

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

[illegible]

```

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

// Enter Command xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
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;
    }
    nCS_H;
}

// Enter Data
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
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
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

unsigned char SPI4W_READDATA(void)
{
    unsigned char scnt,temp;
    P1DIR &=~ R_SDA;
    nCS_H;
    nCS_L;
    SCLK_H;
    nDC_H;
    for(scnt=0;scnt<8;scnt++)
    {
        SCLK_H;
        if(P1IN&R_SDA)
            temp=(temp<<1)|0x01;
        else
            temp=temp<<1;
        SCLK_L;
    }
    nCS_H;
    tempvalue= temp;
    return temp;
}

// Temperature sensing
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

void read_temperure(void) //V1.1
{
    SPI4W_WRITECOM(0x40);
    READBUSY();
    temp1=SPI4W_READDATA();
    temp2=SPI4W_READDATA();
    tempvalue=temp1;
    tempvalue=temp2;
}

void PWM_ADJ1()
{
    SPI4W_WRITECOM(0xFF);
    SPI4W_WRITEDATA(0xA5);

    SPI4W_WRITECOM(0xEF);
    SPI4W_WRITEDATA(0x03); //H0
    SPI4W_WRITEDATA(0x1E); //L0

    SPI4W_WRITEDATA(0x0A); //H1

```

```

SPI4W_WRITEDATA(0x1B); //L1

SPI4W_WRITEDATA(0x0B); //H2
SPI4W_WRITEDATA(0x17); //L2

SPI4W_WRITECOM(0xC3); //PWM en
SPI4W_WRITEDATA(0xFD); //fd

SPI4W_WRITECOM(0xDC);
SPI4W_WRITEDATA(0x01);

SPI4W_WRITECOM(0xDD);
SPI4W_WRITEDATA(0x08);


SPI4W_WRITECOM(0xDE);
SPI4W_WRITEDATA(0x41);

SPI4W_WRITECOM(0xFD);
SPI4W_WRITEDATA(0x01);

SPI4W_WRITECOM(0xE8);
SPI4W_WRITEDATA(0x03);

SPI4W_WRITECOM(0xDA);
SPI4W_WRITEDATA(0x0F);


SPI4W_WRITECOM(0xC9);
SPI4W_WRITEDATA(0x00);

SPI4W_WRITECOM(0xA8);
SPI4W_WRITEDATA(0x0E);

SPI4W_WRITECOM(0xFF);
SPI4W_WRITEDATA(0xE3);

}

```

```

void PWM_ADJ2 ()
{
    SPI4W_WRITECOM(0xFF);
    SPI4W_WRITEDATA(0xA5);

    SPI4W_WRITECOM(0xEF);
    SPI4W_WRITEDATA(0x03); //H0
    SPI4W_WRITEDATA(0x1E); //L0

    SPI4W_WRITEDATA(0x0A); //H1
    SPI4W_WRITEDATA(0x1B); //L1

    SPI4W_WRITEDATA(0x0E); //H2
    SPI4W_WRITEDATA(0x19); //L2
}

```

```

    SPI4W_WRITECOM(0xDC); //CPCK EN
    SPI4W_WRITEDATA(0x01);

    SPI4W_WRITECOM(0xDD); //CPCK H
    SPI4W_WRITEDATA(0x06);

    SPI4W_WRITECOM(0xDE); //CPCK L
    SPI4W_WRITEDATA(0x46);

    SPI4W_WRITECOM(0xFF); // Exit Test command      *****
    SPI4W_WRITEDATA(0xE3);

}

void INIT_IST7163()
{
    SPI4W_WRITECOM(0x00);
    SPI4W_WRITEDATA(0xCF);
    SPI4W_WRITEDATA(0x09);

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

    SPI4W_WRITECOM(0x30); // frame rate from OTP
    SPI4W_WRITEDATA(0x08); //

    SPI4W_WRITECOM(0x03);
    SPI4W_WRITEDATA(0x10);
    SPI4W_WRITEDATA(0x54);
    SPI4W_WRITEDATA(0x44);

    SPI4W_WRITECOM(0x06); //bst level modify
    SPI4W_WRITEDATA(0x40); //PWM0[7:6]  x  x,  x  x  x  x
    default=0x00
    SPI4W_WRITEDATA(0x00); //PWM1[7:6]  x  x,  x  x  x  x
    default=0x00
    SPI4W_WRITEDATA(0x00); //x  x  x  x,  x  x  x  x
    default=0x00

    SPI4W_WRITECOM(0x50);
    SPI4W_WRITEDATA(0x37);

    SPI4W_WRITECOM(0x60);
    SPI4W_WRITEDATA(0x03);
    SPI4W_WRITEDATA(0x03);

    SPI4W_WRITECOM(0x61); // RESOLUTION SETTING
    SPI4W_WRITEDATA(0x00);
    SPI4W_WRITEDATA(0xF0);
    SPI4W_WRITEDATA(0x01);
    SPI4W_WRITEDATA(0xA0);

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

```

```

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

    PWM_ADJ1();
}

// Enter Deep Sleep
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(0x10); // DTM1 Write
    READBUSY();
    pcnt = 0;
    for(col=0; col<416; col++)
    {
        for(row=0; row<60; row++)
        {
            switch (num)
            {
                case PIC_A:
                    SPI4W_WRITEDATA(gImage_3D7_240X416_IMAG4[pcnt]);
                    break;

                case PIC_B:
                    SPI4W_WRITEDATA(0x00);
                    break;

                case PIC_VLINE:
                    if(col<104)
                        SPI4W_WRITEDATA(0xAA);
                    else if(col<208)
                        SPI4W_WRITEDATA(0xFF);
                    else if(col<312)
                        SPI4W_WRITEDATA(0x55);
                    else
                        SPI4W_WRITEDATA(0x00);
            }
        }
    }
}

```

```

        break;

    case PIC_HLINE:
        if(row<15)
            SPI4W_WRITEDATA(0xAA);
        else if(row<30)
            SPI4W_WRITEDATA(0xFF);
        else if(row<45)
            SPI4W_WRITEDATA(0x55);
        else
            SPI4W_WRITEDATA(0x00);
        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);

PWM_ADJ2();

SPI4W_WRITECOM(0x12); // Display Refresh
SPI4W_WRITEDATA(0x00);
DELAY_mS(10);
READBUSY();

SPI4W_WRITECOM(0x02); // Power OFF
SPI4W_WRITEDATA(0x00);
READBUSY();
DELAY_mS(20);
}

```

```
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xx   Main Function
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
//xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
void main( void )
{
    unsigned int i;

    WDTCTL = WDTPW + WDTHOLD;                                     // Stop
watchdog timer to prevent time out reset
    BCSCTL1 = CALBC1_8MHZ;                                       // set DCO
frequency 1MHZ
    DCOCTL = CALDCO_8MHZ;

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

    RESET();
    DELAY_ms(20);
    read_temperure();

    RESET();
    READBUSY();
    INIT_I2C7163();
    dis_img_red(PIC_WHITE);
    enterdeepsleep();
    DELAY_S(60);
    _NOP();

    RESET();
    READBUSY();
    INIT_I2C7163();
    dis_img_red(PIC_A);
    enterdeepsleep();
    DELAY_S(10);
    _NOP();

}
}
```