


```

#define PIC_BLACK      252
#define PIC_WHITE      255
#define PIC_A          1
#define PIC_B          2
#define PIC_HLINE      3
#define PIC_VLINE      4
#define PIC_C          5
#define PIC_D          6
#define PIC_E          7
#define PIC_R          8
#define PIC_Yellow     9

/*****
* Delay time function
*****/
void DELAY_100nS(int delaytime)                // 30us
{
    int i, j;

    for(i=0; i<delaytime; i++)
        for(j=0; j<1; j++);
}

void DELAY_mS(int delaytime)                    // 1ms
{
    int i, j;

    for(i=0; i<delaytime; i++)
        for(j=0; j<1600; j++);
}

void DELAY_S(int delaytime)                     // 1s
{
    int i, j, k;

    for(i=0; i<delaytime; i++)
        for(j=0; j<1000; j++)
            for(k=0; k<1600; k++);
}

/*****
* Reset function
*****/
void RESET()
{
    nRST_L;
    DELAY_mS(20);
    nRST_H;
    DELAY_mS(20);
}

```

```

}

/*****
* Read busy function
*****/
void READBUSY()
{
    while(1)
    {
        _NOP();
        if((P1IN & 0x04)==0x04)
            break;
    }
}

/*****
* 4line SPI Write Command function
*****/
void SPI4W_WRITECOM(unsigned char INIT_COM)
{
    unsigned char TEMPCOM;
    unsigned char scnt;
    P1DIR |= R_SDA;
    TEMPCOM=INIT_COM;
    nCS_H;
    nCS_L;
    SCLK_L;
    nDC_L;
    for(scnt=0;scnt<8;scnt++)
    {
        if(TEMPCOM&0x80)
            SDA_H;
        else
            SDA_L;
        SCLK_H;
        SCLK_L;
        TEMPCOM=TEMPCOM<<1;
    }
    nCS_H;
}

/*****
* 4line SPI Write Data function
*****/
void SPI4W_WRITEDATA(unsigned char INIT_DATA)
{
    unsigned char TEMPCOM;
    unsigned char scnt;
    P1DIR |= R_SDA;

```

```

    TEMPCOM=INIT_DATA;
    nCS_H;
    nCS_L;
    SCLK_L;
    nDC_H;
    for(scnt=0;scnt<8;scnt++)
    {
        if(TEMPCOM&0x80)
            SDA_H;
        else
            SDA_L;
        SCLK_H;
        SCLK_L;
        TEMPCOM=TEMPCOM<<1;
    }
    nCS_H;
    SCLK_L;
    nDC_H;
}

/*****
* 4line SPI Read Data function
*****/
unsigned char SPI4W_READDATA(void)
{
    unsigned char scnt,temp;
    P1DIR &=~ R_SDA;
    nDC_H;
    nCS_H;
    nCS_L;
    SCLK_L;

    for(scnt=0;scnt<8;scnt++)
    {

        if(P1IN&R_SDA)
            temp=(temp<<1)|0x01;
        else
            temp=temp<<1;
        SCLK_H;
        SCLK_L;
    }
    nCS_H;
    tempvalue= temp;
    return temp;
}

```

```
/******  
* Temperature sensing  
*****/
```

```
void read_temperure(void)  
{  
  
    SPI4W_WRITECOM(0x40);  
    DELAY_S(1);  
    temp1=SPI4W_READDATA();  
    temp2=SPI4W_READDATA();  
  
    tempvalue=temp1;  
    tempvalue=temp2;  
  
}
```

```
/******  
* Initialization function  
*****/
```

```
void INIT_JD79676()  
{  
  
    //FITI cmd.  
    SPI4W_WriteCom(0x4D);  
    SPI4W_WriteData(0x78);  
  
    SPI4W_WriteCom(0xAE);  
    SPI4W_WriteData(0xCF);  
  
    SPI4W_WriteCom(0xB0);  
    SPI4W_WriteData(0x13);  
  
    SPI4W_WriteCom(0xBD);  
    SPI4W_WriteData(0x07);  
  
  
    SPI4W_WriteCom(0x00);
```

```

    SPI4W_WriteData(0x0F);
    SPI4W_WriteData(0x29);

//    SPI4W_WriteCom(0x01);                                //POWER
SETTING
//    SPI4W_WriteData(0x07);

    SPI4W_WriteCom(0xE7);
    SPI4W_WriteData(0x1C);

    SPI4W_WriteCom(0xE3);
    SPI4W_WriteData(0x22);

    SPI4W_WriteCom(0xB6);    //
    SPI4W_WriteData(0x0F);

    SPI4W_WriteCom(0xB7);    //
    SPI4W_WriteData(0x7C);

    SPI4W_WriteCom(0xBE);    //
    SPI4W_WriteData(0xFF);

    SPI4W_WriteCom(0xF0);
//Border Parameter Settings
    SPI4W_WriteData(0x65);

    SPI4W_WriteCom(0xA6);
//Border Parameter Settings
    SPI4W_WriteData(0x63);

    SPI4W_WriteCom(0xE9);
    SPI4W_WriteData(0x01);

}

/*****
* Depp-sleep function
*****/
void enterdeepsleep()
{

```

```

    SPI4W_WRITECOM(0x07); // Deep sleep
    SPI4W_WRITEDATA(0xA5);
}

/*****
* Display picture function
*****/

void dis_img(unsigned char num)
{
    unsigned int row, col;
    unsigned int pcnt;

    SPI4W_WRITECOM(0x10); // DTM1 Write
    READBUSY();
    pcnt = 0; // Reset or Save
    Prompt Byte Number
    for(col=0; col<200; col++) // 172 columns in
total // send 128x252bits ram 2D13
    {
        for(row=0; row<24; row++) // A total of 72
lines, with 2 bits per pixel, or 72/4 bytes
        {
            switch (num)
            {
                case PIC_A:

                    SPI4W_WRITEDATA(~gImage_BWRY[pcnt]);

                    break;
                case PIC_B:

                    // SPI4W_WRITEDATA(gImage_2d13_124x250_1[pcnt]);

                    break;

                case PIC_WHITE:
                    SPI4W_WRITEDATA(0x55);
                    break;
            }
        }
    }
}

```

```

        case PIC_BLACK:
            SPI4W_WRITEDATA(0x00);
            break;

        case PIC_R:

            SPI4W_WRITEDATA(0xFF);
            break;

        case PIC_Yellow:

            SPI4W_WRITEDATA(0xAA);
            break;

        default:
            break;
    }
    pcnt++;
}
}

```

```

SPI4W_WRITECOM(0x04); // Power ON
READBUSY();
DELAY_mS(10);
SPI4W_WRITECOM(0x12); // Display Refresh
SPI4W_WRITEDATA(0x00);
DELAY_mS(10);
READBUSY();

SPI4W_WRITECOM(0x02); // Power OFF
SPI4W_WRITEDATA(0x00);
    READBUSY();
    DELAY_mS(20);

```

```

}

```

```

/*****
* Main function
*****/

```

```

void main( void )
{

    unsigned int i;
    WDTCTL = WDTPW + WDTOLD;           // Stop watchdog timer to
prevent time out reset
    BCSCTL1 = CALBC1_8MHZ;             // set DCO frequency 1MHZ
    DCOCTL = CALDCO_8MHZ;

    P1DIR |= 0xF8;                     // set P1.3~7 output


    RESET();
    READBUSY();
    INIT_JD79676();
    dis_img(PIC_A);
    enterdeepsleep();
    DELAY_S(10);
    _NOP();


    while(1);

}

```