

Welink your smart



ME3616

WelinkOpen SDK 安装向导

版本: V1.1

日期: 2018-05-31

NB-IoT 模组



Website: www.gosuncnwelink.com

E-mail: welink@gosuncn.com

修订历史

版本	日期	描述
V1.0	2018-04-18	第一次发布
V1.1	2018-05-31	更新文档名称及文档模板 更新 example 下目录文件名称和功能

目录

修订历史.....	1
1. 概述	3
2. WELINKOPEN RTOS SDK 开发包安装	4
3. 编译器安装.....	6
4. 客户程序添加及编译	10
5. 系统烧录	13
6. 如何查看调试应用程序.....	16
6.1. 如何使用 GENIE TOOL 工具	16
6.2. 用户抓取 LOG	20

1. 概述

本文档介绍使用 windows7/windows10 环境下开发可以运行在 ME3616 模组上基于 OpenCPU 的应用程序。

WelinkOpen RTOS SDK 提供交叉编译链、所需的库文件和头文件、API 和 API 调用示例程序，这些 API 可以实现客户相应的需求。所有的内容将以 SDK 安装包的形式提供给客户，客户只需要按照使用向导解压 SDK 包（**注：SDK 的安装路径中不能包含中文和空格**），然后进行应用程序编写，编译完成后烧入模组开发板中即可调试使用。

WelinkOpen RTOS SDK 的开发环境位于 sdk 目录下，该目录下包含了交叉编译链、库文件、头文件和 API 等。examples 目录包含了 API 的调用示例程序，客户可以参考或直接拷贝所需的代码。

WelinkOpen 目前预留给客户可使用的 flash 及 sram 资源见下表，内存布局见下图。

项目	可用空间大小（字节）
FLASH	256K
SRAM	256K

客户APP内存布局



Preamble: 包含客户APP的一些可定制选项：

- 1、main task栈大小
- 2、callback task栈大小

图 1-1 APP 内存布局图

2. WELINKOPEN RTOS SDK 开发包安装

① 打开 WelinkOpen 提供的自解压 SDK 安装包，双击 welinkopen_rtos_sdk.v0.1.0 应用程序即可。

高新兴物联WelinkOpen_CPU 安装开发...	2018/4/12 10:23	Microsoft Word ...	2,441 KB
高新兴物联WelinkOpen_CPU API手册...	2018/4/11 17:37	Microsoft Word ...	1,024 KB
welinkopen_rtos_sdk.v0.1.0	2018/4/9 11:04	应用程序	80,649 KB

图 2-1 安装程序目录

② 点击 Extract 后，即可进行解压。

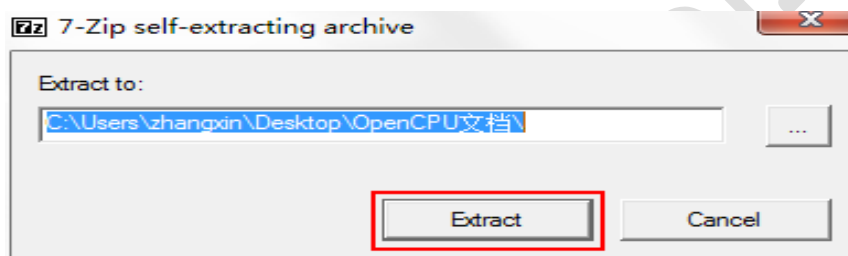


图 2-2 自解压

③ 解压后生成 sdk 目录。

进入 sdk 目录：

“examples” 目录是示例程序展示 API 的使用方法，客户可以参考或拷贝。

“FreeRTOS” 目录是 FreeRTOS 系统源码。

“include” 目录是 OpenCPU 的头文件。

“lib” 目录是 OpenCPU 提供的第三方库。

“lwip” 目录是 OpenCPU 提供的 lwip 相关源码。

“scripts” 目录是 OpenCPU 的编译脚本。

“tools” 目录是 Openlinux 项目的交叉编译环境及其他工具。

“examples” 下面的文件目录和功能见下表：

2-1 example 下目录名称及功能表

examples 下目录名称	目录功能
flash	flash 操作例程，包括 flash 的打开，擦除，读写等功能
Hello	最简单的用户程序，用于输出 Hello world

printf	printf 例程，用户可根据 printf 进行功能调试，输出 log
uart	uart 操作例程，包括 uart 如何配置使用
adc	adc 操作例程，包括 adc 如何配置使用
gpio	gpio 操作例程，包括 gpio 如何配置使用
alarm	alarm 定时器操作例程，包括 alarm 如何配置使用
wefota	Wefota 下载版本操作例程，包括 wefota 如何使用
i2c	i2c 操作例程，包括 i2c master 如何配置使用
iot	iot 业务例程，包括 iot 业务使用建立
spi	spi 操作例程，包括 spi master 如何配置使用
sleepcok	电源管理例程，包括 sleepcok 如何配置使用
tcp	tcp 操作例程，包括 tcp 如何配置使用
tcpdemo	综合业务例程

3. 编译器安装

客户首次进行 WelinkOpen 开发，需准备好编译环境。该交叉编译链工具在该目录下存放 (../sdk/tools/gcc-arm-none-eabi-4_8-2014q3-20140805-win32)。

① 双击 gcc-arm-none-eabi-4_8-2014q3-20140805-win32.exe 应用程序开始安装。

busybox	2018/4/9 11:03	应用程序	431 KB
gcc-arm-none-eabi-4_8-2014q3-2014...	2018/4/9 11:03	应用程序	81,373 KB
make	2018/4/9 11:03	应用程序	257 KB

