此文件为原创，请珍惜作者劳动成果。

学习之前先开几个文件：

* IAR\_EWARM\_user\_manual.pdf
* Cortex-M3权威指南Cn.pdf
* 开发板设计说明.pdf
* 开发板原理图.pdf
* STM32F10xxx\_固件函数库-精简版--去掉了最后两页.pdf
* 到 STM32F107器件手册(en).pdf
* 【中文】STM32F系列ARM内核32位高性能微控制器参考手册V10\_1.pdf

分别简称为：《IAR手册》《权威指南》《开发板设计说明》《开发板原理图》《库函数手册》《107手册》 《微控制器手册》

# 1.创建新工程

1) 按照LED方法，拷备相应文件库。（可将usart整个文件夹拷备入）

2) 在Drivers 文件夹下拷备入以下三个LCD的驱动文件（同单片机课程的驱动）

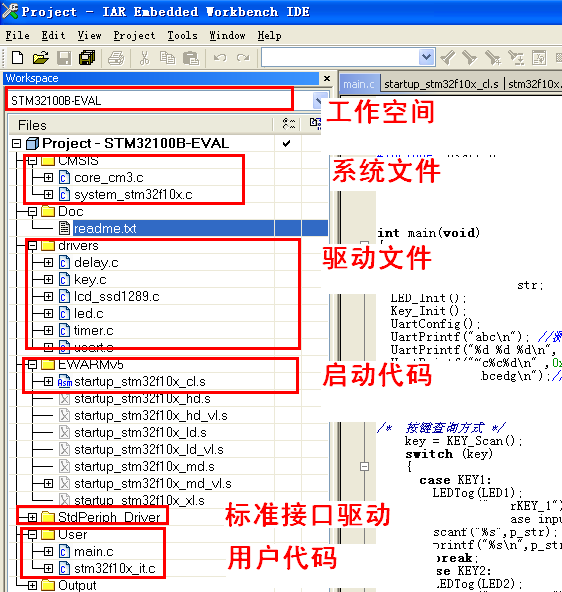
lcd.h lcd\_font.h lcd.c

***液晶屏的驱动头文件***、***触摸屏字体驱动，液晶屏的驱动C文件***、

3)如果需要显示图片及汉字，还需要拷入相应的\*.h文件

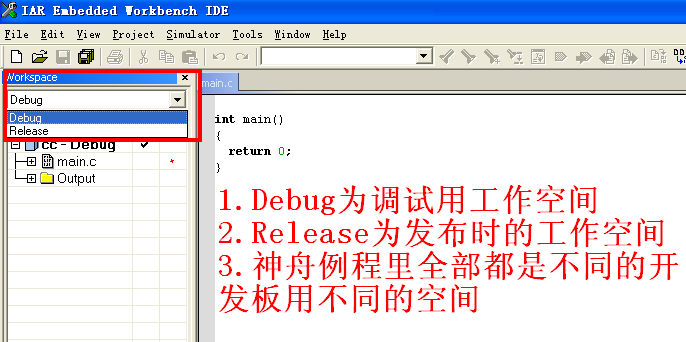
Picture.h hanzi.h

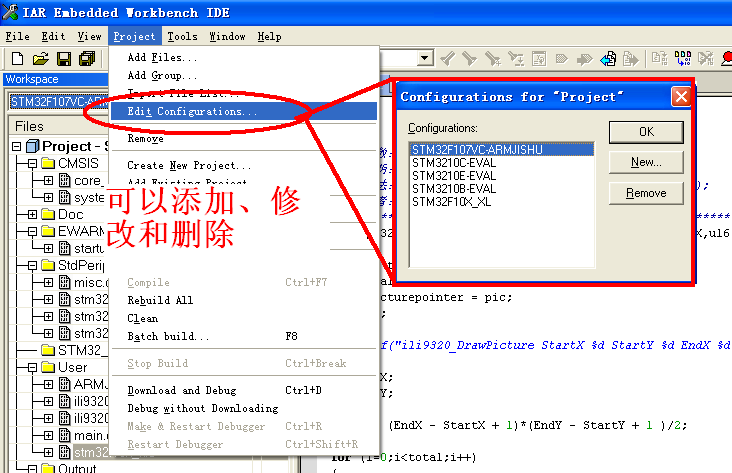
# 2.工程文件添加修改



# 3.工作空间的修改

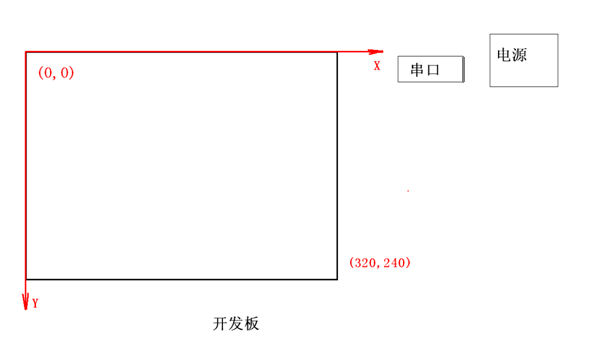
在新建工程时，一般会有两种工作空间，





# 4.图像头文件的建立和修改 使用图像转数组工具ImageLcd

液晶屏幕的坐标方向（0，0） 和 x y方向最大值的点（320,240）坐标位置

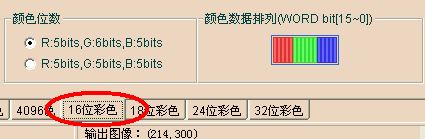


图像显示原理：

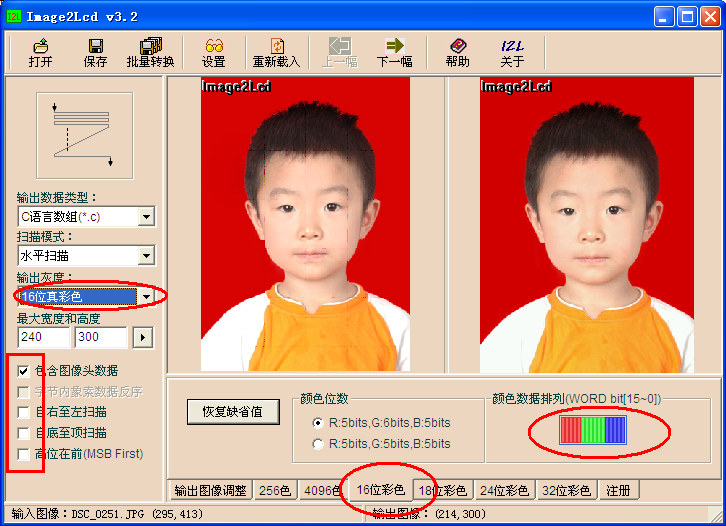
LCD上一共有320行\*240列的点阵，每一个点上有一个颜色，每一种颜色由一个数据表示。如果是24位真彩色，共占24位，B（8位），G（8位） R（8位）



