

---

## N32WB452系列快速开发指南

---

### 简介

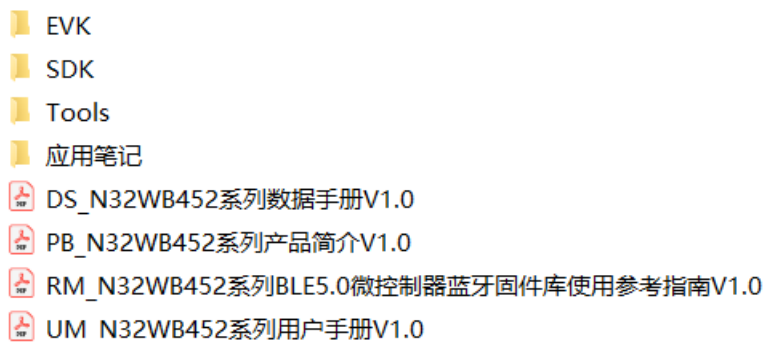
此文档的目的在于让使用者能够快速熟悉 N32WB452 系列 MCU 的开发套件以及 Keil MDK-ARM 的相关设定，以减少开发前期的准备时间，降低开发难度。

# 目录

|                                       |    |
|---------------------------------------|----|
| 目录.....                               | 2  |
| 1. 开发套件简介 .....                       | 3  |
| 2. 开发板介绍 .....                        | 5  |
| 2.1 N32WB45xL-EVB (QFN88)全功能评估板 ..... | 5  |
| 2.1.1 N32WB45xL-EVB 简介 .....          | 5  |
| 2.1.2 N32WB45xL-EVB 开发板接口定义描述 .....   | 6  |
| 2.1.3 N32WB45xL-EVB 开发板跳线功能说明 .....   | 8  |
| 3. 固件开发包介绍 .....                      | 9  |
| 3.1 FIRMWARE .....                    | 9  |
| 3.2 PROJECTS .....                    | 11 |
| 3.3 PROJECTS 框架 .....                 | 12 |
| 4. 编译环境及配置 .....                      | 13 |
| 4.1 编译环境安装 .....                      | 13 |
| 4.2 固件支持包安装 .....                     | 13 |
| 4.3 软件编译 .....                        | 13 |
| 4.5 下载和仿真 .....                       | 19 |
| 5.NS-LINK 调试工具 .....                  | 21 |
| 5.1 下载/仿真接口 .....                     | 21 |
| 5.2 虚拟串口 .....                        | 21 |
| 5.3 常见接口问题 .....                      | 21 |
| 6.申明 .....                            | 22 |

# 1. 开发套件简介

开发套件通常包含相关产品简介,数据手册,用户手册,应用笔记及参考指南等文档,以及开发评估板 EVK,软件开发包 SDK,相关下载工具及说明文档等资料。



- **产品简介:**  
通常以字母“PB”开头,芯片特性介绍文档,用于快速了解芯片资源概况、订购型号、封装信息等,用在项目开发前的芯片选型阶段。
- **数据手册:**  
通常以字母“DS”开头,芯片技术参数文档,用于了解芯片模块资源、引脚定义、电气特性、封装信息等,用在硬件设计阶段和软件开发阶段。
- **用户手册:**  
通常以字母“UM”开头,芯片各功能模块的详细介绍,包括模块内部结构、使用说明、寄存器描述等,作为软件开发阶段的主要参考文件。
- **应用笔记:**  
通常以字母“AN”开头,芯片某个功能模块的详细使用指导,包括实现原理、软件流程、核心代码实现介绍、操作说明等,作为软件开发过程中的主要参考文件。
- **参考指南:**  
通常以字母“RM”开头,芯片特定功能模块固件库的操作指南,包括固件架构、软件流程、核心代码实现介绍、操作说明等,作为软件开发过程中的核心参考文件。
- **EVK:**
  - **N32WB45xL\_EVB (QFN88)**

包含以上所有开发板资料,包括原理图/PCB/开发板使用手册说明。
- **SDK:**  
软件开发设计套件,包括 Keil DFP pack 包、固件库(底层驱动代码和参考例程)、固件库使用说明(常量定义、结构体定义、函数说明等)
- **Tools:**  
配套 PC 端工具,包括 MCU 在线下载工具软件和详细使用操作说明文档

Nations MCU Download tool V1.1.2

帮助(H)

