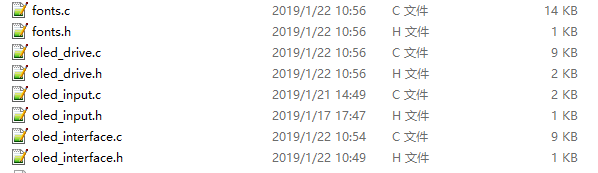
**EAZY-OLED1.0 UI界面库的使用说明**

**1.初阶使用**

1.1 文件结构



1.2 文件作用

fonts.c：保存了字模结构体、英文字模以及自定义图标的字模

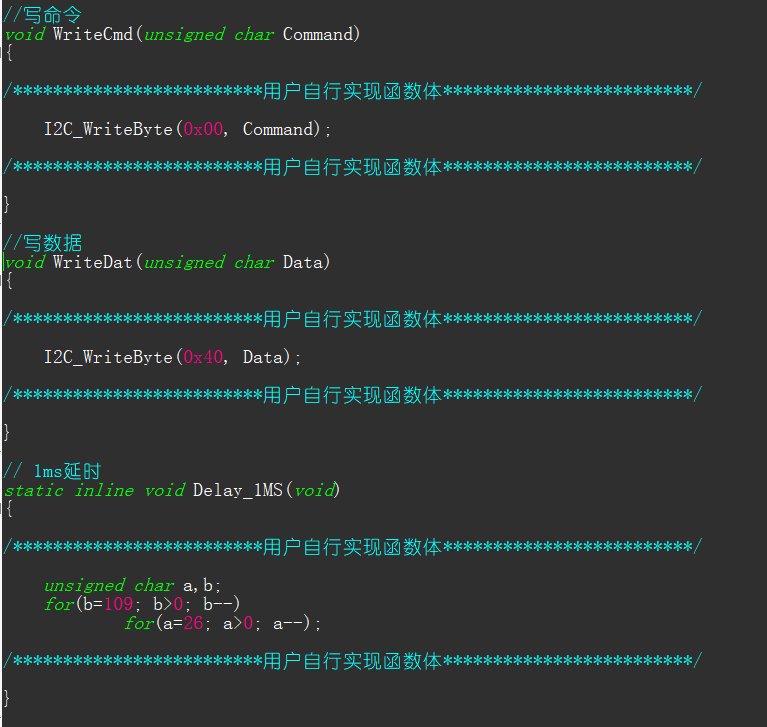
oled\_drive.c ： 底层OLED驱动文件，包含有许多OLED基本的绘图函数

