

## 25.10. 深大实验--硬件 SPI 驱动 OLED 显示图片、图形、字符

### 25.10.1. 实验介绍

1、本实验项目主要目的:

- 硬件 SPI 驱动 OLED 显示
- 如何实现显示图片、6\*8 点阵字符、8\*16 点阵字符、16\*16 点阵汉字
- 如何画直线、方框、圆

2、掌握 Ai8051U 实验箱原理图中 SPI 驱动 OLED 电路

3、了解图片取模的原理,并熟练使用 AIapp-ISP 系统软件中图片取模工具

4、了解字库生成原理,并熟练使用 AIapp-ISP 系统软件中字库生成工具

5、继续熟悉如何管理多文件项目

#### a) Aiapp-ISP 系统软件图片取模工具的使用

打开 AIapp-ISP 系统软件:

- ➡ 点击菜单栏“工具”→“图片取模工具”,打开“图片取模工具”浮窗
- ➡ 点击“打开图片”按钮,选择准备要取模的图片
- ➡ 点击“打开”按钮

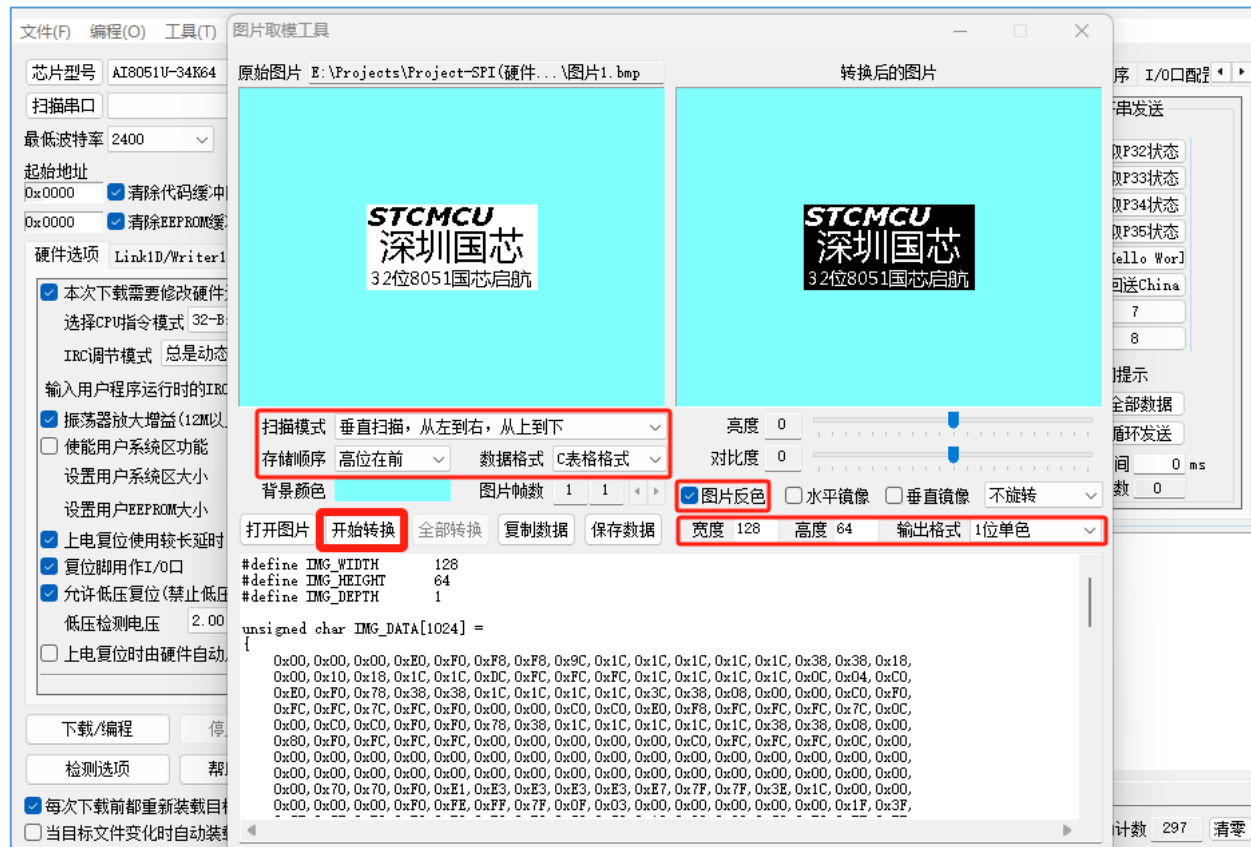
如下图:



此时, 在打开的“图片取模工具”浮窗, 我们看到被选择的图片显示在原始图片区和转换后的图片区:

- 扫描模式: 选择“垂直扫描, 从左到右, 从上到下”
- 存储顺序: 选择“高位在前”
- 数据格式: 选择“C 表格格式”
- 勾选“图片反色”
- 设置: 宽度: 128; 高度: 64; 输出格式: 1 位单色
- 点击“开始转换”

就会生成相应的点阵图片 C 代码, 如下图:



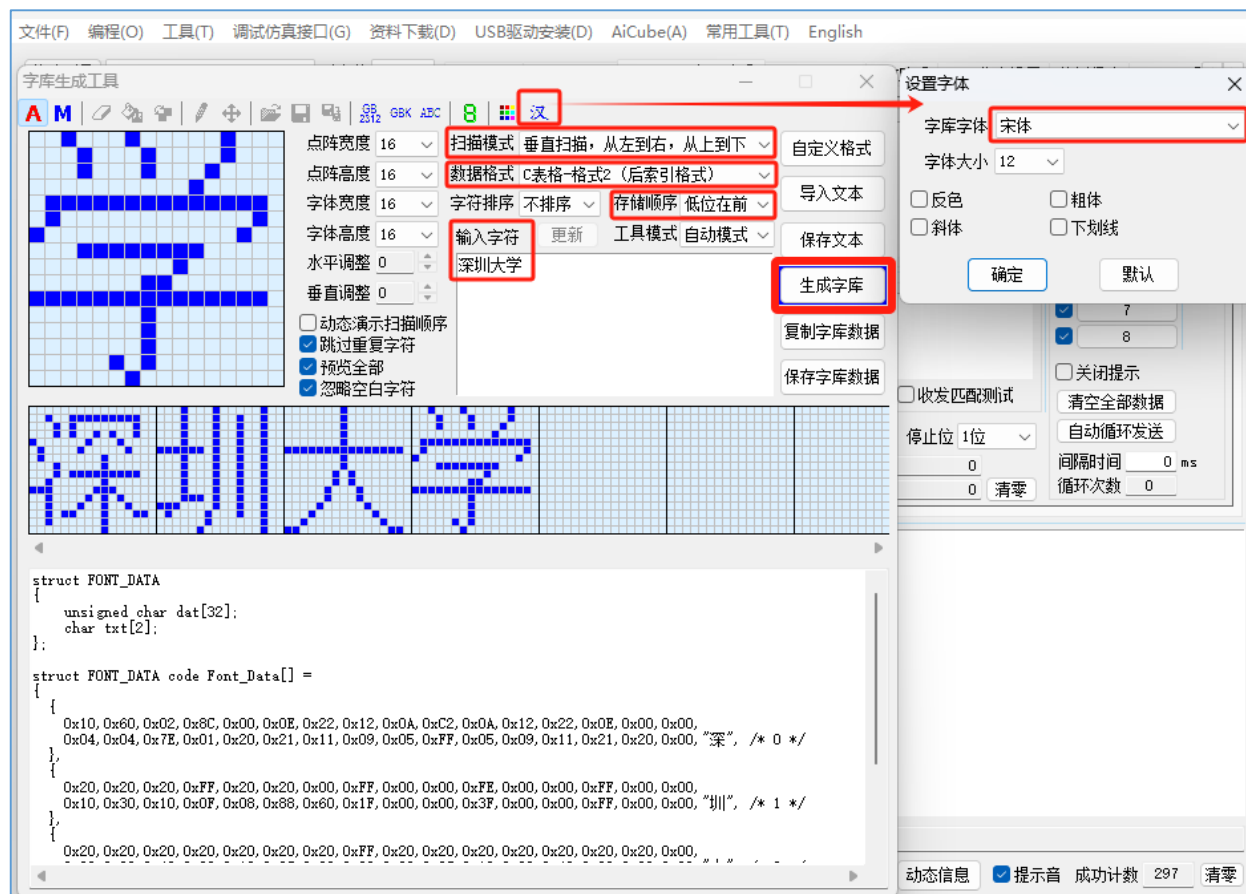
将点阵图片生成的 C 代码数据拷贝出来, 本例是将此段代码放在“pic.h”中, 形成表格数据, 方便程序加载并在 OLED 上显示出来。

## b) Aiapp-ISP 系统软件字库生成工具的使用

打开 Aiapp-ISP 系统软件, 点击菜单栏“工具”→“字库生成工具”, 打开“字库生成工具”浮窗:

- 在“输入字符”区输入想要取模的文字, 本例输入“深圳大学”
- 点击第一行的工具栏上的“汉”按钮, 打开“设置字体”浮窗, 字库字体选择“宋体”
- 扫描模式: 选择“垂直扫描, 从左到右, 从上到下”
- 数据格式: 选择“C 表格-格式 2 (后索引模式)”
- 存储顺序: 选择“低位在前”
- 点击“生成字库”

就会生成相应的点阵字库代码, 如下图:



将点阵字库生成的 C 代码拷贝出来，本例是将此段代码放在“font.h”中，形成表格数据，方便程序查询并提取字库。

### c) 使用 Aiapp-ISP 系统软件图片取模工具生成 ASCII 的点阵字符 C 代码

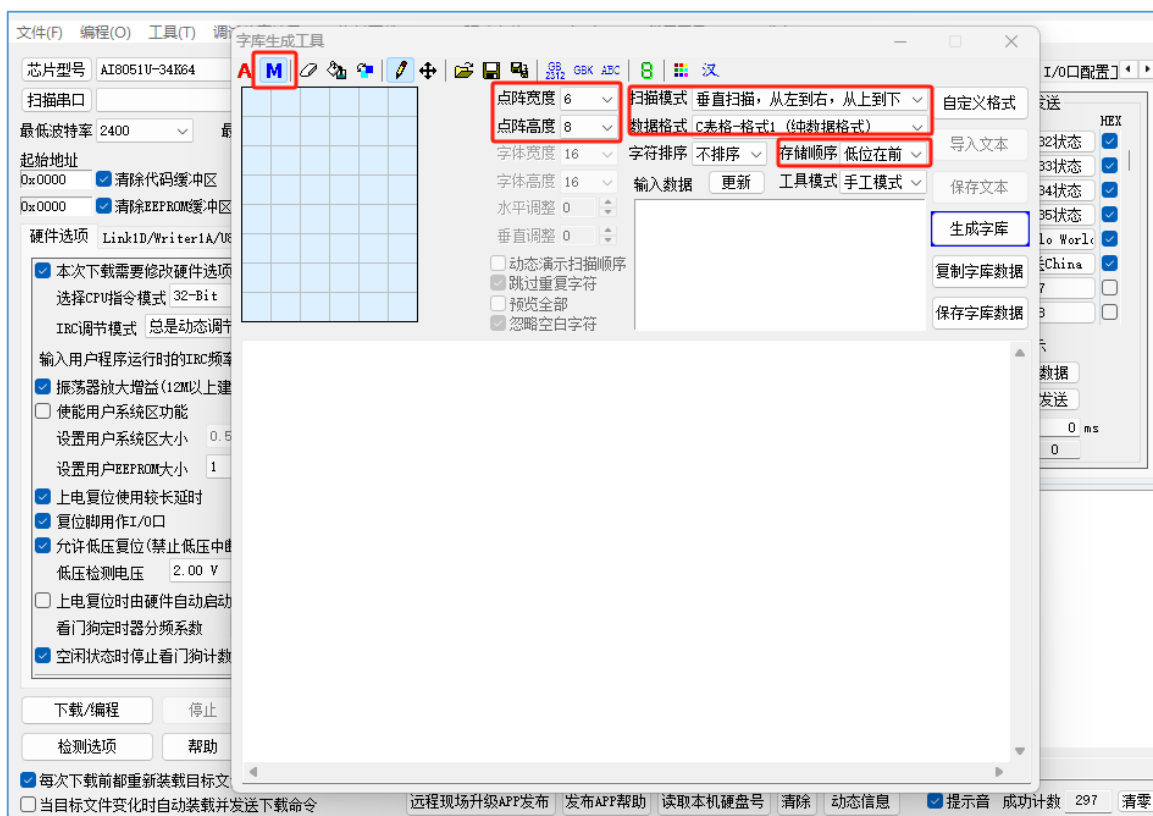
在 Aiapp-ISP 系统软件的字库生成工具中，我们除了可以像上述操作步骤生成标准点阵字符 C 代码，还可以自己手动描点生成自己想要的点阵字符 C 代码。