16位真彩色格式：排列顺序为5-6-5

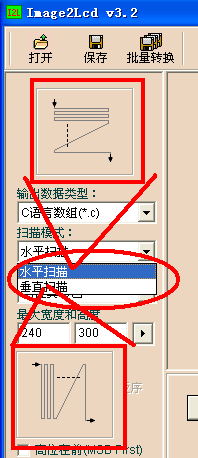


使用图像->数组转换工具 先打开一幅图像

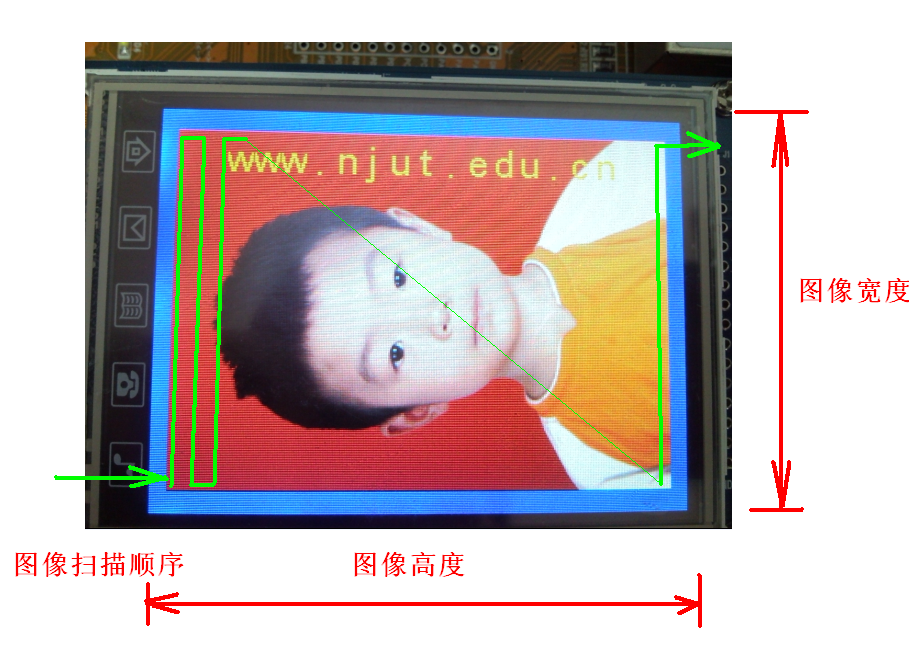


观察几个参数：扫描模式、输出灰度、宽度和高度、扫描方式，修改后看显示结果，并讲解

水平垂直扫描区别：

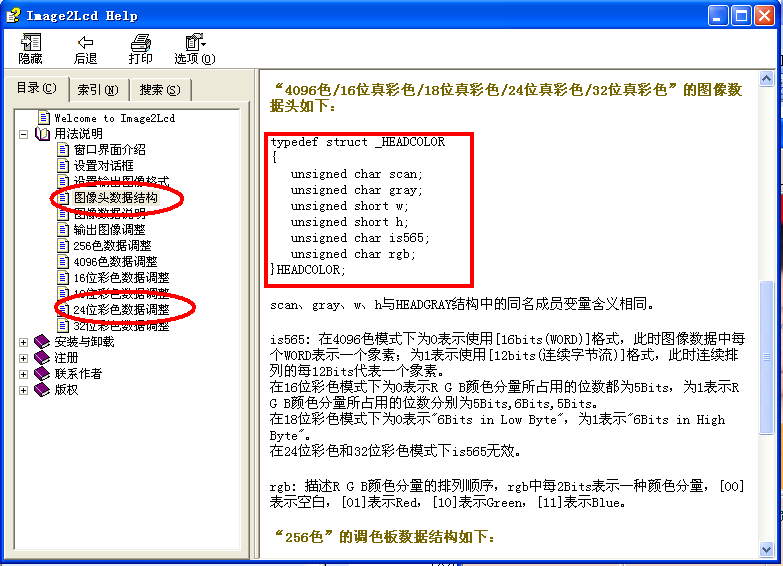


16位真彩色与24位真彩色区别：24色明显要明亮，色彩丰富



高度320 宽度240

学会使用帮助文件，查看头文件信息



scan: 扫描模式

Bit7: 0:自左至右扫描，1:自右至左扫描。

Bit6: 0:自顶至底扫描，1:自底至顶扫描。

Bit5: 0:字节内象素数据从高位到低位排列，1:字节内象素数据从低位到高位排列。

Bit4: 0:WORD类型高低位字节顺序与PC相同，1:WORD类型高低位字节顺序与PC相反。

Bit3~2: 保留。

Bit1~0: [00]水平扫描，[01]垂直扫描，[10]数据水平,字节垂直，[11]数据垂直,字节水平。

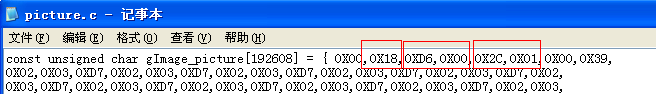
gray: 灰度值

   灰度值，1:单色，2:四灰，4:十六灰，8:256色，12:4096色，16:16位彩色，***24（0x18）:24位彩色***，32:32位彩色。

w: 图像的宽度。***0x00D6***

h: 图像的高度  ***0x012C***

查看一个生成的**24位图像头数据**：



查看数据：0x02 0x03 0xD7 分别为第一个点的B G R ，其中红色成分最多。

其余内容同单片机课程文档：《TFT显示图片及汉字方法.doc》