oled\_input.c ： OLED四个按键（up/down/yes/no）在这个文件内实现

oled\_interface.c ：OLED多级菜单的自定义以及显示的功能

1.3 移植

1.3.1 oled\_drive.c



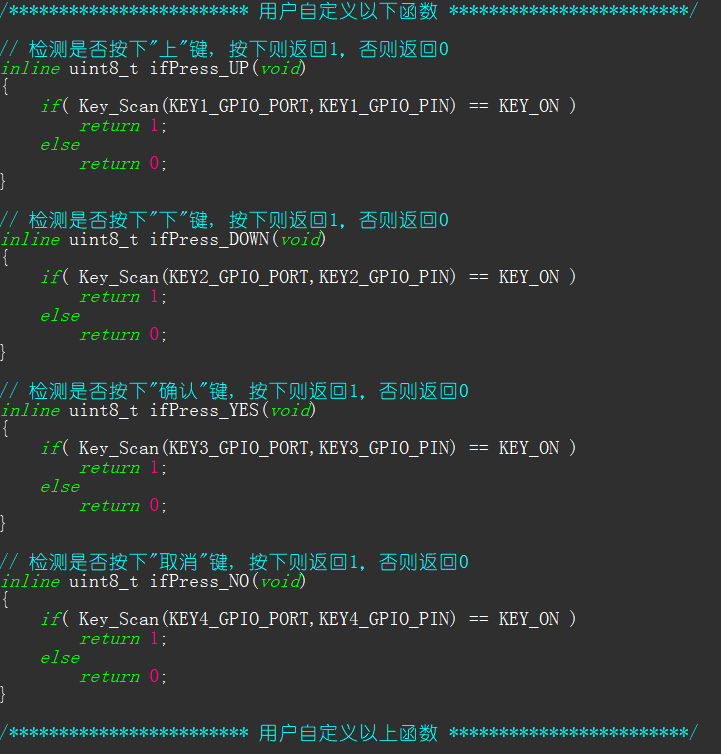
用户需要自行实现以上三个函数，分别是

WrireCmd() ：OLED的写命令函数

WriteDat()：OLED的写数据函数

Delay\_1MS()：用于OLED初始化前的延时，用户根据自身环境实现1ms的延时

1.3.2 oled\_input.c



用户需要自行实现以上四个函数，分别是

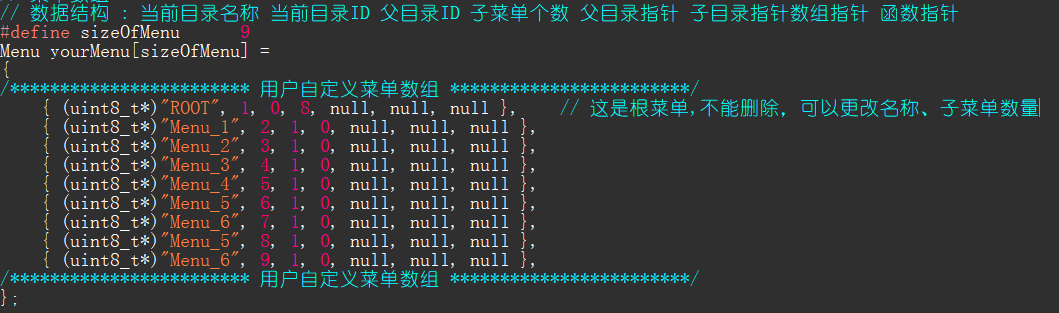
uint8\_t ifPress\_UP：按键“上”的检测函数，按下则返回1，否则返回0

uint8\_t ifPress\_DOWN ：按键“下”的检测函数，按下则返回1，否则返回0

uint8\_t ifPress\_YES：按键“确定”的检测函数，按下则返回1，否则返回0

uint8\_t ifPress\_NO：按键“取消”的检测函数，按下则返回1，否则返回0

1.3.2 oled\_interface.c



用户需要修改两个内容

#define sizeOfMenu 9 ：用户自定义菜单的个数

Menu yourMenu[sizeOfMenu] =

{

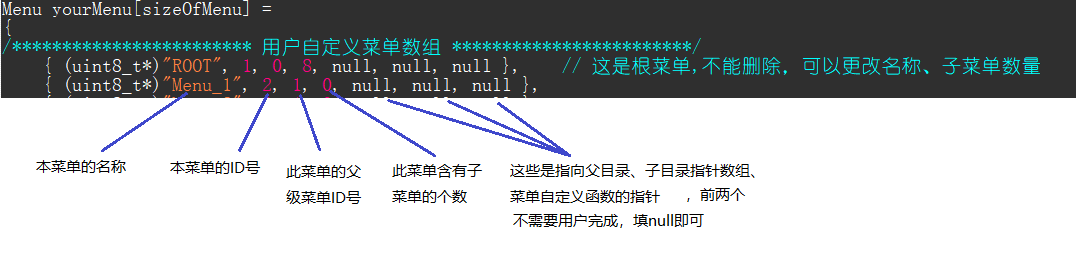
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 用户自定义菜单数组 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

{ (uint8\_t\*)"ROOT", 1, 0, 8, null, null, null }, // 这是根菜单,不能删除，可以更改名称、子菜单数量

… …

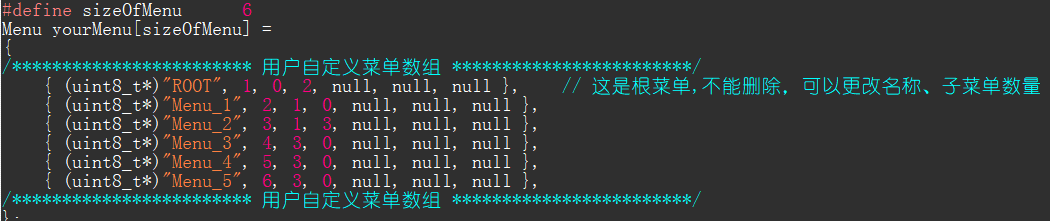
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 用户自定义菜单数组 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

};

各成员意义： 

示例：

代码：



1.3.3 main.c

初始化需要用到一下两个函数

OLED\_Init();

menuInit();

进入菜单需要调用以下函数

enterMainMenu()

