

一、驱动篇

手持机驱动安装

根据需要安装对应的驱动，通常需要安装手持机的通讯驱动，该文件位于“Z:\开发光盘\3、软件\B_通信软件驱动\客户通信包(含驱动).rar”

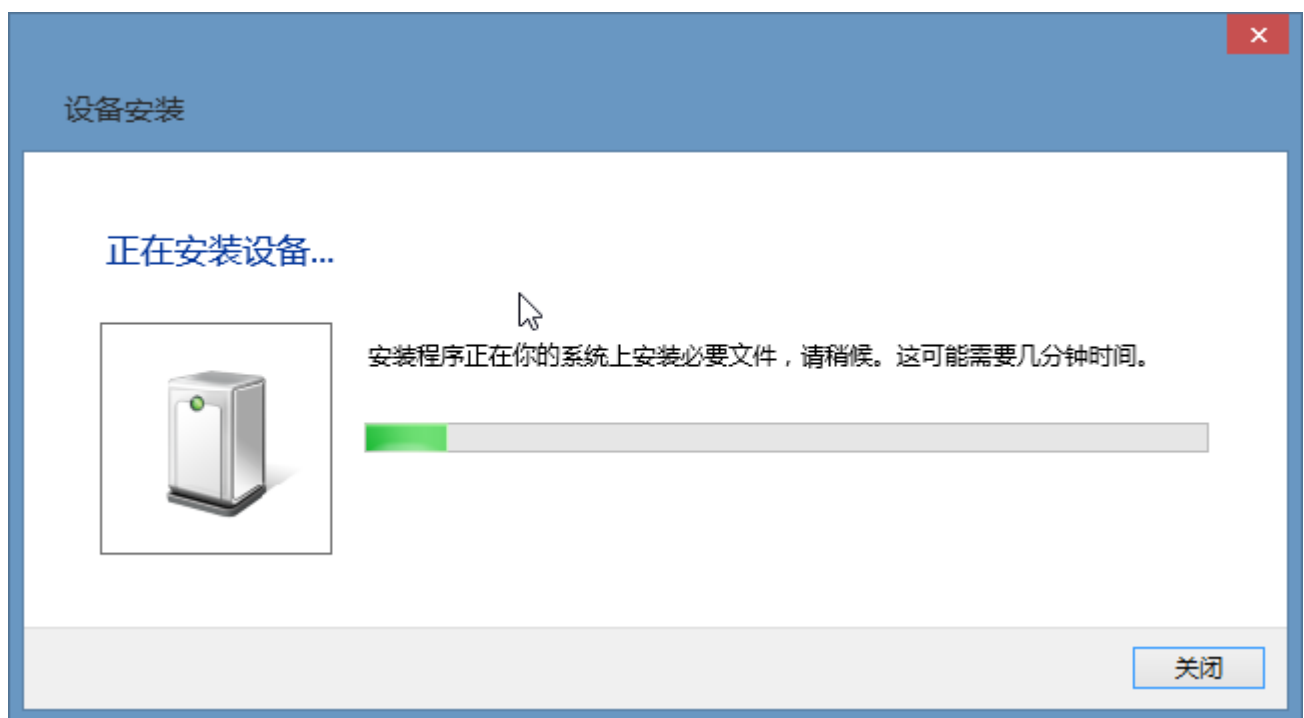
下面以 WINDOWS 8.1 为例安装驱动，其他系统版本雷同，如果有不明白的，可以网上看相关驱动安装资料。

一、准备好一下资料

- 1、手持机
- 2、驱动文件
- 3、USB 通信线
- 4、装有微软系统的电脑

二、连接硬件

- a) 将电池装入手持机中
 - b) 将 USB 的梯形头接入手持机，另一端接入电脑
 - c) 点击手持机中**连接电脑**的图标，让手持机进入通信状态
- 电脑系统可能出现以下提示：点击关闭即可。

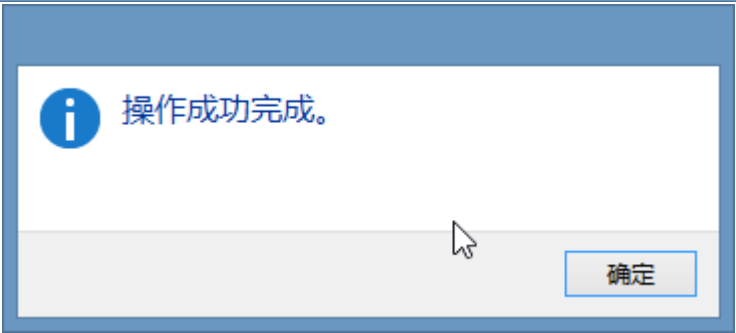
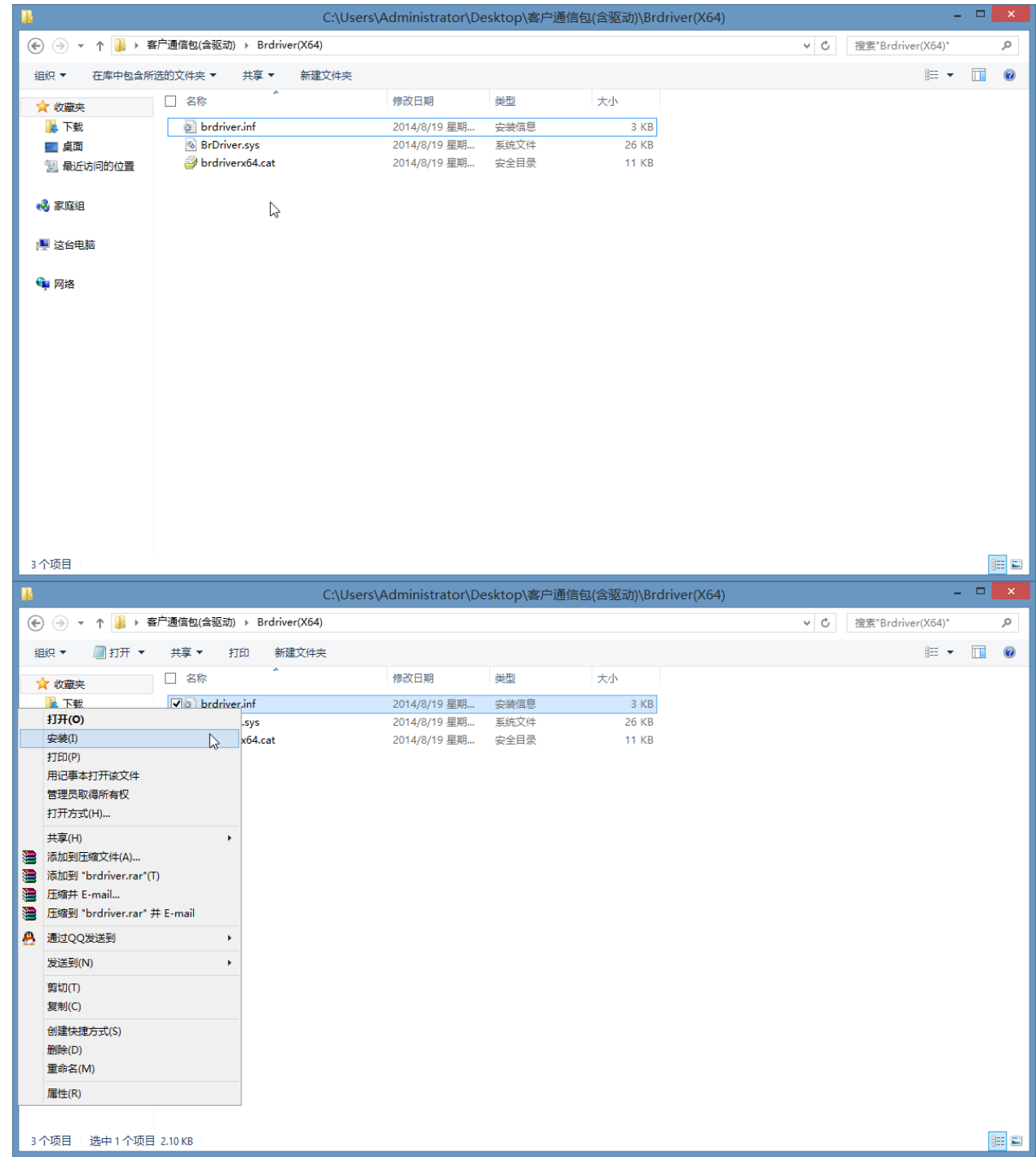


三、电脑端操作

方法一：

解压驱动文件，根据自己的系统选择对应的驱动，32 位的系统选择 BrDriver(X86)文件夹下面的驱动，64 位系统选择 Brdriver(X64)下的驱动。

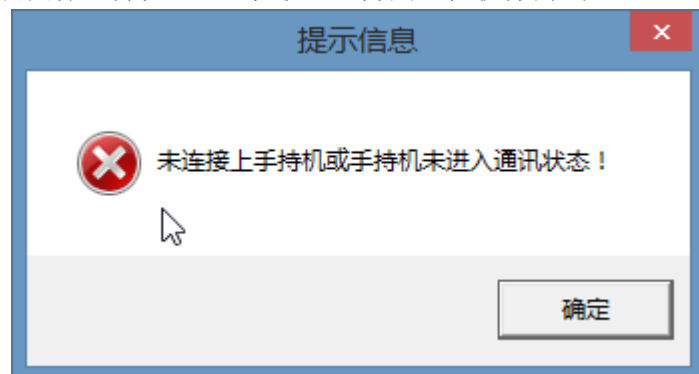
在 brdriver.inf 文件上单击**右键**，在弹出的菜单中选择安装。系统提示安装成功。



打开通信软件 CLComm.exe 进行验证驱动是否安装好。



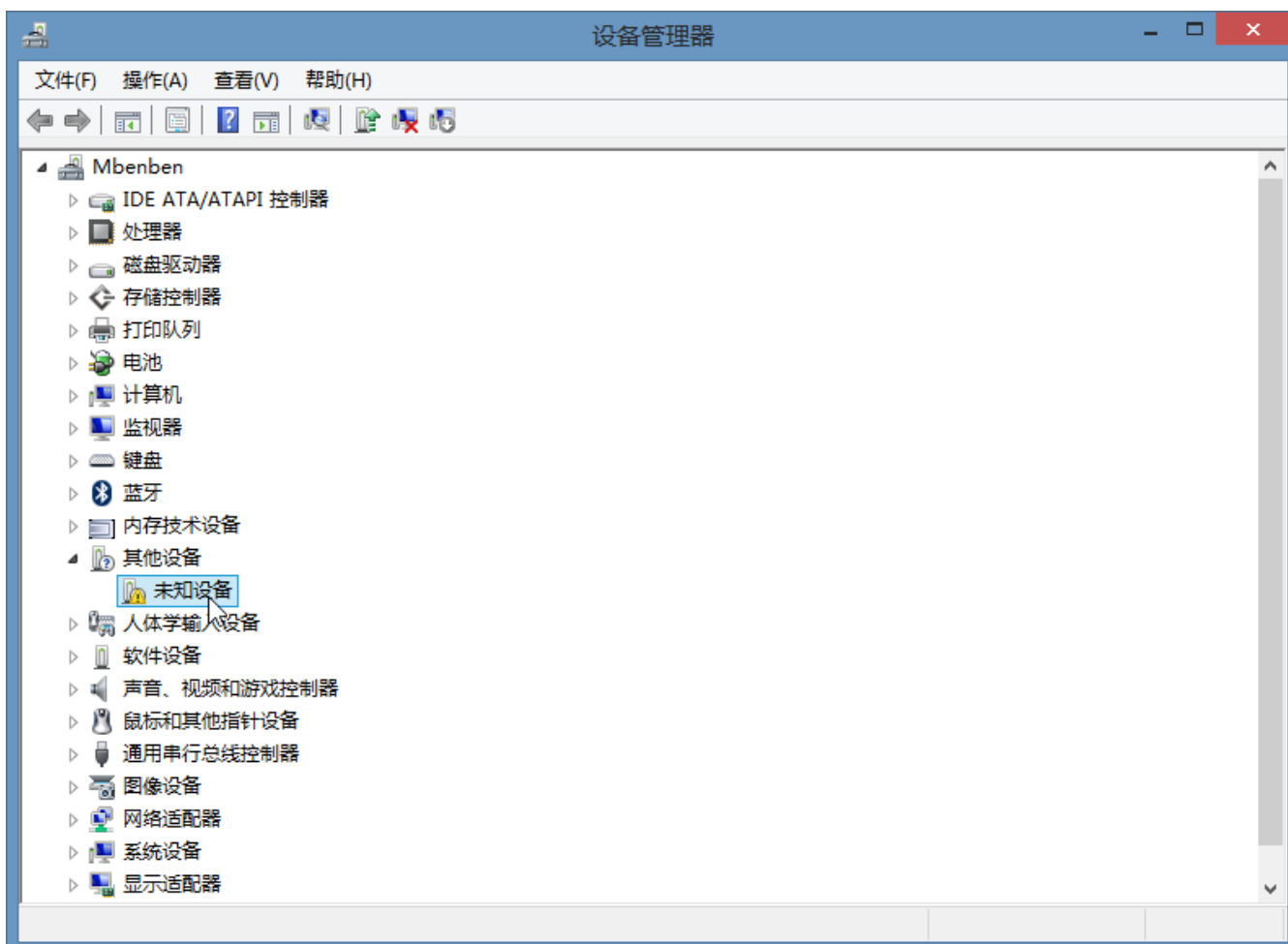
如果弹出以下菜单，说明驱动安装不成功。如果不成功，可以退出通信软件，然后点击手持机退出通信状态，再点击手持机的连接电脑图标，再次进入通信状态，并用通信软件测试。

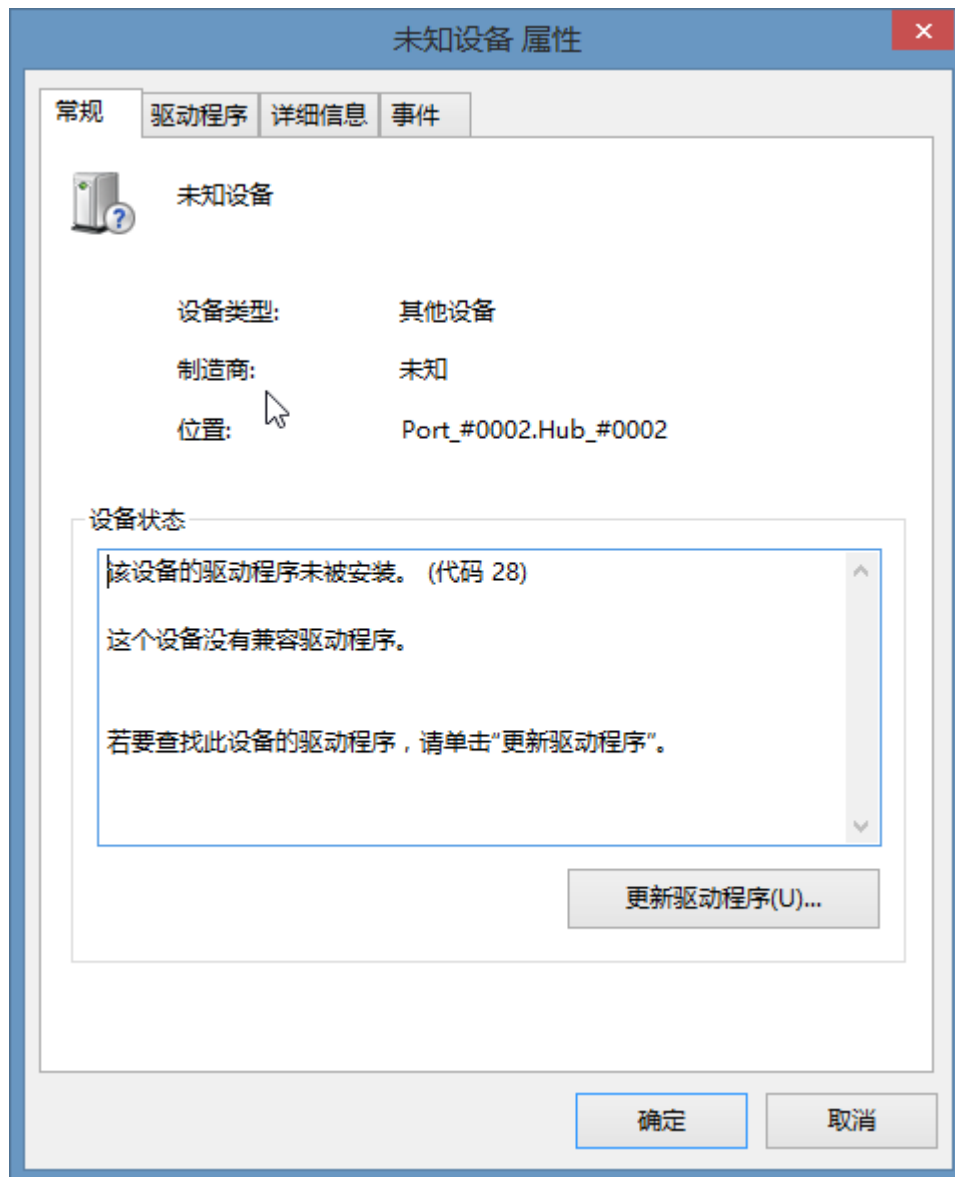


一般不成功，可能是您的电脑安装了 360 软件，驱动被拦截，没有添加到信任列表，查看信任列表，并添加到信任，或者退出 360 再次测试。

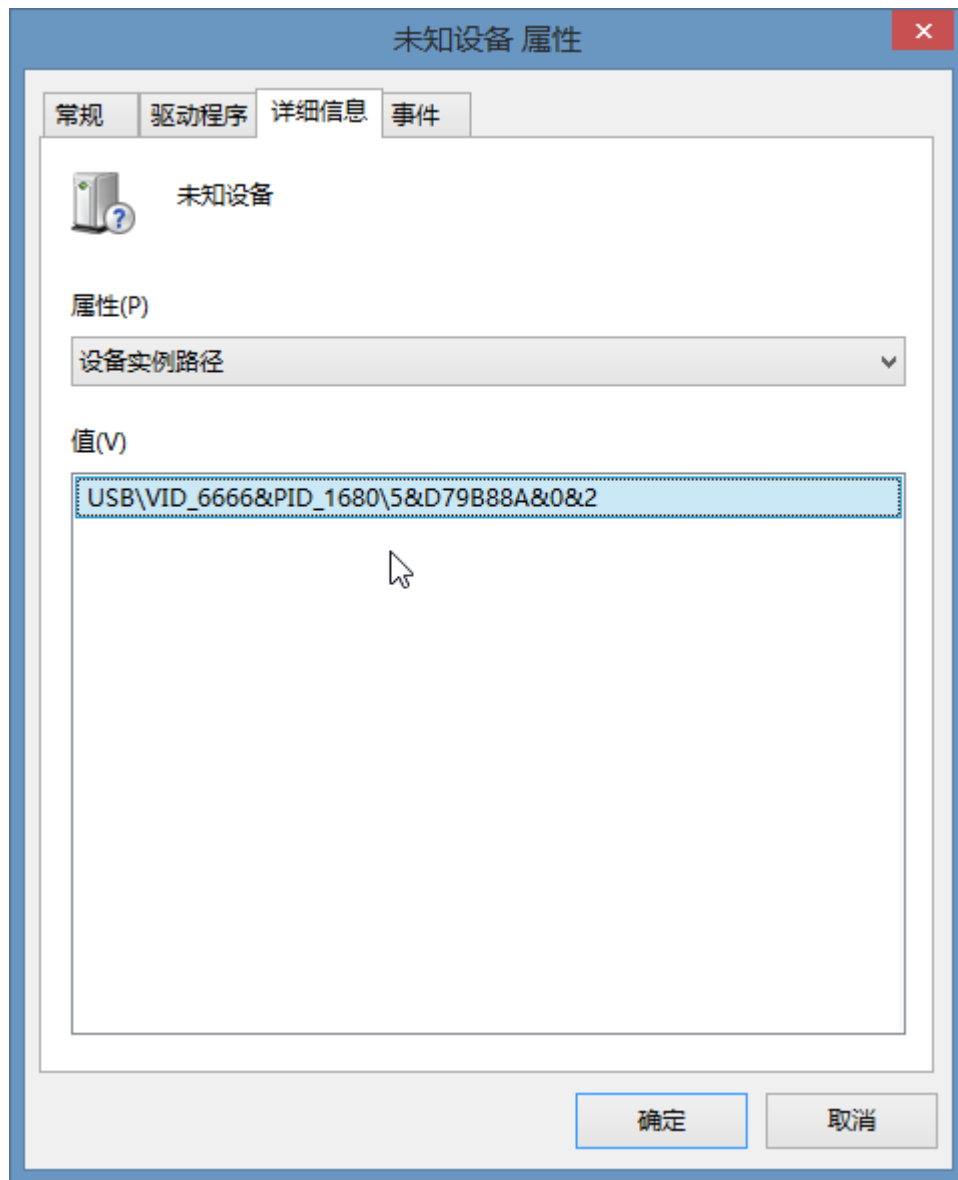
方法二：

手持机进入通信状态后，电脑端出现带有感叹号的未知设备。通过以下顺序可以查看，并手动进行驱动更新。

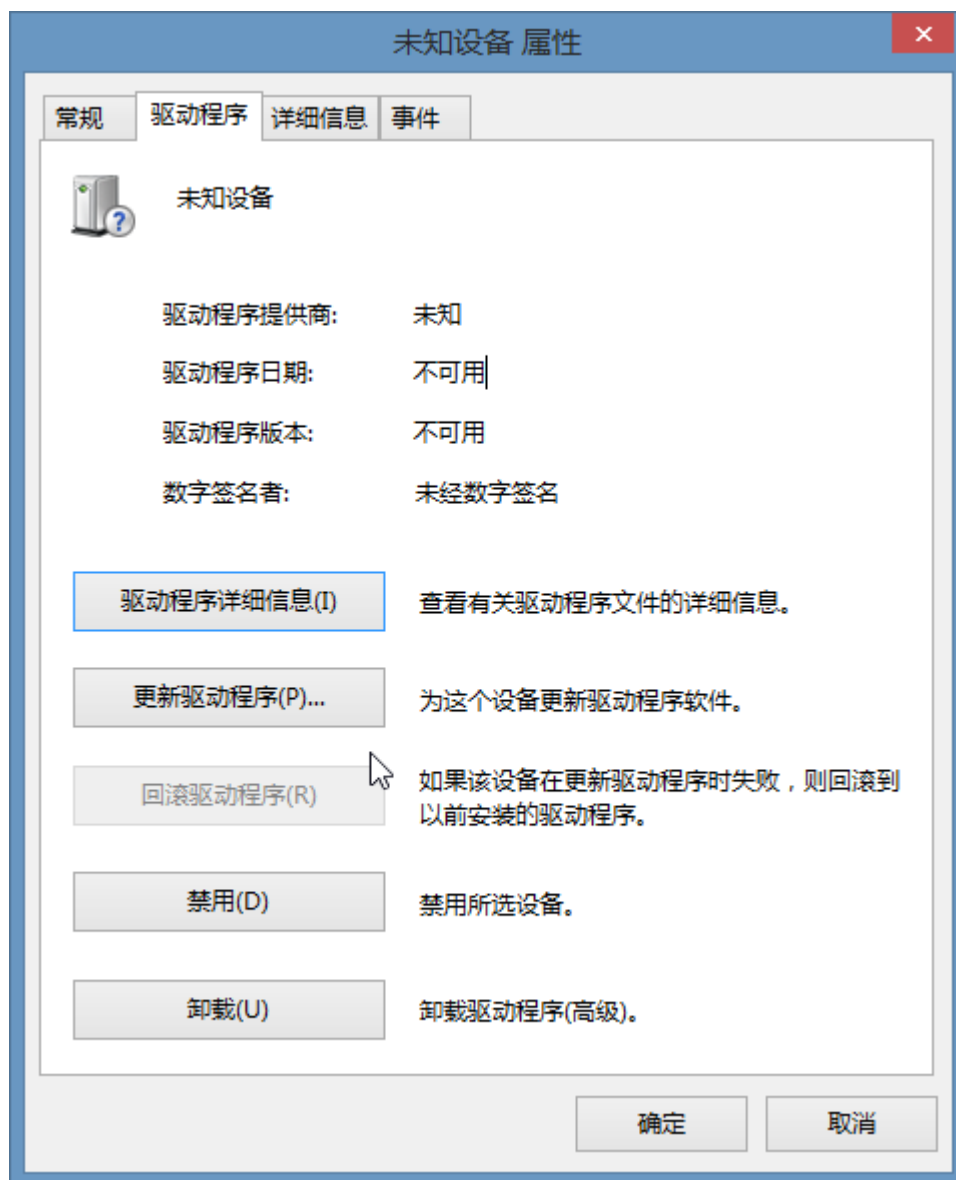




查看详细信息，确保该设备为本公司的手持机设备。



回到驱动程序页面，点击更新驱动程序按钮。





更新驱动程序软件 - 未知设备



你希望如何搜索驱动程序软件？



自动搜索更新的驱动程序软件(S)

