Good Display' co.,

E-ink 电子纸开发板使用说明手册

Edition 1.1.0

大连佳显电子有限公司

大连市沙河口区工华街 17 号

T: 86 411 84619565 **DD:** +86 411 84526151 **E:** sales@good-display.com

摘要

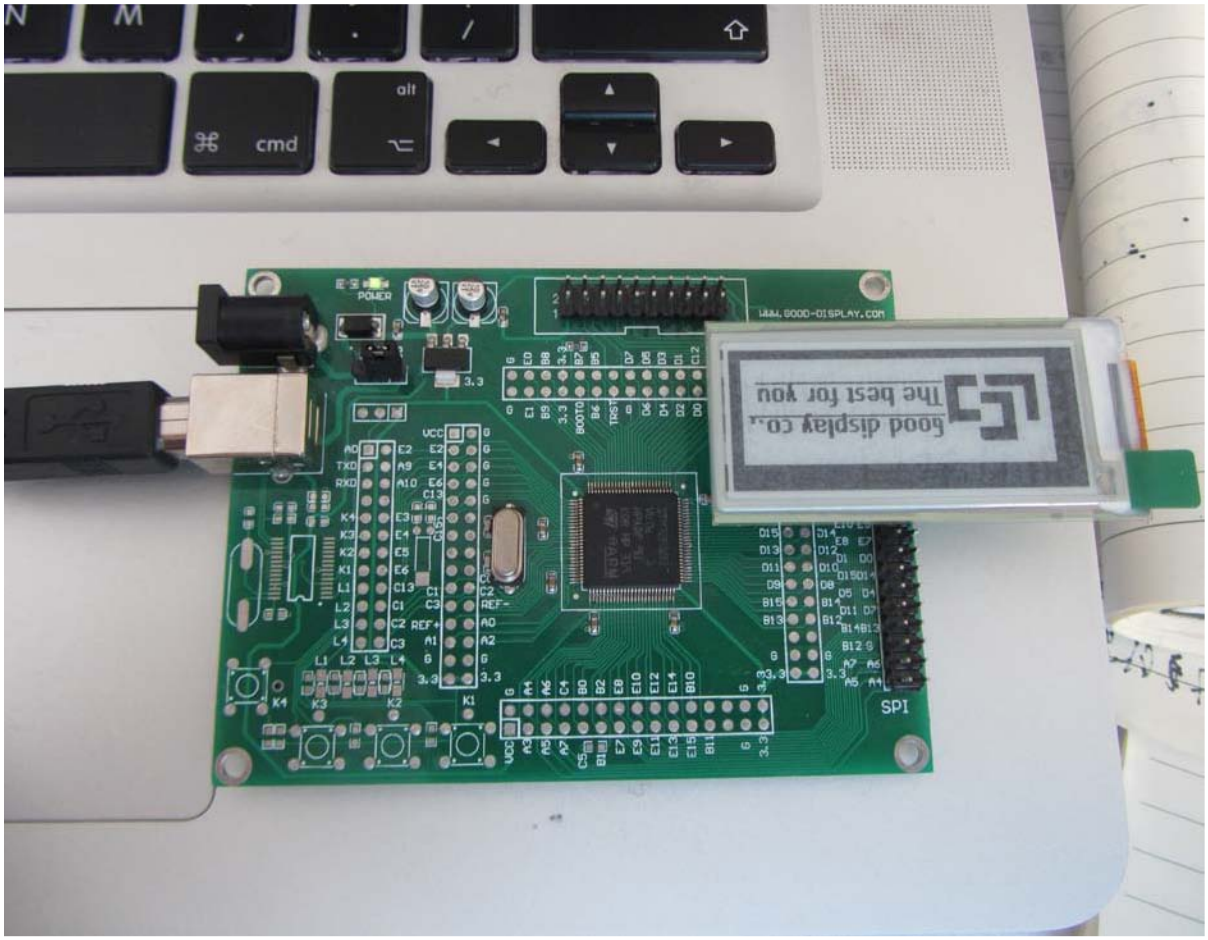
大连佳显电子有限公司 2013 年推出的 2.1 寸、3.5 寸、4.3 寸、6 寸和 8 寸的 E-ink 电子纸开发板是让用户简单上手容易使用电子纸为目的而研制的，对接触过二代计算器语言和 C 语言环境开发的工程师，可以很快了解并应用，此外针对 C 语言的系统函数它可以非常灵活地在该产品上进行控制和操作。Kiel 软件是在很多系统底层编程时需要用到的流行软件，所以我们建议使用该软件进行编译，整个编译和下载过程借助 Debugger 和 Kiel3 的结合完成。同时大连佳显电子有限公司也提供相关的程序例程供客户参考，用户可以向与你联系的销售工程师索取。

针对不同尺寸的电子纸，我们有不同的 Demo kit 可以提供，使用 Arm 单片机存储可以容纳多张图片交替刷新，我们以 2.1 寸的 Demo 开发板为例来说明此类 Demo 开发板的使用方法，其它的 Demo 开发板的使用都与此类似。

