

# MN316

## OpenCPU 开发指导手册

NB-IoT 系列

版本：V1.0.0

日期：2020 年 12 月

# 服务与支持

如果您有任何关于模组产品及产品手册的评论、疑问、想法，或者任何无法从本手册中找到答案的疑问，请通过以下方式联系我们。



## 中移物联网有限公司

OneMO 官网: [onemo10086.com](http://onemo10086.com)

邮箱: [SmartModule@cmiot.chinamobile.com](mailto:SmartModule@cmiot.chinamobile.com)

客户服务热线: 400-110-0866

微信公众号: CMOneMO



中国移动  
China Mobile

# 文档声明

## 注意

本手册描述的产品及其附件特性和功能，取决于当地网络设计或网络性能，同时也取决于用户预先安装的各种软件。由于当地网络运营商、ISP，或当地网络设置等原因，可能也会造成本手册中描述的全部或部分产品及其附件特性和功能未包含在您的购买或使用范围之内。

## 责任限制

除非合同另有约定，中移物联网有限公司对本文档内容不做任何明示或暗示的声明或保证，并且不对特定目的适销性及适用性或者任何间接的、特殊的或连带的损失承担任何责任。

在适用法律允许的范围内，在任何情况下，中移物联网有限公司均不对用户因使用本手册内容和本手册中描述的产品而引起的任何特殊的、间接的、附带的或后果性的损坏、利润损失、数据丢失、声誉和预期的节省而负责。

因使用本手册中所述的产品而引起的中移物联网有限公司对用户的最大赔偿（除在涉及人身伤害的情况中根据适用法律规定的损害赔偿外），不应超过用户为购买此产品而支付的金额。

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。公司保留随时修改本手册中任何信息的权利，无需进行提前通知且不承担任何责任。

## 商标声明



中国移动  
China Mobile 为中国移动注册商标。

本手册和本手册描述的产品中出现的其他商标、产品名称、服务名称和公司名称，均为其各自所有者的财产。

## 进出口法规

出口、转口或进口本手册中描述的产品（包括但不限于产品软件和技术数据），用户应遵守相关进出口法律和法规。

## 隐私保护

关于我们如何保护用户的个人信息等隐私情况，请查看相关隐私政策。

## 操作系统更新声明

操作系统仅支持官方升级；如用户自己刷非官方系统，导致安全风险和损失由用户负责。

## 固件包完整性风险声明

固件仅支持官方升级；如用户自己刷非官方固件，导致安全风险和损失由用户负责。

## 版权所有©中移物联网有限公司。保留一切权利。

本手册中描述的产品，可能包含中移物联网有限公司及其存在的许可人享有版权的软件，除非获得相关权利人的许可，否则，非经本公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并以任何形式传播。



# 关于文档

## 修订记录

版本	日期	作者	描述
V1.0.0	2020/12/8	张雄威/任亚洲	初版



中国移动  
China Mobile

# 目录

服务与支持.....	2
文档声明.....	3
关于文档.....	5
修订记录.....	5
目录.....	6
图示索引.....	7
1 概述.....	8
2 MN316 OPENCPU SDK 开发包介绍.....	9
3 GCC 编译环境安装.....	10
4 程序编译.....	11
5 固件烧录.....	13
6 DEMO 例程演示.....	16
7 SDK 开发指导.....	17
7.1 OpenCPU 模式启动方法.....	17
7.2 用户程序执行流程.....	17
7.3 增加源文件方法.....	18
7.4 低功耗模式下的编程.....	19
7.5 SDK 硬件版本说明.....	19
7.6 调试及疑难问题解决.....	19
7.7 异常场景下 NV 的处理情况.....	20
8 SDK API 分类说明.....	21
8.1 模组操作系统.....	21
8.2 模组网络相关接口.....	21
8.3 模组基础功能接口.....	21
8.4 硬件外设接口.....	21
9 LOG 工具使用.....	22

# 图示索引

图 2-1: OC SDK 目录结构 .....9

图 3-1: 编译工具版本 ..... 10

图 4-1: 编译 DBRDH0 版本..... 11

图 4-2: 编译生成 bin 文件 ..... 11

图 4-3: 清除编译..... 12

图 5-1: 开发板介绍..... 13

图 5-2: 下载界面..... 14

图 5-3: 选择合并文件 ..... 14

图 5-4: 固件下载..... 15

图 6-1: 示例运行..... 16

图 9-1: 连接 Log 工具..... 22

图 9-2: 输出 Log ..... 22

图 9-3: 导出 Wireshark 数据 ..... 23

图 9-4: 导出 Dump 数据 ..... 23



# 1 概述

MN316 OpenCPU SDK 以压缩包形式交付，包含交叉编译链、必要的库文件和头文件、接口文档等资料，并在源码中集成使用示例以帮助用户快速搭建开发环境，缩短软件开发周期。按照使用指导解压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。