打开 Aiapp-ISP 系统软件，点击菜单栏“工具”→“字库生成工具”，打开“字库生成工具”浮窗。

先进行相应的设置：

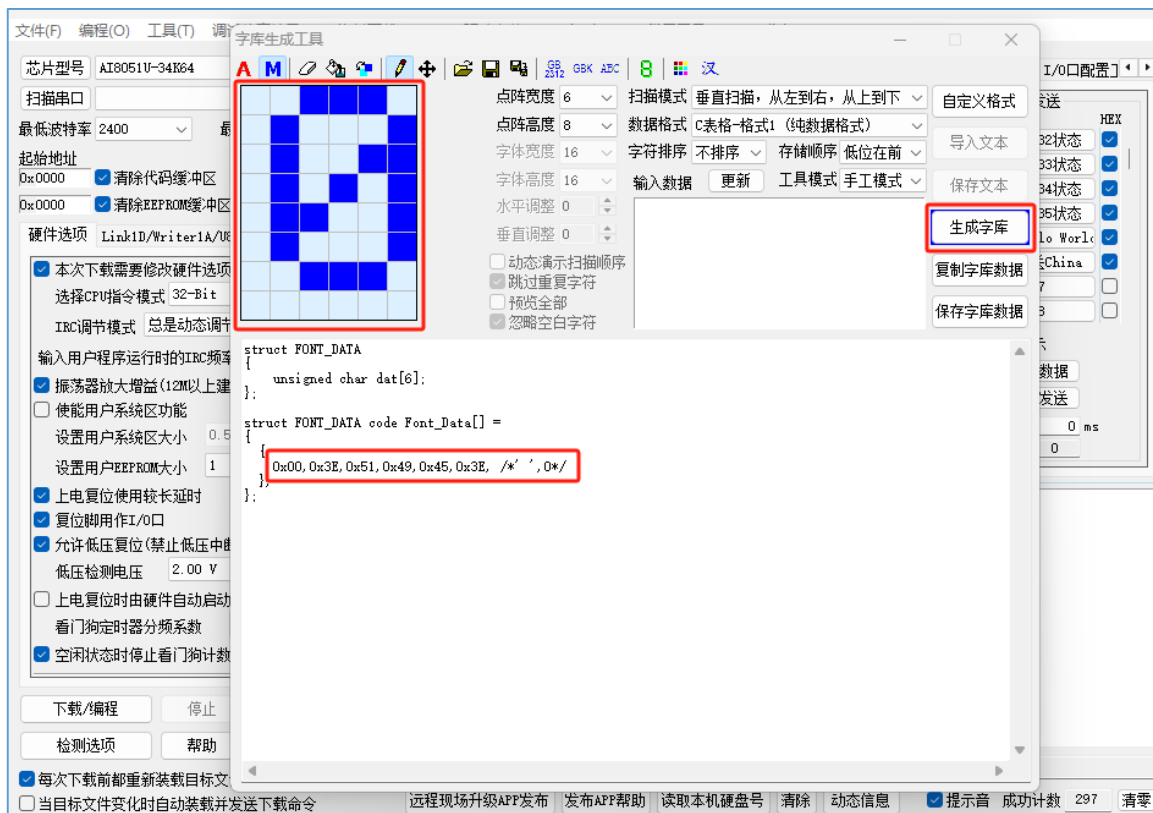
- 点击“字库生成工具”浮窗第一行的“M”按钮，清空字符点阵区域
- 确定点阵宽度：6；点阵高度：8（本例中有 6\*8 的点阵 ASCII 字符 C 代码数据和 8\*16 的点阵 ASCII 字符代码数据）
- 扫描模式：选择“垂直扫描，从左到右，从上到下”
- 数据格式：选择“C 表格-格式 1（纯数据格式）”
- 存储顺序：选择“低位在前”

如下图：



现在手动用鼠标在窗口左上角点阵描图区域, 进行描图。

以 ASCII 码的“0”为例, 进行如下图中的描绘, 再点击“生成字库”按钮, 即可生成相应的字库 C 代码数据。以此类推, 生成自己想要的全部 ASCII 码的点阵字库 C 代码数据。

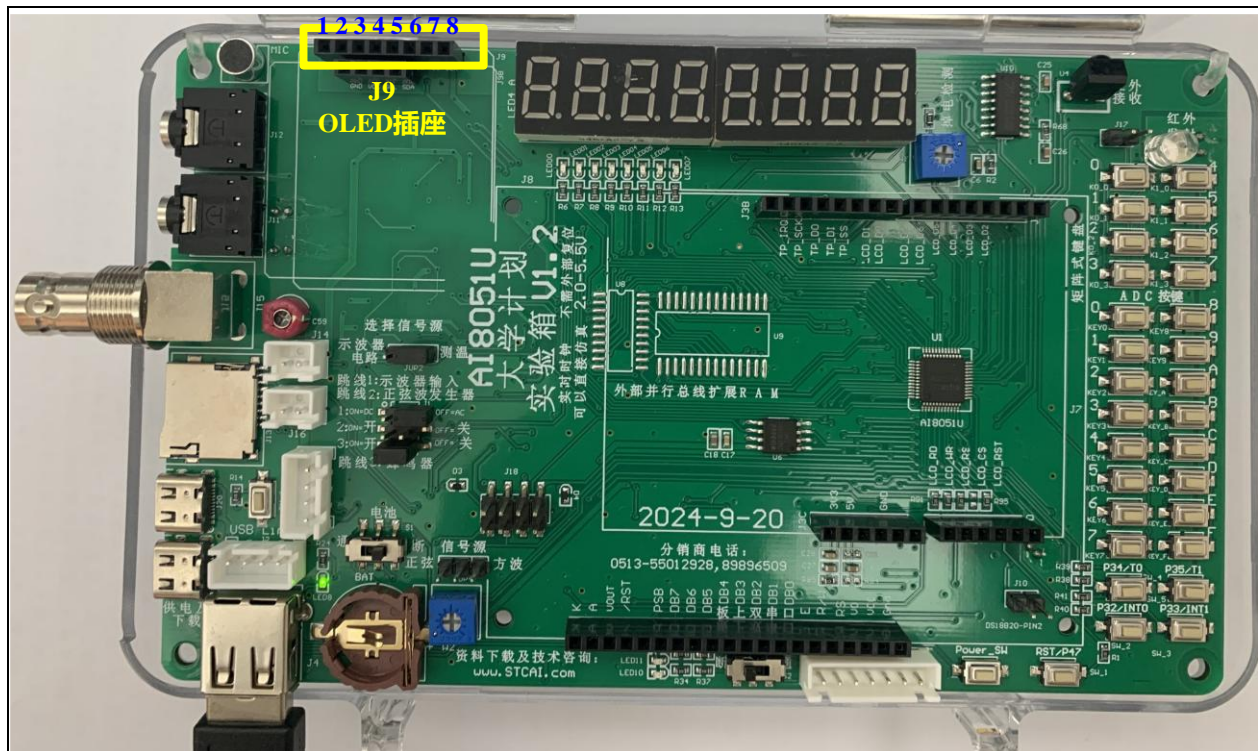


将点阵字库生成的 C 代码拷贝出来, 本例是将此段代码放在“ascii.h”中, 形成表格数据, 方便程序查询并提取字库。



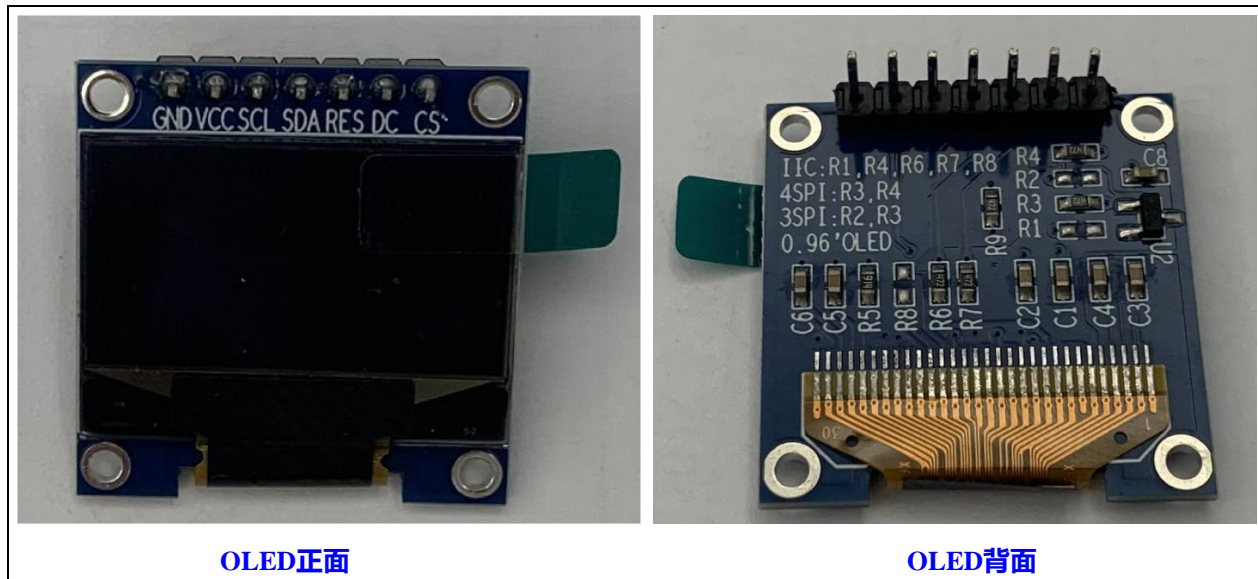
#### d) 认识 Ai8051U 实验箱和 OLED 屏

### Ai8051U 实验箱:



J9: OLED 插座。Ai8051U 实验箱上的 J9 插座可插 7 线或 8 线的 OLED 屏，本实验使用的 OLED 屏是 7 线，第 8 脚浮空。详见 Ai8051U 实验箱中 J9 原理图。

## OLED 屏



### e) Keil 环境下多文件项目管理说明

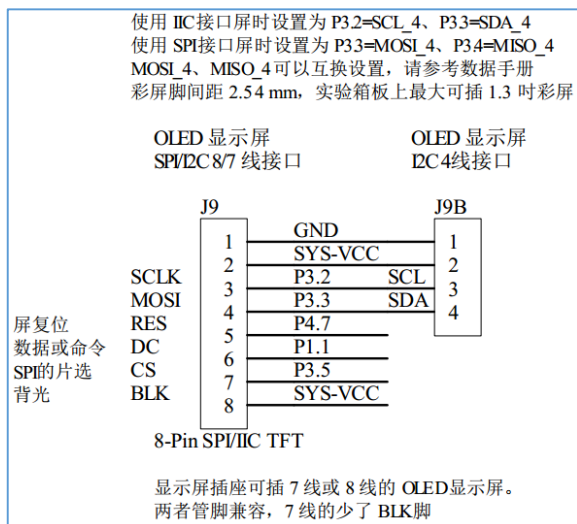
当项目的功能比较复杂时，就需要在 Keil 中建立多文件项目，以方便分工合作、代码复用、模块化管理、增强可读性和可维护性。