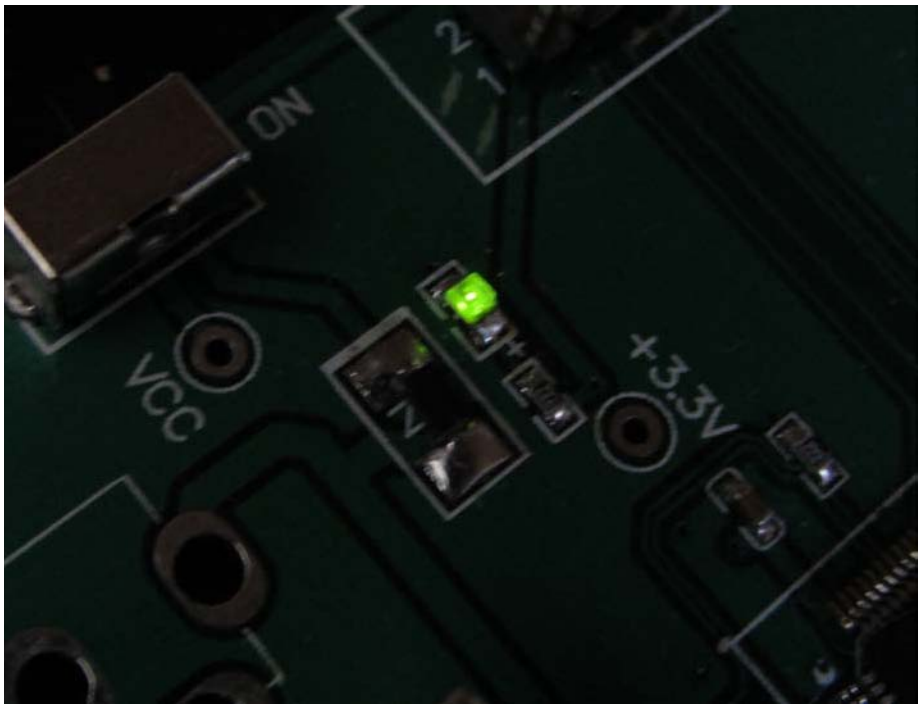
目录:

结构部分：

如果仅仅是想显示内容，只要将 Demo 开发板和电子纸在未上电情况下连接，之后将 Demo 开发板 的 USB 供电插入 PC 的 USB 总线进行供电就可使用了。

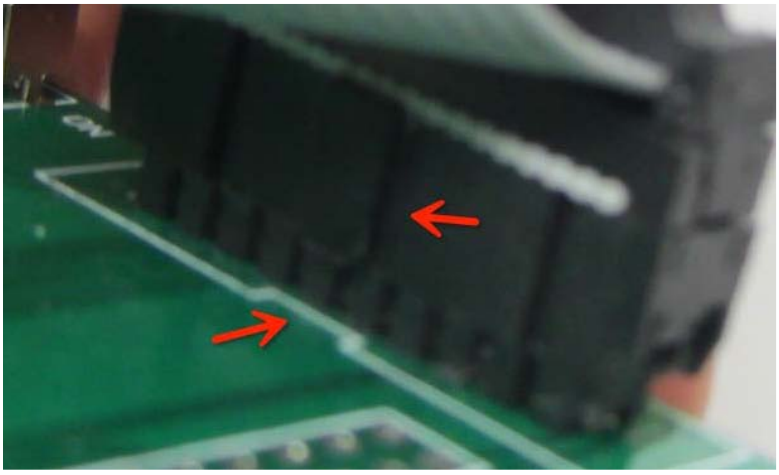


每一个 Demo kit 上留有一枚工作电源指示灯，在上电工作情况下它会保持绿灯常亮，断电后瞬间熄灭，而且在供电源区域内可见一个跳线，用户可以选择使用电源适配器或 USB 供电。



Demo kit 上的 USB 连接线不参与编译后下载，只能供电。下载接口部分是采用标准 JTAG 接口，此处需要与 Debugger 的 JTAG 连接，再用 Debugger 的另一端用 USB 连接计算器的 COM 口。