图 3-1 交叉编译工具安装程序

② 打开安装程序后，选择默认的 English 选项，点击 OK，进行下一步。

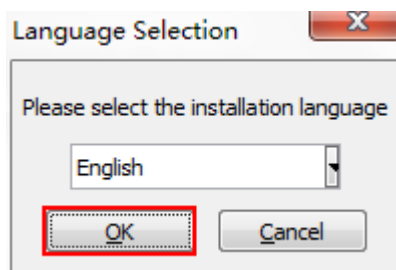


图 3-2 打开交叉编译安装程序

③ 确认继续进行安装，点击 Yes，进行下一步。

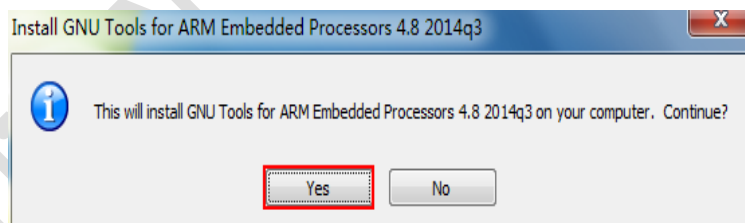


图 3-3 选择继续安装

④ 此时出现如下界面，继续选择 Next。

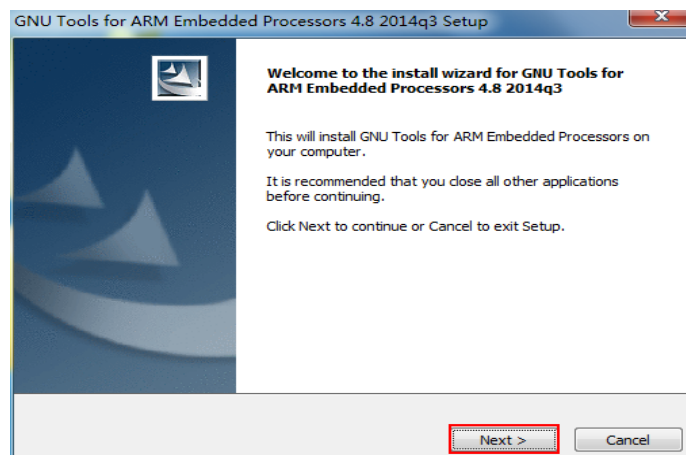


图 3-4 安装界面

⑤ 选择接受协议，继续 Next 安装。

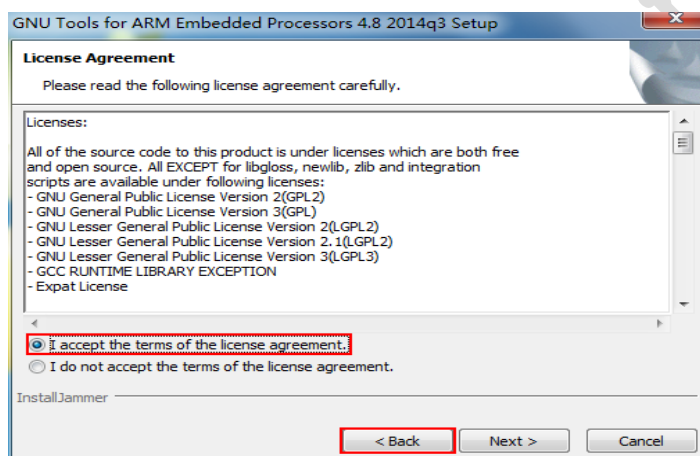


图 3-5 接受协议界面

⑥ 确认安装路径界面。**注：切勿修改安装程序默认的安装路径**

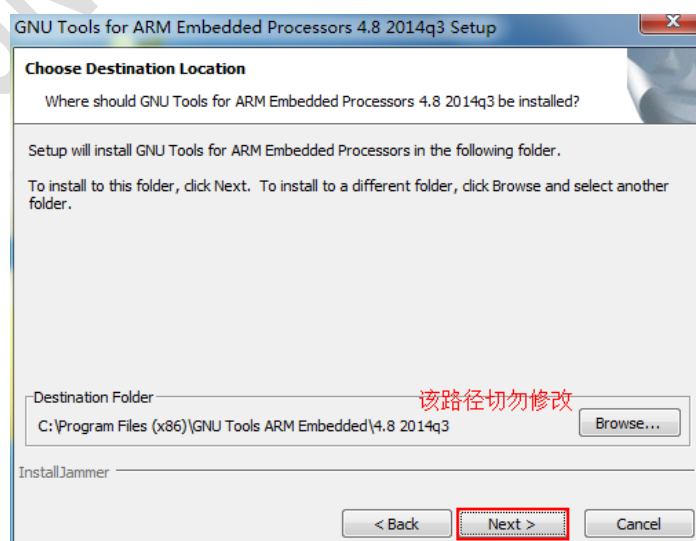


图 3-6 安装路径界面

⑦ 继续选择 Next 进行安装。

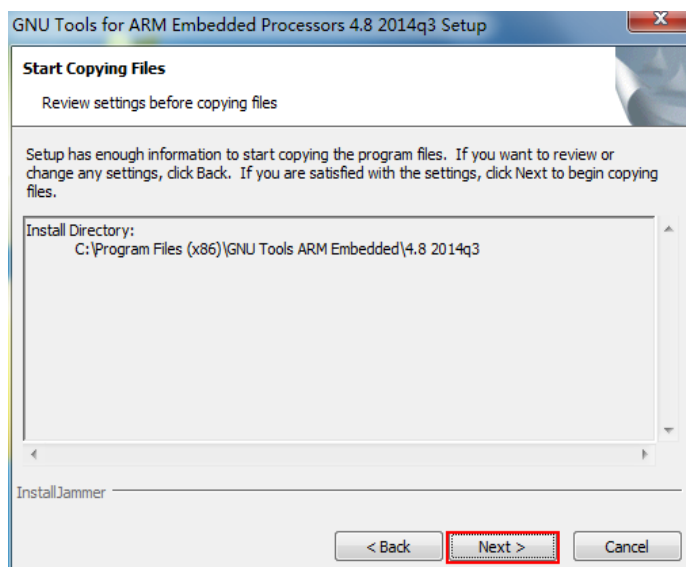


图 3-7 安装界面

⑧ 等待安装完成，选择 Finish 完成安装。

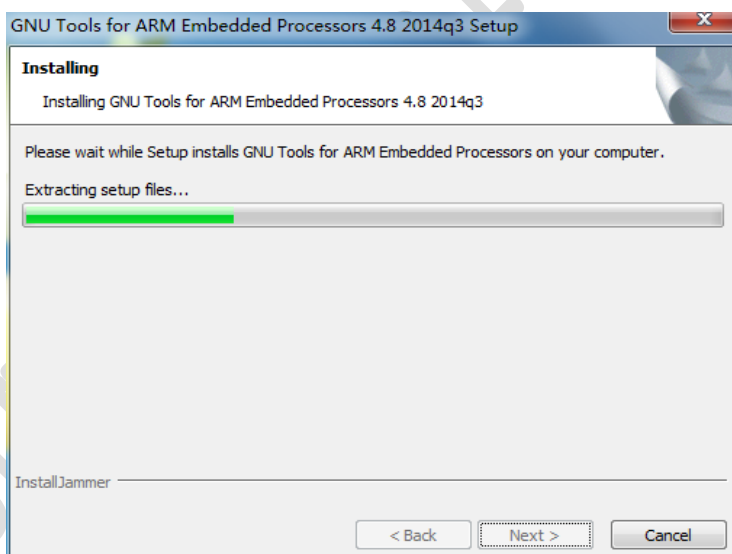


图 3-8 安装界面

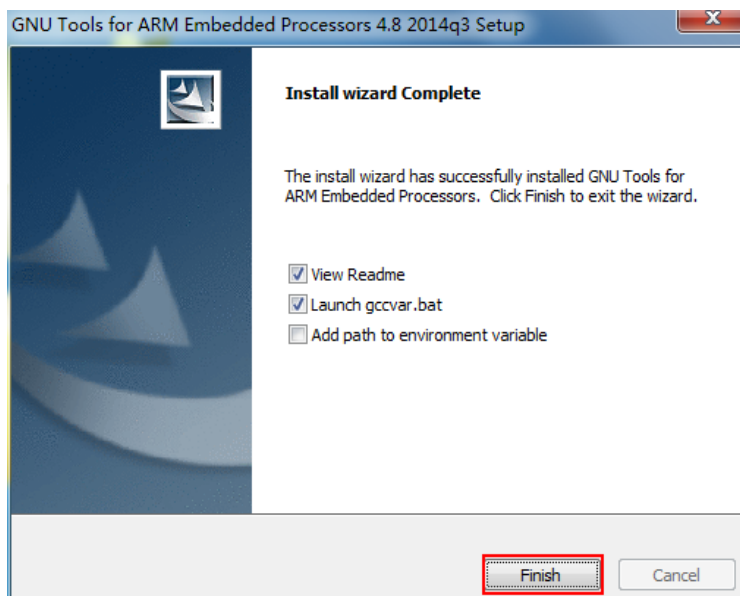


图 3-9 安装完成界面

到此 WelinkOpen 编译工具安装完成，可以使用 examples 目录下的 build 脚本对所有例程进行编译来检测交叉编译工具是否安装成功，成功编译后在例程的目录下会生成 bin 目录，在该目录下生成的即为烧录文件。同时用户也可使用 clean 脚本对例程的编译结果进行清除操作。

4. 客户程序添加及编译

在安装好交叉编译环境后即可进行用户应用程序的编译。

① 在 examples 目录下创建自己的目录，例如添加 test 例程。

flash	2018/4/9 11:03	文件夹	
Hello	2018/4/9 11:03	文件夹	
printf	2018/4/9 11:03	文件夹	
test	2018/4/12 11:11	文件夹	
uart	2018/4/9 11:03	文件夹	
build	2018/4/9 11:03	Windows 批处理...	1 KB
clean	2018/4/9 11:03	Windows 批处理...	1 KB
Makefile	2018/4/9 11:03	文件	2 KB

图 4-1 创建用户目录

② 在该目录下编写自己需要实现的.c 文件以及 makefile 文件，Makefile 文件可参考其他例程中提供的 Makefile。

```
#####
SDK_PATH := $(realpath ../../) sdk路径
#####
# target
#####
TARGET := flash 目标名称
#####
# common variables
#####
DEBUG := 1 DEBUG标志
#####
# source
#####
C_SOURCES := flash.c 添加源文件
S_SOURCES :=
#####
# CFLAGS
#####
C_DEFS :=
C_INCLUDES :=
#####
# ASFLAGS
#####
# macros for gcc
AS_DEFS :=
#####
# includes for gcc
AS_INCLUDES :=
#####
# LDFLAGS
#####
# libraries
LD_LIBS := -lgsdk 添加库
#####
# Build rule
#####
include $(SDK_PATH)/scripts/rules.mk
```

图 4-2 makefile 例程

③以 test 为例，编写完 test.c 后，进行 makefile 编写，修改点见下图：

```
#####
# target
#####
TARGET := test 修改点1

#####
# common variables
#####
DEBUG := 1

#####
# source
#####
C_SOURCES := test.c 修改点2

S_SOURCES :=
```

图 4-3 test 例程对应的 makefile 修改点

