

//Note: For the best effect, we suggest you view the program with the Source Insight software.

```
//xx Includes
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
#include "msp430x22x4.h"
#include "image1.h"

//xx Private macro
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

#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 DELAY_TIME  200                             // The Time after
image remains on screen (unit: seconds)

//#define MODEL1                                     // panel scan
direction

#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

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx Delay Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void DELAY_100nS(int delaytime)                      // 30us
{
    int i,j;
```

[illegible]

[illegible]

```
    READBUSY();  
    SPI4W_WRITECOM(0x07);  
    SPI4W_WRITEDATA(0xA5);  
}  
  
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
//xx Image Display Function  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx  
void dis_img_red(unsigned char num)  
{  
    unsigned int row, col;  
    unsigned int pcnt;  
  
    SPI4W_WRITECOM(0x61);  
    SPI4W_WRITEDATA(0xB0); //176  
    SPI4W_WRITEDATA(0x01);  
    SPI4W_WRITEDATA(0xE0); //480  
  
    SPI4W_WRITECOM(0x13);  
    READBUSY();  
    pcnt = 0; //  
Reset or Save Prompt Byte Number  
for(col=0; col<480; col++) // 172  
columns in total // send 128x252bits ram 2D13  
{  
for(row=0; row<22; row++) // A  
total of 72 lines, with 2 bits per pixel, or 72/4 bytes  
{ switch (num)  
{  
  
case PIC_BLACK:  
SPI4W_WRITEDATA(0x00);  
break;  
  
case PIC_WHITE:  
SPI4W_WRITEDATA(0xFF);  
break;  
default:  
break;  
}  
pcnt++;  
}  
}  
  
SPI4W_WRITECOM(0x04);  
READBUSY();  
DELAY_mS(100);  
SPI4W_WRITECOM(0x12);
```

```

READBUSY();
SPI4W_WRITECOM(0x02);
DELAY_mS(1);
}
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx Main Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void main( void )
{
    int i=10;
        WDTCTL = WDTPW + WDTXOLD; // Stop watchdog
timer to prevent time out reset
        BCSCTL1 = CALBC1_12MHZ; // set DCO
frequency 1MHz
        DCOCTL = CALDCO_12MHZ;

        P1DIR |=0xF8; // set P1.3~7
output
        P3DIR |= 0x10; // p3.4,p3.5 Set
as a input


                RESET(); //
E-paper Controller Reset
                INIT_UC8151();
                dis_img_red(PIC_WHITE);
                enterdeepsleep();
                DELAY_S(3);
                _NOP();
}

```