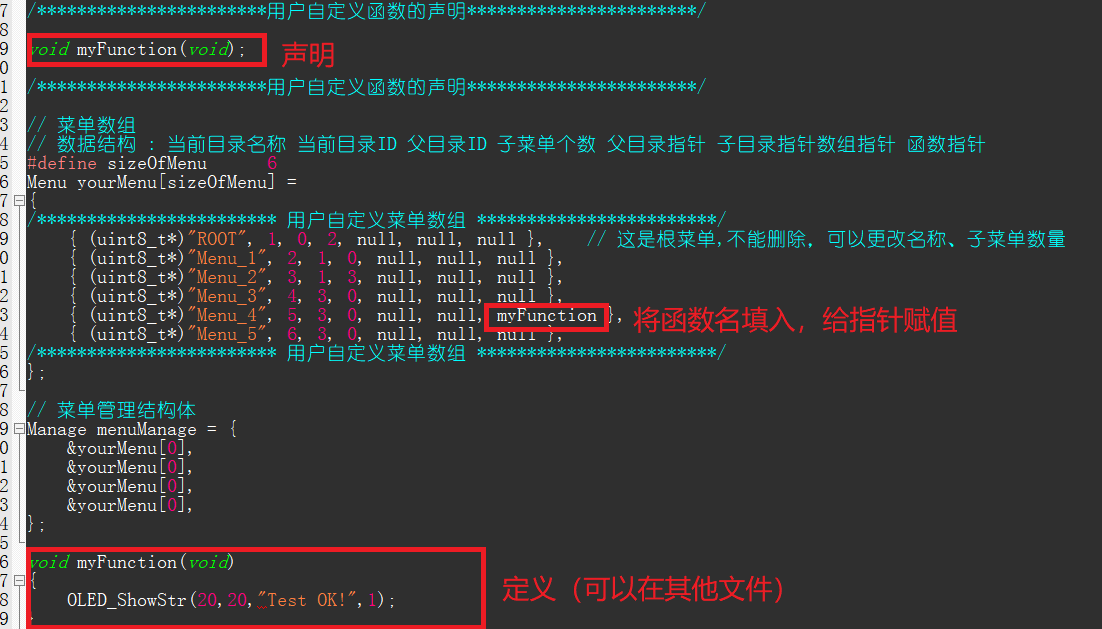
**2.中阶使用**

2.1 用户自定义函数

用户进入最后一级菜单（没有子菜单时），再按下确认键可以调用用户自行编写的自定义函数。实现方法如下：

2.1.1 定义函数

用户可以在任何地方定义函数，但必须在oled\_interface.c中的MenuyourMenu[sizeOfMenu]定义之前进行声明。可以在其他文件定义。

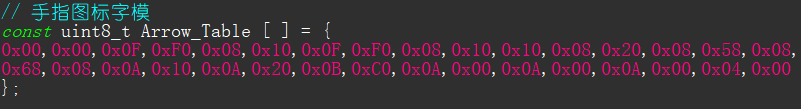


2.2 自定义图标

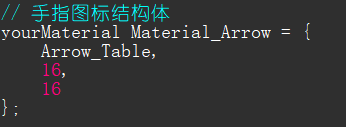
本ui界面支持自定义图标，且封面也是一个图标，也就是封面也可以方便的自定义。

定义一个图标的方法：

2.2.1 在fonts.c声明一个数组，格式如下



2.2.2定义一个结构体，将刚刚声明的字模数组名作为数据的指针，第二，第三个参数分别为宽和高。



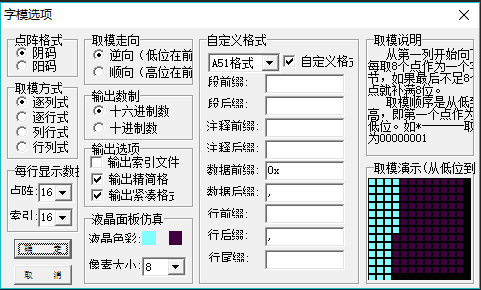
2.2.3 使用的方法：

完整的定义好一个图标结构体之后，便可调用函数来显示。参数为结构体的地址。



2.2.4 取模方法

可以采用软件 PCtoLCD 进行取模，取模的配置如下



PCtoLCD支持文字、bmp图片取模，还支持从txt批量取模文字，比较方便。

**3.高阶使用**

自行修改源码，然后想怎么用怎么用。

附录：

**ASCII从32到126的字符**

!"#$%&'()\*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz{|}~

（！号前有个空格，~号后面没有回车）