开发过程中遇到问题请联系中移物联网技术支持。





## 2 MN316 OpenCPU SDK

# 开发包介绍

SDK 目录详情见下图。Windows 环境安装 gcc 编译环境后可编译软件版本。其中，build.bat 为批处理编译脚本，如无必要请勿修改。

custom	2020/12/3 10:24	文件夹	
docs	2020/12/3 10:24	文件夹	
examples	2020/12/3 10:24	文件夹	
include	2020/12/3 10:24	文件夹	
out	2020/12/4 15:30	文件夹	
prebuild	2020/12/3 10:24	文件夹	
third-party	2020/12/3 10:24	文件夹	
tools	2020/12/3 10:24	文件夹	
build	2020/12/4 15:14	Windows 批处理...	1 KB
readme	2020/12/3 9:26	MD 文件	1 KB

图 2-1: OC SDK 目录结构

其它主要目录详情说明如下。

目录名称	描述
custom	用户代码存放
inc	头文件
src	源文件
custom.mk	用户 makefile
docs	软件相关文档
examples	示例程序
uart	串口示例程序
spi	spi 示例程序
...	
examples.mk	示例程序 makefile
include	系统头文件，请勿修改。
kernel	内核相关
lwip	lwip 协议栈
cmiot	cmiot 头文件
out	编译生成文件，动态创建。
objs	编译.o.d
logs	编译日志
images	编译生成的 bin 文件
prebuild	预构建文件，请勿修改。
third-party	三方库文件
tools	编译工具及脚本集合，请勿修改。

### 3 gcc 编译环境安装

OpenCPU SDK 采用 ARM-GCC 交叉编译工具，版本为 7.3.1 20180622-release 及以上。安装步骤如下：

- (1) 解压“GNU Tools ARM Embedded.zip”文件至“C:\Program Files (x86)”目录下；
- (2) 添加“C:\Program Files (x86)\GNU Tools ARM Embedded\7 2018-q2-update\bin”路径至系统环境变量中；
- (3) gcc 交叉编译环境安装完成后，使用“arm-none-eabi-gcc -v”命令检测编译工具是否安装成功。

```
E:\MN316\软件代码\MN316-OPENCPU1116>arm-none-eabi-gcc -v
Using built-in specs.
COLLECT_GCC=arm-none-eabi-gcc
COLLECT_LTO_WRAPPER=c:/program files (x86)/gnu\ tools\ arm\ embedded\7\ 2018-q2-update\bin/./lib/gcc/arm-none-eabi/7.3.1/lto-wrapper.exe
Target: arm-none-eabi
Configured with: /tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/src/gcc/configure --build=x86_64-linux-gnu --host=i686-w64-mingw32 --target=arm-none-eabi --prefix=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw --libexecdir=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/lib --infodir=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/share/doc/gcc-arm-none-eabi/info --mandir=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/share/doc/gcc-arm-none-eabi/man --htmldir=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/share/doc/gcc-arm-none-eabi/html --pdfdir=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/share/doc/gcc-arm-none-eabi/pdf --enable-languages=c,c++ --enable-mingw-wildcard --disable-decimal-float --disable-libffi --disable-libgomp --disable-libmudflap --disable-libquadmath --disable-libssp --disable-libstdc++-pch --disable-nls --disable-shared --disable-threads --disable-tls --with-gnu-as --with-gnu-ld --with-headers=yes --with-newlib --with-python-dir=share/gcc-arm-none-eabi --with-sysroot=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/install-mingw/arm-none-eabi --with-libiconv-prefix=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/build-mingw/host-libs/usr --with-gmp=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/build-mingw/host-libs/usr --with-mpfr=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/build-mingw/host-libs/usr --with-mpc=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/build-mingw/host-libs/usr --with-libelf=/tmp/jenkins/jenkins-GCC-7-build_toolchain_docker-775_20180622_1529687456/build-mingw/host-libs/usr --with-host-libstdc++=' -static-libgcc -Wl,-Bstatic,-lstdc++,-Bdynamic -lm' --with-pkgversion='GNU Tools for Arm Embedded Processors 7-2018-q2-update' --with-multilib-list=rmprofile
Thread model: single
gcc version 7.3.1 20180622 (release) [ARM/embedded-7-branch revision 261907] (GNU Tools for Arm Embedded Processors 7-2018-q2-update)
```

