STM32 MCU Development

STM32 单片机开发

-- I2C 协议之 OLED 显示屏显示

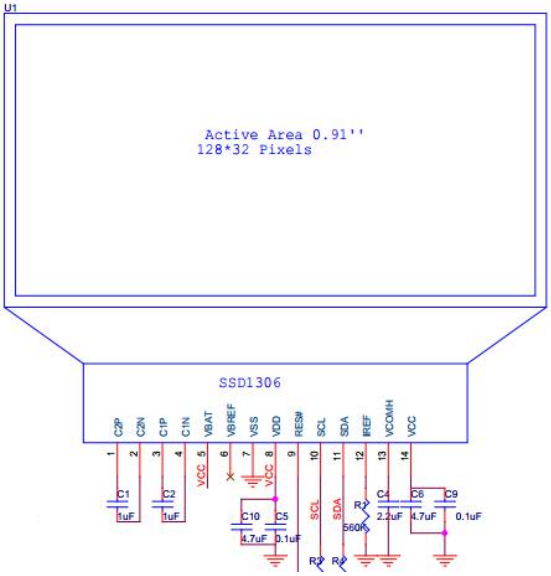
目录

1. OLED 显示屏介绍

1.1. OLED 介绍

小熊座 NB-IoT 开发板上配备了一块 I2C 接口的中景园电子 0.91 寸 128x32 像素 OLED 显示屏，通过该显示屏我们可以实时显示采样的温湿度、噪音值等。OLED 即有机发光二极管（Organic Light-Emitting Diode），又称为有机电激光显示（Organic Electroluminesence Display，OELD）。OLED 由于同时具备自发光，不需背光源、对比度高、厚度薄、视角广、反应速度快、可用于挠曲性面板、使用温度范围广、构造及制程较简单等优异之特性。

该 OLED 显示屏采用 SSD1309 控制器，该芯片具有 256 级亮度控制，专为共阴极 OLED 面板设计，它是一个单片 CMOS OLED/PLED 驱动芯片，可以驱动有机/聚合发光二极管点阵图形显示系统。SSD1309 中嵌入了对比度控制器、显示 RAM 和晶振，并因此减少了外部器件和功耗。数据/命令的发送有三种接口可选择：6800/8000 串口，I2C 接口或 SPI 接口。



1.2. 显示原理

1.2.1. 像素

在计算机中，每一个字符或图片都是由无数个像素点组成，而每一个像素点都表现一种颜色，这样我们所看到的图片都是由这无数个像素点堆叠形成。

如左下图所示是一朵油菜花，然后如右图所示把它放的特别大，则会看到它是有一个一个像素组成。

我们视觉上看到的正常图片其实是由无数个这样的小色块组成，我们管这个小色块叫像素(pixel)。每个像素都有其在图片中对应的坐标，我们电脑所储存的信息就是这些坐标和坐标所包含的 RGB 系数，然后再通过数字压缩算法对我们的图片数据进行压缩。所谓的 PNG 等格式就是对我们图片数据压缩的多种格式。

前面我们提到的小熊座 NB-IoT 开发板的 OLED 为 0.91 寸 128x32 像素显示屏，这些参数的说明如下图所示，OLED 显示屏的对角线物理尺寸大小 0.91 寸，其中横向(x 轴)有 128 个像素，纵向(y 轴)有 32 个像素。故共有 4096 个像素。一个显示屏上横向和纵向各有多少个像素叫做分辨率，所以该屏的分辨率为 128*32 像素。

1.2.2. 像素深度

在光学中我们普遍认为大自然色可以通过通过纯的红(Red)、绿(Green)、蓝(Blue)三种颜色可以组合成任一种颜色。比如红色与绿色组成黄色，蓝色与红色组成紫色，而三种颜色放在一起变成白色。改变每种颜色的亮度也会影响所形成的新颜色，这就是计算机可以产生任何颜色的原理！