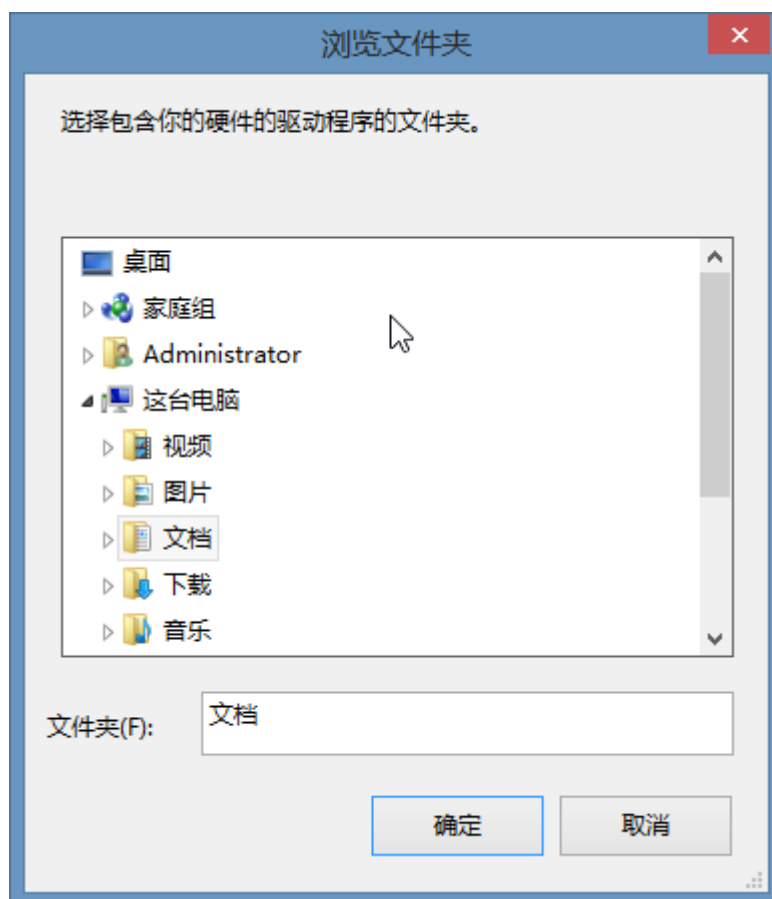
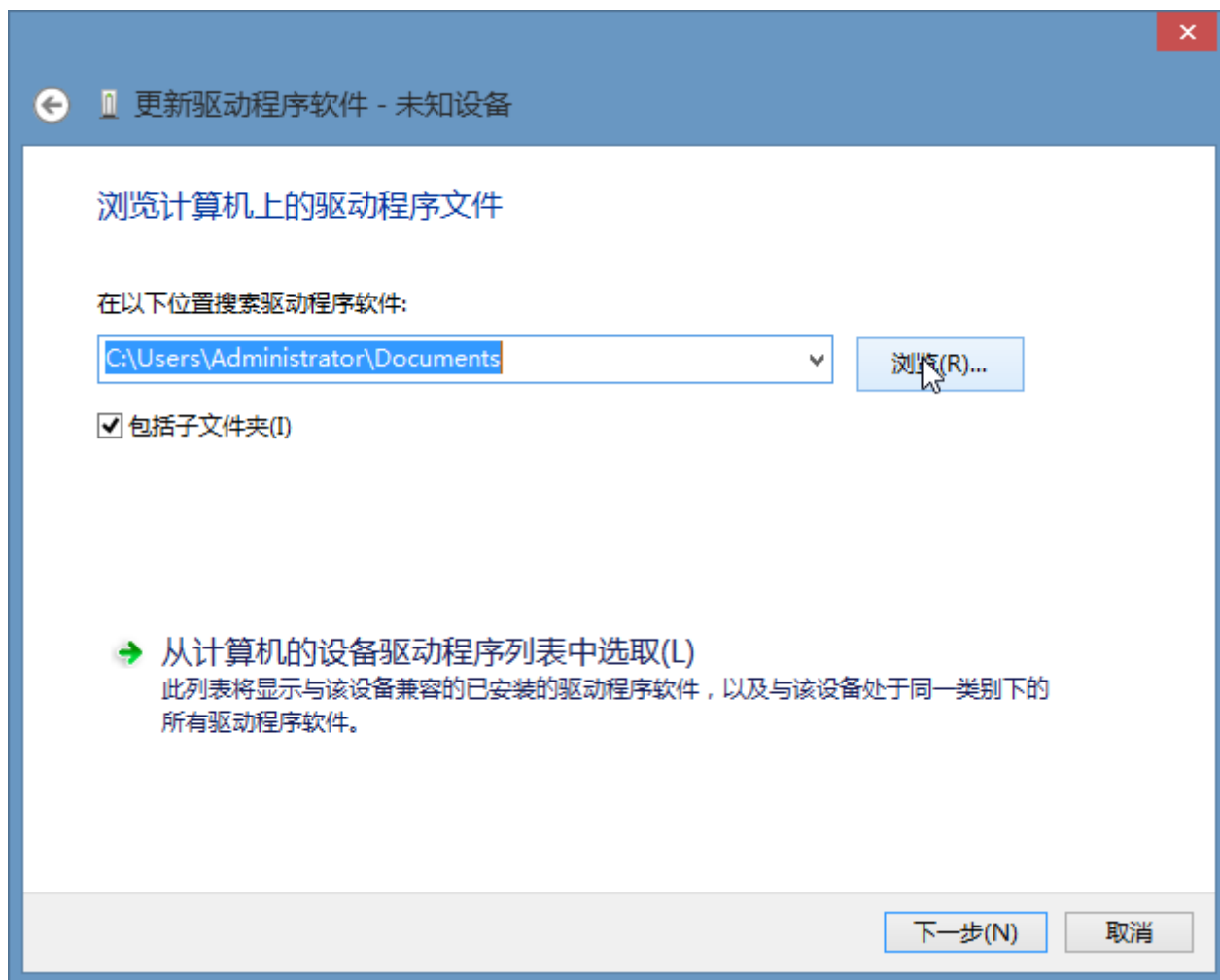
Windows 将在你的计算机和 Internet 上查找用于相关设备的最新驱动程序软件，除非在设备安装设备中禁用该功能。

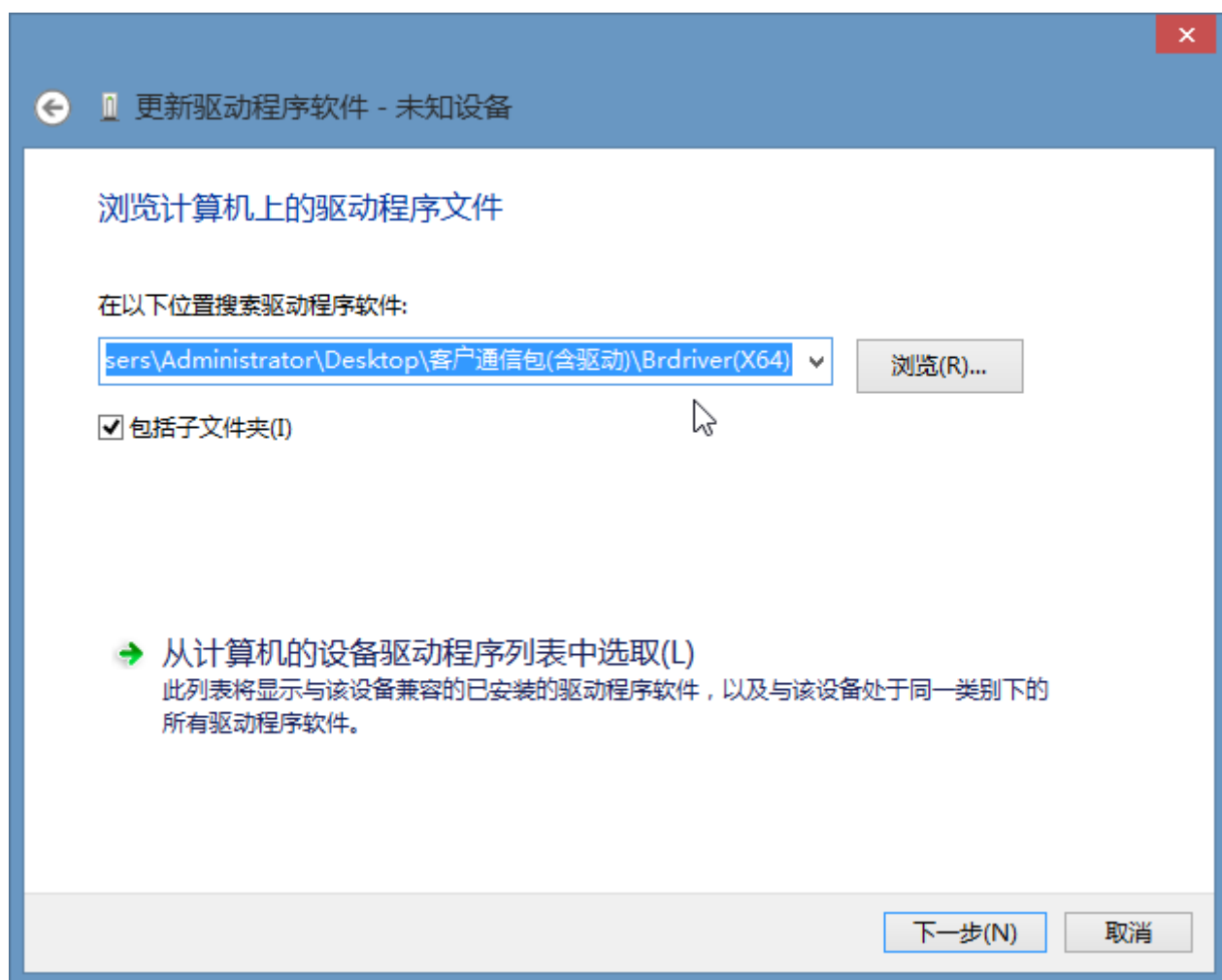
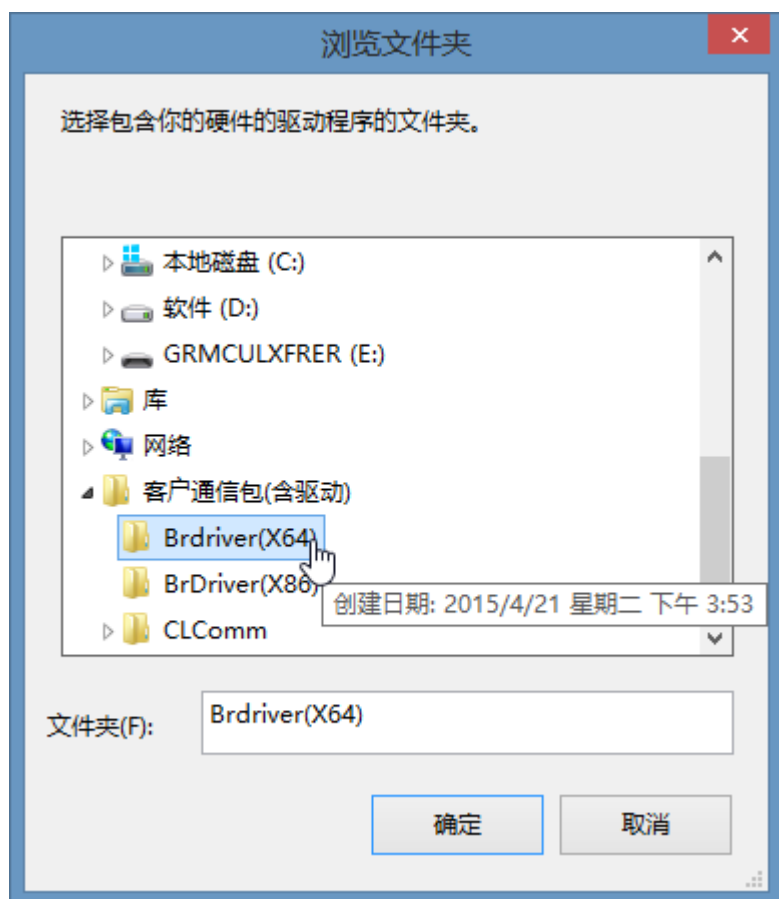


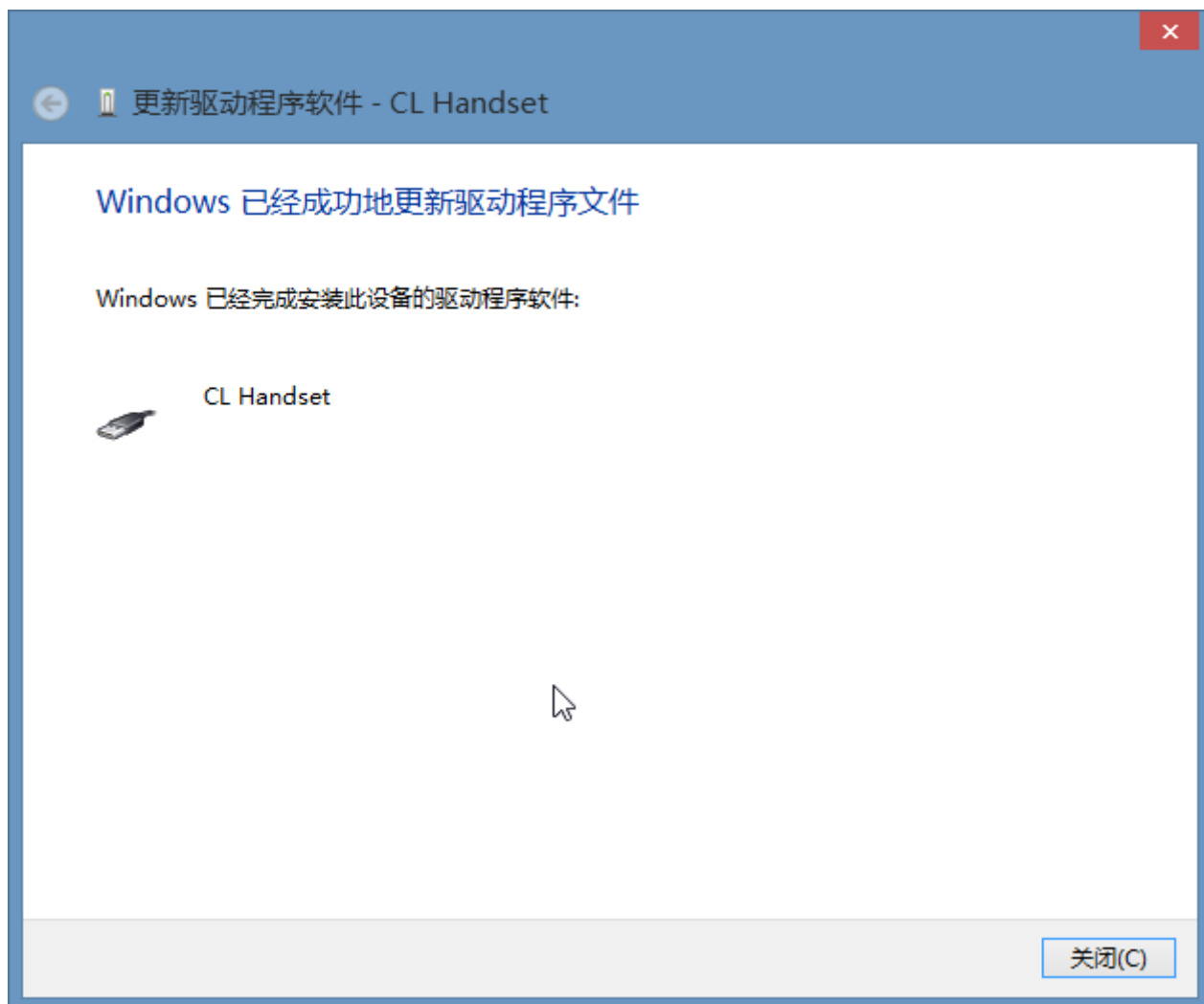
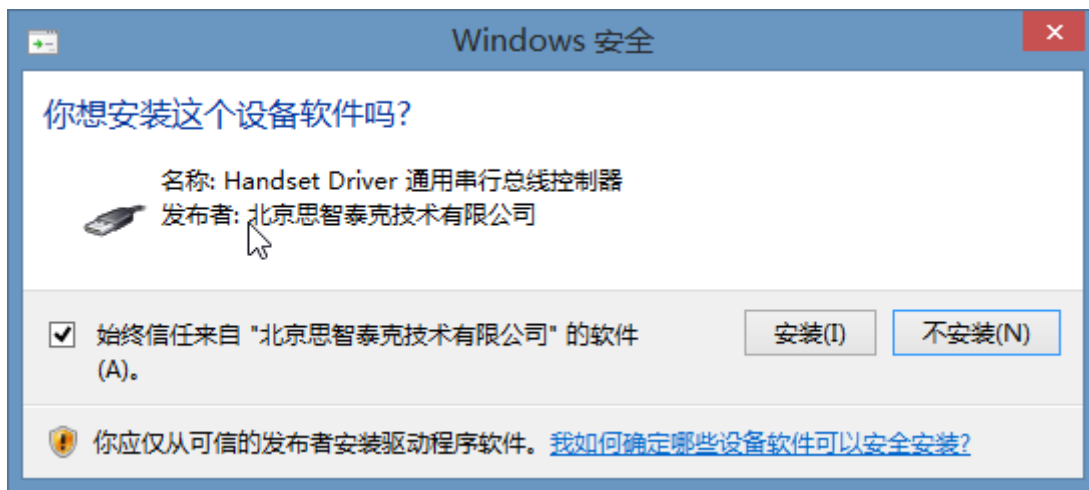
浏览计算机以查找驱动程序软件(R)

手动查找并安装驱动程序软件

取消







CL Handset 属性



常规 驱动程序 详细信息 事件



CL Handset

事件(E)

时间戳	描述
2015/4/21 星期二 ...	已配置设备 (null)
2015/4/21 星期二 ...	已配置设备 (oem15.inf)
2015/4/21 星期二 ...	已启动设备 (BrDriver)
2015/4/21 星期二 ...	已安装设备 (brdriver.inf)

信息(I)

已配置设备 USB\VID_6666&PID_1680\5&d79b88a&0&2。

驱动程序名称: null

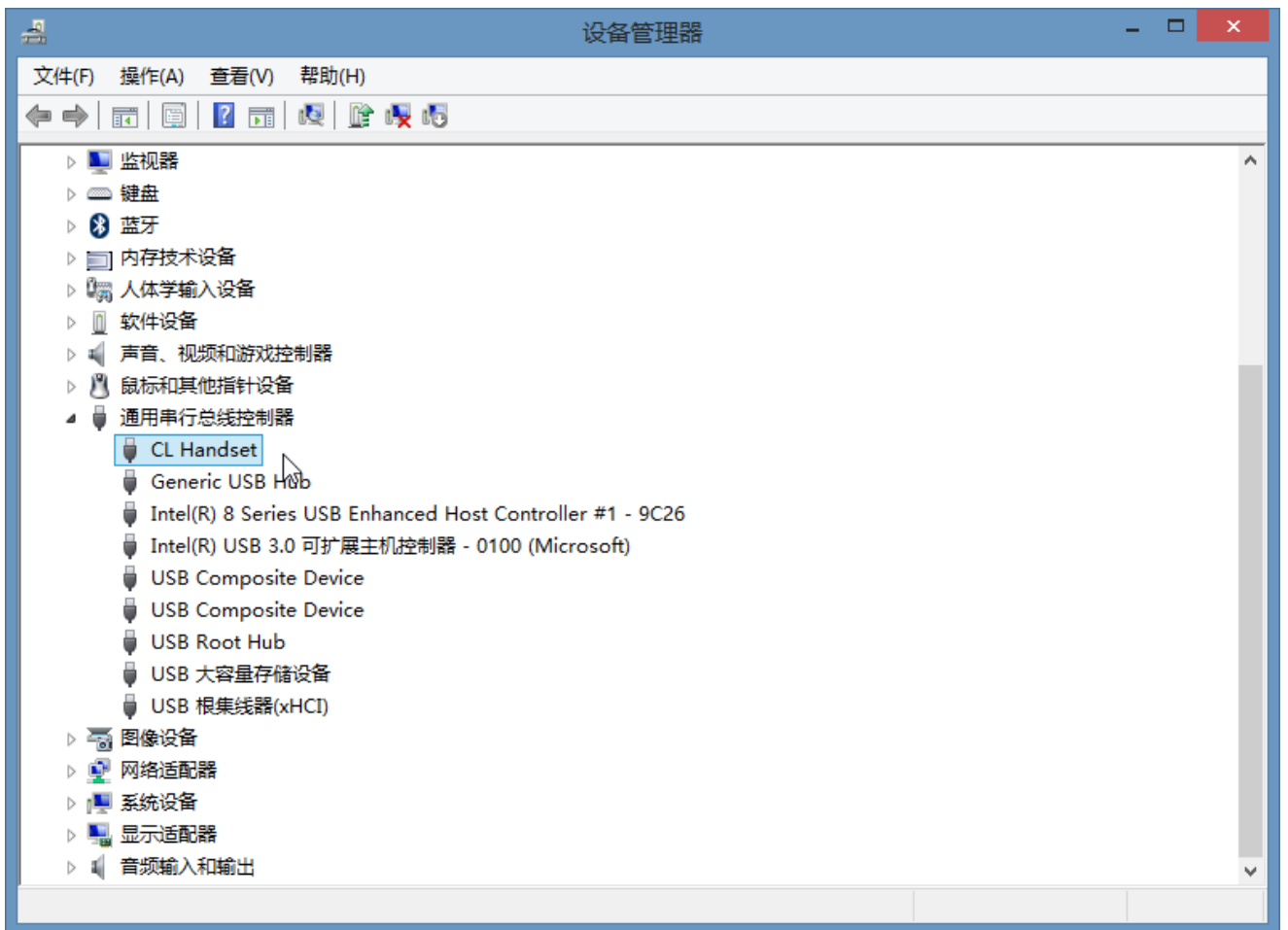
类 GUID: {00000000-0000-0000-0000-000000000000}

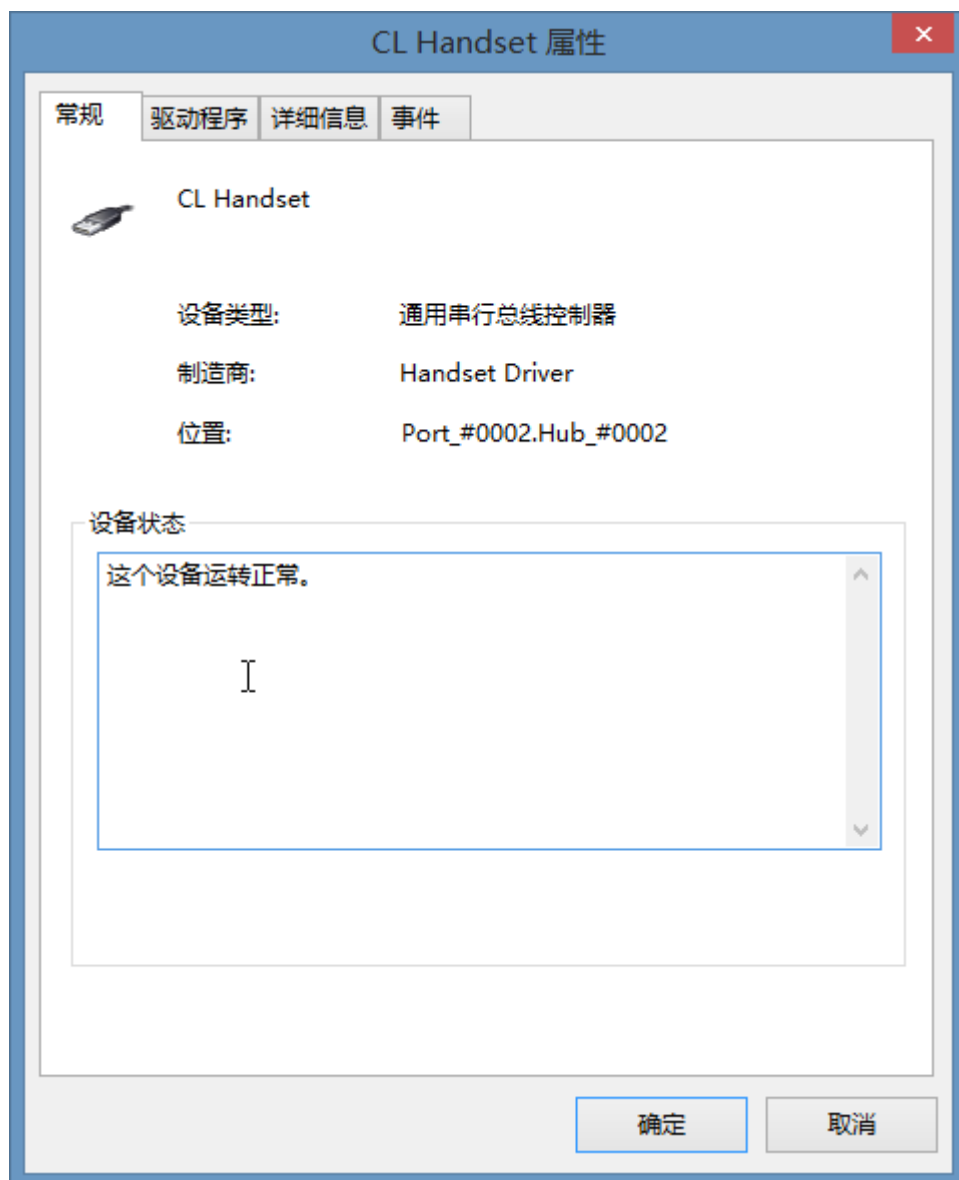
驱动程序日期:

