

付録2 C言語による自動航行プログラム

```
/* -----
```

```
ソーラーボートプロジェクト 電力制御プログラム
```

```
Yukio Yoshita. 1997. Summer
```

```
このプログラムは1997年8月に開催された浜名湖大会耐久レース用に  
制作したものであり、以下にその主な機能を示す。
```

1. 残量計算
2. 理想電力算出
3. オート/マニュアルモード選択
4. 定電力制御
5. 充電監視機能 etc

```
-----*/
```

```
#include <conio.h>  
#include <stdio.h>  
#include <stdlib.h>  
#include <dos.h>  
#include <string.h>  
#include <ctype.h>
```

```
/*===== 関数プロトタイプ =====*/
```

```
void lcdset(void);  
void startmsg(void);  
void scanjob(void);  
void swjob(void);  
void safetyjob(void);  
void killerror(void);  
void swerror(void);  
void startlcd(void);  
void timejob(void);  
void adjob(void);  
void relayjob(void);  
void battjob(void);  
void dutyjob(void);  
void manujob(void);  
void autojob(void);  
void dajob(void);  
void datajob(void);  
void lcdjob(void);  
void lcd_a(void);  
void lcd_b1(void);  
void lcd_b2(void);  
void lcd_c(void);  
void lcd_d(void);  
void lcd_e(void);  
void lcd_p(void);  
void lcd_f(void);  
void lcd_g(void);  
void lcd_cha(void);
```

```

void lcd_auto(void);
void lcd_time(void);
void Dispjob(void);
void endmsg(void);
void keep1(void);
void keep2(void);
int in(int a, int b);
void out1(int a, int b, int c);
void out2(int a, int b);

/*===== グローバル定義 =====*/

#define IO_BIOS 0x7F
#define locate(x, y) printf("%x1B[%d;%dH", y, x) /* カーソル位置 */
#define CLS printf("%x1B[2J") /* 画面消去 */

/*===== グローバルデータ =====*/

union REGS regs;
union REGS Regs; /* メイン側でのレジスタ変数 */
char string[256];
struct time c; /* 時計読み込み */

long zensin=0, kousin=0, battery=0, bypass=0,
    kill=0, autosw=0, keisoku=0, chager=0,
    t1=0, t2=0, tused=0, tlast=0, tall=3601,
    bv=0, bi=0, pi=0, mi=0, ci=0, temp1=0, temp2=0, vol=0, pv=0,
    obv=0, obi=0, opi=0, omi=0, otemp1=0, otemp2=0, ovol=0, opv=0,
    Bpw=0, gosa=0, dapul=0, output=0, output1=0,
    datamem=1, automem=0, keisan=0,
    j1=0, j2=0, k1=0, k2=0, l=0, l1=0, l2=0, m1=0, m2=0,
    zen=0, kou=0, bat=0, byp=0, rtime=15,
    llda=0, acnt=0, push=5, volmem=0, brake1=0, brake2=0,
    Amax=5000, Vmax=720, Vmin=1800, /*---リミット設定---*/
    shohi=0, yoryo=0, psent=0, lastpsent=100000000,
    risopsent=0, Bz=0,
    Vpw=0, Ppw=0, Mpw=0, Cpw=0, Cpsent=0, Cyoryo=0,
    stlcd=0, endlcd=0,
    farst=0, endtime=0,
    psent1=0, lcdpsent2=0,
    min=0, sec=0, min1=0, min2=0, sec1=0, sec2=0, bin=0;

float V1=0, V2=0, l1=0, l2=0, l3=0, l4=0, lcdpsent1=0;

/*===== 関数定義 =====*/

void lcdset() /*---LCD初期化&リセット---*/
{
    regs.x.ax=0x0003;
    regs.x.bx=0x0300;
    int86(IO_BIOS, &regs, &regs);

    regs.x.ax=0x0003;
    regs.x.bx=0x0400;
    int86(IO_BIOS, &regs, &regs);
}