图 3-1：编译工具版本

## 4 程序编译

进入 SDK 根目录，打开命令行工具，输入 “build.bat xxxx” 命令，即可执行编译，xxxx 为对应的编译版本，目前支持 DBRDH0（双频硬件 H0 版本）及 XGCDH2（多频硬件 H2 版本）两种，用户根据使用的硬件选择不同的版本进行编译。

```
E:\MN316\软件代码\MN316-OPENCPU1207\OC-SDK>build.bat DBRDH0
compile custom/src/cm_main.c
compile examples/3gpp/cm_demo_3gpp.c
compile examples/flash/cm_demo_flash.c
compile examples/gpio/cm_demo_gpio.c
compile examples/i2c/cm_demo_i2c.c
compile examples/net/cm_demo_net.c
compile examples/onenet/cm_demo_onenet.c
compile examples/onenet/cm_demo_onenet_fota.c
compile examples/os/cm_demo_os.c
compile examples/spi/cm_demo_spi.c
compile examples/timer/cm_demo_timer.c
compile examples/uart/cm_demo_uart.c
start link...
done!!
arm-none-eabi-objcopy -O binary -j .flash.text xinyiNBSoc_M3.axf ./out/objs/xinyiNBSoc_M3/flash.bin
arm-none-eabi-objcopy -O binary -j .text -j .data xinyiNBSoc_M3.axf ./out/objs/xinyiNBSoc_M3/ram.bin
./tools/toolchain/cp.exe ./out/objs/xinyiNBSoc_M3/xinyiNBSoc_M3.axf ./out/images/DBRD/H0/xinyiNBSoc_M3.axf
./tools/toolchain/cp.exe ./out/objs/xinyiNBSoc_M3/ram.bin ./out/images/DBRD/H0/ram.bin
./tools/toolchain/cp.exe ./out/objs/xinyiNBSoc_M3/flash.bin ./out/images/DBRD/H0/flash.bin
```

图 4-1：编译 DBRDH0 版本

编译成功后，out 目录下自动生成编译文件，并在 “out/images” 文件夹中生成最新的 flash.bin、ram.bin 等文件。

FACTORY_NV.Hex	2020/11/6 17:07	HEX 文件	10 KB
flash	2020/11/16 9:51	BIN 文件	361 KB
loginfo.info	2020/11/6 17:07	INFO 文件	77 KB
ram	2020/11/16 9:51	BIN 文件	72 KB
readme	2020/10/21 9:12	TXT 文件	0 KB
secondary_bootloader_backup	2020/10/21 9:12	BIN 文件	26 KB
secondary_bootloader_prime	2020/10/21 9:12	BIN 文件	26 KB
xinyi	2020/10/9 10:25	配置设置	2 KB
xinyi_dsp	2020/10/9 10:25	配置设置	1 KB
Xinyi_NBIoT_Master_Branch_V1100B0...	2020/11/6 17:07	文件	6,967 KB
Xinyi_NBIoT_Master_Branch_V1100B0...	2020/11/6 17:07	文件	69 KB
Xinyi_NBIoT_Master_Branch_V1100B0...	2020/11/6 17:07	文件	1,690 KB
xinyiNBSoc_dsp	2020/11/6 17:07	好压 IMG 压缩文件	799 KB
xinyiNBSoc_M3	2020/11/6 17:07	AXF 文件	7,190 KB

图 4-2：编译生成 bin 文件

在命令行中输入“build.bat clean”，即可删除编译文件。

```
E:\MN316\软件代码\MN316-OPENCPU1203\OC-SDK>build.bat clean  
clear compiled files
```

图 4-3：清除编译



## 5 固件烧录

开发板如下图所示，开发时使用 micro-USB 线连接“供电/串口”及 PC 机 USB 口，“电源”开关拨至“USB1”方向，“开机”开关拨至“VBAT\_H”方向，“唤醒电平”拨至“H”方向。开机后，“开机”开关旁 LED 灯常亮，PC 机资源管理器将识别出两个串口，一路为主串口（UART0），另一路为 Log 口，可使用串口工具判断两个串口的实际功能。

当使用 Logview 下载固件无法正常进行时，可通过如下步骤让模组进入下载模式。

- (1) 将“开机”开关拨至“VBAT\_L”侧关机；
- (2) 短暂按住“BOOT”按钮；
- (3) 将“开机”开关拨至“VBAT\_H”侧开机，立即松开“BOOT”按键。