[查看所有事件\(V\)...](#)

关闭

取消





二、系统篇

系统备份说明

手持机提供二次开发功能，包括软件和硬件的开发功能。因为提供仿真功能，在进行程序编写时，有会由于各种原因，如程序员不小心擦除了单片机内部的程序，出现该情况后，系统将无法通过正常升级方法进行系统还原，为了避免出现该情况，我们建议在进行系统编程时将系统备份出来，有备无患。

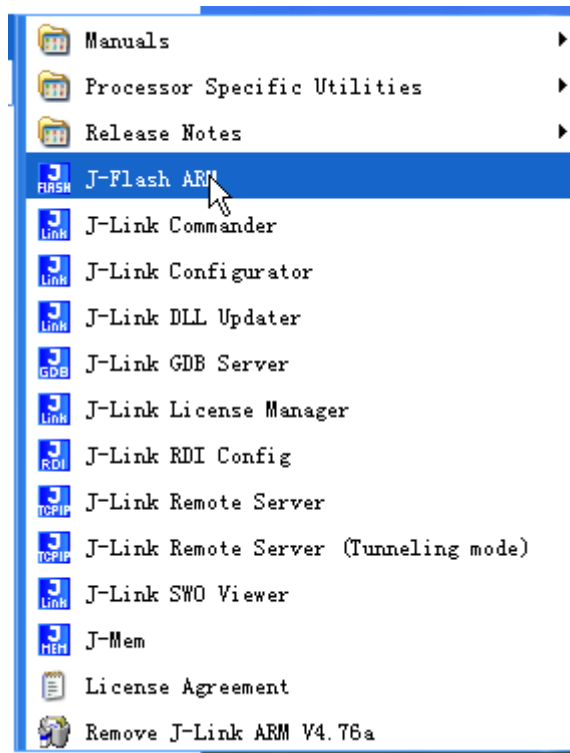
注意：

- 1、备份出来的系统，只能够用于该机器，而不能够应用到其他机器。
- 2、一般编程，请直接使用我们提供范本工程，这样将不会出现系统被破坏或者擦除。

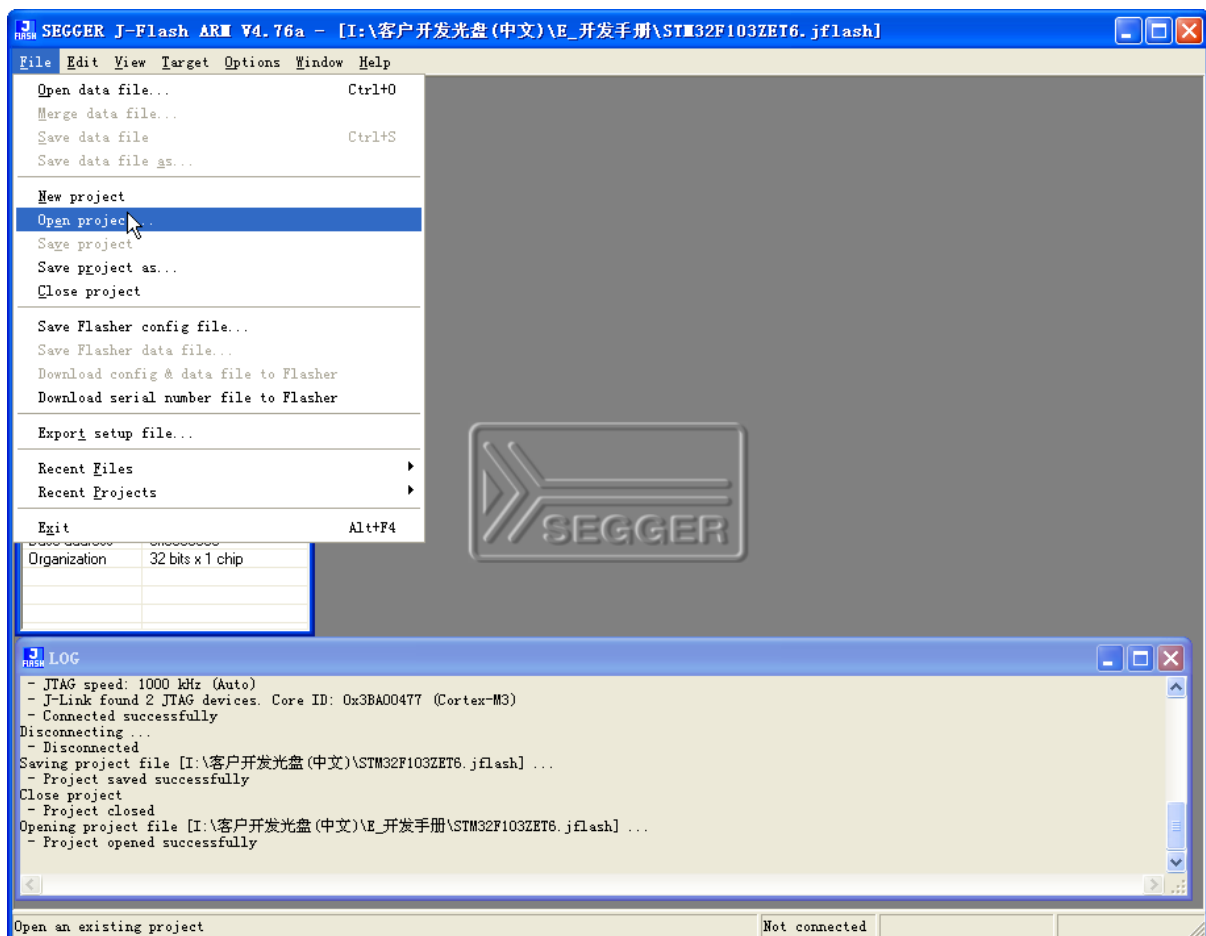
一、备份系统：

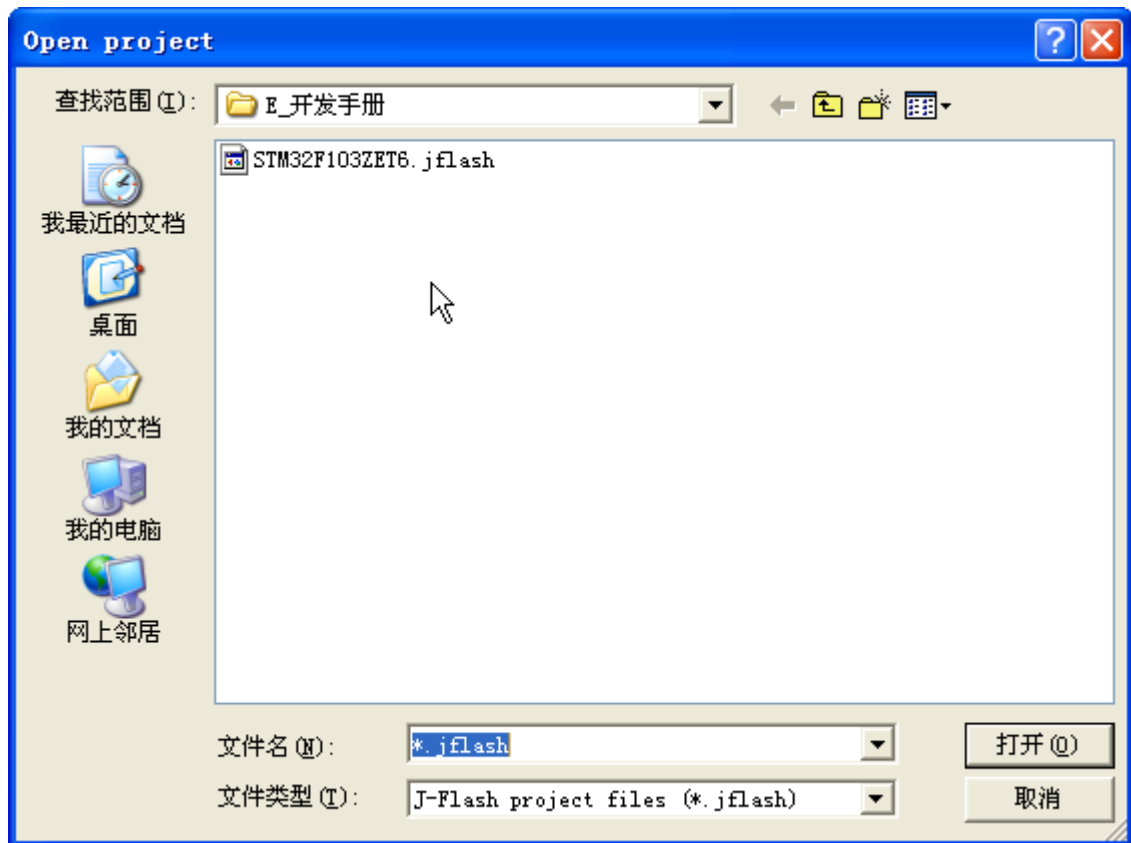
- 1、安装 Setup_JLink_V502l.exe，该软件在“Z:\开发光盘\3、软件\C_编译开发环境\Setup_JLink_V502l（仿真驱动）.zip”

2、运行程序：

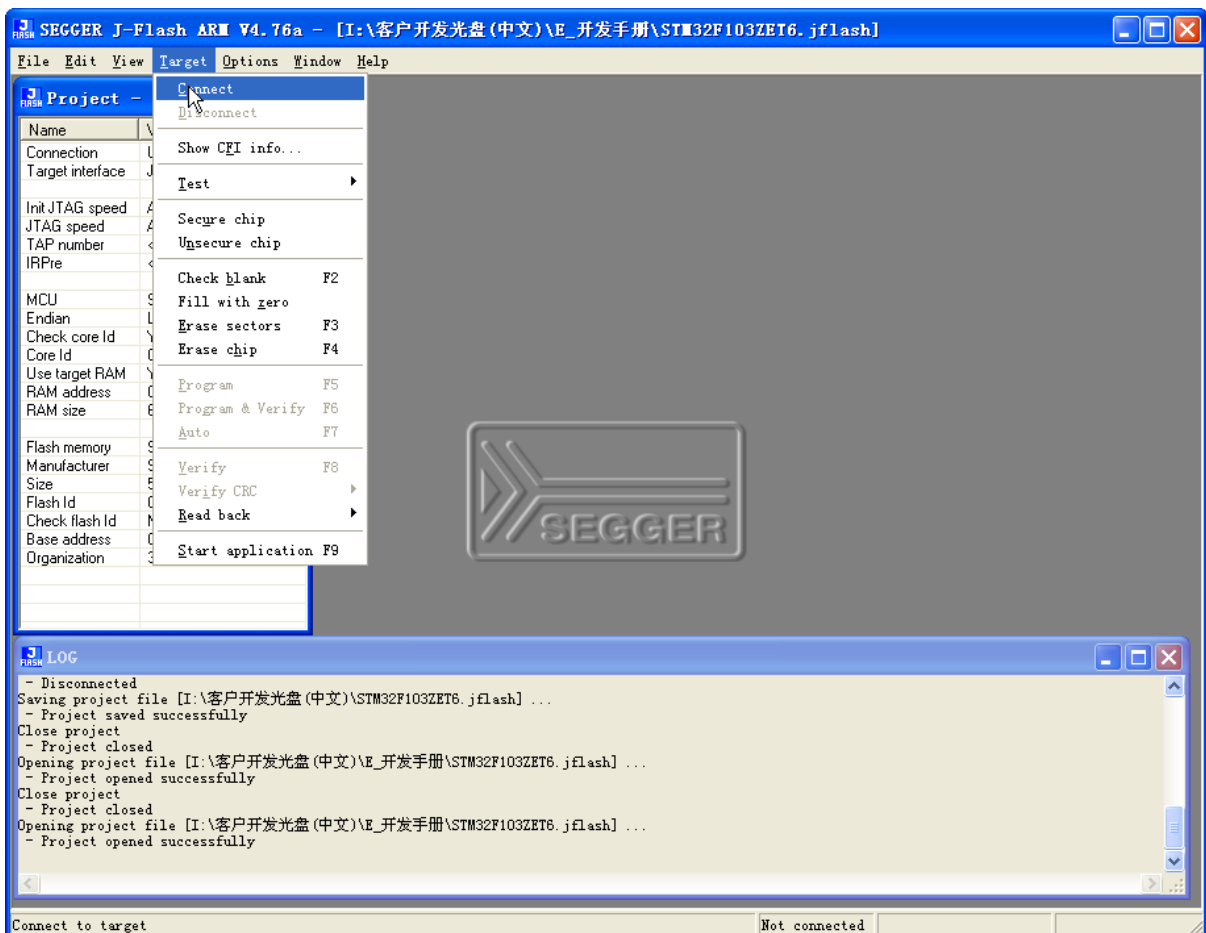


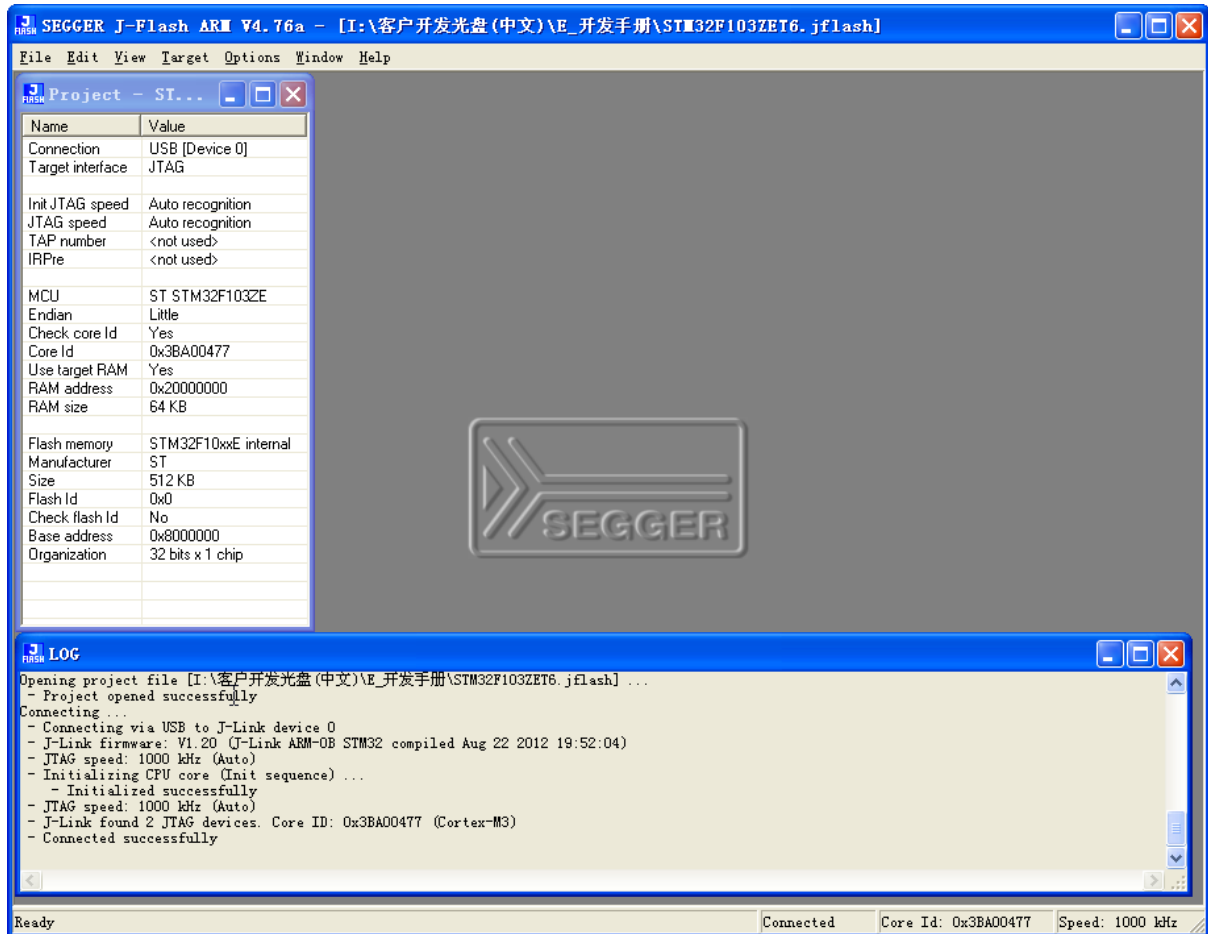
3、打开配置好的工程文件 STM32F103ZET6.jflash，该文件在“Z:\开发光盘\1、文档\B_开发文档手册”文件夹内



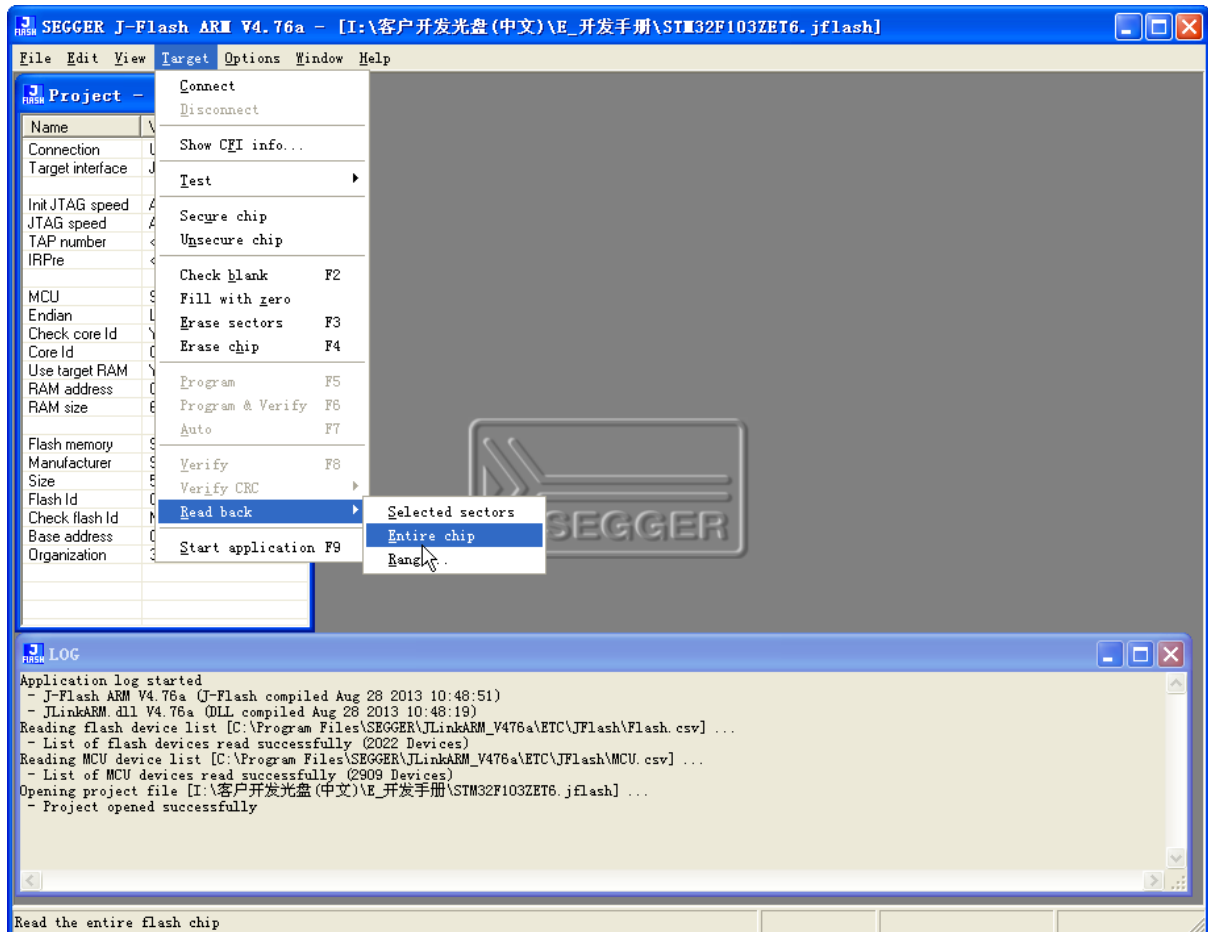


- 4、在进行下面的操作前，先使用 Jlink 连接好电脑和手持机，如需安装 Jlink 的驱动，请先安装好，该驱动在“Z:\开发光盘\3、软件\B_通信软件驱动\MiniJTAG 仿真器驱动.rar”文件内。点击 **Connect 连接手持机**。由于手持机会自动管理功耗，可能处于待机下无法连接，可以让手持机进入 USB 通讯状态，对应 CL998 机型，点击屏幕的连接电脑图标即可。

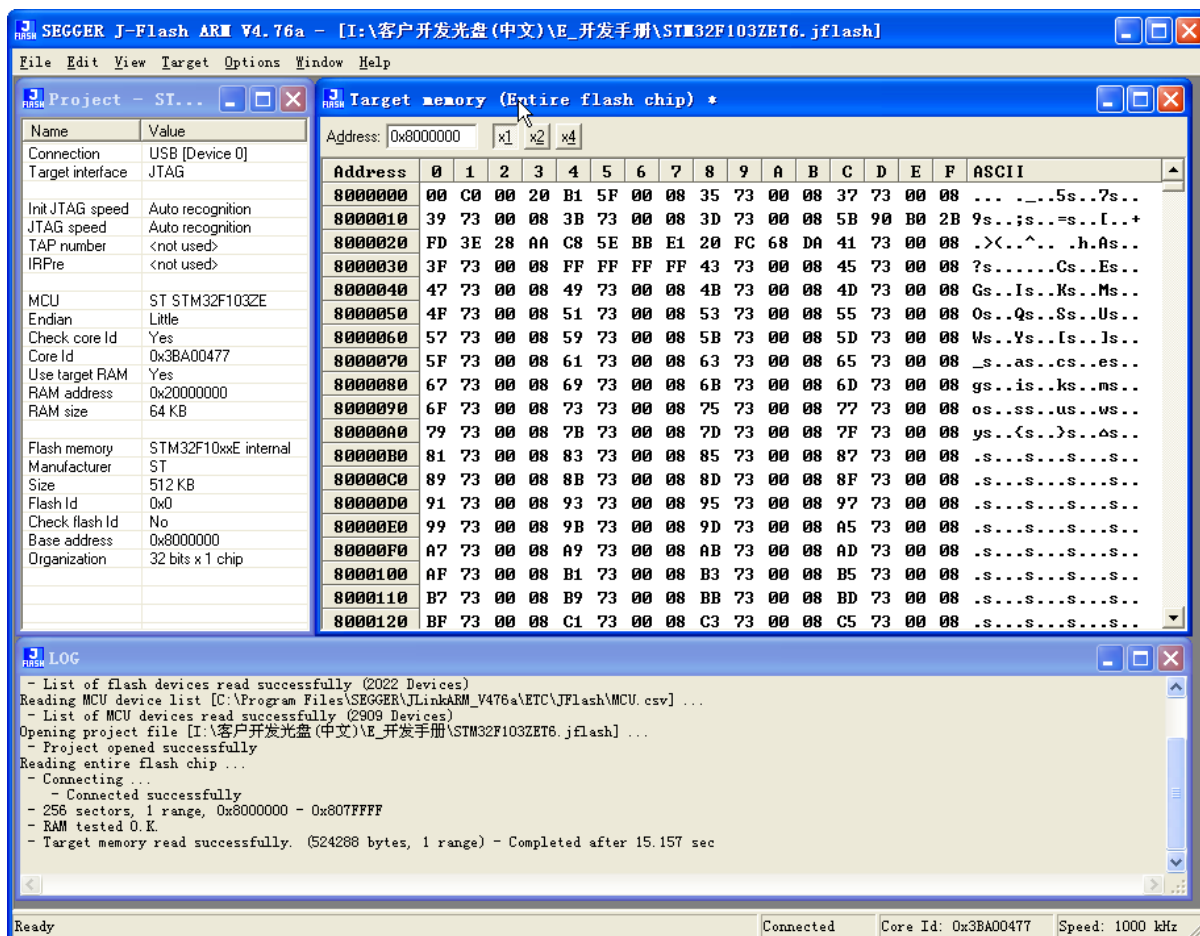
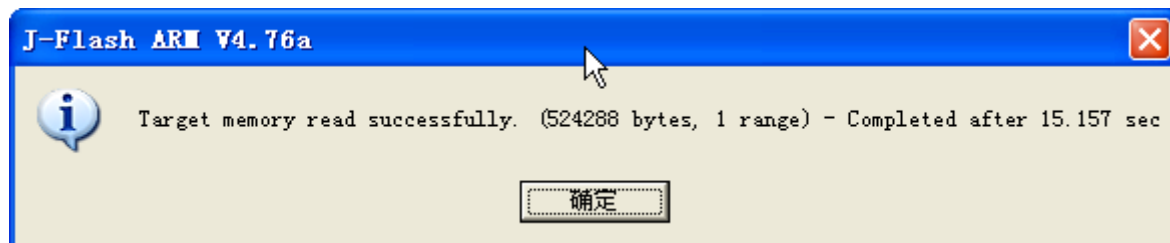
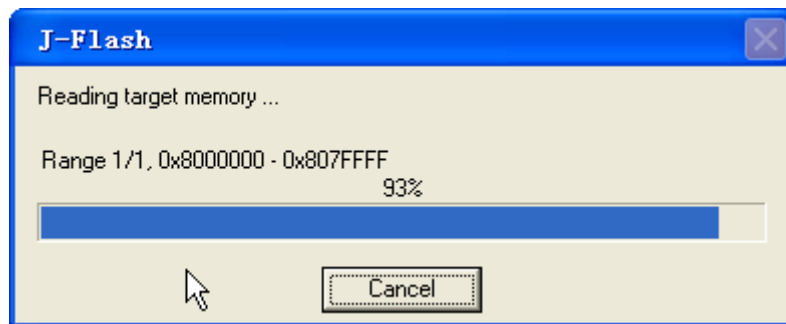