# 5. 图像显示示例 添加和代码修改

## 1.代码添加testlcd函数

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 名 称：testlcd （测试函数）

\* 功 能：调用图像，显示，调用字符串并显示

\* 入口参数：无

\* 出口参数：无

\* 说 明：

\* 调用方法：在finsh中实现

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void testlcd(void)

{

int i;

lcd\_config();

lcd\_clr\_screen(BLUE);

for( i=0;i<319;i++)

{

lcd\_set\_point(i, 30, GREEN);

}

lcd\_put\_string\_16x24(0, 35, "picture 120X168 :", WHITE, BLUE);

for( i=0;i<319;i++)

{

lcd\_set\_point(i, 60, RED);

}

//lcd\_draw\_picture(0,0,0XA7,0X77,(u16 \*)gImage\_picture+8);

DrawPicture\_Center((u16 \*)gImage\_picture);

for(i=0;i<319;i++)

{

lcd\_set\_point(i, 180, RED);

}

lcd\_draw\_hanzi(5,180,(u8\*)Hanzi, WHITE, HYALINE);

for(i=0;i<319;i++)

{

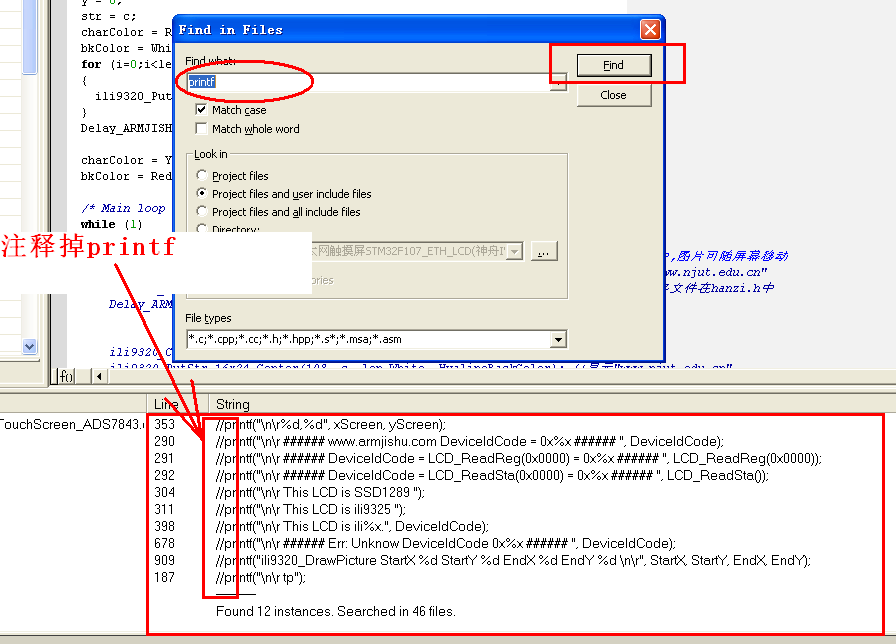
lcd\_set\_point(i, 210, GREEN);

}

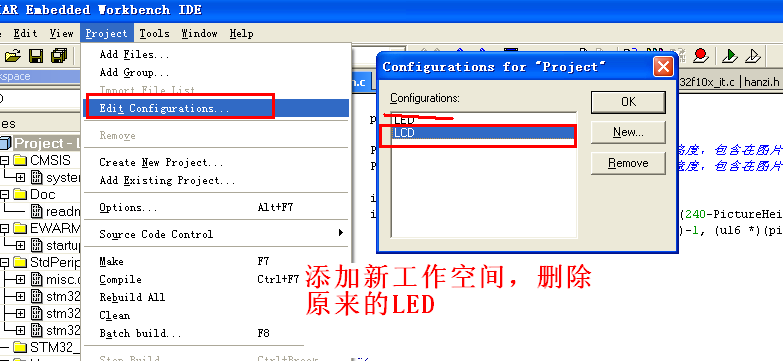
}

## 2.代码添加 main.c

testlcd();

