

รายการอ้างอิง

1. IEEE Transactions on nuclear science. Vol. Ns-35, No.1, february, 1987.
2. Environmental radiation monitoring around tokai reprocessing plant.
environmental protection section, health and safety division, tokai works, power reactor and nuclear fuel development corporation.
3. Lamarsh, John R. Introduction to nuclear engineering. 2nd ed.
California : Addison- Wesley, publishing company, 1983.
4. Nuclear and radiation protection system. berthold, EG&G. nuclear instruments division.
5. Winch, Robert G. Telecommunication transmission systems. New York :
Mc Graw-Hill, C 1993.
6. สุชาติ กังวารจิตต์ม. หลักการทำงานเครื่องรับส่งวิทยุและระบบวิทยุสื่อสาร. พิมพ์ครั้งที่ 1, กรุงเทพฯ, บริษัท ซีเอ็ดดูเคชั่น จำกัด, พ.ศ.2532.
7. Dungan, Frank R. Electronic communications systems. 2nd ed. Delmar publishers inc, 1993.
8. Hall, Douglas V. Digital circuits and systems. New York : Mc Graw-Hill, c1989.
9. National instruments IEEE-488 and VXIbus control data acquisition and analysis, National instruments, 1993.
10. พิพัฒน์ เลหาสงคราม. ไมโครคอนโทรลเลอร์ MCS-48 MCS-51.
พิมพ์ครั้งที่ 2, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พ.ศ. 2537.

ศูนย์วิทยุวิทยุ
จุฬาลงกรณ์มหาวิทยาลัย



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.1

```

*****
;* PROGRAM CHANGE SIGNALS ETC. TO RS-232 *
*****

    ORG 0000H ;RESET START
    JMP START
    ORG 0023H ;SERIAL PROT INTERRUPT
    JMP SERIALS

;
;
PORTA EQU 8101H ;PORT A
PORTB EQU 8102H ;PORT B
PORTC EQU 8103H ;PORT C
CONTR EQU 8100H ;WRITE CONTROL WORD

;
;
NLINE EQU 50H ;NUMBER ANALOG LINE
BCD_C EQU 51H ;COMPARE OLD VALUE
STUW EQU 52H ;WORK STATUS
STAP EQU 40H ;STATUS PORT PI
SELEC EQU 41H ;SELECT SEND , REQUEST
;
;
*****
;* MAIN PROGRAM
*****
START: CLR STAP
      CLR STUW
      CLR BCD_C
      CALL INITS
      CALL RD_ST
EXTT1: MOV A,SELEC
      CJNE A,#0AAH,EXTT2
      CALL TOSE
      JMP EXTT1
EXTT2: MOV A,SELEC
      CJNE A,#55H,START
      CALL RQ-QUEST
      CALL SUB_D
      JMP EXT2

TOSE: CLR A
EXT1: MOV A,STUW
      CJNE A,#08H,EXT2
      MOV DPTR,#0E000H ;ADDRESS IEEE488
      MOVX A,@DPTR ;CALL PROCESS IEEE488
      CALL SEND
      MOV A,STAP
      CLR ACC.7
      MOV STAP,A
      MOV PI,STAP
      JMP EXT1
EXT2: MOV A,STUW
      CJNE A,#04H,EXT3
      MOV DPTR,#0C000H ;ADDRESS BCD_CODE
      MOVX A,@DPTR
      CJNE A,#0BFFH,COMP
      CALL SUB_D
      JMP EXT2
COMP: MOVX A,@DPTR
      CJNE A,BCD_C,P_RD ;COMPARE OLD VALUE
      CALL SUB_D
      JMP EXT2
P_RD: MOVX A,@DPTR

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

MOV BCD_C,A
CALL SEND
CALL SUB_D
MOV A,STAP
CLR ACC.7
MOV STAP,A
MOV P1,STAP
JMP EXT2
EXT3: MOV A,STUW
JNB ACC.7,LINE1
MOV 30H,#31H
CALL ANALOG
LINE1: MOV A,STUW
JNB ACC.6,LINE2
MOV 30H,#32H
CALL ANALOG
LINE2: MOV A,STUW
JNB ACC.5,LINE3
MOV 30H,#33H
CALL ANALOG
CALL SUB_D
LINE3: MOV A,STUW
JNB ACC.4,EXTED
MOV 30H,#34H
CALL ANALOG
EXTED: MOV A,STAP
CLR ACC.7
MOV STAP,A
MOV P1,STAP
CALL T_DELAY
JMP EXT3
RET
:
REQUEST: MOV A,R3
CJNE A,#0FH,REQ1
MOV A,STUW
JNB ACC.7,REQ1
MOV 30H,#31H
CALL ANALOG
CALL SUB_D
REQ1: MOV A,R3
CJNE A,#0FH,REQ2
MOV A,STUW
JNB ACC.6,REQ2
MOV 30H,#32H
CALL ANALOG
CALL SUB_D
REQ2: MOV A,R3
CJNE A,#0FH,REQ3
MOV A,STUW
JNB ACC.5,REQ3
MOV 30H,#33H
CALL ANALOG
CALL SUB_D
REQ3: MOV A,R3
CJNE A,#0FH,REQ4
MOV A,STUW
JNB ACC.4,REQ4
MOV 30H,#34H
CALL ANALOG
CALL SUB_D
REQ4: MOV A,STAP
CLR ACC.7
MOV STAP,A
MOV P1,STAP

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

                                RET
;
;-----
;*          INITIAL SERIALS PORT
;-----
INITS:   CLR          ES
         CLR          ET1
         CLR          TH1
         SETB         TR1
         SETB         ES
         SETB         EA
         RET

B150:   MOV          TMOD,#20H ;timer 1 operate in mode 2
         MOV          TH1,#40H ;code TH1= 40H baud rate = 150 baud
         CLR          ES
         CLR          ET1
         SETB         TR1
         MOV          SCON,#50H ;mode 1 serial port
         SETB         ES
         SETB         EA
         RET

B300:   MOV          TMOD,#20H ;timer 1 operate in mode 2
         MOV          TH1,#0A0H ;code TH1= 0A0H baud rate =300 baud
         CLR          ES
         CLR          ET1
         SETB         TR1
         MOV          SCON,#50H ;mode 1 serial port
         SETB         ES
         SETB         EA
         RET
;
;-----
;*          SERIALS PORT INTERRUPT
;-----
;
SERIALS: CLR          RI
         JNB          RI,#         ;loop to receive
         CLR          RI           ;between receive RI=0 if final receive RI = 1
         CLR          EA
         CLR          ES
         MOV          A,SBUF ;data in to the sbuf
         MOV          R3,A
         SETB         ES
         SETB         EA
         RET

;-----
;*          SERIALS PORT SEND DATA
;-----
;
SEND:   MOV          R7,A
         MOV          A,STAP
         SETB         ACC.7
         MOV          STAP,A
         MOV          P1,STAP
         MOV          A,R7
         CLR          EA
         CLR          ES
         MOV          SBUF,A
         JNB          TI,#
         CLR          TI
         SETB         ES
         SETB         EA


```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

1          RET
-----
1*        READ WORK STATUS PORT
-----
1
RD_ST: CLR      A
          MOV     P1,#00H
          CALL   CTRL
          MOV     DPTR,#PORTA
          MOVX   A,@DPTR
          MOV     STUW,A
          ANL    A,#01H
          CJNE   A,#01H,BAUD1
          CALL   B150
          JMP    READ0
BAUD1: CALL   B300
READ0: MOV     A,STUW
          ANL    A,#02H
          CJNE   A,#02H,READ1
          MOV     SELEC,#55H
          JMP    READ2
READ1: MOV     SELEC,#0AAH
READ2: MOV     DPTR,#PORTB
          MOV     A,#55H
          MOVX   @DPTR,A
          CALL   T_DELAY
          MOV     A,#0AAH
          MOVX   @DPTR,A
          CALL   T_DELAY
          MOV     A,STUW
          ANL    A,#0FH
          MOV     STUW,A
          CJNE   A,#0FH,READ_E
          MOV     A,#40H
          MOVX   @DPTR,A
          MOV     STUW,#00H
READ_E: MOVX   @DPTR,A
          RET
-----
1*        DELAY TIME AND CONTROL PORT
-----
1
T_DELAY: MOV     R7,#04H
DE_2:  MOV     R6,#0FFH
DE_1:  MOV     R5,#0FH
DE_0:  DEC     R5
          CJNE   R5,#00H,DE_0
          DEC     R6
          CJNE   R6,#00H,DE_1
          DEC     R7
          CJNE   R7,#00H,DE_2
          RET
-----
1
SUB_D: MOV     R7,#1FH
SU_D:  DEC     R7
          CJNE   R7,#00H,SU_D
          RET
-----
1
CTRL:  MOV     A,#0FH
          MOV     DPTR,#CNTRL
          MOVX   @DPTR,A
          RET
-----
1

```



 ศูนย์วิทยทรัพยากร
 จุฬาลงกรณ์มหาวิทยาลัย

```

.....
2* SEND ANALOG 1-4 LINE
.....
:
ANALOG: MOV     A,#2EH
              CALL SEND
              MOV  A,#41H
              CALL SEND
              MOV  A,30H
              CALL SEND
              MOV  A,#20H
              CALL SEND
              CALL CTRL
              MOV  DPTR,#PORTC
              MOV  A,30H
              CJNE A,#31H,AN1
              MOV  A,#00H
              MOVX @DPTR,A
AN1:  MOV  A,30H
              CJNE A,#32H,AN2
              MOV  A,#01H
              MOVX @DPTR,A
AN2:  MOV  A,30H
              CJNE A,#33H,AN3
              MOV  A,#02H
              MOVX @DPTR,A
AN3:  MOV  A,30H
              CJNE A,#34H,ANE
              MOV  A,#03H
              MOVX @DPTR,A
ANE:  MOV  DPTR,#0A000H
              MOVX A,@DPTR
              CALL SEND
              RET
:
END

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก.2

```
#include <dos.h>
#include <graphics.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>

#define LNEqual(value,low,high) ((value<low)&&(value!=high))
#define BTW(value,low,high) ((value<low)&&(value>high))
#define CharLong 16
#define CharWidth 8
#define UP 0x4800
#define DOWN 0x5000
#define RIGHT 0x4D00
#define LEFT 0x4B00
#define TAB 0x0F09
#define ENTER 0x1C0D
#define ESC 0x011B
#define MOUSE 0x0000
#define M_KISO 0x0001
#define M_KIS1 0x0002
#define M_KIS2 0x0003
#define M_KIS 0x0004
#define ENG 0x0010
#define THAI 0x0020

void Initial_graph(void);
void Clock(int x,int y);
void port_int(int port, unsigned char code);
void sport(int port,int CH);
void sportX(int port,int CH);
void PrintEg(int x,int y,int color,char *char_size,int len);
void PrintDh(int x,int y,int font L);
void Load_font(void);
void Load_Config(void);
void Show_Menu(void);
void Menu(void);
void PrintTha(int x,int y);
void PrintThaI(int x,int y,int color,char *str,int size);
void Time_Delay(float sec);
void Window(int x,int y,int x1,int y1,int h,int w,int bk);
void Config(void);
void Save_Config(void);
void Int_Plot(void);
void Plot_Graph(void);
void Init_mouse(void);
void Mouse_on(void);
void Mouse_active(void);
void Mouse_off(void);
void Close(int c);
void Read_RS232(int ser);
void ReadI_RS232(int ser);
void Expand(int exp);
int sport(int port);
int Isalpha(unsigned char temp);

FILE *Typ;
unsigned char AlphN[256][16];
unsigned char CONF[4900];
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย


```

unsigned char DATA[1000][10];
unsigned int X[2][L,RR];
unsigned char RS_DAT;
char D_FILE[12]="NONAME.DAT";
char *Sh_menu[4]={"1. Thai or English ", "2. Set Configuration",
                 "3. Show Signal Graph", "4. Exit Program "};
char *Sh_menu[4]={"1. ไทย หรือ อังกฤษ", "2. ตั้งค่าโปรแกรม",
                 "3. แสดงสัญญาณไฟ", "4. ออกโปรแกรม "};

char *T[5]={"0 ", "60 ", "120 ", "180 ", "240 "};
char *Band[11]={"110", "150", "300", "600", "1200", "2400", "4800", "9600"};
char *Vel[10]={"55 km", "60 km", "65 km", "70 km", "75 km",
             "80 km", "85 km", "90 km", "95 km", "100 km"};
char *T_SC[10]={"10 Sec", "15 Sec", "20 Sec", "30 Sec", "1 Min",
              "3 Min", "5 Min", "10 Min", "15 Min", "20 Min"};
char *SET_C[6]={"COM1", "300", "BCDB421", "80 km", "CH0 G 0.1", "5 Sec"};
char *CNEL[10]={"CH0 ", "CH1 ", "CH2 ", "CH3 ", "CH4 ",
              "CH5 ", "CH6 ", "CH7 ", "CH8 ", "CH9 "};
char *GAIN[5]={"G 0.1", "G 0.5", "G 1.0", "G 1.5", "G 2.0"};
char R_Port;

float G[5]={0.1,0.5,1.0,1.5,2.0};
int mouse_x,m_y,kis_m;
int FONT_pos,TIME_SCAN,Y_SCAN_IN;
union REGS inregs,outregs;
int shift,ALA_Ac[10],Gain[10];
float Vahac;
int Sol[80] = { 0xF2,0xF3,0xF4,0x00,0xA7,0xF6,0xF7,0xF8,
              0xF9,0xC7,0xA2,0xF3,0xB3,0xA8,0x2F,0x47,
              0x32,0xC0,0xB0,0xD6,0xD6,0xA4,0xB5,0xAB,
              0xC7,0xB2,0xA5,0xCC,0xC6,0xF1,0xC4,0xDA,
              0xA9,0xAF,0xAF,0xE7,0xAC,0xF7,0xB3,0xEB,
              0xC9,0xC8,0xF0,0xF0,0xCF,0xAD,0xF0,0xB1,
              0xA6,0xB8,0xEA,0xC7,0x34,0x29,0x22,0x28,
              0xBA,0xA3,0xC5,0xD9,0xF8,0x1F,0xBF,0xD4,
              0xE1,0xA1,0xD3,0xB4,0xE0,0xE9,0xC3,0xEB,
              0xD2,0xCA,0xB7,0xD7,0xB9,0xC2,0xE6,0xBE,
              0xCB,0xD0,0xD5,0xCD,0xE4,0xBB,0xD1,0xBC,
              0xB0,0xA5};

main()
{
    Load_font();
    Load_config();
    Initial_graph(); FONT=ENG; ALA =0;
    Init_mouse();
    Menu();
    Mouse_click();
    Closegraph();
    printf("Good Bye!\n");

    /* Initial Graph Mode */
void Initial_graph()
{ int GraphDriver = DETECT;
  int ErrorCode =0;
  int GraphMode;
  inigraph(&GraphDriver,&GraphMode,"C:\\LANG\\BGI");
  ErrorCode = graphresult();
  if (ErrorCode != gOK)
  { printf("No Graphics error %s",grapherrmsg(ErrorCode));
    exit(1); }
}

/* Prints English Character */
void PrintFig(int x,int y,int color,char *ch,int size,int fon)
{ int a,b,c;
  if(size>4)size=4;
}

```



```

for(b=1;b<=3;b++)
{
    if(b==1){a=0;c=1;}
    if(b==2){a=15;c=3;}
    if(b==3){a=color;c=2;}
    setcolor(a);
    settextjustify(1,1);
    settextstyle(font,0,size);
    outtextxy(x-c,y-c,ch);
}

/* Load font bit map of character */
void Load_Font()
{
    FILE *fp;
    char *filename = "fhai.fon";
    if(!fp=fopen(filename,"rb"))==NULL)
    {
        printf("Font not found"); exit(1);
    }
    fread(AlphN,3,sizeof AlphN,fp);
    fclose(fp);
}

void Load_Config()
{
    int i;
    FILE *fp;
    char *filename = "GConfig.232";
    if(!fp=fopen(filename,"rb"))==NULL)
    {
        printf("File Config(GCONFIG.232) not found"); exit(1);
    }
    fread(CONF,3,sizeof CONF,fp);
    fclose(fp); Ac[0] = 1;
    Ac[1]=Ac[2]=Ac[3]=Ac[4]=Ac[5]=Ac[6]=Ac[7]=Ac[8]=Ac[9]=0;
    for(i=0;i<9;i++) G[i]=CONF[i]-4890;
}

void Save_Config()
{
    FILE *fp;
    int ps,i,j,Line,row;
    int p,mu;
    unsigned int S0,S1,Color;
    ps=20;
    for(Line=0;Line<10;Line++)
        for(row=0;row<27;row++)
            for(j=184; j<322;j++)
            {
                p=S0+0;
                for(i=374;i<382;i++)
                {
                    if((p==0)mu=1);if((p==1)mu=2);if((p==2)mu=4);if((p==3)mu=6);
                    if((p==4)mu=16);if((p==5)mu=32);if((p==6)mu=64);if((p==7)mu=128);
                    Color=getpixel(i+8*row,j+35*Line);
                    if(Color==15){ S1=S0+mu; S0=S1; j++; }
                    CONF[ps]=S0; ps++; }
                for(i=0;i<9;i++) CONF[i+4890]=G[i];
            }
    fp=fopen("GConfig.232","wb");
    for(i=0;i<4899;i++)
        fprintf(fp,"%c",CONF[i]);
    fclose(fp); FONT=ENG;
}

void PrintH(int x,int y,int h,m L)
{
    int a,b,ps,Line,i,j,row,cp,color;
    unsigned int code; ps=20; cp=1;
    for(Line=0;Line<10;Line++)
    {
        if(L==13){ ps=20+486*L; Line=L; } if(Line<3){cp=5;
        for(row=0;row<27;row++)
            for(i=0;i<18;i++)

```

มหาวิทยาลัยแพทย์
จุฬาลงกรณ์มหาวิทยาลัย

```

code=CONFps); a=code; ps++;
color=cp+Lanc; if(x==274)color=13;
for(j=0;j<CharWidth;j++)
{ b=a%2; if(b==1)
{ a=(a-1)/2;
putpixel(now*CharWidth+x*,i+y*Lanc*b,color);
}
if(b==0) a=a/2;
}
}
if(L!=13)goto Li;
} Li;
/* Print string to graphic mode */
void PrintThai1(int x,int y,int color,char *str,int size)
{
register int i,j,ASCII;
unsigned char code;
int a,b,c,d,i_chr,shift;
int StrL; shift=-1;
StrL = strlen(str);
for(i=0;i<StrL;i++)
{ ASCII=str[i];
if(ASCII+1) ASCII=256+ASCII;
if(Isalpha(ASCII)&&size==2) shift+=2;
if(Isalpha(ASCII)&&size==1) shift++;
for(j=0;j<CharLong;j++)
{
if(size==2){
code=AlphaN[ASCII][i]; a=code;
for(j=CharWidth*2;j>=0;j--2)
{ b=a%2; if(b==1)
{ a=(a-1)/2;
putpixel(x+j*shift*CharWidth,y+i*2,color);
putpixel(x+j*shift*CharWidth,y+i*2+1,color);
putpixel(x+j*shift*CharWidth+1,y+i*2,color);
}
else a=a/2;
}
}
if(size==1){
code=AlphaN[ASCII][i]; a=code;
for(j=CharWidth;j>=0;j--1)
{ b=a%2; if(b==1)
{ a=(a-1)/2;
putpixel(x+j*shift*CharWidth,y+i,color);
}
else a=a/2;
}
}
}
}
/* Print string to graphic mode */
void PrintThai(int x,int y)
{
register int i,j,ASCII;
unsigned char code;
char ch;
int a,b,color; i=shift=j=0;
do {
ch = getch();
if(ch==0x60||ch==0x7E) goto Pass1;
if(ch==0x0D||shift==25)goto Pass;
if(ch=='M') setfillstyle(1,1);

```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

    bar(x+10+shift*8,y+16,x+shift*8,y-2); if(shift!=0) shift--; goto Pass;
ASCII=ch;
if(FONT=="THAI"&&ASCII!=0x20)ASCII = Set(ASCII-35);
if(FONT=="ENG")ASCII = ch;
if(ASCII+1) ASCII=256+ASCII;
if(ASCII!=0xDB&&ASCII!=0xD9){=1;
if(Isalpha(ASCII)) shift++; 1=0;
if(!Isalpha(ASCII)&&shift==0) sound(800);delay(100);
nosound(); goto Pass; }
for(i=0;i<CharLong;i++)
{
code=AlphaN[ASCII][i]; a=code; color=15;
for(j=CharWidth;j>0;j--)
{ b=a%2; if(b==1)
{ a=(a-1)/2;
putpixel(shift*CharWidth+j,ry-1,code);
if((b==0) a=a/2;
}
}
} goto Pass;
Pass1: Window(210,440,270,465,3,1,1);
if(FONT=="ENG");
Printg(240,450,1,"THAI",1,1);
Time_Delay(0.2); FONT="THAI";
Window(210,440,270,465,3,0,1);
Printg(240,450,3,"THAI",1,1); goto Pass;
if(FONT=="THAI");
Printg(245,450,0,"ENG",1,1);
Time_Delay(0.2); FONT="ENG";
Window(210,440,270,465,3,0,1);
Printg(245,450,1,"ENG",1,1); goto Pass; }
Pass:
while(ch!=0x0D);
/* Check argument is alpha or not */
int Isalpha(unsigned char temp)
{
int bo1,bo2;
bo1 = LNEqual(temp,0x04,0xD1);
bo2 = BTW(temp,0xD9,0x17)&&(temp!=0xDB);
return((bo1|bo2|temp==0xED));
}
/* Time Delay Second Unit */
void Time_Delay(float sec)
{
float start_Time,end_Time,rel_Time;
start_Time = clock();
do
{
end_Time = clock();
rel_Time = (end_Time - start_Time) / CLK_TCK;
}
while(rel_Time<sec);
}
void Window(int x,int y,int x1,int y1,int hin,int wh,int bk)
{
int i,c1,c2,c3;
if(bk==0){c1=1;c2=11;c3=9;}
if(bk==1){c1=8;c2=15;c3=7;}
if(bk==2){c1=4;c2=13;c3=5;}
if(bk==3){c1=12;c2=6;c3=4;}
if(bk==4){c1=10;c2=2;c3=3;}
if(bk==5){c1=1;c2=9;c3=3;}
setlinestyle(0,0,1);
for(i=0;i<hin;i++)
{
if(i%2==1) setcolor(c1); else setcolor(c2);
line(x+1,y+i,x+1,y1-1);
line(x1-1,y+i,x1-1,y1);
}
}

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

        if(wh==0) setcolor(c1); else setcolor(c2);
        line(x1-l,y1-l,x1-l,y+1);
        line(x+l,y1-l,x1-l,y1-1);
        setfillstyle(1,c3);
        bar(x+hn,y+hn,x1-hn,y1-hn);
        setcolor(0);
        rectangle(x,y,x1,y1);
        if(hk==1&&wh==1)
        { setcolor(15);
          line(x,y,x,y1-1); line(x,y,x1-1,y);
        }
    }

/* Initial Menu */
void Show_Menu()
{ int i;
  Window(10,10,630,470,3,1,1);
  setfillstyle(1,3);
  bar(20,20,getmaxx()-20,getmaxy()-20);
  Window(160,250,480,460,2,0,0);
  Window(180,280,460,450,3,1,1);
  for(i=0;i<4;i++)
  { Window(200,300+i*35,440,330+i*35,2,0,1);
    if(FONT=="ENG")PrintEg(getmaxx()/2,i*35+315,6,Sh_menu[i],1,1);
    if(FONT=="THAI")PrintThaI(230,i*35+308,15,Sh_menu[i],1,1);
    if(FONT=="THAI")PrintThaI(230,i*35+310,0,Sh_menu[i],1,1);
    if(FONT=="THAI")PrintThaI(230,i*35+309,4,Sh_menu[i],1,1);
  }
  setcolor(10);
  rectangle(199,299+pos*35,441,331+pos*35);
  if(FONT=="ENG")
  PrintEg(getmaxx()/2,50,12,"Environmental doserati increasing",3,1);
  PrintEg(getmaxx()/2,80,12,"Program for NT 2612 interfacing system",3,1);
  PrintEg(getmaxx()/2,160,9,"Department of Nuclear Technology",3,1);
  PrintEg(getmaxx()/2,190,9,"Faculty of Engineering Chulalongkorn University",3,1);

  if(FONT=="THAI")
  PrintThaI(60,55,15,"โปรแกรมตรวจวัดปริมาณรังสีในสิ่งแวดล้อม",2);
  PrintThaI(60,57,15,"โปรแกรมตรวจวัดปริมาณรังสีในสิ่งแวดล้อม",2);
  PrintThaI(60,56,12,"โปรแกรมตรวจวัดปริมาณรังสีในสิ่งแวดล้อม",2);
  PrintThaI(75,90,15,"สำหรับระบบเชื่อมไฮดรอลิก NT 2612",2);
  PrintThaI(75,92,15,"สำหรับระบบเชื่อมไฮดรอลิก NT 2612",2);
  PrintThaI(75,91,12,"สำหรับระบบเชื่อมไฮดรอลิก NT 2612",2);
  PrintThaI(150,155,15,"ภาควิชาวิศวกรรมเครื่องกล",2);
  PrintThaI(150,157,15,"ภาควิชาวิศวกรรมเครื่องกล",2);
  PrintThaI(150,156,9,"ภาควิชาวิศวกรรมเครื่องกล",2);
  PrintThaI(50,190,15,"คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย",2);
  PrintThaI(50,192,15,"คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย",2);
  PrintThaI(50,191,9,"คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย",2);
}

/* Show Main Menu for select function */
void Menu()
{ int i,key,m_act; m_act=0; pos=0;
  Show_Menu();
  do
  { while(!kbhit()) Mouse_active(); if(mouse==0) Mouse_on();
    /* mouse active */
    if(m_x>200&& m_x<440&& m_y>300&& m_y<330&& kks_m==1)
    { key=MOUSE;pos=0;if(m_act==pos)key+ENTER; goto mm;}
    if(m_x>200&& m_x<440&& m_y>335&& m_y<365&& kks_m==1)
    { key=MOUSE;pos=1;if(m_act==pos)key+ENTER; goto mm;}
    if(m_x>200&& m_x<440&& m_y>370&& m_y<400&& kks_m==1)
    { key=MOUSE;pos=2;if(m_act==pos)key+ENTER; goto mm;}
    if(m_x>200&& m_x<440&& m_y>405&& m_y<435&& kks_m==1)
    { key=MOUSE;pos=3;if(m_act==pos)key+ENTER; goto mm;}
  }
}