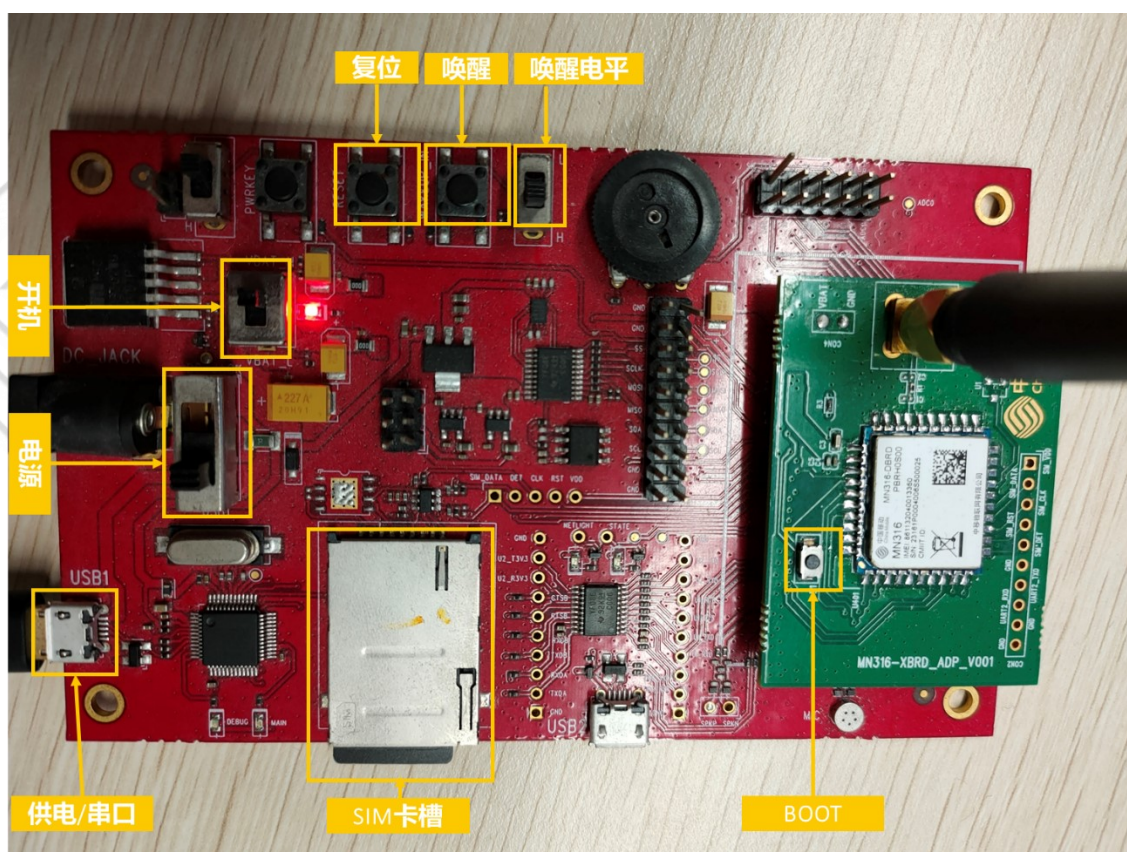


图 5-1：开发板介绍



固件烧录使用 Logview 工具，点击面板中的“Download”选项，即可打开下载工具。

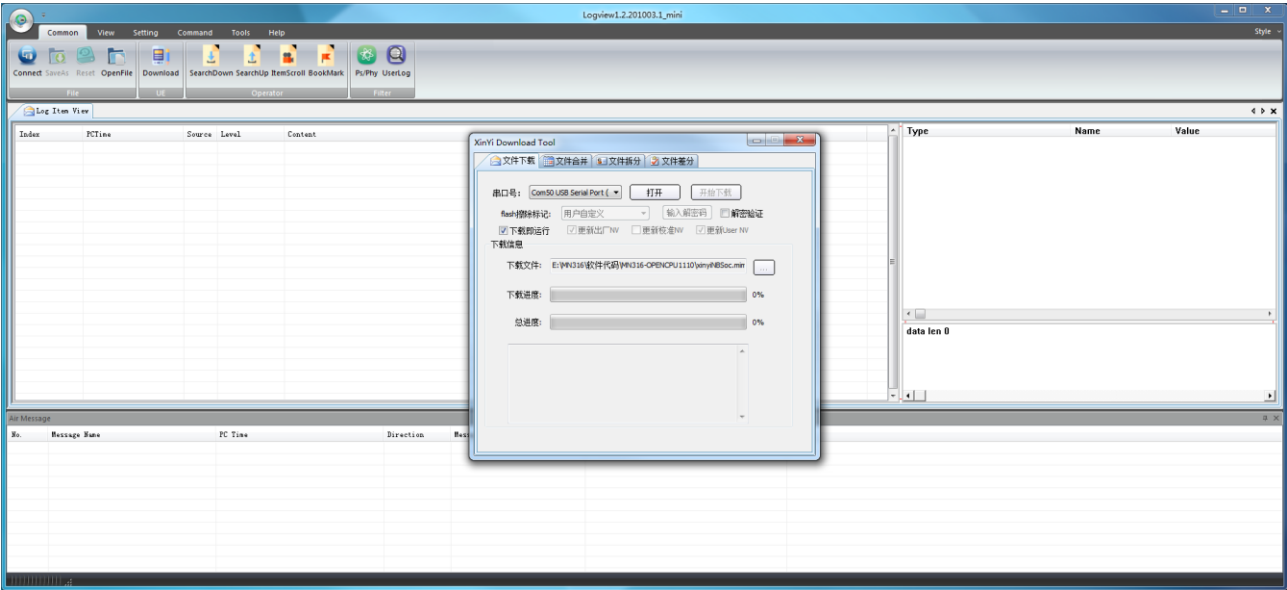


图 5-2：下载界面

执行如下步骤下载固件：

- (1) 点击下载工具中的“文件合并”菜单，将“选择文件夹”中的目录设置为编译生成的 images 目录，将“输出文件名”设置为 SDK 根目录下的“xinyiNBSoc.img”，“加密码”选项无需设置；