- 比较好的建议是将项目功能模块化，不同模块的实现代码放在不同的.c 文件中。

- 一般建议是一个模块对应一个.c 程序文件和一个.h 头文件
- 模块的初始化函数以及相关的数据处理函数都在.c 文件中实现
- 与模块相关的全局变量也必须在.c 文件中进行定义, **一定不能在.h 文件中定义变量**
- 如果有其他模块需要使用本模块定义的变量或函数, 则这些函数和变量都需要在.h 文件中声明。
- 特别提醒: 在.h 文件中声明外部变量必须使用 **extern** 关键字, 否则就变成变量定义了, 这样会出现变量重复定义的错误
- 为防止头文件被多次包含而产生错误或者警告, 在头文件中使用类似如下的条件编译组合语句, 可避免在同一个.c 文件中对同一个.h 头文件进行多次包含:

```
#ifndef XXXX
#define XXXX
...
#endif
```

## 25.10.2. 原理图



## 25.10.3. 实验程序代码

### a) main.c -- 程序主函数

```
//main.c
//程序主函数
//SPI 驱动 OLED 显示图片、图形、字符

#include "config.h"    //头文件中已包含 ai8051u.h, ai_usb.h 以及其他头文件
#include "pic.h"

void delay_ms(unsigned int ms);

void main()
{
    SYS_Init();        //系统初始化
```

```

    OLED_Init();           //初始化
    OLED_CLS();            //清屏
//    OLED_Light(0);        //最低亮度
//    OLED_Light(0xFF);     //最高亮度
    OLED_Light(0x80);      //中间亮度

while (1)
{
    OLED_BMP(0,0,128,8,OLED12864_IMG0[0],1);    //显式 BMP 图
    delay_ms(2000);

    OLED_BMP(0,0,128,8,OLED12864_IMG1[0],0);
    OLED_String(0,0,"AiMCU",0,1);                //图上叠加文字...
    delay_ms(2000);

    OLED_CLS();           //清屏
    OLED_String(5,1,"深圳大学+STC",0,1);          //中西文混合显示
    OLED_String(5,3,"深圳大学+STC",1,1);          //正显/反显
    OLED_Str5x7(16,6,"Ai8051U-34K64",0,1);        //正显
    OLED_Str5x7(16,7,"Ai8051U-34K64",1,1);        //反显
    delay_ms(2000);

    OLED_CLS();           //清屏
    OLED_LineH(10,10,108,0);    //画一水平线 作图算法...
    OLED_LineV(10,10,44,0);     //画一垂直线
    OLED_Line(10,10,118,54,0);  //两点一线 不同方向画线...
    OLED_Line(10,54,118,10,0);  //两点一线
    OLED_Line(118,54,118,10,0); //两点一线
    OLED_Line(118,54,10,54,0);  //两点一线
    OLED_Box(0,0,127,63,1);     //画一方框
    OLED_Circle(64,32,30,1);    //画一个圆
    delay_ms(2000);
}
}

void SYS_Init(void)    //系统初始化函数
{
    EAXFR = 1;          //使能访问扩展 XFR
    WTST = 0x00;        //设置最快速度访问程序代码
    CKCON = 0x00;       //设置最快速度访问内部 XDATA

    P0M1 = 0x00;    P0M0 = 0x00;    //设置为准双向口
    P1M1 = 0x00;    P1M0 = 0x00;    //设置为准双向口
    P2M1 = 0x00;    P2M0 = 0x00;    //设置为准双向口
    P3M1 = 0x00;    P3M0 = 0x2c;    //设置为准双向口, P3.2,P3.3,P3.5 设置为推挽输出
    P4M1 = 0x00;    P4M0 = 0x80;    //设置为准双向口, P4.7 设置为推挽输出

```

```

P5M1 = 0x00;    P5M0 = 0x00;    //设置为准双向口
P6M1 = 0x00;    P6M0 = 0x00;    //设置为准双向口
P7M1 = 0x00;    P7M0 = 0x00;    //设置为准双向口

P3SR = 0x00;    //电平转换速度快（改善 IO 口高速翻转信号）
P3DR = 0xff;    //端口驱动电流增强
// EA = 1; //使能全局中断
}

//=====
// 函数: void delay_ms(unsigned int ms)
// 描述: 毫秒级延时函数。
// 参数: ms,要延时的 ms 数,自动适应主时钟.
//=====
void delay_ms(unsigned int ms)
{
    unsigned int i;
    do{
        i = MAIN_Fosc / 6000;
        while(--i);
    } while(--ms);
}

```

## b) oled.c -- SPI 驱动 OLED 显示图片、图形、字符

```

//oled.c
//SPI 驱动 OLED 显示图片、图形、字符

#include "config.h"    //头文件中已包含 ai8051u.h 以及其他头文件
#include "font.h"
#include "ascii.h"

#define UseSPI 1    //0: IO 模拟 SPI; 1: 使用硬件 SPI

//OLED 屏引脚定义
sbit    OLED_SCK  = P3^2;    //D0 SPI or IIC 的时钟脚
sbit    OLED_SDA  = P3^3;    //D1 SPI or IIC 的数据脚
sbit    OLED_RES  = P4^7;    //RES  复位脚, 低电平复位
sbit    OLED_DC   = P1^1;    //DC数据或命令脚
sbit    OLED_CS   = P3^5;    //CS 片选脚

//SPI 定义
sbit    SPI_SCK = P3^2;    //PIN6
sbit    SPI_MOSI = P3^3;    //PIN5

```



```
unsigned char xdata ShowBuf[8][128];    //OLED 全局缓存
```

```
extern void delay_ms(unsigned int ms);
```

```
//=====
// 函数名称: OLED_WR_CMD(unsigned char Cmd)
// 函数功能: OLED_写命令字
//=====
void OLED_WR_CMD(unsigned char Cmd)
{
#ifdef UseSPI==0
    unsigned char i;
#endif

    OLED_SCK= 0;    //时钟低;
    OLED_DC = 0;    //写命令
    OLED_CS = 0;    //片选中

#ifdef UseSPI
    SPDAT = Cmd;
    while(SPIF == 0);
    SPIF = 1;        //清 SPIF 标志
    WCOL = 1;        //清 WCOL 标志
#else
    for(i=0;i<8;i++)
    {
        if(Cmd&0x80)
            OLED_SDA=1;
        else
            OLED_SDA=0;

        Cmd <<= 1;    //左移位 //
        OLED_SCK = 1;
        OLED_SCK = 0;
    }
#endif
}

#ifdef

    OLED_CS = 1;        //片不选
}

//=====
```

```
// 函数名称: OLED_WR_DAT(unsigned char Dat)
// 函数功能: OLED_写数据字
//=====
```

```
void OLED_WR_DAT(unsigned char Dat)
{
```

```
#if(UseSPI==0)
    unsigned char i;
#endif

    OLED_SCK=0;    //时钟低;
    OLED_DC = 1;    //写数据
    OLED_CS = 0;    //片选中

#if(UseSPI)
    SPDAT = Dat;
    while(SPIF == 0);
    SPIF = 1;        //清 SPIF 标志
    WCOL = 1;        //清 WCOL 标志
#else
    for(i=0;i<8;i++)
    {
        if(Dat&0x80)
            OLED_SDA=1;
        else
            OLED_SDA=0;

        Dat <<= 1;    //左移位
        OLED_SCK = 1;
        OLED_SCK = 0;
    }
#endif

    OLED_CS = 1;    //片不选
}

//=====
// 函数名称: OLED_Disp(unsigned char OnOff)
// 函数功能: OLED 开关显示 参数: 1:开 0:关
//=====
void OLED_Disp(unsigned char On) //开关显示
{
    if(On==1)
    {
        OLED_WR_CMD(0x8D); //电荷泵使能
        OLED_WR_CMD(0x14); //开启电荷泵
        OLED_WR_CMD(0xAF); //点亮屏幕
    }
    else
    {
        OLED_WR_CMD(0x8D); //电荷泵使能
```

```

        OLED_WR_CMD(0x10);    //关闭电荷泵
        OLED_WR_CMD(0xAF);    //关闭屏幕
    }
}

//=====
// 函数名称: OLED_Init() 显示屏初始化函数
//=====

void OLED_Init(void)
{
    #if(UseSPI)
        P_SW1 = (P_SW1 & ~(3<<2)) | (3<<2);    //IO 口切换.  0: P1.4 P1.5 P1.6 P1.7,  1: P2.4 P2.5 P2.6 P2.7,
                                                //                2: P4.0 P4.1 P4.2 P4.3,  3: P3.5 P3.4 P3.3 P3.2

        HSSPI_CFG2 |= 0x40;    //交换 MOSI 与 MISO 脚
        SSIG = 1;    //忽略 SS 引脚功能, 使用 MSTR 确定器件是主机还是从机
        SPEN = 1;    //使能 SPI 功能
        DORD = 0;    //先发送/接收数据的高位 ( MSB )
        MSTR = 1;    //设置主机模式
        CPOL = 0;    //SCLK 空闲时为低电平, SCLK 的前时钟沿为上升沿, 后时钟沿为下降沿
        CPHA = 0;    //数据 SS 管脚为低电平驱动第一位数据并在 SCLK 的后时钟沿改变数据
        SPCTL = (SPCTL & ~3) | 3; //SPI 时钟频率选择, 0: 4T,  1: 8T,   2: 16T,   3: 2T
        SPI_SCK = 0;    //时钟信号置低
        SPI_MOSI = 1;
        SPIF = 1;    //清 SPIF 标志
        WCOL = 1;    //清 WCOL 标志
    #endif

    OLED_SCK = 0;    //时钟端低
    OLED_SDA = 1; delay_ms(100);    //数据端高
    OLED_RES = 1; delay_ms(200);
    OLED_RES = 0; delay_ms(200);    //复位
    OLED_RES = 1;

    OLED_WR_CMD(0xAE);    //--turn off oled panel
    OLED_WR_CMD(0x00);    //--set low column address
    OLED_WR_CMD(0x10);    //--set high column address
    OLED_WR_CMD(0x40);    //--set start line address  Set Mapping RAM Display Start Line (0x00~0x3F)
    OLED_WR_CMD(0x81);    //--set contrast control register
    OLED_WR_CMD(0xCF);    //--Set SEG Output Current Brightness 亮度...
    OLED_WR_CMD(0xA1);    //--Set SEG/Column Mapping      0xA0 左右反转 0xA1 正常
    OLED_WR_CMD(0xC8);    //--Set COM/Row Scan Direction 0xC0 上下反转 0xC8 正常
    OLED_WR_CMD(0xA6);    //--set normal display
    OLED_WR_CMD(0xA8);    //--set multiplex ratio(1 to 64)
    OLED_WR_CMD(0x3f);    //--1/64 duty
    OLED_WR_CMD(0xD3);    //--set display offsetShift Mapping RAM Counter (0x00~0x3F)