```

```

key=brskkey(0); mm: Mouse_onf();
switch(key)
{ case UP: setcolor(7);
rectangle(199,299+pos*35,441,331+pos*35);
pos = (pos==0) ? 3 : --pos; delay(300);
setcolor(10);
rectangle(199,299+pos*35,441,331+pos*35);
break;
case DOWN: setcolor(7);
rectangle(199,299+pos*35,441,331+pos*35);
pos = (pos==3) ? 0 : ++pos; delay(300);
setcolor(10);
rectangle(199,299+pos*35,441,331+pos*35);
break;
case MOUSE: setcolor(7); for(j=0;(j<4);j++)
rectangle(199,299+pos*35,441,331+pos*35);
setcolor(10); m_act=pos;
rectangle(199,299+pos*35,441,331+pos*35);
break;
case ENTER:
Window(200,300+pos*35,440,330+pos*35,2,1,1);
if(FONT==ENG) PrintEg(getmaxx()/2,pos*35+315,6,Sh_menu[pos],1,1);
Time_Delay(0.3);
Window(200,300+pos*35,440,330+pos*35,2,0,1);
if(FONT==THAI)
if(FONT==THAI)PrintTha1(230,pos*35+308,15,Sh_menu[pos],1);
if(FONT==THAI)PrintTha1(230,pos*35+310,0,Sh_menu[pos],1);
if(FONT==THAI)PrintTha1(230,pos*35+309,4,Sh_menu[pos],1);
}
if(FONT==ENG) PrintEg(getmaxx()/2,pos*35+315,6,Sh_menu[pos],1,1);
Time_Delay(0.5);
if(pos==0) Close(0); if(FONT==THAI)FONT=ENG; else FONT=THAI;
if(pos==1) Close(0);Config(2);Close(0);
if(pos==2) Close(0);Plot_Graphic();Close(0);
if(pos==3) key = ESC; goto Exit;
Show_Menu(); Exit;
break;
default: break;
} while(key!=ESC);
Close(0);
void Close(int cp)
{ int c;
setcolor(cp); setlinestyle(0,0,1);
for(i=0;(i<640);i+=2)
{ line(i,0,i,480);
line(639-i,0,639-i,480); delay(10); }
}
void Init_mouse()
{ int x,y;
union REGS regs;
regs.x.ax=0;
int86(0x33,&regs,&regs);
}
void Mouse_onf()
{ union REGS regs;
regs.x.ax=1;
int86(0x33,&regs,&regs);
mouse=1;
}
void Mouse_active()
{ union REGS regs;

```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

regs.ax=3;
int86(0x33,&regs,&regs);
kv_m=regs.ebx & 3;
m_x=regs.eax;
m_y=regs.edi;

void Mouse_off()
{
union REGS regs;
regs.ax=2;
int86(0x33,&regs,&regs);
mouse=0;
}

void Config()
{
char *sh_conf[6]={"PORT","BAUDRATE","S_CODE",
"ALARM_Lc","CALIBRATE",
"TIME_SCAN"};

int i,key0,pos0,cp,Bp,Tp,Ch,Ga,Pa,A;
for(i=1; i<6; i++) | if(CONF[i]>0)pos0=CONF[i];
if(CONF[1]==0x01)SET_C[0]="COM.2";
SET_CT[1]=Baud[CONF[2]];
if(CONF[3]==0x01)SET_C[2]="ASCII";
SET_CT[3]=Per[CONF[4]];
SET_CT[5]=T_SC[CONF[6]];
for(i=0; i<11; i++)
D_FILE[i]=CONF[7+i];
Bp=CONF[2];Pa=CONF[4];Ch=0;Ga=0;Tp=CONF[6];
Window(2,2,638,478,3,1,1);
CONF[i]=setfillstyle(1,3);key0=pos0=0;
bar(10,10,getmaxx()-10,getmaxy()-10);
if(FONT=="ENG")
| Printg(getmaxx()/2,25,13,"Set Configuration",0,1);
Printg(140,80,3,"Set Communication",2,1);
Printg(480,80,3,"EDF Station",2,1);
if(FONT=="THAI")
PrintThai(180,20,15,"จัดรูปแบบระบบ",2);
PrintThai(180,23,1,"จัดรูปแบบระบบ",2);
PrintThai(180,21,9,"จัดรูปแบบระบบ",2);
PrintThai(100,90,15,"ระบบสื่อสาร",1);
PrintThai(100,92,1,"ระบบสื่อสาร",1);
PrintThai(100,91,13,"ระบบสื่อสาร",1);
PrintThai(400,90,15,"สถานีตรวจวัด",1);
PrintThai(400,92,1,"สถานีตรวจวัด",1);
PrintThai(400,91,13,"สถานีตรวจวัด",1);
|
for(i=0; i<6; i++)
| Window(20,120+45*i,120,150+45*i,2,1,2);
Window(140,120+45*i,250,150+45*i,2,0,0);
Window(145,123+45*i,245,147+45*i,2,1,0);
Window(123,120+45*i,135,133+45*i,3,0,1);
Window(123,137+45*i,135,150+45*i,3,0,1);
Printg(73,135+45*i,2,sh_conf[i],1,1);
Printg(200,135+45*i,2,SET_CT[i],1,1); |
setcolor(15);
settextstyle(DEFAULT_FONT,0,1); outtextxy(550,100,"C.F.");
cp=1;
settextstyle(DEFAULT_FONT,0,1); setcolor(14);
outtextxy(412,465,"Inactive"); outtextxy(535,465,"Active");
Window(370,461,378,470,3,0,1); Window(500,461,508,470,3,0,0);
for(i=0; i<10; i++)
| Window(330,115+35*i,345,130+35*i,3,0,2);
if(Ac[i]==0) Window(320,118+35*i,328,127+35*i,3,0,1);
if(Ac[i]==1) Window(320,118+35*i,328,127+35*i,3,0,0);
Window(350,110+35*i,600,135+35*i,2,1,1);
if(i<3)cp=5; setfillstyle(1,cp+i);


```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

bar(355,120+i*35,360,125+i*35);
Window(30,440,90,465,3,0,1); PrintEg(60,450,2,"Close",1,1);
Window(120,440,180,465,3,0,1);PrintEg(150,450,4,"SAVE",1,1);
Window(210,440,270,465,3,0,1);
PrintEg(245,450,9,"ENG",1,1);
setcolor(4); outtextxy(75,400,"DATA FILE");
setfillstyle(1,0); bar(125,390,260,415); setcolor(9);
settextjustify(0,0); outtextxy(134,407,D.FILE);
PrintE(374,114,35,13);
do
| while(!kbhit()) Mouse_active(); if(mouse==0) Mouse_on();
/* mouse active */
if(m_x>330&& m_x<345&& m_y>115&& m_y<130&& kism_m==1)
| key0=M_KISO;pos0=0; goto mm0;
if(m_x>330&& m_x<345&& m_y>150&& m_y<165&& kism_m==1)
| key0=M_KISO;pos0=1; goto mm0;
if(m_x>330&& m_x<345&& m_y>185&& m_y<200&& kism_m==1)
| key0=M_KISO;pos0=2; goto mm0;
if(m_x>330&& m_x<345&& m_y>220&& m_y<235&& kism_m==1)
| key0=M_KISO;pos0=3; goto mm0;
if(m_x>330&& m_x<345&& m_y>255&& m_y<270&& kism_m==1)
| key0=M_KISO;pos0=4; goto mm0;
if(m_x>330&& m_x<345&& m_y>290&& m_y<305&& kism_m==1)
| key0=M_KISO;pos0=5; goto mm0;
if(m_x>330&& m_x<345&& m_y>325&& m_y<340&& kism_m==1)
| key0=M_KISO;pos0=6; goto mm0;
if(m_x>330&& m_x<345&& m_y>360&& m_y<375&& kism_m==1)
| key0=M_KISO;pos0=7; goto mm0;
if(m_x>330&& m_x<345&& m_y>395&& m_y<410&& kism_m==1)
| key0=M_KISO;pos0=8; goto mm0;
if(m_x>330&& m_x<345&& m_y>430&& m_y>445&& kism_m==1)
| key0=M_KISO;pos0=9; goto mm0;
if(m_x>30&& m_x<90&& m_y>440&& m_y<465&& kism_m==1)
| key0=M_KIS1;pos0=10; goto mm0;
if(m_x>120&& m_x<180&& m_y>440&& m_y>465&& kism_m==1)
| key0=M_KIS1;pos0=11; goto mm0;
if(m_x>210&& m_x<270&& m_y>440&& m_y>465&& kism_m==1)
| key0=M_KIS1;pos0=12; goto mm0;
if(m_x>123&& m_x<135&& m_y>120&& m_y<133&& kism_m==1)
| key0=M_KIS2;pos0=13; goto mm0;
if(m_x>123&& m_x<135&& m_y>165&& m_y<178&& kism_m==1)
| key0=M_KIS2;pos0=14; goto mm0;
if(m_x>123&& m_x<135&& m_y>210&& m_y>223&& kism_m==1)
| key0=M_KIS2;pos0=15; goto mm0;
if(m_x>123&& m_x<135&& m_y>255&& m_y>268&& kism_m==1)
| key0=M_KIS2;pos0=16; goto mm0;
if(m_x>123&& m_x<135&& m_y>300&& m_y>313&& kism_m==1)
| key0=M_KIS2;pos0=17; goto mm0;
if(m_x>123&& m_x<135&& m_y>345&& m_y>358&& kism_m==1)
| key0=M_KIS2;pos0=18; goto mm0;
if(m_x>123&& m_x<135&& m_y>137&& m_y<150&& kism_m==1)
| key0=M_KIS2;pos0=19; goto mm0;
if(m_x>123&& m_x<135&& m_y>182&& m_y<195&& kism_m==1)
| key0=M_KIS2;pos0=20; goto mm0;
if(m_x>123&& m_x<135&& m_y>227&& m_y>240&& kism_m==1)
| key0=M_KIS2;pos0=21; goto mm0;
if(m_x>123&& m_x<135&& m_y>272&& m_y>285&& kism_m==1)
| key0=M_KIS2;pos0=22; goto mm0;
if(m_x>123&& m_x<135&& m_y>317&& m_y>330&& kism_m==1)
| key0=M_KIS2;pos0=23; goto mm0;
if(m_x>123&& m_x<135&& m_y>362&& m_y>375&& kism_m==1)
| key0=M_KIS2;pos0=24; goto mm0;
if(m_x>30&& m_x<115&& m_y>395&& m_y>405&& kism_m==1)
| key0=M_KIS2;pos0=25; goto mm0;
for(i=0;i<10;i++)