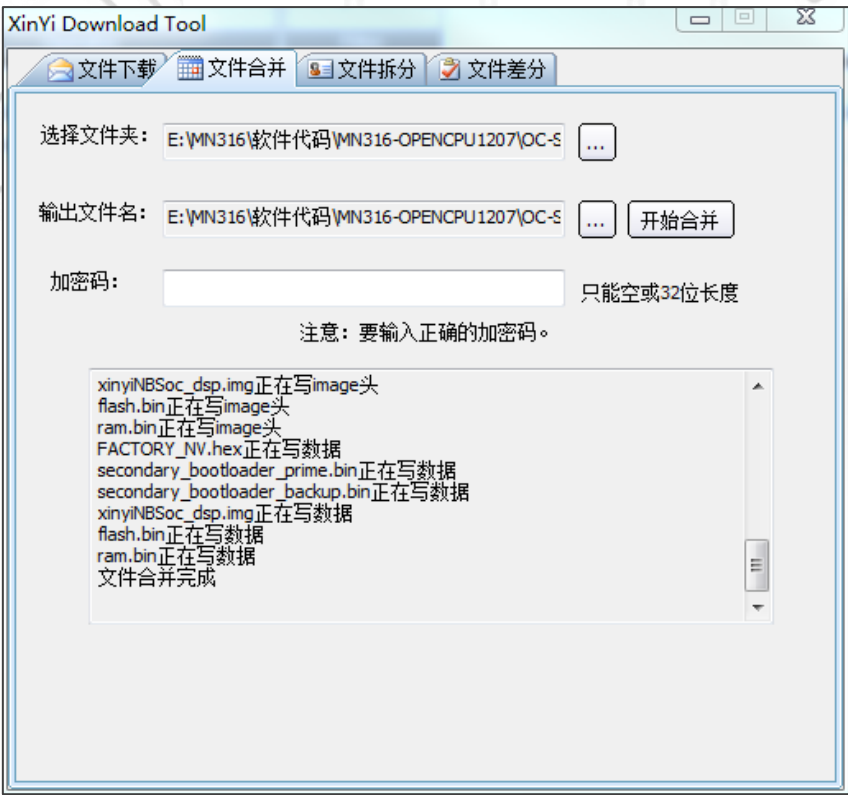


图 5-3：选择合并文件

- (2) 单击“输出文件名”右侧的“开始合并”，SDK 根目录下将生成“xinyiNBSoc.img”文件；
- (3) 切换至“文件下载”菜单，打开主串口（非 Log 口），“下载文件”选择上述生成的“xinyiNBSoc.img”文件，勾选“下载即运行”，单击“开始下载”即可开启固件下载，等待进度条到达 100% 下载完成。

