

WZ203CS-V4.0 开发板 support 文档

文档版本：V1.6

联系人：郝经理

联系电话：18913534803

QQ：1925334941

发布日期：2018.07.12

版权声明:

版权所有©苏州煜璘电子科技有限公司 2017。保留一切权利。

非经本公司书面许可,任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部,并不得以任何形式传播。

注意:

由于产品版本升级或其他原因,本文档内容会不定期进行更新。除非另有约定,本文档仅作为使用指导,本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

版本记录:

版本记录:

版本号	作者	版本说明	日期
V1.1	yangqiang	修正版	2017.10.13
V1.2	yangqiang	更新版	2017.11.01
V1.3	yangqiang	更新版	2018.02.22
V1.4	yangqiang	更新版	2018.04.09
V1.5	yangqiang	更新版	2018.05.08
V1.6	yangqiang	更新版	2018.05.15
V1.7	yangqiang	更新版	2018.07.12

版本信息:

- V_1.1: 初始版本
- V_1.2: WZ203CS+电池+串口模块版本
- V_1.3: WZ203CS-V3.0 版本
- V_1.4: 增加部分介绍
- V_1.5: 增加 mqtt 版本写 SN 号的方法
- V_1.6: 增加环境搭建
- V_1.7: 4.0 开发板适配

注释: 以往版本参考介绍请查看 WZ203CS-FAQ 文档!

目录

- 第一章. 介绍.....4
- 第二章. 使用方法.....5
- 第三章. 烧写方法.....8
- 第四章 . 环境搭建.....9
- 第五章. stm8 烧写方法..... 18
- 第六章. 服务与支持.....20
 - 6.1 产品咨询.....20
 - 6.2 产品保修服务.....20
 - 6.3 技术支持.....21

第一章. 介绍

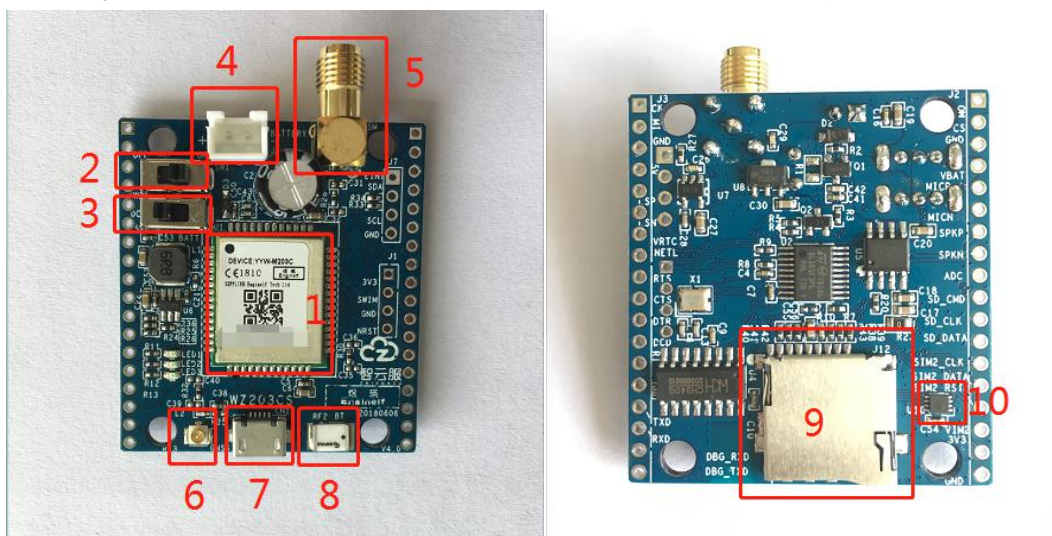
WZ203CS-V3.0 是一款具有低功耗、体积小、定位快等功能的一款优秀开发板。本文将帮助你了解 WZ203CS-V3.0 开发板的使用方法和烧写方法。

WZ203CS-V3.0 开发板采用了我司 YYW_M_203C 模块。

网盘资料：链接：<http://pan.baidu.com/s/1c14zZzE> 密码：qsyd

WZ203CS 技术交流群：657996991

对照发货清单核对物品，然后按照图片中的连接方式连接好天线和 sim 卡。



- 1: YYW_M203C 模块
- 2: 开发板电源开关，左边 OFF，右边 ON
- 3: 电源选择开关，左边为 DC5V 供电即单独使用 USB 供电，右边为锂电池供电。拨到左边不能插上锂电池（这时候电池会被充电，电池分走部分电流，但是电池没有为开发板供电，可能会导致 USB 供电不足），拨到右边可以插上锂电池并且插上 USB 为锂电池充电。
- 4: 锂电池插口
- 5: GSM 天线，现在开发板配套的是板载天线，则不需要插此天线。
- 6: GNSS 天线
- 7: USB 插口
- 8: 蓝牙天线（贴片式）
- 9: 此图中为 SD 卡槽，另外一种为 SIM 卡槽
- 10: 2*2 的贴片 SIM 卡，则如果是 SIM 卡槽版本，此处若焊了贴片 SIM 卡，SIM 卡槽无需再插一张 SIM 卡。SD 卡槽版本此处必须焊有此贴片 SIM 卡。

第二章. 使用方法

开发板发到每个使用者手里时是烧写好的，只需要写一下 SN 号，就能使用，下面介绍如何对开发板进行写 SN 号的操作。这里涉及两个版本的问题，开发板的 SDK 包有几个工程，可以通过配置 Makefile 进行编译成不同的版本。前不久我司在 mqtt-ble 版本上实现了通过扫描二维码绑定 SN 的功能。至于如何编译成不同的版本请大家观看智云服视频教程，里面不仅有教大家如何编译，更有如何使用 SDK 进行简单快速的开发。具体的烧写包可以从 QQ 群公告进入 WZ203CS 开发板资料下载界面，选择下载。

对于 TLV 软件版本必须使用下面方式对开发板写一下 SN 号：

拿到 WZ203CS 开发板后，SN 号是需要手动通过串口写一下的，使用丁丁串口调试助手。用 USB 线连接好电脑，打开串口调试助手，**波特率调至 115200，取消发送新行**，然后会有 log 打印信息出现。发送命令 `$$setsn xxxxxxxx` (xxxxxxx 代表模块贴纸上的 SN 号，**贴纸上的 0x 不算入 SN 号里面**) 比如下图。如果是设置错误提示已经设置 SN 号，请使用 `^^setsn xxxxxxxxxx`。



对于 MQTT 软件版本上面对于 TLV 格式的可以使用，而且有另外一种更简单的方法：

首先给没有写 SN 号的开发板上电开机，这个时候通过串口可以看到有 log 提示设置 SN 号。这个时候可以打开微信小程序，然后点击下方的绑定 SN 号扫描开发板模块上的标签的二位码就可以自动写入 SN 号了。操作完成之后可以通过串口信息查看是否写入。



然后保证插上 SIM 卡（不支持电信）和天线，插好 USB 给开发板稳定供电后，然后把上面开关拨到 ON，下面开关拨到对应位置（查看第一章介绍），等待一会就可以自动联网。联网之后板子会自发进行 GPS 定位。如果联网失败请查看卡是否插牢，GSM 天线是否插好。如果 GPS 定位时间过长，请确定测试环境信号强弱。通常办公室里的信号较弱无法定位，请移至空旷处测试。