选择设备

接口: USART1

设备: COM10 关闭串口

波特率: 115200 数据位: 8 停止位: 1

下载操作

分区: USER1认证下载

☐ 分区认证使能 ☐ 加密下载使能 ☐ 下载文件为密文

文件路径: 浏览

起始地址: 0x08000000 ☒ CRC校验 下载

其它操作

获取芯片信息 擦除

更新密钥 配置分区

配置选项字节 跳转

复位

清空显示 退出系统 下载 隐藏<<

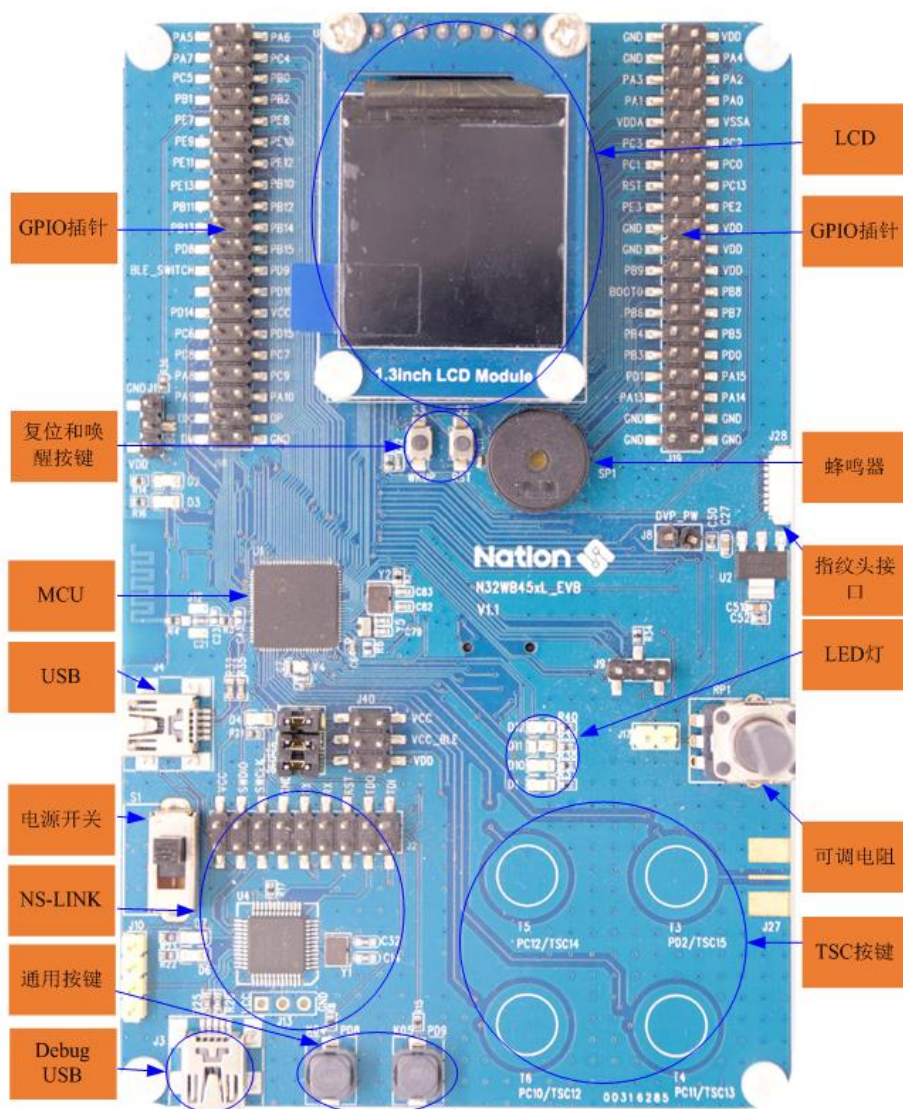
## 2. 开发板介绍

### 2.1 N32WB45xL-EVB (QFN88)全功能评估板

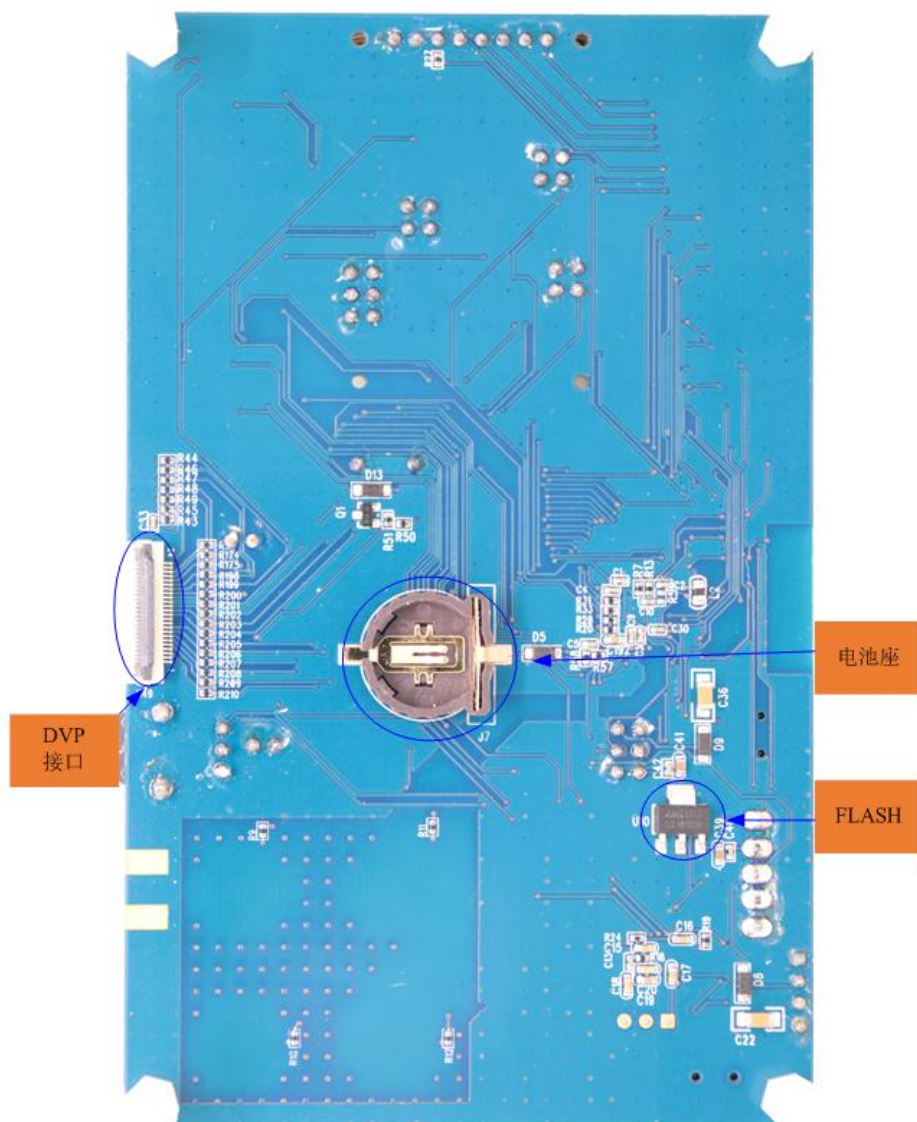
#### 2.1.1 N32WB45xL-EVB 简介

N32WB45xL-EVB 为国民技术 BLE 5.0 SOC 芯片 N32WB452LEQ6（QFN88）的全功能评估板，开发板功能结构图说明如下：

N32WB45xL-EVB 开发板正面图



N32WB45xL-EVB 开发板反面图



## 2.1.2 N32WB45xL-EVB 开发板接口定义描述

### 1) 开发板的供电

开发板可选用USB接口（J4）或Dedug USB接口（J3）供电，通过开关S1连接到3.3V LDO输入口，输出端分为三路，第一路为给MCU供电的电源VCC\_MCU，接口为J6；第二路为给除MCU外其余外设芯片供电的电源VCC33，接口为J5；第三路为给蓝牙预留的电源VCC\_BLE，接口为J15。

### 2) USB接口（J4）

采用MicroUSB接口（J4），连接MCU DP DM，可用于USB接口通讯。

### 3) NSLINK接口 (J2&J20)

NSLINK接口可以用于程序下载调试,支持两种模式SWD/JTAG。J20为SWD模式下载口,J2为JTAG模式下载口。

### 4) 复位和唤醒按键 (S2, S3)

S2, S3分别为复位按键和唤醒按键,分别连接芯片的NRST管脚和PA0-WKUP管脚,用于芯片复位和唤醒功能。

### 5) 通用按键 (K04, K05)

K04, K05分别连接芯片PD8, PD0管脚,作为通用GPIO按键。

### 6) BOOT (J9, J11)

J9和J11分别为MCU主芯片的BOOT0和BOOT1引脚跳线连接到预留的上下拉电阻。

### 7) 电池座 J7(BAT)

电池座可放一颗CR1220电池,连接到芯片VBAT管脚提供电源。

### 8) GPIO口 (J14、J16)

芯片GPIO接口全部引出,插针并预留GND。接口的具体定义参见原理图。

### 9) FLASH芯片 (U7)

开发板板载SPI flash芯片,位号U7,通过J30、J32接口连接MCU SPI管脚。

### 10) 温湿度传感器 (U8)

开发板板载温湿度传感器HDC2010,位号U8,通过I2C2与MCU通讯。

### 11) 触控按键TSC (T3、T4、T5、T6)



开发板板载四路TSC，位号T3、T4、T5、T6，可供客户直接使用调试。

## 12) G-SENSOR (U5)

开发板板载G-SENSOR芯片QMA7981，位号U5，通过I2C1与MCU通讯。

## 13) DVP接口 (J149)

开发板板载DVP接口，位号J149，24pin连接器，通过软排线连接摄像头。

## 14) LCD (U6)

开发板板载LCD屏幕，位号U6，可供客户直接使用调试。

## 2.1.3 N32WB45xL-EVB 开发板跳线功能说明

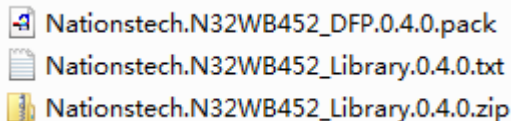
开发板跳线功能说明

| No. | 跳线位号      | 跳线功能         | 使用说明  |
|-----|-----------|--------------|---|
| 1   | J5、J6、J15 | 3.3V 供电跳线    | J6: MCU 供电接口<br>J5: 其余外设供电接口<br>J15: BLE 预留供电接口 |
| 2   | J30、J32   | SPI FLASH 跳线 | 若使用 FLASH，需连接 J30、J32                           |
| 3   | J8        | DVP 供电接口     | 若使用 DVP，需连接 J8                                  |
| 4   | J2、J20    | NSLINK 接口    | J1: SWD 模式下载口<br>J2: JTAG 模式下载口                 |
| 5   | J9,J11    | BOOT 跳线      | J9: BOOT0;<br>J11: BOOT1。                       |
| 6   | J17       | 可调电阻跳线       | 若使用 ADC 功能，连接该跳线                                |