```

```

    OLED_WR_CMD(0x00);    //--not offset
    OLED_WR_CMD(0xd5);    //--set display clock divide ratio/oscillator frequency
    OLED_WR_CMD(0x80);    //--set divide ratio, Set Clock as 100 Frames/Sec
    OLED_WR_CMD(0xD9);    //--set pre-charge period
    OLED_WR_CMD(0xF1);    //--Set Pre-Charge as 15 Clocks & Discharge as 1 Clock
    OLED_WR_CMD(0xDA);    //--set com pins hardware configuration
    OLED_WR_CMD(0x12);
    OLED_WR_CMD(0xDB);    //--set vcomh
    OLED_WR_CMD(0x40);    //--Set VCOM Deselect Level
    OLED_WR_CMD(0x20);    //--Set Page Addressing Mode (0x00/0x01/0x02)
    OLED_WR_CMD(0x00);
    OLED_WR_CMD(0x8D);    //--set Charge Pump enable/disable
    OLED_WR_CMD(0x14);    //--set(0x10) disable
    OLED_WR_CMD(0xA4);    //--Disable Entire Display On (0xa4/0xa5)
    OLED_WR_CMD(0xA6);    //--Disable Inverse Display On (0xa6/a7)
    OLED_WR_CMD(0xAF);
    OLED_Dis(1);          //开显示
}

```

```

//=====

```

```

// 函数名称: OLED_SetXY

```

```

// 函数功能: OLED 设置显示位置

```

```

// 入口参数: X-水平坐标 0..127,Y-垂直坐标 0..63

```

```

//=====

```

```

void OLED_SetXY(unsigned char X, unsigned char Y)

```

```

{
    OLED_WR_CMD((u8)(0xB0+Y));
    OLED_WR_CMD((u8)(((0xF0&X)>>4)|0x10));
    OLED_WR_CMD((u8)(((0x0f&X)|0x01)));
}

```

```

//=====

```

```

// 函数名称: OLED_Show

```

```

// 函数功能: OLED 刷新显示(将显示缓存的数据刷到显示屏显示出来)

```

```

//=====

```

```

void OLED_Show(void)

```

```

{
    unsigned char xdata i,n;

    for(i=0;i<8;i++)          //8 行(页)循环...
    {
        OLED_WR_CMD((u8)(0xB0+i));    //设置行起始地址
        OLED_WR_CMD(0x00);            //设置低列起始地址
        OLED_WR_CMD(0x10);            //设置高列起始地址
    }
}

```

```
        for(n=0;n<128;n++)                //每行(页)128 点(列)
            OLED_WR_DAT(ShowBuf[i][n]);    //写数据...
    }
}

//=====
// 函数名称: OLED_CLS 清屏函数
// 函数功能: OLED 刷新显示(将显示缓存的 0 数据刷到显示屏显示出来)
//=====
void OLED_CLS(void)
{
    memset(ShowBuf,0,128*8);    //清空缓存
    OLED_Show();                //刷新显示
}

//=====
// 函数名称: OLED_Light(unsigned char num)
// 函数功能: OLED 亮度级设置
//=====
void OLED_Light(unsigned char Level)
{
    OLED_WR_CMD(0x81);
    OLED_WR_CMD(Level);
    OLED_WR_CMD(0xDB);        //--set vcomh
    OLED_WR_CMD(0x20);        //Set VCOM Deselect Level
}

//=====
// 函数名称: OLED_Point
// 函数功能: OLED 显示一个点
// 入口参数: X: 水平点 0..127 Y:垂直点 0..63
//=====
void OLED_Point(unsigned char X,unsigned char Y,u8 mode)
{
    ShowBuf[Y/8][X] |= 1<<(Y%8);    //垂直 8 点,高位在下,低位在上.
    if(mode)
        OLED_Show();
}

//=====
// 函数名称: OLED_BMP
// 函数功能: OLED 缓存写入 BMP 格式图片
// 入口参数: X: 水平起点 0..127 Y:垂直起点 0..63 W: 宽度 H:高度 *BMP 图片数组
//=====
void OLED_BMP(unsigned char X, unsigned char Y,unsigned char W, unsigned char H,unsigned char BMP[],u8
```



```
mode)
{
    unsigned int xdata num=0;
    unsigned char i,l;

    for( i=0;i<H;i++ )
    {
        for(l=0;l<W;l++)
        {
            ShowBuf[Y+i][X+l] = BMP[num++];
        }
    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 画一条水平线
//=====
void OLED_LineH(unsigned char X,unsigned char Y,unsigned char W,u8 mode)
{
    unsigned char i;

    for(i=0;i<W;i++)
    {
        OLED_Point((u8)(X+i),Y,0);
    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 画一条垂直线
//=====
void OLED_LineV(unsigned char X,unsigned char Y,unsigned char H,u8 mode)
{
    unsigned char i;

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

```

    {
        OLED_Point(X,(u8)(Y+i),0);
    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 函数名称: OLED_BuffShowPoint
// 函数功能: OLED 显示一条线
// 入口参数: X0 起点 Y0 起点 X1 终点 Y1 终点
//=====

void OLED_Line( unsigned char X0, unsigned char Y0, unsigned char X1,unsigned char Y1,u8 mode)
{
    unsigned char x,y;
    if(X0>X1) //为从左到右而交换坐标.
    {
        x=X0;X0=X1;X1=x;
        y=Y0;Y0=Y1;Y1=y;
    }

    if(X0!=X1)
    {
        for(x=0;x<(X1-X0);x++ )
        {
            if(Y1>Y0)
                OLED_Point((u8)(X0+x),(u8)(Y0+(u16)(Y1-Y0)*(u16)x/(u16)(X1-X0)),0);
            else
                OLED_Point((u8)(X0+x),(u8)(Y0-(u16)(Y0-Y1)*(u16)x/(u16)(X1-X0)),0);
        }
    }
    else
    {
        if(Y0>Y1)
        {
            for(y=Y1; y<=Y0; y++ )
                OLED_Point(X0,y,0);
        }
        else
        {
            for(y=Y0; y<=Y1; y++ )
                OLED_Point(X0,y,0);
        }
    }
}

```

```

    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 画一个方框
//=====

void OLED_Box(unsigned char X,unsigned char Y,unsigned char W,unsigned char H,u8 mode)
{
    OLED_LineH(X,Y,W,0);
    OLED_LineV(X,Y,H,0);
    OLED_LineH(X,(u8)(Y+H),W,0);
    OLED_LineV((u8)(X+W),Y,H,0);

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 函数名称: OLED_Circle
// 函数功能: OLED 显示一个圆形
// 入口参数: X 点 Y 点 r:半径
//      turn: 0:正显示 1:反显示 mode: 1:立即刷新显示 0:仅写缓存
//=====

void OLED_Circle(unsigned char X,unsigned char Y,unsigned char r,u8 mode)
{
    int a, b, di;
    a = 0;
    b = r;
    di = 3 - (r << 1);    //判断下个点位置的标志

    while (a <= b)
    {
        OLED_Point((u8)(X+a),(u8)(Y-b),0);    //5
        OLED_Point((u8)(X+b),(u8)(Y-a),0);    //0
        OLED_Point((u8)(X+b),(u8)(Y+a),0);    //4
        OLED_Point((u8)(X+a),(u8)(Y+b),0);    //6
        OLED_Point((u8)(X-a),(u8)(Y+b),0);    //1
        OLED_Point((u8)(X-b),(u8)(Y+a),0);    //3
        OLED_Point((u8)(X-a),(u8)(Y-b),0);    //2
        OLED_Point((u8)(X-b),(u8)(Y-a),0);    //7
    }
}

```

```
a++;

//使用 Bresenham 算法画圆
if (di < 0)
{
    di += 4 * a + 6;
}
else
{
    di += 10 + 4 * (a - b);
    b--;
}
}

if(mode)
{
    OLED_Show();
}
}

//=====
// 函数名称: OLED_A08
// 函数功能: OLED 显示一个 6*8 字符
// 入口参数: X: 水平坐标列 0..127 Y: 垂直坐标行(页) 0..7
//      turn: 0:正显示 1:反显示 mode: 1:立即刷新显示 0:仅写缓存
//=====
void OLED_A08(unsigned char X,unsigned char Y,char Ch,u8 turn,u8 mode)    //OLED 显示一个 5*7 字符
{
    unsigned char i;
    for( i=0;i<6;i++)
    {
        if(turn==0)
            ShowBuf[Y][X+i]= ASC8[Ch-' '][i];
        else
            ShowBuf[Y][X+i]=~ASC8[Ch-' '][i];
    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 函数名称: OLED_A16
// 函数功能: OLED 显示一个 8*16 字符
// 入口参数: X: 水平坐标列 0..127 Y: 垂直坐标行(页) 0..7
```

```
//      turn: 0:正显示 1:反显示  mode: 1:立即刷新显示 0:仅写缓存
//=====
void OLED_A16(unsigned char X,unsigned char Y,char Ch,u8 turn,u8 mode)    //OLED 显示一个 8*16 字
符
{
    unsigned char i,j;
    for( j=0;j<2;j++)
    {
        for( i=0;i<8;i++)
        {
            if(turn==0)
                ShowBuf[Y+j][X+i]= ASC16[Ch-' '][i+j*8];
            else
                ShowBuf[Y+j][X+i]=~ASC16[Ch-' '][i+j*8];
        }
    }

    if(mode)
    {
        OLED_Show();
    }
}
//=====
// 函数名称: OLED_HZ16
// 函数功能: OLED 显示一个 16*16 汉字
// 入口参数: X: 水平坐标列 0..127  Y: 垂直坐标行(页) 0..7
//      turn: 0:正显示 1:反显示  mode: 1:立即刷新显示 0:仅写缓存
//=====
void OLED_HZ16(unsigned char X,unsigned char Y,char *Hz,u8 turn,u8 mode)
{
    unsigned char i,j;
    unsigned int m;

    //查找字库汉字表
    for(m=0;m<FONT_NUM;m++)
    {
        if((Font_Data[m].txt[0]==Hz[0])&&(Font_Data[m].txt[1]==Hz[1]))
        {
            for( j=0;j<2;j++)
            {
                for( i=0;i<16;i++)
                {
                    if(turn==0)
                        ShowBuf[Y+j][X+i]= Font_Data[m].dat[i+j*16];
                    else

```



```

        ShowBuf[Y+j][X+i]=~Font_Data[m].dat[i+j*16];
    }
}
}

if(mode)
{
    OLED_Show();
}
}

//=====
// 函数名称: OLED_Str5x7
// 函数功能: OLED 显示 5*7 点阵字符串
// 入口参数: X: 水平坐标列 0..127  Y: 垂直坐标行(页) 0..7
//      turn: 0:正显示 1:反显示  mode: 1:立即刷新显示 0:仅写缓存
//=====
void OLED_Str5x7(unsigned char X,unsigned char Y,char *s,u8 turn,u8 mode)
{
    while(*s != '\0')    //字符串不为空, 循环
    {
        OLED_A08(X,Y,*s,turn,0);
        X += 6;
        s++;
    }

    if(mode)
    {
        OLED_Show();
    }
}

//=====
// 函数名称: OLED_String
// 函数功能: OLED 显示字符串。
// 入口参数: X: 水平坐标列 0..127  Y: 垂直坐标行(页) 0..7
//      turn: 0:正显示 1:反显示  mode: 1:立即刷新显示 0:仅写缓存
//=====
void OLED_String(unsigned char X,unsigned char Y,char *s,u8 turn,u8 mode)
{
    char hz[2];
    while(*s != '\0')    //字符串不为空, 循环
    {
        if ((unsigned char)*s < 0x80)    //根据输入数据的大小判断是字符还是汉字,
        {
            OLED_A16(X,Y,*s,turn,0);

```

```
        X += 8;
        s++;
    }
    else
    {
        hz[0] = *s;
        hz[1] = *(s+1);
        OLED_HZ16(X,Y,hz,turn,0);
        X += 16;
        s += 2;
    }

    if(X>127)        //行
    {
        X = 0;
        Y += 2;
    }
}

if(mode)
{
    OLED_Show();
}
}
```

### c) config.h -- 项目配置的头文件

```
#ifndef __CONFIG_H__        //防止头文件被重复包含
#define __CONFIG_H__

#define HIRC                40000000UL
#define FOSC                40000000UL
#define SYSCLK              FOSC
#define MAIN_Fosc          FOSC