如果这些均正常工作。则可以通过微信查看相关数据。第一次使用，必须要使用标签上的 SN 号并且联网成功，这样才能把这个没用过的 SN 号激活并注册到智云服产品下面去。然后可以打开手机微信，扫描下方二维码，输入 SN 号或者扫描标签上的二维码进入智云服小程序查询信息。



设备编号不正确原因：

扫描提示设备编号不正确的原因：说明模块上贴的 SN 号是没有用过的，需要烧写 3.0 开发包的程序，然后给开发板供电，写上贴纸上的 SN 号，千万不能写错，注意写的过程中要取消发送新行，不然会多加一个换行依然会报错。写 SN 号的命令下一章有写。写好 SN 号之后，重新开关机，等待开发板联网，采集数据上报数据。之后就可以使用微信小程序扫描二维码或者直接输入 SN 号即可查询。

小程序传输间隔介绍：

关于传输间隔，设置大于 300s，开发板处于可休眠状态，程序在联网成功之后会立即采集数据，采集到了就会立即上报，若一段时间没采集到 GPS 信号，上报到服务器的数据就不含 GPS 数据。休眠时间为设置的时间。设置小于 300s，开发板为不休眠状态，并且每隔设置的间隔时间会扫描一次上报一次。

开发板上的 LED:

刚通电三个灯全亮，休眠三个灯全灭。

LED1: 闪烁代表模块正常工作，快速闪烁代表发送数据；

LED2: 小程序的 LED 灯可以控制此灯的亮灭；

LED3: 亮表示 GPS 定位，LBS 定位不会亮；

灯亮的逻辑是由软件配置的。

第三章. 烧写方法

收到的开发板是已经烧好程序的，并且基本配置信息已经设置好了，所以如果不是特别需要可以不用重新烧写。需要烧写的使用者可以自行去网盘下载对应的烧写包，MQTT 版本比 TLV 版本多了小程序绑定 SN 号的功能，然后协议用的是 MQTT 协议，其它的没有什么不同，TLV 版本使用的就是 TLV 协议。

烧写方法：

烧写底包：先把上面开关拨到右侧让板子断电，然后插上 USB 线，打开 QFlash_V4.0, 点击 loadFWFiles 选择 3.0 开发包里的底包+ATCMD 中的文件。选好串口，点击 START，开关拨到左侧上电，即可烧写。

注：底包目前有两个，一个是支持 SD 卡的底包，但是不支持蓝牙，烧写支持 SD 卡的底包需要烧写对应的 APP 包；另一个支持蓝牙，但是不支持 SD 卡。支持蓝牙不支持 SD 卡的包也可以烧写到 SD 卡版本的板子，只是无法使用 SD 卡的功能，其它功能都是可以的。

烧写 APP：先把上面开关拨到右侧让板子断电，然后插上 USB 线，打开 QFlash_V4.0, 点击 loadFWFiles 选择 3.0 开发包里的 build/gcc 下的文件。Module type 选择 M66 或以下。点击 START，开关拨回左侧上电，即可烧写。

注：如果无法烧写，请按照上述步骤多试几次。

第四章. 环境搭建

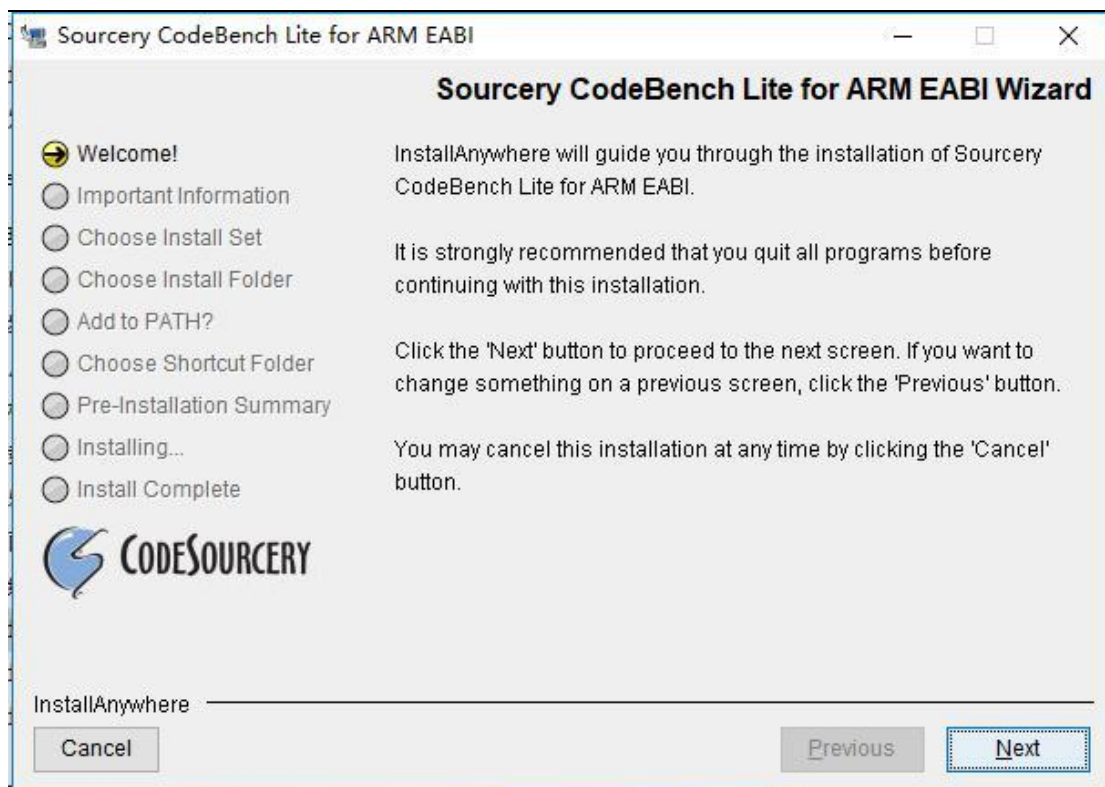
1、先找到这连个文件 arm-2011.09-69-arm-none-eabi.exe, 如图 1 所示。arm-2011.09-69-arm-none-eabi.exe 为需要安装编译环境。