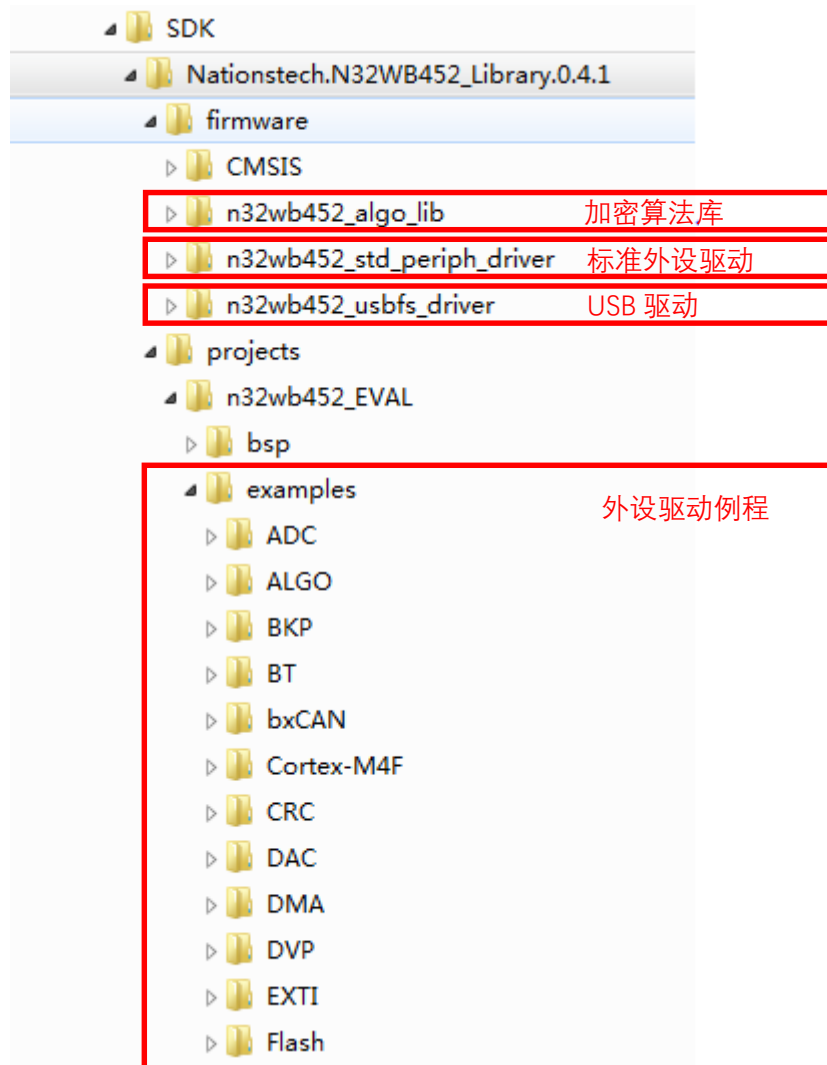


## 3. 固件开发包介绍

通常 SDK 提供一个 Keil 的 PACK 包，一个 chm 格式的 SDK 说明文档，以及 SDK 源代码压缩文件包。



SDK 目录下是一个以固件库版本命名的文件，其中目录结构及介绍如下：



### 3.1 Firmware

- **CMSIS:**

即微控制器软件接口标准，是 Cortex-M 处理器系列的与供应商无关的硬件抽象层，CMSIS 提供了内核与外设、实时操作系统和中间设备之间的通用接口。

- 包含了用来访问内核的寄存器设备的名称定义，地址定义和配置函数。该接口包括调试通道定义。
- 提供片上所有外设的定义，包括所有外设寄存器头文件、启动文件、系统初始化模板文件。

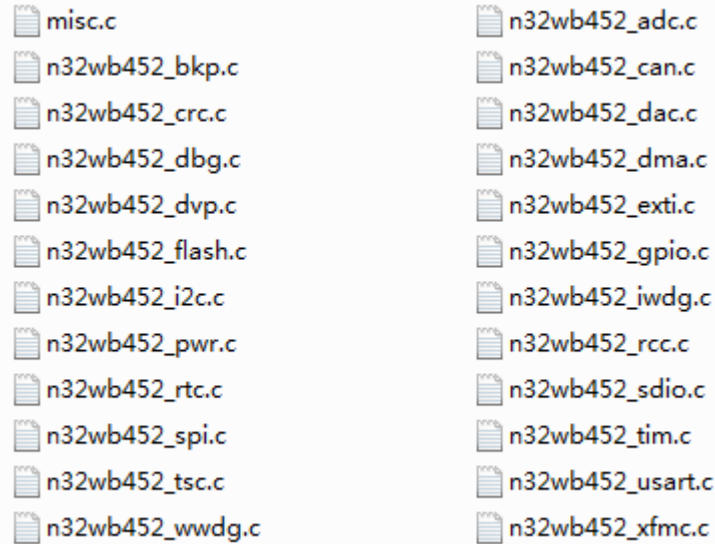
➤ DSP库，提供优化的信号处理算法。

● **n32WB452 系列\_algo\_lib:**

➤ 算法库文件，包括：加密算法、HASH 算法、随机数算法等。

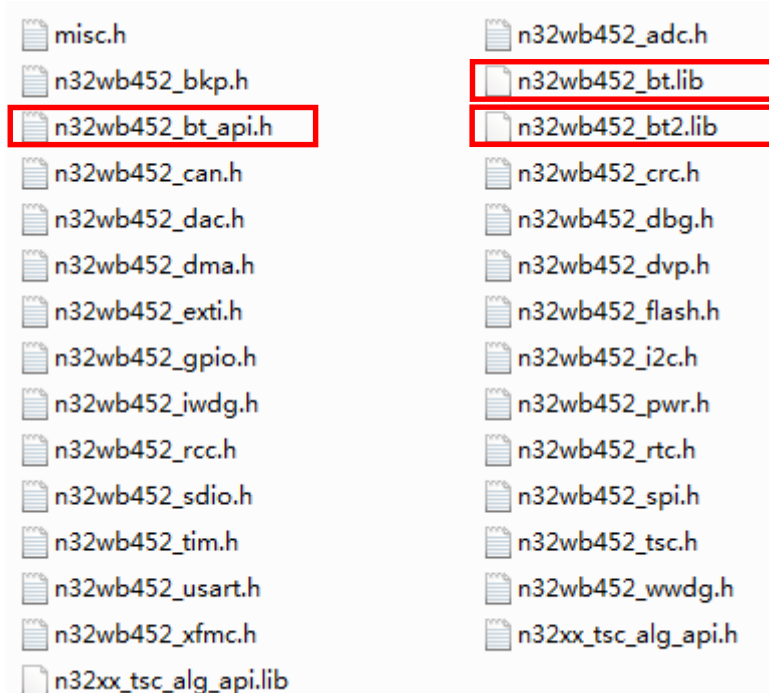
● **n32WB452 系列\_std\_periph\_driver:**

➤ 芯片外设的标准驱动函数，包括.c 的源文件和.h 的头文件。用户可移植到项目中，快速完成对某个外设模块的使用。



➤ N32WB452 系列带有蓝牙协处理器，Cortex-M 主处理器与蓝牙协处理器之间通过 QSPI 通信，蓝牙相关处理程序包含在 n32wb452\_bt.lib 中，用户可以通过 n32wb452\_bt\_api.h 里提供的 api 接口调用。另外还提供了 n32wb452\_bt2.lib，只封装了部分蓝牙相关程序，开放大部分的程序给用户使用。