#include <ai8051u.h>        //包含外部头文件
#include <string.h>
#include <stdlib.h>

#include "def.h"            //包含项目头文件
#include "oled.h"

void SYS_Init(void);        //函数声明

#endif
```

## d) oled.h -- 项目配置文件的头文件

```
#ifndef __OLED_H__          //防止头文件被重复包含
#define __OLED_H__

/*  OLED Pixel */
#define WIDTH 128
#define HEIGHT 64
#define PAGES 8

/*  OLED Brightness */
#define BRIGHTNESS_MIN 1
#define BRIGHTNESS_MAX 25

/*  OLED Driver */
void OLED_Init(void);      //初始化
void OLED_Show(void);     //显示(缓存数据刷屏)
void OLED_Disb(unsigned char On);    //开关显示
void OLED_CLS(void);      //清屏
void OLED_Light(unsigned char Level); //亮度设置
void OLED_SetXY(unsigned char X,unsigned char Y);      //设定坐标
void OLED_Point(unsigned char X,unsigned char Y,u8 mode); //画一个点
void OLED_LineH(unsigned char X,unsigned char Y,unsigned char W,u8 mode); //画一条水平线
void OLED_LineV(unsigned char X,unsigned char Y,unsigned char H,u8 mode); //画一条垂直线
void OLED_Line( unsigned char X0, unsigned char Y0, unsigned char X1,unsigned char Y1,u8 mode);
void OLED_Box(unsigned char X,unsigned char Y,unsigned char W,unsigned char H,u8 mode); //画一个方框
void OLED_Circle(unsigned char X, unsigned char Y, unsigned char r,u8 mode); //画一个圆
void OLED_BMP(unsigned char X,unsigned char Y,unsigned char W,unsigned char H,unsigned char BMP[],u8 mode);
void OLED_A08(unsigned char X,unsigned char Y,char Ch, u8 turn,u8 mode); //OLED 显示一个 5*7 字符
void OLED_A16(unsigned char X,unsigned char Y,char Ch, u8 turn,u8 mode); //OLED 显示一个 8*16 字符
void OLED_HZ16(unsigned char X,unsigned char Y,char *Hz, u8 turn,u8 mode); //OLED 显示一个 16*16 汉字
void OLED_Str5x7(unsigned char X,unsigned char Y,char *s,u8 turn,u8 mode); //OLED 显示 5*7 点阵字符串
void OLED_String(unsigned char X,unsigned char Y,char *s,u8 turn,u8 mode); //OLED 显示 16 点阵高字符串

#endif
```

## e) font.h -- 项目配置文件的头文件

```
#ifndef __FONT_H__          //防止头文件被重复包含
#define __FONT_H__

/* 自定义中文字库, 16*16, 宋体 */
#define FONT_NUM    sizeof(Font_Data)/sizeof(Font_Data[0])    //自动统计汉字数目
```

```

struct FONT_DATA
{
    unsigned char dat[32];
    char txt[2];
};

struct FONT_DATA code Font_Data[] =
{
    {
        0x10,0x60,0x02,0x8C,0x00,0x0E,0x22,0x12,0x0A,0xC2,0x0A,0x12,0x22,0x0E,0x00,0x00,
        0x04,0x04,0x7E,0x01,0x20,0x21,0x11,0x09,0x05,0xFF,0x05,0x09,0x11,0x21,0x20,0x00,"深", /* 0 */
    },
    {
        0x20,0x20,0x20,0xFF,0x20,0x20,0x00,0xFF,0x00,0x00,0xFE,0x00,0x00,0xFF,0x00,0x00,
        0x10,0x30,0x10,0x0F,0x08,0x88,0x60,0x1F,0x00,0x00,0x3F,0x00,0x00,0xFF,0x00,0x00,"圳", /* 1 */
    },
    {
        0x20,0x20,0x20,0x20,0x20,0x20,0x20,0xFF,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x00,
        0x80,0x80,0x40,0x20,0x10,0x0C,0x03,0x00,0x03,0x0C,0x10,0x20,0x40,0x80,0x80,0x00,"大", /* 2 */
    },
    {
        0x40,0x30,0x11,0x96,0x90,0x90,0x91,0x96,0x90,0x90,0x98,0x14,0x13,0x50,0x30,0x00,
        0x04,0x04,0x04,0x04,0x04,0x44,0x84,0x7E,0x06,0x05,0x04,0x04,0x04,0x04,0x04,0x00,"学", /* 3 */
    },
};

#endif

```

## f) pic.h -- 项目配置文件的头文件

```

#ifndef __PIC_H__          //防止头文件被重复包含
#define __PIC_H__

```

```

u8 code OLED12864_IMG0[8][128] =
{
    0x00,0x00,0x00,0xe0,0xf0,0xf8,0xf8,0x9c,0x1c,0x1c,0x1c,0x1c,0x1c,0x38,0x38,0x18,
    0x00,0x10,0x18,0x1c,0x1c,0xdc,0xfc,0xfc,0xfc,0x1c,0x1c,0x1c,0x1c,0x0c,0x04,0xc0,
    0xe0,0xf0,0x78,0x38,0x38,0x1c,0x1c,0x1c,0x1c,0x3c,0x38,0x08,0x00,0x00,0xc0,0xf0,
    0xfc,0xfc,0x7c,0xfc,0xf0,0x00,0x00,0xc0,0xc0,0xe0,0xf8,0xfc,0xfc,0xfc,0x7c,0x0c,
    0x00,0xc0,0xc0,0xf0,0xf0,0x78,0x38,0x1c,0x1c,0x1c,0x1c,0x1c,0x38,0x38,0x08,0x00,
    0x80,0xf0,0xfc,0xfc,0xfc,0x00,0x00,0x00,0x00,0x00,0xc0,0xfc,0xfc,0xfc,0x0c,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
    0x00,0x70,0x70,0xf0,0xe1,0xe3,0xe3,0xe3,0xe7,0x7f,0x7f,0x3e,0x1c,0x00,0x00,
    0x00,0x00,0x00,0xf0,0xfe,0xff,0x7f,0x0f,0x03,0x00,0x00,0x00,0x00,0x00,0x1f,0x3f,

```

0x7f,0x7f,0xf0,0xe0,0xe0,0xe0,0xe0,0x70,0x70,0x10,0x00,0x00,0xc0,0xf0,0xff,0xff,  
0x7f,0x03,0x00,0x03,0x0f,0x0f,0x0f,0x03,0xc3,0xf0,0xff,0xff,0x7f,0x03,0x00,0x00,  
0x1f,0x3f,0x3f,0x7f,0xf0,0xf0,0xe0,0xe0,0xe0,0x70,0x70,0x70,0x00,0x00,0x00,0x3c,  
0x3f,0x7f,0xff,0xf3,0xe0,0xe0,0xe0,0xe0,0x70,0x7e,0x3f,0x1f,0x0f,0x03,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x10,0x18,0x30,0x60,0xe0,  
0x40,0x00,0xf0,0xf0,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,  
0x30,0xf0,0xf0,0x00,0x00,0x00,0x00,0x00,0x00,0xf8,0xf8,0x00,0x00,0x00,0x00,0x00,  
0xf8,0xf8,0x00,0x00,0x00,0xf0,0xf0,0x00,0x00,0x00,0x00,0xf8,0xf8,0x00,0x00,  
0x00,0x00,0xf0,0xf0,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,  
0x30,0x30,0x30,0x30,0x30,0x30,0x30,0xf0,0xf0,0x00,0x00,0x00,0xc0,0xc0,0xc0,0xc0,  
0xc0,0xc0,0xc0,0xfe,0xfe,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,  
0xc0,0xc0,0xc0,0xc0,0xc0,0xc0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x08,0x1c,0x18,0x38,0x70,0x20,  
0x00,0x00,0x90,0xb0,0x98,0x9c,0x8c,0x86,0x83,0x82,0xf0,0xf0,0x82,0x83,0x86,0x8c,  
0x9c,0xb8,0x90,0x80,0x00,0x06,0x06,0x06,0x06,0xff,0xff,0x06,0x06,0x06,0x00,0x00,  
0xff,0xff,0x00,0x00,0x00,0x00,0xff,0xff,0x00,0x00,0x00,0x00,0xff,0xff,0x00,0x00,  
0x00,0x00,0xff,0xff,0x00,0x00,0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,  
0xc3,0xc3,0xc3,0xc3,0xc3,0xc3,0x00,0x00,0xff,0xff,0x00,0x00,0x00,0x00,0x00,0x00,  
0x80,0x00,0x80,0x8f,0x0f,0x00,0x00,0x10,0x78,0xf0,0xc0,0x00,0x00,0x0f,0x0f,0x00,  
0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xf8,0x7e,0x0c,  
0x00,0x00,0x81,0xc1,0xc1,0x61,0x31,0x19,0x0d,0x07,0xff,0xff,0x07,0x1d,0x31,0x61,  
0xc1,0xc1,0x81,0x01,0x00,0x60,0x60,0x30,0x30,0x3f,0x1f,0x18,0x18,0x0c,0xcc,0xf0,  
0x7f,0x0f,0x00,0x00,0x00,0x00,0xff,0xff,0x00,0x00,0x00,0x00,0xff,0xff,0x00,0x00,  
0x00,0x00,0xff,0xff,0x00,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x30,0x3f,0x3f,0x30,  
0x33,0x36,0x3e,0x34,0x30,0x30,0x00,0xff,0xff,0x00,0x00,0x00,0x40,0xf0,0x7e,0x0f,  
0x01,0x00,0xff,0xff,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x07,0x02,0x00,0x00,0xf0,0xe0,  
0x01,0x07,0x3f,0xf8,0x40,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x0f,0x03,0x00,0x00,  
0x01,0x03,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x0f,0x0f,0x00,0x00,0x00,0x00,  
0x00,0x01,0x03,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x0e,0x07,0x03,0x01,  
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x01,0x00,0x00,0x00,0x0f,0x0f,0x00,0x00,  
0x00,0x00,0x0f,0x0f,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x06,  
0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x0f,0x0f,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x07,0x0f,0x0c,0x0c,0x0c,0x0c,0x0c,0x0c,0x0c,0x0c,0x0c,0x0f,0x07,0x00,  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,  
0x00,0x00,0x00,0x00,0x08,0x44,0x44,0xc4,0xb8,0x00,0x00,0x00,0x00,0x08,0x04,0x04,  
0xc4,0x78,0x00,0xc0,0x60,0xfc,0x06,0x08,0x68,0xc8,0x0a,0x0a,0x0c,0x88,0x68,0x08,  
0x00,0x38,0xc4,0x44,0xc4,0xa4,0x38,0x00,0x00,0xf0,0x08,0x04,0x04,0x0c,0xf0,0x00,  
0x00,0x7c,0x44,0x44,0x44,0xc4,0x84,0x00,0x00,0x00,0x08,0x0c,0xfc,0x00,0x00,0x00,  
0xfe,0x02,0x4a,0x4a,0x4a,0x4a,0xfa,0x4a,0xca,0x4a,0x4a,0x02,0xfe,0x00,0x08,0x08,  
0x88,0x08,0xbe,0x08,0x08,0xc8,0x88,0x08,0x3e,0x08,0x88,0x08,0x08,0x00,0x00,0xfc,  
0x24,0x24,0x24,0x24,0x26,0x26,0x24,0x24,0x24,0x24,0x24,0x3c,0x00,0x80,0xfc,0x84,0x96,



[illegible]

