

// This example program drives single HT16D31B via I2C interface and display the LED matrix in checkerboard pattern.

```
#include "HT66F2390.h"
```

```
#define SDA                _pa3
#define SDAC                _pac3
#define SCL                _pa5
#define s1                  _pc0
#define s2                  _pc1
#define s3                  _pc2
#define s4                  _pc3
#define s5                  _pc4
#define s6                  _pc5
```

```
void IO_INIT();
```

```
void LED_INIT();
```

```
void clear_ramdata();
```

```
_Bool CheckACK(void);
```

```
void LED_ON(unsigned char number,unsigned char ADDR);
```

```
void option_led_command(unsigned char number,unsigned char CMD,unsigned char ADDR);
```

```
void func_led_command(unsigned char number,unsigned char CMD,unsigned char ADDR,unsigned char DATA);
```

```
void software_Restart(unsigned char number);
```

```
void main()
```

```
{
```

```
    volatile unsigned char i,j,k;
```

```
    _wdtc=0xAF; //close WDT
```

```
    IO_INIT();
```

```
    LED_INIT();
```

```
    j=0xff;
```

```
    while(1)
```

```
    {
```

```
        if(j==0)
```

```
        {
```

```
            for(i=0;i<144;i++)
```

```
            {
```

```
                func_led_command(1,0x80,i,0xff);
```

```
            }
```

```
            j=255;
```

```
            GCC_DELAY(250000);
```

```
        }else
```

```
        {
```

```
            clear_ramdata();
```

```
            for(i=0;i<144;i=i+32)
```

```
            {
```

```
                func_led_command(1,0x80,i,255);
```

```

        func_led_command(1,0x80,i+2,255);
        func_led_command(1,0x80,i+4,255);
        func_led_command(1,0x80,i+6,255);
        func_led_command(1,0x80,i+0x11,255);
        func_led_command(1,0x80,i+0x13,255);
        func_led_command(1,0x80,i+0x15,255);
        func_led_command(1,0x80,i+0x17,255);
    }
    j=0;
    GCC_DELAY(250000);
}
GCC_DELAY(250000);
}
}
void IO_INIT()
{
    _pas0=0x00;
    _pcc=0xff;
    _pcpu=0xff;
    _pac=0x00;
}
void option_led_command(unsigned char number,unsigned char
CMD,unsigned char ADDR)
{
    unsigned char i,Send_temp = 0,slave;
    SDA=0;
    SCL=0;
    slave=214+number*2;
    if(number==5)
    slave=0x5c;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = slave&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        slave = slave<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=1;
    SDAC=0;

    for(i =0;i<8;i++)

```

```

    {
        SCL=0;
        Send_temp = CMD&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        CMD = CMD<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=1;
    SDAC=0;

    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = ADDR&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        ADDR = ADDR<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;

error:
    SCL=0;
    GCC_DELAY(4);
    SCL=1;
    GCC_DELAY(4);
    SDA=1;
}
void func_led_command(unsigned char number,unsigned char
CMD,unsigned char ADDR,unsigned char DATA)
{
    unsigned char i,Send_temp = 0,slave;

```

```

    SDA=0;
    SCL=0;
    slave=214+number*2;
    if(number==5)
    slave=0x5c;
    for(i =0;i<8;i++)
{
    SCL=0;
    Send_temp = slave&0x80;
    SDA = 1;
    if(Send_temp == 0)
    SDA = 0;
    GCC_NOP();
    SCL=1;
    slave = slave<<1;
}
SCL=0;
SDA=1;
SDAC=1;
if(!CheckACK()) goto error;

SDA=0;
SDAC=0;

for(i =0;i<8;i++)
{
    SCL=0;
    Send_temp = CMD&0x80;
    SDA = 1;
    if(Send_temp == 0)
    SDA = 0;
    GCC_NOP();
    SCL=1;
    CMD = CMD<<1;
}
SCL=0;
SDA=1;
SDAC=1;
if(!CheckACK()) goto error;

SDA=0;
SDAC=0;

for(i =0;i<8;i++)
{
    SCL=0;
    Send_temp = ADDR&0x80;
    SDA = 1;
    if(Send_temp == 0)