JTAG 接的连接方向如图所示，将凹槽对像连接线的凸边：

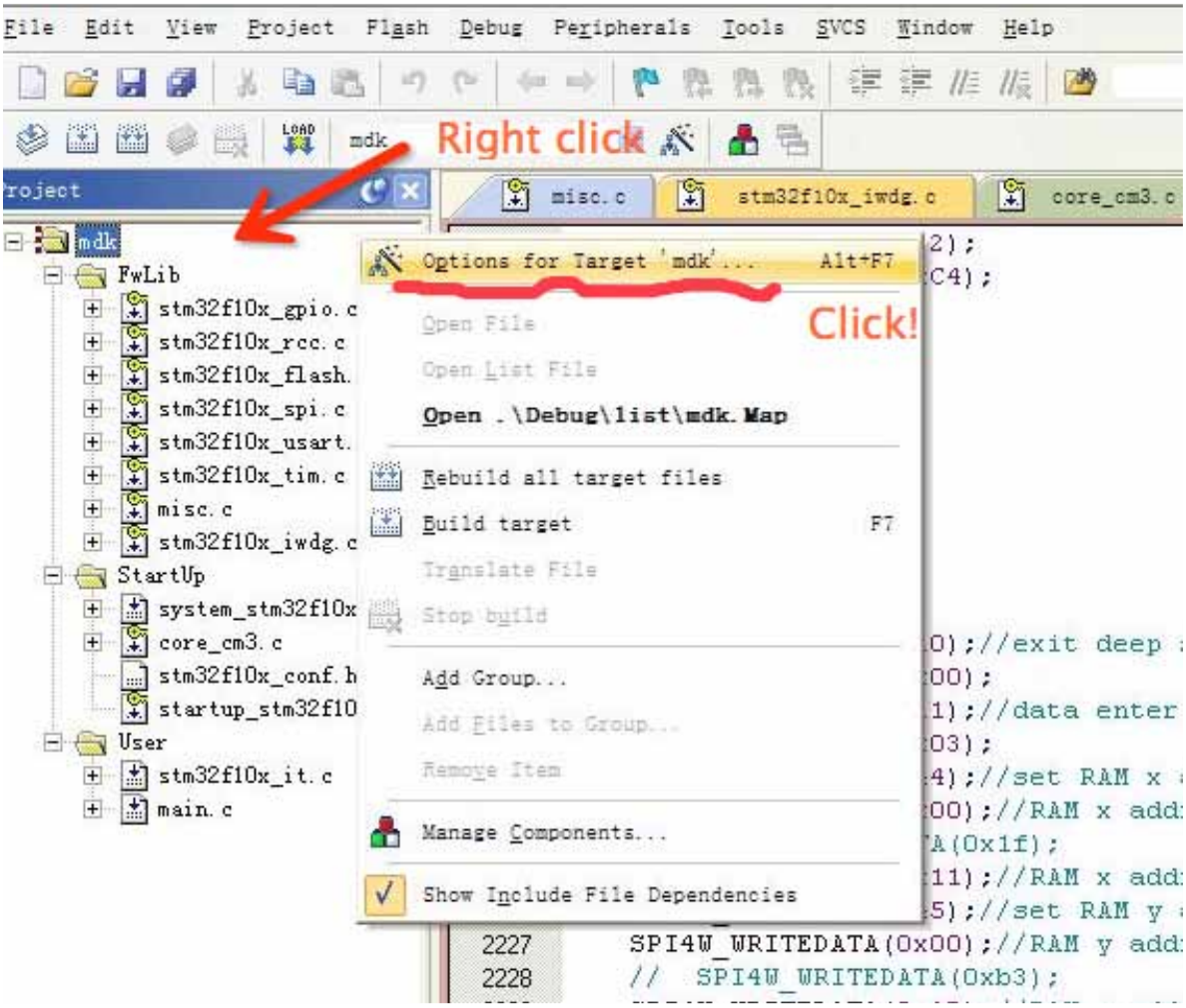


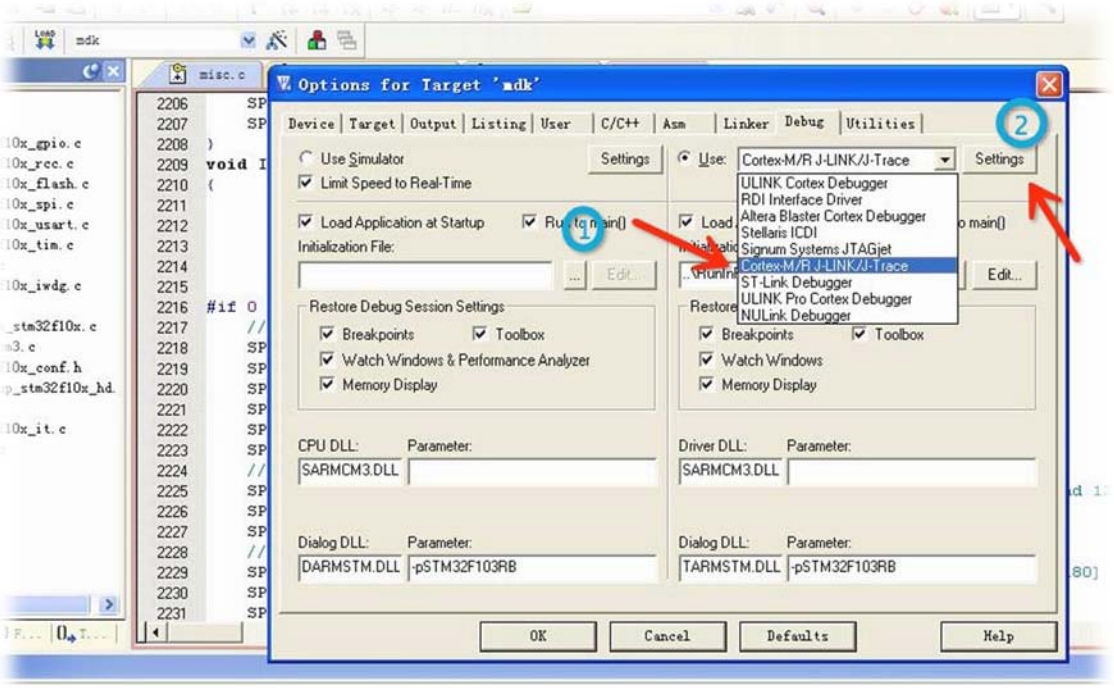
就下载和调试阶段而言，您需要具备一台安装了 Keils3 及以上版本（已注册）的计算机，除此之外还需要连接 Debugger 和 Demo 开发板，最后需要 Demo kit 和 E-ink 电子纸的正确连接。（如图）