（注意：n32wb452\_bt.lib 和 n32wb452\_bt2.lib 不能同时使用）



- **n32WB452 系列\_usbfs\_driver**

- USB 设备驱动库文件，包括.c 的源文件和.h 的头文件。在开发 USB 设备时的必要文件，完成 USB 底层协议栈的搭建，用户只需要关心应用层的开发。

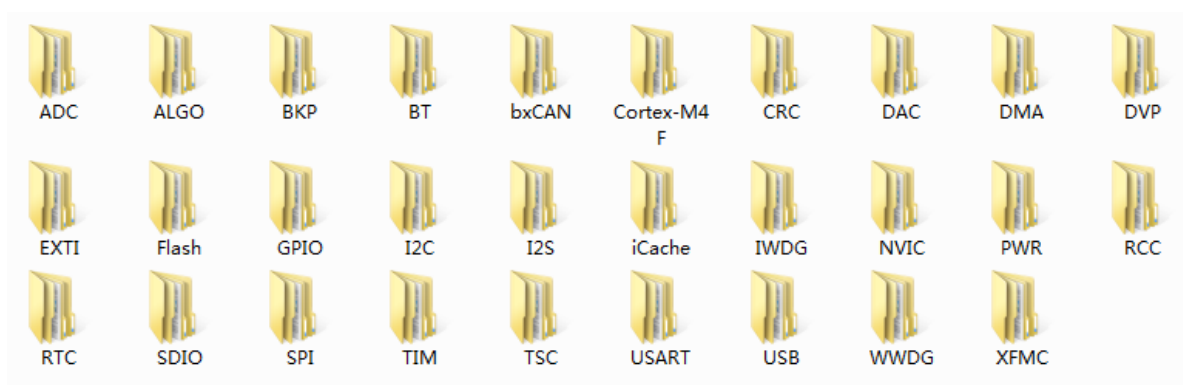
## 3.2 Projects

- **BSP:**

- 包含了调试串口打印功能文件/LOG，用于在调试过程中打印各种调试信息

- **Examples:**

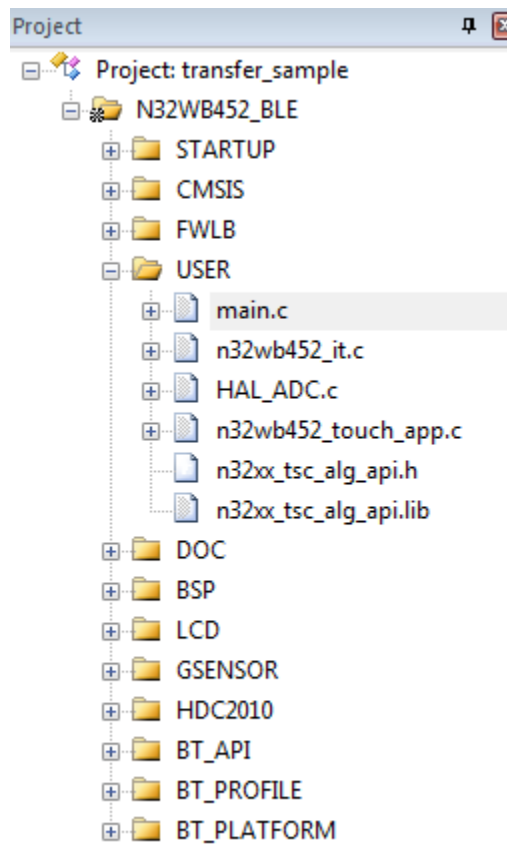
- 包含各个外设功能模块的例程项目，是开发工程师重点关注的部分，实现了每个外设模块的基本应用开发，用户可通过这些例程项目快速的了解芯片用法、编程技巧、模块移植。



一般情况下用户可根据实际需求选择一个应用最为接近的项目作为初始代码进行开发。

- 特别需要说明的是，BT2 例程里包含两个例程 transfer 和 transfer2，为用户提供蓝牙功能的使用方法。transfer 例程使用的是 n32wb452\_bt.lib，用户可以通过 n32wb452\_bt\_api.h 里提供的 api 接口函数进行蓝牙功能开发。Transfer2 例程使用的是 n32wb452\_bt2.lib，Transfer2 可以使用 n32wb452\_bt\_api.h 提供的 api 接口函数开发蓝牙功能，也可以使用例程中开放的蓝牙代码灵活修改。

### 3.3 Projects 框架



- **LCD:**
  - 包含了开发板搭载的 LCD 显示屏驱动例程
- **GSENSOR:**
  - 包含了开发板搭载的加速度传感器型号的例程
- **GSENSOR:**
  - 包含了开发板搭载的温湿度传感器型号的例程

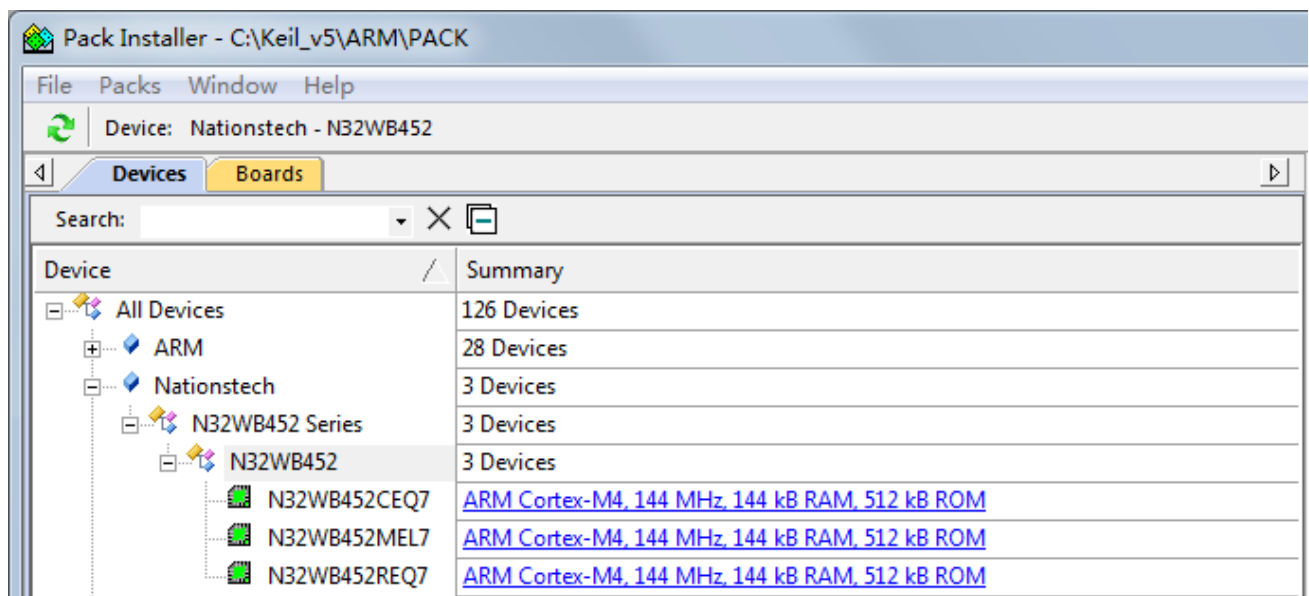
## 4. 编译环境及配置

### 4.1 编译环境安装

请安装 KEIL MDK-ARM 开发环境，版本要求为 V5.26 以上。MDKV5 之后的版本的 CMSIS 架构可支持线上更新功能，未来 Nations 会在线上放置最新的 CMSIS 版本，用户可直接通过 Keil 环境的 Pack Installer 进行更新。（如编译代码量超过 32K，需要购买 KEIL 产品许可密匙）

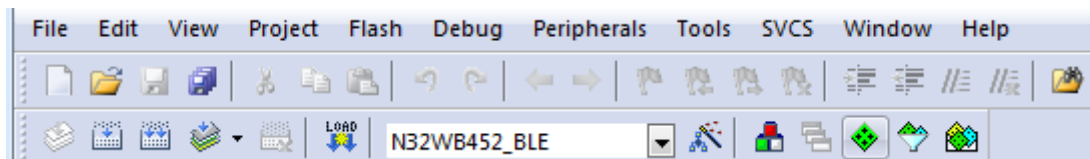
### 4.2 固件支持包安装

请安装固件支持包/DFP [Nationtech.N32WB452\\_DFP.0.4.0.pack](#)，安装成功后，在 KEIL 的 Pack Installer 界面中可以看到支持的设备列表