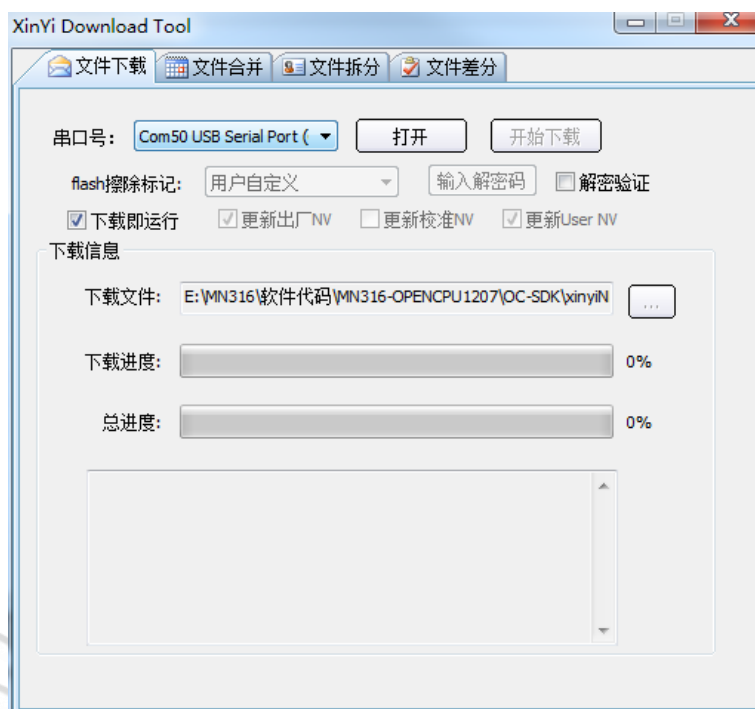


图 5-4：固件下载

## 6 DEMO 例程演示

SDK 默认提供示例程序，可直接编译烧录运行。将原始 SDK 编译烧录后，打开主串口（非 LOG 口，波特率：9600bps；停止位：1；校验位：无；数据位：8；流控：无。），示例程序运行如下。输入'H'字符，执行 3GPP 相关使用示例。

```
[2020-11-16_10:27:07:617]boot reason:0
[2020-11-16_10:27:07:641]MN316 opencpu ready
[2020-11-16_10:27:07:666]Current Version:1.2.5.2011121204_alpha
[2020-11-16_10:27:07:691]Please input cmds
[2020-11-16_10:27:23:690]H
[2020-11-16_10:27:23:812]cgmm:MN316
[2020-11-16_10:27:23:812]cgmi:CMCC
[2020-11-16_10:27:23:812]imei:861132040003187
[2020-11-16_10:27:23:837]sn:23161P0004006S500025
[2020-11-16_10:27:23:862]imsi:460047568206881
[2020-11-16_10:27:23:887]iccid:89860455261990356881
[2020-11-16_10:27:23:912]csq:31,99
[2020-11-16_10:27:23:937]cgact:1
[2020-11-16_10:27:23:937]cereg:1,0x2a2a,0x0d1a84fe,9,0,0,0x1e,0x48
[2020-11-16_10:27:23:986]cscon:1
[2020-11-16_10:27:23:986]plmn:0,46000
[2020-11-16_10:27:24:011]cell info:3688,498,1,-514,-117,-397,254
[2020-11-16_10:27:24:036]cpin:0
[2020-11-16_10:27:24:088]psm,mode:1,tau:72,active_time:33
[2020-11-16_10:27:24:138]edrx,type:5,edrx_value:2,paging_time_window:3
```

图 6-1：示例运行



# 7 SDK 开发指导

## 7.1 OpenCPU 模式启动方法

模组初次使用时，SDK 程序烧录后默认以 AT 模式运行。在 AT 模式下输入 AT+ATCLOSE 命令，模组将自动重启，进入 OpenCPU 模式；在 OpenCPU 模式下使用 cm\_open\_at()接口调用，模组将自动重启，进入 AT 模式，即可进行调试。



提供 OpenCPU 模式切换至 AT 模式的方法仅用于调试目的，请勿使用切换后的 AT 模式进行开发。

## 7.2 用户程序执行流程

SDK 预留以下函数接口作为 OpenCPU 用户程序的入口，用户可对线程属性进行配置（保证线程优先级  $\geq 10$ ），**请勿修改此函数名称，请勿在此函数中执行 Sleep 操作或其他耗时操作。**

```
int cm_main(int argc, char *argv[])
{
    UNUSED(argc);
    UNUSED(argv);
    cm_task_create(cm_task_loop,
                  "OpenCPU-DEMO",
                  1024 * 4,
                  NULL,
                  10,
                  NULL);
}
```

“cm\_task\_loop”由用户自行开发，所有用户业务均在此线程中实现。

## 7.3 增加源文件方法

在“custom/src”目录下增加源文件，并修改 custom.mk 编译文件，将源文件增加至编译文件中。同理，若需要添加自定义源文件目录，则将自定义的源文件增加至 custom.mk 中，或自行编写 user.mk 文件，通过 include 方式将此 user.mk 文件添加至 custom.mk 中。

例如需要增加源文件 user.c 在 src 目录中，custom.mk 内容增加如下。

```
###C source file on flash
C_FILES_FLASH += custom/src/cm_main.c \
                custom/src/user.c

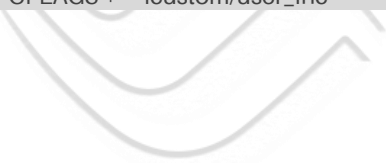
###Public include path for other module
CFLAGS += -Icustom/inc
```

在“custom/inc”目录下增加头文件，无需修改.mk 文件。若需要在 custom 目录中增加自定义头文件目录，则需要将自定义的头文件目录增加至 CFLAGS 变量中。

例如需要在 custom 目录下增加 user\_inc 头文件目录，custom.mk 内容增加如下。

```
###C source file on flash
C_FILES_FLASH += custom/src/cm_main.c \
                custom/src/user.c

###Public include path for other module
CFLAGS += -Icustom/inc
CFLAGS += -Icustom/user_inc
```



China Mobile

## 7.4 低功耗模式下的编程

OpenCPU 程序运行时默认设置睡眠锁，即模组不会自动进入深睡眠；用户业务执行完毕，调用 cm\_work\_unlock 释放睡眠锁，程序会根据 PSM 参数设置自行进入深睡眠状态。