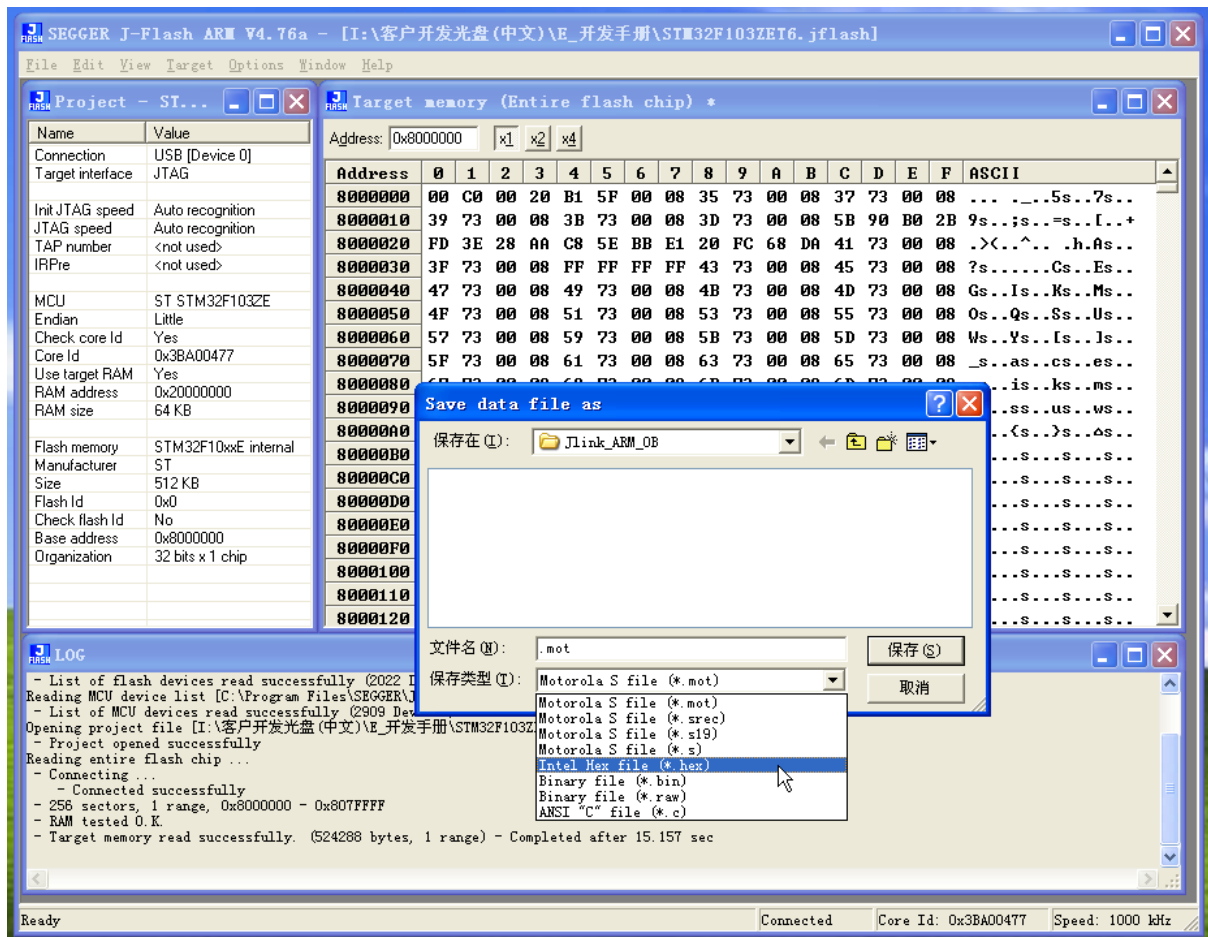
5、选择 Target 下的 Read back 功能，选择读取整个芯片（Entire chip）



软件将整个芯片读取到内存里面



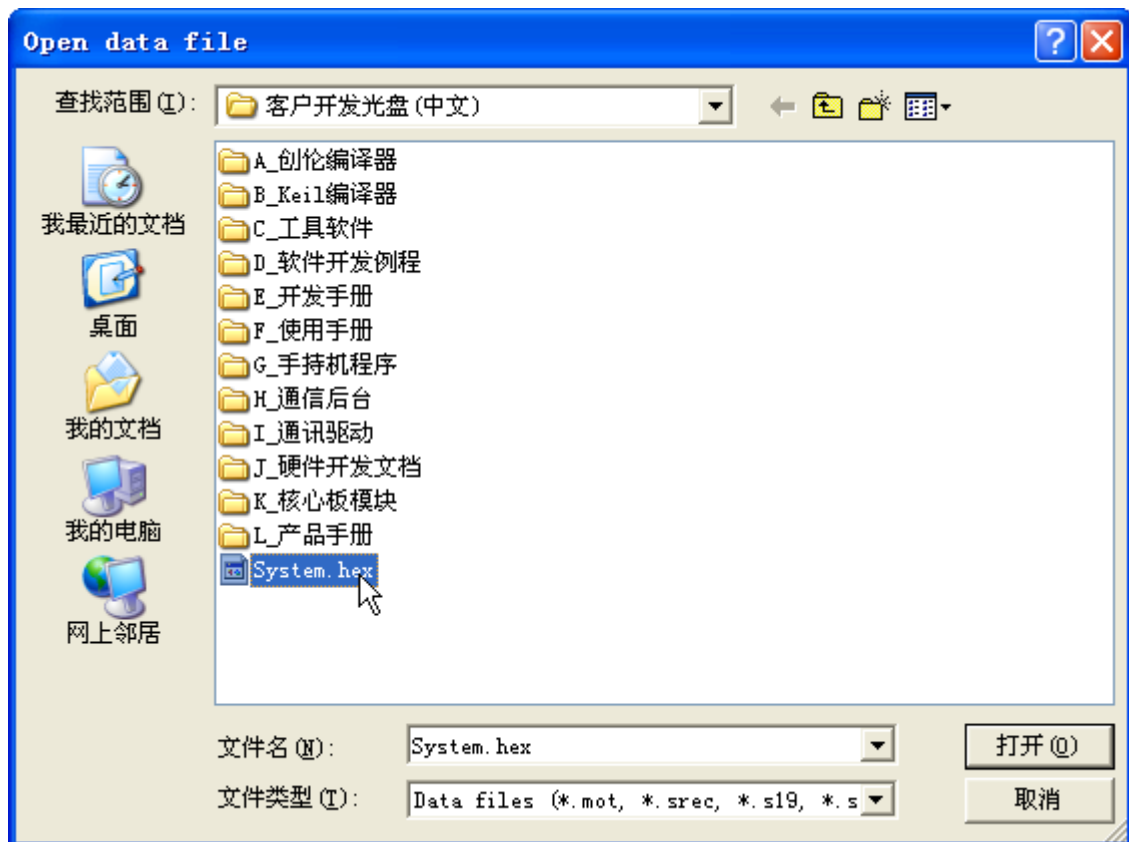
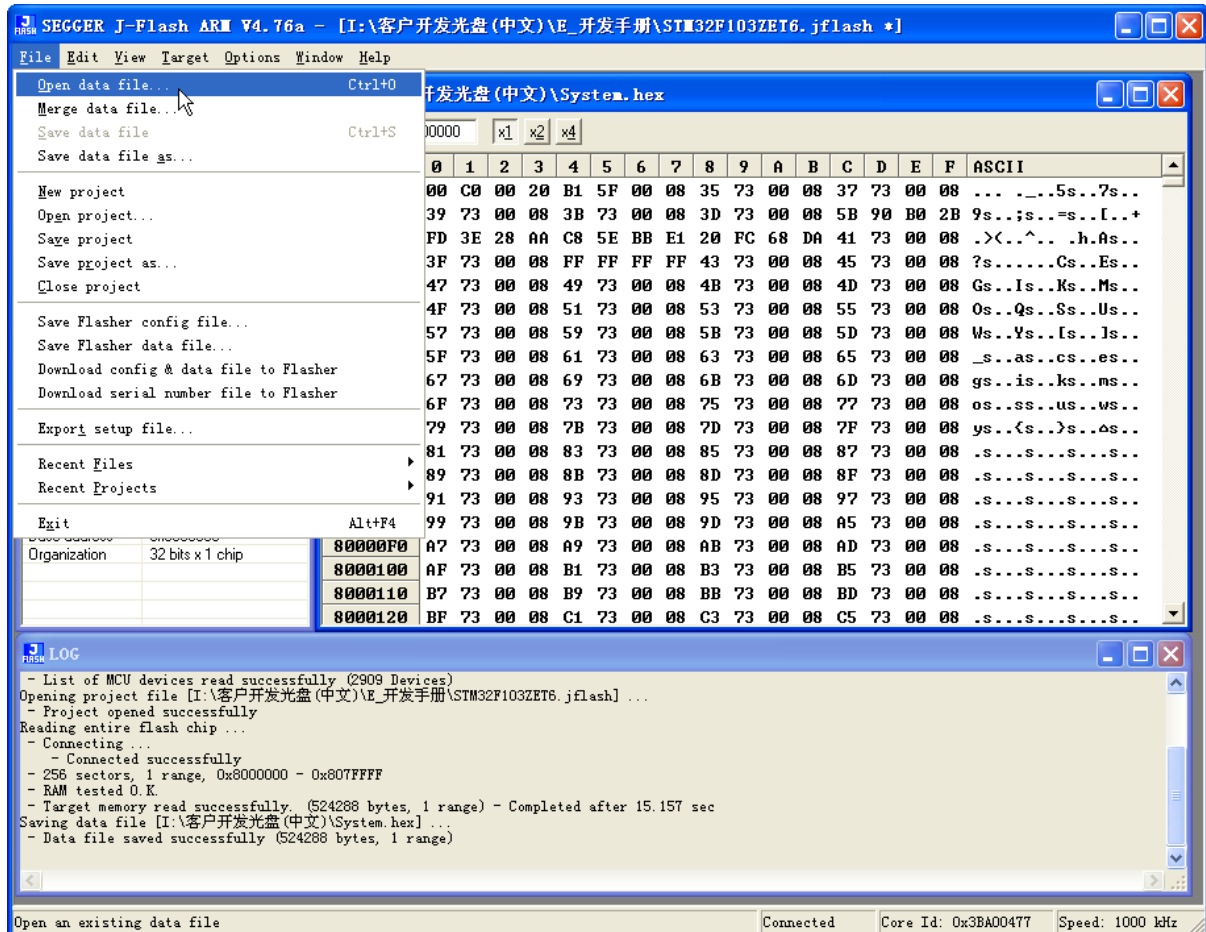
- 6、选择 File 下的 Save data file，将数据保存成文件。一般可以选择保存为 Hex（十六进制）格式，该文件类型包含地址信息，便于回写到芯片内。



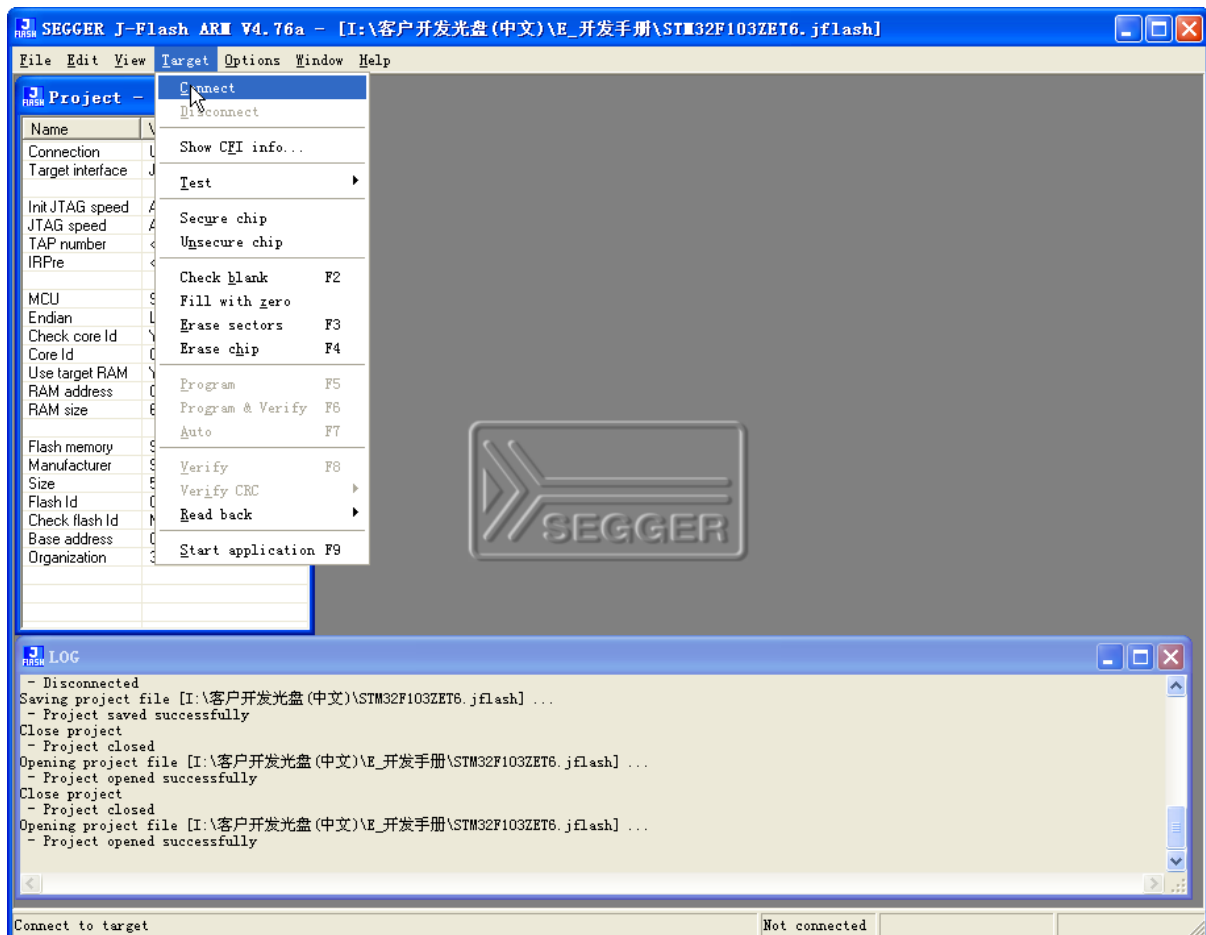
假设我们保存的文件名为 System.hex，选择路径进行保存。

二、还原系统

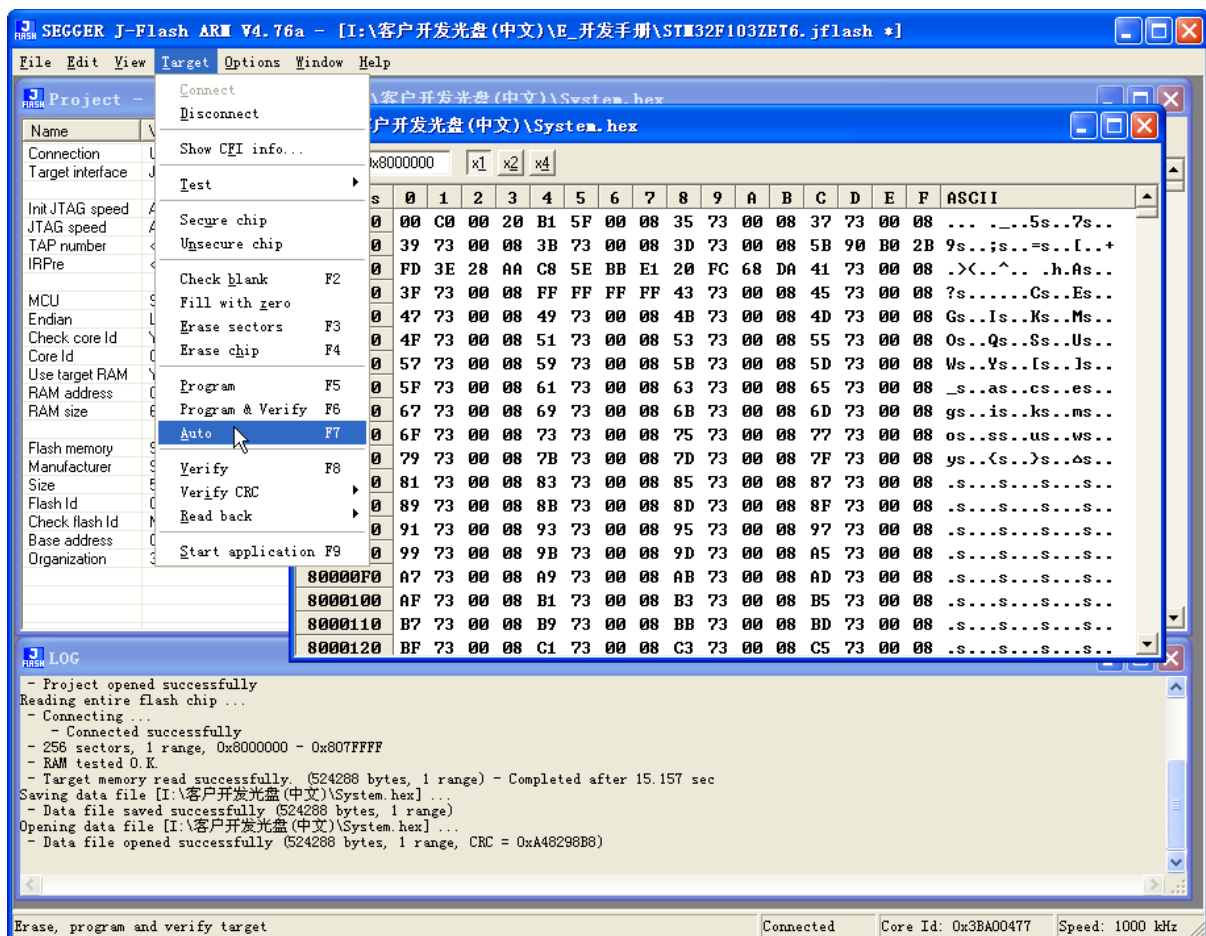
- 1、参考备份系统，先打开软件，再打开工程文件。
- 2、打开需要还原到手持机的系统，比如之前备份的 System.hex.

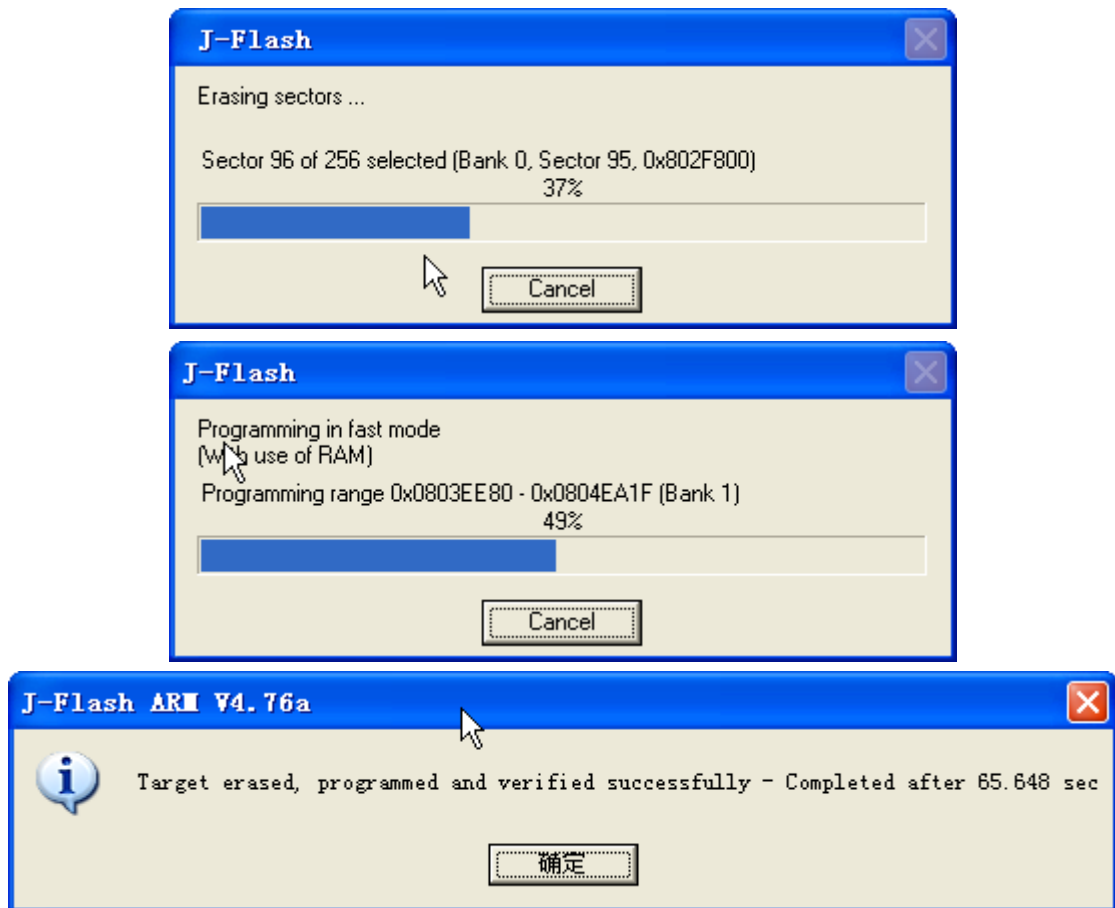


3、然后点击 Target 下面 Connect 连接手持机，（Connect 前，先将 jlink 仿真器和手持机连接）



4、选择 Target 下面的 Auto 进行系统回写。





写入成功后，按手持机的复位键（绿色箭头）进行复位即可。

三、软件篇

软件开发环境及步骤说明

准备工作

Z:\开发光盘\3、软件\C_编译开发环境\Keil V4.54 （C 盘根目录）.rar

Z:\开发光盘\2、例程\A_工程范本\KEILProject.rar

首先您需要获取到以上的几个文件。

Keil V4.54 （C 盘根目录）.rar 是开发时需要的编译器，完成程序的编译，调试，下载的工作。**必须放在 C 盘根目录，即 C:\Keil。**

KEILProject.rar

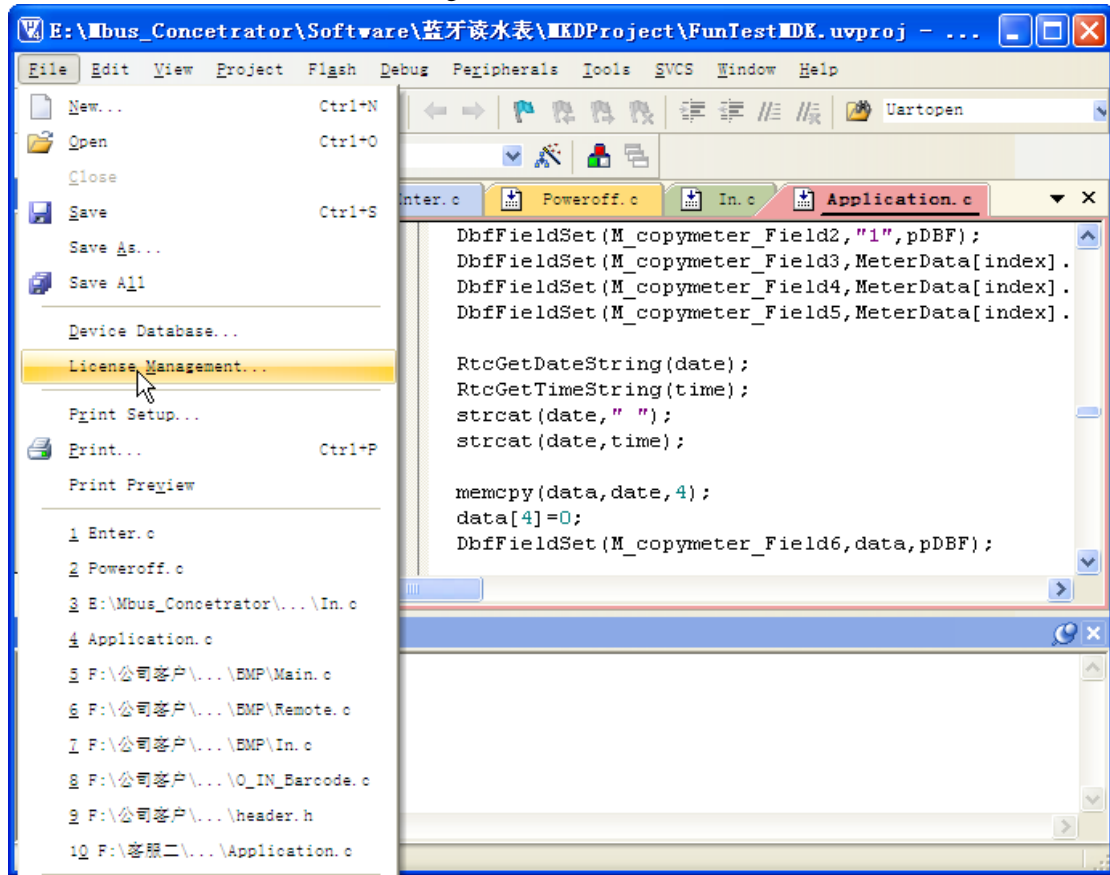
该文件为工程范例文件，里面包含了工程配置，例程源码，调试配置等内容，您可以直接基于该工程进行程序开发，而无需重新建立工程。因为我们的工程是针对我们的开发平台的，有些特殊的配置功能。如果您需要开始新的项目，只需要简单拷贝工程后，添加自己的源代码即可。