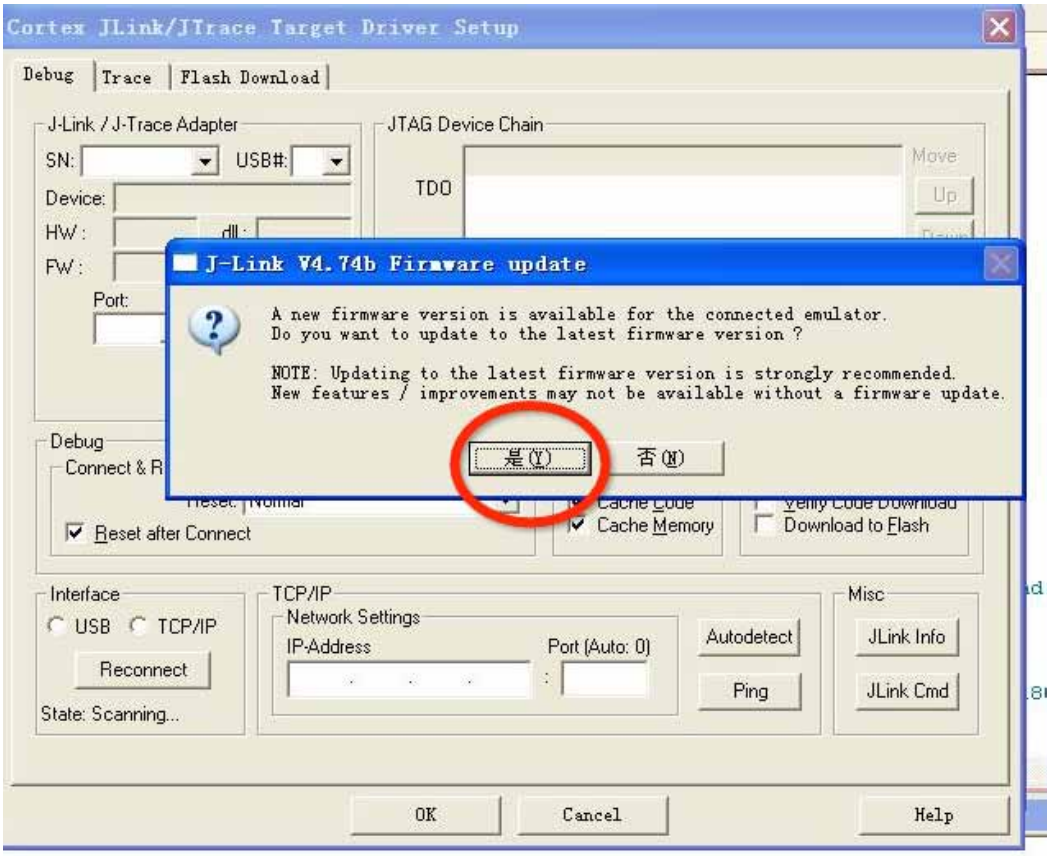
设置程序：

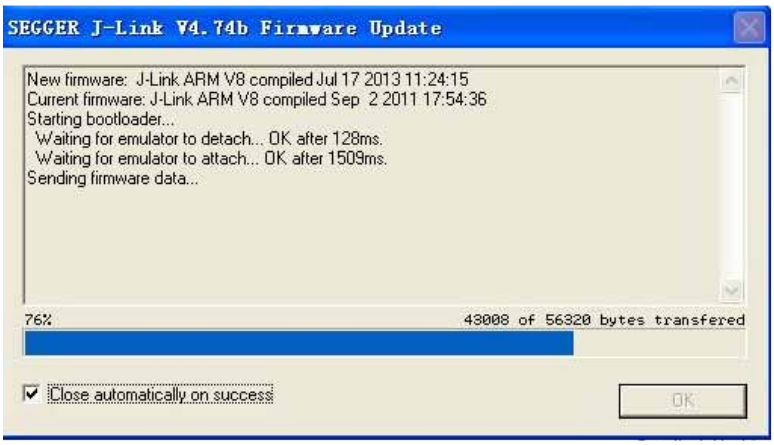
打开我们的 Kiel 软件后

1. 打开工程>光标右击左侧工程>选择 Options for Target >进入 Debug 选项卡，找到 JTAG debugger，选择 settings。


File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
mdk
Right click
misc.c
stm32f10x_iwdg.c
core_cm3.c
mdk
FwLib
stm32f10x_gpio.c
stm32f10x_rcc.c
stm32f10x_flash.
stm32f10x_spi.c
stm32f10x_usart.
stm32f10x_tim.c
misc.c
stm32f10x_iwdg.c
StartUp
system_stm32f10x
core_cm3.c
stm32f10x_conf.h
startup_stm32f10
User
stm32f10x_it.c
main.c
Options for Target 'mdk'... Alt+F7
Click!
Open File
Open List File
Open .\Debug\list\mdk.Map
Rebuild all target files
Build target F7
Translate File
Stop build
Add Group...
Add Files to Group...
Remove Item
Manage Components...
Show Include File Dependencies
2227 SPI4W_WRITEDATA(0x00);//RAM y add
2228 // SPI4W_WRITEDATA(0xb3);


Options for Target 'mdk'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Use Simulator
Limit Speed to Real-Time
Settings
Use:
Cortex-M/R J-LINK/J-Trace
ULINK Cortex Debugger
RDI Interface Driver
Altera Blaster Cortex Debugger
Stellaris ICDI
Signum Systems JTAGjet
Cortex-M/R J-LINK/J-Trace
ST-Link Debugger
ULINK Pro Cortex Debugger
NULink Debugger
Load Application at Startup
Initialization File:
Edit...
Restore Debug Session Settings
Breakpoints
Toolbox
Watch Windows & Performance Analyzer
Memory Display
Watch Windows
CPU DLL:
Parameter:
SARMCM3.DLL
Driver DLL:
DARMSTM.DLL
-pSTM32F103RB
TARMSTM.DLL
Dialog DLL:
OK
Cancel
Defaults
Help


Cortex JLink/JTrace Target Driver Setup
Debug
Trace
Flash Download
J-Link / J-Trace Adapter
SN:
USB#:
Device:
HW:
FW:
Port:
JTAG Device Chain
TDO
Move
Up
J-Link V4.74b Firmware update
A new firmware version is available for the connected emulator.
Do you want to update to the latest firmware version ?
NOTE: Updating to the latest firmware version is strongly recommended.
New features / improvements may not be available without a firmware update.
是(Y)
否(N)
Debug
Reset after Connect
Cache Code
Cache Memory
Download to Flash
Interface
USB
TCP/IP
Reconnect
State: Scanning...
Network Settings
IP-Address
Port (Auto: 0)
Autodetect
Ping
Misc
JLink Info
JLink Cmd
OK
Cancel
Help