```

```

void startmsg()          /*---パソコン用スタートメッセージ---*/
{
    locate(14,6);
    printf("*****");
    locate(14,7);
    printf("1997年ソーラーボート「GOLDEN EAGLE97」");
    locate(14,8);
    printf("浜名湖大会耐久レース用電力制御プログラム");
    locate(14,9);
    printf("このプログラムの主な機能は以下のとおりである。");
    locate(14,10);
    printf("残量計算、理想電力算出、オートクルーズ");
    locate(14,11);
    printf("定電力制御、充電監視機能  etc");
    ");
    locate(14,12);
    printf("*****");

    keep1();
    CLS;

    locate(14,7);
    printf("***** 警告 *****");
    locate(14,8);
    printf("なお、プログラム中に printf 文を含んでいる場合は");
    locate(14,9);
    printf("実行するにあたりパソコン本体が必要となります。");
    locate(14,10);
    printf("*****");

    keep1();
    CLS;

    locate(14,7);
    printf("***** 注意 *****");
    locate(14,8);
    printf("また、このプログラムは先に設定された時間内で、");
    locate(14,9);
    printf("バッテリーを使い切ることを目的としたものである。");
    locate(14,10);
    printf("*****");

    keep1();
    CLS;
}

```

```

void scanjob()
{
    long time1=1;
    printf("最大電力を設定します¥n");

    SCAN:

    do{
        printf("0 から 1200 の範囲で数字を入力してください¥n");
    }
}

```

```

        printf("例 255W → 255 [Return] ¥n");
        scanf("%ld",&Pmax);
    } while(0>Pmax||Pmax>1200);

    printf("%ld W でよろしいですか? < Yes / No >",Pmax);

    scanf("%s",&x);
    if (x=='y' ||x=='Y')
        {
            Vmax=Pmax;
            printf("設定値は%ld W に決定しました¥n",Vmax);
        }
    else
        {
            printf("無効な数字が入力されました¥n¥a");
            Amax=0;
            goto SCAN;
        }
    for (time1=1;time1<10000;time1++);
    time1=1;
}

int in(int a, int b)
{
    int c;
    regs.x.ax=a;
    regs.x.bx=b;
    int86(10_BIOS,&regs,&regs);
    c=regs.x.cx;
    return(c);
}

void out1(int a, int b, int c)
{
    regs.x.ax=a;
    regs.x.bx=b;
    regs.x.cx=c;
    int86(10_BIOS,&regs,&regs);
}

void out2(int a, int b)
{
    regs.x.ax=a;
    regs.x.bx=b;
    int86(10_BIOS,&regs,&regs);
}

void swjob() /*---接点入力 CX: ON時 0FFFFh、off時 0000h----*/
{
    zensin=in (0x2b,0x00);
    kousin=in (0x2b,0x01);
    battery=in (0x2b,0x02);
    bypass=in (0x2b,0x03);
    kill=in (0x2b,0x04);
    autosw=in (0x2b,0x05);
    keisoku=in (0x2b,0x06);
}

```

```

        chager=in (0x2b,0x07);
    }

void killerror() /*---@CHECK_KILL_SW!@---*/
{
/*          out ( lcd , ascii, lcd-no); */
    out1 (0x0003, 0x40, 0x0000);
    out1 (0x0003, 0x43, 0x0001);
    out1 (0x0003, 0x48, 0x0002);
    out1 (0x0003, 0x45, 0x0003);
    out1 (0x0003, 0x43, 0x0004);
    out1 (0x0003, 0x4B, 0x0005);
    out1 (0x0003, 0x5F, 0x0006);
    out1 (0x0003, 0x4B, 0x0007);
    out1 (0x0003, 0x49, 0x0008);
    out1 (0x0003, 0x4C, 0x0009);
    out1 (0x0003, 0x4C, 0x000a);
    out1 (0x0003, 0x5F, 0x000b);
    out1 (0x0003, 0x53, 0x000c);
    out1 (0x0003, 0x57, 0x000d);
    out1 (0x0003, 0x21, 0x000e);
    out1 (0x0003, 0x40, 0x000f);
}

void swerror() /*---スイッチ・エラーメッセージ---*/
{
/*---@SWICH ERROR!!!@---*/
    out1 (0x0003, 0x40, 0x0000);
    out1 (0x0003, 0x53, 0x0001);
    out1 (0x0003, 0x57, 0x0002);
    out1 (0x0003, 0x49, 0x0003);
    out1 (0x0003, 0x43, 0x0004);
    out1 (0x0003, 0x48, 0x0005);
    out1 (0x0003, 0x20, 0x0006);
    out1 (0x0003, 0x45, 0x0007);
    out1 (0x0003, 0x52, 0x0008);
    out1 (0x0003, 0x52, 0x0009);
    out1 (0x0003, 0x4f, 0x000a);
    out1 (0x0003, 0x52, 0x000b);
    out1 (0x0003, 0x21, 0x000c);
    out1 (0x0003, 0x21, 0x000d);
    out1 (0x0003, 0x21, 0x000e);
    out1 (0x0003, 0x40, 0x000f);
}

void startlcd()
{
    if(farst==0)
    {
        out1 (0x0003, 0x57, 0x0000);
        out1 (0x0003, 0x45, 0x0001);
        out1 (0x0003, 0x4c, 0x0002);
        out1 (0x0003, 0x4c, 0x0003);
        out1 (0x0003, 0x20, 0x0004);
        out1 (0x0003, 0x43, 0x0005);
        out1 (0x0003, 0x4F, 0x0006);
    }
}