一、安装Keil uVision4 for ARM 的版本，请选择4.23 以上的版本，建议选用4.54 的版本，该版本优化了目标代码的加载速度，进行仿真时加载比较快。或者直接使用我们提供精简版本，绿色软件，解压可用。请解压到 C 盘根目录下。

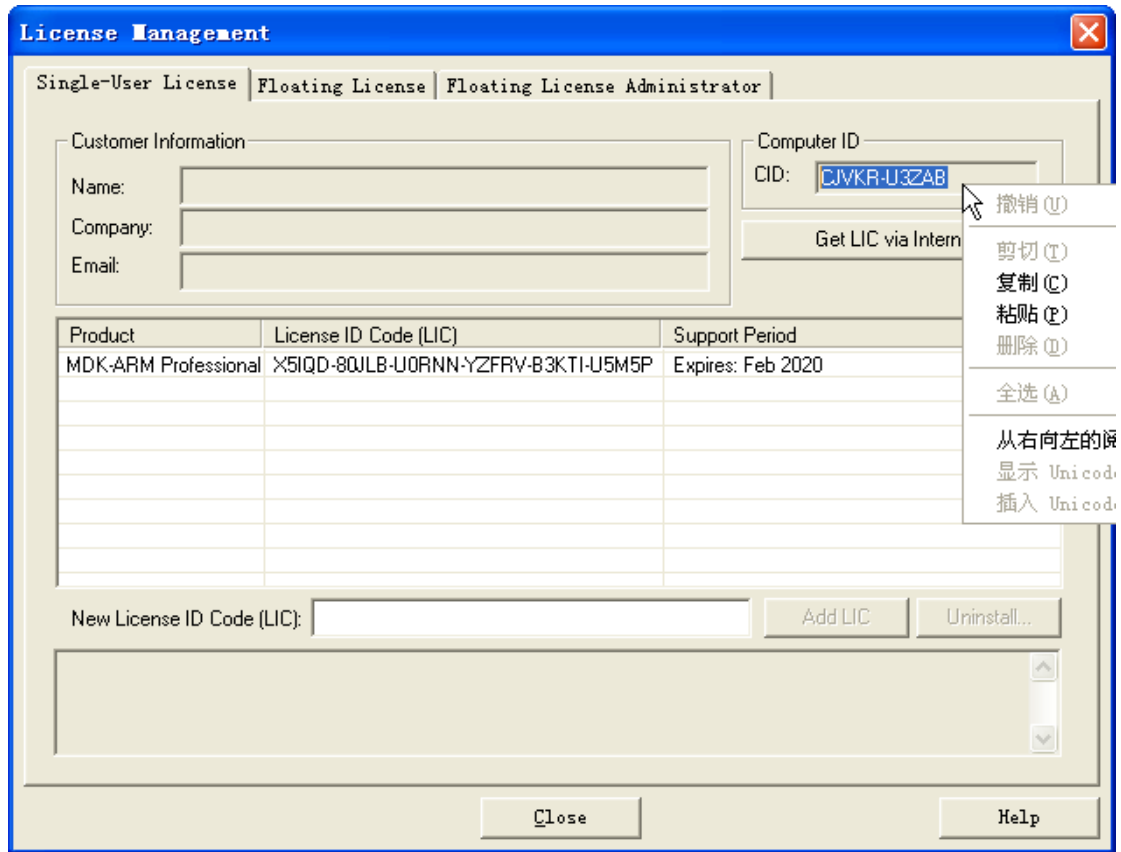
注意：Keil uVision4 for ARM 需要要进行注册使用，如果不注册，只能够编译 32K 的代码，对于编写应用程序，一般是不够的。

注册的方法如下：

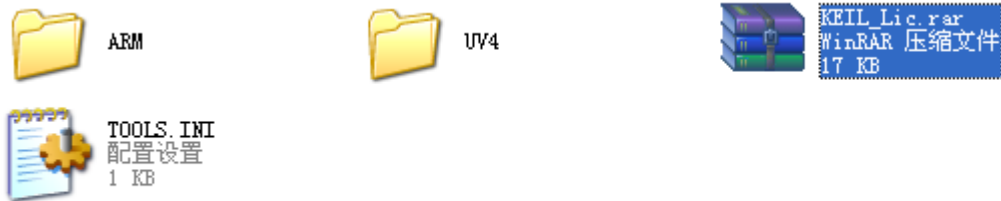
1、打开 KEIL，在 File 菜单下选择 License Management



2、拷贝 Computer ID



3、解压 KEIL_Lic.rar



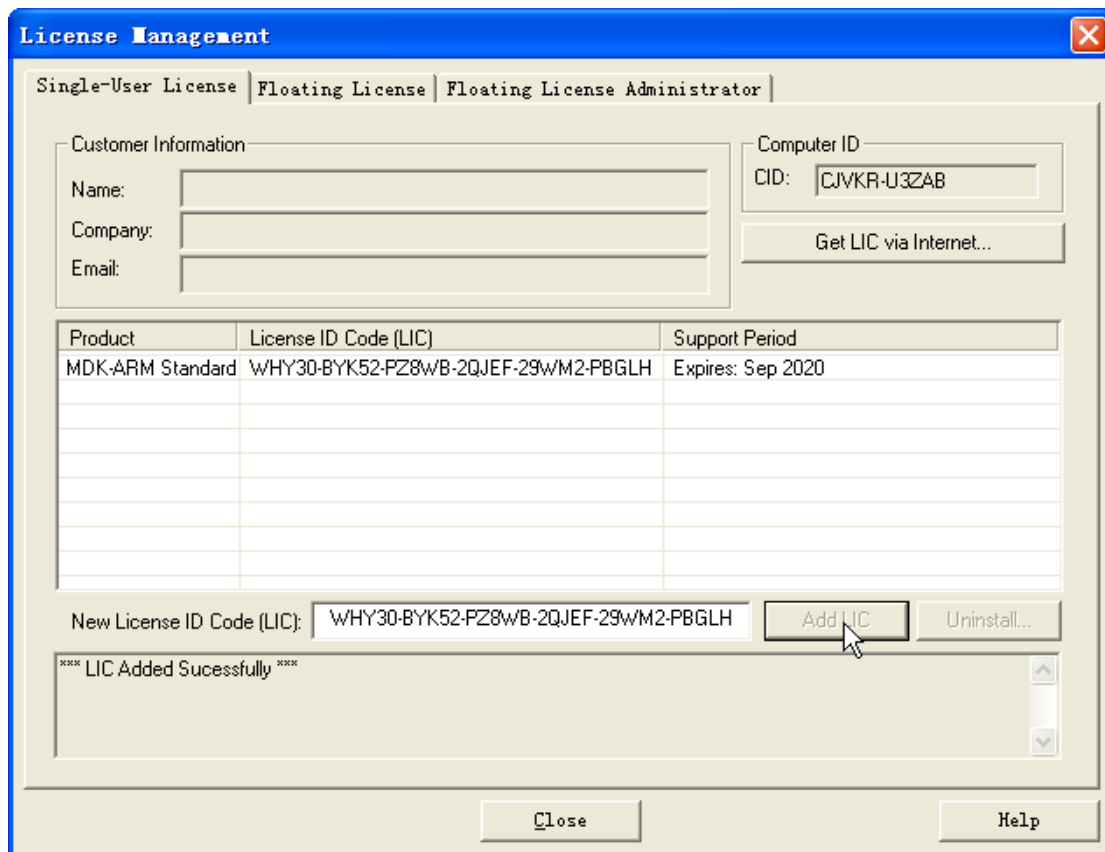
4、运行解压后的文件，并在 Target 中选择 ARM



5、将之前拷贝的 Computer ID (CID) 粘贴到 CID 编辑框，然后点击 Generate 生成 License。

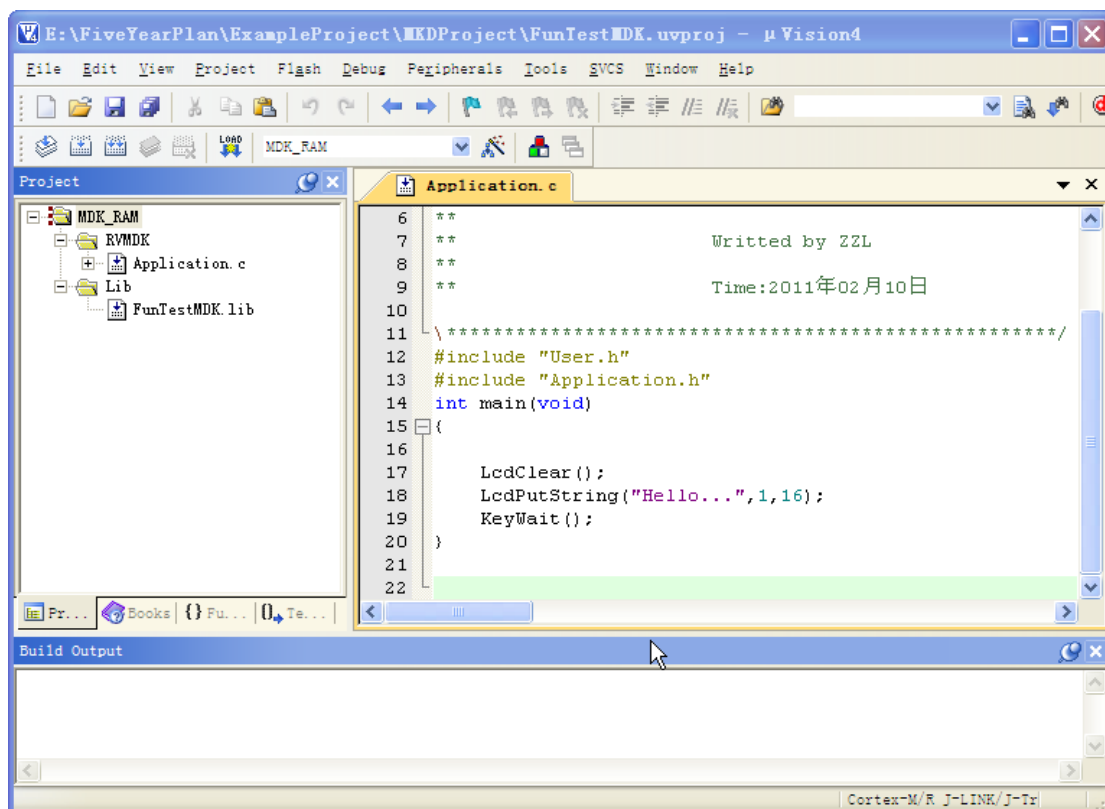


6、将生成的 License 拷贝粘贴到 New License ID Code(LIC)编辑框，并单击 Add LIC 按钮，添加注册码。

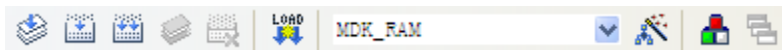


7、信息框提示***LIC Added Sucessfully***表示注册码有效，注册成功。这样就可以正常就代码进行编译。

二、打开 MKDProject 文件夹下的 FunTestMDK.uvproj 工程。出现下面的工程界面，里面有一个简单例程，用户只需要修改 Application.c 的内容，可以添加其他 c 或者 c++ 文件，但是该库文件 FunTestMDK.lib 不能够删除，否则将不能够生成可以运行在手持机中的程序。



三、对代码修改后可以使用下面的工具栏上按键进行编译。



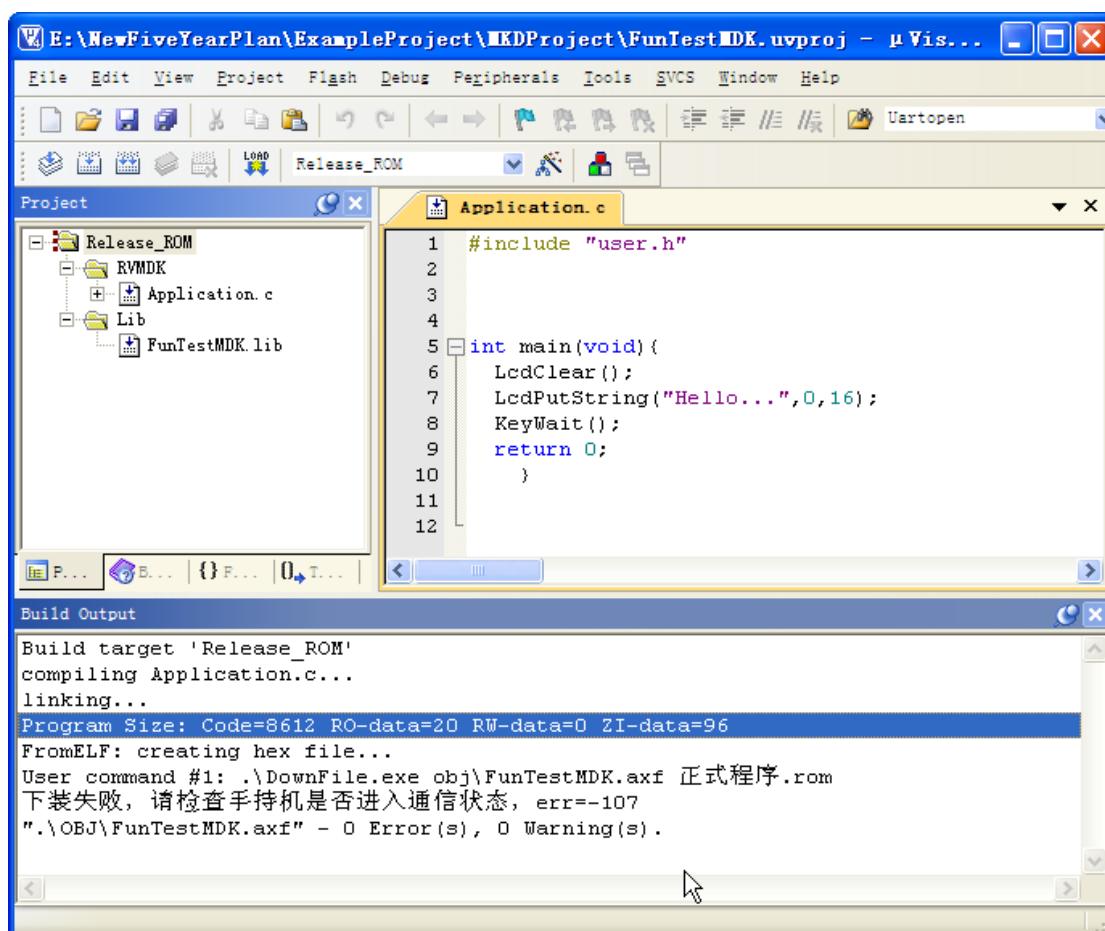
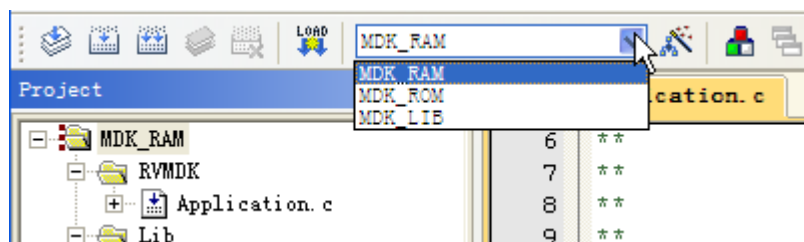
关于 Keil 的编译，仿真调试功能的使用，可以网上搜索。

四、选择项目类型，下拉列表中出现 MDK_RAM、MDK_ROM、MDK_LIB 或者 DEBUG_RAM、RELEASE_ROM、LIBRARY_LIB 三个工程类型，。其中

RAM 可以用于仿真调试程序，也可以生成 ram 文件，但不建议使用。生成的文件在 MDKProject 文件夹。

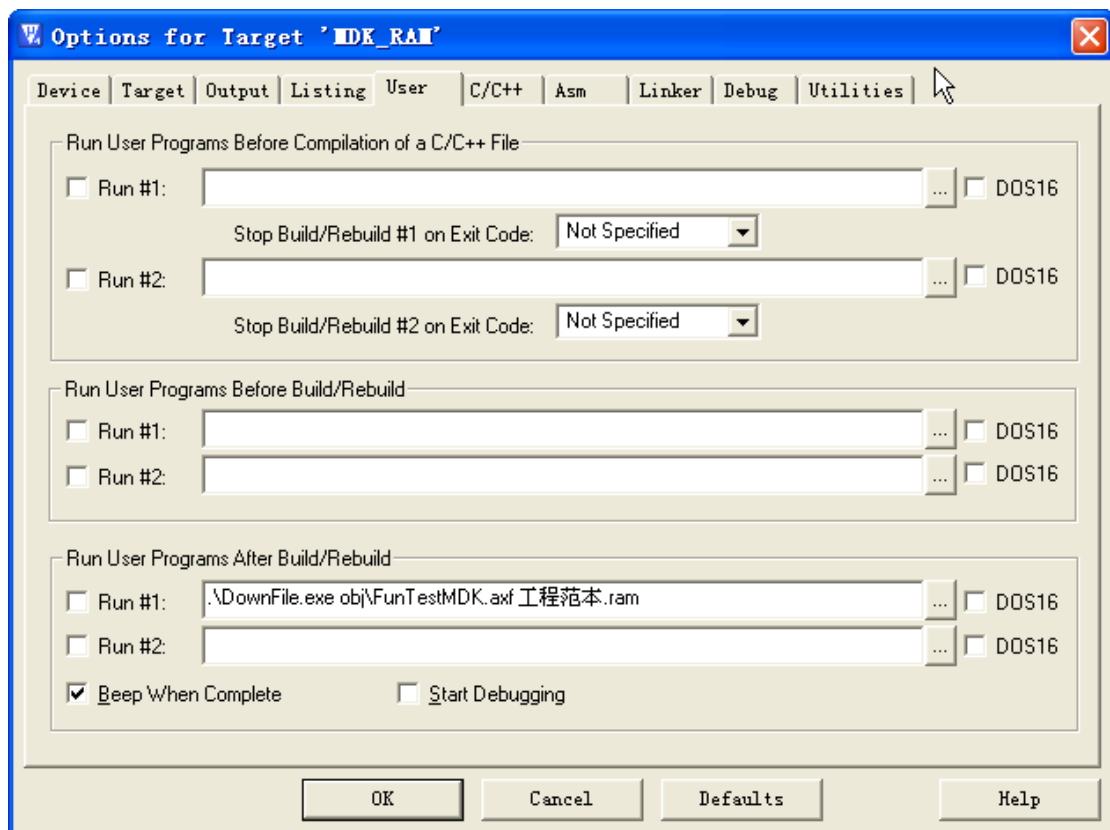
ROM 不可以调试，用于生成可以在手持机中运行的 rom 类型文件。生成的文件在 MDKProject 文件夹

LIB 可以生成库文件，可以进行一些代码封装后提供给第三方使用，生成的文件在 MKDProject\OBJ

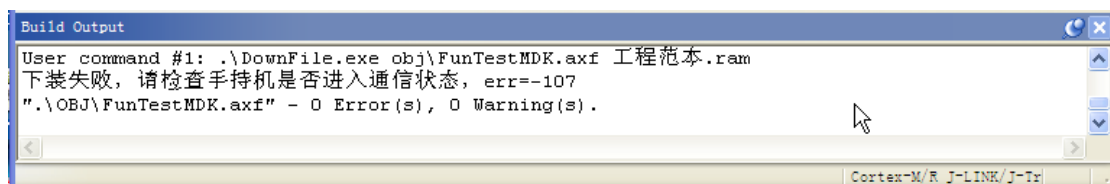


ram 文件和 rom 文件的区别在于 ram 生成的文件在内存中运行，而内存是相对紧缺的资源。全局变量也分配在内存中，能够运行的程序比较小，一般要求编译后小于 96K（上图中蓝色区域 Code+RO+RW+ZI）。rom 文件实在 flash(相当于硬盘)中运行，相对而言资源较多，而且不会和全局变量抢资源。一般可以达到 448K(Code+RO+RW) 另外(RW+ZI 要小于 96K)。

如需要生成 ram 文件，请在下面的配置页面中 **Run#1** 上打上勾。



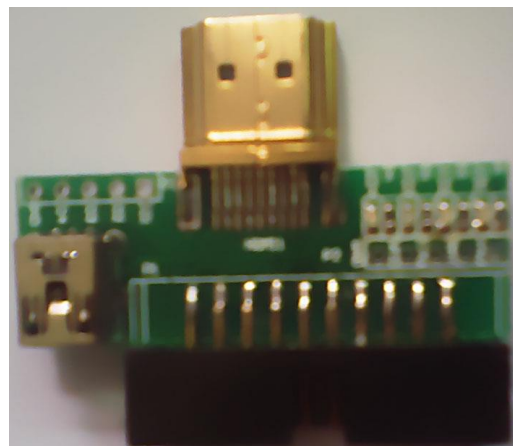
打上勾后，在进行编译时，会自动生成工程范本.ram 并调用通信功能将 ram 文件下载到手持机，如果手持机不在通信状态，会提示以下错误。

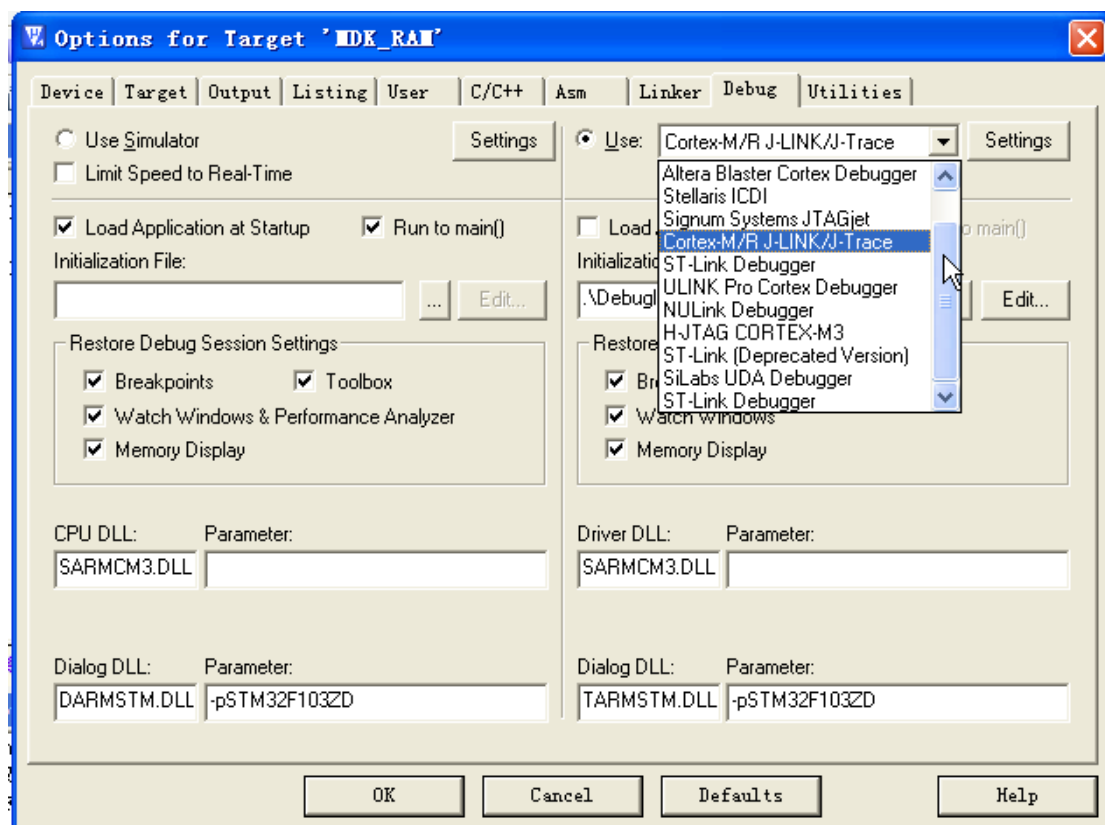



对于 rom 文件的下载配置和 ram 方法是一样的。

五、仿真器配置，在下面的对话框下拉列表中选择您使用的仿真器类型。一般我们配套的仿真器使用 JLINK 仿真器。使用仿真器能够更好的开发调试程序。

下图为 MINI 仿真器和仿真器转换接板，如果您已经有仿真器，可选用仿真转换板。

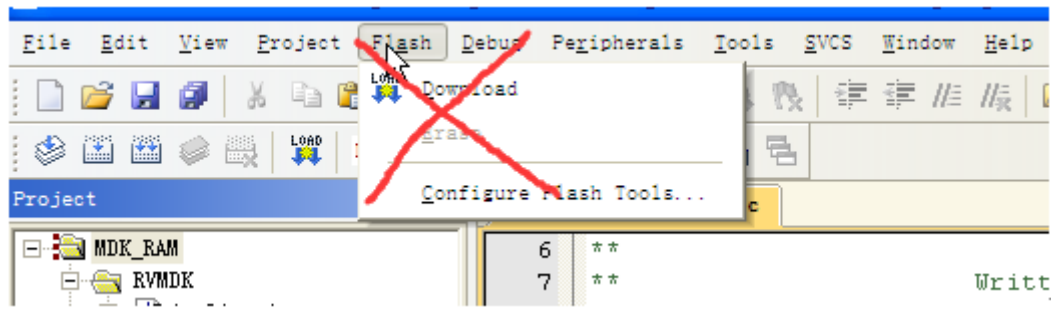




六、点击工具栏中的 按钮进行仿真，手持机上电后第一次进行仿真前，需要将手持机进入通信状态，否则仿真器连接会失败。成功连接后，就不需要再进通信状态。



七、注意，**不要**配置菜单中 **flash** 的内容，更不要使用其中的功能，否则或造成系统被擦除的问题。系统被擦除后，可能需要发回来重新烧写系统，由此产生的费用由用户负责。



