**Image extraction**

简介

在做嵌入式单片机开发的时候需要用LCD/OLED显示一张图片，最直接和简单的方式是用相应的工具将图片转换为相应的数据点（其实是一个2维的数组阵列），再用相应的代码将数组按照一定的时序显示到LCD/OLED显示屏上即可。

操作步骤

1.点击[下载Image2Lcd工具](https://www.waveshare.net/w/upload/b/bd/Image2Lcd2.9.zip" \t "_blank)  
2.双击进入工具界面  
[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_1.jpg)  
3.打开需要提取的图片（这里以1.47inch LCD Module的图片为例）

* 参数设置

输出数据类型：C语言数组(\*.c)

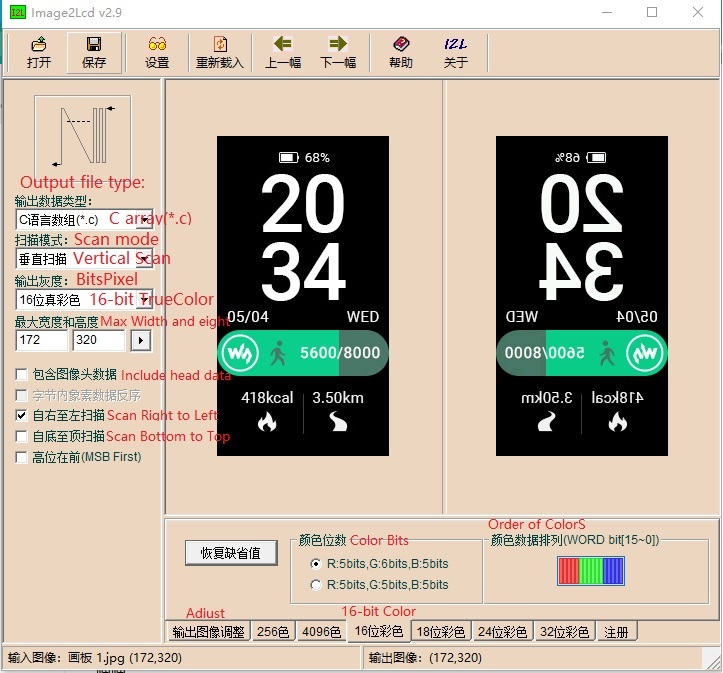
扫描模式：垂直扫描

输出灰度：16位真彩色

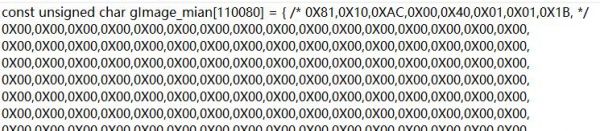
最大宽度和高度(根据图片实际纵横比去设置)：172x320

输出图像调整：默认设置就好，需要颜色反转的勾选颜色反转

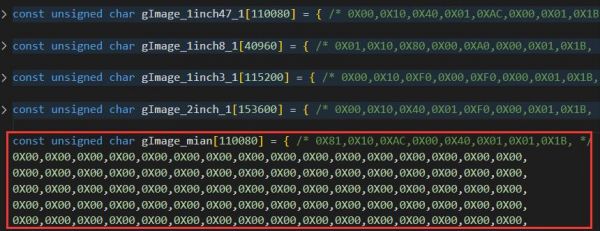
点击16位彩色：颜色位数RGB565,颜色数据排列RGB

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_2.jpg)  
4.设置完参数后选择保存

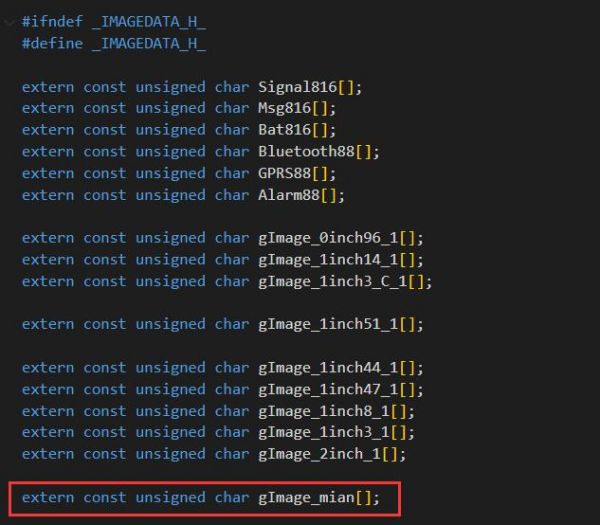
* 路径跟命名由用户自行设置
* 文件类型为.c文件
* 保存后会生成数组，ctrl+A全选后

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_4.jpg)

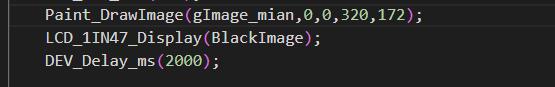
* 复制到Image.c文件里面

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_6.jpg)