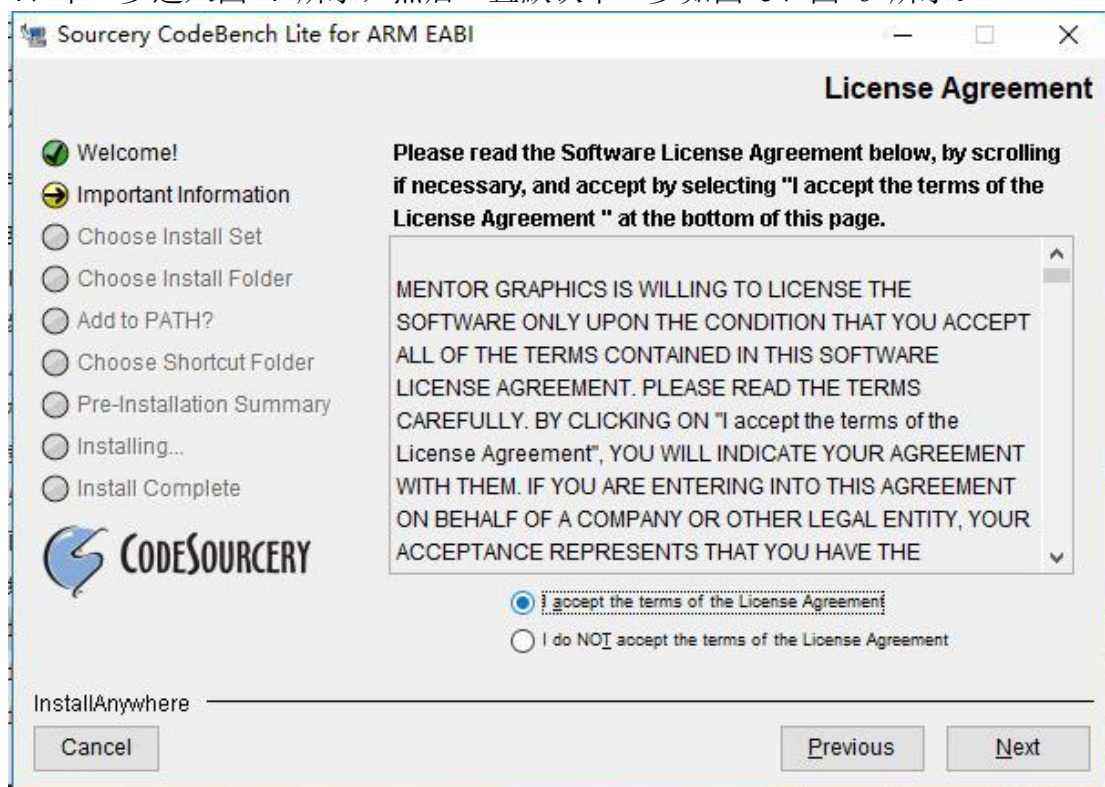
2、设置 arm-2011.09-69-arm-none-eabi.exe 兼容性，兼容 win7 系统，如图 2 所示。

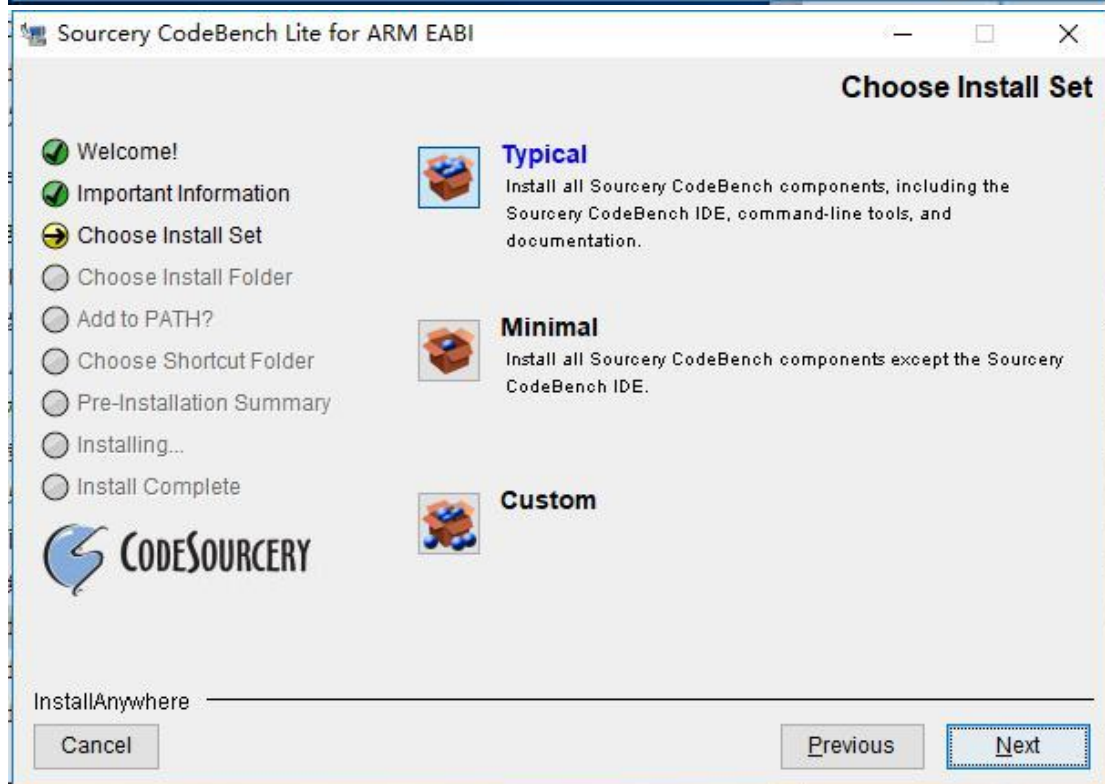
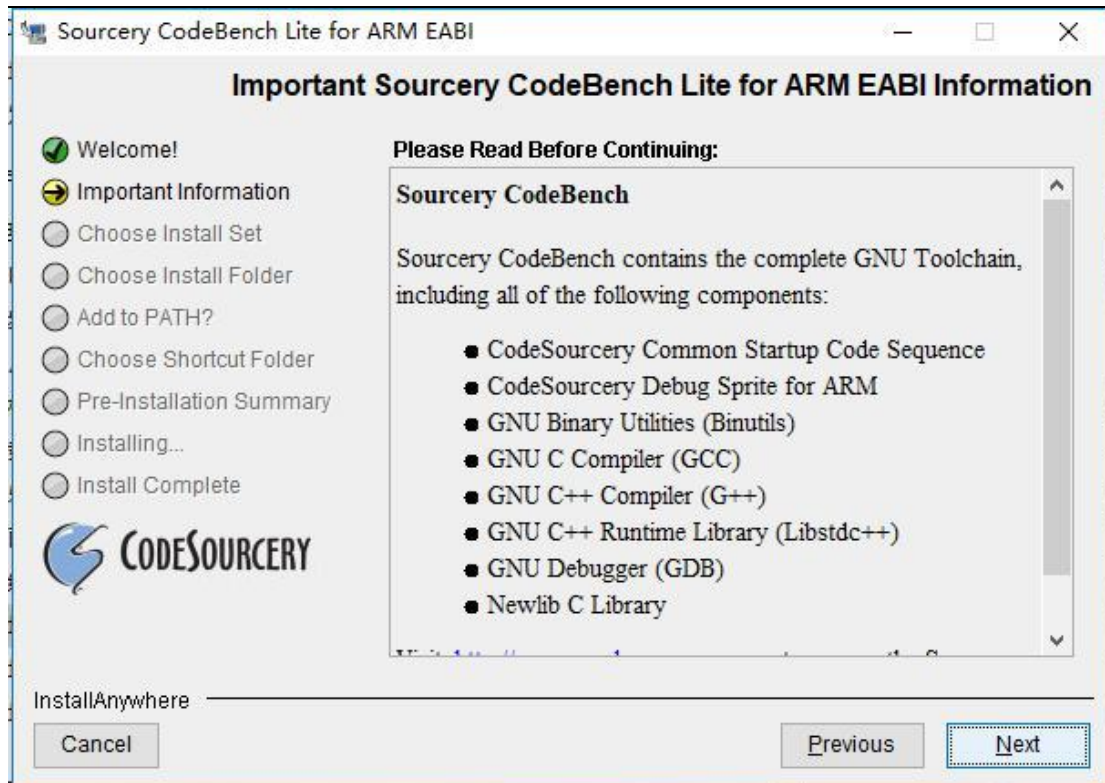