```



ศูนย์วิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย


```

if(m_x>320&& m_x<328&& m_y>118+35*i&& m_y<127+35*i&& k& m==1)
| key0=M_KIS;pos0=26+i; goto mm0; |

| key0 = bmskey(0); mm0: Mouse_off();
| switch(key0)
| case M_KIS: A = pos0-26;
| if(A<pos0-26)A=0; Ac[A]=1;
| Window(320,118+35*A,328,127+35*A,3,0,0); |
| else | Ac[A]=0;
| Window(320,118+35*A,328,127+35*A,3,0,1); | Time_Delay(0.2);
| break;
| case M_KIS0: Window(330,115+35*pos0,345,130+35*pos0,3,1,2);
| setfillstyle(1,1);
| bar(365,115+35*pos0,595,132+35*pos0);
| Print(365,116+35*pos0);
| Time_Delay(0.2);
| Window(330,115+35*pos0,345,130+35*pos0,3,0,2);
| break;
| case M_KIS1: if(pos0==10)|
| Window(30,440,90,465,3,1,1);
| Print(60,450,10,"Close",1,1);
| Time_Delay(0.2); key0=ESC;
| Window(30,440,90,465,3,0,1);
| Print(60,450,2,"Close",1,1);
| if(pos0==11)|
| Window(120,440,180,465,3,1,1);
| Print(150,450,12,"SAVE",1,1);
| CONF[2]-=Bp;CONF[4]-=Pu;CONF[6]-=Tp;
| for(i=0;i<11;i++)
| CONF[7+i]=D_FILE[i];
| Time_Delay(0.2); Save_Config();
| Window(120,440,180,465,3,0,1);
| Print(150,450,4,"SAVE",1,1); |
| if(pos0==12)|
| Window(210,440,270,465,3,1,1);
| if(FONT==ENG);
| Print(240,450,11,"THAI",1,1);
| Time_Delay(0.2); FONT=THAI;
| Window(210,440,270,465,3,0,1);
| Print(240,450,3,"THAI",1,1); goto Non; |
| if(FONT==THAI)|
| Print(245,450,9,"ENG",1,1);
| Time_Delay(0.2); FONT=ENG;
| Window(210,440,270,465,3,0,1);
| Print(245,450,1,"ENG",1,1); |
| Non: Time_Delay(0.1);
| break;
| case M_KIS2: if(pos0==13)| SET_C[0]="COM.1"; CONF[1]=0; |
| if(pos0==19)| SET_C[0]="COM.2"; CONF[1]=1; |
| if(pos0==14)| Bp = (Bp==7) ? 7 : ++Bp; SET_C[1]=Baud[Bp];
| if(pos0==20)| Bp = (Bp==0) ? 0 : --Bp; SET_C[1]=Baud[Bp];
| if(pos0==15)| SET_C[2]="ASCII"; CONF[3]=1;
| if(pos0==21)| SET_C[2]="BCDB421";CONF[3]=0;
| if(pos0==16)| Pu = (Pu==9) ? 9 : ++Pu; SET_C[3]=Par[Pu];
| if(pos0==22)| Pu = (Pu==0) ? 0 : --Pu; SET_C[3]=Par[Pu];
| if(pos0==17)| Ch = (Ch==9) ? 0 : ++Ch;
| strcpy(SET_C[4],CNEL[Ch]); strcat(SET_C[4],GAIN[Ga]); |
| if(pos0==23)| Ga = (Ga==4) ? 0 : ++Ga; Gain[Ch]=Ga;
| strcpy(SET_C[4],CNEL[Ch]); strcat(SET_C[4],GAIN[Ga]); |
| if(pos0==18)| Tp = (Tp==9) ? 9 : ++Tp; SET_C[5]=T_SC[1p];
| if(pos0==24)| Tp = (Tp==0) ? 0 : --Tp; SET_C[5]=T_SC[1p];
| if(pos0==25)| RET: setfillstyle(1,0); bar(125,390,260,415);
| for(A=0;A<11;A++) D_FILE[A]=0x20; A=0;
| do| D_FILE[A] = getch(); | setcolor(12);
| if(D_FILE[A]=="\n")goto RET; | settextjustify(0,0);

```

ศูนย์วิทยาศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

```

        outnxtxy(134,409,D_FILE); A++;
        while(D_FILE[A-1]!='\r'&&A<11; D_FILE[A-1]=0x00;pos0=0;
            if(pos0>19)
                Window(123,120+45*(pos0-13),135,133+45*(pos0-13),3,1,1);
                Window(145,123+45*(pos0-13),245,147+45*(pos0-13),2,1,0);
                Time_Delay(0.2);
                PrintIg(200,135+45*(pos0-13),4,SET_C[pos0-13],1,1);
                Window(123,120+45*(pos0-13),135,133+45*(pos0-13),3,0,1);
                if(pos0>18)
                    Window(123,137+45*(pos0-19),135,150+45*(pos0-19),3,1,1);
                    Window(145,123+45*(pos0-19),245,147+45*(pos0-19),2,1,0);
                    Time_Delay(0.2);
                    PrintIg(200,135+45*(pos0-19),4,SET_C[pos0-19],1,1);
                setfillstyle(1,7); bar(530,115+35*Ch,580,130+35*Ch);
                settextstyle(DEFAULT_FONT,0,1); setcolor(15);
                outnxtxy(550,125+35*Ch,GAIN[Ga]);
                Window(123,137+45*(pos0-19),135,150+45*(pos0-19),3,0,1);
                default; break;
            }
        } while(key0!='ESC');
}

void Int_Plot()
{
    int i,cp;
    int STATUS_BAUD[8]= {0x03,0x23,0x43,0x63,0x83,0xA3,0xC3,0xE3};
    char *percent[11]=" 0%", " 10%", " 20%", " 30%", " 40%",
        " 50%", " 60%", " 70%", " 80%", " 90%", "100%";
        Window(2,2,688,478,3,1,1);
        setfillstyle(1,3); bar(10,10,getmaxx()-10,getmaxy()-10);
        Window(20,30,getmaxx()-20,80,2,0,0);Window(20,100,260,450,2,0,0);
        if(FONT=="ENG")
            PrintIg(getmaxx()/2,50,13,"Environmental Doserate",6,1);
            PrintIg(140,115,3,"Stations",2,1);
        if(FONT=="THAI")
            PrintThai(170,40,14,"ปริมาณรังสีในสิ่งแวดล้อม",2);
            PrintThai(170,43,0,"ปริมาณรังสีในสิ่งแวดล้อม",2);
            PrintThai(170,41,12,"ปริมาณรังสีในสิ่งแวดล้อม",2);
            PrintThai(80,110,10,"สถานีตรวจวัด",1);
        setcolor(15); settextstyle(1,1);
        settextstyle(DEFAULT_FONT,0,1); outnxtxy(225,125, "[C,F,I]");
        Window(27,135,253,440,2,1,2); cp=1; Y=346; X[0]=X[1]=378;
        if(RR!=0)Y=346+RR*10;
        for(i=0;i<10;i++)
            Window(30,140+30*i,250,165+30*i,2,0,1); if(i>3)cp=0;
        setfillstyle(1,cp+i); bar(35,150+i*30,40,155+i*30);
        Window(285,140,620,420,1,1,1);
        Window(290,145,615,415,3,1,2);
        setfillstyle(1,0);bar(295,150,610,410);
        Window(295,108,440,132,2,0,0); PrintIg(325,120,4,"Time",1,1);
        Window(465,108,620,132,2,0,0); PrintIg(495,120,3,"Data",1,1);
        settextstyle(DEFAULT_FONT,0,1); setcolor(14); outnxtxy(575,295,"Message");
        Window(330,440,390,465,3,0,1); PrintIg(360,452,1,"STOP",1,1);
        Window(420,440,480,465,3,0,1);PrintIg(450,452,4,"Close",1,1);
        Window(510,440,570,465,3,0,1); PrintIg(540,452,3,"Exp.",1,1);
        setcolor(14); rectangle(569,154,601,169);
        rectangle(479,154,511,169);
        Window(570,155,600,168,2,0,1);
        Window(480,155,510,168,2,0,1);
        setcolor(15); line(335,175,335,385); line(330,380,590,380);
        settextstyle(0,0,1); outnxtxy(570,370,"Time");
        msvph(324,155,15);msvph(325,155,15);
        msvph(590,112,11);msvph(591,112,11);
        outnxtxy(430,395,"TIME SCAN"); setcolor(4);
        outnxtxy(495,162,"PRN"); outnxtxy(585,162,"ACK");
        setcolor(15);

