

```

*****/
#include "msp430x22x4.h"
#include "image_Temp.h"

/*****
* Private macro
*****/
#define SDA_H      (P1OUT |=BIT7)           // P1.7
#define SDA_L      (P1OUT &=~BIT7)
#define SCLK_H     (P1OUT |=BIT6)           // P1.6
#define SCLK_L     (P1OUT &=~BIT6)
#define nCS_H      (P1OUT |=BIT5)           // P1.5
#define nCS_L      (P1OUT &=~BIT5)
#define nDC_H      (P1OUT |=BIT4)           // P1.4
#define nDC_L      (P1OUT &=~BIT4)
#define nRST_H     (P1OUT |=BIT3)           // P1.3
#define nRST_L     (P1OUT &=~BIT3)

#define R_SDA      0x80    //P1.7

#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

/*****
* custom variables
*****/
unsigned char Vcomvalue;
unsigned char tempvalue;
unsigned char temp1,temp2;

/*****
* Delay_time
*****/
void DELAY_100nS(int delaytime)           // 30us
{
    int i,j;
    for(i=0;i<delaytime;i++)
        for(j=0;j<8;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
*****/
void RESET()
{
    nRST_L;
    DELAY_mS(20);
    nRST_H;
    DELAY_mS(20);
}

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

/*****
* Write-command
*****/
void SPI4W_WRITECOM(unsigned char INIT_COM)
{
    unsigned char TEMPCOM;
    unsigned char sent;

```

```

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;
}

/*****
* Write-data
*****/
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;
}

/*****
* Read-data
*****/

```

[illegible]

```

{
    //FITI cmd.
    SPI4W_WRITECOM(0x4D);
    SPI4W_WRITEDATA(0xAA);

    SPI4W_WRITECOM(0x87);
    SPI4W_WRITEDATA(0x28);

    SPI4W_WRITECOM(0x88);
    SPI4W_WRITEDATA(0x80);

    SPI4W_WRITECOM(0x91);
    SPI4W_WRITEDATA(0x8D);

    SPI4W_WRITECOM(0xAA);           //improve bst performance of low VDD
    SPI4W_WRITEDATA(0x0F);
}
/*****
* Enter-deepsleep
*****/
void enterdeepsleep()
{
    SPI4W_WRITECOM(0x07); // Deep sleep
    SPI4W_WRITEDATA(0xA5);
}

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void dis_img_red(unsigned char num)
{
    unsigned int row, col;
    unsigned int pcnt;
    SPI4W_WRITECOM(0x13);
    READBUSY();
    pcnt = 0; // Reset
or Save Prompt Byte Number
    for(col=0; col<296; col++) // 172 columns in
total // send 128x252bits ram 2D13
    {
        for(row=0; row<19; row++) // A total of
72 lines, with 2 bits per pixel, or 72/4 bytes
        { switch (num)
            {

```

```

        case PIC_VLINE:
            if(col>198)
                SPI4W_WRITEDATA(0xff);
            else
                SPI4W_WRITEDATA(0x00);

            break;

        case PIC_HLINE:
            if(row>12)
                SPI4W_WRITEDATA(0xff);
            else
                SPI4W_WRITEDATA(0x00);
            break;

        case PIC_WHITE:
            SPI4W_WRITEDATA(0xFF);
            break;

        case PIC_BLACK:
            SPI4W_WRITEDATA(0x00);
            break;
        default:
            break;
    }
    pcnt++;
}

}

READBUSY();

SPI4W_WRITECOM(0x04);    // Power ON
READBUSY();
DELAY_mS(10);

SPI4W_WRITECOM(0x12);    // Display Refresh
DELAY_mS(10);
READBUSY();

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

}

```

```

/*****
* VCI setting function
*****/
void VCI_2_3V()
{
    P3DIR &=0x00;
    P3DIR |=0x00;    //2.3V
}

void VCI_2_4V()
{
    P3DIR &=0x00;
    P3DIR |= 0x02;
    P3OUT &=~BIT1;    //2.4V
}

void VCI_2_5V()
{
    P3DIR &=0x00;
    P3DIR |= 0x04;
    P3OUT &=~BIT2;    //2.5V
}

void VCI_3_0V()
{
    P3DIR &=0x00;
    P3DIR |= 0x01;
    P3OUT &=~BIT0;    //3.0V
}

void VCI_3_6V()
{
    P3DIR &=0x00;
    P3DIR |= 0x08;
    P3OUT &=~BIT3;    //3.6V
}

/*****
* Main function
*****/
void main( void )

```

```

{
    int i=30;
    WDTCTL = WDTPW + WDTHOLD; //
    Stop watchdog timer to prevent time out reset
    BCCTL1 = CALBC1_8MHZ; // set DCO
    frequency 12MHZ
    DCOCTL = CALDCO_8MHZ;

    P1DIR |=0x78; // set P1.3~7
    output
    P3DIR |= 0x10; //set p3.4,p3.5
    input

    RESET();
    READBUSY();
    INIT_JD79651();
    dis_img_red(PIC_WHITE);
    enterdeepsleep();
    DELAY_S(5);
    _NOP();

}

```