3、打开 exe 文件开始安装，如图 3 所示为安装第一步。

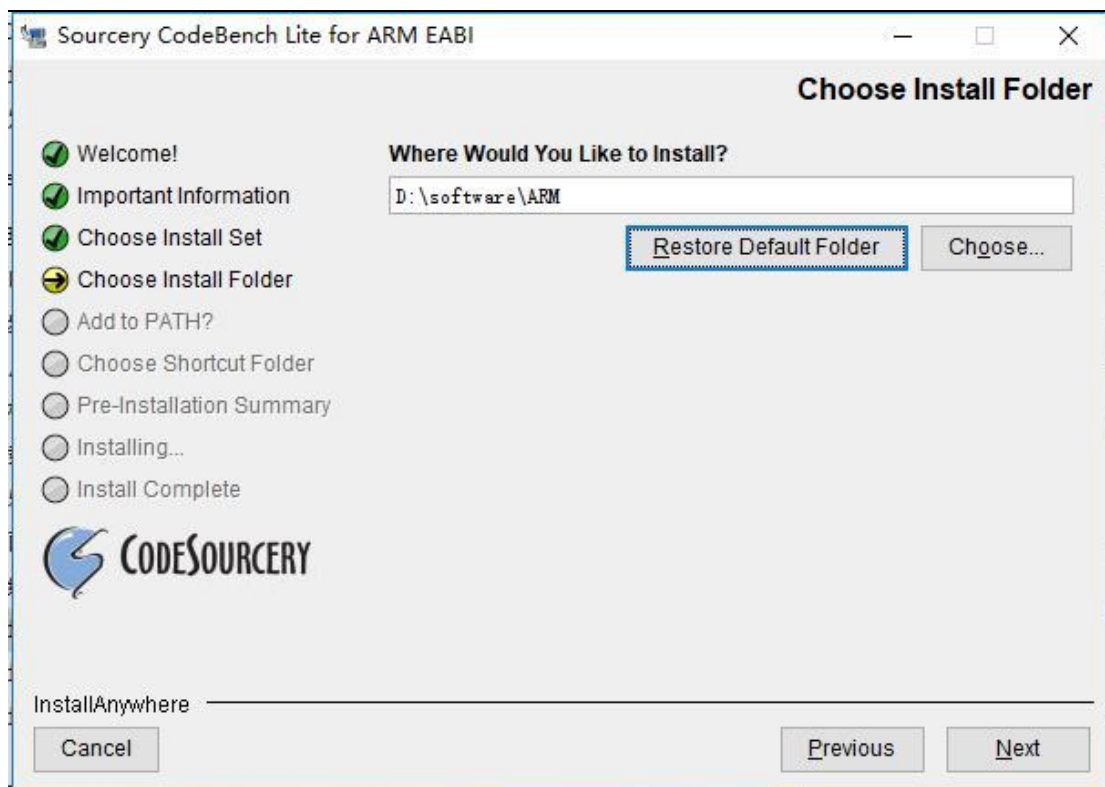


4、下一步进入图 4 所示，然后一直默认下一步如图 5、图 6 所示。

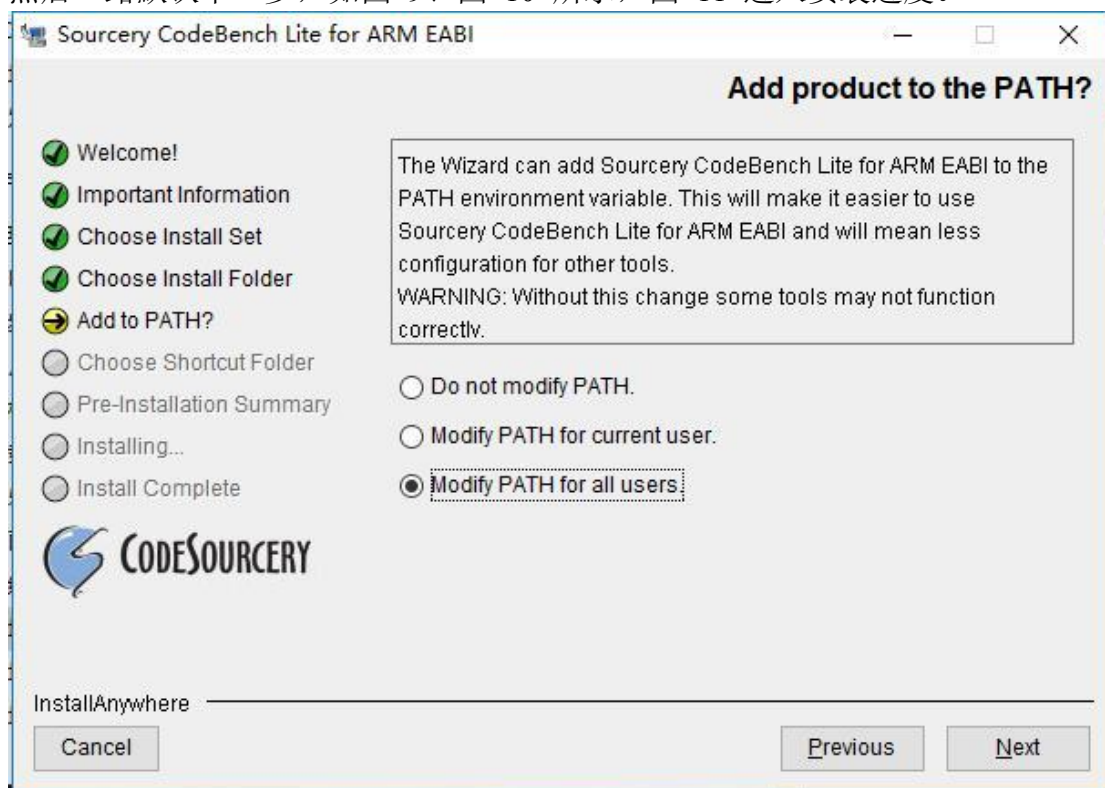


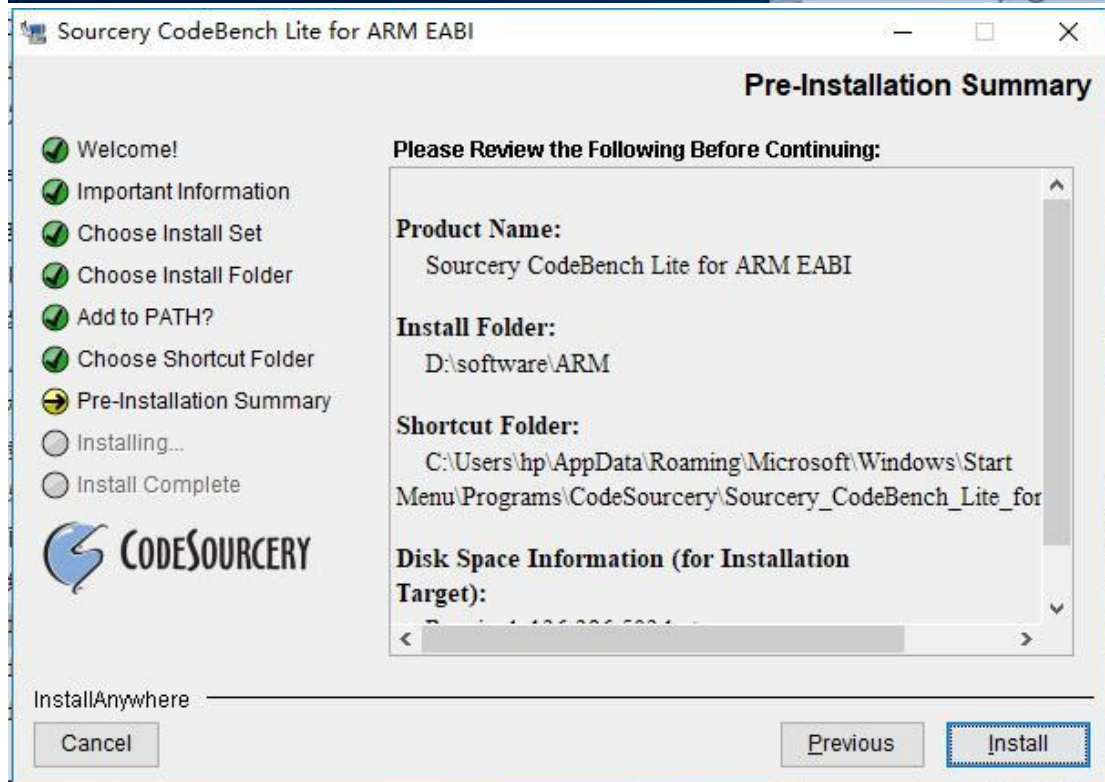
