

BlueX Microelectronics Co., Ltd.

Bluetooth 5.0 LE | MESH SoC



BX2400-dRF0xp-S1x

BX2400-eRF01e-G1x

版本:1.7

日期:2021/08/11



http://www.bluexmicro.com



目录

1.	开发	板说明	
	1.1	BX2400-dRF0xp-S1c 开发板	 3
	1.2	BX2400-eRF01e-G10 开发板	 5
	1.3	BX2400-eRF01e-G1a 开发板	 7
	1.4	10 资源说明	 9
2.	工具	下载	 10
3.	工具	安装	
	3.1	Keil 的安装	 10
	3.2	系统环境变量的添加	 12
	3.3	J-Link 的安装	 14
4.	J-Link	的使用	
	4.1	程序下载	 16
	4.2	擦除 Flash 中的程序	 24
	4.3	Debug 与查看 Log	 25
5.	文档	修改记录	 28
6.	附录		
	6.1	BX2400-dRF0xp-S1c 原理图	 29
	6.2	BX2400-eRF01e-G10 原理图	 30
	6.3	BX2400-eRF01e-G1a 原理图	 31



1. 开发板说明

本章着重介绍各款开发板的功能与用法, 文末附各款开发板原理图。

1.1 BX2400-dRF0xp-S1c 开发板

BX2400-dRF0xp-S1c 开发板适用于 RF01 以外的 BlueX BLE 芯片开发,项目(1)到(18)说明如下:



- (1) Type-C USB 接口
 - A. USB 5V 电源输入
 - B. 此处提供 USB 转 UART 电路, 连接到模块的 UART(P12/ P13) 接口
- (2) 开发板主电源选择开关

#2 与#9 拨动开关用于选择开发板电源,两开关挡位位置与开发板电源的关系如下表:

#2 拨动开关挡位	#9 拨动开关挡位	开发板电源	模块供电电压
上	上	2节5号电池	3.0V
F	上	2节5号电池	3.0V
上	下	DC 5V, #3 处输入	4.2V
下	下	DC 5V, USB 输入	4.2V

由于 RF02/BX2416 的工作电压范围为 2.3V~3.6V, 故这两种模块在开发板上只能使用两节 5 号电池供电。

(3) D5V 外部电源接口

A. 外部电源输入口, 直流 5V 输入

- (4) RFOx 模块电源接口
 - A. RFOx 模块电源外接接口, 用户可从此引 RFOx 模块电压给外部电路使用
 - B. 当开发板使用 USB 电源或 D5V 处(图中#3 处) 5V 电源供电时,此处输出 4.2V 电压
 - C. 当开发板使用 2 节 5 号电池供电时,此处输出电池电压
- (5) 拓展 10 接口

A. RFOx 模块的 IO 和电源外接的接线处



- (6) 模块转接板接口
 - A. BX241x/RF0x 模块焊接到转接板后,从此处插到开发板,如下图:



(7) RF04/RF08 模块焊接区域

A. RF04/RF08 模块, 可在此处直接焊接到开发板

- (8) 蜂鸣器
- (9) 二次电源开关 請參考#2 开发板主电源做比較
- (10) RGB LED
- (11) RFOx 模块复位按键

A. 按下该按键时, 模块会执行芯片复位

- (12) OLED 显示屏
- (13) LIS3DSH 加速度传感器接口
- (14) P16/P23/Watchdog 上下拉拨码开关

#14 拨码位置		-	功能
+	D16	上	芯片从 UARTO(P12(TX)/P13(RX))启动, 用于下载固件
丘	P10	下	芯片从 Flash 启动
中	P23	下	当使用的 BLE 模块芯片为 RF08 时,此位上电前需下拨
		Ŀ	使用板载看门狗芯片,看门狗定时器溢出周期约25s,看门狗
+	Watch dog	1	定时器溢出后模块复位
石	watchdog	Ŧ	不使用板载看门狗芯片,开发板看门狗 IC 默认不焊接,不焊时
		L L	不需要考虑此位

- (15) P10/P11/P21 电路连接选择接口
 - A. P10 可选择连接 RGB LED 的绿灯,或者 OLED 显示屏的 DC 脚
 - B. P11 可选择连接 RGB LED 的蓝灯, 或者 OLED 显示屏的 RST 脚
 - C. P21 可选择连接 RGB LED 的红灯, 或者湿度传感器的 QD 脚
- (16) DS18B20 湿度传感器接口
- (17) UART 接口
 - A. 此处连接到 RFOx 的 UART(P12/P13) 接口
- (18) SWD 接口

A. J-Link SWD 下载调试接口,从此处对 RFOx 模块进行烧录,或软件调试



1.2 BX2400-eRF01e-G10 开发板

BX2400-eRF01e-G10 开发板适用于 RF01 芯片开发,项目(1) 到(11) 说明如下:



- (1) RF01, QFN52, 6*6mm
- (2) 外接天线座
- (3) 板载 3.3V LDO
- (4) 跳线端子
 - A. 分断 VDD_BAT 和 LDO, 不接则 VDD_BAT 与板载 3.3V LDO 断开, 可以用于总的耗电电流 的测量
- (5) 电池供电端
 - A. 靠近跳线端的是 VDD_BAT, 靠近 LDO 的是 GND
 - B. 需要配合項目(4) 跳线来使用(短接跳线则电池给 LDO 供电)
- (6) 10 外接口,接口引脚图,如下:





(7) SWD 调试接口

A. P00-SWCLK, P01-SWDIO

(8) UART 串行接口

A. P12-TXD, P13-RXD

- (9) P16/P23 拨码开关
 - A. 左侧是 P16, 右侧是 P23
 - B. 功能和使用方法同 BX2400-dRF0xp-S1a 开发板 P16/P23 拨码开关,请参考 1.1 节中拨码 开关的用法
- (10) 按键和 LED
 - A. KEY-P15/P17, LED-P02/P03 无特殊用途, 用户可自定义功能
- (11) SPI Flash 芯片
 - A. CS#-P24, DO-P27, WP#-P28, HOLD#-P29, CLK-P25, DI-P26