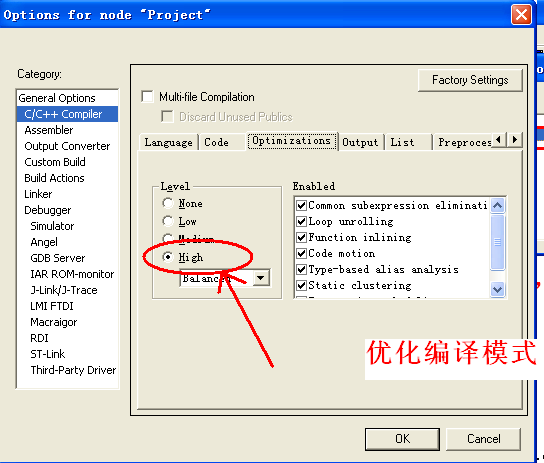


## 4.修改工作空间名称

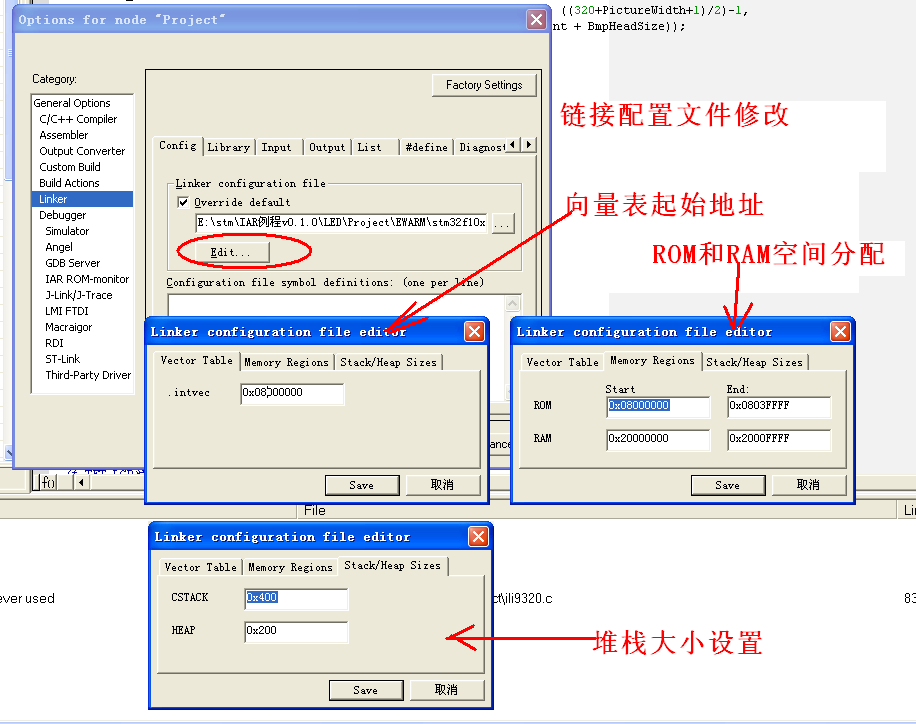


## 5.工程属性修改

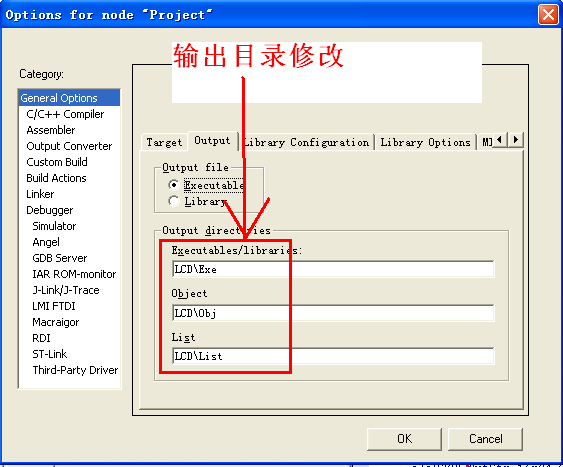
编译模式优化，因为头文件中代码量过大，一共包含三个大型的头数据文件，两个图片，一个文字：picture.h picture1.h hanzi.h，