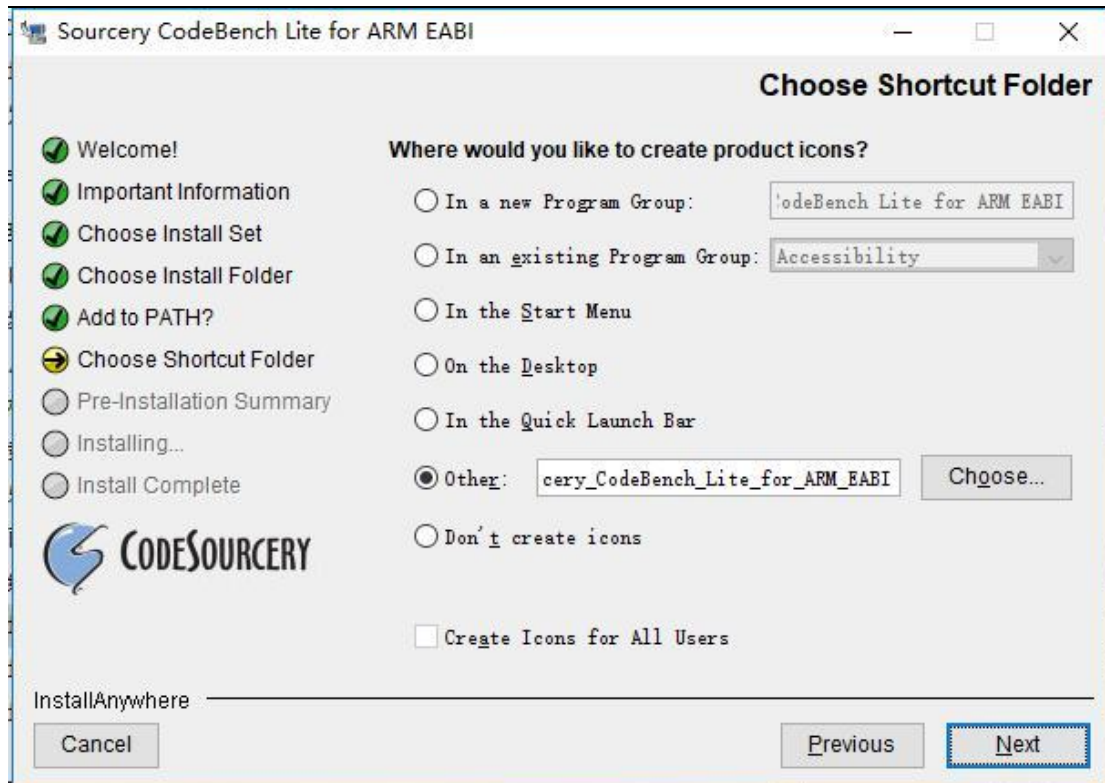


5、选择安装了路径，如图 7 所示，此时路径不要随便选择，记住安装路径。此时我安装的路径为 D:\software\ARM。



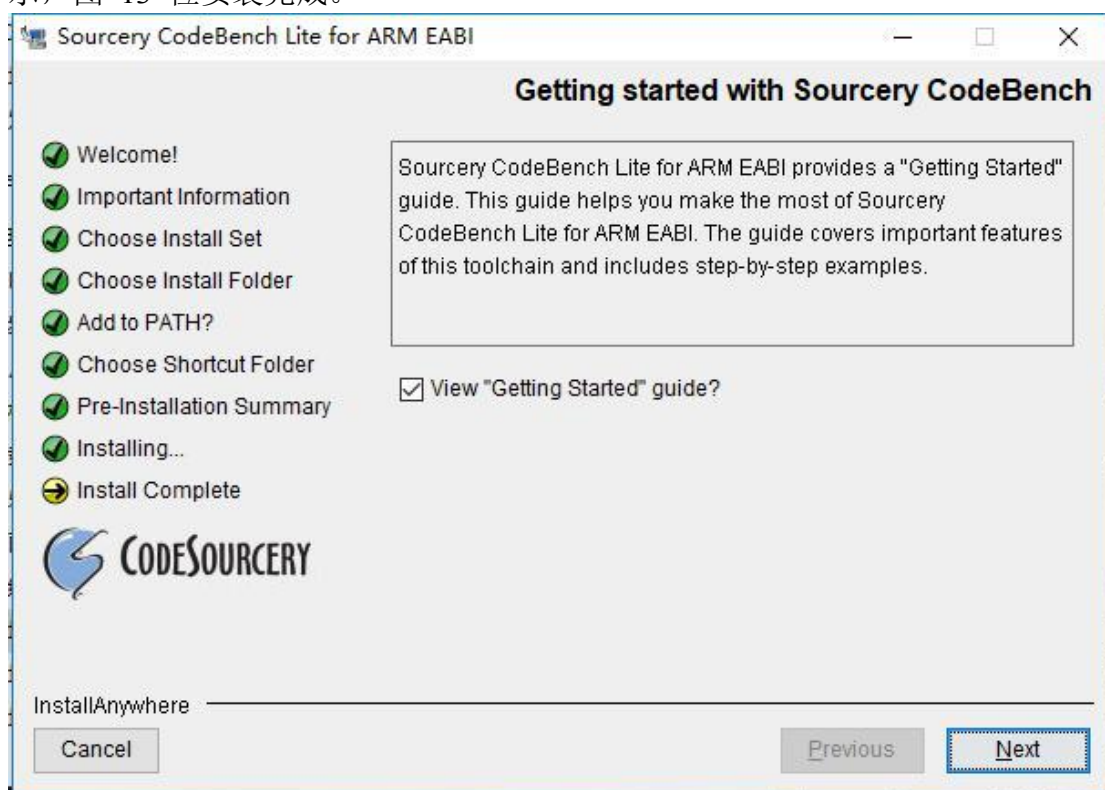
6、选择 Modify PATH for all users 选项，继续下一步，如图 8 所示，然后一路默认下一步，如图 9、图 10 所示，图 11 进入安装进度。