```

```

out1 (0x0003, 0x4d, 0x0007);
out1 (0x0003, 0x45, 0x0008);
out1 (0x0003, 0x20, 0x0009);
out1 (0x0003, 0x54, 0x000a);
out1 (0x0003, 0x4f, 0x000b);
out1 (0x0003, 0x20, 0x000c);
out1 (0x0003, 0x56, 0x000d);
out1 (0x0003, 0x32, 0x000e);
out1 (0x0003, 0x35, 0x000f);

```

```

keep1();
}

```

```

farst=1;
}

```

```

void keep1() /*---時間待ち (lcd表示をラッチを使ってキ-フする)---*/
{
    long keep1;
    for ( keep1=1; keep1<150000; keep1++);
    keep1=1;
}

```

```

void keep2() /*---時間待ち (過渡現象用時間待ち) ---*/
{
    long keep2;
    for ( keep2=1; keep2<2000; keep2++);
    keep2=1;
}

```

```

void timejob()
{
    if(chager!=0 && autosw!=0)
    {tall=3601;
    tused=0;
    lastpsent=100000000;
    }

    regs.x.ax=0x0026;
    regs.x.bx=0x0000;
    int86(10_BIOS, &regs, &regs);
    t1=regs.x.cx; /*---t1は60進数---*/
    if(!(t1==t2))
    {
        tused=tused+1;
        datamem=1;
        keisan=1; /*---データ格納用変数---*/
    }
    t2=t1;
    tlast=tall-tused;
    if(tused==3600)
        {tused=0;}
}

```

```

void adjob()
{
    int i;
}

```

```

for(i=1;i<=10;i++) /*---ad 入力は 10 回分の平均を利用する ---*/
{
    bv=in(0x3e, 0);
    mi=in(0x3e, 1);
    pi=in(0x3e, 2);
    temp1=in(0x3e, 3);
    vol=in(0x3e, 4);
    high=in(0x3e, 5);
    pv=in(0x3e, 6); /*---son---*/
    temp2=in(0x3e, 7);
    obv=obv+bv;
    omi=omi+mi;
    opv=opv+pv;
    opi=opi+pi;
    otemp1=otemp1+temp1;
    otemp2=otemp2+temp2;
    ovol=ovol+vol;
}
bv=obv/10;
mi=omi/10;
pv=opv/10;
pi=opi/10;
temp1=otemp1/10;
temp2=otemp2/10;
vol=ovol/10;
obv=0;
omi=0;
opi=0;
otemp1=0;
ovol=0;
opv=0;
otemp2=0;

mi=1.25*mi; /*---モータ電流---*/
pi=1.25*pi; /*---パネル電流---*/
bi=mi-pi; /*---バッテリー電流---*/
ci=pi-mi-104;
if(ci<0 || battery==0)
{ci=0;}
bv=bv;
vol=vol+1;
high=(75*high)/4095;
temp1=temp1-273;
pv=pv;
temp2=temp2;
if(bi<=0)
{bi=0;}
if(battery==0)
{bv=pv;} /*---パネルのみの時の電圧切り替え---*/
Bpw=(bi*bv)/10000; /*---現在のバッテリー電力---*/
Mpw=(mi*bv)/10000; /*---現在のモータ電力---*/
Ppw=(pi*pv)/10000; /*---現在のパネル電力---*/
Cpw=(ci*bv)/10000; /*---現在の充電電力 ---*/
Vpw=Vmax*vol/4095;
if(temp1>=35)
{out2(0x002f, 0x0008);}