链接文件配置



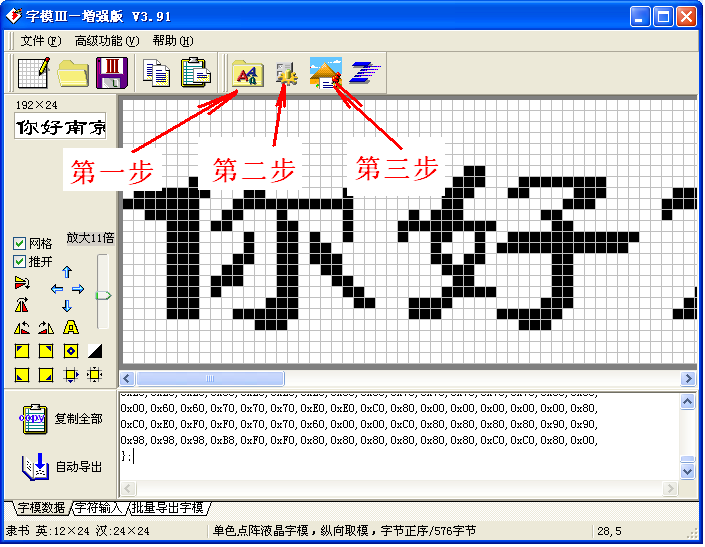
输出目录修改



# 6.汉字显示 使用字模软件

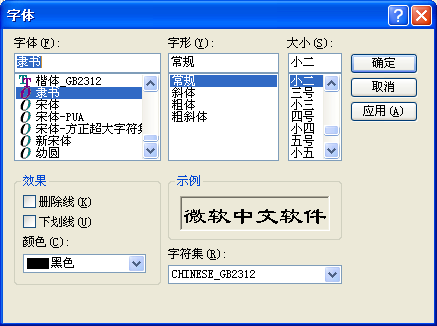
一种方式做成图形文件，同步骤6，但是这种方式占用内存太大

第二种方式，用取字模软件，生成字模的汉字库

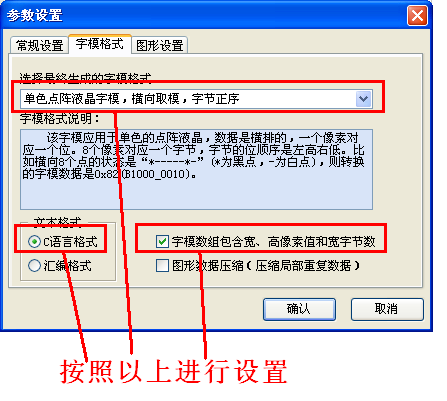


用上学期单片机课程所用的字模生成软件，按步骤设置如下

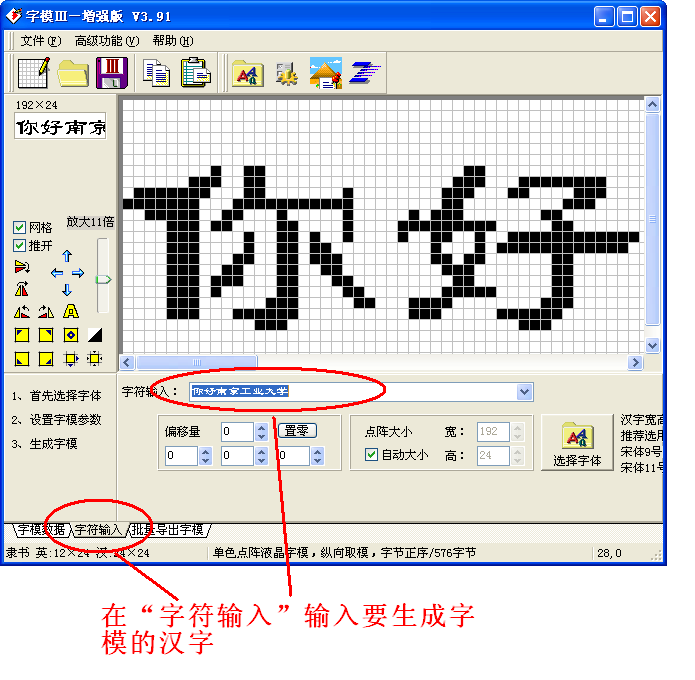
第一步



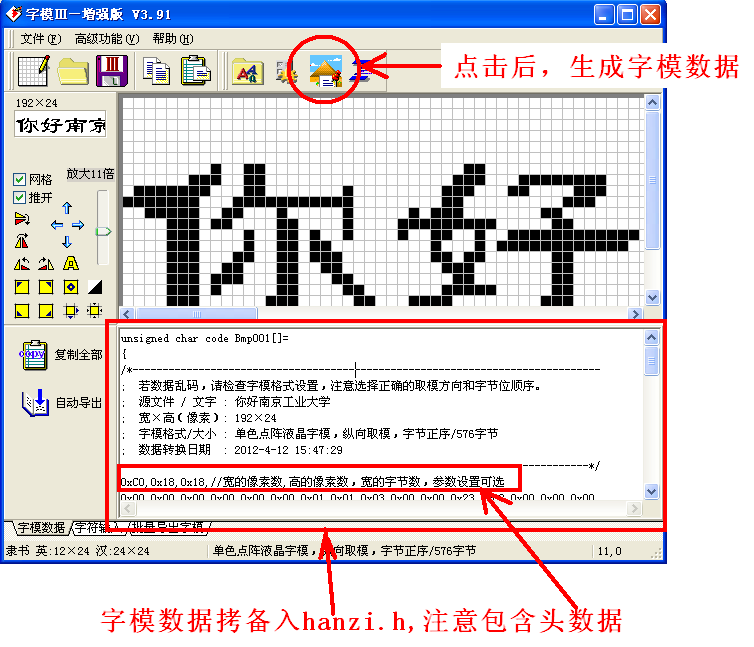
第二步

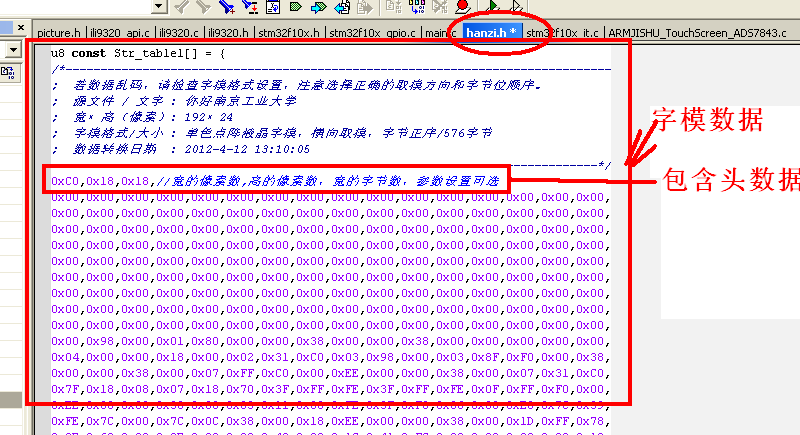


汉字显示软件操作：



第三步，生成字模数据





# 7. 液晶驱动

液晶模块的驱动很多，常用ili9325 SSD1289,不同的驱动模块初始化代码不同，运行时的操作也不同，其模块的驱动可通过读取寄存器0x0000获得。

例程针对以上不同模块作了兼容处理，首先读取ID

DeviceIdCode = lcd\_read\_reg(0x0000);