```

ศูนย์วิทยาศาสตร์สุขภาพ
จุฬาลงกรณ์มหาวิทยาลัย

```

for(i=0;i<201;i+=4)
{ if(i%20==0) line(332,380-i,335,380-i);
outtextxy(318,380-i,percent[i]/20); }
else line(334,380-i,335,380-i); }
for(i=0;i<20;i++)
line(345+10*i,380,345+10*i,382);
setcolor(14); outtextxy(500,395,T_SC[CONF[6]]);
Print(45.143,30,13);
setcolor(10); rectangle(29,139,251,166);
setcolor(14); rectangle(329,439,391,466);
setcolor(15); setlinestyle(1,2,1);
line(337,270-CONF[4]*10,600,270-CONF[4]*10);settextstyle(DEFAULT_FONT,0,1);
outtextxy(580,260-CONF[4]*10,"Alarm");
setlinestyle(0,0,1);

port_init(CONF[11],STATUS_BAUD[CONF[2]]);

void Plot_Graph()
{
int i,pos1,key1,cp,pos2,pos3;
float st,ct,n;
int Time[10] = {10,15,20,30,60,180,300,600,900,1200};
int STATUS_BAUD[8]= {0x03,0x23,0x43,0x63,0x83,0xA3,0xC3,0xE3};

Typ = fopen(D_FILE,"wb");
setfillstyle(1,0); SCAN_IN=pos1+st+pos2+pos3+0;
TIME_SCAN = Time[CONF[6]]; RR = L + 0;
Int_Plot();
st = 0;
do
{ while(!kbhit()) Clock(45.8); Mouse_active(); if(mouse==0) Mouse_on();
Alarm_sound(ALA);
ct = clock(); st = [ct - st] / CLK_TCK ;
if(n-TIME_SCAN&&SCAN_IN==1)
if(L==11)L++; if(L==11){ st=clock();
Y+=10; L=1; RR++;}
if(SCAN_IN==1&&A[L-1]==1)
key1 = M_KIS; pos1 = L-1;
goto mm1; } }
/* mouse active */
for(i=0;i<10;i++)
if(m_x>200&&m_x<250&&m_y>140+30*i&&m_y<166+30*i&&ks_m==1)
key1=M_KIN;pos1=i; goto mm1; }
for(i=0;i<2;i++)
if(m_x>330+*90&&m_x<390+*90&&m_y>440&&m_y<465&&ks_m==1)
key1=M_KIS0;pos2=i; goto mm1; }
if(m_x>570&&m_x<600&&m_y>155&&m_y<170&&ks_m==1)
key1=M_KIS1;pos3=0; goto mm1; }
if(m_x>480&&m_x<510&&m_y>155&&m_y<170&&ks_m==1)
key1=M_KIS1;pos3=1; goto mm1; }

} key1 = bioskey(0);
mm1: Mouse_off();
switch(key1)
{ case M_KIS1: setcolor(2);
if(pos3 == 0 )rectangle(569,154,601,169);
if(pos3 == 1 )rectangle(479,154,511,169);
Time_Delay(0.3);
setcolor(14);
if(pos3 == 0 )(ALA=0; rectangle(569,154,601,169);)
if(pos3 == 1 )( rectangle(479,154,511,169);)
break;
case M_KIS: setcolor(5);
for(i=0;i<10;i++)
rectangle(29,139+i*30,251,166+i*30);

```

```

setcolor(10);
rectangle(29,139+pos1*30,251,166+pos1*30);
setcolor(15); setlinestyle(1,2,1);
line(337,270-CONF[4]*10,600,270-CONF[4]*10);settextstyle(DEFAULT_FONT,0,1);
outtextxy(580,260-CONF[4]*10,"Alarm");
setlinestyle(0,0,1);
if(kis_m!=1) sport(CONF[1],L-1);
break;
case M_KISO: setcolor(0);
for(i=0;i<3;i++)
rectangle(329+i*90,439,391+i*90,466);
delay(200); setcolor(14);
rectangle(329+pos2*90,439,391+pos2*90,466);
if(pos2==0)
Window(330,440,390,465,3,1,1);
Time_Delay(0.2);
port_uart(CONF[1],STATUS_BAUD[CONF[2]]);
if(SCAN_IN==0) Time_Delay(0.2);
Window(330,440,390,465,3,0,1);
PrintEg(360,452,2,"SCAN",1,1);
SCAN_IN=1; goto Stop;
if(SCAN_IN==1) Time_Delay(0.2);
Window(330,440,390,465,3,0,1);
PrintEg(360,452,1,"STOP",1,1); SCAN_IN=0;
Stop: if(pos2==1)
Window(420,440,480,465,3,1,1);
PrintEg(450,452,12,"Close",1,1); Time_Delay(0.2);
Window(420,440,480,465,3,0,1); key1=ESC;
PrintEg(450,452,4,"Close",1,1);
if(pos2==2)
Window(510,440,570,465,3,1,1);
PrintEg(540,452,11,"Exp.",1,1); Time_Delay(0.2);
Window(510,440,570,465,3,0,1);
PrintEg(540,452,3,"Exp.",1,1);
Expand(pos1); Int_Plot(); Extend(T1);
if(SCAN_IN==1)
Window(330,440,390,465,3,0,1);
PrintEg(360,452,2,"SCAN",1,1);
}
break;
default: break;
} while(key1!="ESC"); fclose(fyp);
}
void Expand(int exp)
{ int exp[key4];
char percent[11]={"0","10","20","30","40",
"50","60","70","80","90","100"};
Window(2,2,638,478,3,1,1);
setfillstyle(1,3); bar(10,10,getmaxx()-10,getmaxy()-10);
Window(25,35,625,425,3,1,2); Window(30,40,620,420,1,1,1);
setfillstyle(1,0); bar(34,44,616,416);
setcolor(15); line(80,70,80,400); line(80,380,560,380);
settextstyle(DEFAULT_FONT,0,1); Y = 101;
for(i=0;i<300;i+=6)
{ if(i%30==0) line(70,380-i,79,380-i);
outtextxy(55,380-i,percent[i/30]);
else line(75,380-i,79,380-i);
}
for(i=0;i<21;i++)
line(80+20*i,380,80+20*i,383);
outtextxy(550,370,"Time");
msvph(70,50,15);msvph(71,50,15);
outtextxy(250,395,"TIME SCAN");
setcolor(14); outtextxy(320,395,1_SCAN[CONF[6]]);

```

ศูนย์วิทยทรัพยากร

จุฬาลงกรณ์มหาวิทยาลัย

```

PrintEg(560,195,4,"Time",1,1);
Window(30,440,250,465,2,0,1);
PrintFh(50,440-exp*30,30,exp);
if(exp>3)kp = exp+5; else cp = exp+1;
setfillstyle(1,fp); bar(35,450,40,455);
Window(530,440,600,465,3,0,1);
PrintEg(565,452,1,"Close",1,1);
Extend(exp);
do
{
while(!kbhit()) Clock(67,14); Mouse_active();
if(mouse==0) Mouse_off();
/* mouse active */
if(m_x+530&&m_x+600&&m_y+440&&m_y+465&&kis_m==1)
{
key4=ESC; goto mm3;
}
key4=buskey(0); mm3: Mouse_off();
}while(key4!=ESC);
Window(530,440,600,465,3,1,1);
PrintEg(565,452,9,"Close",1,1); Time_Delay(0,2);
Window(530,440,600,465,3,0,1);
PrintEg(565,452,1,"Close",1,1); Mouse_active();
}
}
void sport(int port,int CH)
{
union REGS r;
int YYu,JJ;
float web,s1,c1;
int CD[10] = {0xF0,0xF1,0xF2,0xF3,0xF4,0xF5,0xF6,0xF7,0xF8,0xF9};
JJ = 0;
r.x.ax = port;
r.h.al = CD[CH];
r.h.ah = 0x01;
int86(0x14,&r,&r);
r.x.ax = port;
r.h.al = CD[CH];
r.h.ah = 0x01;
int86(0x14,&r,&r);
r.x.ax = port;
r.h.al = 0x0D;
r.h.ah = 0x01;
int86(0x14,&r,&r);
if(r.h.ah & 128) { settextjustify(L,1); setcolor(12);
settextstyle(DEFAULT_FONT,0,1);
outtextxy(470,200,"Send Error Detected.");
outtextxy(470,220,"Please Set Configuration again");
outtextxy(470,240,"and Connected Serial Port");
outtextxy(470,260,"to Packet Radio");
s1 = clock(); YYu = 0;
do { c1 = clock(); web = (c1 - s1) / CLK_TCK;
while(check_status(CONF[1])&256) { R_Port = point(CONF[1]);
putc(R_Port,Typ); RS_DAT=R_Port;
if(R_Port==' ') gotoxy(70,20); YYu=0; } YYu++;
if(YYu==3)JJ = R_Port-0x30; printf("%c",R_Port);
if(YYu==5) { DATA[RR][JJ]=R_Port; Read_RS232(JJ);
Value = DATA[RR][JJ]*.4*G[Gain[CH]];
if(Value<=(CONF[4]*5+50)HALA==1)HALA = 1;
gotoxy(66,8); printf(" %5.2f ",Value); } }
}while(web>5);
}
}
int sport(int port)
{
union REGS r;
r.x.ax = port;
r.h.ah = 0x02;
int86(0x14,&r,&r);
}
}

```

คณะวิศวกรรมศาสตร์
 จุฬาลงกรณ์มหาวิทยาลัย

```

        if(r.hah & 128)
            settextstyle(1,1); setcolor(12);
        outtextxy(470,300,"Read Error Detected.");
        return(r.hah);
    }

    check_stat(port)
    int port;
    union REGS r;
    r.x.ax = port;
    r.h.ah = 0x02;
    int86(0x14,&r,&r);
    return(r.x.ax);
}

void port_init(int port, unsigned char code)
{ union REGS r;
  r.x.ax = port;
  r.h.ah = 0x00;
  r.h.al = code;
  int86(0x14,&r,&r);
}

void Read_RS232(int ser)
{ int Pz[10] = {1,2,3,4,9,10,11,12,13,14};
  Value = abs(DATA[RR][ser]*.78*G[Gain[ser]]); X[1]=378-(int)Value;
  if(RR!=0) Value = abs(DATA[RR-1][ser]*.78*G[Gain[ser]]); X[0] = 378-(int)Value;
  setcolor(Pz[ser]);
  line(Y-10,X[0],Y,X[1]);
  if(Value<(CONF[4]*5*50))rectangle(Y-2,X[1]-2,Y*2,X[1]+2);

  if(Y==546)SHIFT();
}

Extend(int ny)
{ int Pz[10] = {1,2,3,4,9,10,11,12,13,14};
  int r,yy,i;
  if(ny==11) { yy=10; r=0; if(RR==20)r = RR-19; X[1]=X[0]+378;
  do {
    Value = abs(DATA[r][ny]*1.18*G[Gain[ny]]); X[1]=378-(int)Value;
    if(r!=0) Value = abs(DATA[r-1][ny]*1.18*G[Gain[ny]]); X[0] = 378-(int)Value;
    setcolor(Pz[ny]);
    line(yy-20,X[0],yy,X[1]);
    if(Value<(CONF[4]*5*50))rectangle(yy-2,X[1]-2,yy*2,X[1]+2);
    r++; yy+=20;
  } while(r==RR-1);

  if(ny==11) { yy=346; r=0; if(RR==20)r = RR-19; X[1]=X[0]+378;
  for(i=0;i=10;i++)
  do {
    Value = abs(DATA[r][i]*.78*G[Gain[i]]); X[1]=378-(int)Value;
    if(r!=0) Value = abs(DATA[r-1][i]*.78*G[Gain[i]]); X[0] = 378-(int)Value;
    setcolor(Pz[i]);
    if(Ac[i]==1)line(yy-10,X[0],yy,X[1]);
    if(Value<(CONF[4]*5*50))rectangle(yy-2,X[1]-2,yy*2,X[1]+2);
    r++; yy+=10;
  } while(r==RR-1);
}

SHIFT()
{ int i,j,ZI;
  setfillstyle(1,0); bar(336,170,346,379);
  for(j=346;j=547;j++)
  for(i=170;i=379;i++)
  { ZI = getpixel(j,i);
    if(ZI!=0) putpixel(j,i,0); putpixel(j-10,i,ZI);
  }
  Y=536;
}