### 4.3 软件编译

在 SDK 中选择一个项目工程并打开，点击全局编译

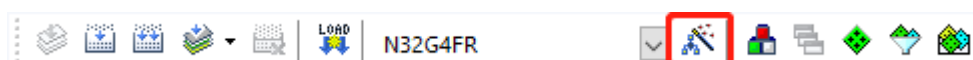


默认无修改的项目都可以编译通过

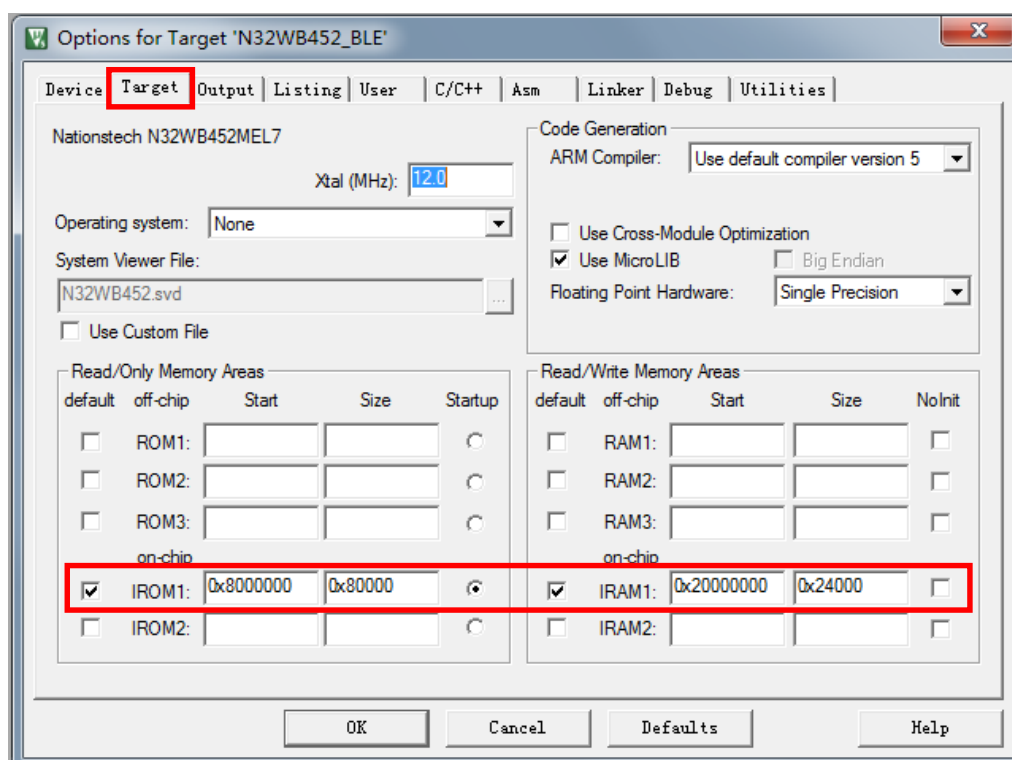
```
compiling lcd_gui.c...
compiling font24.c...
compiling lcd_drv.c...
compiling gsensor.c...
compiling i2c_drv.c...
compiling hdc_i2c_drv.c...
compiling HDC2010.c...
linking...
Program Size: Code=98192 RO-data=18500 RW-data=744 ZI-data=133352
FromELF: creating hex file...
".\Objects\ble_transfer.axf" - 0 Error(s), 0 Warning(s).
Build Time Elapsed: 00:00:21
```

## 4.4 编译环境配置

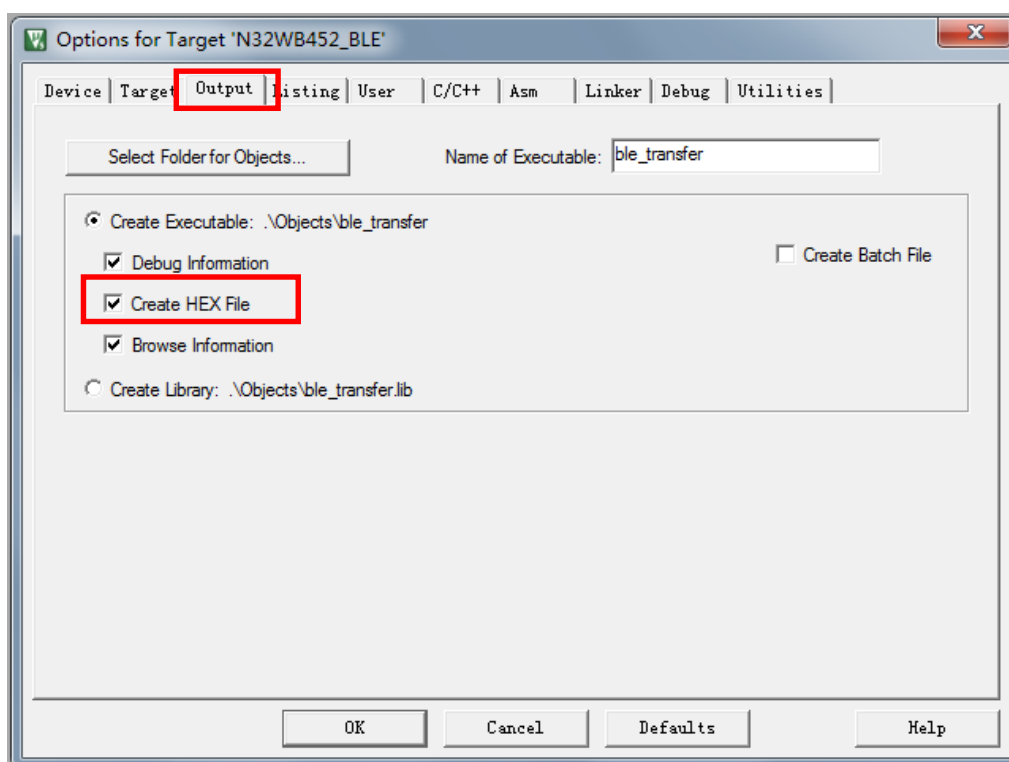
点击魔术棒，打开 OPTION 设定界面



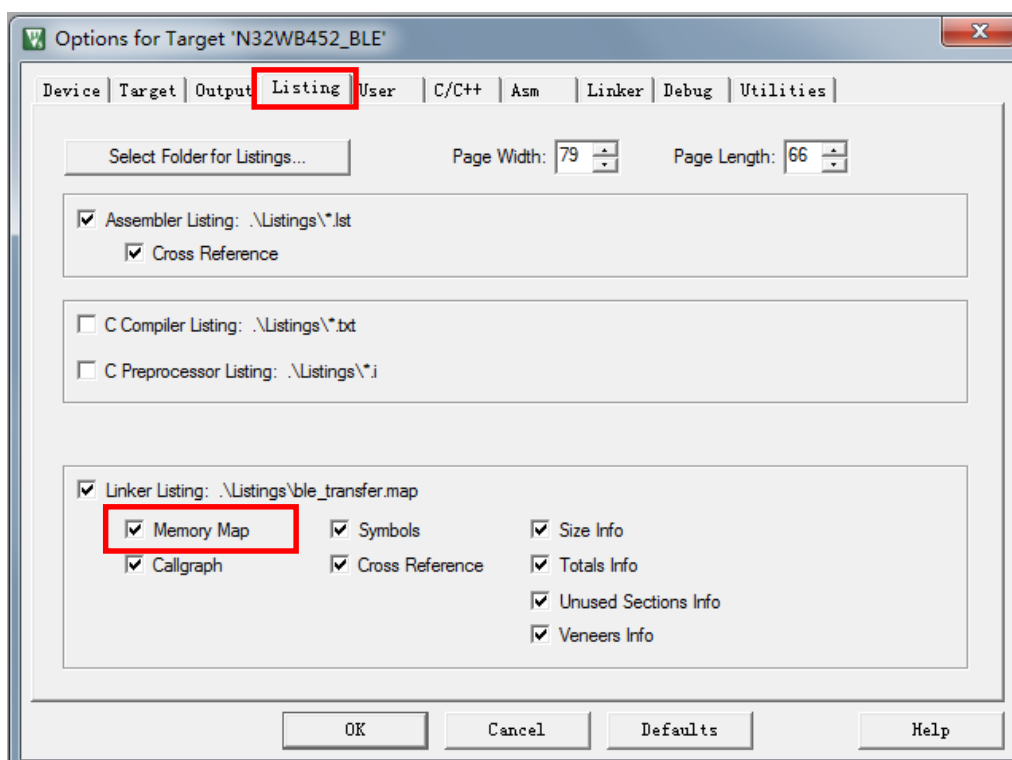
### 4.4.1 在 Target 界面中设定 ROM 和 RAM 的大小（根据不同型号设定）