再根据不同的ID，运行相应的程序。

if(DeviceIdCode==0x8989)//SSD1289

{…

}

else if(DeviceIdCode==0x9325)// ili9325

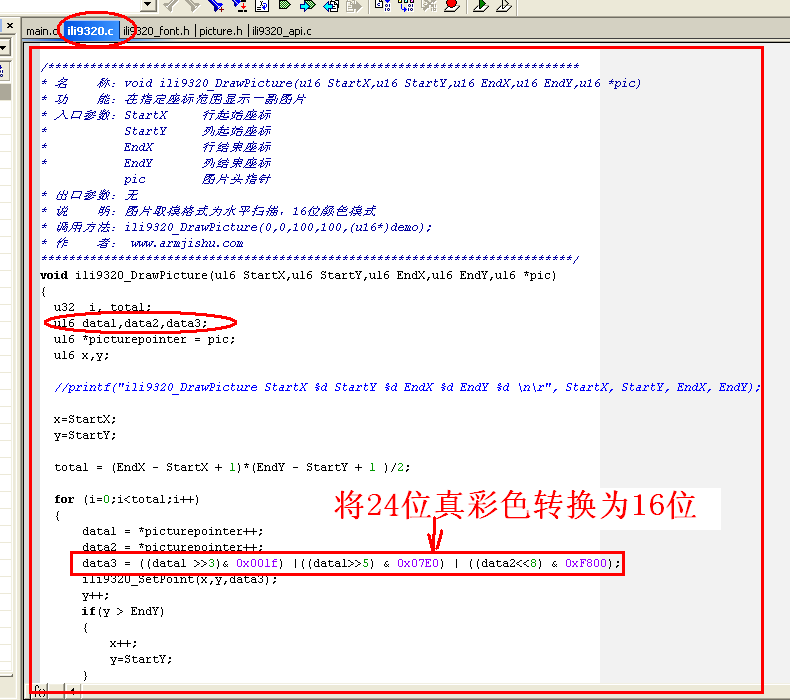
{…

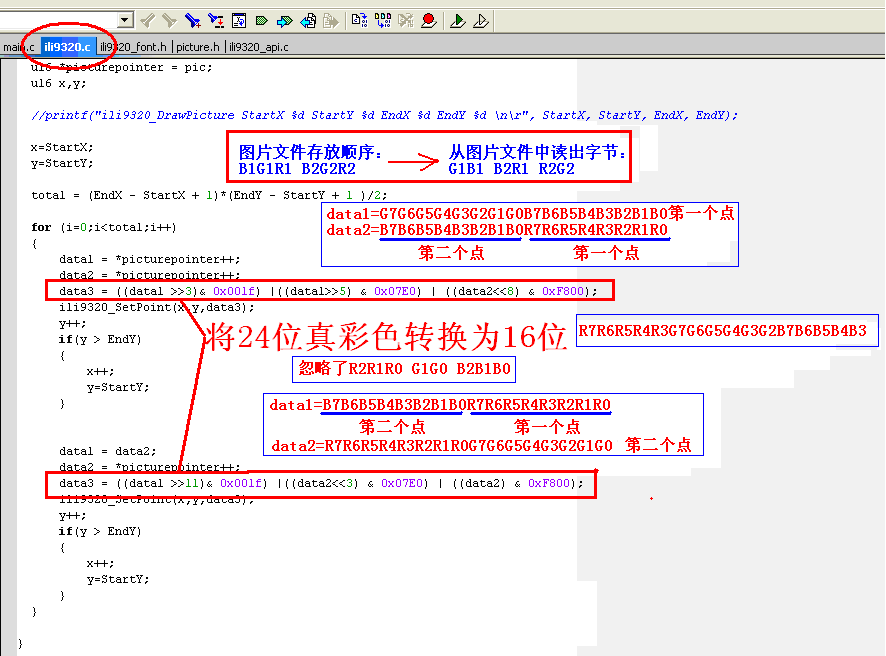
}

# 附录1 神州ili9320驱动解释

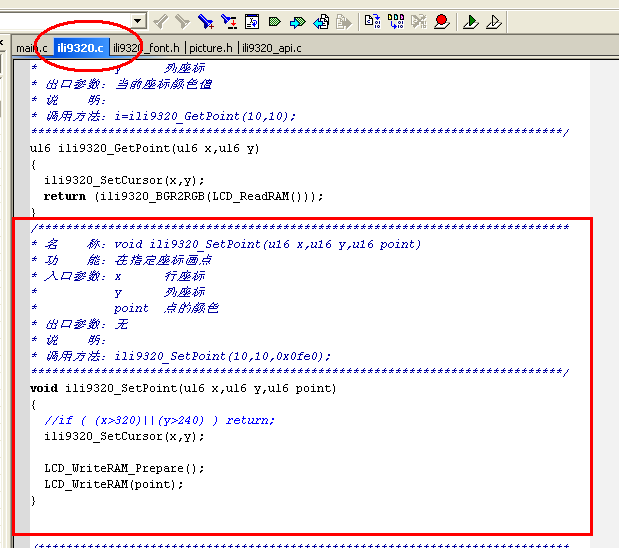
## 1. 图像文件

**---------🡪**

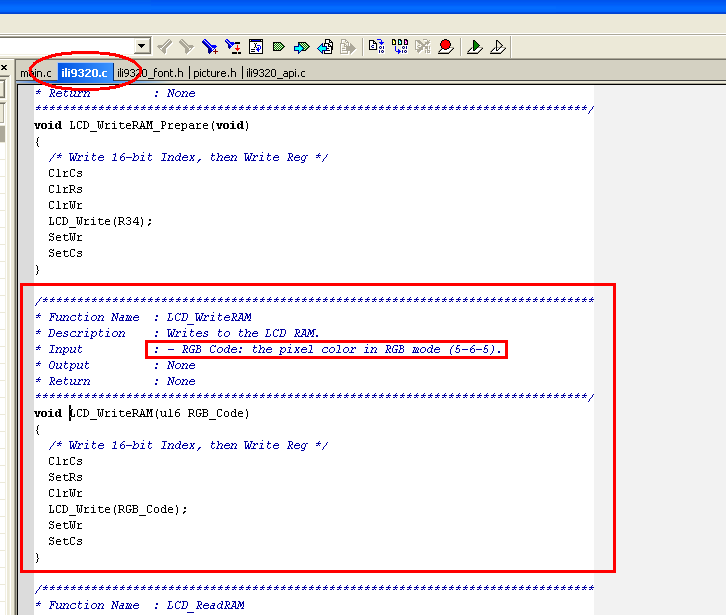




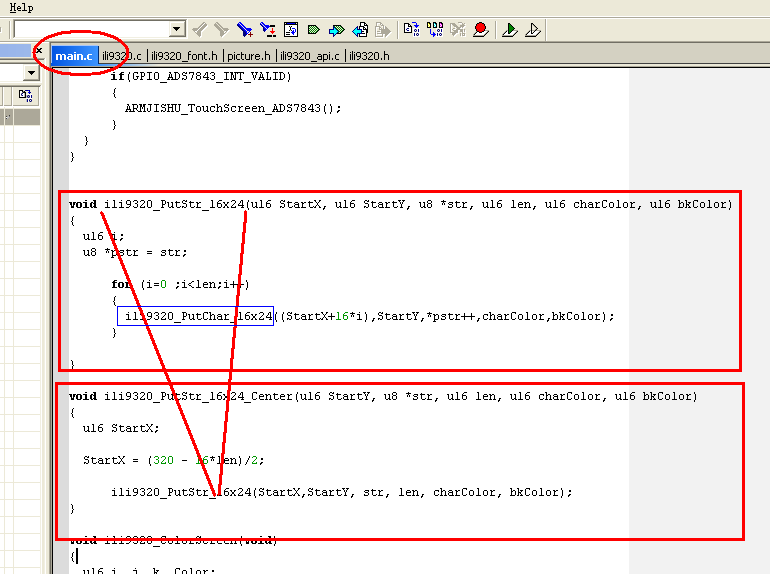