八、

注意：程序是通过 USB 通讯器进行下载的，而不是通过仿真器下载，下载方法见前面的说明，下载时需要安装手持机的相关驱动。

九、使用仿真功能可以进行代码的单步调试，设置断点，查看内存等功能，方便用户进行二次开发。

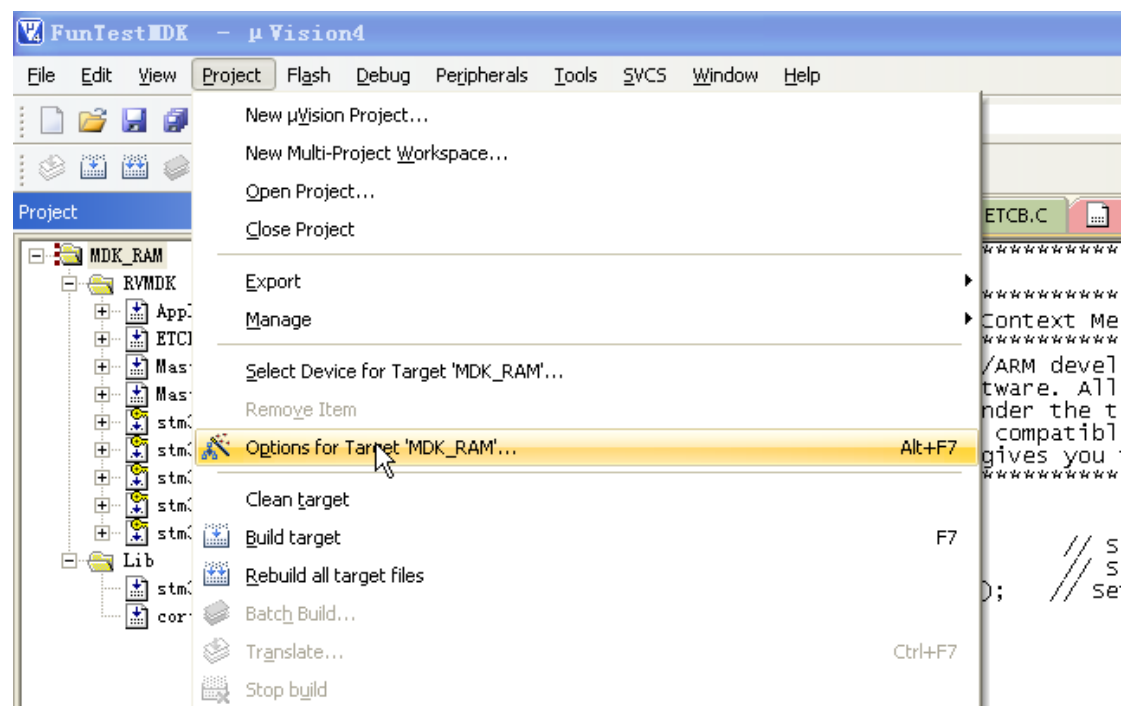
十、不使用仿真功能，调试代码时需要将结果 ram 或者 rom 下载到机器内运行，根据运行的结果进行修改程序，相对仿真功能而言麻烦些。

```
Application.c
6  **
7  **                               Writted by ZZL
8  **
9  **                               Time:2011年02月10日
10
11 \ *****/
12 #include "User.h"
13 #include "Application.h"
14 int main(void)
15 {
16
17     LcdClear();
18     LcdPutString("Hello...",1,16);
19     KeyWait();
20 }
21
22
```

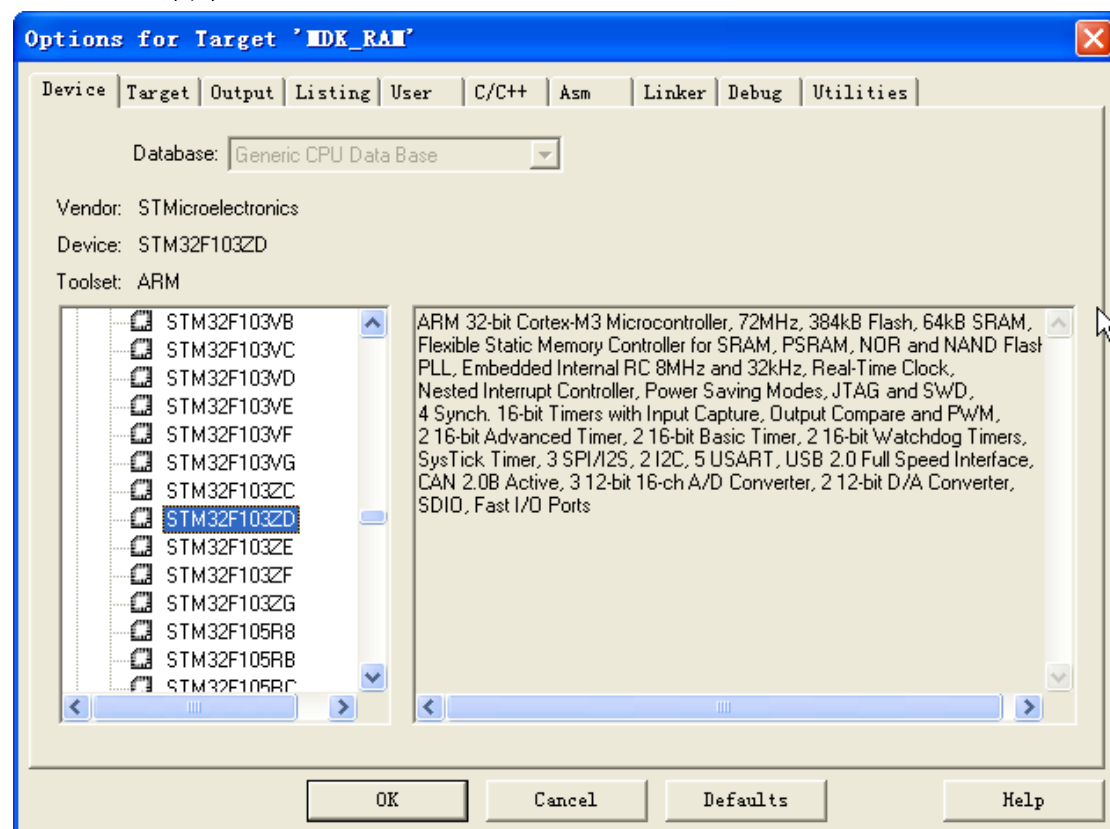
如何修改工程配置说明：

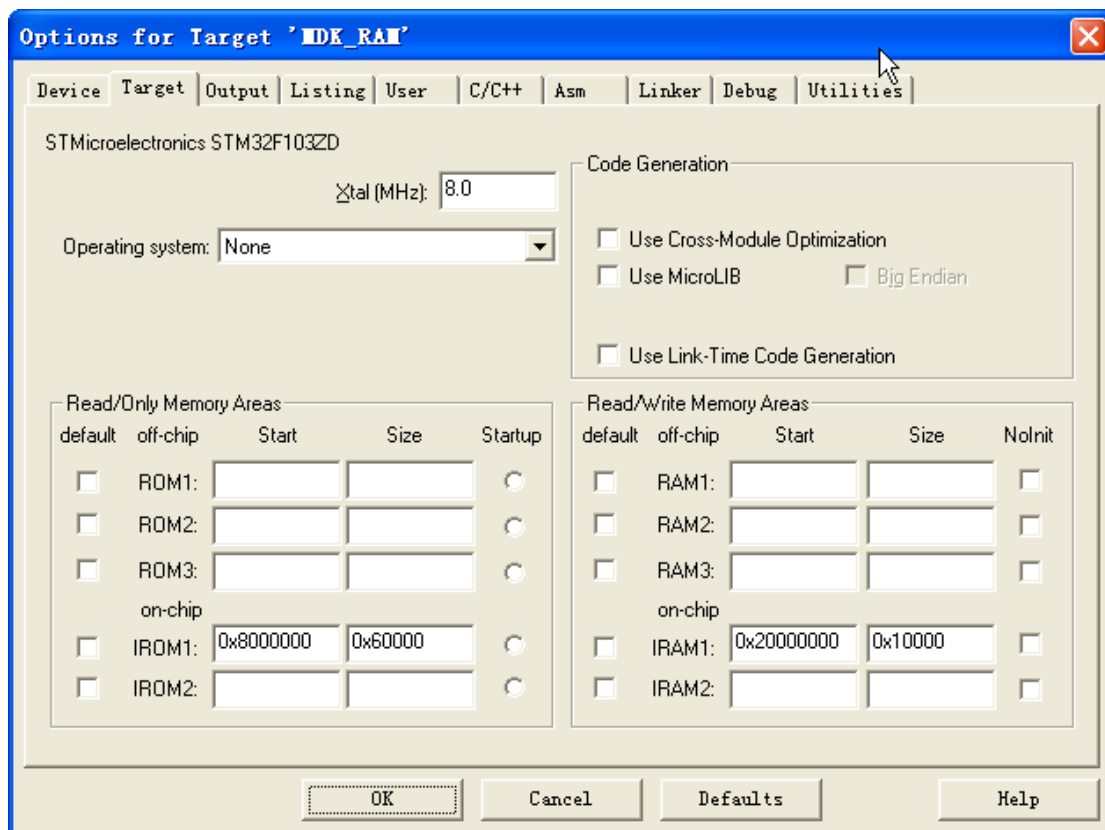
（注意：一般情况下直接使用工程模板即可，无需进行配置）

一、 打开工程配置：

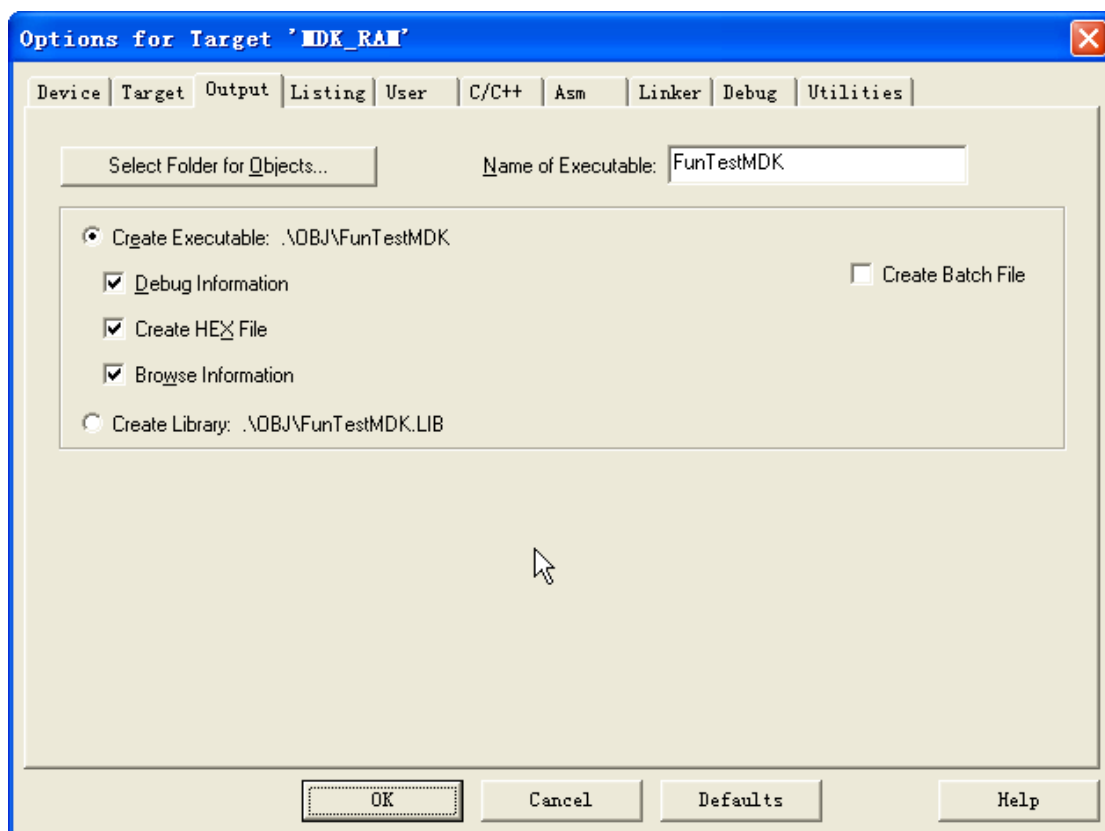


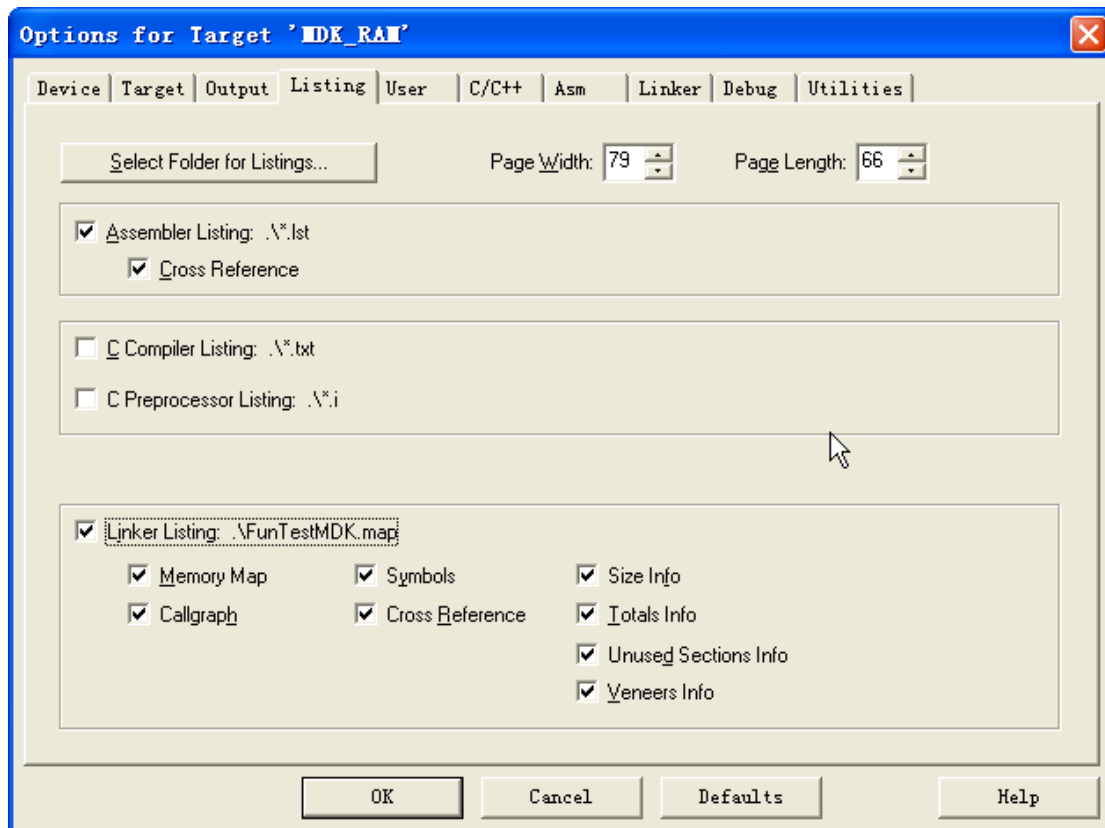
二、选择 STM32F103Z 系列 CPU



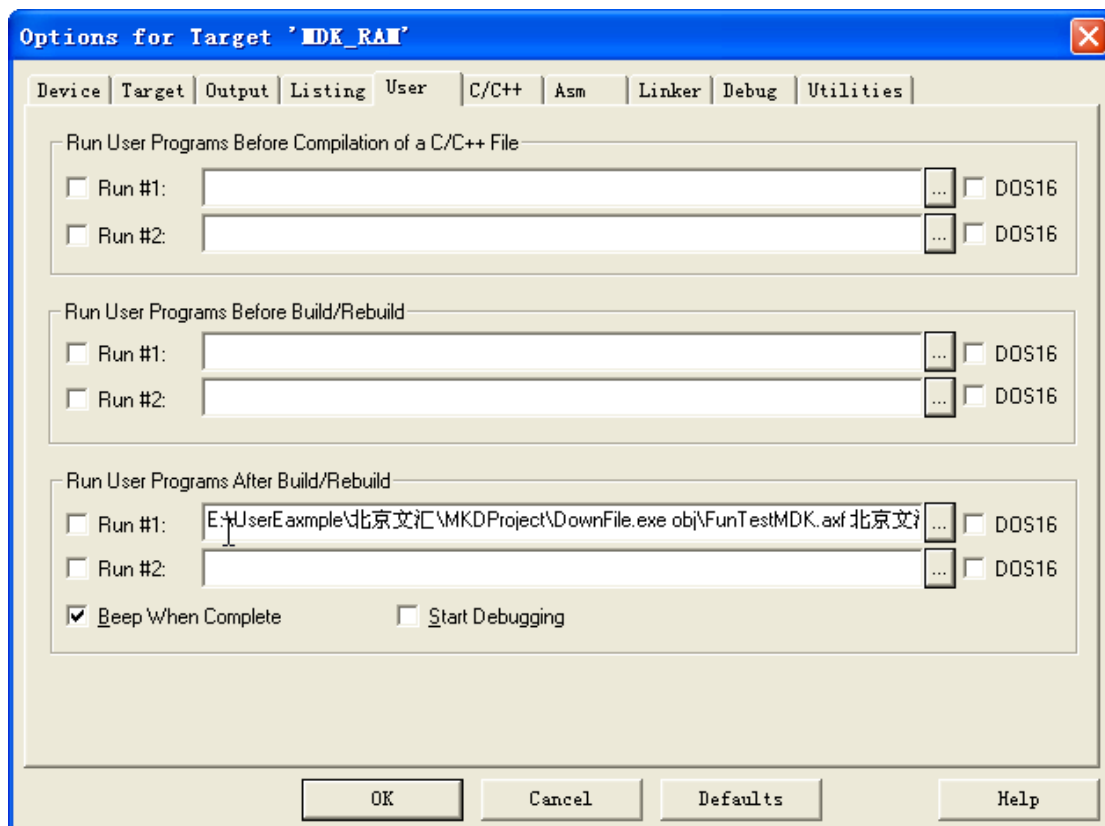


选择 obj 文件存放的文件夹，默认是工程文件夹。一般在工程文件夹下新建一个 obj 文件，将生成的 obj 文件放入该文件夹，这样不会使工程文件夹过于混乱。使用 obj 文件夹时，对于的 downfile.exe 可执行文件参数也需要对应修改

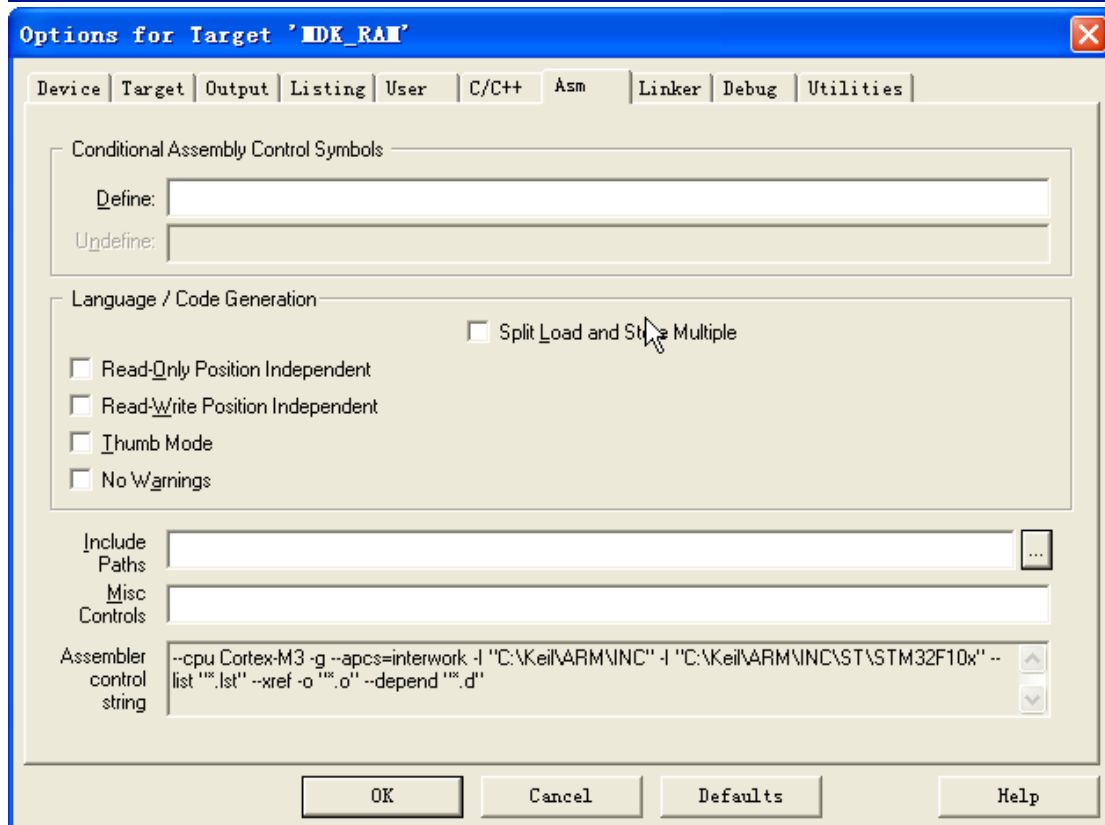
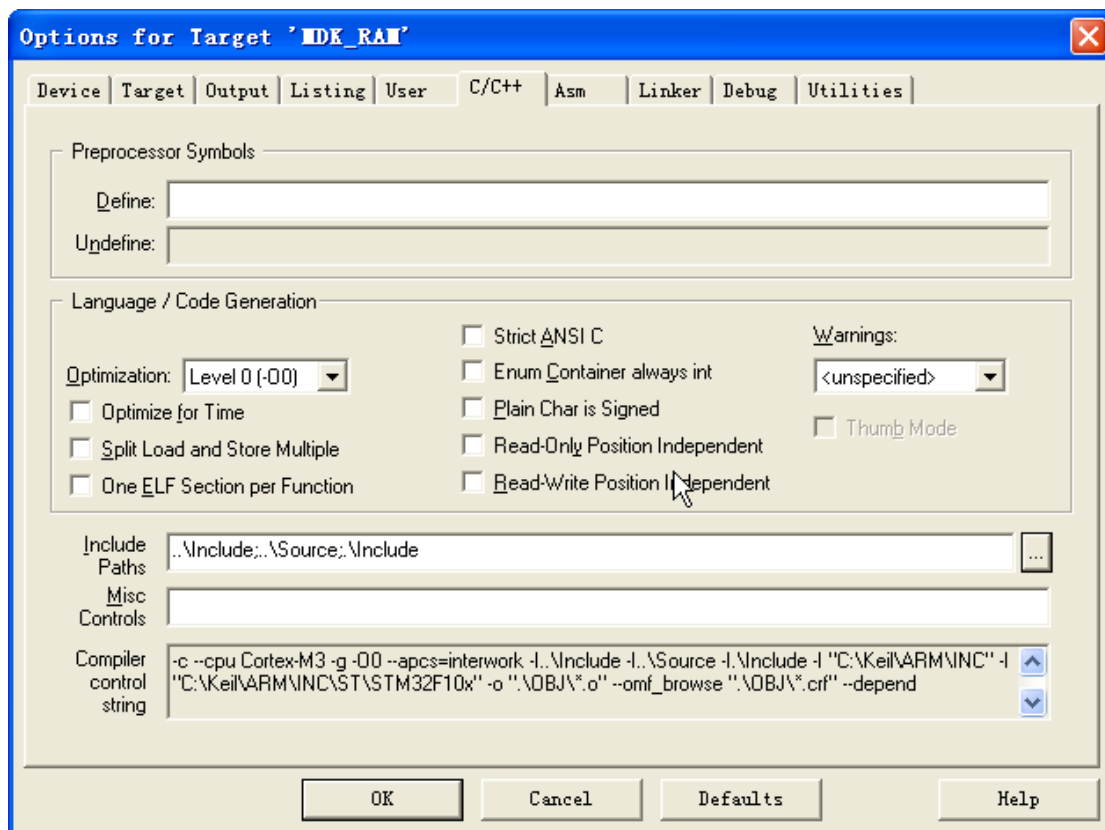