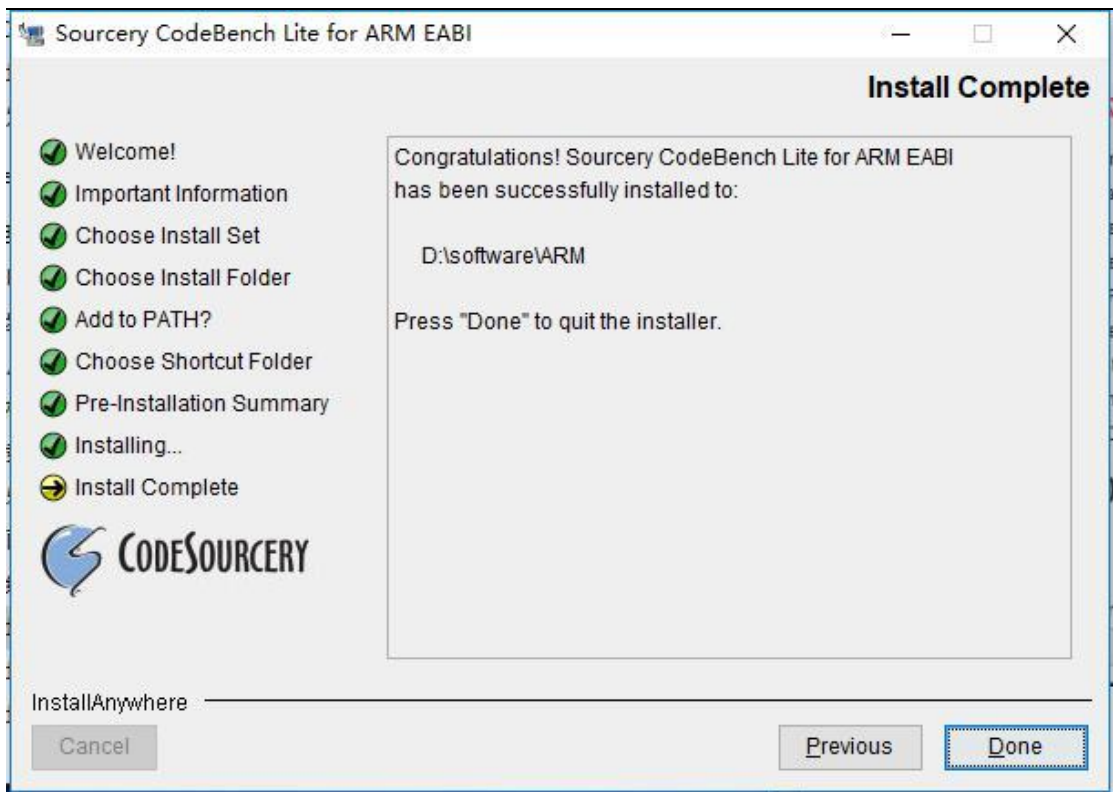






7、此时可以选择够此处选项，也可以不选，没有什么影响如图 12 所示，图 13 位安装完成。

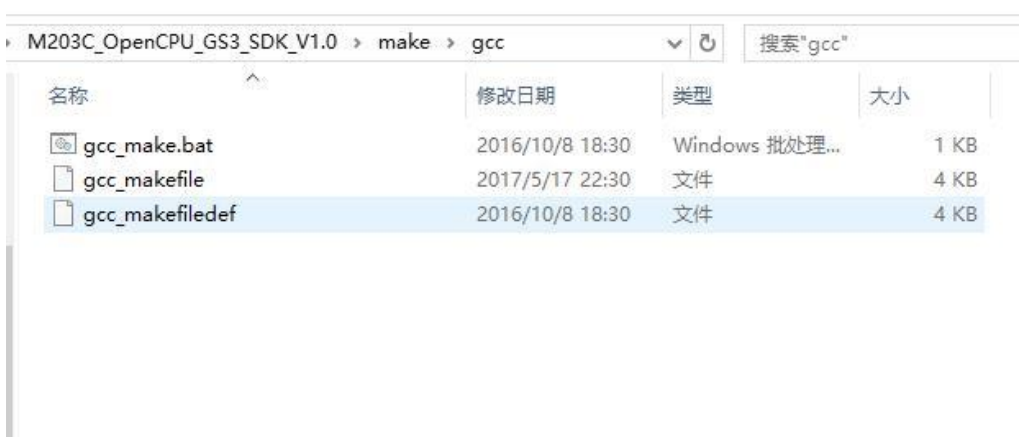




8、到前边七步只是安装完成，还需要配置相关文件。在电脑查找 cmd,进入命令编辑界面，输入命令 `arm-none-eabi-gcc -v`，查看安装版本，结果如图 14，版本号为：4.6.1。

```
/obj/host-libs-2011.09-69-arm-none-eabi-i686-mingw32/usr --wi
ynamic -lm' --with-cloog=/scratch/jwlenke/2011.09-arm-eabi-li
--with-libelf=/scratch/jwlenke/2011.09-arm-eabi-lite/obj/host-
ibgomp --enable-poison-system-directories --with-build-time-to
pc-linux-gnu-2011.09-69-arm-none-eabi-i686-mingw32/arm-none-ea
m-eabi-lite/obj/tools-i686-pc-linux-gnu-2011.09-69-arm-none-ea
Thread model: single
gcc version 4.6.1 (Sourcery CodeBench Lite 2011.09-69)
C:\Users\bn>
```

9、然后进入第 1 步时候的目录 `m203c3.0_sdk_vport_v1.1`（具体名称请按照最新的 SDK 名称为主），进入目录 `m203c3.0_sdk_vport_v1.1\make\gcc`，如图 15 所示。



10、打开文件 gcc_makefile 进行编辑，用记事本或者文本文档的方式就行，第 18 行和 19 行如图 16 所示.

```
5 # Configure GCC installation path, and GCC version.
6 # To execute "arm-none-eabi-gcc -v" in command line can get the current gcc version.
7 #
8 -----
9 GCC_INSTALL_PATH=C:\Program Files (x86)\CodeSourcery\Sourcery_CodeBench_Lite_for_ARM_EABI
10 GCC_VERSION=4.6.1
11
12 C_PREDEF=-D __EXAMPLE_MULTITASK__
13
```

11、18 行安装路径改成我的安装路径 D:\software\ARM，即第 5 步记录的路径；第 19 行版本号改成开始时候得到的版本号 4.6.1，即第 8 步得到的版本号。改完如图 17 所示，然后保存退出。

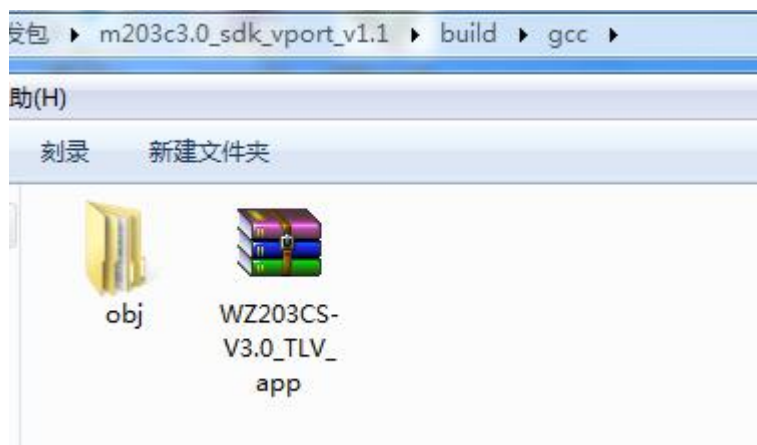
```
18 GCC_INSTALL_PATH=D:\software\ARM
19 GCC_VERSION=4.6.1
```

12、最后进入路径 m203c3.0_sdk_vport_v1.1，即文件源码存放路径，打开 MS-DOS，如图 18 所示 .