```

0xCF,0xEF,0xC7,0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xC3,0xE7,0xCF,
0xCF,0xDF,0xCF,0xCF,0xE7,0xE1,0xE0,0xF0,0xF8,0x78,0x7F,0x3F,0x1F,0x1F,0x00,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x1F,0x1F,0x1F,
0x1F,0x1F,0x87,0xF0,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xF8,0xC3,0x0F,0x1F,0x1F,0x1E,0x00,0x00,0xE0,0xE0,0xE0,
0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xE0,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0x00,0x00,0x00,0x00,0x00,0x00,0x60,0x78,0x7C,0x3E,0x9F,0xEF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0x9F,0x3E,0x7C,0x78,0x70,0x7C,0x3E,0x1F,0x0F,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,
0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x0B,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x01,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0x01,0x01,0x01,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x0F,0x3F,
0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x7F,0x1F,0x07,0x01,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,0x0F,0x0F,0xFF,0xFF,0xFF,0xFF,0x5F,0x0F,
0xFF,0xFF,0xFF,0x0F,0x0F,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,
0xFF,0x01,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0xFF,0xFF,0xFF,
0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,
0xE0,0xE0,0xE1,0xE3,0xE3,0xE7,0xC7,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0xCF,0xEF,
0xE7,0xE7,0xE7,0xE3,0xE1,0xC1,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x1F,0x1F,0x1F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x1F,0x1F,
0x1F,0x00,0x1F,0x1F,0x1F,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,
0x1F,0x1F,0x1F,0x1F,0x0F,0x00,0x0F,0x1F,0x1F,0x1F,0x1F,0x0F,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04,0x1F,0x1F,0x1F,0x1F,0x0F,0x00,0x07,0x1F,0x1F,
0x1F,0x1F,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,
0x1F,0x1F,0x1F,0x1F,0x1F,0x1F,0x0F,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x1F,
0x1F,0x1F,0x1F,0x1F,0x0F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,

```

```
};

#endif
```

## g) ascii.h -- 项目配置文件的头文件

```
#ifndef __ASCII_H__          //防止头文件被重复包含
#define __ASCII_H__
```

```
/******6*8 的点阵 ASCII 字符******/
```

```
unsigned char code ASC8[][6] =
```

```
{
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00,    // sp
    0x00, 0x00, 0x00, 0x2f, 0x00, 0x00,    // !
```

```

0x00, 0x00, 0x07, 0x00, 0x07, 0x00, // "
0x00, 0x14, 0x7f, 0x14, 0x7f, 0x14, // #
0x00, 0x24, 0x2a, 0x7f, 0x2a, 0x12, // $
0x00, 0x62, 0x64, 0x08, 0x13, 0x23, // %
0x00, 0x36, 0x49, 0x55, 0x22, 0x50, // &
0x00, 0x00, 0x05, 0x03, 0x00, 0x00, // '
0x00, 0x00, 0x1c, 0x22, 0x41, 0x00, // (
0x00, 0x00, 0x41, 0x22, 0x1c, 0x00, // )
0x00, 0x14, 0x08, 0x3E, 0x08, 0x14, // *
0x00, 0x08, 0x08, 0x3E, 0x08, 0x08, // +
0x00, 0x00, 0x00, 0xA0, 0x60, 0x00, // ,
0x00, 0x08, 0x08, 0x08, 0x08, 0x08, // -
0x00, 0x00, 0x60, 0x60, 0x00, 0x00, // .
0x00, 0x20, 0x10, 0x08, 0x04, 0x02, // /
0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E, // 0
0x00, 0x00, 0x42, 0x7F, 0x40, 0x00, // 1
0x00, 0x42, 0x61, 0x51, 0x49, 0x46, // 2
0x00, 0x21, 0x41, 0x45, 0x4B, 0x31, // 3
0x00, 0x18, 0x14, 0x12, 0x7F, 0x10, // 4
0x00, 0x27, 0x45, 0x45, 0x45, 0x39, // 5
0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30, // 6
0x00, 0x01, 0x71, 0x09, 0x05, 0x03, // 7
0x00, 0x36, 0x49, 0x49, 0x49, 0x36, // 8
0x00, 0x06, 0x49, 0x49, 0x29, 0x1E, // 9
0x00, 0x00, 0x36, 0x36, 0x00, 0x00, // :
0x00, 0x00, 0x56, 0x36, 0x00, 0x00, // ;
0x00, 0x08, 0x14, 0x22, 0x41, 0x00, // <
0x00, 0x14, 0x14, 0x14, 0x14, 0x14, // =
0x00, 0x00, 0x41, 0x22, 0x14, 0x08, // >
0x00, 0x02, 0x01, 0x51, 0x09, 0x06, // ?
0x00, 0x32, 0x49, 0x59, 0x51, 0x3E, // @
0x00, 0x7C, 0x12, 0x11, 0x12, 0x7C, // A
0x00, 0x7F, 0x49, 0x49, 0x49, 0x36, // B
0x00, 0x3E, 0x41, 0x41, 0x41, 0x22, // C
0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C, // D
0x00, 0x7F, 0x49, 0x49, 0x49, 0x41, // E
0x00, 0x7F, 0x09, 0x09, 0x09, 0x01, // F
0x00, 0x3E, 0x41, 0x49, 0x49, 0x7A, // G
0x00, 0x7F, 0x08, 0x08, 0x08, 0x7F, // H
0x00, 0x00, 0x41, 0x7F, 0x41, 0x00, // I
0x00, 0x20, 0x40, 0x41, 0x3F, 0x01, // J
0x00, 0x7F, 0x08, 0x14, 0x22, 0x41, // K
0x00, 0x7F, 0x40, 0x40, 0x40, 0x40, // L
0x00, 0x7F, 0x02, 0x0C, 0x02, 0x7F, // M
0x00, 0x7F, 0x04, 0x08, 0x10, 0x7F, // N

```

```

0x00, 0x3E, 0x41, 0x41, 0x41, 0x3E, // O
0x00, 0x7F, 0x09, 0x09, 0x09, 0x06, // P
0x00, 0x3E, 0x41, 0x51, 0x21, 0x5E, // Q
0x00, 0x7F, 0x09, 0x19, 0x29, 0x46, // R
0x00, 0x46, 0x49, 0x49, 0x49, 0x31, // S
0x00, 0x01, 0x01, 0x7F, 0x01, 0x01, // T
0x00, 0x3F, 0x40, 0x40, 0x40, 0x3F, // U
0x00, 0x1F, 0x20, 0x40, 0x20, 0x1F, // V
0x00, 0x3F, 0x40, 0x38, 0x40, 0x3F, // W
0x00, 0x63, 0x14, 0x08, 0x14, 0x63, // X
0x00, 0x07, 0x08, 0x70, 0x08, 0x07, // Y
0x00, 0x61, 0x51, 0x49, 0x45, 0x43, // Z
0x00, 0x00, 0x7F, 0x41, 0x41, 0x00, // [
0x00, 0x55, 0x2A, 0x55, 0x2A, 0x55, // \
0x00, 0x00, 0x41, 0x41, 0x7F, 0x00, // ]
0x00, 0x04, 0x02, 0x01, 0x02, 0x04, // ^
0x00, 0x40, 0x40, 0x40, 0x40, 0x40, // _
0x00, 0x00, 0x01, 0x02, 0x04, 0x00, // '
0x00, 0x20, 0x54, 0x54, 0x54, 0x78, // a
0x00, 0x7F, 0x48, 0x44, 0x44, 0x38, // b
0x00, 0x38, 0x44, 0x44, 0x44, 0x20, // c
0x00, 0x38, 0x44, 0x44, 0x48, 0x7F, // d
0x00, 0x38, 0x54, 0x54, 0x54, 0x18, // e
0x00, 0x08, 0x7E, 0x09, 0x01, 0x02, // f
0x00, 0x18, 0xA4, 0xA4, 0xA4, 0x7C, // g
0x00, 0x7F, 0x08, 0x04, 0x04, 0x78, // h
0x00, 0x00, 0x44, 0x7D, 0x40, 0x00, // i
0x00, 0x40, 0x80, 0x84, 0x7D, 0x00, // j
0x00, 0x7F, 0x10, 0x28, 0x44, 0x00, // k
0x00, 0x00, 0x41, 0x7F, 0x40, 0x00, // l
0x00, 0x7C, 0x04, 0x18, 0x04, 0x78, // m
0x00, 0x7C, 0x08, 0x04, 0x04, 0x78, // n
0x00, 0x38, 0x44, 0x44, 0x44, 0x38, // o
0x00, 0xFC, 0x24, 0x24, 0x24, 0x18, // p
0x00, 0x18, 0x24, 0x24, 0x18, 0xFC, // q
0x00, 0x7C, 0x08, 0x04, 0x04, 0x08, // r
0x00, 0x48, 0x54, 0x54, 0x54, 0x20, // s
0x00, 0x04, 0x3F, 0x44, 0x40, 0x20, // t
0x00, 0x3C, 0x40, 0x40, 0x20, 0x7C, // u
0x00, 0x1C, 0x20, 0x40, 0x20, 0x1C, // v
0x00, 0x3C, 0x40, 0x30, 0x40, 0x3C, // w
0x00, 0x44, 0x28, 0x10, 0x28, 0x44, // x
0x00, 0x1C, 0xA0, 0xA0, 0xA0, 0x7C, // y
0x00, 0x44, 0x64, 0x54, 0x4C, 0x44, // z
0x14, 0x14, 0x14, 0x14, 0x14, 0x14, // horiz lines

```

};

/\*\*\*\*\*8\*16 的点阵\*\*\*\*\*/

unsigned char code ASC16[[16]=