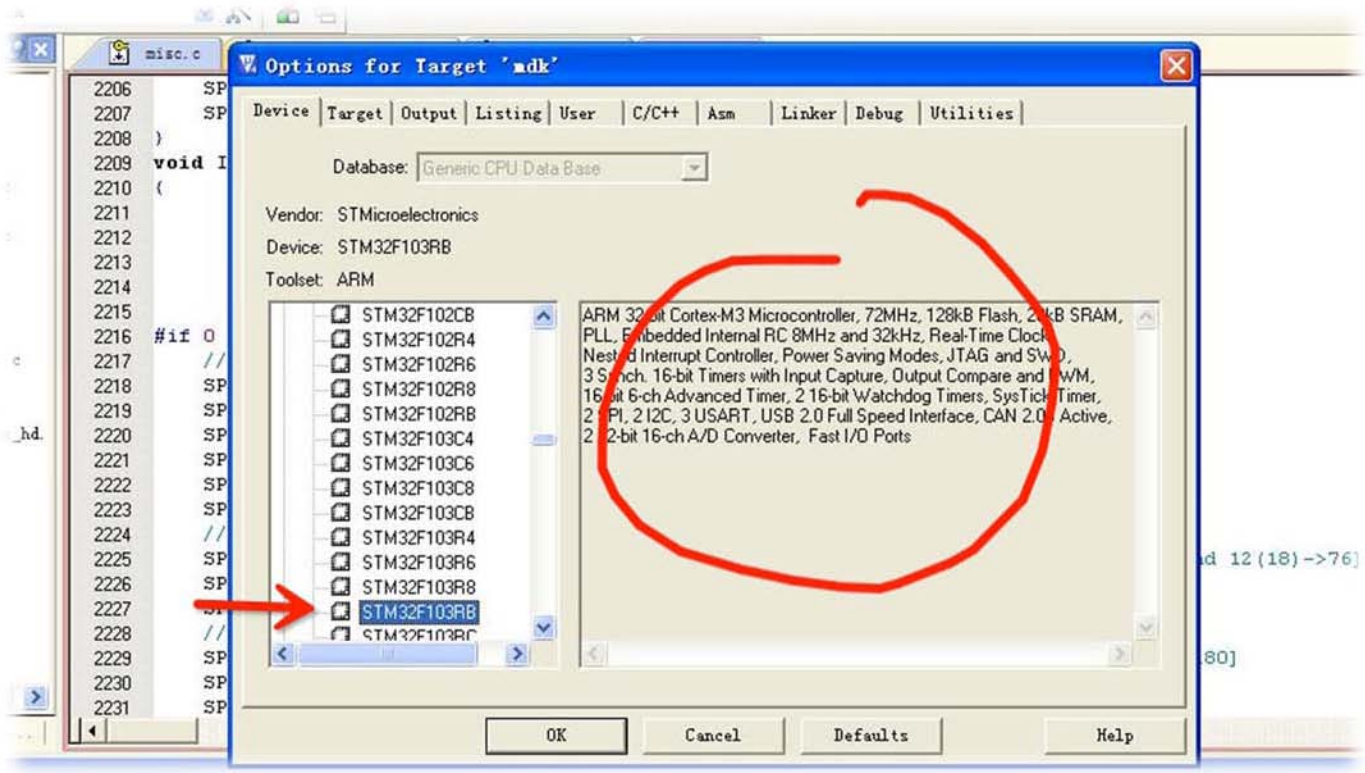


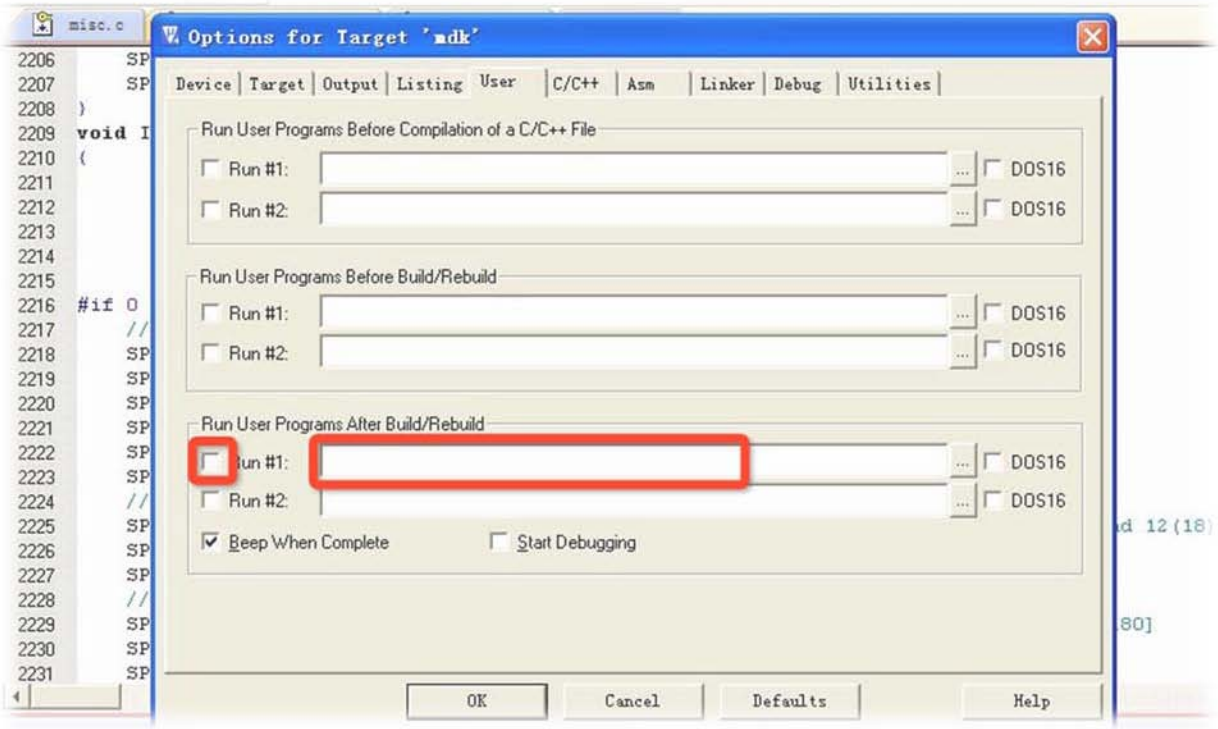
上传完成后需要对内容进行确定保存。

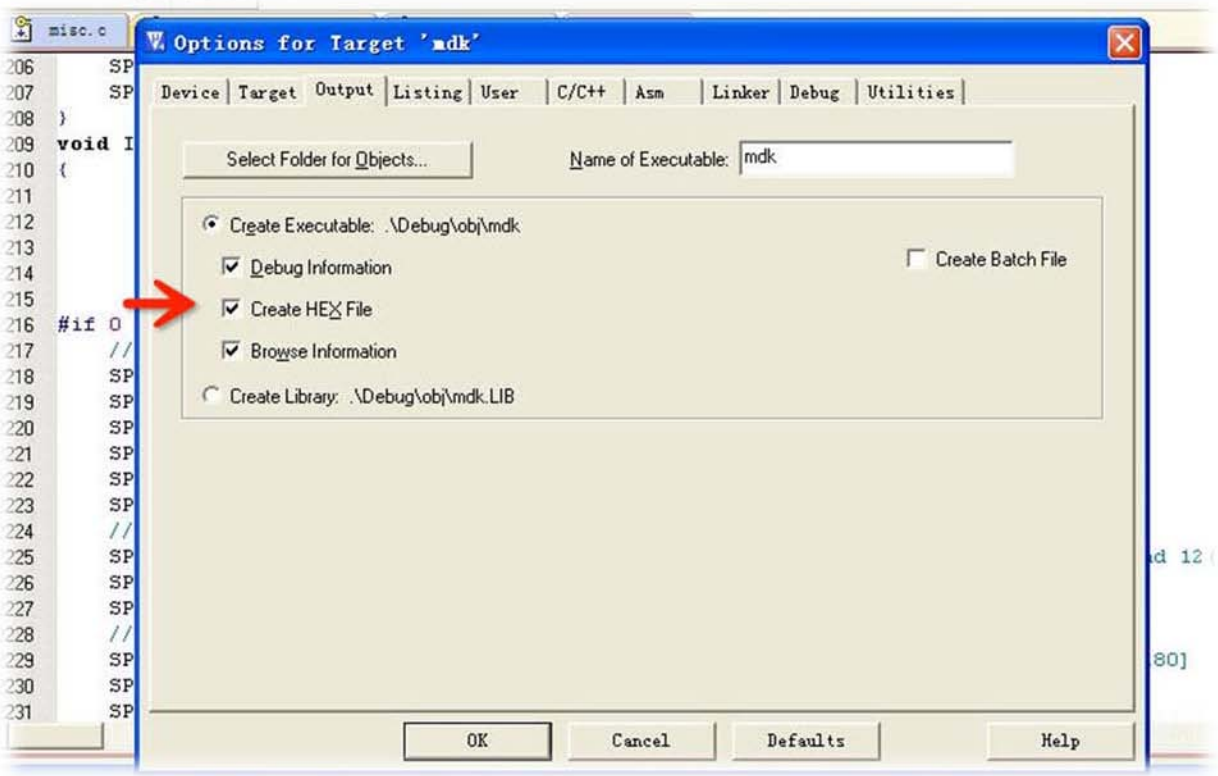
2. 然后在 Utilities 选项卡中同样选择 JTAG 进行类似设定，进入 settings 选项。（此步骤同上）

注意：完成后请检查 Debug 和 Utilities 的 FlashDownload 选项卡如果没有正确添加芯片请手动添加 STM32F10x 128K。

3. Device 选项卡中找出我们所用的单片机规格，完毕后将 User 选项卡中 Run user pro 去掉勾选并在 Output 选项卡中勾选上产生 Hex 十六进制文件。

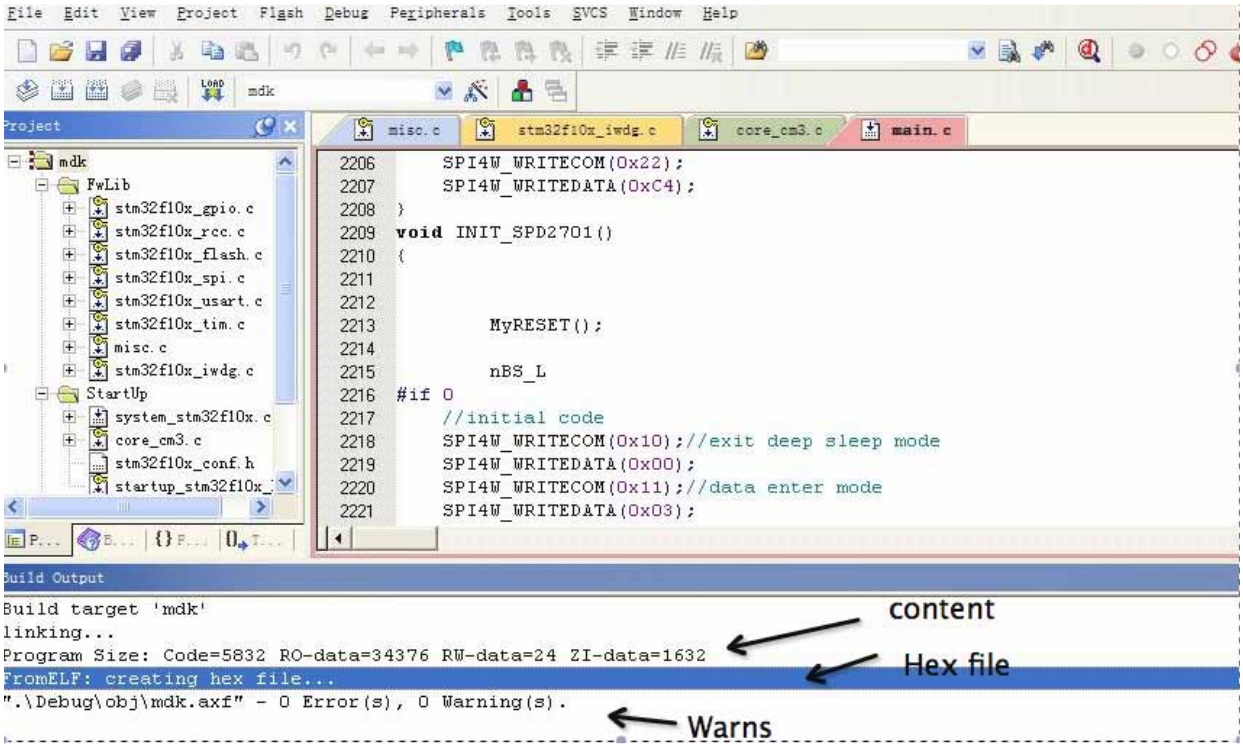



misc.c
Options for Target 'mdk'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Run User Programs Before Compilation of a C/C++ File
Run #1:
Run #2:
DOS16
Run User Programs Before Build/Rebuild
Run User Programs After Build/Rebuild
Beep When Complete
Start Debugging
OK
Cancel
Defaults
Help


misc.c
Options for Target 'mdk'
Device
Target
Output
Listing
User
C/C++
Asm
Linker
Debug
Utilities
Select Folder for Objects...
Name of Executable:
mdk
Create Executable: .\Debug\obj\mdk
Debug Information
Create HEX File
Browse Information
Create Library: .\Debug\obj\mdk.LIB
Create Batch File
OK
Cancel
Defaults
Help

