```
/*****************************************************************************
**************************
*
* MSP430 microcontroller,MSP430G2955/MSP430F2274 devices.
*
* Prorgam: 1D8_BWRY_168x224_SSD2680_OTP
*
* Author : chengwei
*
* Rev. 1.0, Setup,2024.01.15
*
*****************************************************************************
**************************/



/*****************************************************************************
**************************
*Header File
*****************************************************************************
**************************/

#include "msp430x22x4.h"
#include "image.h"



/*****************************************************************************
**************************
* IO Port Define
*****************************************************************************
**************************/

#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


/*****************************************************************************
**************************
* Globle Variable definitions
```

```
*****************************************************************************************
*************************/


unsigned char tempvalue;
unsigned char tempvalue1;
unsigned char temp1;
unsigned char temp2;
unsigned char Byte1,Byte2,Byte3,Byte4;
unsigned char vcomExist=0;
unsigned char vcomBuf=0;

unsigned int opt_data_idx=0;
unsigned char tempBuff [112];
unsigned char paraBuff [50];

unsigned char OTPRECV[112];

unsigned int temp,CRC_temp;
unsigned int Check_MTPdata,Check_Userdata;
unsigned char CRC_Byte;
unsigned char CRC_bite1,CRC_bite2,CRC_bite3,CRC_bite4;
unsigned char UserCRC_bite1,UserCRC_bite2,UserCRC_bite3,UserCRC_bite4;
unsigned char A4_OTP_Byte0;
unsigned char A4_OTP_Byte1;
unsigned char A4_OTP_Byte2;
unsigned char A4_OTP_Byte3;

unsigned char temp1_h,temp1_l;
unsigned char temp2_h,temp2_l;

/*****************************************************************************************
**************************
* Image-Display define
*****************************************************************************************
*************************/


#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
```

```
/*****************************************************************************
*****************************
* MCU delay configuration
*****************************************************************************
*****************************/
void DELAY_100nS(int delaytime)
{
     int i,j;

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

void DELAY_mS(int delaytime)
{
     int i,j;

     for(i=0;i<delaytime;i++)
          for(j=0;j<200;j++);
}

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

     for(i=0;i<delaytime;i++)
          for(j=0;j<1000;j++)
               for(k=0;k<200;k++);
}

/*****************************************************************************
*****************************
* IC reset
*****************************************************************************
*****************************/
void RESET()
{
     nRST_L;
     DELAY_mS(20);
     nRST_H;
     DELAY_mS(20);
}

/*****************************************************************************
*****************************
* IC Read Busy
```

```
*****************************************************************************************
*************************/


void READBUSY()
{
     while(1)
     {
          _NOP();
          if((P1IN & 0x04)==0x04)
          break;
     }
}

/*****************************************************************************************
*************************
* 4line SPI Write Command
*****************************************************************************************
*************************/


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;
          }

}

/*****************************************************************************************
************************
* 4line SPI Write Data
*****************************************************************************************
*************************/
```

```
void SPI4W_WRITEDATA(unsigned char INIT_DATA)
{
     unsigned char TEMPCOM;
     unsigned char scnt;
     P1DIR |= R_SDA;
     TEMPCOM=INIT_DATA;

     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;
          }

}


/*****************************************************************************************
**************************
* 4line SPI Read Data
*****************************************************************************************
**************************/


   unsigned char SPI4W_READDATA(void)
   {
    P1DIR &=~ R_SDA;
     unsigned char scnt,temp;
     temp=0;

     nDC_H;
     for(scnt=0;scnt<8;scnt++)
          {
               SCLK_L;
               if(P1IN&R_SDA)
                    temp=(temp<<1)|0x01;
               else
                    temp=temp<<1;
                    SCLK_H;
                    SCLK_L;
          }
```

```
    return temp;

  }



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


void read_temperture(void)
{


      SPI4W_WRITECOM(0x40);
      DELAY_S(1);