1.3 BX2400-eRF01e-G1a 开发板

BX2400-eRF01e-G1a 开发板适用于 RF01 芯片开发,项目(1)到(10)说明如下:



- (1) RF01, QFN52, 6*6mm
- (2) 外接天线座
- (3) RGBW LED
 - A. 10 连接如下图:



- (4) 电源与 10 排针接口
 - A. 接口引脚图如下:



B. VDD_3V_0 为板载 3.3V LDO 输出端,为 RF01 VDD_BAT/1/2 PIN 供电



(5) 电源, 10 和 SWD 排针接口, 接口引脚图如下:



A. 第15PIN 与第16PIN 为 RF01 的 SWD 接口, 用 Debug 与下载固件

(6) SPI Flash 芯片

A. CS#-P24, DO-P27, WP#-P28, HOLD#-P29, CLK-P25, DI-P26

- (7) 板载 3.3V LDO
- (8) P16/P23 拨码开关
 - A. 左侧是 P16, 右侧是 P23
 - B. 功能和使用方法同 BX2400-dRF0xp-S1a 开发板 P16/P23 拨码开关,请参考 1.1 节中拨码 开关的用法
- (9) 按键
 - A. 10 连接如下图: VDD_3V_0 VDD_3V_0 R10 R11 10k 10k P15 P17 d SW2 SW1 d C28 C29 Q $0.1 \mathrm{uF}$ 0.1uF
- (10) Type-C USB 接口
 - A. USB 5V 电源输入
 - B. 此处提供 USB 转 UART 电路, 连接到模块的 UART(P12/ P13) 接口



BLUEX

- (1) RFOx 与 BX241x 系列 BLE 芯片的 IO 资源说明如下:
 - A. P00-P29 为数字 IO, 可作为 GPIO 使用
 - B. P30-P35 为模拟输入口,不可作为 GPIO 使用
 - C. POO 和 PO1 分别为 SWD 的 SWCLK 和 SWDIO
- (2) UART
 - A. 最多支持两路 UART, 可以在 PO2-P23 中任意管脚配置使用 UART
 - B. P12 和 P13 默认定义为 UARTO 的 TX 和 RX
- (3) SPI
 - A. 最多支持 2 路 SPI, 但使用时 SPIM 和 SPIS 的管脚是固定的, 不可任意配置
 - B. SPIMO 使用 PO2(CS1), PO3(CS0), PO4(CLK), PO5(MISO)& PO6(MOSI)
 - C. SPIM1 使用 P07(CS1), P08(CS0), P09(CLK), P10(MISO)& P11(MOSI)
 - D. SPIS 使用 PO3(CS), PO4(CLK), PO5(MISO)& PO6(MOSI)
- (4) I²C
 - A. 最多2路1²C, 可以在 P02-P23 中任意管脚配置使用 I²C
- (5) PWM

A. 最多支持 5 路 PWM, 可以在 PO2-P23 中任意管脚配置使用 PWM (6) ADC

- A. 最多支持 6 路 ADC 数据采集,均为独立的模拟 IO(P30-P35)
- (7) 外部中断
 - A. P15, P16, P17, P22 和 P23 是支持唤醒的外部中断引脚

2. 工具下载

请自行下载 Keil5 及 Jlink6.3 以上版本安装包

3. 工具安装

- 3.1 Keil 的安装
 - (1) 双击安装包进行安装

名称	*	修改日期	关型	大小
JLink_Windows_V646j		2020/2/4 13:40	应用程序	35,369 KB
		2020/2/4 14:13	好任 RAR 压缩文件	530 KB
d mdk523		2020/2/4 15:55	应用程序	737,407 KB

(2) 点击[Next]



(3) 勾选[I agree all the terms of the preceding License Agreement] 后点击[Next]

License Agreement Please read the following license agreement carefully.	ARM [®] KEIL Microcontroller Tools
To continue with SETUP, you must accept the terms of the Lice agreement, click the check box below.	inse Agreement. To accept the
END USER LICENCE AGREEMENT FOR MDK-ARM	*
THIS END LISER LICENCE ACREEMENT (LICENC	
ARM LIMITED ("ARM") FOR THE USE OF THE SOFT LICENCE. ARM IS ONLY WILLING TO LICENSE T CONDITION THAT YOU ACCEPT ALL OF THE TE CLICKING "I AGREE" OR BY INSTALLING OR OTHE	SINGLE LEGAL ENTITY) AND WARE ACCOMPANYING THIS HE SOFTWARE TO YOU ON ERMS IN THIS LICENCE. BY ERWISE USING OR COPYING +
BETWEEN YOU (EITHER A SINGLE INDIVIDUAL, OR ARM LIMITED ("ARM") FOR THE USE OF THE SOFT LICENCE. ARM IS ONLY WILLING TO LICENSE T CONDITION THAT YOU ACCEPT ALL OF THE TE CLICKING "I AGREE" OR BY INSTALLING OR OTHE	SINGLE LEGAL ENTITY) AND WARE ACCOMPANYING THIS HE SOFTWARE TO YOU ON ERMS IN THIS LICENCE. BY ERWISE USING OR COPYING +



(4) 选择安装路径后点击[Next]

Folder Selection Select the folder where SETUP will install files.	ARM [®] KEIL Microcontroller Tools	
Press 'Next' to install MDK-ARM to these folders. Press 'Brows	se' to select different folders for installation.	
Pestination Folders		
Lore: [C:\Kei_v5	BIowse	
Pack: C:\KeiLv5\ARM\PACK	B <u>r</u> owse	