#### 4.4.2 在 Output 界面中勾选 hex 输出文件



#### 4.4.3 Memory Map 文件生成

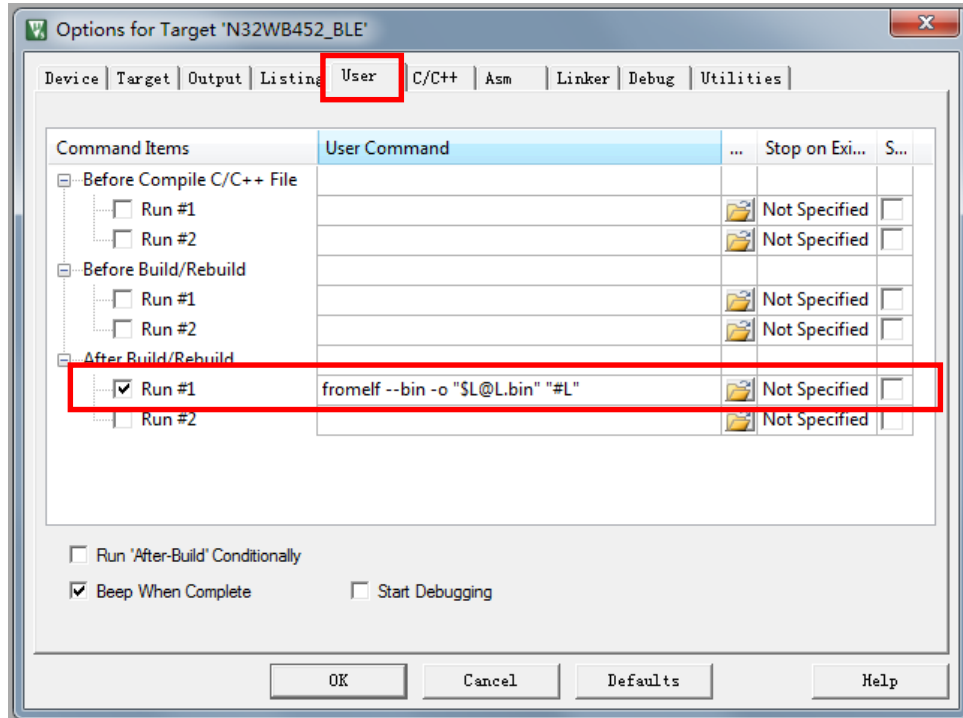


.map 文件记录了程序存储空间使用情况，可方便查看变量、常量、函数存放的地址空间和大小，如下图

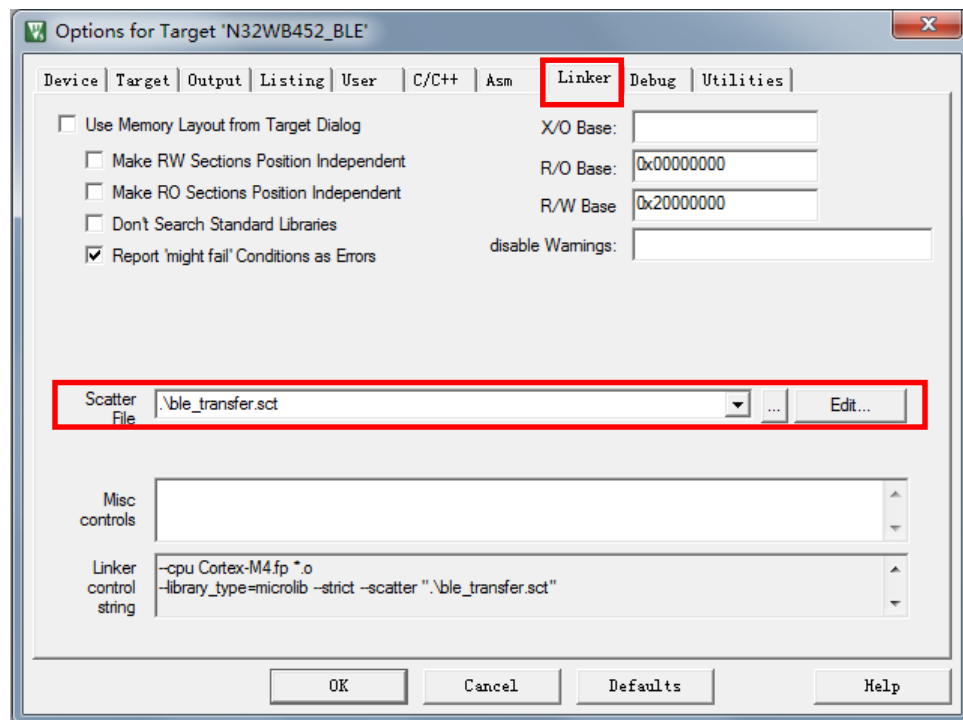


|                       |            |            |     |                      |
|-----------------------|------------|------------|-----|----------------------|
| adc_value_get         | 0x080038e9 | Thumb Code | 58  | main.o(.text)        |
| hdc_i2c_gpio_init     | 0x08003923 | Thumb Code | 88  | main.o(.text)        |
| hdc_i2c_gpio_deinit   | 0x0800397b | Thumb Code | 68  | main.o(.text)        |
| hdc2010_init          | 0x080039bf | Thumb Code | 46  | main.o(.text)        |
| hdc2010_data_read     | 0x080039ed | Thumb Code | 106 | main.o(.text)        |
| mainboard_enter_stop2 | 0x08003a57 | Thumb Code | 32  | main.o(.text)        |
| mainboard_exit_stop2  | 0x08003a77 | Thumb Code | 28  | main.o(.text)        |
| main                  | 0x08003a93 | Thumb Code | 798 | main.o(.text)        |
| NMI_Handler           | 0x08003dd5 | Thumb Code | 2   | n32wb452_it.o(.text) |
| HardFault_Handler     | 0x08003dd7 | Thumb Code | 68  | n32wb452_it.o(.text) |
| MemManage_Handler     | 0x08003e1b | Thumb Code | 4   | n32wb452_it.o(.text) |