      temp1=SPI4W_READDATA();
      temp2=SPI4W_READDATA();


}




/*****************************************************************************************
**************************
* Read Revision function
*****************************************************************************************
**************************/


void read_Revision()
{
  SPI4W_WRITECOM(0x70);
  Byte1=SPI4W_READDATA();//chip id
  Byte2=SPI4W_READDATA();//lut_ver[0:7]
  Byte3=SPI4W_READDATA();//lut_ver[8:15]
  Byte4=SPI4W_READDATA();//lut_ver[23:16]
}
```

```
/*****************************************************************************************
**************************
* EPD init
*****************************************************************************************
**************************/

void INIT_SSD2680()
{


     SPI4W_WRITECOM(0x00);
     SPI4W_WRITEDATA(0x2F);
     SPI4W_WRITEDATA(0x29);
     READBUSY();

     SPI4W_WRITECOM(0x01);
     SPI4W_WRITEDATA(0x07);
     SPI4W_WRITEDATA(0xF0);

     SPI4W_WRITECOM(0x06);     //BTST
     SPI4W_WRITEDATA(0x0F);
     SPI4W_WRITEDATA(0x8B);
     SPI4W_WRITEDATA(0x9C);
     SPI4W_WRITEDATA(0x96);

     SPI4W_WRITECOM(0x30);  //PLL
     SPI4W_WRITEDATA(0x08);

     SPI4W_WRITECOM(0x50);  //CDI Booder 设置   POR:97
     SPI4W_WRITEDATA(0x37);

     SPI4W_WRITECOM(0x60); //TCON
     SPI4W_WRITEDATA(0x02);
     SPI4W_WRITEDATA(0x02);

    SPI4W_WRITECOM(0x61);
    SPI4W_WRITEDATA(0x00);
    SPI4W_WRITEDATA(0xA8);
    SPI4W_WRITEDATA(0x00);
    SPI4W_WRITEDATA(0xE0);


     SPI4W_WRITECOM(0x62);  // HTOTAL
     SPI4W_WRITEDATA(0x59);
     SPI4W_WRITEDATA(0x59);
     SPI4W_WRITEDATA(0x59);
     SPI4W_WRITEDATA(0x45);
```

```
    SPI4W_WRITEDATA(0x77);
    SPI4W_WRITEDATA(0x69);
    SPI4W_WRITEDATA(0x59);
    SPI4W_WRITEDATA(0x4A);


    SPI4W_WRITECOM(0x65);
    SPI4W_WRITEDATA(0x00);
    SPI4W_WRITEDATA(0x00);//08
    SPI4W_WRITEDATA(0x00);
    SPI4W_WRITEDATA(0x00);


    SPI4W_WRITECOM(0xE3);  //PWS
    SPI4W_WRITEDATA(0x22);

    SPI4W_WRITECOM(0xE7);  //PST
    SPI4W_WRITEDATA(0x1C);

    SPI4W_WRITECOM(0xE9);
    SPI4W_WRITEDATA(0x01);


}



/*****************************************************************************************
*************************
* Enter sleep
*****************************************************************************************
************************/

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







/*****************************************************************************************
*************************
* Diplay Image
```

```
**************************************************************************************
*************************/

void dis_img(unsigned char num)
{

     unsigned int row, col;
     unsigned int pcnt;


     SPI4W_WRITECOM(0x10);   // DTM1 Write
          READBUSY();
          pcnt = 0;
          for(col=0; col<224; col++)
          {
               for(row=0; row<42; row++)
               {
                         switch (num)
                    {
                         case PIC_A:

                              SPI4W_WRITEDATA(gImage_test_BWRY[pcnt]);

                             break;
                         case PIC_B:

                              SPI4W_WRITEDATA(gImage_tartan[pcnt]);

                             break;



                          case PIC_VLINE:
                               if(col<74)
                               SPI4W_WRITEDATA(0x00);
                               else if(col<148)
                               SPI4W_WRITEDATA(0xAA);
                                                  else if(col<222)
                                                  SPI4W_WRITEDATA(0xFF);
                               else
                                                  SPI4W_WRITEDATA(0x55);
                               break;




                         case PIC_HLINE:
                               if(row<9)
```

```
                    SPI4W_WRITEDATA(0x00);
                    else if(row<18)
                    SPI4W_WRITEDATA(0xAA);
                                        else if(row<28)
                                        SPI4W_WRITEDATA(0xFF);
                                        else
                                        SPI4W_WRITEDATA(0x55);
                    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);
}




//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx   Main Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxx
void main( void )
{

    unsigned int i;

    WDTCTL = WDTPW + WDTHOLD;                               // Stop watchdog timer to
prevent time out reset
    BCSCTL1 = CALBC1_1MHZ;                                  // set DCO frequency 1MHZ
    DCOCTL = CALDCO_1MHZ;

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



    RESET();
    READBUSY();
    INIT_SSD2680();
    dis_img(PIC_BLACK);
    enterdeepsleep();
    DELAY_S(5);

    RESET();
    READBUSY();
    INIT_SSD2680();
    dis_img(PIC_WHITE);
    enterdeepsleep();
    DELAY_S(5);


    RESET();
    READBUSY();
    INIT_SSD2680();
    dis_img(PIC_A);
    enterdeepsleep();
    DELAY_S(5);
```

```
    _NOP();


}
```