(5) 如下图,框内内容填写后点击[Next]

Customer Information Please enter your information.	ARM KEIL Microcontroller Tools	
Please enter your name, the name of the con	npany for whom you work and your E-mail address.	
First Name:		
Last Name:		
Company Name:		
company reality		
E-mail:		

(6) 等待安装完成

Setup Status	ARM [®] KEIL [®] Microcontroller Tools
MDK-ARM Setup is performing the req	uested operations.
Install Files	
Installing c_we.b.	
-	
Keil MDK-ARM Setup	



(7) 点击[Finish] 完成安装



- 3.2 系统环境变量的添加
 - (1) 在[我的电脑] 上右键鼠标, 点[属性]



(2) 点击 [高级系统设置]

()・ 控制面板 ・	系统和安全 > 系统	▼ 49	度素控制面板
文件(E)编辑(E) 重看(Y) 工	具(1) 帮助(11)		
控制面板主页	Contract and		
😚 设备管理器	Contract of the local division of the local		
😚 远程设置	10000		0
◎ 系统保护	Acres 1.00	STATUS STREET,	
· 高级系统设置	Receiver .		
	系统		
	1000	CONTRACT OF A DESCRIPTION OF A DESCRIPTI	
	100		
	and the second second	the second second	
另语参问	and the second second	Company of the local division of the local d	
摄作中心	10.000	And Income the owner where the second s	
Windows Update	COLUMN TWO IS NOT		
性能信息和工具	++====	PC-201810251HVP	NECCH



(3) 点击[环境变量]



(4) 如图, 在选中[Path] 后点击[编辑]

29,000	值	
OneDrive	C:\Users\96266\OneDrive	
OneDriveConsumer	C:\Users\96266\OneDrive	
Path	C:\Users\96266\AppData\Local\Microsoft\WindowsApps;C:\Use	ers\
TEMP	C:\Users\96266\AppData\Local\Temp	
TMP	C:\Users\96266\AppData\Local\Temp	
	新建(N) 编辑(E) 册	馀(D)
系统变量(<u>5</u>)		
交量	値	1
OS	Windows_NT	
PADS_PROGRAMS	Programs	
PADS ROOT	C:\MentorGraphics\9.5PADS\SDD HOME	_
Path	C:\MentorGraphics\9.5PADS\SDD_HOME\common\win32\bin;0	SA
PATHERT	.COM, EXE, BAT, CMD, VDS, VDE, JS, JSE, WSF, WSH, MSC	_
PROCESSOR_ARCHITECTURE	AMD64	
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 142 Stepping 10, GenuineIntel	
	8	
PROCESSOR TEVEL		

(5) 如图新建, 并选择 keil 安装路径下的 bin 文件夹

	_
C:\MentorGraphics\9.5PADS\SDD_HOME\common\win32\bin	新建(N)
C:\MentorGraphics\9.5PADS\SDD_HOME\common\win32\lib	- Anne Content
%MGC_HOME%\bin	编辑(E)
%MGC_HOME%\lib	
%SystemRoot%\system32	浏览(B)
%SystemRoot%	-
%SystemRoot%System32Whem	删除(D)
D:\keil_v5\ARM\ARMCC\bin	
c.(riogram riles)(or(cind	
	上移(山)
	下移(Q)
	-
	100100-0-1-000
	编辑又本(1).
15h cta	HUSSK

(6) 点击[确定], 完成环境变量配置



- 3.3 J-Link 的安装
 - (1) 双击安装包开始安装

名称	修改日期	类型	大小
JLink_Windows_V646j	2020/2/4 13:40	应用程序	35,369 KB
	2020/2/4 14:13	好性 KAR 法殖文件	530 KB
4 mdk523	2020/2/4 15:55	应用程序	737,407 KB

(2) 点击[Next]



(3) 点击[I Agree]

C	License Agreement Please review the license terms before installing SEGGER - J-Link V6.46j.	J Link
	Press Page Down to see the rest of the agreement.	
	Important - Read carefully:	*
Embedded Studio	DEFINITIONS: For the purpose of this agreement, the terms shall have the following meaning when the entire word is marked bold:	
Powerful C/C++ IDE available for	The "software" means all J-Link related software components included in the J-Link software & documentation pack provided by SEGGER which can be downloaded at: http://www.segger.com/jiink-software.html	
Windows	"Licensor" shall mean SEGGER except under the following circumstances:	-
 macOS Linux 	If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install SEGGER - J-Link V6.46J.	
Download Trial		

(4) 点击[Install]: (注意: 请勿修改安装路径, J-Link 请安装在系统默认的路径下)

SEGGER	Choose optional components Choose optional components to be installed.
loT 🤝	Install USB Driver for J-Link Create entry in start menu Add shortcuts to desktop
A DEN	Choose destination: Update existing installation Install a new instance Destination Folder
Discover SEGGER solutions for the Internet of Things	C: (Program Files (x86) \SEGGER \JLink Browse
Learn More	< Back Instal Cancel



(5) 等待安装

	Installing Please wait while SEGGER - J-Link V6.46j is being installed.	ل Link
Jamac	Extract: JLink_x64.dll 87%	
	Extract: JLinkRTTLogger.exe	*
	Extract: JLinkRTTViewer.exe Extract: JLinkRegistration.exe	
J-Trace PRO	Extract: JLinkRemoteServer.exe	
Trace and Debug in Real Time!	Extract: JLinkSTM32.exe Extract: JLinkSTR91x.exe	
 Streaming Trace Live Profiling 	Extract: JLinkSWOViewer.exe Extract: JLinkSWOViewerCL.exe Extract: JLink_x64.dll 87%	E