```

```

        SDA = 0;
        GCC_NOP();
        SCL=1;
        ADDR = ADDR<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = DATA&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        DATA = DATA<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;

error:
    SDA=0;
    SCL=0;
    GCC_DELAY(4);
    SCL=1;
    GCC_DELAY(4);
    SDA=1;
}
void LED_INIT()
{
    unsigned char i;
    software_Restart(1);
    GCC_NOP();
    software_Restart(2);
    GCC_NOP();
    software_Restart(3);
    GCC_NOP();
    software_Restart(4);
    GCC_NOP();
}

```

```

        opction_led_command(5,0x30,0x00);
        opction_led_command(5,0x11,0x00);
        opction_led_command(5,0x12,0x01);
        opction_led_command(4,0x34,0x02); //
slave
        opction_led_command(3,0x34,0x02); //
slave
        opction_led_command(2,0x34,0x02); //
slave
        opction_led_command(1,0x34,0x01); //
master
        opction_led_command(5,0x31,0x00); //
灰階模式
        LED_ON(5,0);
                //LED ON 控制

        clear_ramdata();
        opction_led_command(5,0x35,0x02); //
SYS ON、DISPLAY OFF

        opction_led_command(5,0x39,0x80);
        opction_led_command(5,0x36,0x0F); //
設定定電流
        opction_led_command(5,0x37,0xFF); //
亮度設定
        opction_led_command(5,0x35,0x03); //
SYS ON、DISPLAY OFF
}

void clear_ramdata()
{
    unsigned char i;
    opction_led_command(5,0xFD,0x00);
    for(i=0;i<144;i++)
    {
        func_led_command(5,0x80,i,0x00);
    }
    opction_led_command(5,0xFD,0x01);
    for(i=0;i<144;i++)
    {
        func_led_command(5,0x80,i,0x00);
    }
    opction_led_command(5,0xFD,0x00);
}

```

```

void software_Restart(unsigned char number)
{
    unsigned char i,Send_temp = 0,slave,CMD;
    SDA=0;
    SCL=0;
    slave=214+number*2;
    if(number==5)
    slave=0x5c;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = slave&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        slave = slave<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=1;
    SDAC=0;
    CMD=0xcc;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = CMD&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        CMD = CMD<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=1;
    SDAC=0;
error:
    SCL=0;
    GCC_DELAY(4);
    SCL=1;
    GCC_DELAY(4);
}

```

```

        SDA=1;
    }
void LED_ON(unsigned char number,unsigned char ADDR)
{
    unsigned char i,j,k,Send_temp = 0,slave,DATA,color_SW,CMD;
    option_led_command(5,0xfd,0x00);
    SDA=0;
    SCL=0;
    CMD=0x84;
    slave=214+number*2;
    if(number==5)
    slave=0x5c;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = slave&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        slave = slave<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;

    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = CMD&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        CMD = CMD<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;

```



```

for(i =0;i<8;i++)
{
    SCL=0;
    Send_temp = ADDR&0x80;
        SDA = 1;
    if(Send_temp == 0)
        SDA = 0;
    GCC_NOP();
    SCL=1;
    ADDR = ADDR<<1;
}
SCL=0;
SDA=1;
SDAC=1;
if(!CheckACK()) goto error;

SDA=0;
SDAC=0;

for(j=0;j<9;j++)                                //DATA START
{
    DATA=0xff;
    for(i =0;i<8;i++)
    {
        SCL=0;
        Send_temp = DATA&0x80;
        SDA = 1;
        if(Send_temp == 0)
            SDA = 0;
        GCC_NOP();
        SCL=1;
        DATA = DATA<<1;
    }
    SCL=0;
    SDA=1;
    SDAC=1;
    if(!CheckACK()) goto error;

    SDA=0;
    SDAC=0;
}

error:
    SDA=0;
    SCL=0;
    GCC_DELAY(4);
    SCL=1;
    GCC_DELAY(4);
    SDA=1;

```

```

}
_Bool CheckACK(void)
{
    unsigned char i=200;
    SDAC=1;
    //設定SDA為輸入模式
    SCL=1;
    //設定 SCL=1
    while(i--)
    {
        if(!SDA) { SCL=0; return 1;} //若裝置回應
ACK則返回並回傳1
        GCC_DELAY(200);
        //200*100uS=20ms 若20ms未回應
    }
    SCL=0;
    //SCL=0
    return 0;
    //若20ms未回應回傳0
}

```