* 并在Image.h里面声明

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_7.jpg)

* 最后在主函数调用，映射到屏幕就可以了

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_5.jpg)

效果展示

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Img2Lcd_3.jpg)

# OLED Draw

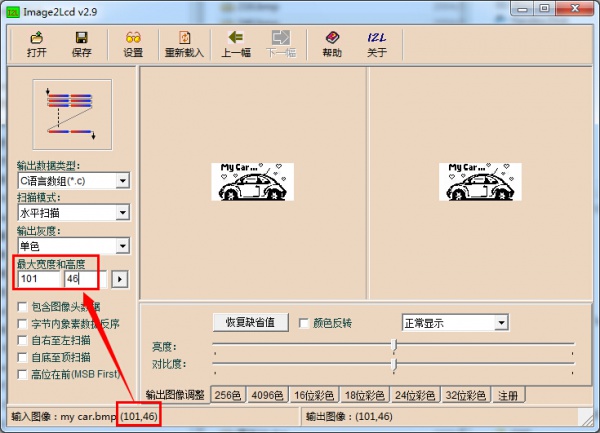
## 图片取模

使用[Image2Lcd](https://www.waveshare.net/w/upload/b/bd/Image2Lcd2.9.zip)进行去取模

运行Img2Lcd.exe，点击打开，选择my car.bmp，输出数据类型：C语言数组（\*.c）,扫描模式：水平扫描，输出灰度：单色。

最大宽度和高度由输入图像的实际值决定，这里是101和46。点击保存，生成相应的数组。

具体设置如下图：

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:OLED_APP01.jpg)

这里以默认的代码为例：

在Fonts.c中添加上面生成的数组：

const uint8\_t c\_my\_car[598] = { */\* 0X00,0X01,0X65,0X00,0X2E,0X00, \*/*

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X14,0X00,0X00,0X00,0X00,0X00,0X00,0X06,0X30,

0X01,0XE0,0X00,0X00,0X2A,0X00,0X00,0XD8,0X00,0X00,0X00,0X0F,0X7B,0X63,0XE0,0X00,

0X00,0X22,0X00,0X01,0X24,0X00,0X00,0X00,0X0F,0XFB,0X63,0X07,0X34,0X00,0X14,0X00,

0X01,0X04,0X00,0X00,0X00,0X0D,0XDB,0X63,0X01,0XBC,0X00,0X08,0X00,0X00,0X88,0X00,

0X00,0X00,0X0C,0X1B,0X63,0X07,0XB0,0X00,0X00,0X00,0X00,0X50,0X00,0X00,0X00,0X0C,

0X1B,0X63,0XED,0XB0,0XDB,0X00,0X01,0X00,0X20,0X00,0X00,0X00,0X0C,0X19,0XE1,0XE7,

0XB0,0XDB,0X00,0X02,0X00,0X00,0X00,0X00,0X00,0X0C,0X19,0XE1,0XE7,0XB0,0XDB,0X00,

0X02,0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X60,0X00,0X00,0X00,0X00,0X04,0X00,0X00,

0X00,0X00,0X00,0X00,0X03,0XC0,0X00,0X00,0X00,0X00,0X08,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X00,0X00,0X00,0X08,0X05,0X00,0X00,0X00,0X00,0X00,0X00,0X00,