④ 完成用户应用程序和 makefile 编写添加后，进行编译。在 examples 目录下执行 bulid 脚本（双击即可）即可对所有例程进行编译，使用 clean 脚本（双击即可）可对例程的编译结果进行清除。编译成功后在对应的目录下会生成对应的 bin 目录以及 obj 目录。bin 目录对应的为烧录所需的文件。

flash	2018/4/12 13:41	文件夹	
Hello	2018/4/12 13:41	文件夹	
printf	2018/4/12 13:41	文件夹	
uart	2018/4/12 13:41	文件夹	
build	2018/4/9 11:03	Windows 批处理...	1 KB
clean	2018/4/9 11:03	Windows 批处理...	1 KB
Makefile	2018/4/9 11:03	文件	2 KB

图 4-4 脚本路径

⑤图 4-5 为以 test 例程为例正确编译后输出的信息，其他例程编译成功后信息基本一致。

```
make -C test
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/test'
      MKDIR    obj
      CC       test.o
      MKDIR    bin
      LN       test.elf
      text      data      bss      dec      hex filename
      680      8      8736      9424      24d0 bin/test.elf
      BIN      test.nocrc
      CKSUM    test.bin
rm bin/test.nocrc
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/test'
```

图 4-5 test 例程编译成功

⑥图 4-6 为对编译结果进行成功清除，在对应用程序进行重新编译时需要先执行 clean 脚本删除此前编译生成的相关文件，成功执行后对应例程目录下的 bin 目录及 obj 目录就会被删除。

```
make -C uart clean
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/uart'
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/uart'
make -C printf clean
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/printf'
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/printf'
make -C flash clean
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/flash'
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/flash'
make -C test clean
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/test'
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/test'
make -C Hello clean
make[1]: Entering directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/Hello'
make[1]: Leaving directory 'C:/Users/zhangxin/Desktop/OpenCPU/sdk/examples/Hello'
请按任意键继续. . .
```

图 4-6 例程 clean 成功

5. 系统烧录

① 硬件上电及连接，具体硬件详情可见 ME3616 开发板指导手册。

- 1) 插入跳线帽如下图 5-2 所示。
- 2) 用 5V 电源适配器给模组供电。
- 3) 插入 micro-USB 线到 Debug_UART 口，另一端连接到 PC 端。

