

```

/*-----*/
/*  Prorgam:3D7BWR_240X416_8253_Test                                */
/*  Version:A0      Description:  First version      Date:2020-10-8    */
/*  Author:Vick cheng                                           */
/*-----*/

/* Includes -----*/

/* Includes -----*/
#include "msp430x22x4.h"
#include "image.h"

#include "msp430g2955.h"

#include "waveinit_T2530.h"

/* IO_Configuration----- */
#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)

/* Display function-----*/

#define PIC_BLACK      252                //Disply black
#define PIC_WHITE      255                //Disply white
#define PIC_A          1                  //Disply char
#define PIC_B          2                  //Disply lebel
#define PIC_HLINE      3                  //Disply hline
#define PIC_VLINE      4                  //Disply vline
#define PIC_R          5                  //Disply rad
#define PIC_0          6                  //Disply rad
#define PIC_1          7
#define PIC_2          8
#define PIC_3          9
#define PIC_4          10
#define PIC_5          11
#define PIC_6          12

#define PIC_C          15

```

```

/* MCU delay configuration -----*/

void DELAY_mS(int delaytime)                // delaytime=1=1ms
{
    int i, j;
    for(i=0; i<delaytime; i++)
        for(j=0; j<2400; j++);
}

void DELAY_S(int delaytime)                // delaytime=1=1s
{
    int i, j, k;
    for(i=0; i<delaytime; i++)
        for(j=0; j<1000; j++)
            for(k=0; k<2400; k++);
}

/*-----*/
/* EPD function-----*/
/*-----*/

/*IC reset -----*/

void RESET()
{
    nRST_H;
    DELAY_mS(1);
    nRST_L;
    DELAY_mS(5);
    nRST_H;
    DELAY_mS(1);
}

/* IC read BUSY -----*/

void READBUSY()
{
    while(1)
    {
        _NOP();
        if((P1IN & 0x04)==0x04)                // Read BUSY electrica level
            break;
    }
}

```

```
/* 4line SPI Write Command -----*/
```

```
void SPI4W_WRITECOM(unsigned char INIT_COM)
{
    unsigned char TEMPCOM;
    unsigned char scnt;

    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 -----*/
```

```
void SPI4W_WRITEDATA(unsigned char INIT_DATA)
{
    unsigned char TEMPCOM;
    unsigned char scnt;

    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;
    }

    /* IC Sleep -----*/

    void enterdeepsleep()
    {
        READBUSY();
        SPI4W_WRITECOM(0x07);
        SPI4W_WRITEDATA(0xA5);
    }

    /* Display configuration -----*/
    void Display_QuickClear_Refresh_Mode()
    {

        SPI4W_WRITECOM(0xE0);
        SPI4W_WRITEDATA(0X02);

        SPI4W_WRITECOM(0xE5);
        SPI4W_WRITEDATA(0x53);    //setting slow update wf data 83
    }


    void dis_img_fast(unsigned char num,unsigned char num1)
    {
        unsigned int row, col;
        unsigned int pcnt;

        SPI4W_WRITECOM(0x50);
        SPI4W_WRITEDATA(0xC7);

        SPI4W_WRITECOM(0x10);
        READBUSY();
        pcnt = 0;
        for(col=0; col<416; col++)
        {
            for(row=0; row<30; row++)
            {
                switch (num)
                {
                    case PIC_A:
                        SPI4W_WRITEDATA(~gImage_3D7_BW_K[pcnt]);
                        break;

```

```

        case PIC_B:
            SPI4W_WRITEDATA(gImage_BW[pcnt]);
        break;

        case PIC_VLINE:
            if(col<208)
                SPI4W_WRITEDATA(0x00);
            else
                SPI4W_WRITEDATA(0xFF);
            break;
        case PIC_HLINE:
            if(row<15)
                SPI4W_WRITEDATA(0x00);
            else
                SPI4W_WRITEDATA(0xFF);
            break;
        case PIC_BLACK:
            SPI4W_WRITEDATA(0xFF);
        break;
        case PIC_WHITE:
            SPI4W_WRITEDATA(0x00);
        break;
        default:
            break;
    }
    pcnt++;
}

SPI4W_WRITECOM(0x13);
READBUSY();
pcnt = 0;
for(col=0; col<416; col++)
{
    for(row=0; row<30; row++)
    {
        switch (num1)
        {
            case PIC_A:
                SPI4W_WRITEDATA(~gImage_3D7_BW_K[pcnt]);
            break;

            case PIC_B:
                SPI4W_WRITEDATA(gImage_BW[pcnt]);
            break;