计算机中用二进制位来表示一个像素的数据，用来表示一个像素的数据位越多，则这个像素的颜色值更加丰富、分的更细，颜色深度就更深。一个像素的颜色在计算机中用多少位来描述，这个单位叫像素深度，简称位深（bits per pix, bpp）。

常见的像素深度：1 位、8 位、16 位、24 位、32 位

1 位：用 1 个二进制位来表示颜色，这种就叫单色显示，如只有黑色或白色两种颜色。示例就是小饭店、理发店门口的 LED 屏

8 位：用 8 个二进制位来表示颜色，此时能表示 256 种颜色，这种叫灰度显示。注意这时候没有彩色是黑白的，我们把纯白到纯黑分别对应 255 到 0，中间的数值对应不同的灰，示例就是以前的黑白电视机。

16 位：用 16 个二进制位表示颜色，此时能表示 65536 种颜色，这时候就可以彩色显示了。一般是 RGB565 的颜色分布（用 5 位表示红色、用 6 位表示绿色、用 5 位表示蓝色）。这种红绿蓝都有的颜色表示法就是一种模拟自然界中所有颜色的表示方式，但是因为 RGB 的颜色表达本身二进制位数不够多（导致红绿蓝三种颜色本身分的都不够细致），所以这样显示的彩色失真比较重，人眼能明显看到显示的不真实，所以 RGB565 叫做假彩色。

24 位：用 24 个二进制位来表示颜色，此时能表示 16777216 种颜色。这种表示方式和 16 位色原理是一样的，只是 RGB 三种颜色各自的精度都更高了（RGB 各 8 位），叫 RGB888，也叫 RGB24。此时颜色比 RGB565 更加真实细腻，虽然说比自然界无数种颜色还是少了很多，不过由于人眼的不理想性所以人眼几乎不能区分 1677 万种颜色和无数种颜色的差别了，于是乎就把这种 RGB888 的表示方法叫做真彩色。

32 位：总共用 32 位二进制来表示颜色，其中 24 位表示红绿蓝三元色（还是 RGB888 分布），剩下 8 位表示透明度。这种显色方式就叫 ARGB（A 是阿尔法，表示透明度），现在 PC 机中一般都用 ARGB 表示颜色。

1.2.3. 扩展知识

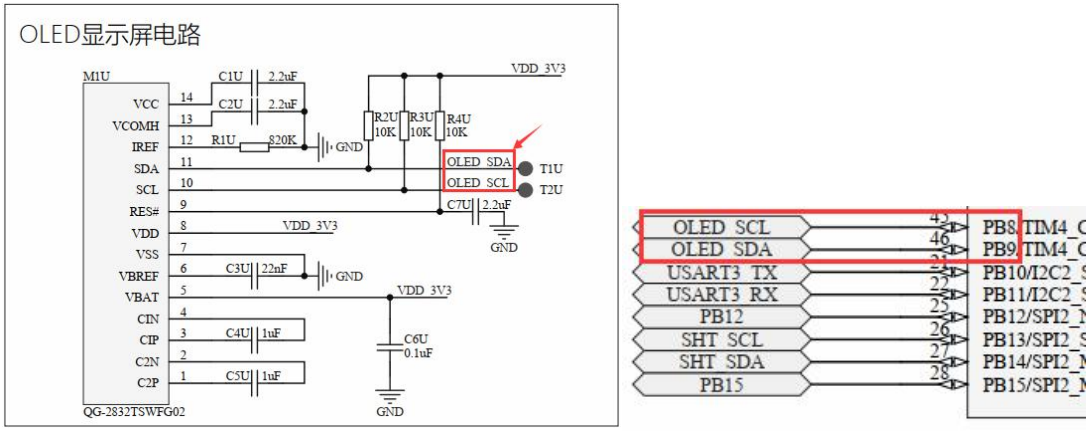
假设某 LCD 显示屏其分辨率为 320x240，颜色位深为 32bpp，请问一屏数据需要占用多大内存空间？