```

{
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // - -
    0x00,0x00,0x38,0xFC,0xFC,0x38,0x00,0x00,0x00,0x00,0x00,0x0D,0x0D,0x00,0x00,0x00, // -!-
    0x00,0x0E,0x1E,0x00,0x00,0x1E,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // -"-
    0x20,0xF8,0xF8,0x20,0xF8,0xF8,0x20,0x00,0x02,0x0F,0x0F,0x02,0x0F,0x0F,0x02,0x00, // -#-
    0x38,0x7C,0x44,0x47,0x47,0xCC,0x98,0x00,0x03,0x06,0x04,0x1C,0x1C,0x07,0x03,0x00, // -$$-
    0x30,0x30,0x00,0x80,0xC0,0x60,0x30,0x00,0x0C,0x06,0x03,0x01,0x00,0x0C,0x0C,0x00, // -%-
    0x80,0xD8,0x7C,0xE4,0xBC,0xD8,0x40,0x00,0x07,0x0F,0x08,0x08,0x07,0x0F,0x08,0x00, // -&-
    0x00,0x10,0x1E,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // -'-
    0x00,0x00,0xF0,0xF8,0x0C,0x04,0x00,0x00,0x00,0x00,0x03,0x07,0x0C,0x08,0x00,0x00, // -(-
    0x00,0x00,0x04,0x0C,0xF8,0xF0,0x00,0x00,0x00,0x00,0x08,0x0C,0x07,0x03,0x00,0x00, // -)-
    0x80,0xA0,0xE0,0xC0,0xC0,0xE0,0xA0,0x80,0x00,0x02,0x03,0x01,0x01,0x03,0x02,0x00, // -*-
    0x00,0x80,0x80,0xE0,0xE0,0x80,0x80,0x00,0x00,0x00,0x00,0x03,0x03,0x00,0x00,0x00, // -+ -
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x10,0x1E,0x0E,0x00,0x00,0x00, // -,-
    0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // ---
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0C,0x0C,0x00,0x00,0x00, // -.-
    0x00,0x00,0x00,0x80,0xC0,0x60,0x30,0x00,0x0C,0x06,0x03,0x01,0x00,0x00,0x00,0x00, // -/-
    0xF8,0xFC,0x04,0xC4,0x24,0xFC,0xF8,0x00,0x07,0x0F,0x09,0x08,0x08,0x0F,0x07,0x00, // -0-
    0x00,0x10,0x18,0xFC,0xFC,0x00,0x00,0x00,0x00,0x08,0x08,0x0F,0x0F,0x08,0x08,0x00, // -1-
    0x08,0x0C,0x84,0xC4,0x64,0x3C,0x18,0x00,0x0E,0x0F,0x09,0x08,0x08,0x0C,0x0C,0x00, // -2-
    0x08,0x0C,0x44,0x44,0x44,0xFC,0xB8,0x00,0x04,0x0C,0x08,0x08,0x08,0x0F,0x07,0x00, // -3-
    0xC0,0xE0,0xB0,0x98,0xFC,0xFC,0x80,0x00,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00, // -4-
    0x7C,0x7C,0x44,0x44,0xC4,0xC4,0x84,0x00,0x04,0x0C,0x08,0x08,0x08,0x0F,0x07,0x00, // -5-
    0xF0,0xF8,0x4C,0x44,0x44,0xC0,0x80,0x00,0x07,0x0F,0x08,0x08,0x08,0x0F,0x07,0x00, // -6-
    0x0C,0x0C,0x04,0x84,0xC4,0x7C,0x3C,0x00,0x00,0x00,0x0F,0x0F,0x00,0x00,0x00,0x00, // -7-
    0xB8,0xFC,0x44,0x44,0x44,0xFC,0xB8,0x00,0x07,0x0F,0x08,0x08,0x08,0x0F,0x07,0x00, // -8-
    0x38,0x7C,0x44,0x44,0x44,0xFC,0xF8,0x00,0x00,0x08,0x08,0x08,0x0C,0x07,0x03,0x00, // -9-
    0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00,0x00,0x00,0x00,0x06,0x06,0x00,0x00,0x00, // -:-
    0x00,0x00,0x00,0x30,0x30,0x00,0x00,0x00,0x00,0x00,0x08,0x0E,0x06,0x00,0x00,0x00, // -;-
    0x00,0x80,0xC0,0x60,0x30,0x18,0x08,0x00,0x00,0x00,0x01,0x03,0x06,0x0C,0x08,0x00, // -<-
    0x40,0x40,0x40,0x40,0x40,0x40,0x40,0x00,0x02,0x02,0x02,0x02,0x02,0x02,0x02,0x00, // -=-
    0x00,0x08,0x18,0x30,0x60,0xC0,0x80,0x00,0x00,0x08,0x0C,0x06,0x03,0x01,0x00,0x00, // ->-
    0x18,0x1C,0x04,0xC4,0xE4,0x3C,0x18,0x00,0x00,0x00,0x00,0x0D,0x0D,0x00,0x00,0x00, // -?-
    0xF0,0xF8,0x08,0xC8,0xC8,0xF8,0xF0,0x00,0x07,0x0F,0x08,0x0B,0x0B,0x0B,0x01,0x00, // -@-
    0xE0,0xF0,0x98,0x8C,0x98,0xF0,0xE0,0x00,0x0F,0x0F,0x00,0x00,0x00,0x0F,0x0F,0x00, // -A-
    0x04,0xFC,0xFC,0x44,0x44,0xFC,0xB8,0x00,0x08,0x0F,0x0F,0x08,0x08,0x0F,0x07,0x00, // -B-
    0xF0,0xF8,0x0C,0x04,0x04,0x0C,0x18,0x00,0x03,0x07,0x0C,0x08,0x08,0x0C,0x06,0x00, // -C-
    0x04,0xFC,0xFC,0x04,0x0C,0xF8,0xF0,0x00,0x08,0x0F,0x0F,0x08,0x0C,0x07,0x03,0x00, // -D-
    0x04,0xFC,0xFC,0x44,0xE4,0x0C,0x1C,0x00,0x08,0x0F,0x0F,0x08,0x08,0x0C,0x0E,0x00, // -E-
    0x04,0xFC,0xFC,0x44,0xE4,0x0C,0x1C,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00,0x00,0x00, // -F-
    0xF0,0xF8,0x0C,0x84,0x84,0x8C,0x98,0x00,0x03,0x07,0x0C,0x08,0x08,0x07,0x0F,0x00, // -G-

```

0xFC,0xFC,0x40,0x40,0x40,0xFC,0xFC,0x00,0x0F,0x0F,0x00,0x00,0x00,0x0F,0x0F,0x00, // -H-  
0x00,0x00,0x04,0xFC,0xFC,0x04,0x00,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00, // -I-  
0x00,0x00,0x00,0x04,0xFC,0xFC,0x04,0x00,0x07,0x0F,0x08,0x08,0x0F,0x07,0x00,0x00, // -J-  
0x04,0xFC,0xFC,0xC0,0xF0,0x3C,0x0C,0x00,0x08,0x0F,0x0F,0x00,0x01,0x0F,0x0E,0x00, // -K-  
0x04,0xFC,0xFC,0x04,0x00,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x08,0x0C,0x0E,0x00, // -L-  
0xFC,0xFC,0x38,0x70,0x38,0xFC,0xFC,0x00,0x0F,0x0F,0x00,0x00,0x00,0x0F,0x0F,0x00, // -M-  
0xFC,0xFC,0x38,0x70,0xE0,0xFC,0xFC,0x00,0x0F,0x0F,0x00,0x00,0x00,0x0F,0x0F,0x00, // -N-  
0xF0,0xF8,0x0C,0x04,0x0C,0xF8,0xF0,0x00,0x03,0x07,0x0C,0x08,0x0C,0x07,0x03,0x00, // -O-  
0x04,0xFC,0xFC,0x44,0x44,0x7C,0x38,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00,0x00,0x00, // -P-  
0xF8,0xFC,0x04,0x04,0x04,0xFC,0xF8,0x00,0x07,0x0F,0x08,0x0E,0x3C,0x3F,0x27,0x00, // -Q-  
0x04,0xFC,0xFC,0x44,0xC4,0xFC,0x38,0x00,0x08,0x0F,0x0F,0x00,0x00,0x0F,0x0F,0x00, // -R-  
0x18,0x3C,0x64,0x44,0xC4,0x9C,0x18,0x00,0x06,0x0E,0x08,0x08,0x08,0x0F,0x07,0x00, // -S-  
0x00,0x1C,0x0C,0xFC,0xFC,0x0C,0x1C,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00, // -T-  
0xFC,0xFC,0x00,0x00,0x00,0xFC,0xFC,0x00,0x07,0x0F,0x08,0x08,0x08,0x0F,0x07,0x00, // -U-  
0xFC,0xFC,0x00,0x00,0x00,0xFC,0xFC,0x00,0x01,0x03,0x06,0x0C,0x06,0x03,0x01,0x00, // -V-  
0xFC,0xFC,0x00,0x80,0x00,0xFC,0xFC,0x00,0x03,0x0F,0x0E,0x03,0x0E,0x0F,0x03,0x00, // -W-  
0x0C,0x3C,0xF0,0xC0,0xF0,0x3C,0x0C,0x00,0x0C,0x0F,0x03,0x00,0x03,0x0F,0x0C,0x00, // -X-  
0x00,0x3C,0x7C,0xC0,0xC0,0x7C,0x3C,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00, // -Y-  
0x1C,0x0C,0x84,0xC4,0x64,0x3C,0x1C,0x00,0x0E,0x0F,0x09,0x08,0x08,0x0C,0x0E,0x00, // -Z-  
0x00,0x00,0xFC,0xFC,0x04,0x04,0x00,0x00,0x00,0x00,0x0F,0x0F,0x08,0x08,0x00,0x00, // -[-  
0x38,0x70,0xE0,0xC0,0x80,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x07,0x0E,0x00, // -\-  
0x00,0x00,0x04,0x04,0xFC,0xFC,0x00,0x00,0x00,0x00,0x08,0x08,0x0F,0x0F,0x00,0x00, // -]-  
0x08,0x0C,0x06,0x03,0x06,0x0C,0x08,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // -^-  
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x20,0x20,0x20,0x20,0x20, // -\_-  
0x00,0x00,0x03,0x07,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // -`-  
0x00,0xA0,0xA0,0xA0,0xE0,0xC0,0x00,0x00,0x07,0x0F,0x08,0x08,0x07,0x0F,0x08,0x00, // -a-  
0x04,0xFC,0xFC,0x20,0x60,0xC0,0x80,0x00,0x08,0x0F,0x07,0x08,0x08,0x0F,0x07,0x00, // -b-  
0xC0,0xE0,0x20,0x20,0x20,0x60,0x40,0x00,0x07,0x0F,0x08,0x08,0x08,0x0C,0x04,0x00, // -c-  
0x80,0xC0,0x60,0x24,0xFC,0xFC,0x00,0x00,0x07,0x0F,0x08,0x08,0x07,0x0F,0x08,0x00, // -d-  
0xC0,0xE0,0xA0,0xA0,0xA0,0xE0,0xC0,0x00,0x07,0x0F,0x08,0x08,0x08,0x0C,0x04,0x00, // -e-  
0x40,0xF8,0xFC,0x44,0x0C,0x18,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00,0x00,0x00, // -f-  
0xC0,0xE0,0x20,0x20,0xC0,0xE0,0x20,0x00,0x27,0x6F,0x48,0x48,0x7F,0x3F,0x00,0x00, // -g-  
0x04,0xFC,0xFC,0x40,0x20,0xE0,0xC0,0x00,0x08,0x0F,0x0F,0x00,0x00,0x0F,0x0F,0x00, // -h-  
0x00,0x00,0x20,0xEC,0xEC,0x00,0x00,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00, // -i-  
0x00,0x00,0x00,0x00,0x20,0xEC,0xEC,0x00,0x00,0x30,0x70,0x40,0x40,0x7F,0x3F,0x00, // -j-  
0x04,0xFC,0xFC,0x80,0xC0,0x60,0x20,0x00,0x08,0x0F,0x0F,0x01,0x03,0x0E,0x0C,0x00, // -k-  
0x00,0x00,0x04,0xFC,0xFC,0x00,0x00,0x00,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00, // -l-  
0xE0,0xE0,0x60,0xC0,0x60,0xE0,0xC0,0x00,0x0F,0x0F,0x00,0x0F,0x00,0x0F,0x0F,0x00, // -m-  
0x20,0xE0,0xC0,0x20,0x20,0xE0,0xC0,0x00,0x00,0x0F,0x0F,0x00,0x00,0x0F,0x0F,0x00, // -n-  
0xC0,0xE0,0x20,0x20,0x20,0xE0,0xC0,0x00,0x07,0x0F,0x08,0x08,0x08,0x0F,0x07,0x00, // -o-  
0x20,0xE0,0xC0,0x20,0x20,0xE0,0xC0,0x00,0x40,0x7F,0x7F,0x48,0x08,0x0F,0x07,0x00, // -p-  
0xC0,0xE0,0x20,0x20,0xC0,0xE0,0x20,0x00,0x07,0x0F,0x08,0x48,0x7F,0x7F,0x40,0x00, // -q-  
0x20,0xE0,0xC0,0x60,0x20,0x60,0xC0,0x00,0x08,0x0F,0x0F,0x08,0x00,0x00,0x00,0x00, // -r-  
0x40,0xE0,0xA0,0x20,0x20,0x60,0x40,0x00,0x04,0x0C,0x09,0x09,0x0B,0x0E,0x04,0x00, // -s-  
0x20,0x20,0xF8,0xFC,0x20,0x20,0x00,0x00,0x00,0x00,0x07,0x0F,0x08,0x0C,0x04,0x00, // -t-