```

```

        case PIC_VLINE:
            if(col<208)
                SPI4W_WRITEDATA(0x00);
            else
                SPI4W_WRITEDATA(0xFF);
            break;
        case PIC_HLINE:
            if(row<15)
                SPI4W_WRITEDATA(0x00);
            else
                SPI4W_WRITEDATA(0xFF);
            break;
        case PIC_BLACK:
            SPI4W_WRITEDATA(0xFF);
            break;
        case PIC_WHITE:
            SPI4W_WRITEDATA(0x00);
            break;
        default:
            break;
    }
    pcnt++;
}
}

```

```

    SPI4W_WRITECOM(0x04);
    READBUSY();

```

```

    SPI4W_WRITECOM(0x12);

```

```

    READBUSY();
    SPI4W_WRITECOM(0x02);
    READBUSY();
    DELAY_mS(10);

```

```

}

```

```

void dis_img(unsigned char num)
{

```

```

unsigned int row, col;
unsigned int pcnt;

SPI4W_WRITECOM(0x13);
READBUSY();
pcnt = 0;
for(col=0; col<416; col++)
{
    for(row=0; row<30; row++)
    {
        switch (num)
        {
            case PIC_A:
                SPI4W_WRITEDATA(~gImage_416X240_ZITI[pcnt]);
                break;

            case PIC_VLINE:
                if(col<208)
                    SPI4W_WRITEDATA(0x00);
                else
                    SPI4W_WRITEDATA(0xFF);
                break;
            case PIC_HLINE:
                if(row<15)
                    SPI4W_WRITEDATA(0x00);
                else
                    SPI4W_WRITEDATA(0xFF);
                break;
            case PIC_BLACK:
                SPI4W_WRITEDATA(0xFF);
                break;
            case PIC_WHITE:
                SPI4W_WRITEDATA(0x00);
                break;
            default:
                break;
        }
        pcnt++;
    }
}

```

```

Display_QuickClear_Refresh_Mode(); //setting slow update

```

```

SPI4W_WRITECOM(0x04);
READBUSY();
DELAY_mS(20);

```

```

    SPI4W_WRITECOM(0x12);
    DELAY_mS(20);
    READBUSY();
    SPI4W_WRITECOM(0x02);
    READBUSY();
    DELAY_mS(20);

}

void INIT_UC8253()
{
    SPI4W_WRITECOM(0x00);    //panel setting
    SPI4W_WRITEDATA(0xD7);
    SPI4W_WRITEDATA(0x0E);

    SPI4W_WRITECOM(0x50);
    SPI4W_WRITEDATA(0x47);
}

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx    Main Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void main(void)
{
    int i=0;
    WDTCTL = WDTPW + WDTHOLD;    // Stop watchdog timer to prevent time out
reset
    BCSCCTL1 = CALBC1_12MHZ;    // set DCO frequency 1MHZ
    DCOCTL = CALDCO_12MHZ;

    P1DIR |=0xF8;    // set P1.3~7 output
    P3DIR = 0xff;    // set P3    output

```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img(PIC_BLACK);
enterdeepsleep();
DELAY_S(1);
_NOP();
```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img_fast(PIC_BLACK, PIC_WHITE);
enterdeepsleep();
DELAY_S(1);
_NOP();
```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img_fast(PIC_WHITE, PIC_BLACK);
enterdeepsleep();
DELAY_S(1);
_NOP();
```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img_fast(PIC_BLACK, PIC_VLINE);
enterdeepsleep();
DELAY_S(1);
_NOP();
```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img_fast(PIC_VLINE, PIC_HLINE);
enterdeepsleep();
DELAY_S(1);
_NOP();
```

```
RESET();
READBUSY();
INIT_UC8253();
dis_img_fast(PIC_HLINE, PIC_VLINE);
enterdeepsleep();
```

```
DELAY_S(1);  
_NOP();
```

```
while(1) {}  
}
```