(6) 点击[finsh] 完成安装

SEGGER - J-Link V6.46	Setup	
Litraco	11	Completing SEGGER - J-Link V6.46j Setup
	a a flink T	The latest version of the J-Link Software & Documentation pack can be found at the location linked below.
J-Trace PRO	-	
Trace and Debug in Real Time!		
 Streaming Trace Live Profiling Code Coverage 	SEGGER	J-Link Software & Documentation pack
Learn More		< Back Finish Cancel



4. J-Link 的使用

J-Link 固件需 6.0 以上版本, 硬件需 9.0 以上版本

4.1 程序下载

以 SDK 3.x 为例, SDK 3.x 下载链接: https://www.bluexmicro.com/download/cid-6.html

- (1) 方法 1: 使用 J-Flash 下载
 - A. 首先找到 SDK3.x 文件路径[SDK3\tools\bluex\prog_tool_v2] 下的[Bluex] 文件夹和
 [JLinkDevices.xml] 文件

	> SDK3 > tools > bl	uex > prog_tool_v2	ٽ ~
称 ^	修改日期	类型	大小
BlueX	2021/3/8 14:08	文件夹	
] JLinkDevices.xml	2021/1/4 16:35	XML 文档	1 KB
ReadMe.txt	2021/1/4 16:35	文本文档	1 KB

B. 将 Bluex 文件夹和[JLinkDevices.xml]文件复制 J-Link 安装路径下, 如图所示:

	SEGGER > JLink_V6	544I		
名称 へ	修改日期	类型	大小	
BlueX	2021/5/26 17:04	文件夹		
Devices	2020/4/25 15:23	文件夹		
Doc	2020/4/25 15:23	文件夹		
ETC	2020/4/25 15:23	文件夹		
GDBServer	2020/4/25 15:23	文件夹		
RDDI	2020/4/25 15:23	文件夹		
🔥 Samples	2020/4/25 15:23	文件夹		
USBDriver	2020/4/25 15:23	文件夹		
JFlash.exe	2019/4/12 23:18	应用程序	704 KB	
🔜 JFlashLite.exe	2019/4/12 23:18	应用程序	345 KB	
🛃 JFlashSPI.exe	2019/4/12 23:18	应用程序	408 KB	
🛃 JFlashSPI_CL.exe	2019/4/12 23:18	应用程序	563 KB	
🔜 JLink.exe	2019/4/12 23:18	应用程序	292 KB	
JLink_x64.dll	2019/4/12 23:19	应用程序扩展	17,268 KB	
JLinkARM.dll	2019/4/12 23:18	应用程序扩展	16,184 KB	
🔜 JLinkConfig.exe	2019/4/12 23:18	应用程序	441 KB	
JLinkDevices.xml	2021/4/15 11:09	XML 文档	1 KB	



BLUEX



(B) 再打开 J-Link 安装路径下的[JLinkDevices.xml]文件,在该文件的最后"</DataBase>" 前粘贴刚才复制的内容即可,如下图:







D. 在弹出的[create a new project] 对话框点击下图箭头指示位置完成以下操作

reate New Project		×
Target Device		
Cortex-M0		
Little endian 💌		1
Target Interface	Speed (kHz)	_
SWD 🗸	4000	•
	OK	
	<u> </u>	(

E. 在弹出窗口,点击下拉箭头,选择[BlueX],如下图:

1anufacturer 🛛 🗧	•			
Manufacturer Abov	^	Core	Flash size	RAM size
Inspecified active-s	emi	ABM7		
Inspecified Altera		ABM9		
Inspecified Ambigh	licro	ABM11		
Inspecified AMS		Contex-A5		
Inspecified Analog		Cortex-A7		
Inspecified AuDook		Cortex-A8		
Inspecified BlueX	Nay .	Cortex-A9		
Inspecified Cirrus I	onic	Cortex-A12		
Inspecified Cypress	ogic l	Cortex-A15		
Inspecified Dialog	Semiconductor	Cortex-A17		
Inspecified Digi		Cortex-A53		
Inspecified DSPGn	pup	Cortex-657		
Inspecified Epson		Cortex-M0		
Inspecified Farada		Cortex-M0		
Inspecified GigaDe	vice	Cortex-M1		
Inspecified Hilsche	r i i i i i i i i i i i i i i i i i i i	Cortex-M3		
Inspecified Int	~	Cortex-M4	1	
Inspecified	Lortex-M7	Cortex-M7		
Inspecified	Cortex-M23	Cortex-M23		
Inspecified	Cortex-M33	Cortex-M33		
Inspecified	Cortex-R4	Cortex-R4	2	
Inspecified	Cortex-R5	Cortex-R5		
Unspecified	Cortex-R8	Cortex-R8		
Inspecified	RX	BX		
Inspecified	RISC-V	RISC-V		
Ahov	AC33M6128	Cortey-M3	128 KB	12 KB

F. 根据所使用芯片 Flash 工作电压选择相应的配置,
 RF01/RF03/RF04/BX2416/BX2417/BX2418 选择 [Apollo_00_3v3],
 RF08 选择[Apollo_00_1v8], 然后点击[OK] 确定

Manufacturer	Device	Core	Flash size	RAM size
BlueX	Apollo_00_1V8	Cortex-M0	8192 KB	192 KB
llueX	Apollo_00_3V3	Cortex-M0	8192 KB	192 KB

G. 再回到[create a new project] 对话框,选择速率[4000kHz], 点击[OK]

Create New Project		×
Target Device		
BlueX Apollo_00_1V8		
Little endian 💌		
Target Interface	Speed (kHz)	
SWD 💌	4000	•
		<u>0</u> K