```

ศูนย์วิทยุทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```

void Clock(int x,int y)
{
  intregs.h.ab=0x2C;
  intregs.h.ch=0x2D;
  intregs.h.ci=0x2E;
  intregs.h.dh=0x2F;
  intregs.h.eh=0x30;
  intregs.h.fh=0x31;
  intregs.h.gi=0x32;
  intregs.h.gj=0x33;
  intregs.h.gk=0x34;
  intregs.h.gl=0x35;
  intregs.h.gm=0x36;
  intregs.h.gn=0x37;
  intregs.h.go=0x38;
  intregs.h.gp=0x39;
  intregs.h.gq=0x3A;
  intregs.h.gr=0x3B;
  intregs.h.gs=0x3C;
  intregs.h.gt=0x3D;
  intregs.h.gu=0x3E;
  intregs.h.gv=0x3F;
  intregs.h.gw=0x40;
  intregs.h.gx=0x41;
  intregs.h.gy=0x42;
  intregs.h.gz=0x43;
  intregs.h.ga=0x44;
  intregs.h.gb=0x45;
  intregs.h.gc=0x46;
  intregs.h.gd=0x47;
  intregs.h.ge=0x48;
  intregs.h.gf=0x49;
  intregs.h.gg=0x4A;
  intregs.h.gh=0x4B;
  intregs.h.gi=0x4C;
  intregs.h.gj=0x4D;
  intregs.h.gk=0x4E;
  intregs.h.gl=0x4F;
  intregs.h.gm=0x50;
  intregs.h.gn=0x51;
  intregs.h.go=0x52;
  intregs.h.gp=0x53;
  intregs.h.gq=0x54;
  intregs.h.gr=0x55;
  intregs.h.gs=0x56;
  intregs.h.gt=0x57;
  intregs.h.gu=0x58;
  intregs.h.gv=0x59;
  intregs.h.gw=0x5A;
  intregs.h.gx=0x5B;
  intregs.h.gy=0x5C;
  intregs.h.gz=0x5D;
  intregs.h.ga=0x5E;
  intregs.h.gb=0x5F;
  intregs.h.gc=0x60;
  intregs.h.gd=0x61;
  intregs.h.ge=0x62;
  intregs.h.gf=0x63;
  intregs.h.gg=0x64;
  intregs.h.gh=0x65;
  intregs.h.gi=0x66;
  intregs.h.gj=0x67;
  intregs.h.gk=0x68;
  intregs.h.gl=0x69;
  intregs.h.gm=0x6A;
  intregs.h.gn=0x6B;
  intregs.h.go=0x6C;
  intregs.h.gp=0x6D;
  intregs.h.gq=0x6E;
  intregs.h.gr=0x6F;
  intregs.h.gs=0x70;
  intregs.h.gt=0x71;
  intregs.h.gu=0x72;
  intregs.h.gv=0x73;
  intregs.h.gw=0x74;
  intregs.h.gx=0x75;
  intregs.h.gy=0x76;
  intregs.h.gz=0x77;
  intregs.h.ga=0x78;
  intregs.h.gb=0x79;
  intregs.h.gc=0x7A;
  intregs.h.gd=0x7B;
  intregs.h.ge=0x7C;
  intregs.h.gf=0x7D;
  intregs.h.gg=0x7E;
  intregs.h.gh=0x7F;
  intregs.h.gi=0x80;
  intregs.h.gj=0x81;
  intregs.h.gk=0x82;
  intregs.h.gl=0x83;
  intregs.h.gm=0x84;
  intregs.h.gn=0x85;
  intregs.h.go=0x86;
  intregs.h.gp=0x87;
  intregs.h.gq=0x88;
  intregs.h.gr=0x89;
  intregs.h.gs=0x8A;
  intregs.h.gt=0x8B;
  intregs.h.gu=0x8C;
  intregs.h.gv=0x8D;
  intregs.h.gw=0x8E;
  intregs.h.gx=0x8F;
  intregs.h.gy=0x90;
  intregs.h.gz=0x91;
  intregs.h.ga=0x92;
  intregs.h.gb=0x93;
  intregs.h.gc=0x94;
  intregs.h.gd=0x95;
  intregs.h.ge=0x96;
  intregs.h.gf=0x97;
  intregs.h.gg=0x98;
  intregs.h.gh=0x99;
  intregs.h.gi=0x9A;
  intregs.h.gj=0x9B;
  intregs.h.gk=0x9C;
  intregs.h.gl=0x9D;
  intregs.h.gm=0x9E;
  intregs.h.gn=0x9F;
  intregs.h.go=0xA0;
  intregs.h.gp=0xA1;
  intregs.h.gq=0xA2;
  intregs.h.gr=0xA3;
  intregs.h.gs=0xA4;
  intregs.h.gt=0xA5;
  intregs.h.gu=0xA6;
  intregs.h.gv=0xA7;
  intregs.h.gw=0xA8;
  intregs.h.gx=0xA9;
  intregs.h.gy=0xAA;
  intregs.h.gz=0xAB;
  intregs.h.ga=0xAC;
  intregs.h.gb=0xAD;
  intregs.h.gc=0xAE;
  intregs.h.gd=0xAF;
  intregs.h.ge=0xB0;
  intregs.h.gf=0xB1;
  intregs.h.gg=0xB2;
  intregs.h.gh=0xB3;
  intregs.h.gi=0xB4;
  intregs.h.gj=0xB5;
  intregs.h.gk=0xB6;
  intregs.h.gl=0xB7;
  intregs.h.gm=0xB8;
  intregs.h.gn=0xB9;
  intregs.h.go=0xBA;
  intregs.h.gp=0xBB;
  intregs.h.gq=0xBC;
  intregs.h.gr=0xBD;
  intregs.h.gs=0xBE;
  intregs.h.gt=0xBF;
  intregs.h.gu=0xC0;
  intregs.h.gv=0xC1;
  intregs.h.gw=0xC2;
  intregs.h.gx=0xC3;
  intregs.h.gy=0xC4;
  intregs.h.gz=0xC5;
  intregs.h.ga=0xC6;
  intregs.h.gb=0xC7;
  intregs.h.gc=0xC8;
  intregs.h.gd=0xC9;
  intregs.h.ge=0xCA;
  intregs.h.gf=0xCB;
  intregs.h.gg=0xCC;
  intregs.h.gh=0xCD;
  intregs.h.gi=0xCE;
  intregs.h.gj=0xCF;
  intregs.h.gk=0xD0;
  intregs.h.gl=0xD1;
  intregs.h.gm=0xD2;
  intregs.h.gn=0xD3;
  intregs.h.go=0xD4;
  intregs.h.gp=0xD5;
  intregs.h.gq=0xD6;
  intregs.h.gr=0xD7;
  intregs.h.gs=0xD8;
  intregs.h.gt=0xD9;
  intregs.h.gu=0xDA;
  intregs.h.gv=0xDB;
  intregs.h.gw=0xDC;
  intregs.h.gx=0xDD;
  intregs.h.gy=0xDE;
  intregs.h.gz=0xDF;
  intregs.h.ga=0xE0;
  intregs.h.gb=0xE1;
  intregs.h.gc=0xE2;
  intregs.h.gd=0xE3;
  intregs.h.ge=0xE4;
  intregs.h.gf=0xE5;
  intregs.h.gg=0xE6;
  intregs.h.gh=0xE7;
  intregs.h.gi=0xE8;
  intregs.h.gj=0xE9;
  intregs.h.gk=0xEA;
  intregs.h.gl=0xEB;
  intregs.h.gm=0xEC;
  intregs.h.gn=0xED;
  intregs.h.go=0xEE;
  intregs.h.gp=0xEF;
  intregs.h.gq=0xF0;
  intregs.h.gr=0xF1;
  intregs.h.gs=0xF2;
  intregs.h.gt=0xF3;
  intregs.h.gu=0xF4;
  intregs.h.gv=0xF5;
  intregs.h.gw=0xF6;
  intregs.h.gx=0xF7;
  intregs.h.gy=0xF8;
  intregs.h.gz=0xF9;
  intregs.h.ga=0xFA;
  intregs.h.gb=0xFB;
  intregs.h.gc=0xFC;
  intregs.h.gd=0xFD;
  intregs.h.ge=0xFE;
  intregs.h.gf=0xFF;
}

movepht(int x,int y,int color)
{
  unsigned int MSVPH[18]= { 0x01C0,0x0220,0x0200,0x0180,0x4849,
                             0x4829,0x4829,0x7426,0x41C2,
                             0x8000,0x3FFF,0x0000,0x0200,0x0200,
                             0x0380,0x0240,0x0240,0x0240 };
  unsigned int i,j,a,b;
  for(i=0;i<18;i++)
  {
    a = MSVPH[i];
    for(j=21;j>0;j--)
    {
      b=a/2; if(b==1)
      {
        a=(a-1)/2;
        putpixel(x+j,i+y,color);
      }
      if(b==0) a=a/2;
    }
  }
}