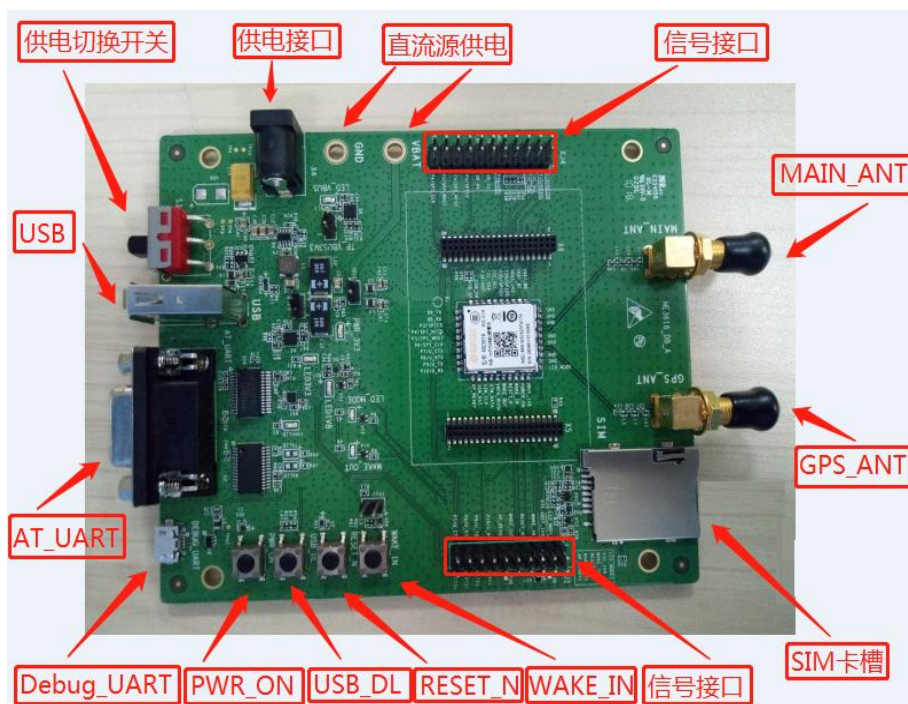


图 5-1 硬件连接图

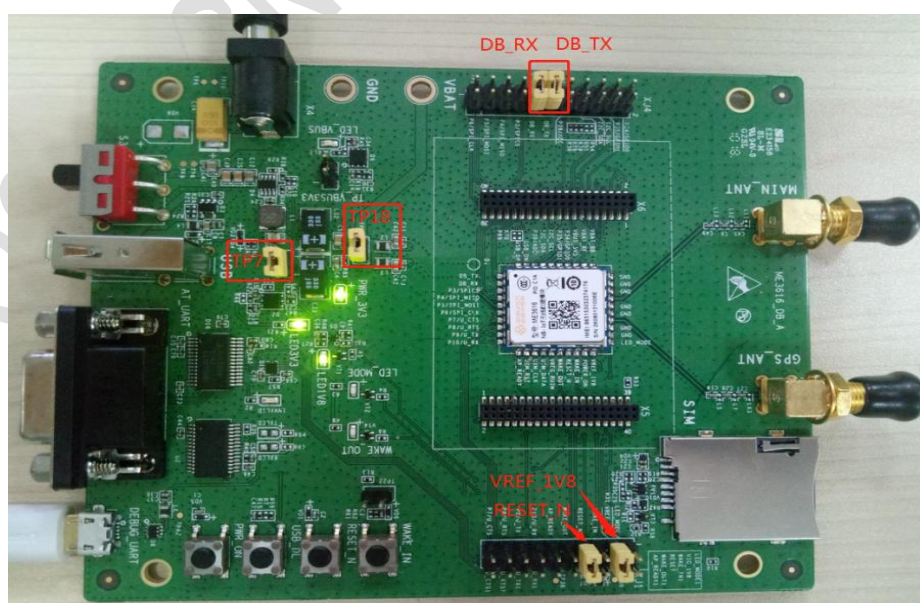


图 5-2 跳线帽连接图

② 驱动安装。

如果模组 TTL 转 USB 驱动安装好后，且正常开机，设备管理器中串口如下 COM43：

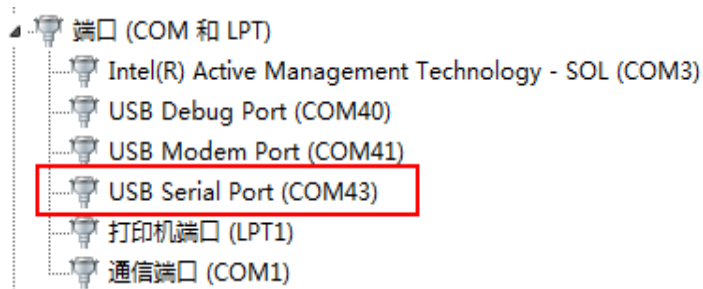


图 5-2 串口驱动目录

③ 双击 download_tool.exe，打开下载工具，出现如图所示的界面。

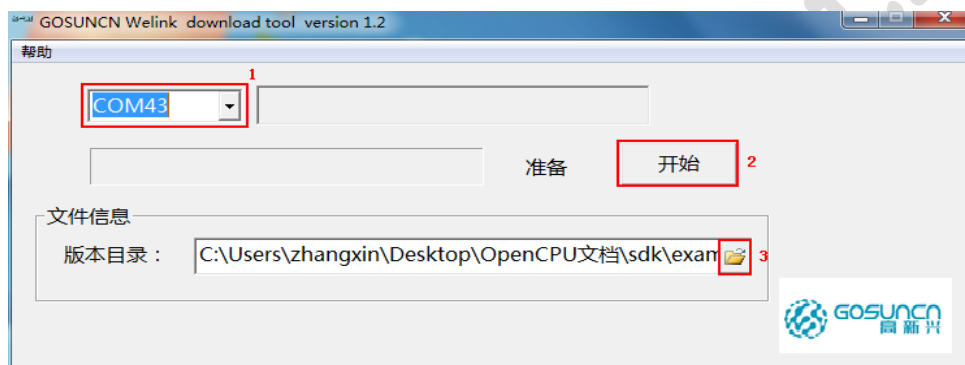


图 5-3 下载工具

④ 点击 1 处的组合框选择 PC 机上设备管理器中对应的端口号，如果 PC 机上对应的端口号大于 256，可手动输入端口，例如：“COM1102”。然后点击 3 处的按钮选择客户应用程序版本所在路径（例如../sdk/examples/flash/bin/download.cfg）。完成以上步骤后点击 2 处的开始按钮，此时对模组复位（按 RESET_IN 键）紧接着再长按 pwoer_on 键直到状态框出现“Download ROM ...”时可松开电源键。

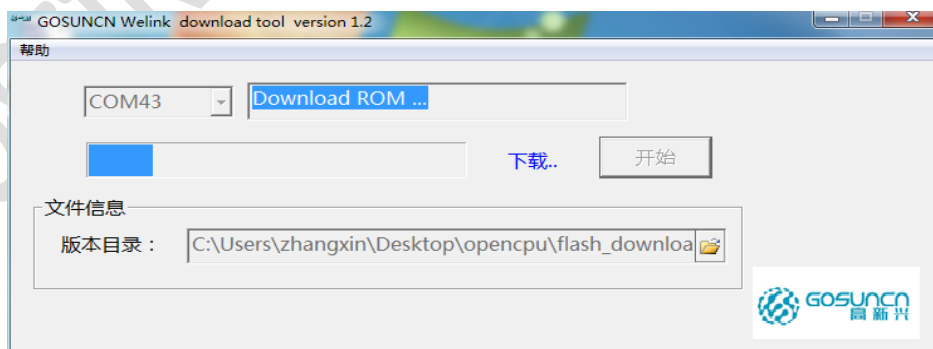


图 5-4 烧录界面

⑤ 成功后会出现如图所示的界面：

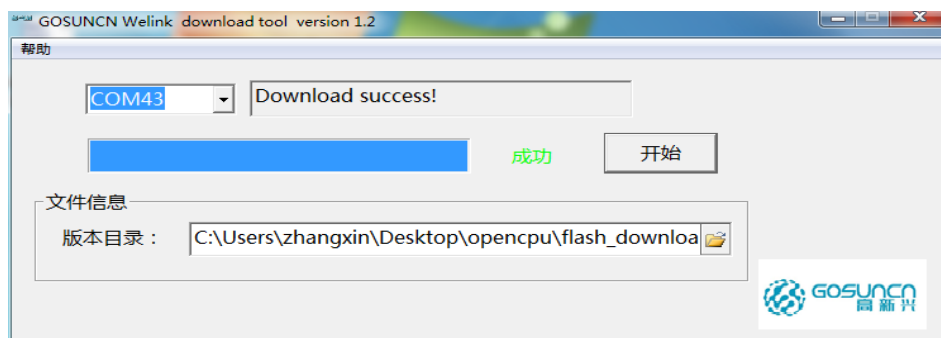


图 5-5 升级成功

- ⑥ 在点击“开始”后，若工具一直没有反应，直到出现图 5-7 所示的提示。说明板子端口不通，或者端口选择错误，或者端口被其他程序占用，或者板子硬件有问题，建议检测板子端口情况。然后关闭工具重新开始。