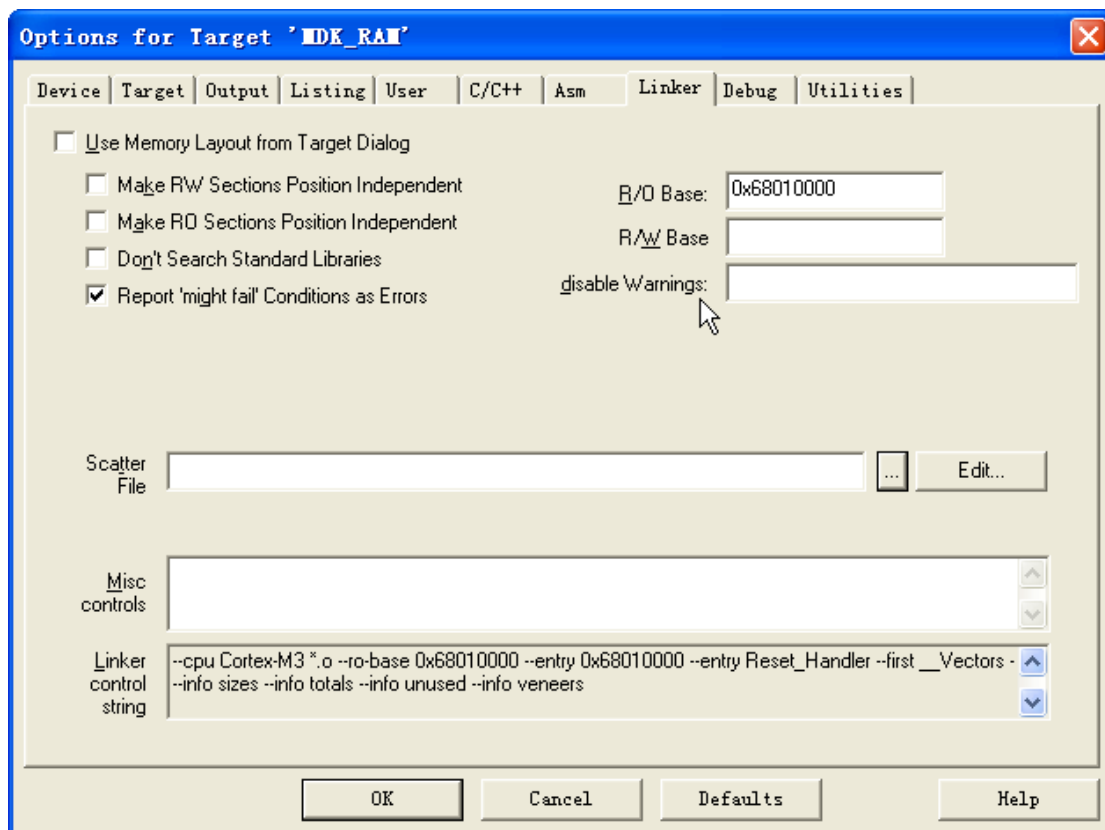
在 Run User Programs After Build/Rebuild 中的 Run#1 加入 downfile.exe 命令，可以在完成编译后自动下载文件到手持机（手持机应处于通信状态，否则会报错），downfile.exe 可执行文件有两个参数，第一个是项目编译后生成的 elf 格式的文件，另外一个是在下载到手持机后的文件名称。如 DownFile.exe obj\FunTestMDK.axf 北京文汇.ram。obj 表示位于 obj 目录下



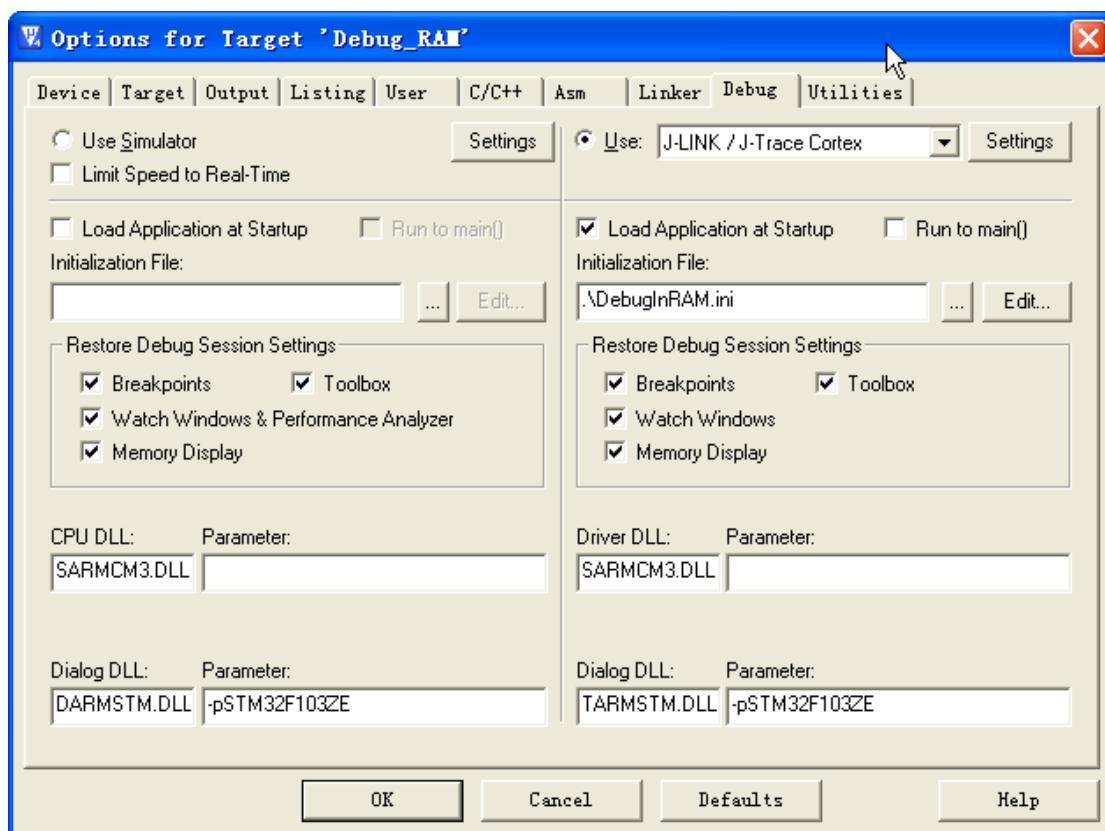
添加工程需要包含的路径，一般是头文件和源文件路径



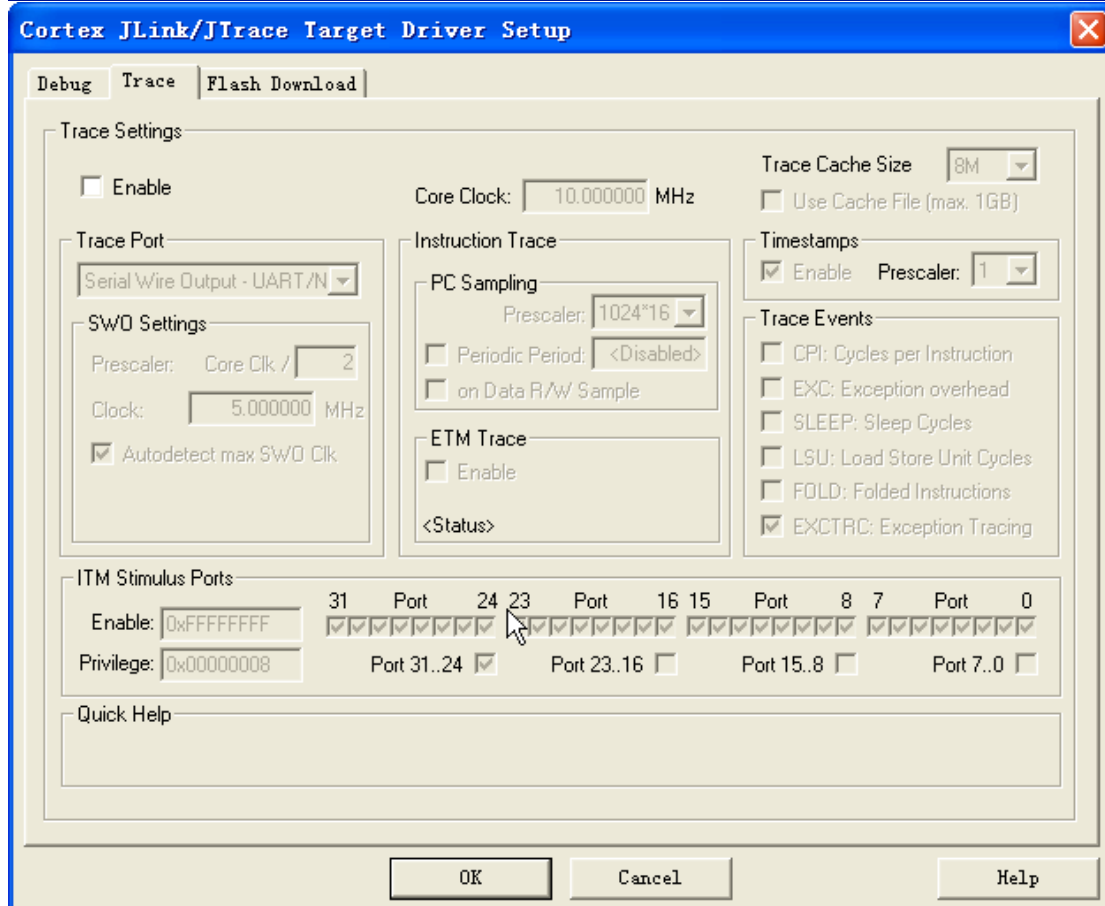
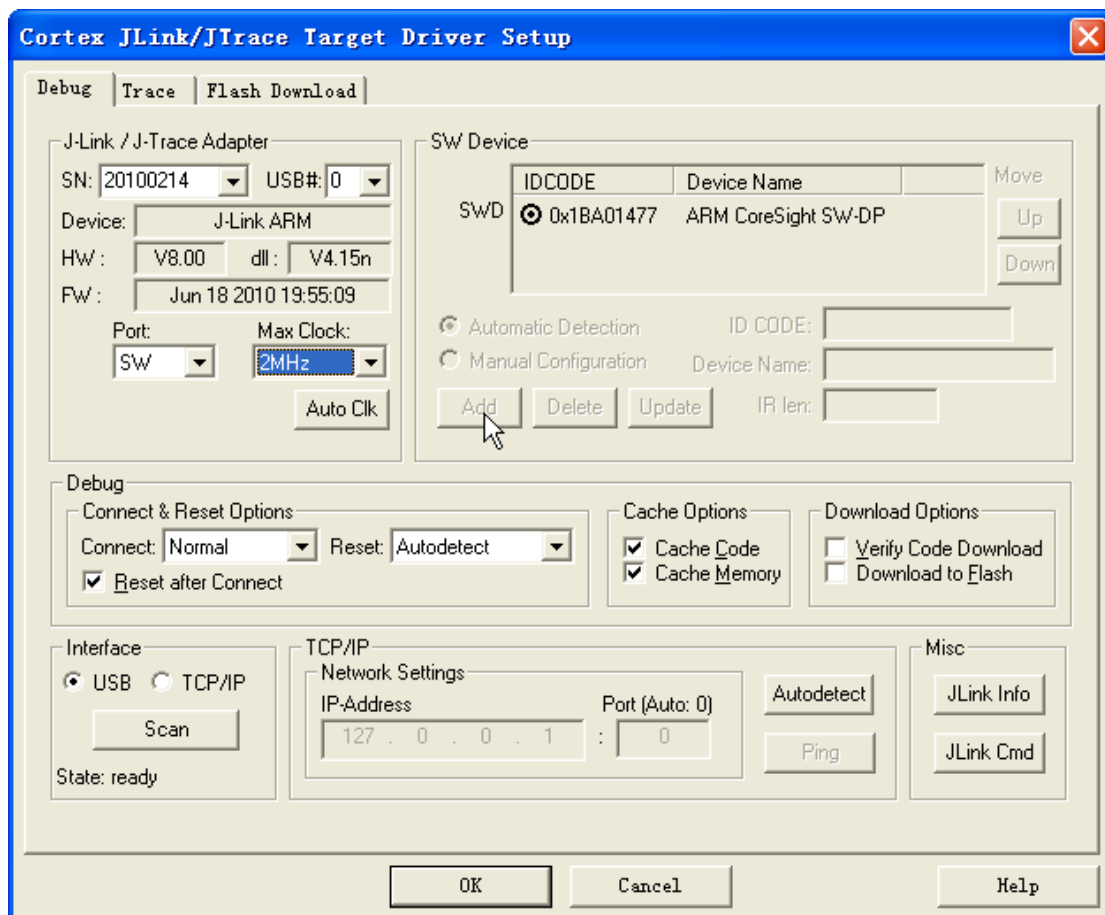
该处 R/O Base:参数必须为 0x68010000,否则程序不能运行。



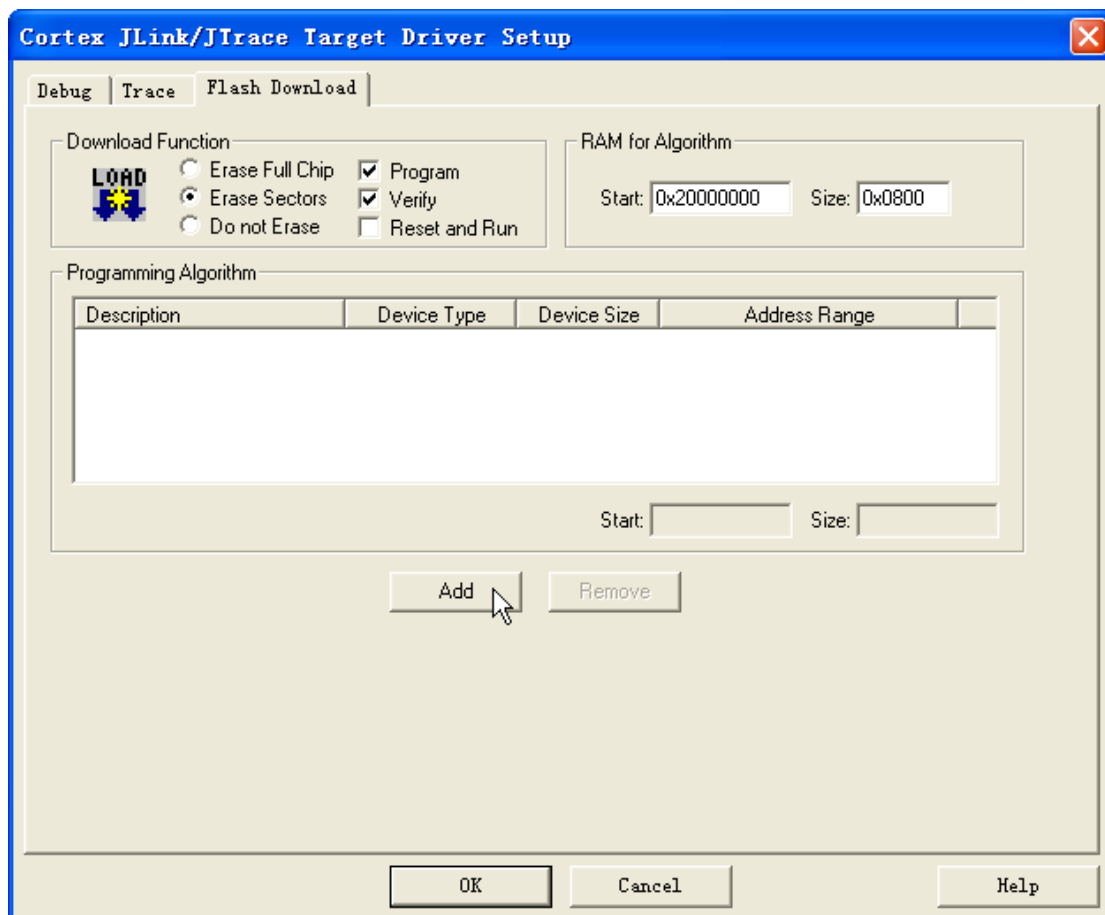
Initialization File 选择 DdebugInRam.ini 文件，否则不能进行仿真调试。仿真器根据需要选择，该例子为 Cortex-M/R J-LINK/J-Trace



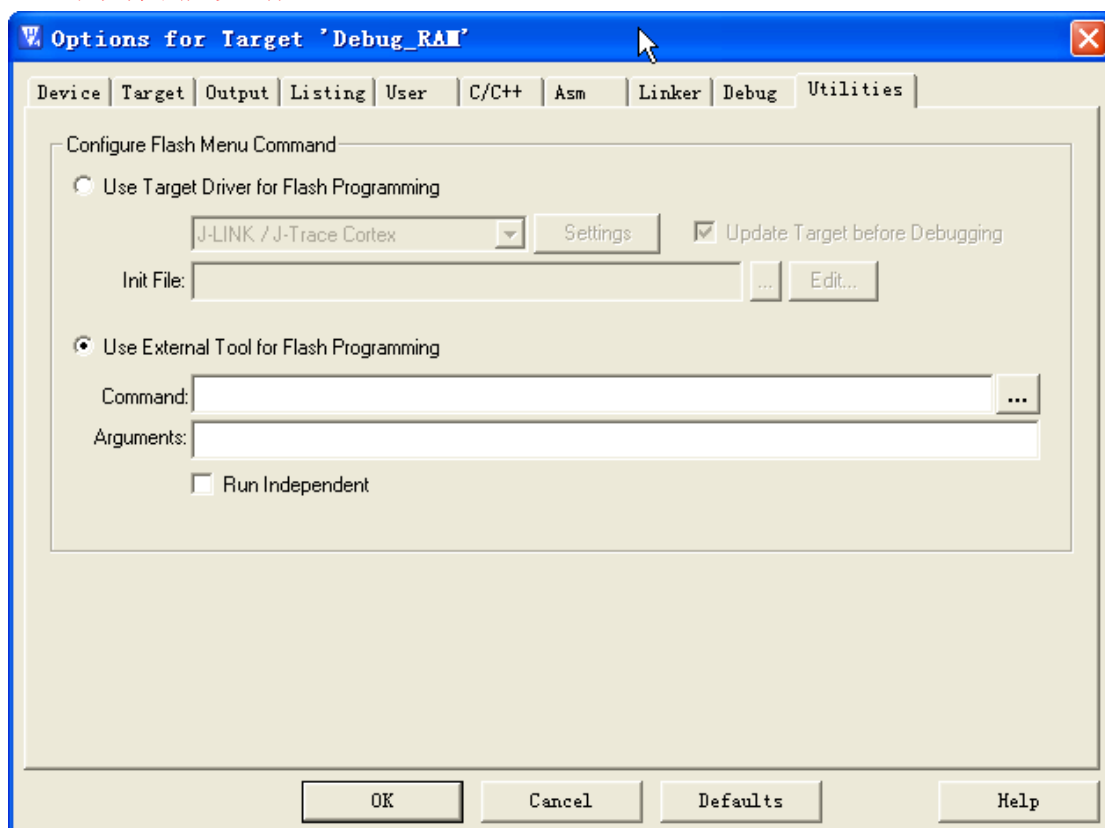
选择 Settings 按键，出现一下界面，Port 选择 SW 模式，Max Clock 选择较低的频率，如 2MHz，以确保能够顺利连接目标系统



Flash Download 不需要添加任何器件。



Utilities 页面配置不需要使用 **Update Target before Debugging**，也就是不要对系统 flash 进行任何操作，否则一旦擦除了系统 flash，系统将不能够运行。



常见问题：

1. 点击调试按钮，出现断点错误。



当使用以前的旧工程在新机器上调试时出现以上对话框，可以将新的 **DebugInRAM.ini** 文件替换原来的。内容如下：

```
FUNC void InitSram (void) {
    _WDWORD(0x40021014, 0x00000114);    //
    _WDWORD(0x40021018, 0x000001E0);
    _WDWORD(0x40011400, 0xB4BB44BB);    //
    _WDWORD(0x40011404, 0BBBBBBBB);    //
    _WDWORD(0x40011800, 0BBBBB4BB);    //A23 not use
    _WDWORD(0x40011804, 0BBBBBBBB);    //
    _WDWORD(0x40011C00, 0x44BBBBBB);    //
    _WDWORD(0x40011C04, 0BBBBB4344);    //PF10 output
    _WDWORD(0x40011C10, 0x00000400);    //SET PF10 POWER ON
    _WDWORD(0x40012000, 0x44BBBBBB);    //
    _WDWORD(0x40012004, 0x334B3BB3);    //
    _WDWORD(0x40012010, 0x00009E00);    //G
    _WDWORD(0x40012014, 0x00004100);    //RESET PG8 PG14 LED ON
    _WDWORD(0xA0000000, 0x00001011);    //Bank1_NAND_FLASH
    _WDWORD(0xA0000004, 0x00000200);    //
    _WDWORD(0xA0000008, 0x00001011);    //Bank2_NOR_FLASH
    _WDWORD(0xA000000C, 0x00000200);    //
    _WDWORD(0xA0000010, 0x00001011);    //Bank13_SRAM
    _WDWORD(0xA0000014, 0x00000200);    //
    _WDWORD(0xA0000018, 0x00001011);    //Bank4_LCD
    _WDWORD(0xA000001C, 0x00000200);    //
    _WDWORD(0x68004FFC, 0x55aaaa55);
}
InitSram();                //初始化外部 SRAM

LOAD obj\FunTestMDK.axf INCREMENTAL    //下载程序 Download

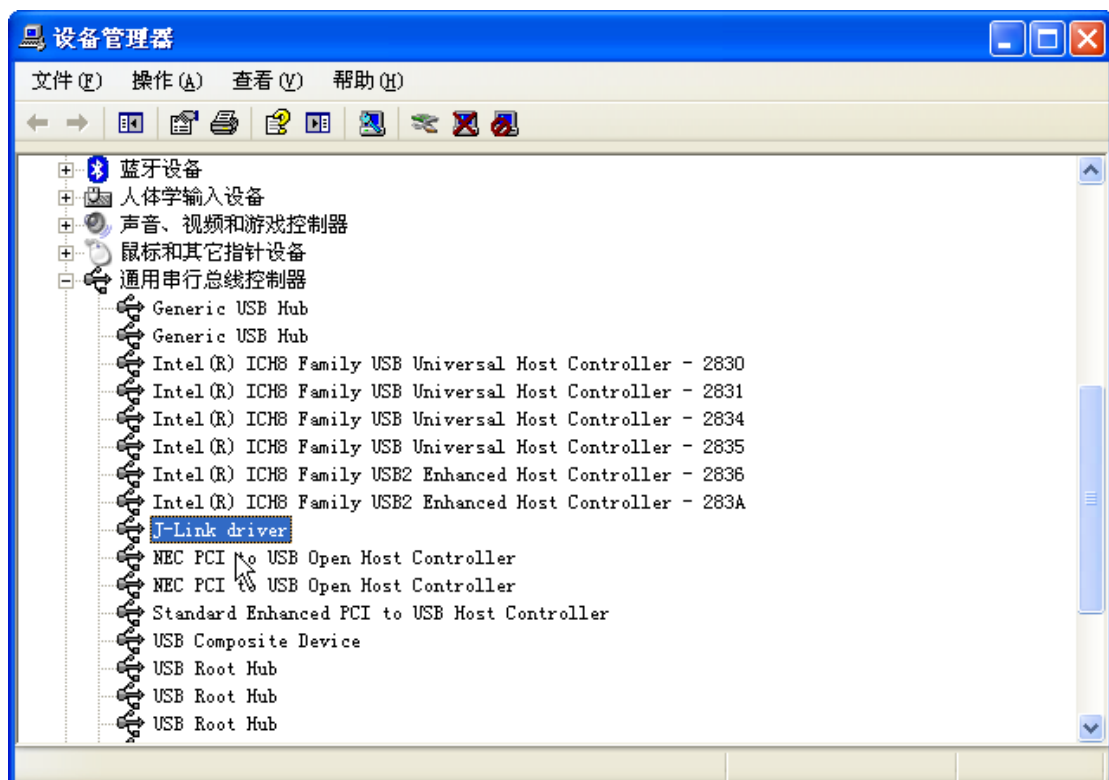
G ,main                //运行
```

2. 点击调试按钮，出现连接错误。



解决方法：

- 1、 请确保手持机放入电池，而不是 usb 直接供电。 因为 usb 供电有限，调试时会由于供电不稳定而造成连接异常。
- 2、 确保仿真器和手持机连接可靠，仿真器驱动正常安装好。



Options for Target 'Debug_RAM'

Device | Target | Output | Listing | User | C/C++ | Asm | Linker | Debug | Utilities

☐ Use Simulator Settings
☐ Limit Speed to Real-Time

☒ Load Application at Startup ☒ Run to main()
 Initialization File: ... Edit...