0X00,0X00,0X00,0X00,0X10,0X0A,0X80,0X00,0X00,0X00,0X00,0X00,0X0A,0X00,0X00,0XFF,

0XF8,0XE0,0X05,0X00,0X00,0X00,0X00,0X06,0XC0,0X15,0X00,0X1F,0XFF,0XFF,0XC0,0X02,

0X00,0X00,0X00,0X00,0X09,0X20,0X11,0X00,0XFB,0XFF,0XE1,0XF8,0X00,0X00,0X00,0X00,

0X00,0X08,0X20,0X0A,0X03,0X9F,0X00,0X9E,0X3E,0X00,0X00,0X00,0X00,0X00,0X04,0X40,

0X04,0X0E,0X70,0X00,0X81,0XC7,0X80,0X01,0X40,0X00,0X00,0X02,0X80,0X00,0X1D,0X80,

0X00,0XE0,0X61,0XE0,0X02,0XA0,0X00,0X00,0X01,0X00,0X00,0X77,0X9F,0XFC,0XF0,0X18,

0XF8,0X02,0X20,0X00,0X00,0X00,0X00,0X01,0XDF,0X00,0X00,0XF3,0X0C,0X3C,0X01,0X40,

0X00,0X00,0X00,0X00,0X03,0X9E,0X00,0X00,0XF8,0X06,0X1E,0X00,0X80,0X00,0X00,0X00,

0X00,0X07,0X0E,0X30,0X01,0XFC,0X7F,0X07,0X00,0X00,0X00,0X00,0X00,0X00,0X07,0X0E,

0X30,0X01,0XFC,0X7F,0X07,0X00,0X00,0X00,0X00,0X00,0X00,0X7E,0XFF,0XFF,0XFF,0XFF,

0XFF,0X83,0XC0,0X00,0X00,0X00,0X00,0X0F,0XFD,0XFF,0XFF,0XFF,0XFF,0XFC,0X01,0XF0,

0X00,0X00,0X00,0X00,0X3F,0X81,0XFF,0X00,0X00,0X00,0X00,0X1F,0XF0,0X00,0X00,0X00,

0X00,0XFE,0XF0,0X00,0X00,0X01,0X00,0X00,0XC0,0X1C,0X00,0X00,0X00,0X03,0XC0,0X07,

0X80,0X00,0X03,0XE0,0X07,0X00,0X1C,0X00,0X00,0X00,0X03,0X80,0X03,0X40,0X00,0X03,

0XE0,0X0C,0X00,0X0E,0X00,0X00,0X00,0X07,0X00,0X00,0XC0,0X00,0X00,0X20,0X18,0X00,

0X07,0X00,0X00,0X00,0X09,0X01,0X80,0X60,0X00,0X00,0X20,0X73,0X9F,0X03,0X80,0X00,

0X00,0X1E,0X0F,0XF2,0X20,0X00,0X00,0X20,0X67,0XFF,0XC1,0XC0,0X00,0X00,0X1E,0X1F,

0XF9,0X30,0X00,0X00,0X20,0XEF,0XFF,0XE0,0XE0,0X00,0X00,0X38,0X3F,0XFC,0X90,0X00,

0X00,0X20,0XCB,0XFF,0XF9,0XF0,0X00,0X00,0X7F,0X7F,0XFE,0X10,0X00,0X00,0X20,0XDB,

0XFF,0XFF,0XF0,0X00,0X00,0X6E,0XF9,0XBF,0X10,0X00,0X00,0X20,0XF7,0XED,0XFF,0XF0,

0X00,0X00,0X7C,0XFC,0X3F,0X10,0X00,0X00,0X20,0X67,0XE1,0XFD,0XE0,0X00,0X00,0X7C,

0XFC,0X3F,0X10,0X00,0X00,0X20,0X67,0XE1,0XFD,0XE0,0X00,0X00,0X38,0XFC,0X3F,0X18,

0X00,0X00,0X20,0X87,0XE0,0XFD,0XC0,0X00,0X00,0X38,0XF1,0X8F,0X18,0X00,0X1F,0XE7,

0X87,0X9C,0X7D,0XC0,0X00,0X00,0X1C,0XFE,0X1F,0X81,0XFF,0XFF,0XC0,0X0F,0XE0,0XFF,

0X80,0X00,0X00,0X0C,0XFC,0X3F,0X80,0X00,0X00,0X00,0X0F,0XE1,0XF8,0X00,0X00,0X00,

0X00,0XF9,0XBF,0XFF,0XFF,0XFF,0XFF,0XFF,0XEC,0XFE,0X00,0X00,0X00,0X03,0XF9,0XBF,

0XFF,0XFF,0XFF,0XFF,0XFF,0XFF,0XFF,0XC0,0X00,0X00,0X0F,0XFF,0XFF,0XFF,0XFF,0XFF,

0XFF,0XFF,0XFF,0XFF,0XC0,0X00,};

在Fonts.h中添加声明：

extern const uint8\_t c\_my\_car[598];

在main.c中修改代码如下即可：

#include "LIB\_Config.h"

int main(void)

{

system\_init();

ssd1306\_clear\_screen(0x00);*//清屏*

ssd1306\_draw\_bitmap(0, 0, &c\_my\_car[0], 101, 46);*//显示坐标为（0，0）*

ssd1306\_refresh\_gram();*//更新GRAM*

while (1) {

}

}

具体代码：[0.96inch-OLED-Code-Draw](https://www.waveshare.net/w/upload/6/61/0.96inch-OLED-Code-Draw.zip)

## 扩展

同样，汉字取模方法类似，这里使用LcmZimo(链接：<https://pan.baidu.com/s/10sSyEJGG-ltjWEdqr7hiKw> 提取码：hj21 )进行取模，默认按下图设置

其中红色部分不可修改，其它可按实际需要选择，点击确认参数后，开始输入相应的中文字符串进行取模。用显示图片类似的方法保存到工程中，调用时注意数组的边界，如使用16点阵，则在显示图片的大小都设置为16。

用户也可在函数ssd1306\_draw\_bitmap的基础上修改为相应的显示中文字符函数。

[](https://www.waveshare.net/wiki/%E6%96%87%E4%BB%B6:Oled_draw01.jpg)