```

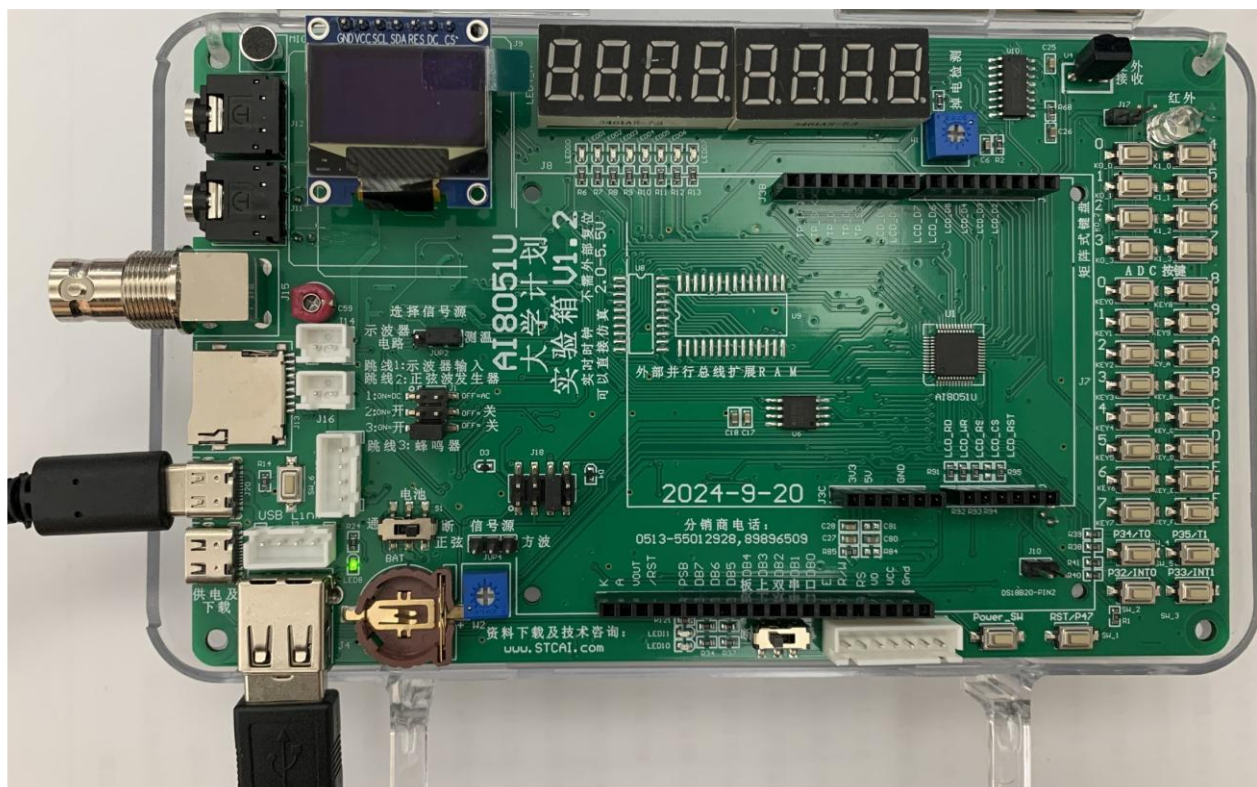
0xE0,0xE0,0x00,0x00,0xE0,0xE0,0x00,0x00,0x07,0x0F,0x08,0x08,0x07,0x0F,0x08,0x00, // -u-
0x00,0xE0,0xE0,0x00,0x00,0xE0,0xE0,0x00,0x00,0x03,0x07,0x0C,0x0C,0x07,0x03,0x00, // -v-
0xE0,0xE0,0x00,0x00,0x00,0xE0,0xE0,0x00,0x07,0x0F,0x0C,0x07,0x0C,0x0F,0x07,0x00, // -w-
0x20,0x60,0xC0,0x80,0xC0,0x60,0x20,0x00,0x08,0x0C,0x07,0x03,0x07,0x0C,0x08,0x00, // -x-
0xE0,0xE0,0x00,0x00,0x00,0xE0,0xE0,0x00,0x47,0x4F,0x48,0x48,0x68,0x3F,0x1F,0x00, // -y-
0x60,0x60,0x20,0xA0,0xE0,0x60,0x20,0x00,0x0C,0x0E,0x0B,0x09,0x08,0x0C,0x0C,0x00, // -z-
0x00,0x40,0x40,0xF8,0xBC,0x04,0x04,0x00,0x00,0x00,0x00,0x07,0x0F,0x08,0x08,0x00, // -{-
0x00,0x00,0x00,0xBC,0xBC,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0x0F,0x00,0x00,0x00, // -|-
0x00,0x04,0x04,0xBC,0xF8,0x40,0x40,0x00,0x00,0x08,0x08,0x0F,0x07,0x00,0x00,0x00, // -}-
0x08,0x0C,0x04,0x0C,0x08,0x0C,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, // ~-
0x80,0xC0,0x60,0x30,0x60,0xC0,0x80,0x00,0x07,0x07,0x04,0x04,0x04,0x07,0x07,0x00, // -□-
};

```

```
#endif
```

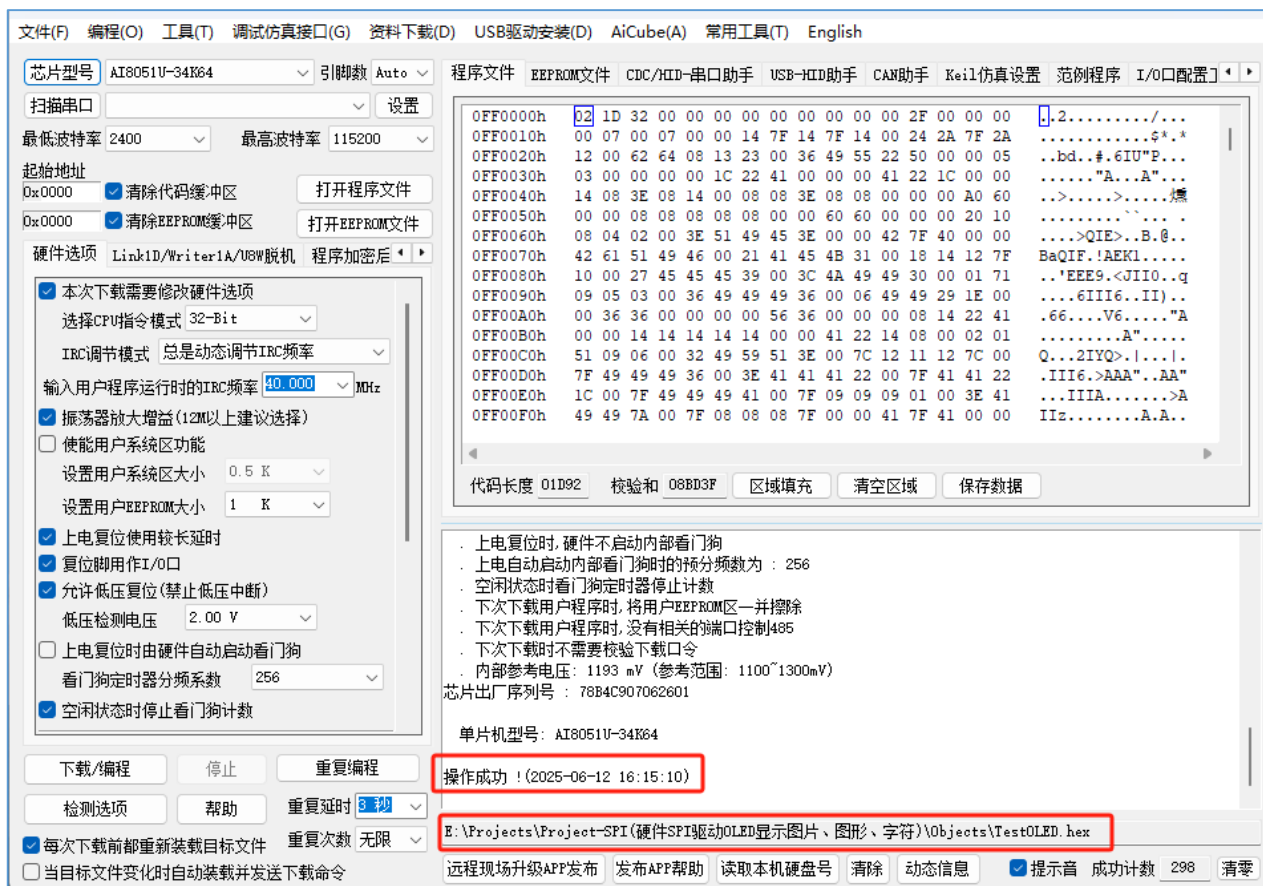
## 25.10.4. 程序下载，观察实验现象，验证实验效果

录入代码，保存，编译。将 OLED 屏正确的插到 Ai8051U 实验箱的 J9 接口，此时 OLED 屏上无任何图案显示，如下图：

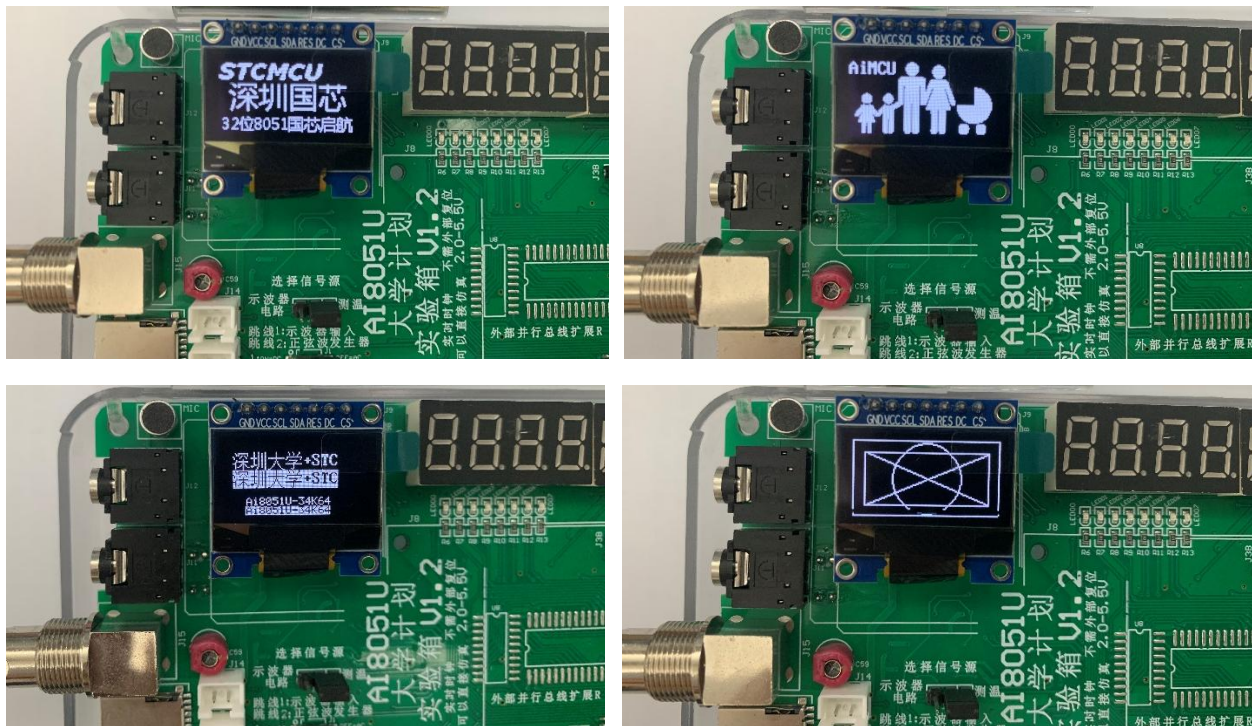


运行 AIapp-ISP 软件系统，按步骤打开并下载程序“TestOLED.hex”，如下图：





现在观察 Ai8051U 实验箱上的 OLED 屏幕, 下列 4 幅图片在 OLED 屏上循环显示:



完成实验验证。