H. 完成工程的配置后, 再点击[File] 选择[Open data file], 选择需要下载的 hex 固件



1. Hex 固件选择之后 J-Flash,显示如下:

	indow Help																					^
Project - new p																S	DK3\	exam	ple			x
Name Value	Address	0.000	000		-	t																
Host connection USB (Device 0)	Agaress.	0,000	000			1.47	<u>~</u>	~=														
(, , , , , , , , , , , , , , , , , , ,	Address	0	1	2	3	4	5	6	7	8	9	A	В	С	D	E	F	ASC	II			-
Target interface SWD	800000	42	58	32	34	ØØ	80	12	ØØ	10	14	ØØ	ØØ	19	82	12	QQ	BX2	4			
Init SWD speed 4000 kHz	000000		00		00	00	00	00	00	20	20	20		25		~		2.1.0		••••	. \e	
SWD speed 4000 kHz	000010	66	99		99	66	99	97	99	00	00	30	69	35	11	61	27			• • • •	,	,
	800020	BØ	B2	69	46	42	78	00	AF	92	06	92	ØF	D3	10	DB	08	i	FBx.	• • • •		-
MCU BlueX Apollo_00_1V8	800030	DB	00	СВ	18	9D	46	05	00	69	46	00	78	01	FØ	8E	F9		F	iF	.×	-
Core Cortex-M0	800040	07	21	01	20	6B	78	6C	46	1 B	07	1 B	ØF	DA	10	ØB	40		kx.	1F		9
Endian Little	800050	99	40	02	50	BD	46	10	40	43	1 E	99	41	CO	82	RØ	BD		N F	BC.	۵.	
Lheck core IU No	000050	70	10				10	10	10	-13			-11	200	102	200	00					-
Use target HAM 192 KB @ 0x100000	800020	F8	B2	ьн	46	44	-78	00	HF	H4	øь	H4	øг	E3	10	DR	68	· · J	FDx.	• • • •		-
Flash memory Jubrard have 0	800070	DB	00	D3	1A	9D	46	05	00	22	00	00	21	68	46	D9	F6		F	".	.thF.	-
Page address 0v900000	800080	4E	FD	07	22	6B	78	6E	46	1B	07	1B	ØF	D9	10	13	40	Ν	"kxi	nF		9
Flach size 8192 KB	800090	06	38	9A	40	28	78	72	54	69	46	22	ØØ	01	FØ	69	F9	. : .	e(xi	rTiF	"i	
013210	900000	PD	40	E0	DD	10	DE	86	40	80	40	90	42	90	n 2	10	PD			τ.ν	P	-
	800000	60	-10	ro	60	10	63	80	40	80	70	76	74		5	10	50	· · ·			- B	-
	8 NON BO	62	68	20	68	90	21	D9	F6	32	FD	68	34	F4	E2	сø	46	bh	h.†	2.	.4	F
	800000	04	94	12	00	ØC	94	12	00	70	B5	05	00	07	48	ØE	00			p.	н.	-
	800000	14	00	00	FØ	F3	F8	01	20	32	00	29	00	40	42	00	FØ			. 2.	>.@B.	
	800050	11	F 9	00	1 B	43	42	58	41	CO	B 2	70	BD	87	1 D	C1	84		CB	20	n	
	000000			50	10	15	10	30		100	00		202	20			01		. 00		P	-
	800010	80	21	FØ	B2	31	48	40	40	18	bУ	85	RO	22	00	18	90	- 7 -	. r кı	86.1	"	
	800100	49	00	00	93	01	FØ	84	F8	3C	4B	3D	4D	22	68	2B	60	Ι		<k< th=""><th>=M''h+</th><th>•</th></k<>	=M''h+	•
	800110	3C	4B	9A	42	4C	D1	3C	4B	3C	48	2B	60	00	9B	62	68	<к.	BL.	(K <h< td=""><td>+`b</td><td>h i</td></h<>	+`b	h i
· · · · · · · · · · · · · · · · · · ·	000100	64	٢0	10	10	DD	69	ъø	DD	88	20	CC	ΠØ	00	9 9	20	AD	L		1	A 100	v 🔳
																						83
- JLinkARM.dll V6.4- Opening project fil. - Project opened successrv Failed to open data file	12																				1	Î
Close project - Project closed Creating new project																						
- New project creation of the second	100																					
Data file opened succes.																						×
<																						٠
leady																						



J. 此时,通过 J-Link SWD 接口连接电路板, 点击[Target] 选择[connect]

Droject - per	Connect			1	*11*	2/生相	l¥\s	סאס	RXM	cro-	SDK	3-rel		- 1/3 2	-202	1010			example	x
Name	Disconnect				e4 (3					0.0-	SDR.	5-161	case	- 43.2	-202	1010		51(3)	example	-
Host connection	Test			- 10			×1	<u>x2</u>	×4											
The connector	lest			1	2	3	4	5	6	7	8	9	A	в	С	D	E	F	ASCII	T
Target interface	Production Pro	gramming	F7	38	32	34	00	80	12	00	10	14	00	00	19	82	12	00	BX24	-
Init SWD speed	Manual Progra	mmina		90	04	00	00	00	07	00	08	00	3B	60	35	11	01	29		
SWD speed	. Inianual Progra	PPOPAP		-25	69	46	42	78	00	AF	92	06	92	ØF	D3	1 D	DB	08	iFBx	
мси	BlueXApollo 00 1V8	800030	DB	RR	CB	18	9D	46	Ø 5	88	69	46	RЯ	78	P 1	FØ	8E	F9		
Core	Cortex-M0	800040	07	21	R1	20	6B	78	60	46	18	07	18	ØF	DA	10	ØB	40	t kylF 0	
Endian	Little	000000	00	40	01	50	nn	40	10	40	42	107	00	44	00	10	na	10	0 \ E 0C 0	
Check core ID	No	800050	98	40	HZ	50	RD	46	10	40	43	1E	38	41	CØ	BZ	80	RD	.e.\.F.ecH	
Use target RAM	192 KB @ 0x100000	800060	F8	B5	6A	46	44	78	00	AF	A4	06	A4	ØF	E3	1D	DB	08	jFD×	
		800070	DB	00	D3	1A	9D	46	05	00	22	00	00	21	68	46	D9	F6	FhF	
Flash memory	Internal bank U	800080	4E	FD	07	22	6B	78	6 E	46	1B	07	1B	ØF	D9	10	13	40	N"kxnF0	
Base address	0102 KD	800090	86	30	90	40	28	78	72	54	69	46	22	00	01	FØ	69	29	· R(vaTiF" i	
riasri size	0132 ND	000070	nn	46	70	DD	10	DE	~	40	~	40	00	40	00	10				
		800000	RD	46	F8	BD	10	82	96	40	96	48	AC	42	99	D3	10	RD	.FL.K.B	
		8000B0	62	68	20	68	ии	21	D9	F6	32	FD	68	34	F4	E7	сø	46	bh h.¶24F	
		800000	04	94	12	00	ØC	94	12	00	70	B5	05	00	07	48	ØE	00	н	
		8000D0	14	00	00	FØ	F3	F8	01	20	32	00	29	00	40	42	00	FØ	2.).@B	
		8000E0	11	F9	RR	1 B	43	42	58	41	ся	B2	70	BD	B7	1 D	C1	84	CBXA	
		900020	00	24	PO	DE	20	40	40	40	1 10	60	OE.	DO	22	00	10	00	• 2V01 - "	
		000010	00	41	гØ	D2	31	7.0	-10	-10	1.0	67	05	DU	66	00	10	60		
		800100	49	90	90	A 3	61	гØ	84	F8	3C	4B	3D	4D	22	ь8	ZB	ь0	1 <k=m°h+< td=""><td></td></k=m°h+<>	
		800110	3C	4B	9A	42	4C	D1	3C	4B	3C	48	2B	60	00	9B	62	68	<k.bl.<k<h+`bh< td=""><td></td></k.bl.<k<h+`bh<>	
		000100	0.4	CO	40	4.0	7777	777	na	1111	00	20	E 12	na	00	20	20	40	1. /^ UOV	- 12

K. 连接成功后显示, 如下图:

SEGGER J-Fla	ash V6.44f - [new project	*]																	-		×
Ele Edit View Register von State Name Host connection Target interface Init SWD speed MCU Core Endien MCU Use target RAM Use target RAM Base addess Flash size	SN V0.447 - (new project I arget Options Wi Value US8 (Device 0) SW0 4000 kHz 4000 kHz 4000 kHz 4000 kHz 000 kHz 100 kHz 4000 kHz 100 kHz 100 kHz 100 kHz 100 kHz 100 kHz 110 kHz 110 kHz 110 kHz 110 kHz 111 kHz 112 kHz 0x00000 112 kHz 0x00000 112 kHz 110 kHz 0x00000 111 kHz 112 kHz 111 kHz 111 kHz 111 kHz 112 kHz 111 kHz </th <th>1 ndow Help 2. ExBULEX Address 800000 800010 800000 800000000</th> <th>0 80 9 9 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8</th> <th>档资 0000 1 58 00 21 40 85 00 21 40 85 00 70 30 46 68 94 00 FD 30 79 21</th> <th> □ □</th> <th>相关\ ×1 3 4 4 00 6 42 6 42 6 44 6 9D 6 44 6 9D 1 6 8 00 0 6 7 3 8 00 0 6 7 1 8 00 0 6 7 1 8 00 0 6 8 00 0 0 0 7 3 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0</th> <th>SDK/ x2 5 80 00 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 78 78 46 78 78 78 78 78 46 78 78 78 78 78 78 78 78 78 78</th> <th>8XM x4 6 12 07 00 05 6C 10 05 6C 10 05 6E 72 06 12 05 12 05 10 05 10 05 10 05 12 05 10 05 10 05 12 05 10 10 10 10 10 10 10 10 10 10</th> <th>7 00 00 AF 00 46 40 46 40 46 46 46 46 46 46 46 46 46 40 20 41 40</th> <th>8 1C 08 92 69 1B 43 44 22 1B 69 06 32 70 32 C0 1B</th> <th>9 14 00 46 07 1E 06 00 07 46 4B FD B5 00 B2 69</th> <th>A 00 3B 92 00 1B 98 A4 00 1B 22 9C 08 05 29 70 85 29</th> <th>B 00 60 0F 78 0F 41 0F 21 0F 21 0F 42 34 00 0D BD BD</th> <th>-202 19 35 01 DA CØ E3 68 09 01 00 F4 00 F4 07 40 B7 22</th> <th>1010 B2 11 10 FØ 10 B2 10 46 10 FØ D3 E7 48 42 10 00 00</th> <th>E 12 01 DB 8E 0B B0 D9 13 69 10 C0 00 C1 18 8 20 C1</th> <th>F 00 29 08 F9 40 8D 08 F6 40 8D 46 00 64 00 64 00 64 00 64 00 64 00 64 00 00 00 00 00 00 00 00 00 0</th> <th>ASCII BX24. iFB iFB N'k c. c. c. c. c. c.</th> <th></th> <th>; `5) ; `5) ?.×</th> <th></th>	1 ndow Help 2. ExBULEX Address 800000 800010 800000 800000000	0 80 9 9 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8	档资 0000 1 58 00 21 40 85 00 21 40 85 00 70 30 46 68 94 00 FD 30 79 21	 □ □	相关\ ×1 3 4 4 00 6 42 6 42 6 44 6 9D 6 44 6 9D 1 6 8 00 0 6 7 3 8 00 0 6 7 1 8 00 0 6 7 1 8 00 0 6 8 00 0 0 0 7 3 7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0	SDK/ x2 5 80 00 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 46 78 78 78 78 46 78 78 78 78 78 46 78 78 78 78 78 78 78 78 78 78	8XM x4 6 12 07 00 05 6C 10 05 6C 10 05 6E 72 06 12 05 12 05 10 05 10 05 10 05 12 05 10 05 10 05 12 05 10 10 10 10 10 10 10 10 10 10	7 00 00 AF 00 46 40 46 40 46 46 46 46 46 46 46 46 46 40 20 41 40	8 1C 08 92 69 1B 43 44 22 1B 69 06 32 70 32 C0 1B	9 14 00 46 07 1E 06 00 07 46 4B FD B5 00 B2 69	A 00 3B 92 00 1B 98 A4 00 1B 22 9C 08 05 29 70 85 29	B 00 60 0F 78 0F 41 0F 21 0F 21 0F 42 34 00 0D BD BD	-202 19 35 01 DA CØ E3 68 09 01 00 F4 00 F4 07 40 B7 22	1010 B2 11 10 FØ 10 B2 10 46 10 FØ D3 E7 48 42 10 00 00	E 12 01 DB 8E 0B B0 D9 13 69 10 C0 00 C1 18 8 20 C1	F 00 29 08 F9 40 8D 08 F6 40 8D 46 00 64 00 64 00 64 00 64 00 64 00 64 00 00 00 00 00 00 00 00 00 0	ASCII BX24. iFB iFB N'k c. c. c. c. c. c.		; `5) ; `5) ?.×	
		800100	49 3C	00 4В	9A 4	3 Ø1 2 40	FØ D1	84 3C	F8 4B	3C 3C	4B 48	3D 2B	4D 60	22 00	68 9B	2B 62	6Ø	1 <k.bl< td=""><td><}</td><td>(=M"h+ +`b}</td><td></td></k.bl<>	<}	(=M"h+ +`b}	
		000100	01	٤0	10 1	0 77	79	ħØ	DD	66	20	EE	ъø	00	22	20	AD	L.		·^	, •
R LOG																					83
 FFUnit: 4 coor CoreSight coor ROMTH1(0) @ F ROMTH1(0) [0]: ROMTH1(0)[1]: ROMTH1(0)[2]: Executing initialized interior Target interior Target interior Connected succession 	de (BF) slots and 0 lite proments: 500FF000 E000F0000, CTD: B105E00 E00001000, CTD: B105E00 E00002000, CTD: B105E00 it sequence. J S0002000, CTD: B105E00 it sequence. J Successfully Ene speed: 40000 kHz (Fi J JTMC Aurite. Core ID: coessfully	ral slots D, PID: 000BBG D, PID: 000BBG D, PID: 000BBG xed) 0x0BC11477 (1	008 S 00A D 00B F None)	CS WT PB																	<
<																					>
leady											С	onne	ected	1	Cor	e Id:	0x0	BC11477	Spe	ed: 400	0 kH