```

```

else
    {out2(0x0030, 0x0008);}
}

void relayjob()/*---V25 のラッチ機能を利用したタイマ-式リレ-プログラ-ム---*/
{
    if(zensin==0 && kousin==0)
    {
        out2(0x0030, 0x0001);           /*---Timer Relay---*/
        out2(0x0030, 0x0002);           /*---Forwad Relay---*/
        out2(0x0030, 0x0003);           /*---Reverse Relay---*/

        zen=0;
        kou=0;
    }
    else if(kousin==0 && !(zensin==0))
    {
        zen=zen+1;
        if(zen==1)
        {
            out2(0x002f, 0x0001);
        }
        if (zen==runtime)
        {
            out2(0x0030, 0x0001);
            zen=1;
        }

        out2(0x002f, 0x0002);
        out2(0x0030, 0x0003);

        kou=0;
    }
    else if(zensin==0 && !(kousin==0))
    {
        kou=kou+1;
        if(kou==1)
        {
            out2(0x002f, 0x0001);
        }
        if(kou==runtime)
        {
            out2(0x0030, 0x0001);
            kou=1;
        }

        out2(0x0030, 0x0002);
        out2(0x002f, 0x0003);

        zen=0;
    }

    if(!(battery==0))
    {
        bat=bat+1;
        if(bat==1)
    }
}

```



```

        {
            out2(0x002f, 0x0006);
        }
        if(bat==runtime)
        {
            out2(0x0030, 0x0006);
            bat=1;
        }

        out2(0x002f, 0x0007);
    }
    else if(battery==0)
    {
        out2(0x0030, 0x0007);
        bat=0;
    }

    if(!(bypass==0))
    {
        byp=byp+1;
        if(byp==1)
        {
            out2(0x002f, 0x0004);
        }
        if(byp==runtime)
        {
            out2(0x0030, 0x0004);
            byp=1;
        }
        out2(0x002f, 0x0005);
    }
    else if(bypass==0)
    {
        out2(0x0030, 0x0005);
        byp=0;
    }
}

void battjob()
{
    if(keisan==1)
    {
        if (bi>=0)
        {
            shohi=Bpw+15.6+9.36 ;
            /*---1秒間の消費電力量:controller+Mppt=(0.65+0.39)*24=24.96W ---*/
            yoryo=1.6817*shohi*shohi-1976.15*shohi+1155442;
            /*その電力での総電力量*/
            psent=100000000/yoryo*shohi;
            /*1秒間の消費率 100000000分
率*/

            lastpsent=lastpsent-psent;

            if(lastpsent<0)
            {lastpsent=0;}

```

```

        risopsent=lastpsent/tlast;
率 */
        Bz=0.00569147*risopsent+52.23;
        keisan=0;
    }
    if(battery!=0 && ci>0)
    {
        ci=ci; /*---V25の消費電流を減算---*/
        Cpw=bv*ci/10000; /*---充電電力---*/
        Cyoryo=1.6817*Cpw*Cpw-1976.15*Cpw+1155442;
        Cpsent=(100000000/Cyoryo*Cpw)*0.9;
率
        /*その電力での総電力量*/
        /*1秒間の充電率 100000000分
        充電
        効率は90%とする */
        lastpsent=lastpsent+Cpsent;
        risopsent=lastpsent/tlast;
率 */
        Bz=0.00569147*risopsent+52.23;
        /*理想電力算出*/

        if(lastpsent>100000000)
        {lastpsent=100000000;}
        ci=ci+65+39;
        Cpw=bv*ci/10000;
        keisan=0;
    }
}

void dutyjob()
{
    if(automem==1)
        goto AT;

    if(autosw==0) /*----- オートクルーズ判定 -----*/
    {
        lcda=0;
        automem=0;
        acnt=0;
        push=5;
        manujob();
        goto NEXT;
    }

    AT:

    acnt=acnt+1;
    if(acnt>1000)