图 5-6 错误提示

⚠ 注意： 该下载工具不要放在有任何中文字符的路径下，且下载的版本也不要放在有中文的目录下！

6. 查看调试应用程序

6.1. 如何使用 GENIE TOOL 工具

- ① 打开 Genie Tool 工具软件。例如 E:\nbiot\tools\core\genie\genie.exe。
- ② 配置 Config 文件。如第一次使用 LOG 工具，没有配置过 Config 文件，选择 “New Config”。

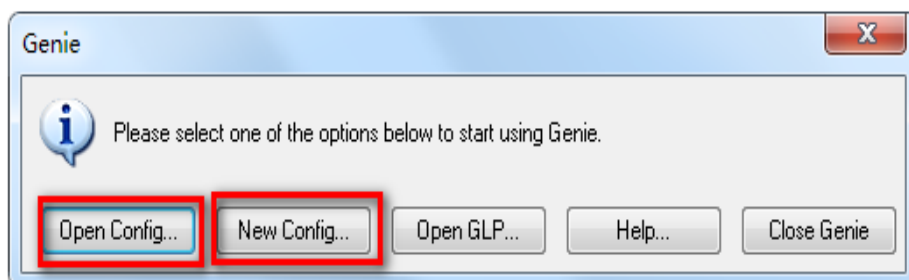


图 6-1 genie 工具 config 界面

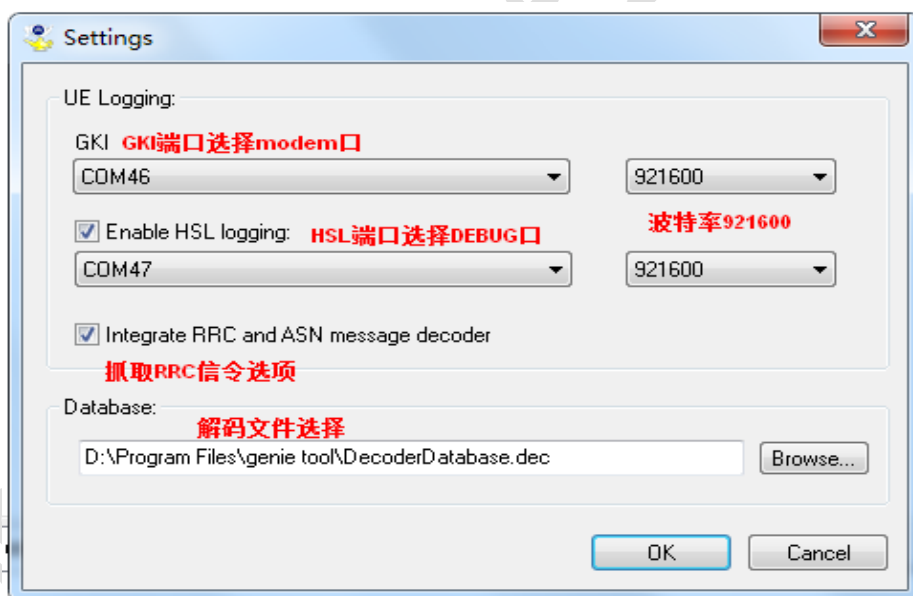


图 6-2 genie 工具 config 界面