睡眠锁的申请和释放需配对使用，配对使用时无需考虑默认设置的睡眠锁。例如：用户执行三次申请，只需执行三次释放，即可解除睡眠锁，模组将自动进入深睡眠。

模组进入深睡眠，RTC 时钟正常运行，其它部分均断电，程序不运行；模组被唤醒，程序由 main 函数重新执行。

唤醒深睡眠模组的方式包括：

- (1) WakeUp 引脚高电平持续时间 100uS 至 5S；
- (2) RTC 定时器到期；
- (3) TAU 到期。

判断程序是否是从深睡眠唤醒可在启动时使用“cm\_is\_boot\_from\_deepsleep”接口查询。使用深睡眠的典型示例流程如下。

```
usr_task()
{
    if(cm_is_boot_from_deepsleep)
    {
        data_resume();
    }
    usr_proxy();           //用户业务操作
    rtc_set();             //rtc 设置下一次唤醒时间
    cm_work_unlock();      //允许进入深睡眠
}
```

## 7.5 SDK 硬件版本说明

当前 SDK 仅适用于 MN316-DBRD-MBRH0SXX 及 MN316-XGCD-MBRH2SXX 模组硬件，请勿在其他型号的模组上使用此 SDK。

## 7.6 调试及疑难问题解决

用户在进行程序调试设计时，应注意在程序关键处加入日志信息，便于用 Log 工具实时查看程序运行状态。当出现异常问题，且 Log 信息无法定位原因时，可使用 Log 工具断言功能动态导出 Dump 文件，然后使用 TRACE32 工具对 Dump 文件进行进一步分析。Log 工具的详细使用方法见第 9 章节。

## 7.7 异常场景下 NV 的处理情况

- **PIN\_RESET**: 按键重启, 3GPP 相关的 NV 内容保存, 正常开机 Attach。
- **WDT\_RESET**: 看门狗异常重启, 属于严重异常, 擦除所有工作态 NV, 仅保持出厂 NV 有效, 重启后会进行较长时间的小区搜索驻留动作。
- **ASSERT**: 软件断言, 属于严重异常, 擦除所有工作态 NV, 仅保持出厂 NV 有效, 重启后会进行较长时间的小区搜索驻留动作。
- **SOFT\_RESET**: 由 API 触发软重启, 重启之前, 将会保存 3GPP 相关的所有 NV, 但清空平台业务相关 NV, 重启后正常开机 Attach。
- **异常断电**: 发生了异常断电, 将会擦除所有工作态 NV, 仅保持出厂 NV 有效, 会进行较长时间的小区搜索驻留动作。
- **Flash 操作异常**: 将会擦除所有工作态 NV, 仅保持出厂 NV 有效, 会进行较长时间的小区搜索驻留动作。



# 8 SDK API 分类说明

## 8.1 模组操作系统

MN316 OPENCPU SDK 已对原有操作系统进行二次封装。相关接口及数据结构等可以参考 inc 目录下 os\_api.h 头文件。

参考文档：

MN316\_OPENCPU MANUAL.chm，模块/os。

## 8.2 模组网络相关接口

MN316-OPENCPU SDK 使用 LWIP 协议栈，Socket 接口符合 POSIX 标准，用户可参照相关例程进行应用程序开发。

示例参考文档：

MN316\_OPENCPU MANUAL.chm，示例/cm\_demo\_net.c。

## 8.3 模组基础功能接口

模组基础功能接口包含基本的模组信息获取，模组控制，模组状态设置等。

参考文档：

MN316\_OPENCPU MANUAL.chm，模块/utils。

## 8.4 硬件外设接口

硬件外设接口包括 GPIO、I2C、SPI、ADC、FLASH、UART 等。使用这些外设前，必须使用 cm\_set\_standby(0) 接口关闭 Standby 功能。

外设接口参考文档：

MN316\_OPENCPU MANUAL.chm，模块/gpio，模块/i2c，模块/spi，模块/adc，模块/flash，模块/uart。

外设说明参考文档：

《MN316\_OpenCPU 资源综述》



## ■ Wireshark 数据导出

Log 工具集成了 Wireshark 空口数据导出的功能，便于用户分析上下行的网络数据。导出 Wireshark 数据的方法如下：

- (1) 在日志抓取完成后，点击“connect”关闭日志输出；
- (2) 选择“Tools”选项，点击“Export By Air Message”，选择导出文件位置即可；
- (3) 使用 Wireshark 工具打开步骤 2 导出的数据进行分析。

