


```
#define PIC_D      6
#define PIC_E      7
#define PIC_R      8


//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx    Delay Function
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
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<200;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<200;k++);
}

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx    E-paper Driver Functions
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

// Reset
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void RESET()
{
    nRST_L;
    DELAY_mS(20);
    nRST_H;
    DELAY_mS(20);
}
```

```

// Check Busy
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void READBUSY ()
{
    while(1)
    {
        NOP ();
        if((P1IN & 0x04)==0x04)
            break;
    }
}

// Enter command xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
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;
}

// Enter data
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
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;
}

```

```

// E-Paper Driver Initialization

```

```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

```

void INIT_EK79651 ()

```

```

{
    //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 deeps leep

```

```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

```

void enterdeepsleep ()

```

```

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

```

```

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

```

//xx Image Display Function

```

```

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

```

//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

```

void dis_img_red(unsigned char num)

```

```

{
    unsigned int row, col;
    unsigned int pcnt;

```

```

        SPI4W_WRITECOM(0x13); // DTM2 Write
        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<16; row++) // A
total of 72 lines, with 2 bits per pixel, or 72/4 bytes
            { switch (num)
              {

                case PIC_R:
                    SPI4W_WRITEDATA(0xff);
                    break;

                case PIC_VLINE:
                    if(col>198)
                        SPI4W_WRITEDATA(0xff);
                    else
                        SPI4W_WRITEDATA(0x00);
                    break;
                case PIC_HLINE:
                    if(row>10)
                        SPI4W_WRITEDATA(0xff);
                    else
                        SPI4W_WRITEDATA(0x00);
                    break;
                case PIC_WHITE:
                    SPI4W_WRITEDATA(0x00);
                    break;

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

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

    DELAY_mS(10);

```

```

SPI4W_WRITECOM(0x02); // Power OFF
}
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx Main Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void main( void )
{
    //int i;
        WDTCTL = WDTPW + WDTHOLD;                                // Stop
watchdog timer to prevent time out reset
        BCSCCTL1 = CALBC1_1MHZ;                                    // set DCO
frequency 1MHZ
        DCOCTL = CALDCO_1MHZ;

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

RESET();
    READBUSY();
    INIT_EK79651();
    dis_img_red(PIC_WHITE);
    enterdeepsleep();
    DELAY_S(3);
    _NOP();
}

```