Alarm_sound(int ala)
{
  if(ala==1) sound(500); delay(1000); nosound();
  else nosound();
}

```



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



80C51BH/80C51BH-2 CHMOS SINGLE COMPONENT 8-BIT MICROCONTROLLER with Factory Mask-Programmable ROM

80C31BH/80C31BH-2 CHMOS SINGLE COMPONENT 8-BIT CONTROL-ORIENTED CPU WITH RAM AND I/O

80C51BH/80C31BH — 3.5 to 12 MHz $V_{CC} = 5V \pm 20\%$

80C51BH-2/80C31BH-2 — 0.5 to 12 MHz $V_{CC} = 5V \pm 20\%$

- 128 X 8 RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- 64K Program Memory Space
- Boolean Processor
- 5 Interrupt Sources
- Programmable Serial Port
- 64K Data Memory Space

The MCS[®]-51 CHMOS products are fabricated on Intel's advanced CHMOS III process and are functionally compatible with the standard MCS-51 HMOS and EPROM products. CHMOS III is a technology which combines the high speed and density characteristics of HMOS with the low power attributes of CMOS. This combination expands the effectiveness of the powerful MCS-51 architecture and instruction set.

Like the MCS-51 EPROM and HMOS, the MCS-51 CHMOS products have the following features: 4K of ROM (80C51BH/80C51BH-2 only); 128 bytes of RAM; 32 I/O lines; two 16-bit timer/counters; a five-source two-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuitry. In addition, the MCS-51 CHMOS products have two software selectable modes of reduced activity for further power reduction — Idle and Power Down.

Idle mode freezes the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. Power Down mode saves the RAM contents but freezes the oscillator causing all other chip functions to be inoperative.

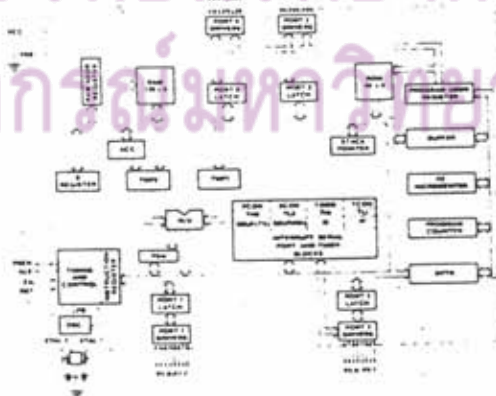


Figure 1. Block Diagram



80C51BH/80C51BH-2
80C31BH/80C31BH-2

ADVANCE INFORMATION

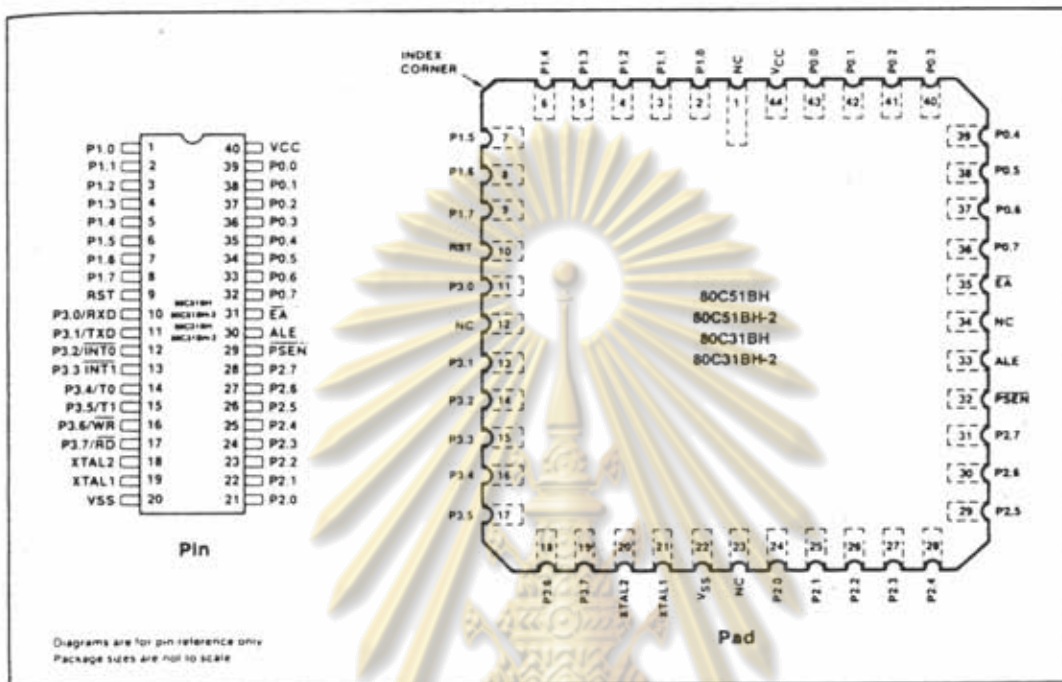


Figure 2. Configurations

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation freezes the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is halted.

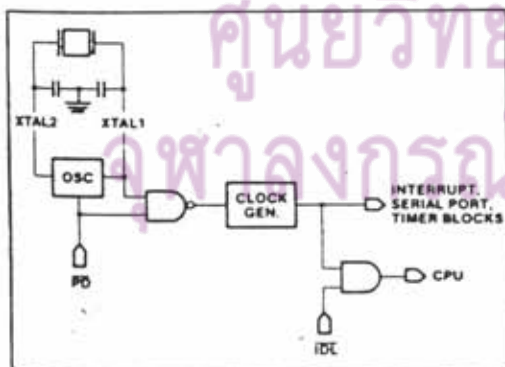


Figure 3. Idle and Power Down Hardware

These special modes are activated by software via the Special Function Register, PCON. Its hardware address is 87H. PCON is not bit addressable.

PCON: Power Control Register

(MSB)							(LSB)
SMOD	—	—	—	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
—	PCON.6	(Reserved)
—	PCON.5	(Reserved)
—	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX0000).

MCS²-51 EXPRESS

PRELIMINARY

Table 1 — Prefix Identification

Prefix	Package Type	Temperature Range	Burn-In
P	plastic	commercial	no
D	cerdip	commercial	no
C	ceramic	commercial	no
TP	plastic	extended	no
TD	cerdip	extended	no
TC	ceramic	extended	no
QP	plastic	commercial	yes
QD	cerdip	commercial	yes
QC	ceramic	commercial	yes
LP	plastic	extended	yes
LD	cerdip	extended	yes
LC	ceramic	extended	yes

Please note:

- Commercial temperature range is 0° to 70°C. Extended temperature range is -40° to +85°C.
- Burn-in is dynamic, for a minimum time of 160 hours at 125°C, $V_{CC} = 5.5V \pm 0.5V$, following guidelines in MIL-STD-883 Method 1015 (Test Condition D).
- The following devices are not available in plastic packages:
8751H, 8751H
- The following devices are not available in ceramic packages:
8051AH, 8031AH
8052AH, 8032AH

Examples: P8031AH indicates 8031AH in a plastic package and specified for commercial temperature range, without burn-in. LD8751H indicates 8751H in a cerdip package and specified for extended temperature range with burn-in.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



PRELIMINARY

**8031AH/8051AH
8032AH/8052AH
8751H/8751H**

EXPRESS

- Extended Temperature Range
- Burn-in

The Intel EXPRESS system offers enhancements to the operational specifications of the MCS[®]-51 family of microcontrollers. These EXPRESS products are designed to meet the needs of those applications whose operating requirements exceed commercial standards.

The EXPRESS program includes the commercial standard temperature range with burn-in, and an extended temperature range with or without burn-in.

With the commercial standard temperature range operational characteristics are guaranteed over the temperature range of 0°C to 70°C. With the extended temperature range option, operational characteristics are guaranteed over the range of -40°C to +85°C.

The optional burn-in is dynamic, for a minimum time of 160 hours at 125°C with $V_{CC} = 5.5V \pm 0.5V$, following guidelines in MIL-STD-883, Method 1015.

Package types and EXPRESS versions are identified by a one- or two-letter prefix to the part number. The prefixes are listed in Table 1.

For the extended temperature range option, this data sheet specifies the parameters which deviate from their commercial temperature range limits. The commercial temperature range data sheets are applicable for all parameters not listed here.

Electrical Deviations from Commercial Specifications for Extended Temperature Range

D.C. and A.C. parameters not included here are the same as in the commercial temperature range data sheets.

D.C. CHARACTERISTICS: ($T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$; $V_{CC} = 5V \pm 10\%$; $V_{SS} = 0V$)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.75	V	
V_{IH}	Input High Voltage (Except XTAL2, RST)	2.1	$V_{CC} + 0.5$	V	
I_{CC}	Power Supply Current: 8051AH,8031AH 8052AH,8032AH 8751H,8751H		135 175 265	ma ma ma	All Outputs Disconnected; $E\bar{A} = V_{CC}$



80C51BH/80C51BH-2
80C31BH/80C31BH-2

ADVANCE INFORMATION

Table 3. Instruction Opcodes in Hexidecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4	E6	1	MOV	A,@R0
CD	1	XCH	A,R5	E7	1	MOV	A,@R1
CE	1	XCH	A,R6	E8	1	MOV	A,R0
CF	1	XCH	A,R7	E9	1	MOV	A,R1
D0	2	POP	data addr	EA	1	MOV	A,R2
D1	2	ACALL	code addr	EB	1	MOV	A,R3
D2	2	SETB	bit addr	EC	1	MOV	A,R4
D3	1	SETB	C	ED	1	MOV	A,R5
D4	1	DA	A	EE	1	MOV	A,R6
D5	3	DJNZ	data addr,code addr	EF	1	MOV	A,R7
D6	1	XCHD	A,@R0	F0	1	MOVX	@DPTR,A
D7	1	XCHD	A,@R1	F1	2	ACALL	code addr
D8	2	DJNZ	R0,code addr	F2	1	MOVX	@R0,A
D9	2	DJNZ	R1,code addr	F3	1	MOVX	@R1,A
DA	2	DJNZ	R2,code addr	F4	1	CPL	A
DB	2	DJNZ	R3,code addr	F5	2	MOV	data addr,A
DC	2	DJNZ	R4,code addr	F6	1	MOV	@R0,A
DD	2	DJNZ	R5,code addr	F7	1	MOV	@R1,A
DE	2	DJNZ	R6,code addr	F8	1	MOV	R0,A
DF	2	DJNZ	R7,code addr	F9	1	MOV	R1,A
E0	1	MOVX	A,@DPTR	FA	1	MOV	R2,A
E1	2	AJMP	code addr	FB	1	MOV	R3,A
E2	1	MOVX	A,@R0	FC	1	MOV	R4,A
E3	1	MOVX	A,@R1	FD	1	MOV	R5,A
E4	1	CLR	A	FE	1	MOV	R6,A
E5	2	MOV	A,data addr	FF	1	MOV	R7,A

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



80C51BH/80C51BH-2
80C31BH/80C31BH-2

ADVANCE INFORMATION

Table 3. Instruction Opcodes in Hexadecimal Order (Continued)

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0	99	1	SUBB	A,R1
67	1	XRL	A,@R1	9A	1	SUBB	A,R2
68	1	XRL	A,R0	9B	1	SUBB	A,R3
69	1	XRL	A,R1	9C	1	SUBB	A,R4
6A	1	XRL	A,R2	9D	1	SUBB	A,R5
6B	1	XRL	A,R3	9E	1	SUBB	A,R6
6C	1	XRL	A,R4	9F	1	SUBB	A,R7
6D	1	XRL	A,R5	A0	2	ORL	C,/bit addr
6E	1	XRL	A,R6	A1	2	AJMP	code addr
6F	1	XRL	A,R7	A2	2	MOV	C,bit addr
70	2	JNZ	code addr	A3	1	INC	DPTR
71	2	ACALL	code addr	A4	1	MUL	AB
72	2	ORL	C,bit addr	A5		reserved	
73	1	JMP	@A+DPTR	A6	2	MOV	@R0,data addr
74	2	MOV	A,#data	A7	2	MOV	@R1,data addr
75	3	MOV	data addr,#data	A8	2	MOV	R0,data addr
76	2	MOV	@R0,#data	A9	2	MOV	R1,data addr
77	2	MOV	@R1,#data	AA	2	MOV	R2,data addr
78	2	MOV	R0,#data	AB	2	MOV	R3,data addr
79	2	MOV	R1,#data	AC	2	MOV	R4,data addr
7A	2	MOV	R2,#data	AD	2	MOV	R5,data addr
7B	2	MOV	R3,#data	AE	2	MOV	R6,data addr
7C	2	MOV	R4,#data	AF	2	MOV	R7,data addr
7D	2	MOV	R5,#data	B0	2	ANL	C,/bit addr
7E	2	MOV	R6,#data	B1	2	ACALL	code addr
7F	2	MOV	R7,#data	B2	2	CPL	bit addr
80	2	SJMP	code addr	B3	1	CPL	C
81	2	AJMP	code addr	B4	3	CJNE	A,#data,code addr
82	2	ANL	C,bit addr	B5	3	CJNE	A,data addr,code addr
83	1	MOVC	A,@A+PC	B6	3	CJNE	@R0,#data,code addr
84	1	DIV	AB	B7	3	CJNE	@R1,#data,code addr
85	3	MOV	data addr, data addr	B8	3	CJNE	R0,#data,code addr
86	2	MOV	data addr,@R0	B9	3	CJNE	R1,#data,code addr
87	2	MOV	data addr,@R1	BA	3	CJNE	R2,#data,code addr
88	2	MOV	data addr,R0	BB	3	CJNE	R3,#data,code addr
89	2	MOV	data addr,R1	BC	3	CJNE	R4,#data,code addr
8A	2	MOV	data addr,R2	BD	3	CJNE	R5,#data,code addr
8B	2	MOV	data addr,R3	BE	3	CJNE	R6,#data,code addr
8C	2	MOV	data addr,R4	BF	3	CJNE	R7,#data,code addr
8D	2	MOV	data addr,R5	C0	2	PUSH	data addr
8E	2	MOV	data addr,R6	C1	2	AJMP	code addr
8F	2	MOV	data addr,R7	C2	2	CLR	bit addr
90	3	MOV	DPTR,#data	C3	1	CLR	C
91	2	ACALL	code addr	C4	1	SWAP	A
92	2	MOV	bit addr,C	C5	2	XCH	A,data addr
93	1	MOVC	A,@A+DPTR	C6	1	XCH	A,@R0
94	2	SUBB	A,#data	C7	1	XCH	A,@R1