```

```

        acnt=100;
if(acnt==1)      /*----- ボリューム位置保存判定 -----*/
{
        lcda=1;
        automem=1;
        volmem=vol;
}

if(vol>volmem+25 || volmem-25>vol) /*---オート解除判定---*/
        brake1=1;      /*---解除する---*/
else
        brake1=0;      /*---解除しない---*/

if (push==0 && autosw!=0)
        brake2=1;      /*---解除する---*/
else
        brake2=0;      /*---解除しない---*/

push=autosw;

if(brake1==1 || brake2==1)
{
        if(autosw!=0)
        {push=0;}
        else
        {push=5;}

        lcda=0;
        automem=0;
        acnt=0;
        manujob();
        goto NEXT;
}
else /*---オートクルーズモード---*/
{
        lcda=1;
        automem=1;
        Vpw=Bz;
        Mpw=Bpw;
        manujob();
        Mpw=bv*mi/10000;
}

NEXT:

}

void manujob()
{
        if(chager!=0)
                {Vpw=Vpw*1.2;}

        gosa=Mpw-Vpw;

```

```

if(gosa<0)
    {gosa=gosa*(-1);}

if(Mpw<Vpw && gosa>31)
    {dapul=dapul+80;}
else if(Mpw<Vpw && gosa>23)
    {dapul=dapul+64;}
else if(Mpw<Vpw && gosa>17)
    {dapul=dapul+48;}
else if(Mpw<Vpw && gosa>11)
    {dapul=dapul+32;}
else if((Mpw<Vpw && gosa>5) && datamem==1)
    {dapul=dapul+16;}
else if((Mpw<Vpw && gosa>5) && datamem==0)
    {dapul=dapul;}

else if((Mpw>Vpw && gosa>31) || Vpw==0)
    {dapul=dapul-80;}
else if(Mpw>Vpw && gosa>23)
    {dapul=dapul-64;}
else if(Mpw>Vpw && gosa>17)
    {dapul=dapul-48;}
else if(Mpw>Vpw && gosa>11)
    {dapul=dapul-32;}
else if((Mpw>Vpw && gosa>5) && datamem==1)
    {dapul=dapul-16;}
else if((Mpw>Vpw && gosa>5) && datamem==0)
    {dapul=dapul;}

else if(Mpw==Vpw || gosa<=5)
    {dapul=dapul;}

if(dapul>4095)
    {dapul=4095;}
if(dapul<0)
    {dapul=0;}
}

```

```
void autojob()
```

```

{
    if(chager!=0)
        {Bz=Bz*1.2;}

    gosa=Bpw-Bz;

    if(gosa<0)
        {gosa=gosa*(-1);}

    if(Bpw<Bz && gosa>31)
        {dapul=dapul+80;}
    else if(Bpw<Bz && gosa>23)
        {dapul=dapul+64;}
    else if(Bpw<Bz && gosa>17)
        {dapul=dapul+48;}
    else if(Bpw<Bz && gosa>11)
        {dapul=dapul+32;}
}

```

```

else if((Bpw<Bz && gosa>5) && datamem==1)
    {dapul=dapul+16;}
else if((Bpw<Bz && gosa>5) && datamem==0)
    {dapul=dapul;}

else if((Bpw>Bz && gosa>31) || Vpw==0)
    {dapul=dapul-80;}
else if(Bpw>Bz && gosa>23)
    {dapul=dapul-64;}
else if(Bpw>Bz && gosa>17)
    {dapul=dapul-48;}
else if(Bpw>Bz && gosa>11)
    {dapul=dapul-32;}
else if((Bpw>Bz && gosa>5) && datamem==1)
    {dapul=dapul-16;}
else if((Bpw>Bz && gosa>5) && datamem==0)
    {dapul=dapul;}

else if(Bpw==Bz || gosa<=5)
    {dapul=dapul;}

if(dapul>4095)
    {dapul=4095;}
if(dapul<0)
    {dapul=0;}
}

void dajob()
{

if(zensin==0 && kousin==0)
    {dapul=0;}

/*if(bypass!=0)
    {dapul=4095;} */

output=dapul/16;          /*---8ビット 256段階変換---*/
output1=((214*output)/255)+41;

out1(0x17, 0x02, output1);

}

void datajob()
{

psent1=lastpsent/10000;
if(datamem==1 && keisoku!=0)
{

FILE *fp;
if((fp=fopen("b:data", "at"))==NULL)
{
exit(1);
}

fprintf(fp, "a%d ", bv); /*---データの区切りはタブキーを利用---*/
fprintf(fp, "b%d ", bi);
}
}