用SetPoint 进行显示点设置

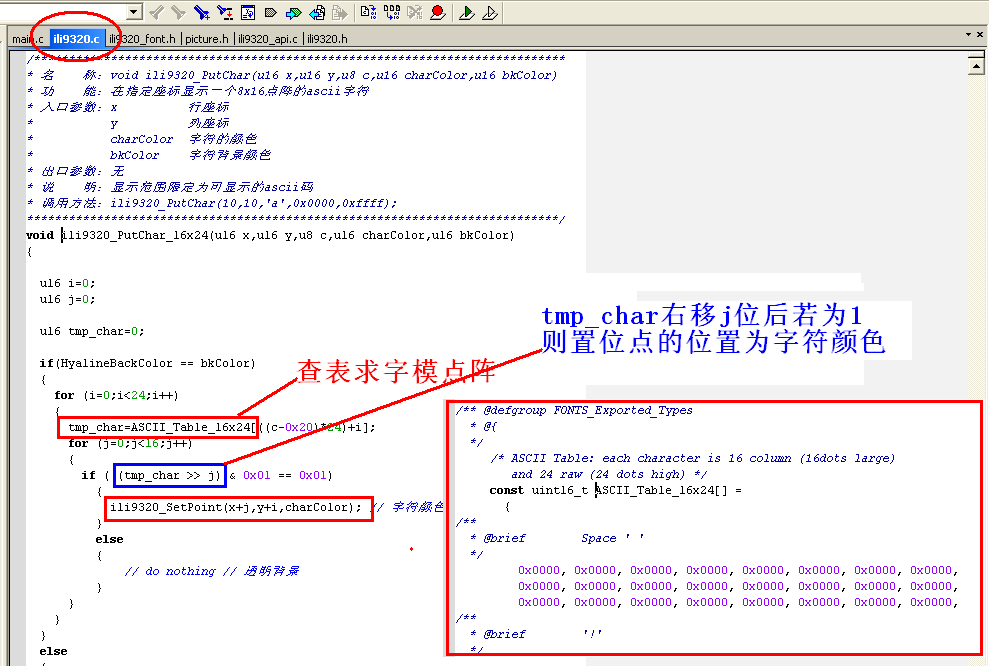


写显示控制器



## 2.字符显示





16行 24列

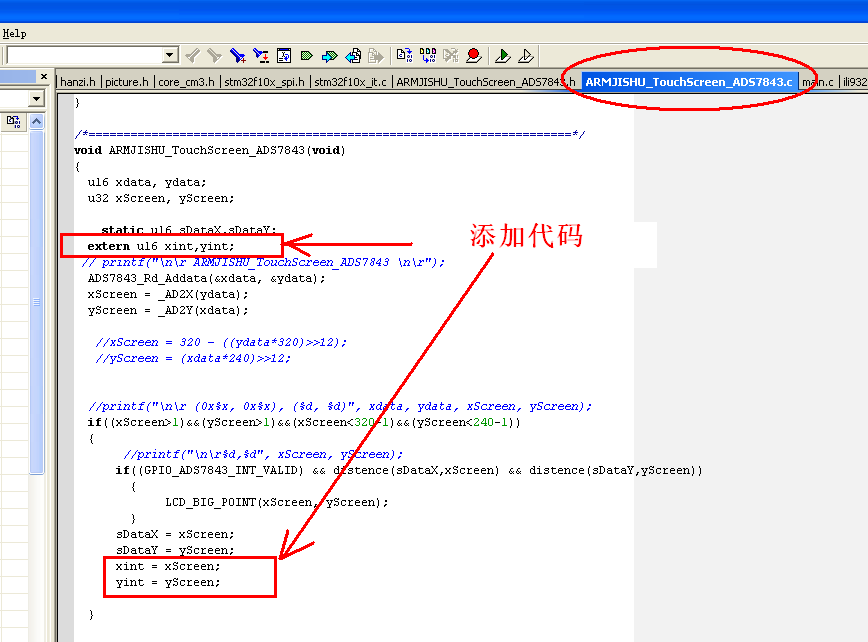
X 方向从0-23

Y方向从0-15

## 3.触摸屏设置 照片随触点移动

功能实现照片显示位置随着触摸位置改变而变化：

改动代码在ARMJISHU\_TouchScreen\_ADS7843.c文件的334行：



在中断服务程序中添加代码：

头部：

extern void ARMJISHU\_TouchScreen\_ADS7843(void);

尾部添加：

void EXTI9\_5\_IRQHandler(void)