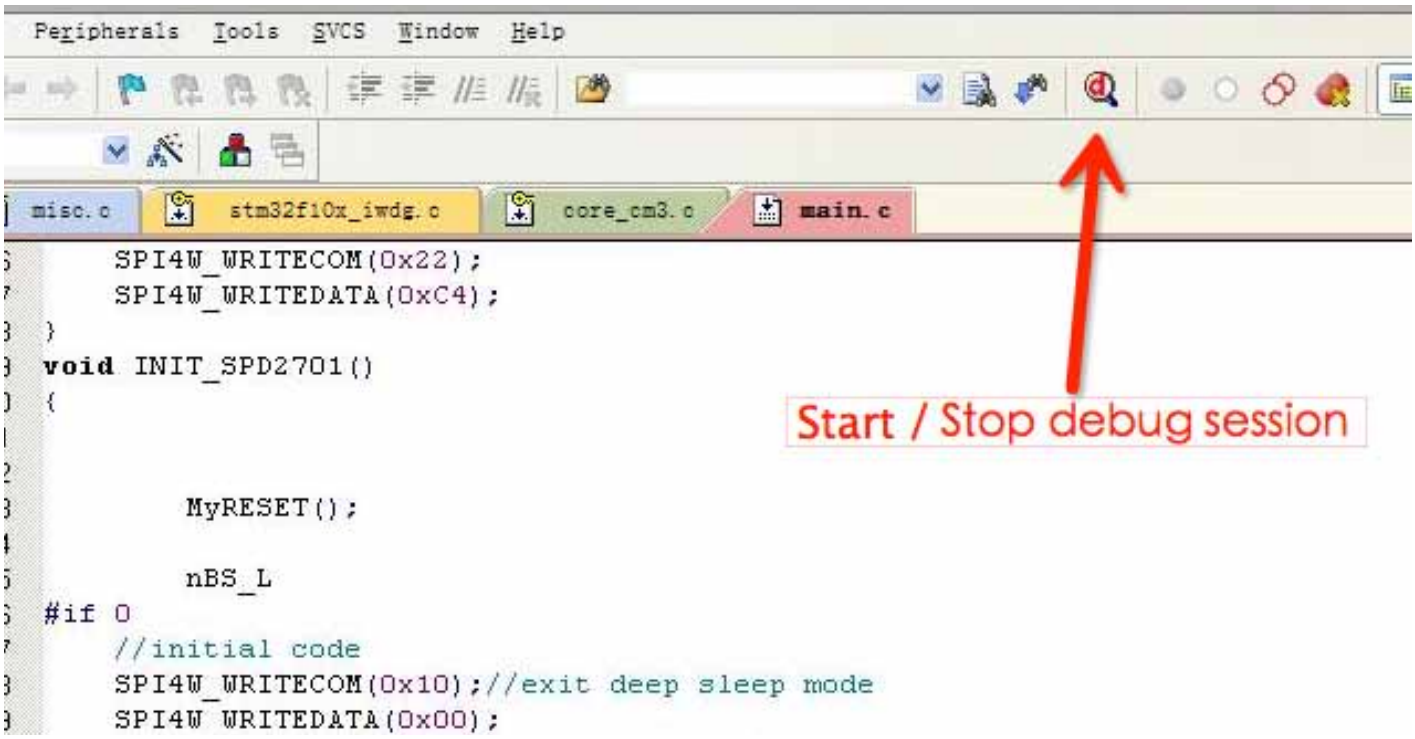
编译过程：

编译你的程序可以在左上角点击 build 按键，正式编译后会看到窗口下方的 Build Output 窗口内显示成功编译字样，如果存在错误，也会提示错误所在的命令行及函数体，且 Hex 文件不会被成功创建。



下载：

请再一次确认 Debugger 和 Demo kit 正确连接且都已上电，工作指示灯闪烁或常亮。点击 Start/Stop Debugger session 按钮后自动将已编译好的文件下载到 Arm 单片机中去。