```

```

fprintf(fp, "c%d", pv);
fprintf(fp, "d%d", pi);
fprintf(fp, "e%d", Vpw);
fprintf(fp, "f%d", Bz);
fprintf(fp, "g%d", output);
fprintf(fp, "h%d ¥n", psent1);
fprintf(fp, "i%d ¥n", temp1);
fprintf(fp, "ne%d ¥n", temp2);
fclose(fp);
}
datamem=0;
}
void Dispjob() /*---各データを表示---*/
{
/* V1=bv/100, V2=pv/100;
I1=bi/100, I2=pi/100, I3=mi/100, I4=ci/100;*/
lcdpsent1=lastpsent/100000;
lcdpsent2=(lastpsent/10000)-(lcdpsent1*100);
min=tlast/60;
sec=tlast%60;

locate(1,1);
printf("'97 浜名湖大会 耐久レース用プログラム HAMANAKO.exe ");
if(automem==1)
{
locate(1,3);
printf("制御モード: マニュアル放電モード マニュアル充電モード *オートクルージングモード* ");
}
if(automem==0)
{
if(battery!=0 && ci>0)
{
locate(1,3);
printf("制御モード: *マニュアル放電モード* マニュアル充電モード オートクルージングモード ");
}
else
{
locate(1,3);
printf("制御モード: *マニュアル放電モード* マニュアル充電モード オートクルージングモード ");
}
}

gettime(&c);
locate(1,5);
printf("現在の時刻 %2d:%02d:%02d残り時間 %ld分%ld秒 ¥n", c.ti_hour, c.ti_min,
c.ti_sec, min, sec);
locate(1,7);
printf("バッテリー電圧 %ld[V] アンペア電圧 %ld[V] ", bv, pv);
locate(1,9);
printf("バッテリー電流 %ld[A] アンペア電流 %ld[A] モーター電流 %ld[A] 充電電流 %ld[A] ", bi, pi, mi, ci);
locate(1,11);
printf("バッテリー電力 %ld[W] アンペア電力 %ld[W] モーター電力 %ld[W] 充電電力 %ld[W] ");
}

```

```

    ", Bpw, Ppw, Mpw, Cpw);
    locate(1, 13);
    printf("設定電力 %ld[W] 理想電力 %ld[W]          ", Vpw, Bz);
    locate(1, 15);
    printf("IC化温度 %ld[°C] 熱電対 %ld[°C]          ", temp1, temp2);
    locate(1, 17);
    printf("デューティ比 %ld[段階] (Max 255)          ", output);
    locate(1, 19);
    printf("バッテリー残量 %3.0f.%ld[%%]          ", lcdpsent1, lcdpsent2);
}

```

```
void lcdjob()
```

```

{
    bin=t1%2;          /*---2進カウンタ、1秒間隔で0と1を繰り返す---*/

    if(automem==0)
    {
        if(battery!=0 && ci>0)          /*---charge mode---*/
        {
            if(datamem==1)
            {
                lcd_a();          /*---Moter Power---*/
                lcd_f();          /*---chage Power---*/

                if(bin==0)
                {lcd_p();          /*---percentage---*/
                  lcd_cha();}      /*---chag---*/
                else
                {lcd_b2();          /*---batt voltage---*/
                  lcd_time();}
                }

            else
            {
                if(datamem==1)
                {
                    lcd_a();          /*---Moter Power---*/
                    lcd_b1();          /*---batt Volt---*/
                    lcd_c();          /*---Moter current---*/
                }
                lcd_d();          /*---Set Power---*/
            }
        }

        if(automem==1)
        {
            if(datamem==1)
            {
                lcd_a();          /*---Moter Power---*/
                lcd_e();          /*---Batt Power---*/
                if(bin==0)
                {lcd_p();          /*---%----*/
                  lcd_auto();      /*---auto---*/ }
                else
            }
        }
    }
}