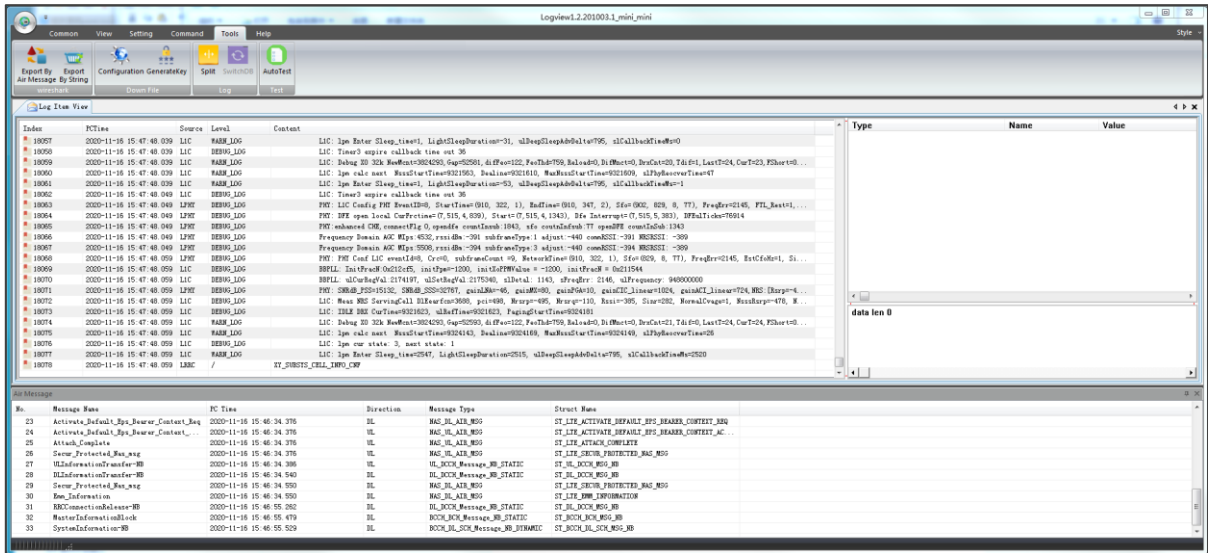


图 9-3：导出 Wireshark 数据

## ■ Dump 文件导出

在 Log 工具已连接的情况下，选择主面板“Command”选项，点击“UEAssert”即可导出当前的 Dump 信息。Dump 信息默认保存在 Log 工具“LogTmpData/Catched Data/Logviewxxxxx\_当前时间”目录下。

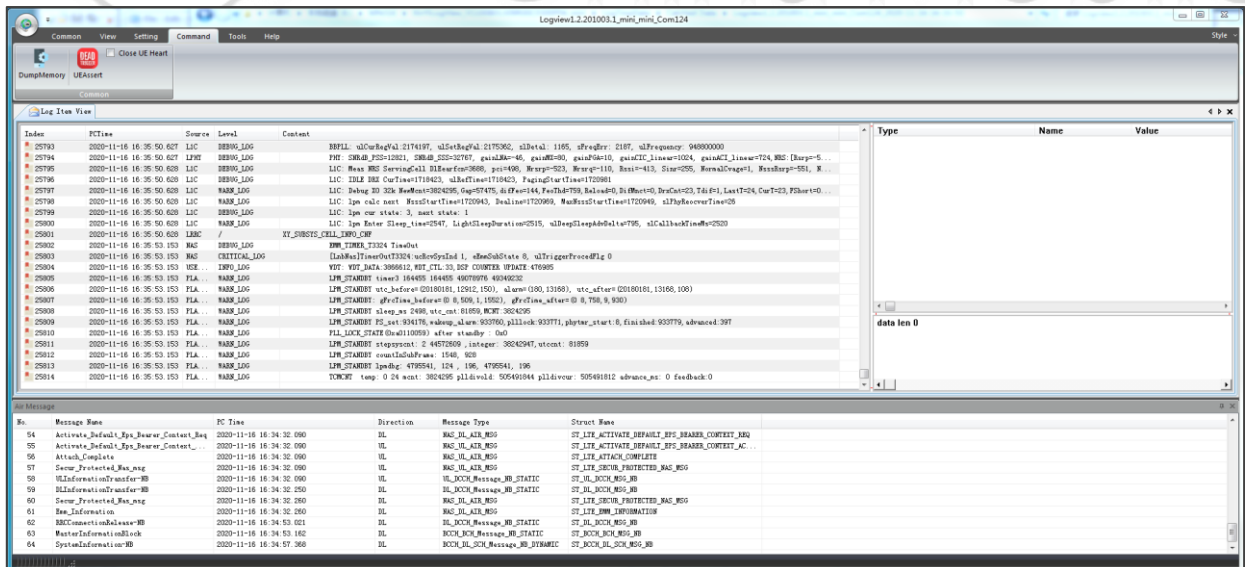


图 9-4：导出 Dump 数据

## ■ 固件下载

固件下载方法已在第 5 章描述，此处不再赘述。