答： 320x240(像素) * 32(bpp) / 8 (一个字节为 8 个位) = 307200 字节

2. OLED 显示

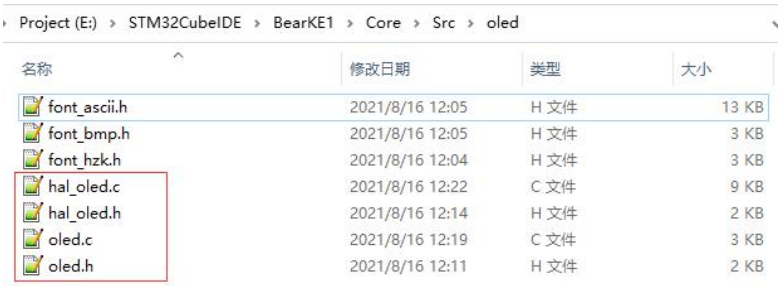
2.1. 原理图分析

小熊座 NB-IoT 开发板上的 OLED 显示屏通过 I2C 接口连接到 STM32 单片机的 PB8 和 PB9 管脚上，为了保证程序的可移植性，我们在 OLED 驱动代码里采用 GPIO 模拟 I2C 接口，而不是直接使用 STM32 单片机上的 I2C 总线控制器。这样今后该 OLED 显示屏不管连到 STM32 单片机的哪两个管脚上都能正常工作。



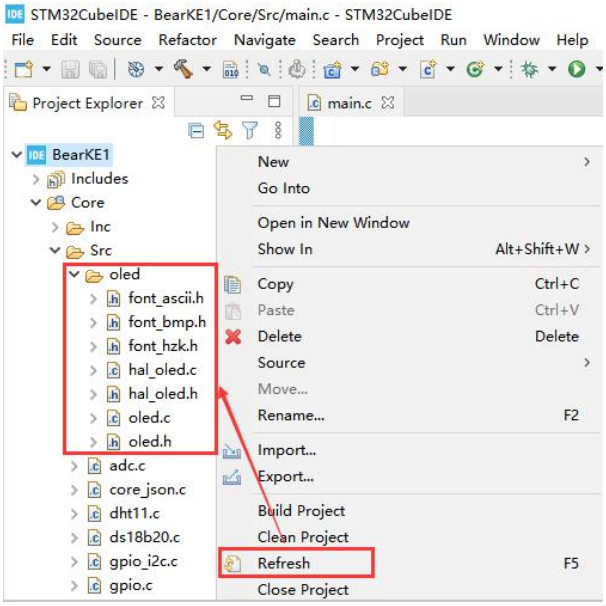
2.2. 添加 OLED 驱动文件

将 OLED 显示屏驱动文件夹 oled 拷贝到 Core/Src 路径下，为了便于移植和管理，我们将.c/.h 文件都放到同一个文件夹下。



- hal_oled.c/h 这里是 OLED 底层硬件相关的驱动文件，在该文件里使用 GPIO 模拟 I2C 提供了 OLED 操作的接口，并实现了 oled 初始化、字符串显示、汉字显示、BMP 图片显示等功能；
- oled.c/h 这个源文件封装了 OLED 显示屏上电时显示的 logo 信息，以及显示温度、湿度、噪音、光强等采样信息函数；
- font_ascii.h 文件提供了 ASCII 可打印字符的字库, oled.c 中 OLED_ShowBanner()函数会调用；
- font_bmp.h 文件提供了凌云实验室 BMP 格式 logo 的字库, oled.c 中 OLED_ShowBanner()函数会调用；
- Font_hzk.h 文件提供了汉字“凌云实验室”的字库，, oled.c 中 OLED_ShowBanner()函数会调用；

添加代码之后，鼠标右键点击项目并刷新，项目中将会出现 oled 的相关代码。



2.3. 测试代码修改与编译

修改 main.c 文件，在进入 while()循环之前添加 oled 显示函数调用：

```
... ...
在文件开始处添加头文件 oled.h ：
/* USER CODE BEGIN Includes */
#include <string.h>
#include "dht11.h"
#include "ds18b20.h"
#include "core_json.h"
#include "oled.h"
/* USER CODE END Includes */

... ...

int main(void)
{
  ... ...

  /* USER CODE BEGIN WHILE */
  sysled_hearbeat();
  beep_start(2, 300);
  printf("Start BearKE1 5G NB-IoT Board Example Program v1.0\r\n");
```