```

```

        {lcd_b2();
        lcd_time();}
    }
}
void lcd_a()
{
    j1=Mpw/1000;
    j2=j1+0x0030;
    k1=(Mpw-1000*j1)/100;
    k2=k1+0x0030;
    l1=(Mpw-1000*j1-100*k1)/10;
    l2=l1+0x0030;
    m2=Mpw-1000*j1-100*k1-10*l1+0x0030;

    out1(0x0003, k2, 0x0000);
    out1(0x0003, l2, 0x0001);
    out1(0x0003, m2, 0x0002);
    out1(0x0003, 0x57, 0x0003);
}

void lcd_b1()
{
    j1=bv/1000;
    j2=j1+0x0030;
    k1=(bv-1000*j1)/100;
    k2=k1+0x0030;
    l1=(bv-1000*j1-100*k1)/10;
    l2=l1+0x0030;

    out1(0x0003, j2, 0x0004);
    out1(0x0003, k2, 0x0005);
    out1(0x0003, l2, 0x0006);
    out1(0x0003, 0x56, 0x0007);
}

void lcd_b2()
{
    j1=bv/1000;
    j2=j1+0x0030;
    k1=(bv-1000*j1)/100;
    k2=k1+0x0030;
    l1=(bv-1000*j1-100*k1)/10;
    l2=l1+0x0030;

    out1(0x0003, j2, 0x0008);
    out1(0x0003, k2, 0x0009);
    out1(0x0003, l2, 0x000a);
    out1(0x0003, 0x56, 0x000b);
}

void lcd_c()
{
    j1=mi/1000;
    j2=j1+0x0030;
    k1=(mi-1000*j1)/100;
    k2=k1+0x0030;

```



```

        l1=(mi-1000*j1-100*k1)/10;
        l2=l1+0x0030;

        out1(0x0003, j2, 0x0008);
        out1(0x0003, k2, 0x0009);
        out1(0x0003, l2, 0x000a);
        out1(0x0003, 0x41, 0x000b);
    }

void lcd_d()
{
    j1=Vpw/100;
    j2=j1+0x0030;
    k1=(Vpw-100*j1)/10;
    k2=k1+0x0030;
    l1=Vpw-100*j1-10*k1;
    l2=l1+0x0030;

    out1(0x0003, j2, 0x000c);
    out1(0x0003, k2, 0x000d);
    out1(0x0003, l2, 0x000e);
    out1(0x0003, 0x57, 0x000f);
}

void lcd_e()
{
    j1=Bpw/1000;
    j2=j1+0x0030;
    k1=(Bpw-1000*j1)/100;
    k2=k1+0x0030;
    l1=(Bpw-1000*j1-100*k1)/10;
    l2=l1+0x0030;
    m2=Bpw-1000*j1-100*k1-10*l1+0x0030;

    out1(0x0003, k2, 0x0004);
    out1(0x0003, l2, 0x0005);
    out1(0x0003, m2, 0x0006);
    out1(0x0003, 0x57, 0x0007);
}

void lcd_p()
{
    psent1=lastpsent/100000;
    if(psent1>=1000)
    {
        j2=1+0x0030;
        k2=0+0x0030;
        l2=0+0x0030;
    }
    if(psent1>=0 && psent1<1000)
    {
        j1=psent1/100;
        j2=j1+0x0030;
        k1=(psent1-100*j1)/10;
        k2=k1+0x0030;
        l1=psent1-100*j1-10*k1;
        l2=l1+0x0030;
    }
}

```

```

        if(lastpsent==0)                /*--- emp ---*/
        {
            j2=0x45;
            k2=0x4d;
            l2= 0x50;
        }
        out1(0x0003, j2, 0x0008);
        out1(0x0003, k2, 0x0009);
        out1(0x0003, l2, 0x000a);
        out1(0x0003, 0x25, 0x000b);    /*--- % ---*/
    }
void lcd_auto()
    {
        j2=0x41;
        k2=0x55;
        l2=0x54;
        m2=0x4f;

        out1(0x0003, j2, 0x000c);
        out1(0x0003, k2, 0x000d);
        out1(0x0003, l2, 0x000e);
        out1(0x0003, m2, 0x000f);
    }
void lcd_f()
    {
        j1=Cpw/1000;
        j2=j1+0x0030;
        k1=(Cpw-1000*j1)/100;
        k2=k1+0x0030;
        l1=(Cpw-1000*j1-100*k1)/10;
        l2=l1+0x0030;
        m2=Cpw-1000*j1-100*k1-10*l1+0x0030;

        out1(0x0003, k2, 0x0004);
        out1(0x0003, l2, 0x0005);
        out1(0x0003, m2, 0x0006);
        out1(0x0003, 0x57, 0x0007);
    }
void lcd_time()
    {
        min=tlast/60;
        sec=tlast%60;
        min1=min/10+0x0030;
        min2=min%10+0x0030;
        sec1=sec/10+0x0030;
        sec2=sec%10+0x0030;

        out1(0x0003, min1, 0x000c);
        out1(0x0003, min2, 0x000d);
        out1(0x0003, sec1, 0x000e);
        out1(0x0003, sec2, 0x000f);
    }
void lcd_cha()
    {
        j2=0x43;
        k2=0x48;
    }

```

```

    l2=0x41;
    m2=0x47;

    out1(0x0003, j2, 0x000c);
    out1(0x0003, k2, 0x000d);
    out1(0x0003, l2, 0x000e);
    out1(0x0003, m2, 0x000f);
}

void endmsg() /*---終了メッセージ---*/
{
    if (endtime==0)
    {
        out1 (0x0003, 0x40, 0x0000);
        out1 (0x0003, 0x53, 0x0001);
        out1 (0x0003, 0x45, 0x0002);
        out1 (0x0003, 0x45, 0x0003);
        out1 (0x0003, 0x20, 0x0004);
        out1 (0x0003, 0x59, 0x0005);
        out1 (0x0003, 0x4F, 0x0006);
        out1 (0x0003, 0x55, 0x0007);
        out1 (0x0003, 0x20, 0x0008);
        out1 (0x0003, 0x41, 0x0009);
        out1 (0x0003, 0x47, 0x000a);
        out1 (0x0003, 0x41, 0x000b);
        out1 (0x0003, 0x49, 0x000c);
        out1 (0x0003, 0x4e, 0x000d);
        out1 (0x0003, 0x21, 0x000e);
        out1 (0x0003, 0x40, 0x000f);

    }

    keep1();

    endtime=0;
    farst=0;
}

/*-----*/
void main()
{
    CLS; /*--- 初期設定(前面消去) ---*/
    lcdset(); /*--- L C D初期化 ---*/
    startmsg(); /*--- メッセージ表示 (パソコン) ---*/
    scanjob(); /* --- キ-キ'-ト'による最大電力入力---*/

    START:

    swjob(); /*--- 各スイッチの接点入力確認 ---*/

    /*---安全確認---*/

    if((kill==0&&autosw==0)|| (kill==0&&chager==0))
    {

```

```

        zensin=0;
        kousin=0;
        battery=0;
        bypass=0;
        dapul=0;
        relayjob();
        dajob();
        killerror();
        goto START;
    }
    if((zensin==0&&kousin==0)&&!(battery==0)||!(bypass==0))
    {
        swerror();
        goto START;
    }
/*-----安全確認終了---*/

    startlcd();      /*--- メッセージ表示 (LCD) ---*/
    timejob();       /*--- 時間計算 ---*/
    relayjob();      /*--- リレーの動作 ---*/
    adjob();         /*--- A/D入力 ---*/
    battjob();       /*--- 残量計算 ---*/
    dutyjob();       /*--- パルス幅決定 ---*/
    dajob();         /*--- D/A出力 ---*/
    lcdjob();        /*--- LCD表示 ---*/
    Dispjob();       /*--- ディスプレイ表示 ---*/
    datajob();       /*--- データ格納 ---*/
    keep2();         /*--- 過渡反応時間修正 ---*/
    if (kill==0&&!(autosw==0)&&!(chager==0))
    {
        zensin=0;
        kousin=0;
        battery=0;
        bypass=0;
        dapul=0;
        relayjob();
        dajob();
        endmsg();
        lcdset();
        goto END;
    }

    goto START;

END:
CLS;

printf("%n%nプログラムは終了しました\n");
printf("お疲れさまでした\n");
}

```