95	2	SUBB	A,data addr	C8	1	XCH	A,R0
96	1	SUBB	A,@R0	C9	1	XCH	A,R1
97	1	SUBB	A,@R1	CA	1	XCH	A,R2
98	1	SUBB	A,R0	CB	1	XCH	A,R3



80C51BH/80C51BH-2
80C31BH/80C31BH-2

ADVANCE INFORMATION

Table 3. Instruction Opcodes in Hexidecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands	Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP		33	1	RLC	A
01	2	AJMP	code addr	34	2	ADDC	A,#data
02	3	LJMP	code addr	35	2	ADDC	A,data addr
03	1	RR	A	36	1	ADDC	A,@R0
04	1	INC	A	37	1	ADDC	A,@R1
05	2	INC	data addr	38	1	ADDC	A,R0
06	1	INC	@R0	39	1	ADDC	A,R1
07	1	INC	@R1	3A	1	ADDC	A,R2
08	1	INC	R0	3B	1	ADDC	A,R3
09	1	INC	R1	3C	1	ADDC	A,R4
0A	1	INC	R2	3D	1	ADDC	A,R5
0B	1	INC	R3	3E	1	ADDC	A,R6
0C	1	INC	R4	3F	1	ADDC	A,R7
0D	1	INC	R5	40	2	JC	code addr
0E	1	INC	R6	41	2	AJMP	code addr
0F	1	INC	R7	42	2	ORL	data addr,A
10	3	JBC	bit addr, code addr	43	3	ORL	data addr,#data
11	2	ACALL	code addr	44	2	ORL	A,#data
12	3	LCALL	code addr	45	2	ORL	A,data addr
13	1	RRC	A	46	1	ORL	A,@R0
14	1	DEC	A	47	1	ORL	A,@R1
15	2	DEC	data addr	48	1	ORL	A,R0
16	1	DEC	@R0	49	1	ORL	A,R1
17	1	DEC	@R1	4A	1	ORL	A,R2
18	1	DEC	R0	4B	1	ORL	A,R3
19	1	DEC	R1	4C	1	ORL	A,R4
1A	1	DEC	R2	4D	1	ORL	A,R5
1B	1	DEC	R3	4E	1	ORL	A,R6
1C	1	DEC	R4	4F	1	ORL	A,R7
1D	1	DEC	R5	50	2	JNC	code addr
1E	1	DEC	R6	51	2	ACALL	code addr
1F	1	DEC	R7	52	2	ANL	data addr,A
20	3	JB	bit addr, code addr	53	3	ANL	data addr,#data
21	2	AJMP	code addr	54	2	ANL	A,#data
22	1	RET		55	2	ANL	A,data addr
23	1	RL	A	56	1	ANL	A,@R0
24	2	ADD	A,#data	57	1	ANL	A,@R1
25	2	ADD	A,data addr	58	1	ANL	A,R0
26	1	ADD	A,@R0	59	1	ANL	A,R1
27	1	ADD	A,@R1	5A	1	ANL	A,R2
28	1	ADD	A,R0	5B	1	ANL	A,R3
29	1	ADD	A,R1	5C	1	ANL	A,R4
2A	1	ADD	A,R2	5D	1	ANL	A,R5
2B	1	ADD	A,R3	5E	1	ANL	A,R6
2C	1	ADD	A,R4	5F	1	ANL	A,R7
2D	1	ADD	A,R5	60	2	JZ	code addr
2E	1	ADD	A,R6	61	2	AJMP	code addr
2F	1	ADD	A,R7	62	2	XRL	data addr,A
30	3	JNB	bit addr, code addr	63	3	XRL	data addr,#data
31	2	ACALL	code addr	64	2	XRL	A,#data
32	1	RETI		65	2	XRL	A,data addr



80C51BH/80C51BH-2
80C31BH/80C31BH-2

Table 2. MCS[®]-51 Instruction Set Description (Continued)

DATA TRANSFER (CONTINUED)				PROGRAM AND MACHINE CONTROL			
Mnemonic		Description	Byte Cyc	Mnemonic		Description	Byte Cyc
MOV	direct, #data	Move immediate data to direct byte	3 2	ACALL	addr11	Absolute Subroutine Call	2 2
MOV	@Ri,A	Move Accumulator to indirect RAM	1 1	LCALL	addr16	Long Subroutine Call	3 2
MOV	@Ri,direct	Move direct byte to indirect RAM	2 2	RET		Return from subroutine	1 2
MOV	@Ri,#data	Move immediate data to indirect RAM	2 1	RETI		Return from interrupt	1 2
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3 2	AJMP	addr11	Absolute Jump	2 2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1 2	LJMP	addr16	Long Jump	3 2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1 2	SJMP	rel	Short Jump (relative addr)	2 2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1 2	JMP	@A+DPTR	Jump indirect relative to the DPTR	1 2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1 2	JZ	rel	Jump if Accumulator is Zero	2 2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1 2	JNZ	rel	Jump if Accumulator is Not Zero	2 2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1 2	JC	rel	Jump if Carry flag is set	2 2
PUSH	direct	Push direct byte onto stack	2 2	JNC	rel	Jump if No Carry flag	2 2
POP	direct	Pop direct byte from stack	2 2	JB	bit,rel	Jump if direct Bit set	3 2
XCH	A,Rn	Exchange register with Accumulator	1 1	JNB	bit,rel	Jump if direct Bit Not set	3 2
XCH	A,direct	Exchange direct byte with Accumulator	2 1	JBC	bit,rel	Jump if direct Bit is set & Clear bit	3 2
XCH	A,@Ri	Exchange indirect RAM with A	1 1	CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3 2
XCHD	A,@Ri	Exchange low-order Digit ind RAM w A	1 1	CJNE	A,#data,rel	Comp, immed, to A & Jump if Not Equal	3 2
BOOLEAN VARIABLE MANIPULATION				CJNE	Rn,#data,rel	Comp, immed, to reg & Jump if Not Equal	3 2
CLR	C	Clear Carry flag	1 1	CJNE	@Ri,#data,rel	Comp, immed, to ind. & Jump if Not Equal	3 2
CLR	bit	Clear direct bit	2 1	DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2 2
SETB	C	Set Carry flag	1 1	DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3 2
SETB	bit	Set direct Bit	2 1	NOP		No operation	1 1
CPL	C	Complement Carry flag	1 1	Notes on data addressing modes:			
CPL	bit	Complement direct bit	2 1	Rn		—Working register R0-R7	
ANL	C,bit	AND direct bit to Carry flag	2 2	direct		—128 internal RAM locations, any I/O port, control or status register	
ANL	C/bit	AND complement of direct bit to Carry flag	2 2	@Ri		—Indirect internal RAM location addressed by register R0 or R1	
ORL	C/bit	OR direct bit to Carry flag	2 2	#data		—8-bit constant included in instruction	
ORL	C/bit	OR complement of direct bit to Carry flag	2 2	#data16		—16-bit constant included as bytes 2 & 3 of instruction	
MOV	C/bit	Move direct bit to Carry flag	2 1	bit		—128 software flags, any I/O pin, control or status bit	
MOV	bit,C	Move Carry flag to direct bit	2 2	Notes on program addressing modes:			
				addr16		—Destination address for LCALL & LJMP may be anywhere within the 64-K program memory address space	
				Addr11		—Destination address for ACALL & AJMP will be within the same 2-K page of program memory as the first byte of the following instruction	
				rel		—SJMP and all conditional jumps include an 8-bit offset byte. Range is +127-128 bytes relative to first byte of the following instruction	

All mnemonics copyrighted © Intel Corporation 1979



80C51BH/80C51BH-2
80C31BH/80C31BH-2

ADVANCE INFORMATION

Table 2. MCS[®]-51 Instruction Set Description

ARITHMETIC OPERATIONS				LOGICAL OPERATIONS (CONTINUED)					
Mnemonic		Description	Byte	Cyc	Mnemonic	Destination	Byte	Cyc	
ADD	A,Rn	Add register to Accumulator	1	1	ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1	ORL	A,#data	OR immediate data to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1	ORL	direct,A	OR Accumulator to direct byte	2	1
ADD	A,#data	Add immediate data to Accumulator	2	1	ORL	direct,#data	OR immediate data to direct byte	3	2
ADDC	A,Rn	Add register to Accumulator with Carry	1	1	XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1	XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1	XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1	XRL	A,#data	Exclusive-OR immediate data to A	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1	XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1	XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1	CLR	A	Clear Accumulator	1	1
SUBB	A,#data	Subtract immed data from A with Borrow	2	1	CPL	A	Complement Accumulator	1	1
INC	A	Increment Accumulator	1	1	RL	A	Rotate Accumulator Left	1	1
INC	Rn	Increment register	1	1	RLC	A	Rotate A Left through the Carry flag	1	1
INC	direct	Increment direct byte	2	1	RR	A	Rotate Accumulator Right	1	1
INC	@Ri	Increment indirect RAM	1	1	RRC	A	Rotate A Right through Carry flag	1	1
INC	DPTR	Increment Data Pointer	1	2	SWAP	A	Swap nibbles within the Accumulator	1	1
DEC	A	Decrement Accumulator	1	1					
DEC	Rn	Decrement register	1	1					
DEC	direct	Decrement direct byte	2	1					
DEC	@Ri	Decrement indirect RAM	1	1					
MUL	AB	Multiply A & B	1	4					
DIV	AB	Divide A by B	1	4					
DA	A	Decimal Adjust Accumulator	1	1					
LOGICAL OPERATIONS				DATA TRANSFER					
Mnemonic		Destination	Byte	Cyc	Mnemonic	Description	Byte	Cyc	
ANL	A,Rn	AND register to Accumulator	1	1	MOV	A,Rn	Move register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1	MOV	A,direct	Move direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1	MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1	MOV	A,#data	Move immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1	MOV	Rn,A	Move Accumulator to register	1	1
ANL	direct,#data	AND immediate data to direct byte	3	2	MOV	Rn,direct	Move direct byte to register	2	2
ORL	A,Rn	OR register to Accumulator	1	1	MOV	Rn,#data	Move immediate data to register	2	1
ORL	A,direct	OR direct byte to Accumulator	2	1	MOV	direct,A	Move Accumulator to direct byte	2	1
					MOV	direct,Rn	Move register to direct byte	2	2
					MOV	direct,direct	Move direct byte to direct	3	2
					MOV	direct,@Ri	Move indirect RAM to direct byte	2	2

ประวัติผู้เขียน

นายธนากร อรัญศิริ เกิดวันที่ 27 พฤศจิกายน พ.ศ. 2509 ที่อำเภอปทุมวัน จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีราชมงคล(วิทยาเขตเทเวศน์) ในปีการศึกษา 2534 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชา นิวเคลียร์เทคโนโลยี คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2536 ปัจจุบัน เป็นพนักงานรัฐวิสาหกิจ ที่ฝ่ายระบบสื่อสาร การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย