

---

## N32WB452 系列快速开发指南

---

### 简介

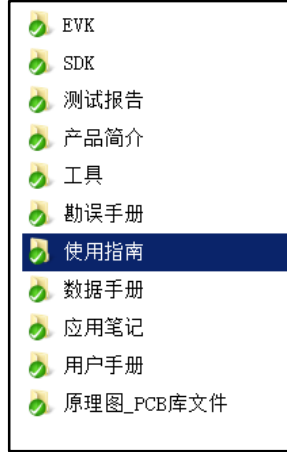
此文档的目的在于让使用者能够快速熟悉 N32WB452 系列 MCU 的开发套件以及 Keil MDK-ARM 的相关设定，以减少开发前期的准备时间，降低开发难度。

# 目录

<b>1 开发套件简介 .....</b>	<b>1</b>
<b>2 开发板介绍 .....</b>	<b>2</b>
2.1 N32WB45xL_EVB (QFN88)全功能评估板 .....	2
2.1.1 N32WB45xL_EVB 简介 .....	2
2.1.2 N32WB45xL_EVB 开发板接口定义描述 .....	3
2.1.3 N32WB45xL_EVB 开发板跳线功能说明 .....	4
<b>3 固件开发包介绍 .....</b>	<b>6</b>
3.1 Firmware .....	6
3.2 Projects .....	7
3.2.1 Projects 框架 .....	8
<b>4 编译环境及配置 .....</b>	<b>10</b>
4.1 编译环境安装 .....	10
4.2 固件支持包安装 .....	10
4.3 软件编译 .....	10
4.4 编译环境配置 .....	10
4.5 下载和仿真 .....	15
4.5.1 下载固件 .....	15
4.5.2 仿真调试 .....	15
<b>5 NS-LINK 调试工具 .....</b>	<b>18</b>
5.1 下载/仿真接口 .....	18
5.2 虚拟串口 .....	18
5.3 常见接口问题 .....	18
<b>6 版本历史 .....</b>	<b>19</b>
<b>7 声明 .....</b>	<b>20</b>

# 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。

- **使用指南:**

通常以字母“UG”开头, 芯片特定功能模块固件库的使用指南, 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)。

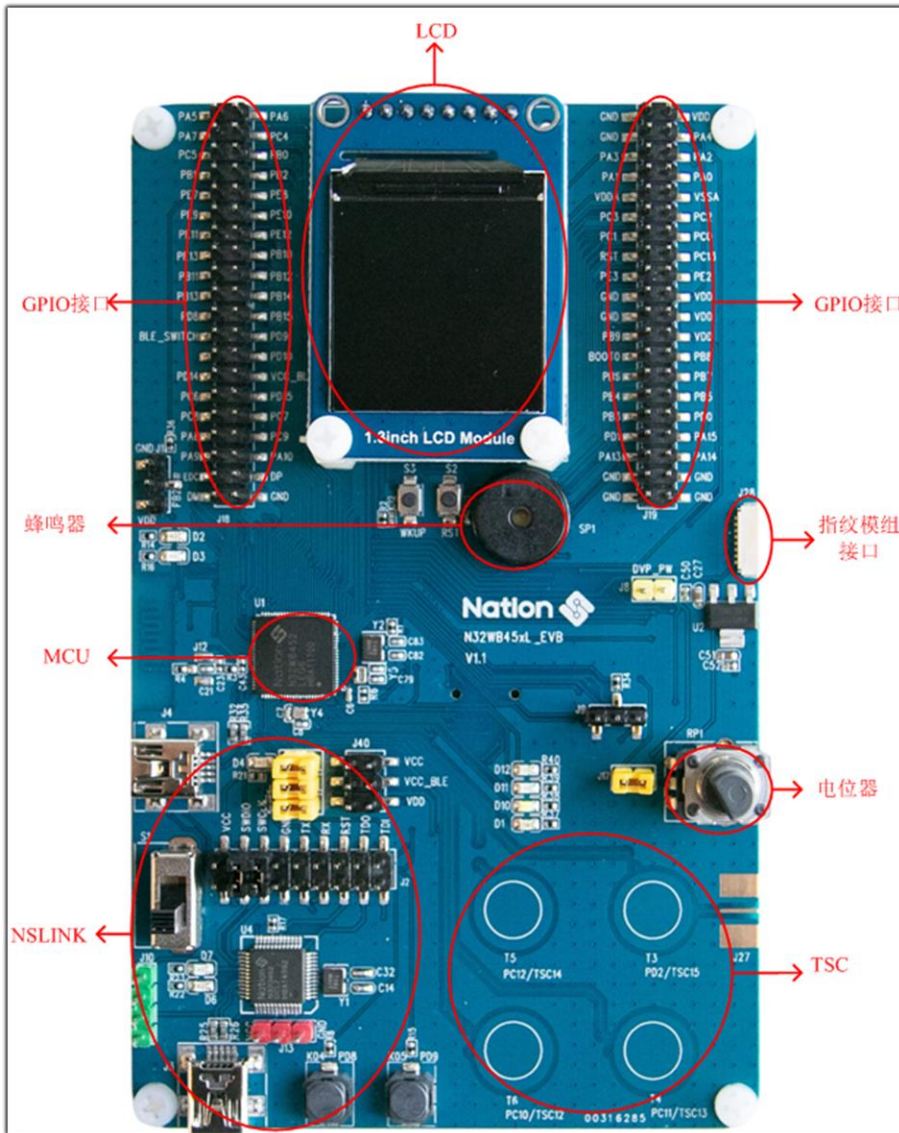
## 2 开发板介绍

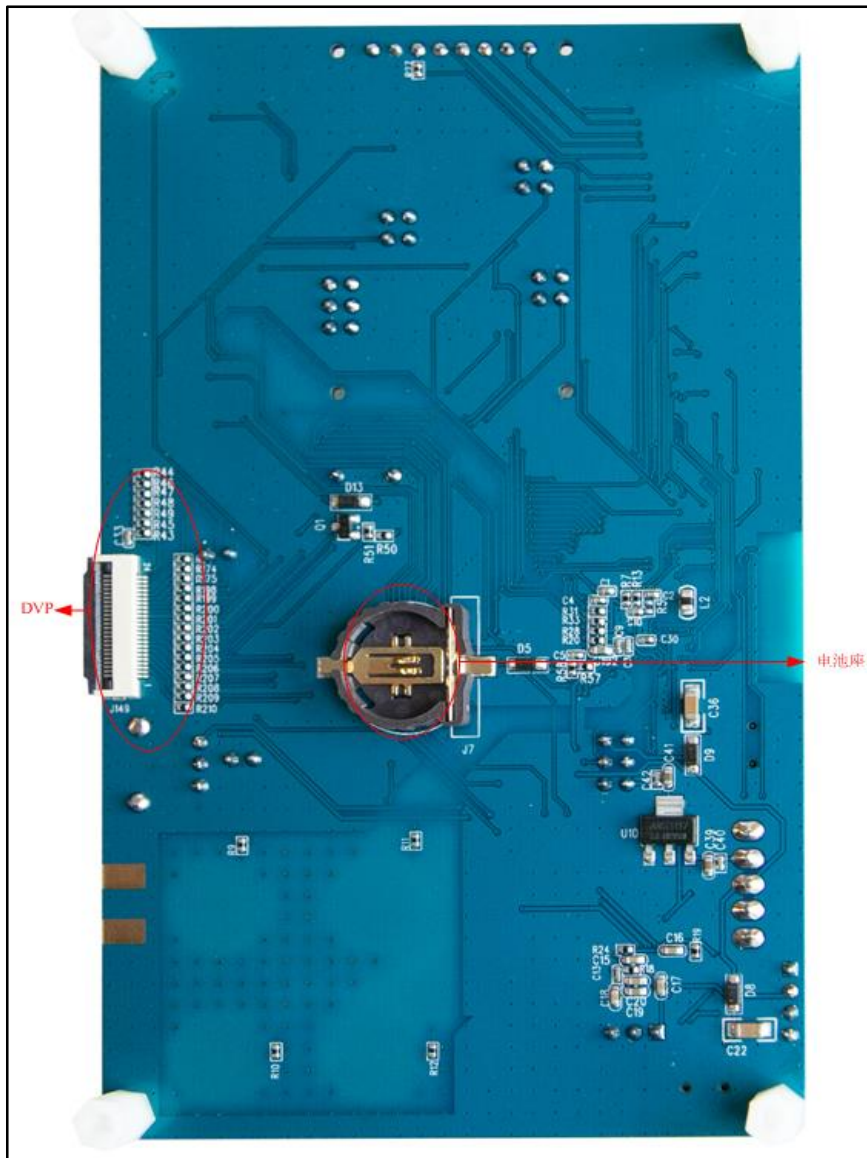
### 2.1 N32WB45xL\_EVB (QFN88)全功能评估板

#### 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：

图 2-1 N32WB45xL\_EVB V1.1 开发板正面图



**图 2-2 N32WB45xL\_EVB V1.1 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（J3或J4）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VCC3.3，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（J3）

采用MicroUSB接口（J4），连接板载NSLINK芯片（U4），可用于程序下载调试，使用时注意连接J2、J20跳线。

#### 3) USB接口（J4）

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

#### 4) 调试接口（J2&J20）

用于连接其他调试工具进行程序下载调试，支持两种模式SWD/JTAG。J20为SWD模式下载口，使用时需要断开与NSLINK芯片的连接。J20&J2 可作为JTAG模式下载口。

#### 5) 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。

#### 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、J19、J16、J18)

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J1、J21、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) G-SENSOR (U5)

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

#### 12) 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。

#### 13) LCD (U6)

开发板板载LCD屏幕，位号U6，可供客户直接使用调试。

#### 14) 指纹模组 (J28)

开发板板载一路指纹模组接口，位号 J28，可外接指纹模组供客户调试用。

### 2.1.3 N32WB45xL\_EVB 开发板跳线功能说明


表 2-1 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 J5: 其余外设供电接口 J15: BLE 预留供电接口
2	J1、J21、J30、J32	SPI FLASH 跳线	若使用 FLASH, 需连接 J1、J21、J30、J32
3	J8	DVP 供电接口	若使用 DVP, 需连接 J8

No.	跳线位号	跳线功能	使用说明
4	J2、J20	调试接口	J20: SWD 模式下载口 J20&J2: JTAG 模式下载口
5	J9,J11	BOOT 跳线	J9: BOOT0 J11: BOOT1
6	J17	可调电阻跳线	若使用 ADC 功能, 连接该跳线

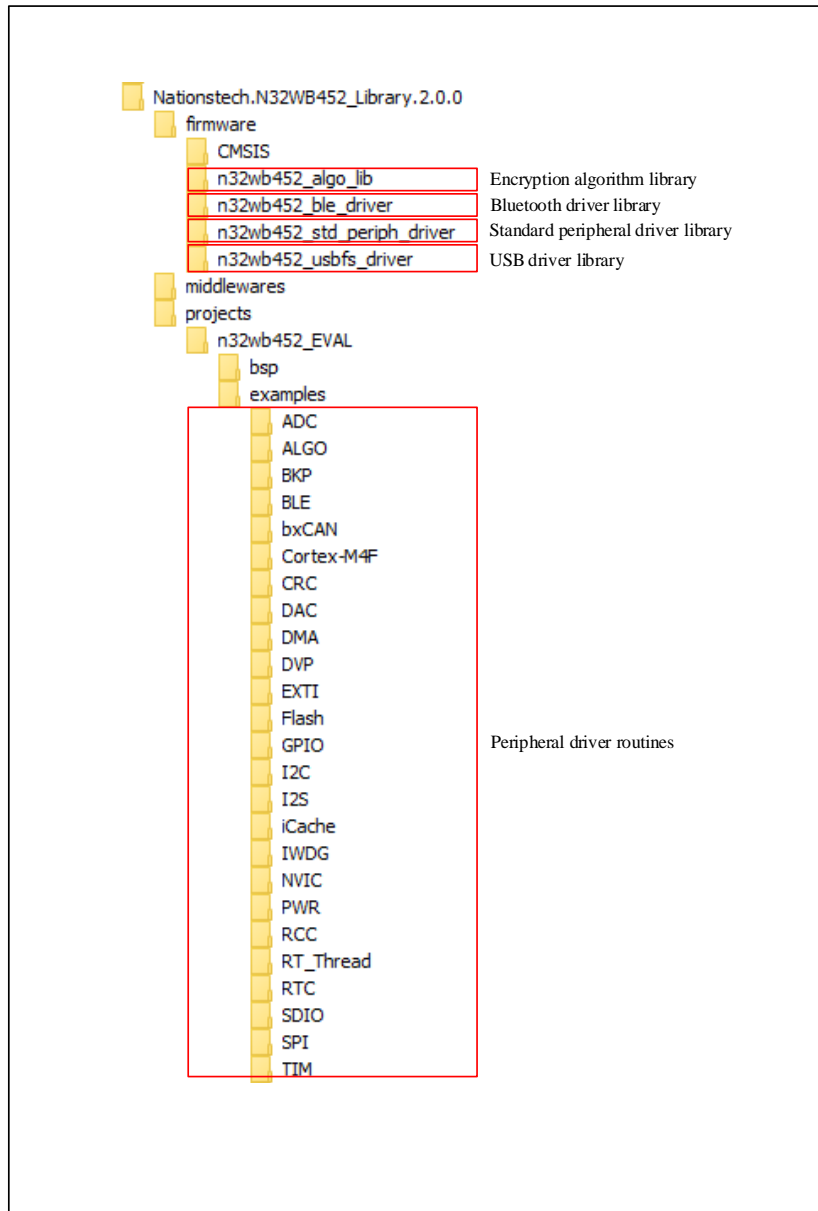
### 3 固件开发包介绍

通常 SDK 提供一个 Keil 的 PACK 包，以及 SDK 源代码压缩文件包。

 Nationstech.N32WB452\_DFP.1.0.3.pack

 Nationstech.N32WB452\_Library.2.0.0.7z

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:

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。

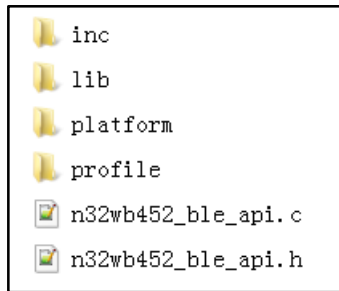
- 包含了用来访问内核的寄存器设备的名称定义，地址定义和配置函数。该接口包括调试通道定义。
- 提供片上所有外设的定义，包括所有外设寄存器头文件、启动文件、系统初始化模板文件。
- DSP库，提供优化的信号处理算法。

- **n32wb452\_algo\_lib:**

- 算法库文件，包括：加密算法、HASH 算法、随机数算法等。

- **n32wb452\_ble driver:**

- BLE 驱动文件，包括 ble 算法库用到的头文件 inc，蓝牙驱动库 lib，用到芯片相关资源文件 platform，和 ble fprofile。蓝牙驱动库 lib 中包含 IAR 与 KEIL 两种开发环境的库文件，包括蓝牙协议栈驱动库（host.lib、host\_16bit.lib、host\_16bit.a）和蓝牙底层驱动库（n32wb452\_ble.lib、n32wb452\_ble\_16bit.lib、n32wb452\_ble\_16bit.a）。
- 用户在使用 BLE 功能时，可以调用 n32wb452\_ble\_api.c 里的 api 函数，实现蓝牙功能的注册、回调、广播、连接等。



- **n32wb452\_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\_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

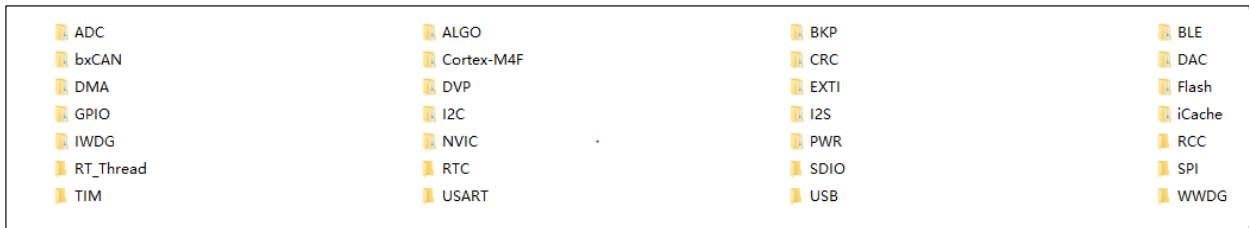
- **BSP:**

- 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

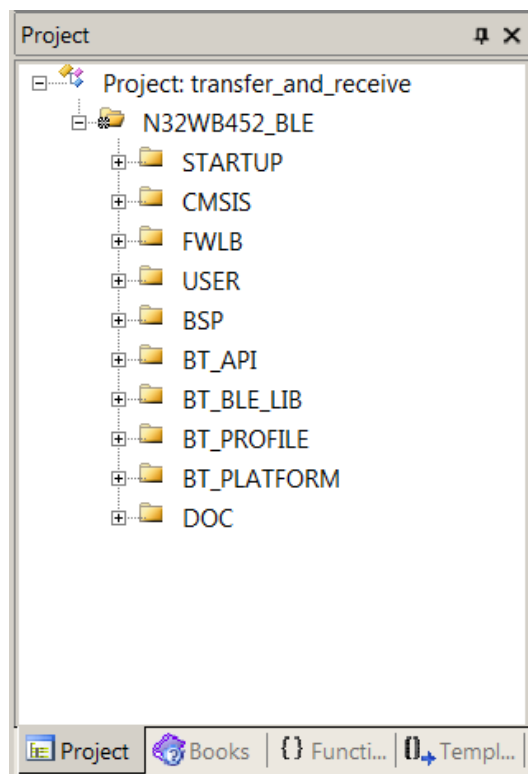
应用开发，用户可通过这些例程项目快速的了解芯片用法、编程技巧、模块移植。



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。

### 3.2.1 Projects 框架

以 BLE 的 Slave 工程为例：



- **STARTUP:**
  - 包含了 N32WB452 的启动文件。
- **CMSIS**
  - 包含了 system\_n32wb452.c 源文件。
- **FWLIB**
  - 包含了 n32wb452 的驱动文件。
- **USER**
  - 包含了开发板搭载用户应用层文件。
- **BSP**

- 包含了 n32wb452 的 BSP 文件。
- **BT\_API**
  - 包含了 BLE 的 API 接口文件。
- **BT\_BLE\_LIB**
  - 包含了 BLE 的蓝牙协议栈驱动库与底层驱动库文件。
- **BT\_PROFILE**
  - 包含了 BLE 的 Profile 文件。
- **BT\_PLATFORM**
  - 包含了 BLE 使用 n32wb452 芯片相关资源的源文件，
- **DOC**
  - 例程简要说明文档。


## 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 Nsing 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 产品许可密钥）。

### 4.2 固件支持包安装

请安装固件支持包：

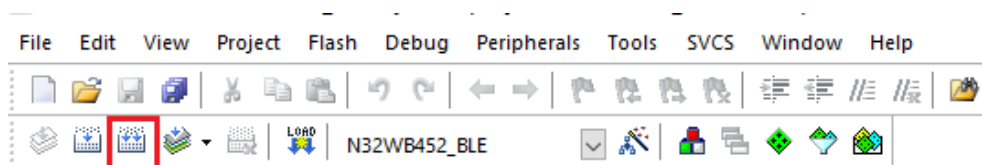
 Nationstech.N32WB452\_DFP.0.4.0.pack

安装成功后，在 KEIL 的 Pack Installer 界面中可以看到支持的设备列表：

N32WB452 Series	3 Devices
N32WB452	3 Devices
N32WB452CEQ6	<a href="#">ARM Cortex-M4, 144 MHz, 144 kB RAM, 512 kB ROM</a>
N32WB452LEQ6	<a href="#">ARM Cortex-M4, 144 MHz, 144 kB RAM, 512 kB ROM</a>
N32WB452REQ6	<a href="#">ARM Cortex-M4, 144 MHz, 144 kB RAM, 512 kB ROM</a>

### 4.3 软件编译

在 SDK 中选择一个工程并打开，点击全局编译。



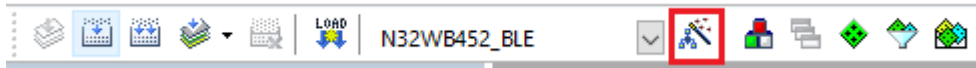
默认无修改的项目都可以编译通过：

```

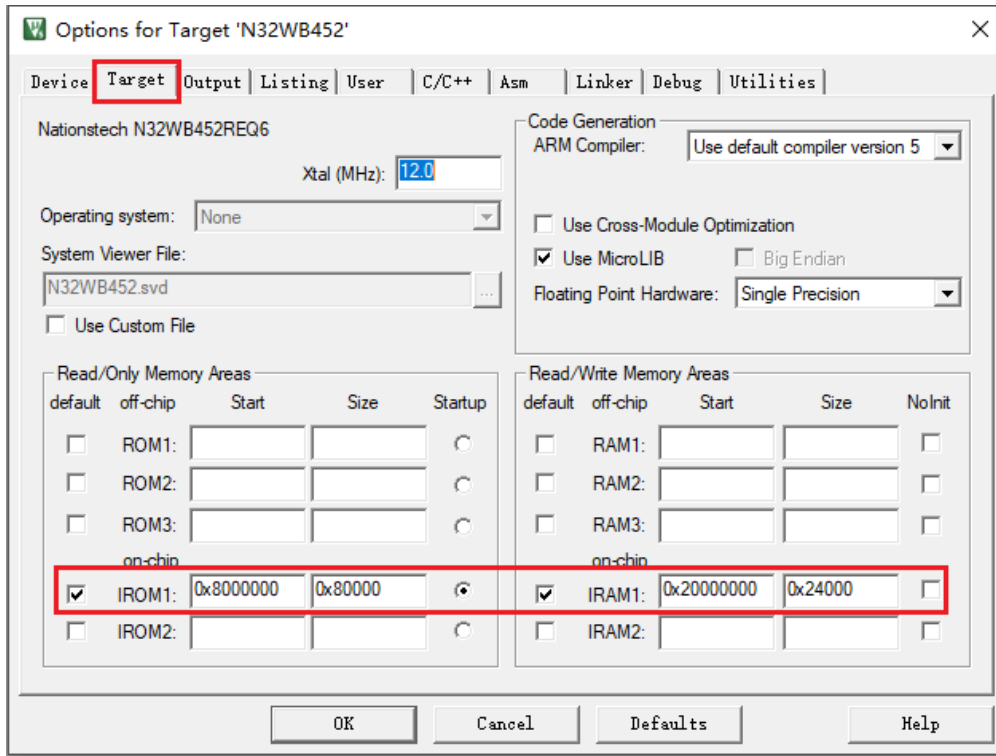
compiling lcd_gui.c...
compiling font24.c...
compiling lcd_drv.c...
compiling gsensor.c...
compiling i2c_drv.c...
compiling hdc_i2c_drv.c...
compiling HDC2010.c...
linking...
Program Size: Code=98192 RO-data=18500 RW-data=744 ZI-data=133352
FromELF: creating hex file...
".\Objects\ble_transfer.axf" -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

### 4.4 编译环境配置

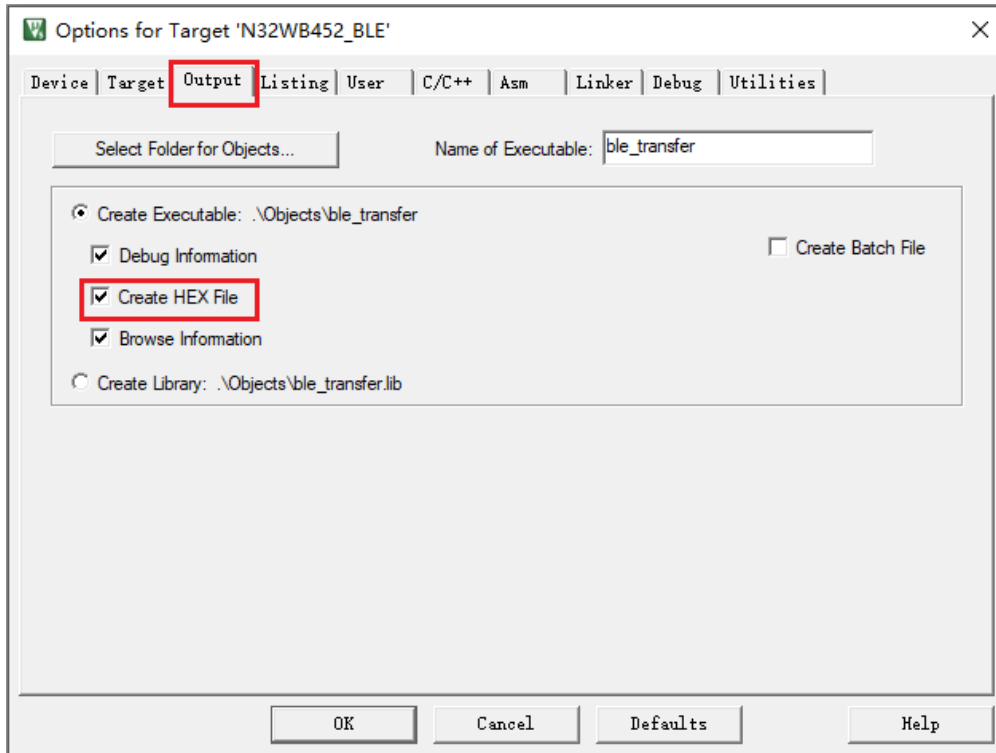
点击魔术棒，打开 OPTION 设定界面。



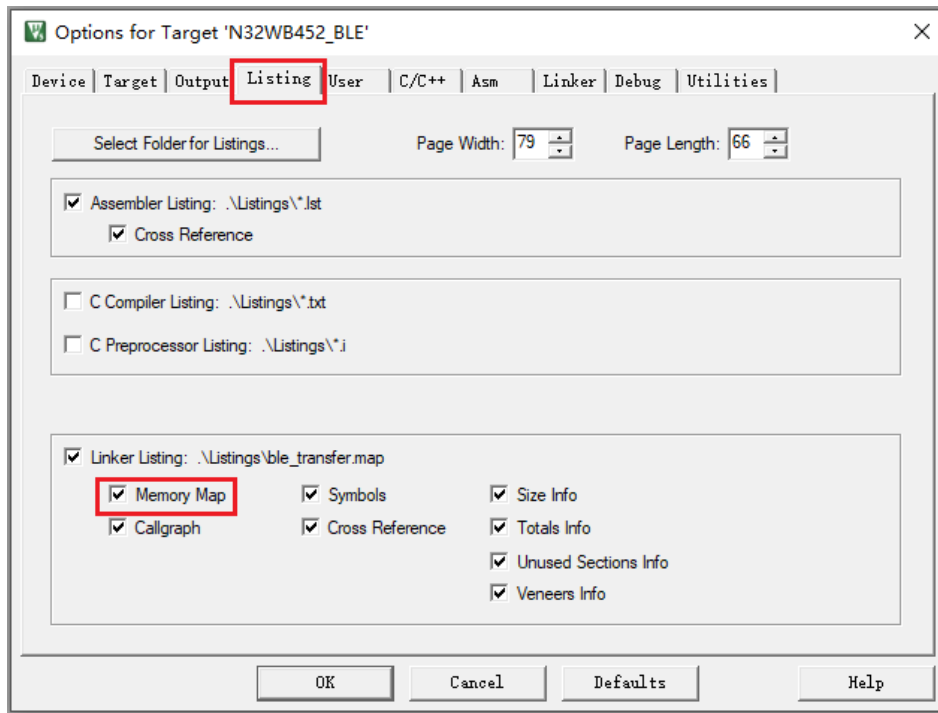
1) 在 Target 界面中设定 ROM 和 RAM 的大小（根据不同型号设定）



2) 在 Output 界面中勾选 hex 输出文件



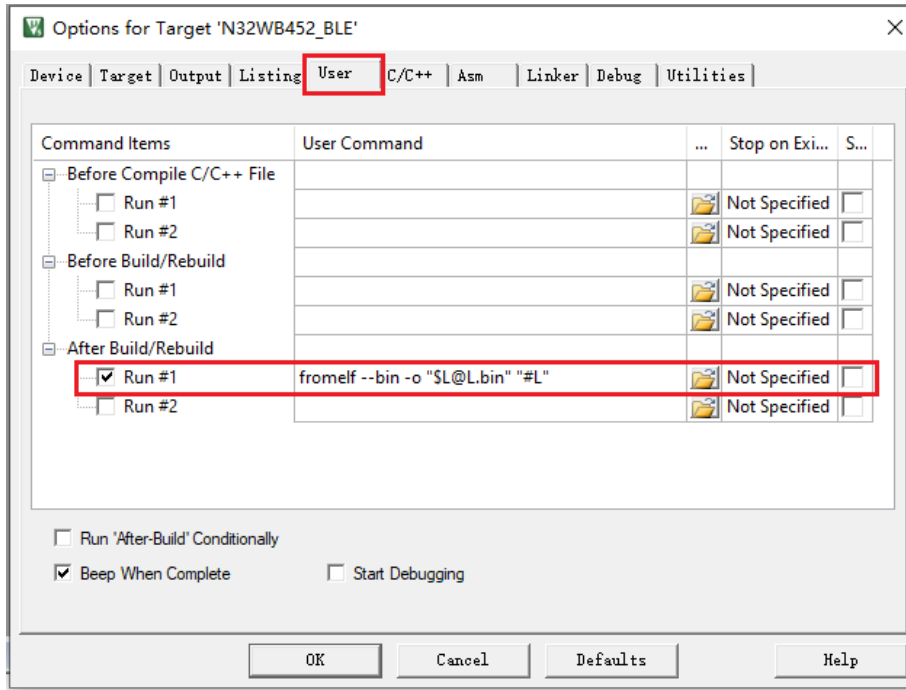
### 3) Memory Map 文件生成



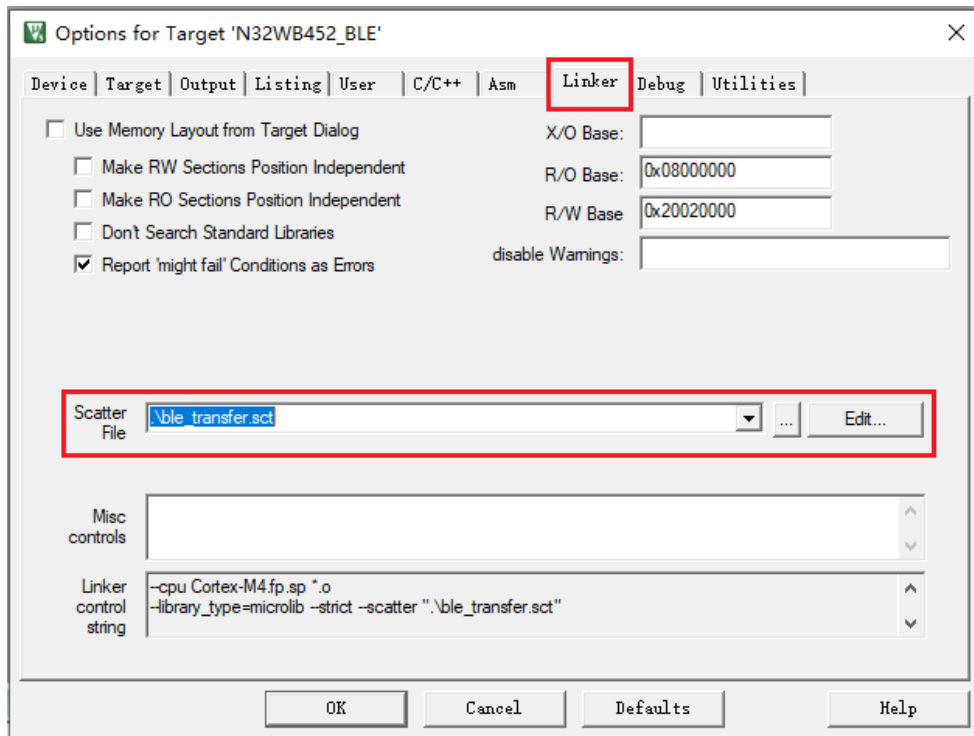
.map 文件记录了程序存储空间使用情况，可方便查看变量、常量、函数存放的地址空间和大小，如下图。

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)
<b>main</b>	<b>0x08003a93</b>	<b>Thumb Code</b>	<b>798</b>	<b>main.o(.text)</b>
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) 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



5) Linker Scatter File 设置（根据项目实际需求设置）

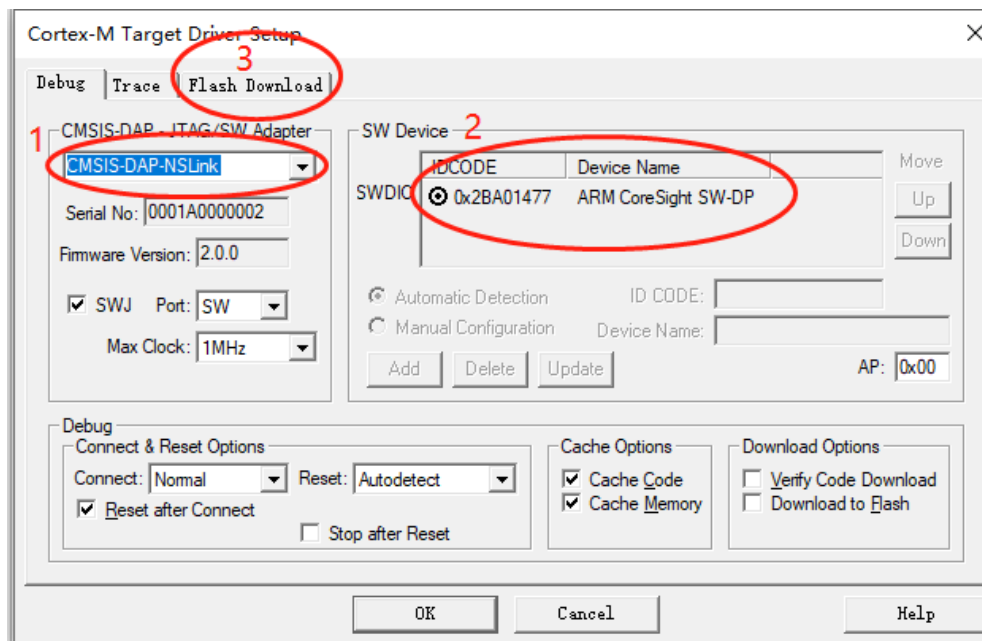
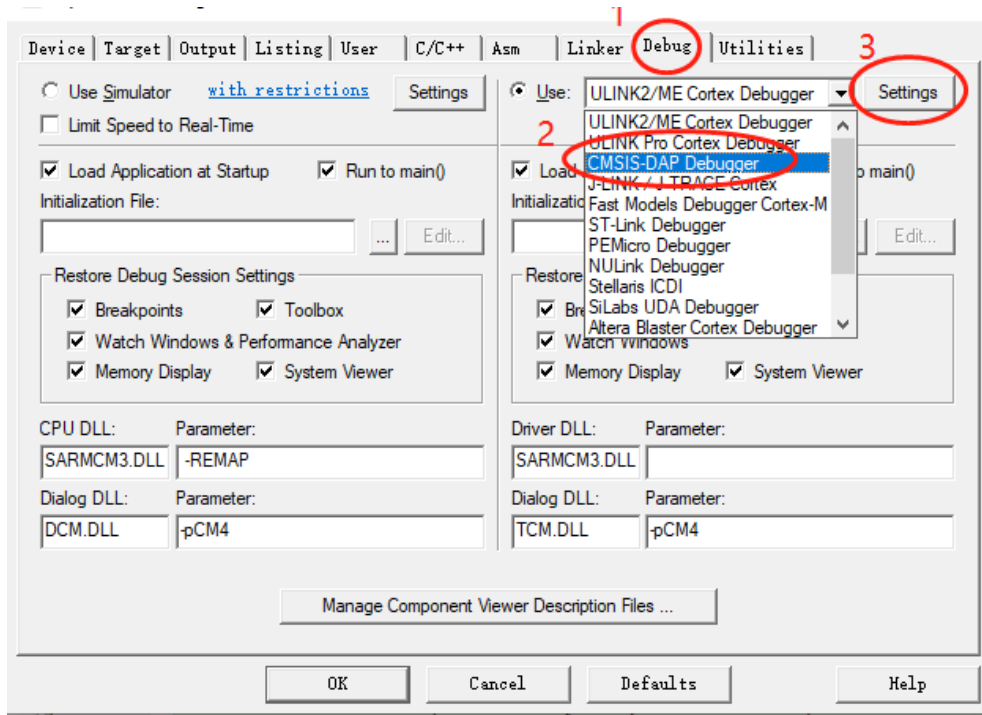


.sct 一个文本文件，通过编写一个分散加载文件来指定 ARM 连接器在生成映像文件时如何分配 RO,RW,ZI 等数据的存放地址。一般情况下 ARM 连接器会按照默认的方式来生成映像文件。

某些场合，我们希望把某些数据放在指定的地址位置

- 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)。

- 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。
  - 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。
  - 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。
- 6) 在 Debug 界面中进行仿真相关设置

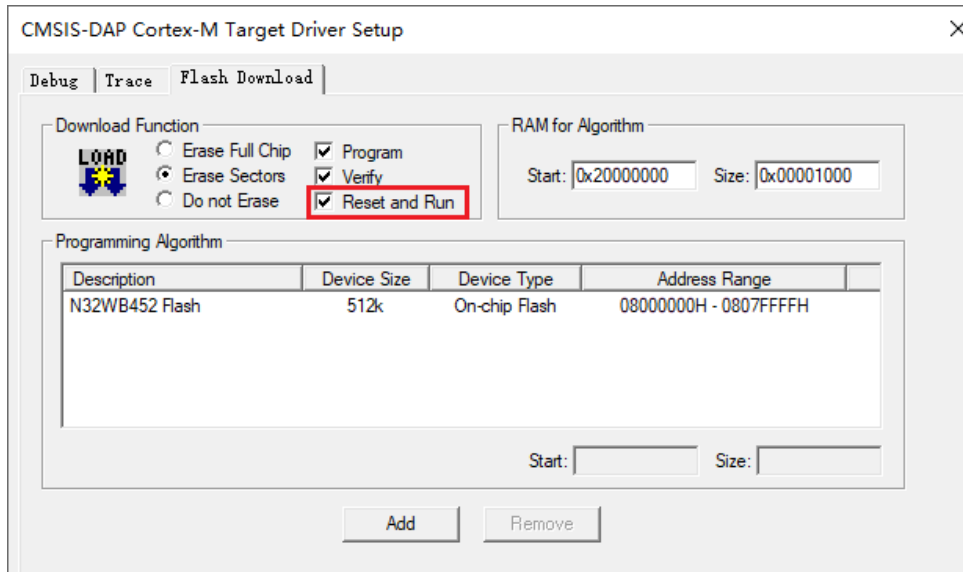


a) 选择 NSLINK 仿真器（如连接其他 LINK，则选择对应的 LINK 种类）。



- b) 如连接成功，在位置 2 出可以显示出仿真器的设备序列号。
- c) 在 Flash Download 界面中勾选复位运行。

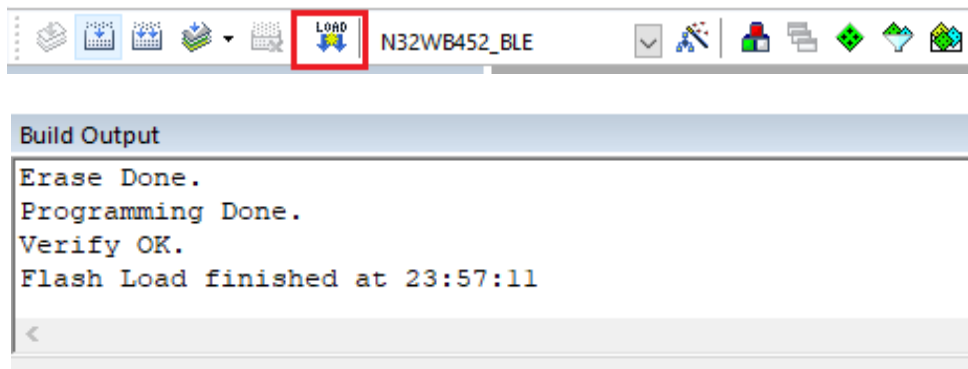
点击 Flash Download 设置,勾选“Reset and Run”，代码下载后自动复位和运行。



到这里编译环境设置完成。

## 4.5 下载和仿真

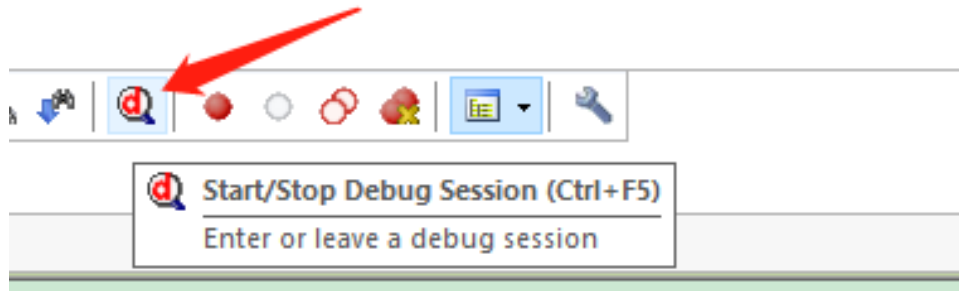
### 4.5.1 下载固件



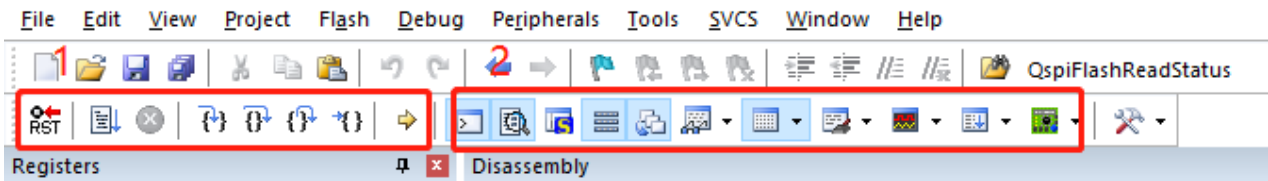
如上图所示，此时开发板/调试板已经在运行程序。

### 4.5.2 仿真调试

当仿真器连接且程序编译通过后，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**



- 1) 点击按钮 1，打开变量查看窗口

Watch 1			变量地址和值
Name	变量名称	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

- 2) 点击按钮 2，打开存储器查看窗口

Memory 1																				
Address: <input type="text" value="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	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	

3) 点击按钮 3 打开寄存器查看窗口

Property	Value
<input checked="" type="checkbox"/> GPIOx_PL_CFG	0xB BBB0000
<input checked="" type="checkbox"/> GPIOx_PH_CFG	0x88800000
<input checked="" type="checkbox"/> GPIOx_PID	0x0000C010
<input checked="" type="checkbox"/> GPIOx_POD	0x0000A000
<input checked="" type="checkbox"/> GPIOx_PBSC	0
<input checked="" type="checkbox"/> GPIOx_PBC	0
<input checked="" type="checkbox"/> GPIOx_PLOC...	0
<input checked="" type="checkbox"/> GPIOx_DS_CFG	0
<input checked="" type="checkbox"/> 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。Nsing 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板，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 Nsing 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 版本历史

日期	版本	备注
2020-06-15	V1.0	创建文档
2020-07-30	V1.1	升级 N32WB45xL-EVB 开发板到 V1.1 版本
2022-06-24	V1.2	1、修改了 USB、调试接口、GPIO、SPI FLASH 错误的接口及跳线说明 2、更新了与当前 SDK 不匹配的示例截图及相关说明 3、修改 Linker Scatter File 设置里的文本文件为.sct

## 7 声明

国民技术股份有限公司（下称“国民技术”）对此文档拥有专属产权。依据中华人民共和国的法律、条约以及世界其他法域相适用的管辖，此文档及其中描述的国民技术产品（下称“产品”）为公司所有。

国民技术在此并未授予专利权、著作权、商标权或其他任何知识产权许可。所提到或引用的第三方名称或品牌（如有）仅用作区别之目的。

国民技术保留随时变更、订正、增强、修改和改良此文档的权利，恕不另行通知。请使用者在下单购买前联系国民技术获取此文档的最新版本。

国民技术竭力提供准确可信的资讯，但即便如此，并不推定国民技术对此文档准确性和可靠性承担责任。

使用此文档信息以及生成产品时，使用者应当进行合理的设计、编程并测试其功能性和安全性，国民技术不对任何因使用此文档或本产品而产生的任何直接、间接、意外、特殊、惩罚性或衍生性损害结果承担责任。

国民技术对于产品在系统或设备中的应用效果没有任何故意或保证，如有任何应用在其发生操作不当或故障情况下，有可能致使人员伤亡、人身伤害或严重财产损失，则此类应用被视为“不安全使用”。

不安全使用包括但不限于：外科手术设备、原子能控制仪器、飞机或宇宙飞船仪器、所有类型的安全装置以及其他旨在支持或维持生命的应用。

所有不安全使用的风险应由使用者承担，同时使用者应使国民技术免于因为这类不安全使用而导致被诉、支付费用、发生损害或承担责任时的赔偿。

对于此文档和产品的任何明示、默示之保证，包括但不限于适销性、特定用途适用性和不侵权的保证，国民技术可在法律允许范围内进行免责。

未经明确许可，任何人不得以任何理由对此文档的全部或部分进行使用、复制、修改、抄录和传播。