L. 此时, 点击[Target] 再点击[Produciton Programming] 即可完成固件的下载

SEGGER J-Fla	sh V6.44f - [new projec	:t *]														_		×
File Edit View	Target Options V	/indow Help																
Project - nev	Connect		「御料へ	软件框]¥\SD	K\BXM	licro-	SDK3	-relea	se-v3	.2-20	21010	04\SI	DK3\	example	. 🕞		×
Name	Disconnect			_														_
Hest compaction			- 00		X	2 ×4												
Host connection	lest	>	1 2	3	4	5 6	2	8	9 6	A B	C	D	E	F	ASCII			
Target interface	Production Prog	ramming F7	8 32	34	00 8	80 12	00	10	14 0	0 0	3 19	82	12	00	BX24			-13
SWD speed	Manual Program	nmina >	00 04	00	00 0	90 07	00	08	00 3	B 60	3 35	11	01	29		;	`5>	
SWD speed		1 000060 100	35 69	46	42 3	78 00	AF	92	06 9	2 01	F D3	1D	DB	08	iFB×			
MCU	BlueXApollo_00_1V8	800030 DB	00 CI	18	9D 4	16 Ø5	00	69	46 Ø	0 7	8 Ø1	FØ	8E	F9	F	iF.	×	
Core	Cortex-M0	800040 07	21 81	20	68 5	78 60	46	18	07 1	R A	FDO	10	ØR	40	• kv	18		
Endian	Little	000010 00	40 00	E0	DD /	10 00	40	42	1 . 0	0 4		n 0	DO	DD	PA		~	
Check core ID	No	888858 78	40 H2	50	80 .	10 10	40	43	IE 7	0 1.	1 00	BZ	60	вр	.e. \.r		н	
Use target RAM	192 KB @ 0x100000	800060 F8	B5 6F	46	44	78 00	AF	84	Ø6 A	4 0	F E3	1D	DB	68	jFD×			
Elsek memory	Internal havels 0	800070 DB	00 D3	1A	9D 4	16 Ø5	00	22	00 0	0 2:	L 68	46	D9	F6	F	"	! hF	
Rase address	0v800000	800080 4E	FD 07	22	6B 1	78 6E	46	1B	07 1	B Ø1	F D9	10	13	40	N"kx	nF	0	1
Flash size	8192 KB	800090 06	3A 96	40	28 3	78 72	54	69	46 2	2 0	ð Ø1	FØ	69	F9	.:.0(x	rTiF"	i.	
		8000A0 BD	46 F8	BD	10 1	BS 196	4G	Ø 6	4R 9	G 43	2 00	D 3	10	BD	. F	. L. К.	B	
		900000 62	69 26	69	00 3	01 00	EC.	22	ED 0	0 2	4 124	F7	CO	46	hh h +	2	4 E	,
		000000 04	04 40	00	00 0		00	70	DE O		1 11	40	ar	-10	<i>m n</i>		4	
		800000 04	74 12	00	00 3	4 12	00	70	B5 0	15 101	9 197	48	ØE	00		p		
		800000 14	00 06	FØ	F3 1	8 01	20	32	00 2	9 0	8 40	42	00	FØ		. 2.)	.08	
		8000E0 11	F9 ØØ	1 B	43	12 58	41	CØ	B2 7	10 B	D B7	1D	C1	04	CB	XAp		
		8000F0 80	21 F@	B5	3F 4	1B 40	4C	1B	698	5 B	3 22	00	18	00	.t?K	CL.i.	."	
		800100 49	00 00	93	Ø1 1	FØ 84	F8	3C	4B 3	D 41	D 22	68	2B	60	I	<k=< td=""><td>M"h+`</td><td></td></k=<>	M"h+`	
		800110 3C	4B 96	42	4C 1	D1 3C	4B	3C	48 2	B 60	a 00	9B	62	68	<k.bl.< td=""><td>< K<h+< td=""><td>`bh</td><td></td></h+<></td></k.bl.<>	< K <h+< td=""><td>`bh</td><td></td></h+<>	`bh	
-		000120 01	20 10	10		מח כיי	DD	00	70 E	E D	a on	22	20	AD	h.	10	100	
LOG																0	•	83
- FPUnit: 4 cod	le (BP) slots and O lit	eral slots																^
- CoreSight com	ponents:																	
- ROMIBILOJ @ 2	E000F0000 CTD: B105F0	OD. PTD: OCOBBOOB SI	27															
- ROMTEL[0][1]:	E0001000, CID: B105E0	OD, PID: OOOBBOOA D	Τ															
- ROMTb1[0][2]:	E0002000, CID: B105E0	OD, PID: OCOBBOOB F	PB															
- Executing ini	t sequence																	
- Target interf	face speed: 4000 kHz (F	ixed)																
- J-Link found	1 JTAG device. Core ID	: OxOBC11477 (None)																
- Connected suc	cessfully																	~
<																		>
Erase, program a	nd verify target								Cor	nnect	ed	Cor	e Id:	0x0	BC11477	Spee	d: 4000) kH