- ③ 如果已经配置了 Config 文件，选择 “Open Config”，如下所示：

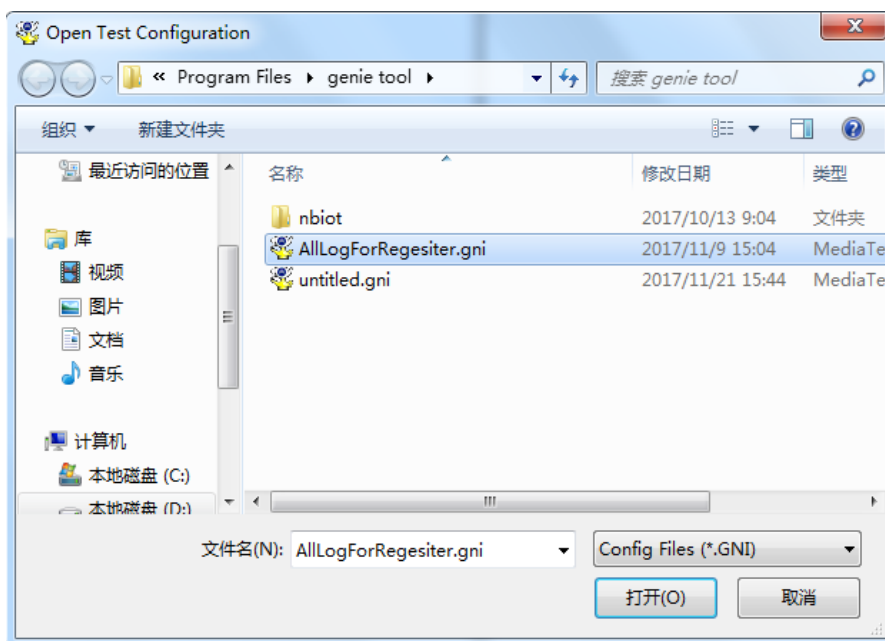


图 6-3 config 文件

④ 成功配置 config 后，抓 LOG 界面如下。

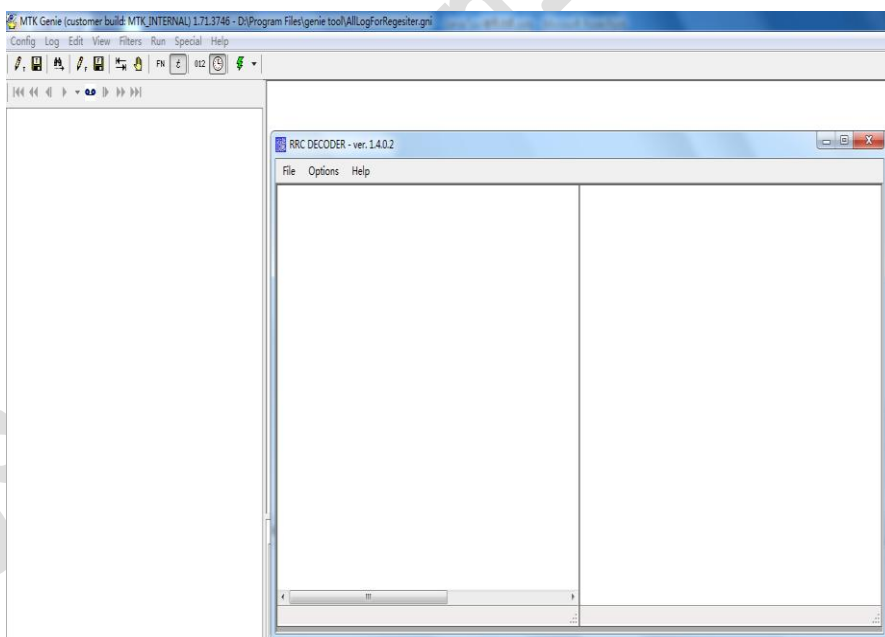


图 6-4 genie 界面

⑤ 设置 LOG 过滤方式，如第一次使用 LOG 工具，没有配置过 Filter 文件，选择 “Edit Filter” 。

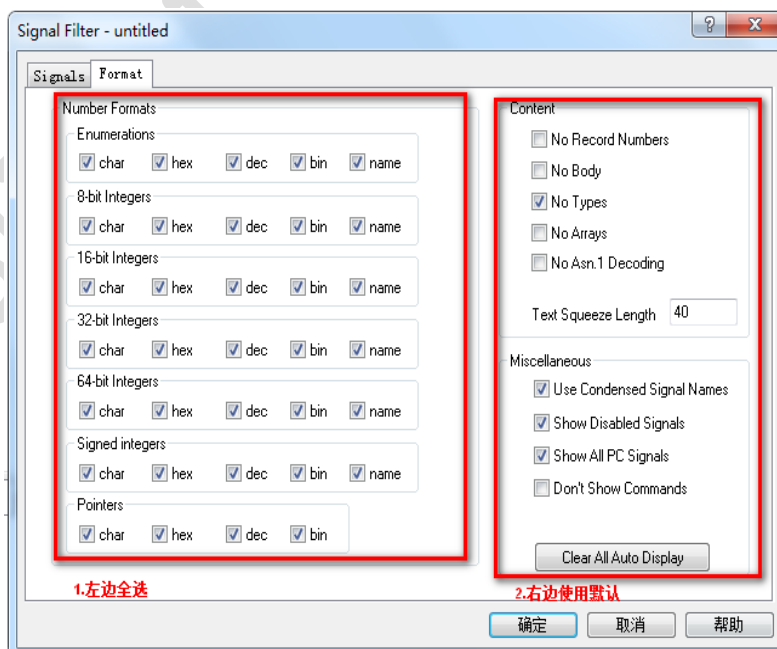
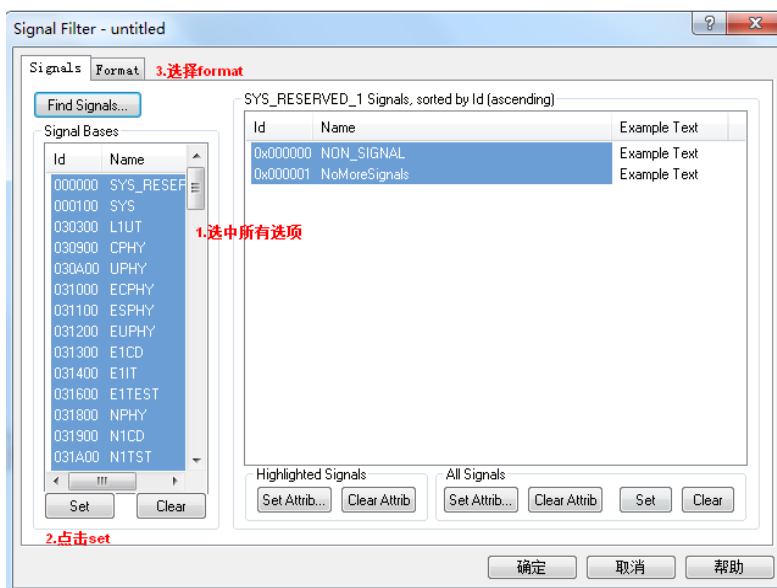
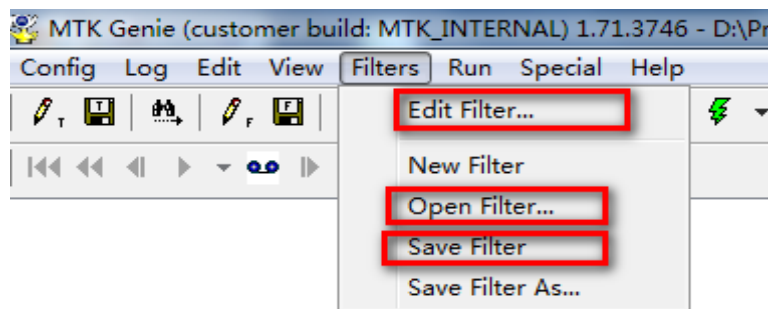