4.4.4 如需要生成 bin 格式的烧录文件，将 **fromelf --bin -o "\$L@L.bin" "#L"** 拷贝至如下图：



4.4.5 Linker Scatter File 设置（根据项目实际需求设置）

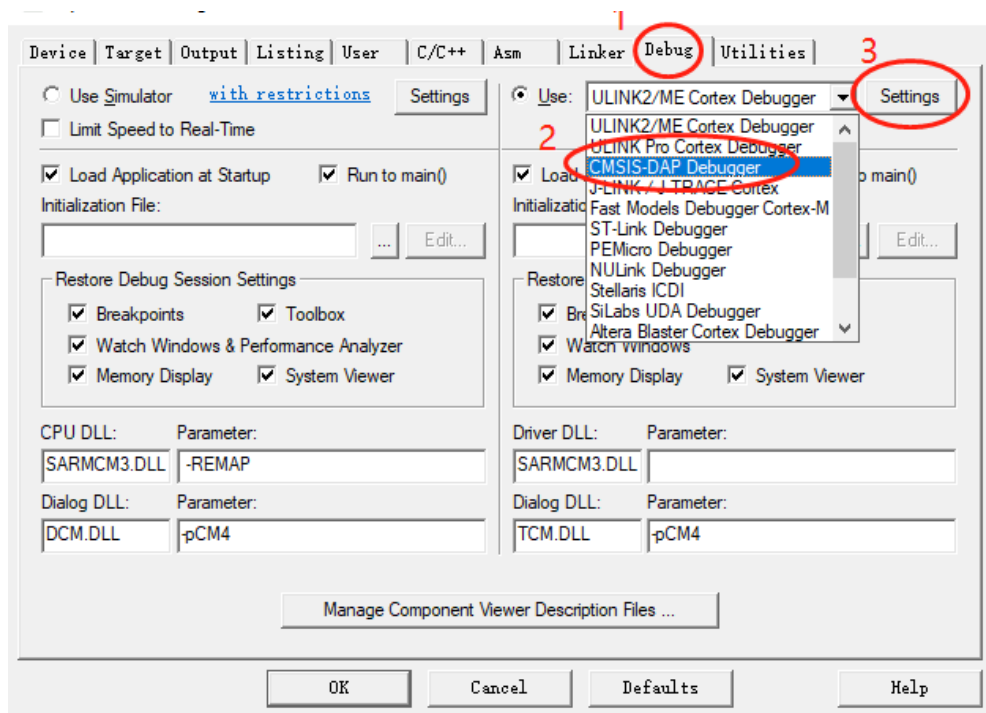


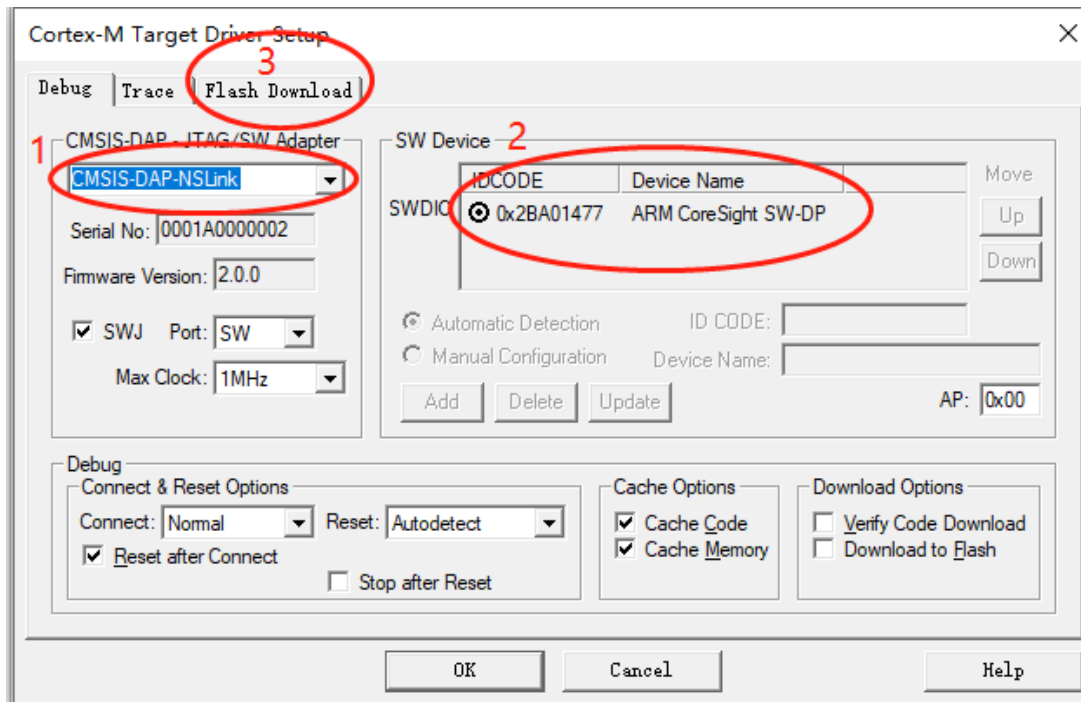
.scf 是一个文本文件，通过编写一个分散加载文件来指定 ARM 连接器在生成映像文件时如何分配 RO,RW,ZI 等数据的存放地址。一般情况下 ARM 连接器会按照默认的方式来生成映像文件。

某些场合，我们希望把某些数据放在指定的地址位置

- 1) 存在复杂的地址映射：例如代码和数据需要分开存放在多个区域(一般用在引导代码)。
- 2) 存在多个存储器类型：包含 Flash, ROM, SDRAM 等。根据代码与数据的特性把他们发在不同的存储器中。
- 3) 利用 Scatter file 实现把某个函数放在固定地址，不管其应用程序是否已经改变或重新编译。
- 4) 内存映射的 IO：利用 scatter file 可以实现把某个数据段放在精确的地址位置。

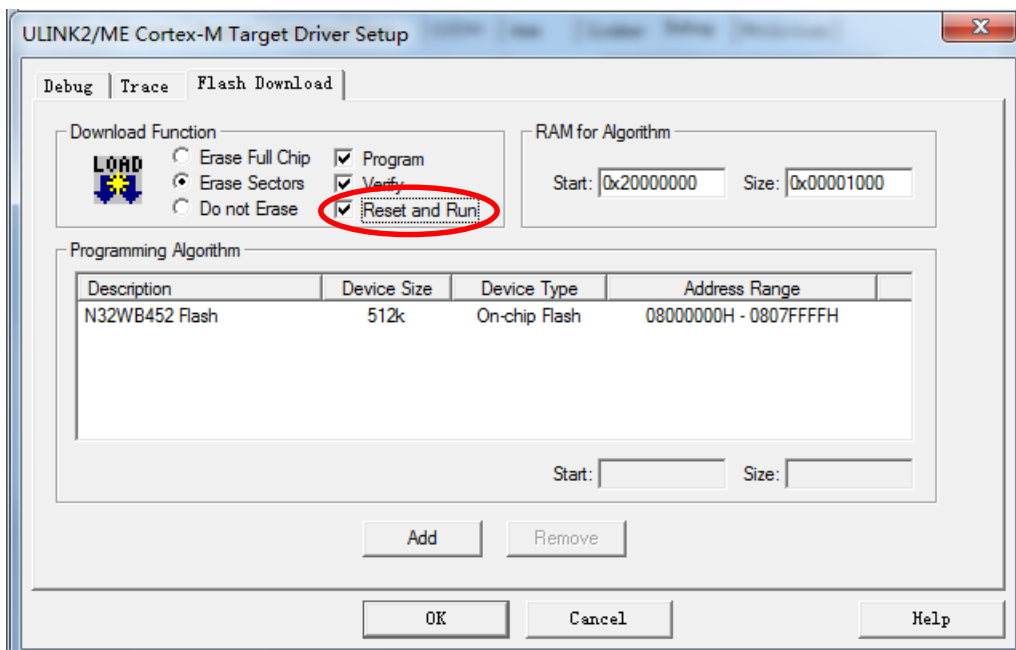
#### 4.4.6 在 Dubg 界面中进行仿真相关设置





- 1) 选择 NSLINK 仿真器（如连接其他 LINK，则选择对应的 LINK 种类）
- 2) 如连接成功，在位置 2 出可以显示出仿真器的设备序列号
- 3) 在 Flash Download 界面中勾选复位运行