{

if(EXTI\_GetITStatus(EXTI\_Line5) != RESET)

{

//printf("\n\r tp");

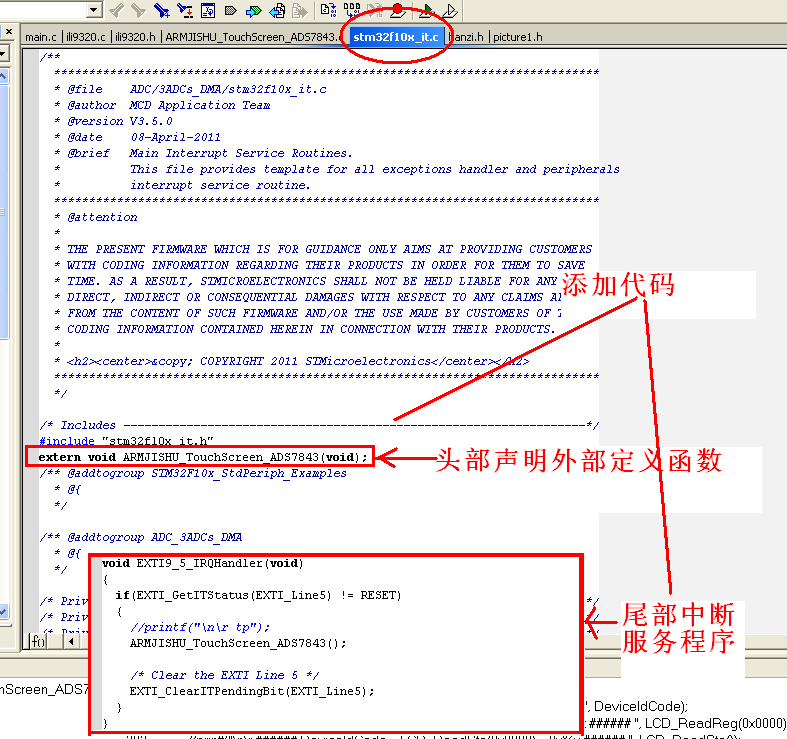
ARMJISHU\_TouchScreen\_ADS7843();

/\* Clear the EXTI Line 5 \*/

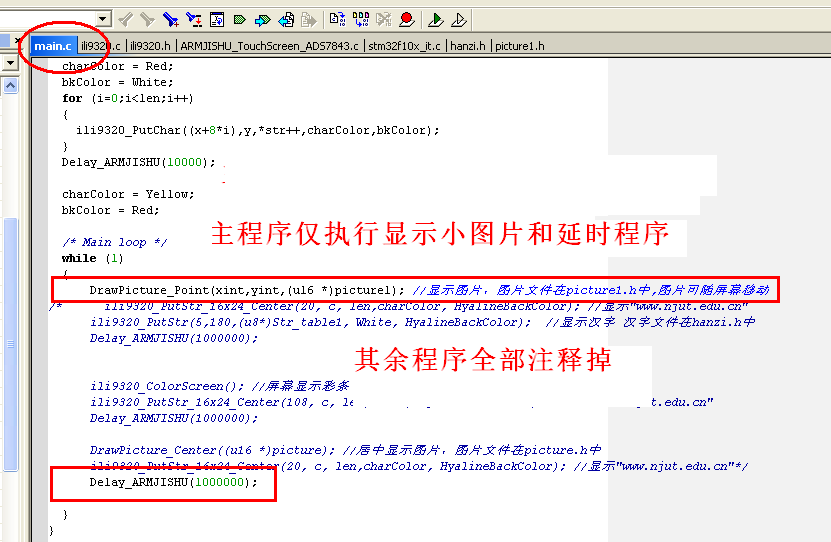
EXTI\_ClearITPendingBit(EXTI\_Line5);

}

｝



在主程序中修改代码：



作业：

1.学会制作不同的图片数组，生成图片头文件，包含入工程，按触摸屏，则翻动图片显示。

2.学会制作图片随触点移动