图 6-5 filters 界面

⑥ 设置好后,选择保存 Filter 设置,如下所示,下次使用时,直接 open filter 即可,不需要重新配置 Filter。

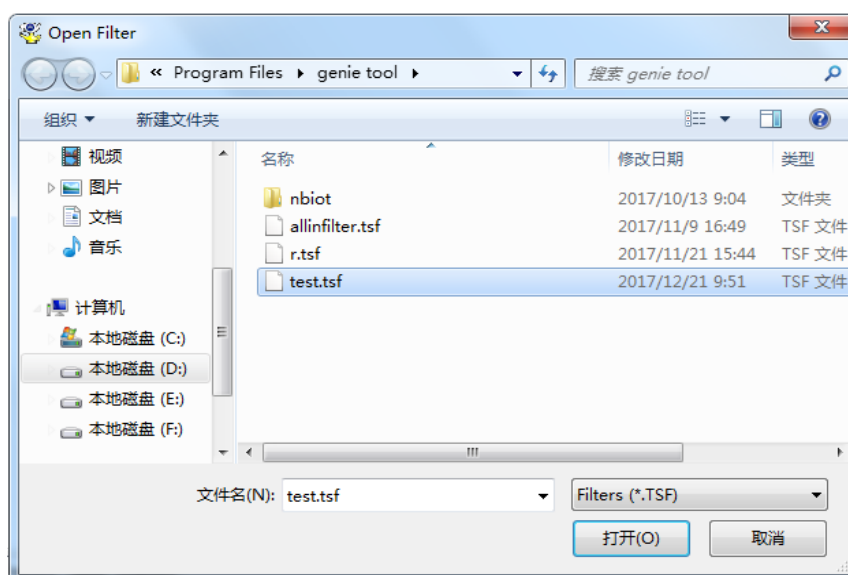


图 6-6 filters 配置文件

⑦ 单击  启动抓包流程，Log 如下图所示：

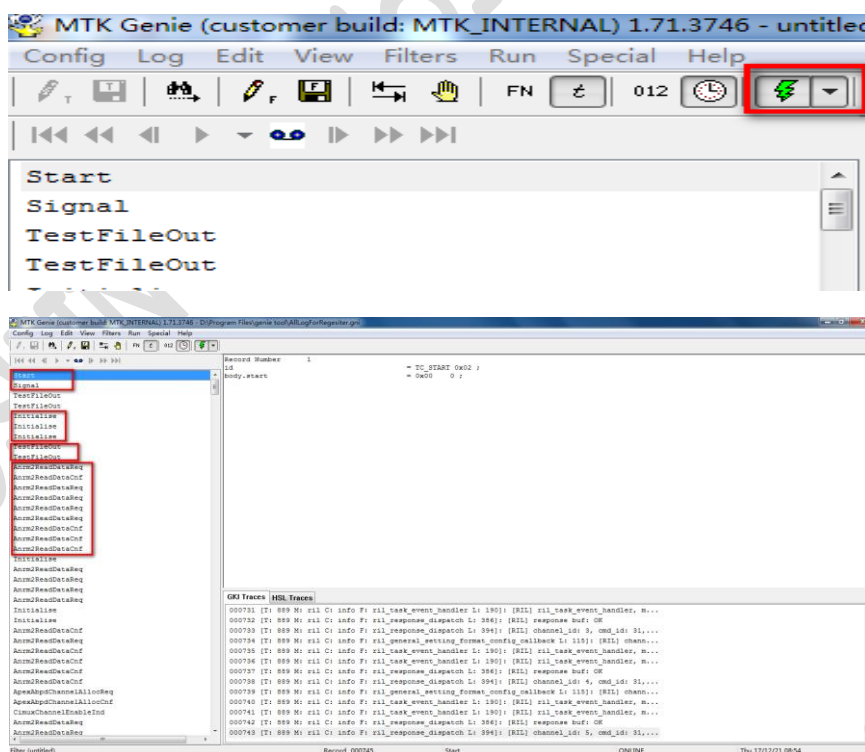


图 6-7 启动 log 抓取

⑧ 保存 LOG。

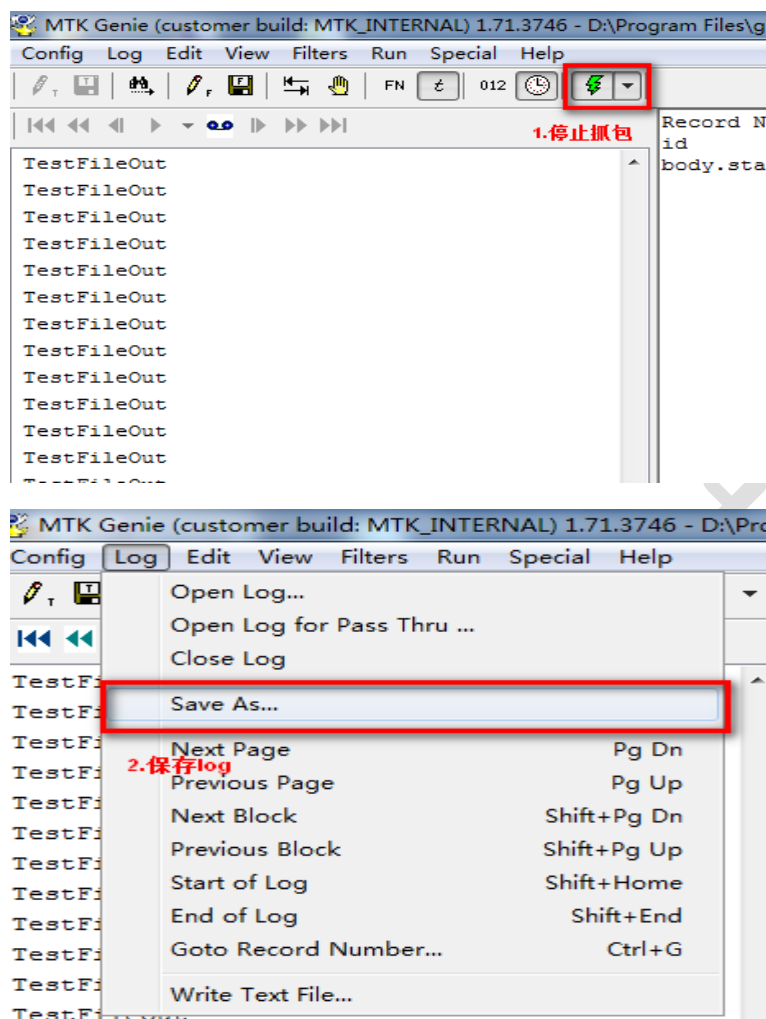


图 6-8 保存 log

6.2. 用户抓取 LOG

① 用户在编写自己的应用程序时，可参照 examples 目录下的 Hello 例程，对代码需要输出 log 处调用 syslog_printf 接口函数，输出调试 log。

② 模组通过开发板 USB 口（见图 5-1）连接 PC 端，使用 genie 工具，查看输出 log，使用 AT_UART 口可发送 AT 命令，具体硬件连接详情可参考 ME3616 开发板使用手册。