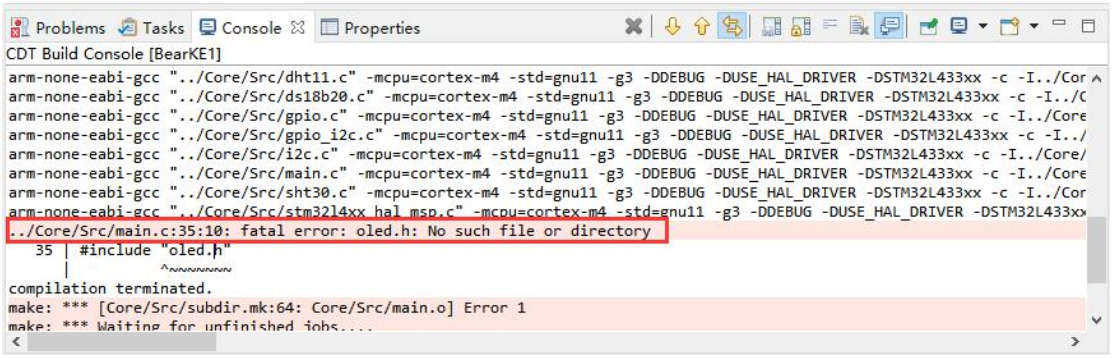
```
  OLED_Init();
  OLED_ShowBanner(TIME_1S*2);
  while (1)
  {
      ... ...
  }
  ... ...
}

... ...
```

修改 report_tempRH_json()函数，在采样温湿度值之后让 OLED 显示屏输出 ：

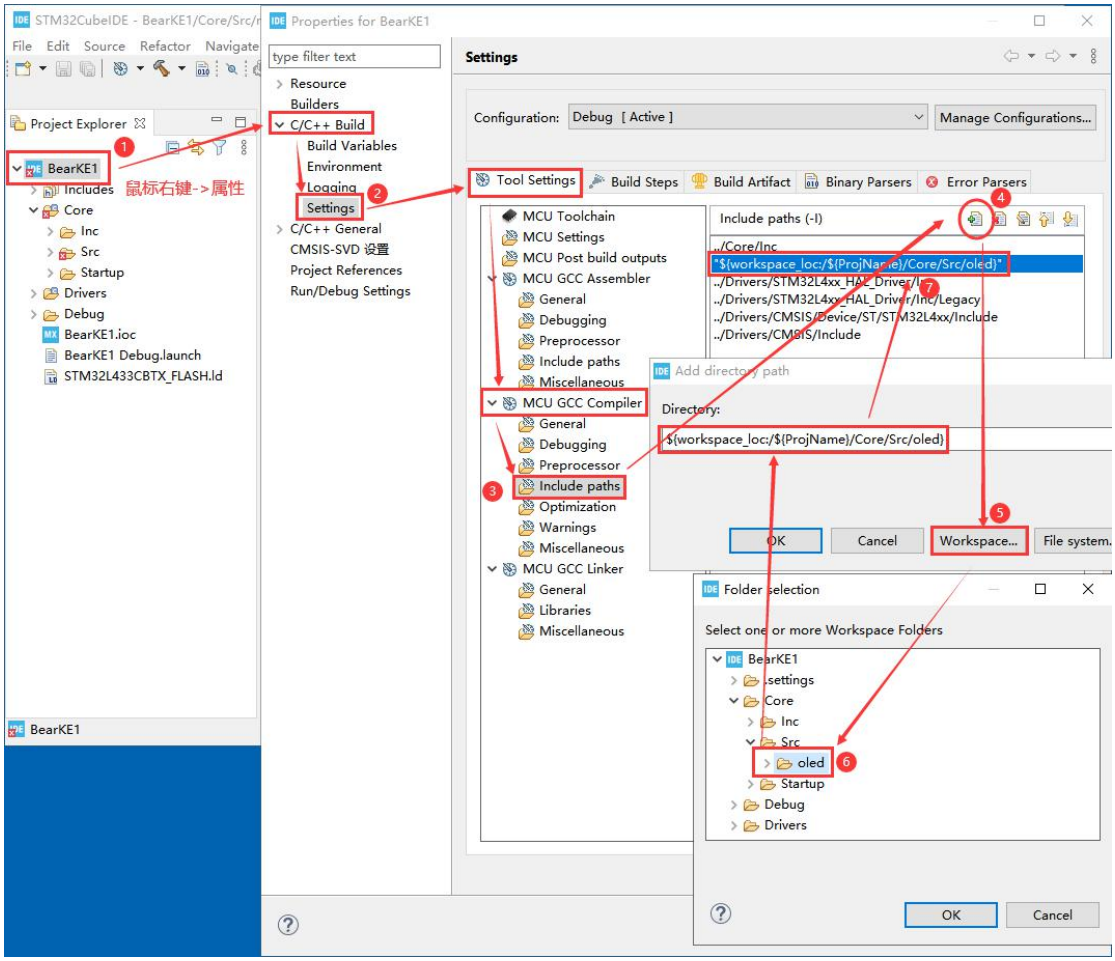
```
int report_tempRH_json(void)
{
    char         buf[128];
    float        temperature, humidity=0.0;
    uint32_t     temp, humd;

   if ( DHT11_SampleData(&temperature, &humidity) < 0 )
    {
         return -1;
    }

    memset(buf, 0, sizeof(buf));
    snprintf(buf, sizeof(buf), "{\"Temperature\":\"%.2f\", \"Humidity\":\"%.2f\"}", temperature,
humidity);

    temp = (int)(temperature*100);
    humd = (int)(humidity*100);
    OLED_ShowTempHumdity(temp, humd, TIME_1S*2);

    HAL_UART_Transmit(&huart1 , (uint8_t *)buf, strlen(buf), 0xFFFF);

    return 0;
}
```