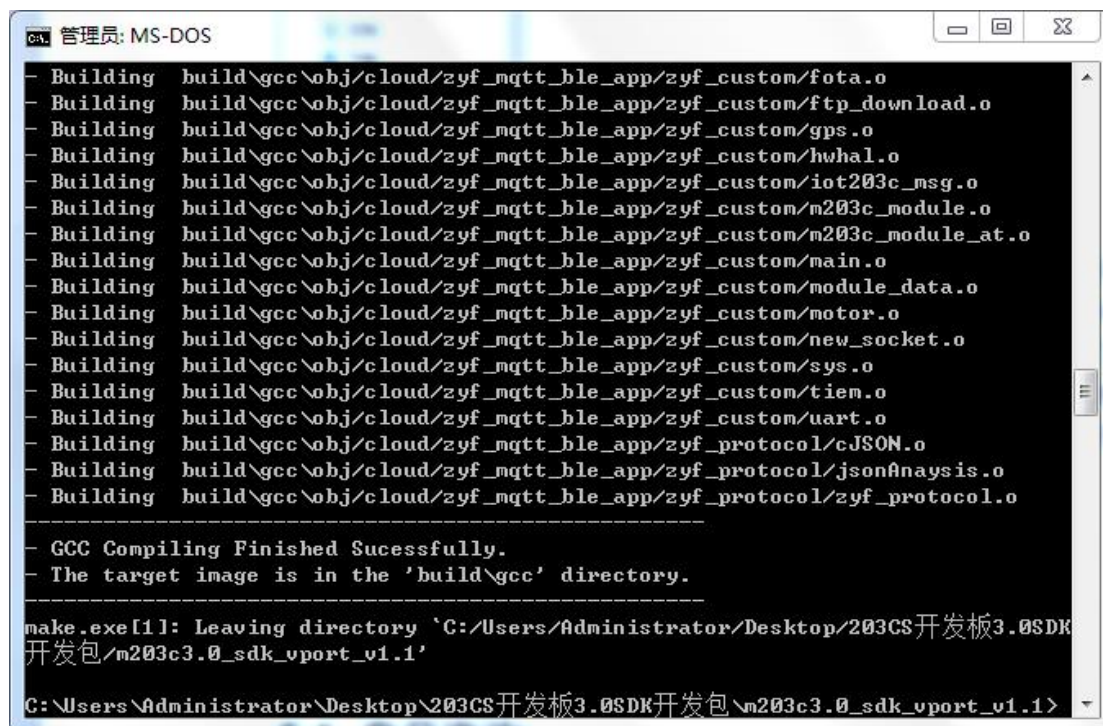


13、因为我们上一步修改了 Makefile，现在需要重新编译，先输入命令 make clean，把以前生成文件删除，如图 19 文件删除成功，此时查看路径 m203c3.0_sdk_vport_v1.1\build\gcc，文件显示如图 20 所示。

```
C:\Users\Administrator\Desktop\203CS开发板3.0SDK开发包\m203c3.0_sdk_vport_v1.1>make clean
-----
clean finished.
-----
C:\Users\Administrator\Desktop\203CS开发板3.0SDK开发包\m203c3.0_sdk_vport_v1.1>
```



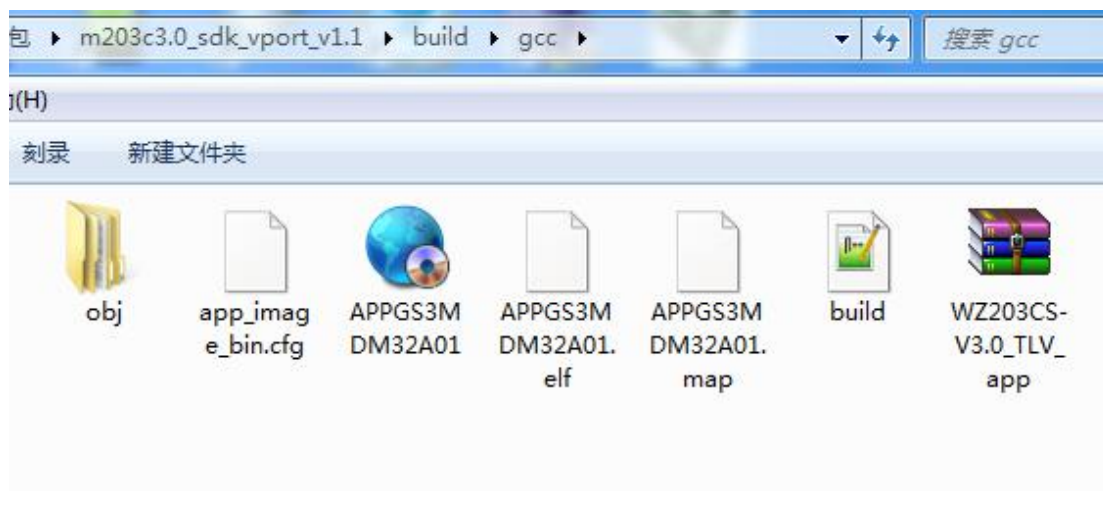
14、重新编译，输入命令 make new，编译成功如图 21 所示。
此时重新查看路径 D:\ARM\ZF-203C_OpenCPU_GS3_SDK_V1.0\build\gcc，发现多了几个文件，如图 22 所示。其中这三个文件 app_image_bin.cfg、APPGS3MDM32A01.bin 和 APPGS3MDM32A01.elf 是需要烧写的文件，build.log 是日志文件，编译过程的错误提示会保存在这个文件中。至此环境搭建成功。



```

C:\Users\Administrator\Desktop\203CS开发板3.0SDK开发包\m203c3.0_sdk_vport_v1.1>
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\fota.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\ftp_download.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\gps.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\hwhal.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\iot203c_msg.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\m203c_module.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\m203c_module_at.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\main.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\module_data.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\motor.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\new_socket.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\sys.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\tiem.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_custom\uart.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_protocol\cJSON.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_protocol\jsonAnalysis.o
Building build\gcc\obj\cloud\zyf_mqtt_ble_app\zyf_protocol\zyf_protocol.o

-----
GCC Compiling Finished Successfully.
The target image is in the 'build\gcc' directory.
-----
make.exe[1]: Leaving directory 'C:/Users/Administrator/Desktop/203CS开发板3.0SDK开发包/m203c3.0_sdk_vport_v1.1'
C:\Users\Administrator\Desktop\203CS开发板3.0SDK开发包\m203c3.0_sdk_vport_v1.1>
    
```



第五章. stm8 烧写方法

首先打开 STM8 烧写包，选择此烧写包的工程文件并打开（如图 3.1），会自动进入 IAR 工具的编程环境（如图 3.2）。然后插上 ST-LINK 烧写线，最后点击烧写图标进行烧写（如图 3.2），烧写过程中会有进度条显示，初次烧写的过程中需要选择烧写包里的 main.c，请自行选择（如图 3.3）。