☒ Breakpoints ☒ Toolbox
☒ Watch Windows & Performance Analyzer
☒ Memory Display

CPU DLL: Parameter:
 SARMCM3.DLL

Dialog DLL: Parameter:
 DARMSTM.DLL -pSTM32F103ZE

☐ Use: J-LINK / J-Trace Cortex Settings
☒ Load Application at Startup ☐ Run to main()
 Initialization File: ... Edit...

☒ Breakpoints ☒ Toolbox
☒ Watch Windows
☒ Memory Display

Driver DLL: Parameter:
 SARMCM3.DLL

Dialog DLL: Parameter:
 TARMSTM.DLL -pSTM32F103ZE

OK Cancel Defaults Help

Cortex JLink/JTrace Target Driver Setup

Debug | Trace | Flash Download

J-Link / J-Trace Adapter
 SN: USB#:
 Device: J-Link ARM-OB STM32
 HW: V7.00 dll: V4.40c
 FW: J-Link ARM-OB STM32 comp
 Port: Max Clock:
 SW 10MHz
Auto Clk

SW Device

IDCODE	Device Name	Move
0x1BA01477	ARM CoreSight SW-DP	Up Down

☒ Automatic Detection ID CODE:
☐ Manual Configuration Device Name:
Add Delete Update IR len:

Debug
 Connect & Reset Options
 Reset:
☒ Reset after Connect

Cache Options
☒ Cache Code
☒ Cache Memory

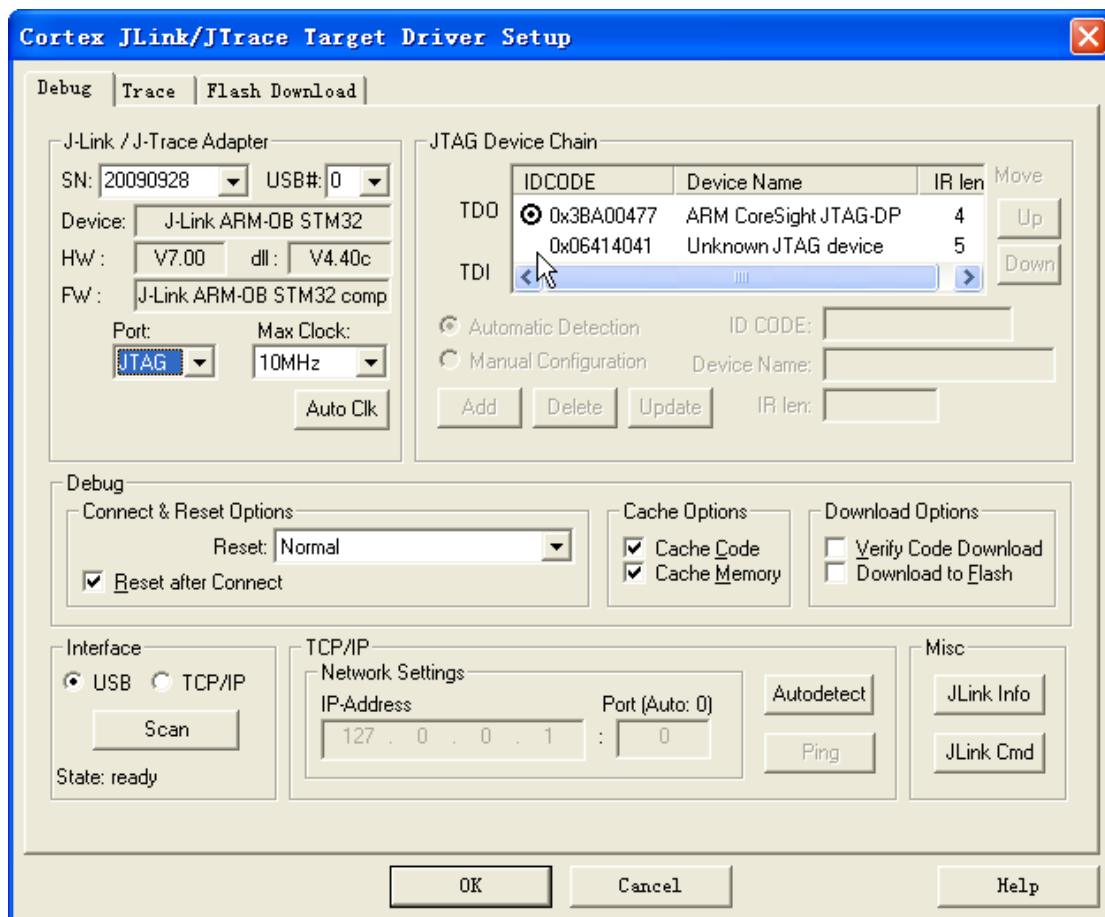
Download Options
☐ Verify Code Download
☐ Download to Flash

Interface
☒ USB ☐ TCP/IP
Scan
 State: ready

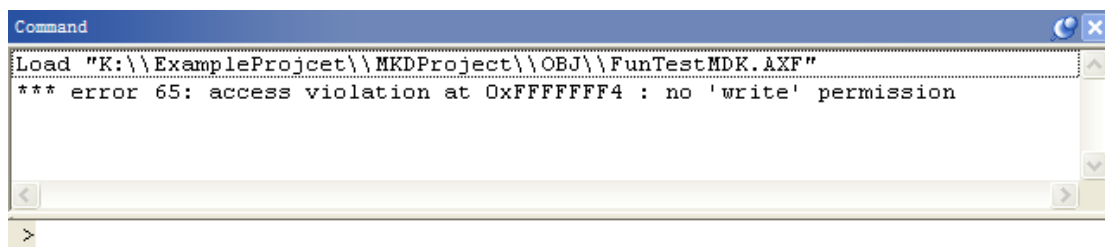
TCP/IP
 Network Settings
 IP-Address: Port (Auto: 0):
 127 . 0 . 0 . 1 : 0
Autodetect Ping

Misc
JLink Info JLink Cmd

OK Cancel Help

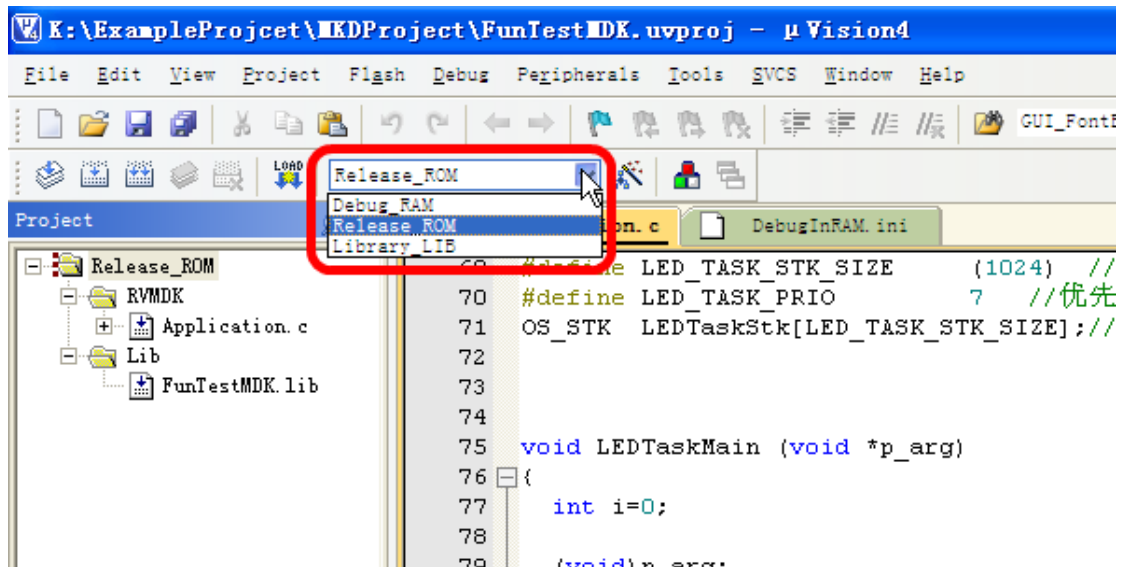


3. 点击调试按钮，无法调试，底部信息提示如下：



解决方法：

- 1、 在下拉菜单选择带 RAM 字样的工程，只有 ram 工程能够进行调试。ROM 工程不能调试，LIB 工程生成文件为库文件。



四、硬件篇

手持机硬件开发说明

1、内部扩展口说明

CL 系列手持机具有很高的开放性，用户可以轻松将自己的功能模块加入的到手持机中，形成自己的手持机。

手持机采用 STM32F103Z 系列 MCU 作为主控，通过双排 2.54mm 排针扩展了常用的硬件接口，如 UART，IIC，SPI，ADC，DAC，PWM，IO，JTAG，CAN，SDIO 等。

板上左侧扩展口 P111 资源：

STM32F103Z 管脚定义

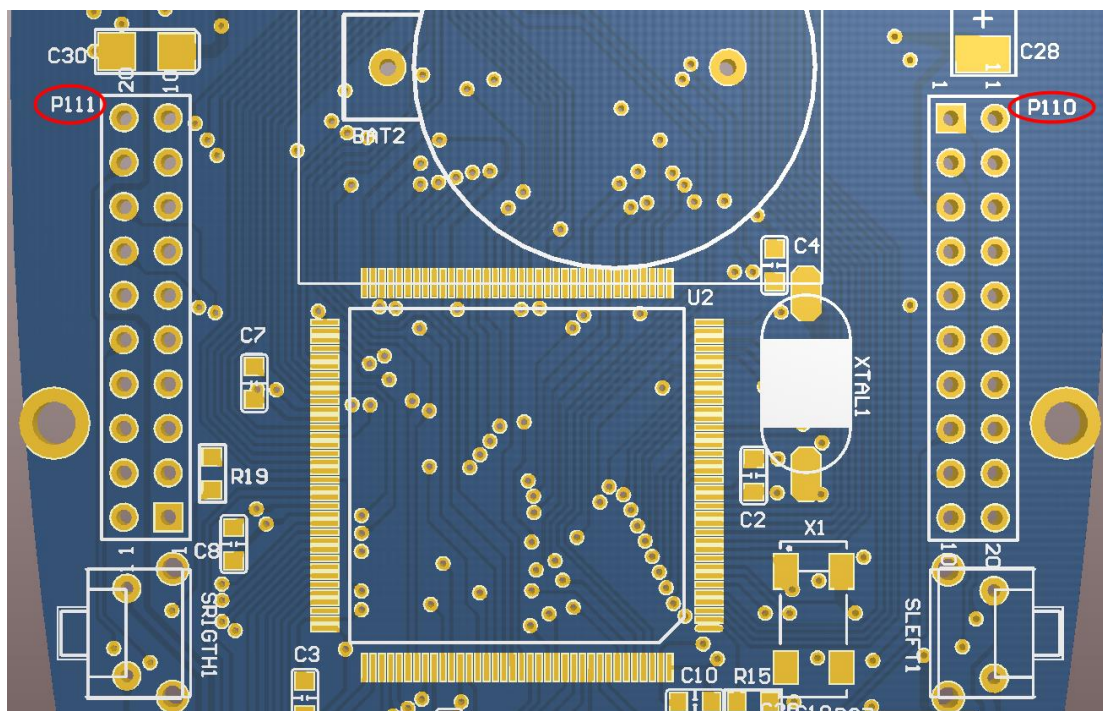
扩展口排针顺序	STM32F103Z 管脚定义						
	Pins	Pin name	Type	I/ O Level	Main function (after reset)	Alternate functions	
						Default	Remap
1		CPUvcc	Cpu 的电源，3.3V，关机后电源保持				
2	98	PC8	I/O	FT	PC8	TIM8_CH3/SDIO_D0	TIM3_CH3
3	99	PC9	I/O	FT	PC9	TIM8_CH4/SDIO_D1	TIM3_CH4
4	111	PC10	I/O	FT	PC10	UART4_TX/SDIO_D2	USART3_TX
5	112	PC11	I/O	FT	PC11	UART4_RX/SDIO_D3	USART3_RX
6	116	PD2	I/O	FT	PD2	TIM3_ETR/UART5_RX SDIO_CMD	
7	113	PC12	I/O	FT	PC12	UART5_TX/SDIO_CK	USART3_CK
8	49	PF11	I/O		PF11	FSMC_NIOS16	
9	122	PD6	I/O	FT	PD6	FSMC_NWAIT	USART2_RX

10	110	PA15/JTD I	I/O	FT	JTDI	PA15/SPI3_NSS/ I2S3_WS	TIM2_CH1_ETR SPI1_NSS
11		VCC	外设电源，3.3V，关机后无输出				
12		GND	系统电源地				
13	96	PC6	I/O	FT	PC6	I2S2_MCK/ TIM8_CH1/SDIO_D6	TIM3_CH1
14	97	PC7	I/O	FT	PC7	I2S3_MCK/ TIM8_CH2/SDIO_D7	TIM3_CH2
15	100	PA8	I/O	FT	PA8	USART1_CK/ TIM1_CH1(6)/MCO	
16	135	PB5	I/O		PB5	I2C1_SMBAL/ SPI3_MOSI/I2S3_SD	TIM3_CH2 / SPI1_MOSI
17	41	PA5	I/O		PA5	SPI1_SCK(6) DAC_OUT2 ADC12_IN5	
18	139	PB8	I/O	FT	PB8	TIM4_CH3(6)/SDIO_D4	I2C1_SCL/ CANRX
19	134	PB4/JNT RST	I/O	FT	JNTRST	PB4/SPI3_MISO	TIM3_CH1 / SPI1_MISO
20	133	PB3/JTD O	I/O	FT	JTDO	PB3/TRACESWO JTDO SPI3_SCK/I2S3_CK/	TIM2_CH2 / SPI1_SCK

右侧扩展口 P110 定义：

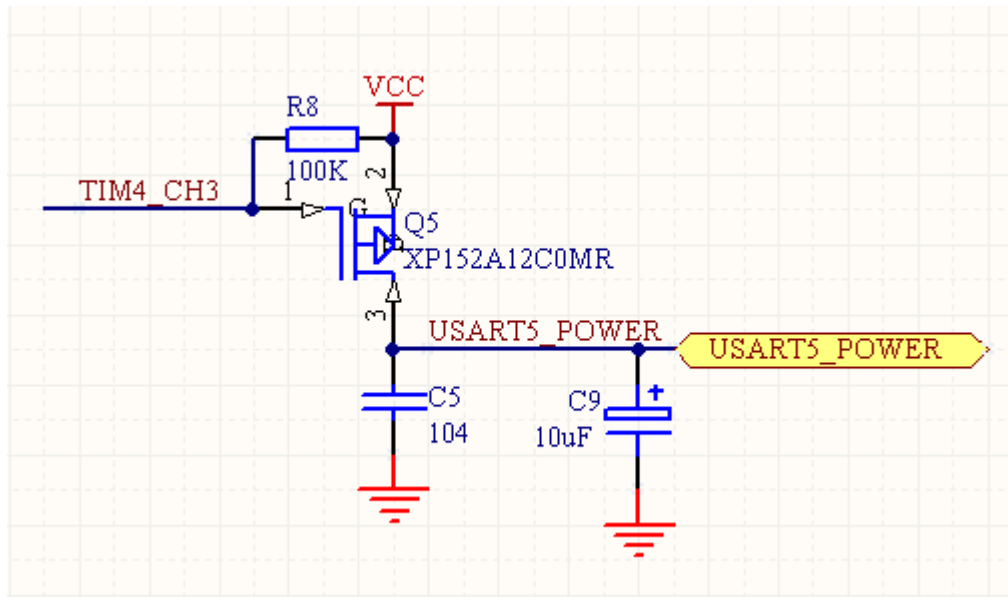
扩展口 排针 顺序	STM32F103Z 管脚定义						
	Pins	Pin name	Type	I/ O Lev el	Main function (after reset)	Alternate functions	
						Default	Remap
1							
2							
3		VCC	外设电源，3.3V，关机后无输出				
4	36	PA2	I/O		PA2	USART2_TX(6)/ TIM5_CH3/ADC123_IN 2/ TIM2_CH3 (6)	
5	69	PB10	I/O	FT	PB10	I2C2_SCL USART3_TX(6)	TIM2_CH3
6	73	PB12	I/O	FT	PB12	SPI2_NSS/I2S2_WS/ I2C2_SMBAL/ USART3_CK(6)/ TIM1_BKIN(6)	
7	75	PB14	I/O	FT	PB14	SPI2_MISO/TIM1_CH2 N USART3_RTS(6)	
8	137	PB7	I/O	FT	PB7	I2C1_SDA(6)/	USART1_RX

						FSMC_NADV/ TIM4_CH2(6)	
9	136	PB6	I/O	FT	PB6	I2C1_SCL(6)/ TIM4_CH1(6)	USART1_TX
10		RTCvcc	内存及时钟后备电池，内部通过二极管防止反冲				
11							
12							
13		GND	系统电源地				
14	37	PA3	I/O		PA3	USART2_RX(6)/ TIM5_CH4/ADC123_IN 3 TIM2_CH4(6)	
15	70	PB11	I/O	FT	PB11	I2C2_SDA USART3_RX	TIM2_CH4
16	74	PB13	I/O	FT	PB13	SPI2_SCK/I2S2_CK USART3_CTS(6)/ TIM1_CH1N	
17	76	PB15	I/O	FT	PB15	SPI2_MOSI/I2S2_SD TIM1_CH3N	
18	102	PA10	I/O	FT	PA10	USART1_RX(6)/ TIM1_CH3(6)	
19	101	PA9	I/O	FT	PA9	USART1_TX(6)/ TIM1_CH2(6)	
20		BATTER	电池供电电源				

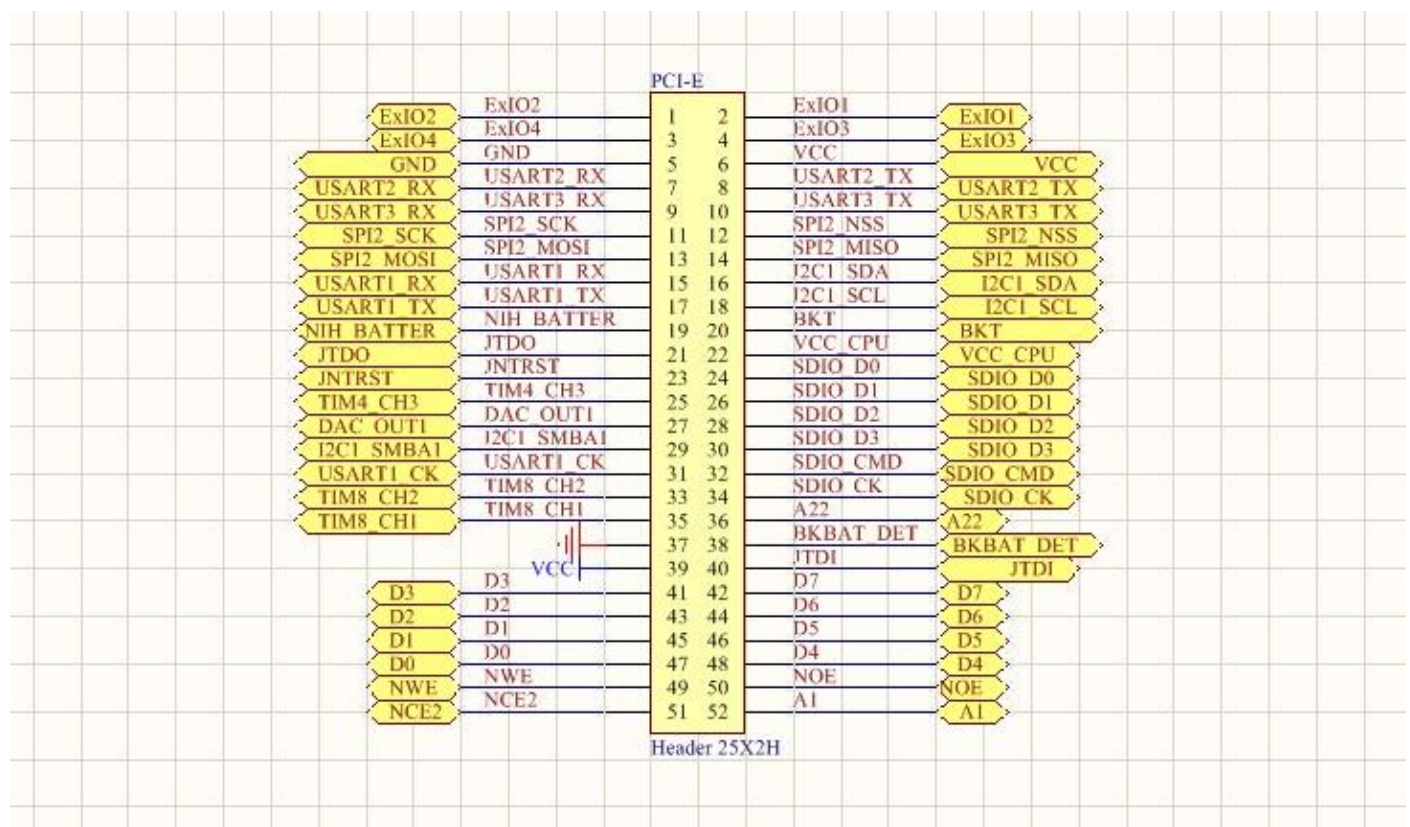


电源控制图:

为了控制外部器件的电源，可以使用一下的电路。在程序不使用外设时，可以控制 MOS 管的 G 极关闭电源输出，使功耗降低。



2、PCI-E 扩展口说明



3、HDMI 扩展口说明



五、问答篇

1、何谓二次开发：

二次开发，简单的说就是在现有的软件上进行定制修改，功能的扩展，然后达到自己想要的功能，一般来说都不会改变原有系统的内核。我们提供了编译器和建好的工程，只要用编译器打开工程，然后在主函数 `int main(void){}` 里面添加你需要实现的功能即可。我们提供的软件功能包括文件系统，DBF 数据库，通信管理，显示，键盘，输入，条码，RFID，WIFI，GPRS 等，你只要参照相应的文档和例程进行调用即可，极大的方便开发，缩短开发周期。例如条码功能：



2、DBF 数据库：

一种特殊的文件格式！表示数据库文件，Foxbase，Dbase，Visual FoxPro,等数据库处理系统所产生的数据库文件！也可被 execl，wps 等软打开编辑。

Chaobiao - Microsoft Visual FoxPro

文件(F) 编辑(E) 显示(V) 工具(T) 程序(P) 表(A) 窗口(W) 帮助(H)

用户表号	电表编号	用户姓名	用户地址	用电类别	倍率	上月度数	本月度数	上
20091527	5205180	汤维辉	朝阳南路	生活用水	1	955.0	0.0	0
20091528	00	邓佑琼	朝阳南路	生活用水	1	67.0	0.0	0
20091529	111111	雷向飞	朝阳南路	生活用水	1	623.0	0.0	0
20091530	111111	王良于	朝阳南路	生活用水	1	306.0	0.0	0
20091531	113673	章奎	朝阳南路	生活用水	1	392.0	0.0	0
20091532	195732	李姓	朝阳南路	生活用水	1	377.0	0.0	0
20091533	111111	李兴会	朝阳南路	生活用水	1	360.0	0.0	0
20097769	383344	胡长江	朝阳南路	生活用水	1	0.0	0.0	0
20097774	385462	雷中波	朝阳南路	生活用水	1	0.0	0.0	0
20091534	111111	超市	朝阳南路	生活用水	1	326.0	0.0	0
20091535	178661	黄兴华	朝阳南路	生活用水	1	420.0	0.0	0
20097775	384527	兰秀明	朝阳南路	生活用水	1	0.0	0.0	0
20097776	383397	姜天生	朝阳南路	生活用水	1	0.0	0.0	0
20091536	193892	李兴平	朝阳南路	生活用水	1	659.0	0.0	0
20093868	110329	谢光耀	朝阳南路	生活用水	1	469.0	0.0	0
20091537	169388	李兴军	朝阳南路	生活用水	1	121.0	0.0	0
20091538	124794	殷优明	朝阳南路	生活用水	1	122.0	0.0	0
20091539	150625	韩中林	朝阳南路	生活用水	1	649.0	0.0	0
20091540	152287	黄吉海	朝阳南路	生活用水	1	401.0	0.0	0
20091541	106137	王贤亮	朝阳南路	生活用水	1	1370.0	0.0	0
20091542	147801	代宜才	朝阳南路	生活用水	1	186.0	0.0	0
20099085	172676	胡林	朝阳南路	生活用水	1	35.0	0.0	0
20091543	130554	曾祥碧	朝阳南路	生活用水	1	760.0	0.0	0
20091544	140863	刘登文	朝阳南路	生活用水	1	340.0	0.0	0
20091545	141212	刘兴禄	朝阳南路	生活用水	1	308.0	0.0	0
20091546	144304	范治华	朝阳南路	生活用水	1	234.0	0.0	0
20091547	105823	孔凡泽	朝阳南路	生活用水	1	226.0	0.0	0
20091548	106178	戴光祥	朝阳南路	生活用水	1	297.0	0.0	0
20091549	150580	欧祖文	朝阳南路	生活用水	1	780.0	0.0	0

Chaobiao (d:_application\industryapply\通用抄表抄表程序\chs\记录:1/668 Exclusive NUM

3、开发需要什么知识

一般而言，开发人员只要掌握编程需要的 C 语言知识就可以，然后根据相关文档和例程实行功能。如果具备一些单片机知识，硬件开发知识，就可以自己进行硬件定制开发，实行具体的项目需要的功能，如增加特殊的无线通讯功能。