2.4. 程序编译

代码修改好之后，点击编译将会出现头文件找不到的错误。这是因为 STM32CubeIDE 默认会到 Core/Inc 路径下查找相应的头文件，而现在 OLED 驱动的相关头文件为了便于移植，统一放到了 Core/oled 文件夹下。



这时我们可以鼠标右键点在项目上，选择最下面的属性。在编译器设置里添加其他头文件所在的路径。



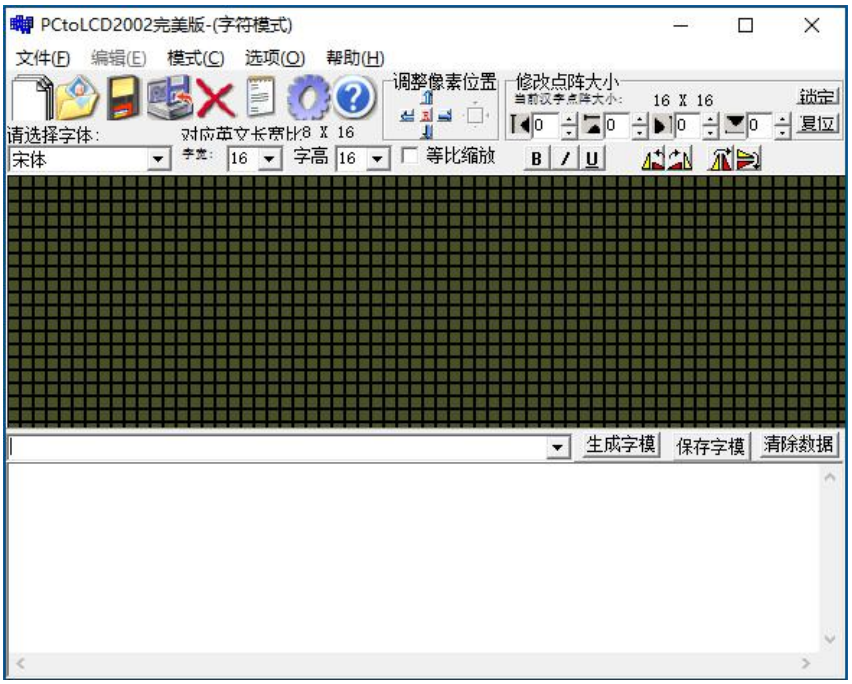
2.5. 运行测试

程序编译后重新烧录运行，就可以看到 OLED 输出实验室的 logo 和中英文显示信息，同时每隔 3s 会实时显示当前的温度值。

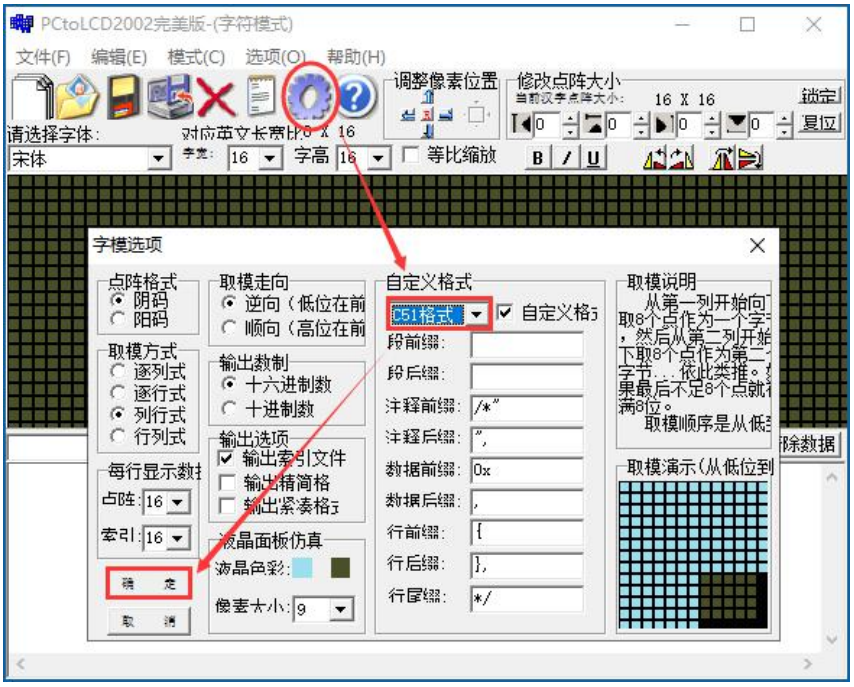
3. 创建 OLED 显示字库

3.1. 字库生成软件

字模提取软件(PCtoLCD2002) 可以根据输入的字符或汉字，以及 BMP 图片文件自动生成相应的 LCD 显示字模，支持汇编和 C 语言代码输出。