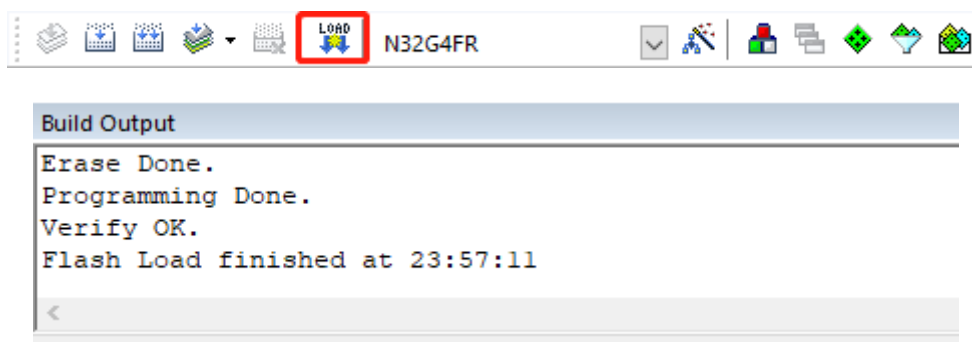
击 Flash Download 设置,勾选“Rest and Run”，代码下载后自动复位和运行



到这里编译环境设置完成。

## 4.5 下载和仿真

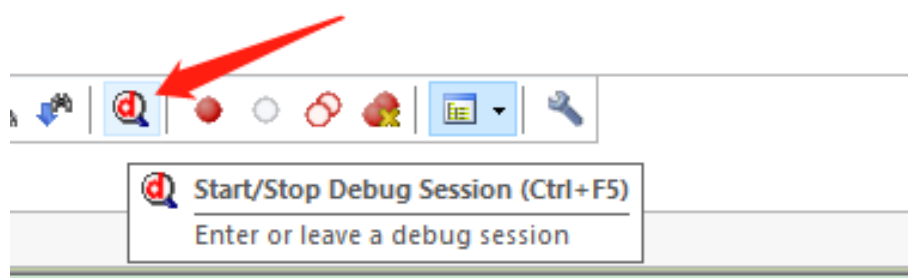
### 4.5.1 下载固件



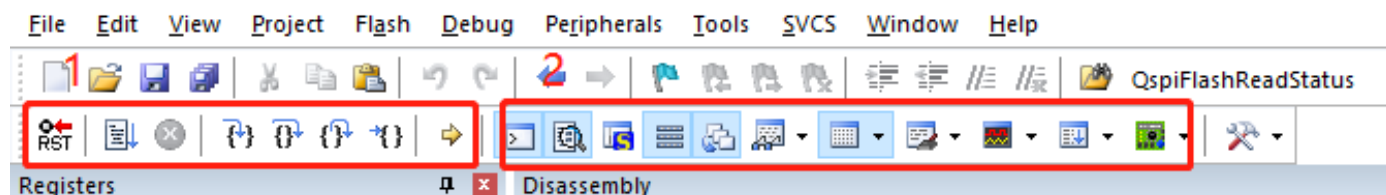
如上图所示，此时开发板/调试板已经在运行程序。

### 4.5.2 仿真调试

当仿真器连接且程序编译通过，点击仿真按钮



进入下图仿真界面，工具栏中出现仿真工具按钮



- 1) 执行仿真步骤，控制执行过程（单步、逐行、全速、复位等）
- 2) 监控仿真状态，如寄存器、变量、FLASH 等信息

### KEIL 常用仿真工具



点击按钮 1，打开变量查看窗口

| Watch 1 |            |                |
|---------|------------|----------------|
| 变量地址和值  |            |                |
| Name    | Value      | Type           |
| src     | 0x20001400 | unsigned in... |
| [0]     | 0x0000009F | unsigned int   |
| [1]     | 0x000000FF | unsigned int   |
| [2]     | 0x000000FF | unsigned int   |
| [3]     | 0x000000FF | unsigned int   |
| [4]     | 0x00000000 | unsigned int   |
| [5]     | 0x00000001 | unsigned int   |
| [6]     | 0x00000002 | unsigned int   |
| [7]     | 0x00000003 | unsigned int   |

点击按钮 2，打开存储器查看窗口

| Memory 1    |  |
|-------------|--|
| Address:    | 0x000000   |
| 0x00000000: | 10 15 00 20 AD 01 00 08 D5 22 00 08 D7 22 00 08 DB 22 00 |
| 0x0000001D: | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 E7 22 00 08 |
| 0x0000003A: | 00 08 EB 22 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 |
| 0x00000057: | 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 |
| 0x00000074: | C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 |
| 0x00000091: | 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 |
| 0x000000AE: | 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 01 00 08 C7 |

点击按钮 3 打开寄存器查看窗口

| Property      | Value      |
|---------------|------------|
| GPIOx_PL_CFG  | 0xBBBB0000 |
| GPIOx_PH_CFG  | 0x88800000 |
| GPIOx_PID     | 0x0000C010 |
| GPIOx_POD     | 0x0000A000 |
| GPIOx_PBSC    | 0          |
| GPIOx_PBC     | 0          |
| GPIOx_PLOC... | 0          |
| GPIOx_DS_CFG  | 0          |
| GPIOx_SR_CFG  | 0x0000FFFF |

通过以上方法能够在仿真过程中直观的观察程序运行过程和状态，加快开发进度

## 5.NS-LINK 调试工具

NS-LINK 采用 ARM 公司的标准软件调试访问接口：CMSIS-DAP，支持程序下载、仿真和串口调试功能。Nations 所有的开发板中都集成了一个 NS-LINK 电路，支持 SWD 调试。

### 5.1 下载/仿真接口

只需要一条 USB MINI 连接线使用者就能够方便的下载和调试程序。可以通过跳线帽将 SWD 接口与开发板核心 MCU 芯片连接，也可通过杜邦线连接其他调试版，作为一个独立的 LINK 调试工具使用。

### 5.2 虚拟串口

NS-LINK 支持 USB 串口功能，默认与芯片 **USART2** 通过跳线帽连接，用户可通过包含 BSP 文件夹中的 LOG 文件调用下列输出函数方便的输出串口调试信息

```
#define log_info(...)
#define log_warning(...)
#define log_error(...)
#define log_debug(...)
#define log_init()
```

### 5.3 常见接口问题

- 连接调试 USB，无任何设备枚举  
可能是 NS-LINK 控制芯片未烧录固件，请联系 Nations 技术支持团队解决
- 跳线帽没有正确连接导致接口无法工作  
在"EVT"文件夹中查阅开发板使用文档，按照要求连接
- 串口无法打印信息  
可能 1：代码将 USART2 作为其他用途，导致无法输出 LOG 打印信息  
可能 2：串口跳线帽没有正确连接  
可能 3：串口通讯堵塞，需要重新打开串口助手
- 下载程序后无法继续再下载  
可能 1：代码中将 SWDIO/SWDCLK 用作其他用途，导致仿真接口失效。一般情况下不建议使用仿真引脚用作其他功能设计  
可能 2：芯片进入低功耗掉电模式，此时仿真接口失效，需要在唤醒状态下下载。

## 6. 申明

国民技术股份有限公司（以下简称国民技术）保有在不事先通知而修改这份文档的权利。国民技术认为提供的信息是准确可信的。尽管这样，国民技术对文档中可能出现的错误不承担任何责任。在购买前请联系国民技术获取该器件说明的最新版本。对于使用该器件引起的专利纠纷及第三方侵权国民技术不承担任何责任。另外，国民技术的产品不建议应用于生命相关的设备和系统，在使用该器件中因为设备或系统运转失灵而导致的损失国民技术不承担任何责任。国民技术对本手册拥有版权等知识产权，受法律保护。未经国民技术许可，任何单位及个人不得以任何方式或理由对本手册进行使用、复制、修改、抄录、传播等。