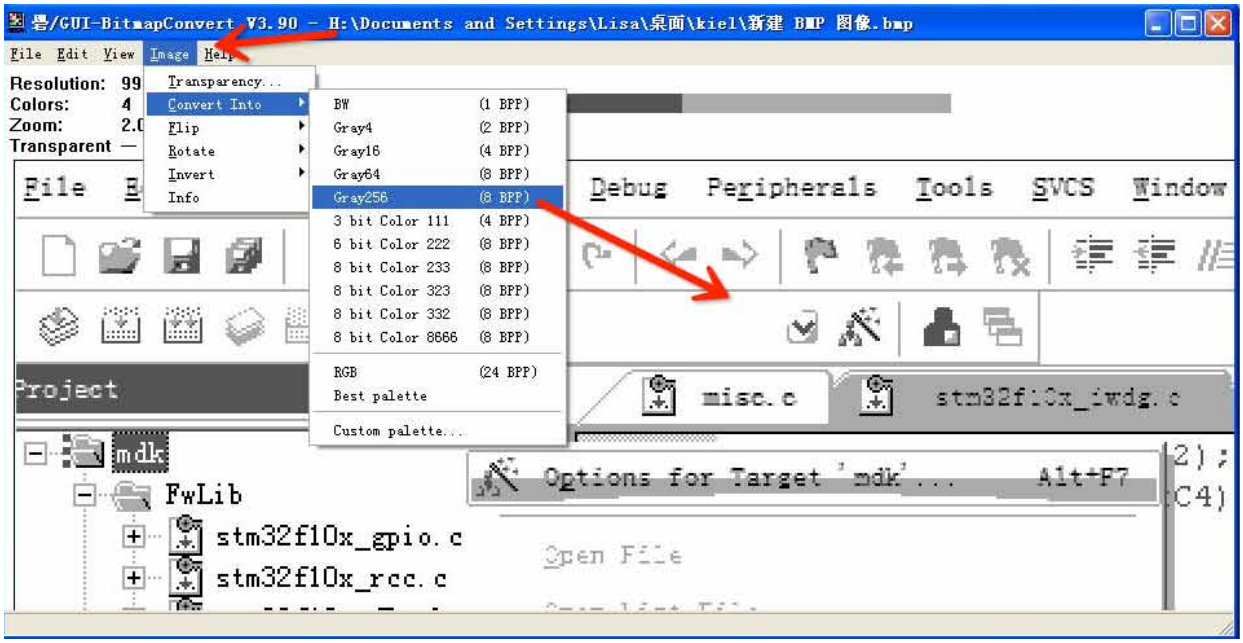
下载只可能在编译后执行。

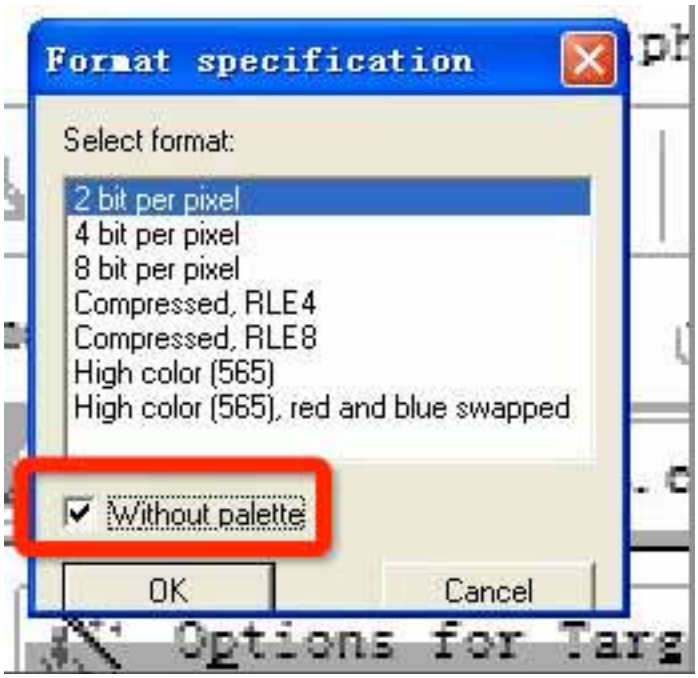
调试程序：

可以使用双击命令行前端设置一个节点，用 step 功能检查工作情况，建议每次调试后都编译后下载再使用，这是为了保证单片机中的程序和工程同步。

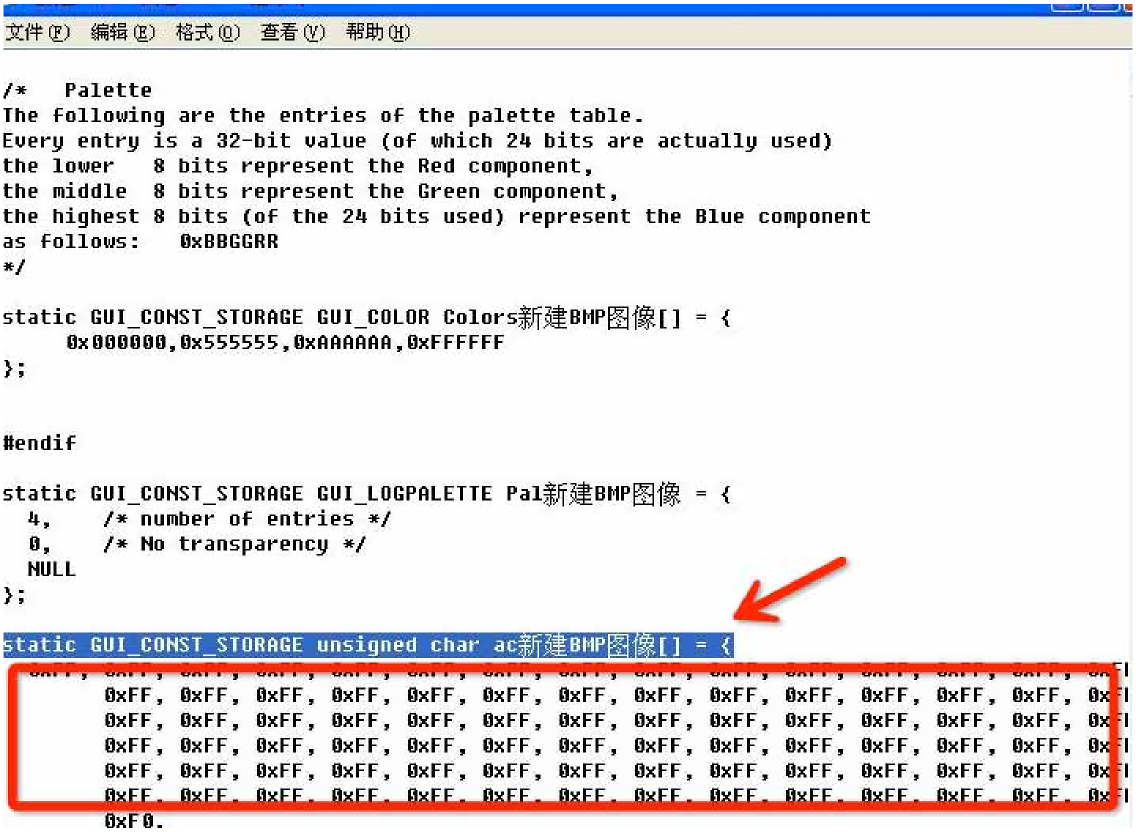
生成一个图片：

借助任意一款网络上的图片转码软件都可以，这里建议使用我们的小软件 BitmapCovert，只要将.bmp 格式的绘图加载到软件中进行格式化处理，再将图片转存.c 文件，记住勾选 without template 选项可以确保产生的.c 文件内没有重复的混淆使用的数据。将图片部分产生的十六进制数组全部拷贝到工程内就可以正确的替换一张内置图片，编译下载完成。





这里的选项请注意是 per pixel「每像素比特数」，所以请了解产品可支持的灰阶等级后进行转换。



此段数组就是转换成的代码，需要将程序部份替换。