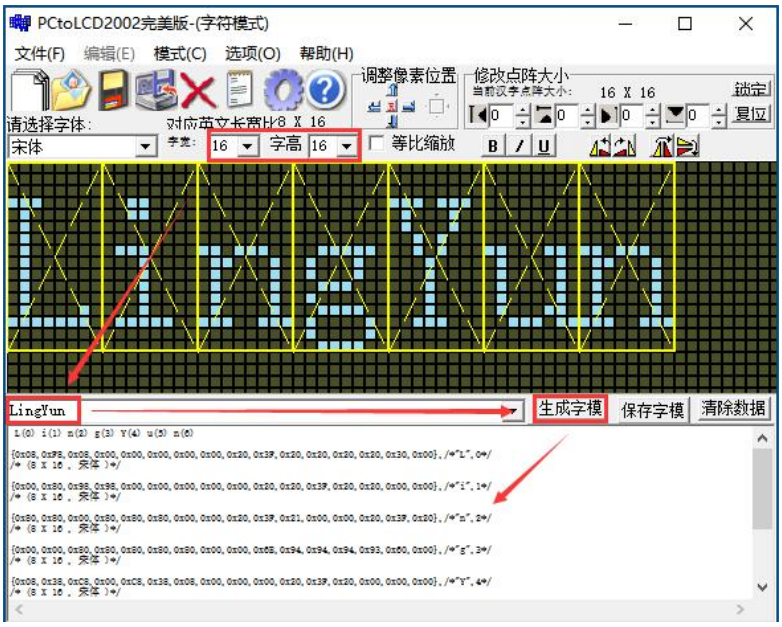


在开始使用它生成我们所需要的字库代码之前，我们首先需要如下图所示更改格式为 C51 格式。



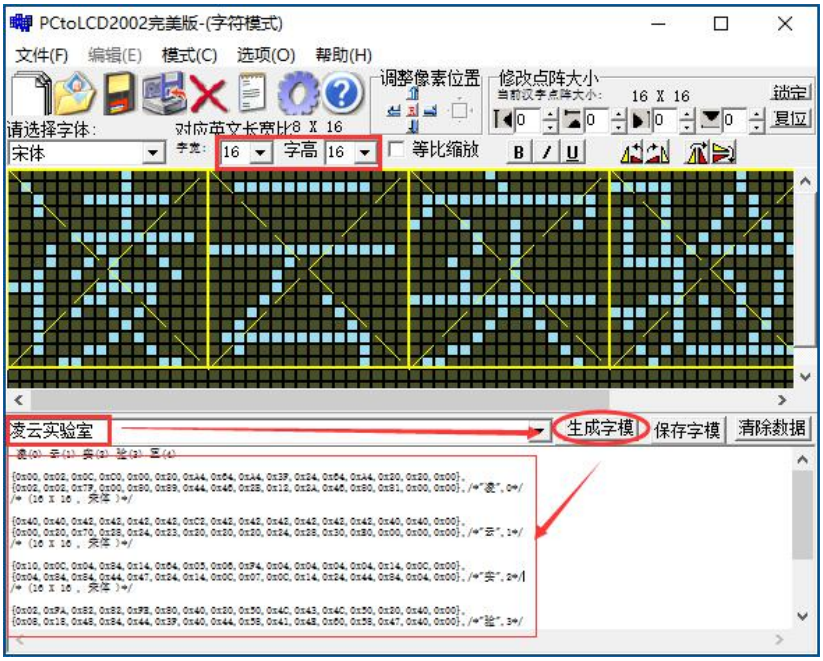
3.2. 生成英文字模

在输入框里输入可打印的 ASCII 字符，点击“生成字模”即可生成要显示的英文字模及相应符号的字模，font_ascii.h 中的字模就是使用这种方式生成。接下来可以调用 hal_oled.c/h 中提供的 OLED_ShowString()函数来显示英文字符串。



3.3. 生成汉字字模

在输入框里输入要打印的汉字，点击“生成字模”即可生成要显示的汉字字模，把它拷贝到相应的头文件(font_hzk.h)中即可使用。接下来可以调用 hal_oled.c/h 中提供的 OLED_ShowChinese()函数来显示。



3.4. 生成 BMP 图片字模

BMP(Bitmap)是 Windows 操作系统中的标准图像文件格式，能够被多种 Windows 应用程序所支持。PCtoLCD2002 软件支持 BMP 格式图片生成字模，这样我们就可以在 OLED 显示屏上显示相应的 Logo。

因为我们的 OLED 显示屏是 128x32 像素，所以这里我们首先需要使用 PhotoShop 或 GIMP 软件生成 128x32 像素的 BMP 格式图片，其位深度调整为 1 位 bpp。

接下来点击 PCtoLCD2002 软件上的“文件”菜单按钮，点击“打开”选中准备好的 BMP 图片文件，如果想修改也可以直接点击上方的小方格来选中或取消该像素显示。然后点击“生成字模”即可生成要显示的图片字模。接下来可以调用 hal_oled.c/h 中提供的 OLED_DrawBMP()函数来显示 BMP 格式图片。