- 注释：（1）IAR 工具的编程环境需要提前安装好
（链接：<https://pan.baidu.com/s/1sFi551q051CLiBT4Ave0-g> 密码：mqby）
（2）ST-LINK 烧写线的连接需注意 ST-LINK 烧写线与开发板的引脚的对应详见图 3.4
（3）发货的板子已经烧写好 STM8, 正常情况，不需要再次烧写。



图 3.1 打开相应的 stm8 工程

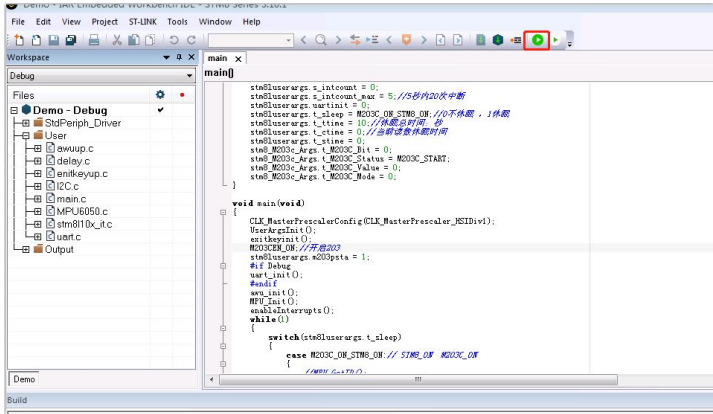


图 3.2 点击烧写图标

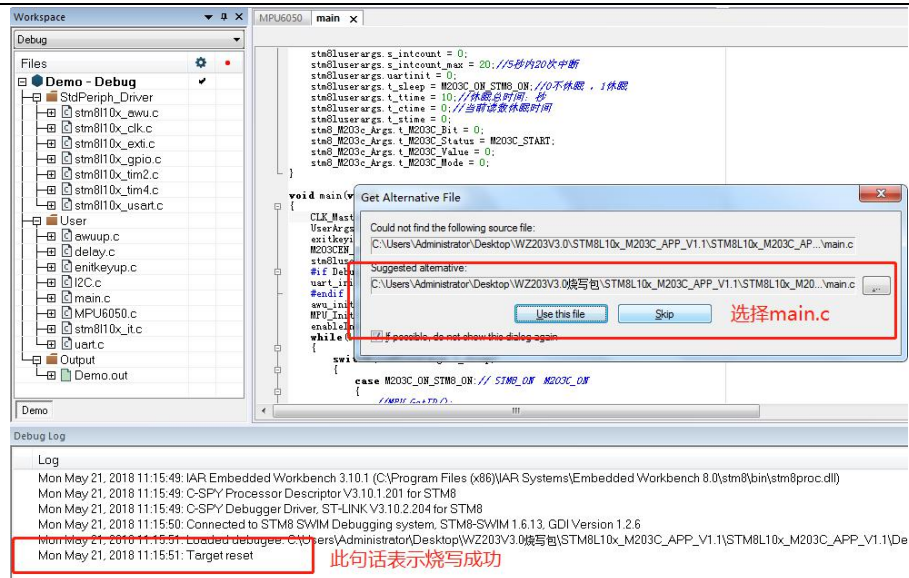


图 3.3 选择烧写文件

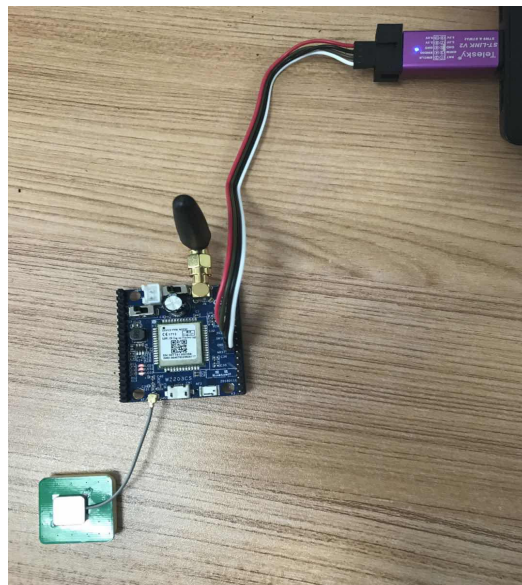


图 3.4 ST-LINK 烧写线的连接

第六章. 服务与支持

6.1 产品咨询

用户在选购时，可通过服务热线电话、淘宝阿里旺旺、Email 的联系方式直接联系我公司的销售人员和工程师，我们将以最大的热情为您解答售前的所有问题。

公司：苏州煜璘电子科技有限公司

地址：苏州市工业园区独墅湖高教区仁爱路 99 号 C501

电话：郝远琪，18913534803

淘宝：[首页-煜璘电子科技有限公司-淘宝网](#)

邮箱：sale@engineelf.com

QQ：1925334941

6.2 产品保修服务

凡是通过本公司购买的产品可享受以下权益：

- 6 个月免费保修服务
- 终身免费技术支持服务
- 终身维修服务
- 免费享有所购买产品配套的软件升级服务
- 免费享有所购买产品配套的源代码，以及本公司开发的部分开源代码
- 可直接从本公司购买主要的芯片样品，简单、方便、快速；免去从代理商处购买的漫长等待周期
- 自购买之日起成为本公司的永久客户，享有再次购买本公司任何一款产品的优惠政策。

•OEM/ODM 服务

如有以下情况之一，则不享有免费保修服务：

- 1.超过免费保修服务周期
- 2.无产品序列号或无产品有效的购买单据
- 3.进液、受潮、发霉或腐蚀
- 4.受撞击、挤压、摔落、刮伤等非产品本身质量问题引起的故障和损坏

5.擅自更改硬件、错误上电、错误操作引起的故障和损坏

6.由不可抗拒自然因素引起的故障和损坏

关于产品返修：

用户在使用过程中由于产品故障、损坏或其他异常现象，在寄回维修之前，请先致电本公司，与工程师进行沟通以确认问题，避免故障判断错误造成不必要的运费损失及周期的耽误。

维修周期：收到返修产品后，我们将即日安排工程师进行检测，我们将在最短的时间内维修或更换并寄回。一般的故障维修周期为 3 个工作日（自我司收到物品之日起，不计运输过程时间），由于特殊故障导致无法短期内维修的产品，我们会与用户另行沟通并确认维修周期。

维修费用：在免费保修期内的产品，由于产品质量问题引起的故障，不收任何维修费用；不属于免费保修范围内的故障或损坏，在检测确认问题后，我们将与客户沟通并确认维修费用，我们仅收取元器件材料费，不收取维修服务费；超过保修期限的产品，根据实际损坏的程度来确定收取的元器件材料费和维修服务费。

运输费用：产品正常保修时，用户寄回的运费由用户承担，维修后寄回给用户的费用由我司承担。非正常保修产品来回运费均有用户承担

6.3 技术支持

我们配备多名具有多年嵌入式研发经验的工程师来专职做技术支持，能够专业、快速的为您解决在产品使用过程中遇到的问题。

我们提供的技术支持范围仅局限于开发板的功能范围，我们将帮助您将我们提供的功能在产品上正常的运行起来，在这之外的其他技术问题，可相互之间进行交流，我们很乐意跟我们的用户分享我们在产品软、硬件研发过程中的点滴经验。