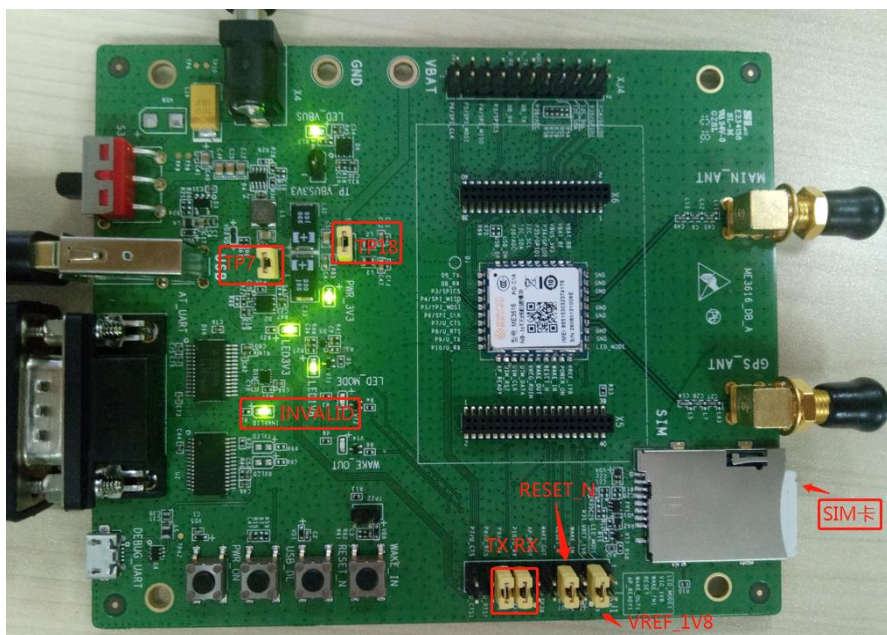



图 6-9 使用 AT_UART 跳线示意图

③ 模组开机，按下 power_on 键，单击  启动抓取 log，此时可以看到通过串口助手的一些启动信息和 genie 工具上的启动 log。

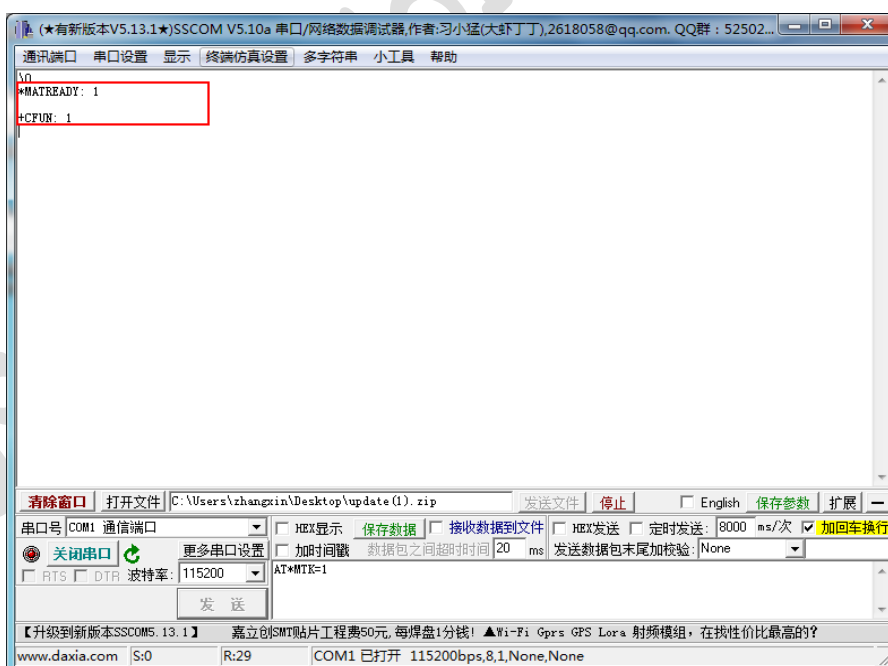


图 6-10 启动信息

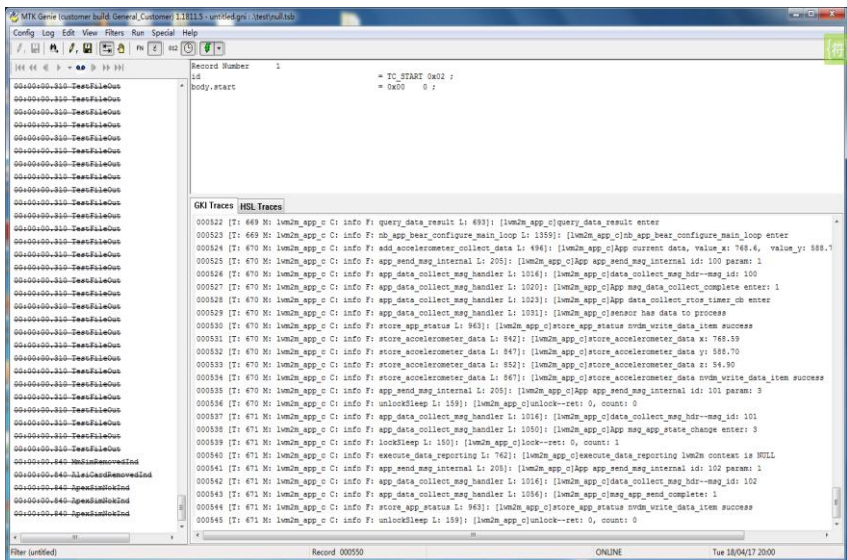


图 6-11 启动 log 信息

④ 在串口助手上发送启动应用程序命令 AT+ZBOOTAPPAUTO=1，返回 OK 信息表示客户程序开始运行（**重新启动后客户程序可以自动运行**），以 Hello 例程为例，这时在 genie 工具上可看到用户程序的 log 信息，这里的 log 信息为 Hello world！

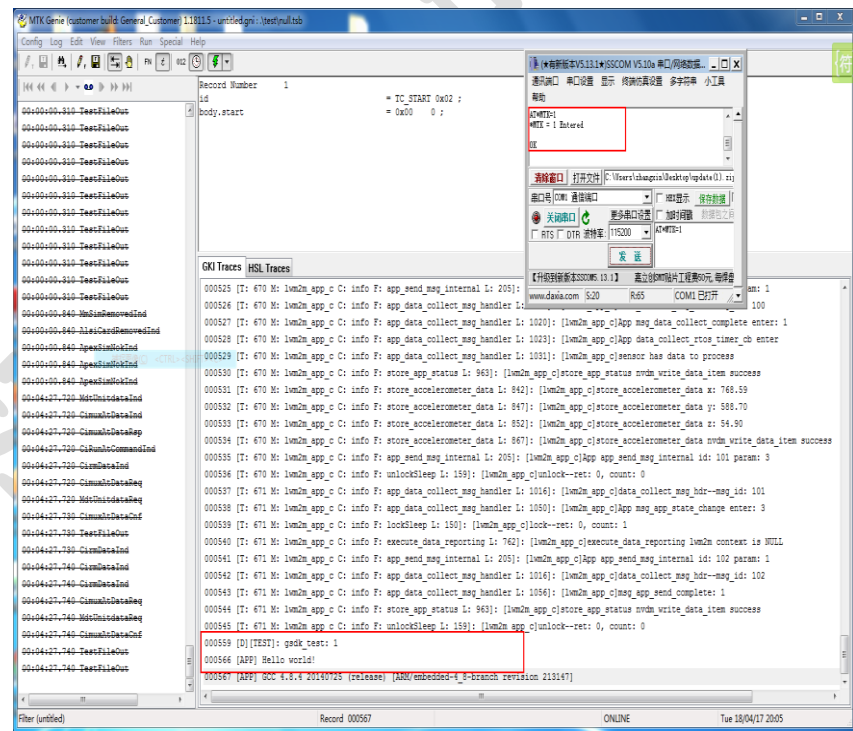


图 6-12 用户 log 信息

以上通过图文结合的方式，详细说明了 WelinkOpen RTOS SDK 安装包从安装到开发，再到编译，最后将程序烧录到板子的四个过程。旨在指导客户更好更快地开发项目。