(2) 方法 2: 使用 keil 下载

BLUEX

A. 将 BlueX 文件夹中的[APOLLO_00_1V8.FLM] 和[APOLLO_00_3V3.FLM] 拷贝到 keil 文件目 录的安装位置, Keil_v5/ARM/Flash 如下图:

	Windows (C:) > Keil_v5 > A	RM → Flash →			~
	名称 ^	修改日期	类型	大小	
	 M29W640F	2020/5/29 16:36	文件夹		
<i>ж</i>	RC28F640J3x_x2	2020/5/29 16:36	文件夹		
R	S29GL064Nx2	2020/5/29 16:36	文件夹		
*	SP29JL032H	2020/5/29 16:36	文件夹		
*	AM29F160DB.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29F160DT.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29F320DB.FLX	2015/7/8 16:30	FLX 文件	14 KB	
529	AM29F320DBx2.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29F320DT.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29F320DTx2.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x033.FLX	2015/7/8 16:30	FLX 文件	13 KB	
	AM29x128.FLM	2015/7/8 16:30	FLM 文件	13 KB	
	AM29x128.FLX	2015/7/8 16:30	FLX 文件	13 KB	
	AM29x800BB.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x800BBx2.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x800BT.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x800BTx2.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x800DB.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	AM29x800DBx2.FLX	2015/7/8 16:30	FLX 文件	14 KB	
	APOLLO_00_1V8.FLM	2019/9/27 8:59	FLM 文件	90 KB	
	APOLLO_00_3V3.FLM	2019/9/27 8:59	FLM 文件	90 KB	

B. 打开 keil, 根据下图序号顺序操作:

File Edit View Project Flash Debug	Peripherals Tools SVCS Window Help
- C - C - C - C - C - C - C - C - C - C	← → 作 弛 弛 弛 症 症 //ε //ε 🙋 advData 🛛 🔍 🗟 🌮 🔍 🖕 ○ 🔗 🗶 💼 - 🔍
🛞 🍱 🎬 🥔 🚟 🕍 template	
Project 4	by cost of a contract by contract by contract by the cost of the c
Project: ble_base	
🖃 🚂 template	2
😥 🧰 bx/core	3 * @file : .h
🗈 🧰 bx/ble	5 * Southor : 2.
😑 🦢 bx/log	6 * Anrief ·
⊕ nr_micro_shell.c	7 🛛 Options for Target 'template' X **
	9 Davido Target Dutant Listing Mean C/C++ Are Liston Tahung Weiliting
⊕ <u></u> ansi.c	10 bereg raiger journel out job jam jame and other
bx_shell.c	11 C Use Simulator <u>with restrictions</u> Settings C Use: J-LINK /J-TRACE Cortex Settings
i bxsh_uart.c	12 Limit Speed to Real-Time
	14 🔽 Load Application at Startup 🔽 Run to main() 🖾 Load Application at Startup 🔽 Run to main()
SEGGER RTT printf c	15 Initialization File: Initialization File:
bx dbg asm.s	10 III III IIII IIII IIII IIII IIII III
bx/drivers	18 🛱 Restore Debug Session Settings
components	19 V Breakpoints V Toolbox
	21 F Watch Windows & Performance Analyzer F Watch Windows
🗈 🧰 user/profiles	22 🗄 🔽 Memory Display 🔽 System Viewer 🖾 Memory Display 🔽 System Viewer
🖨 🦢 user/service	23
	24 CPU DLL: Parameter: Driver DLL: Parameter:
🖃 🦢 user/app	26 SARMCM3.DLL SARMCM3.DLL
user_app.c	27 Dialog DLL: Parameter: Dialog DLL: Parameter: */
⊕- <u></u> user_ble.c	29 DARMCM1.DLL pCM0+
user_ble_task.c	30
· iser_ble_profile.c	31 Manage Component Viewer Description Files
user/drivers	32
ere crg	34 OK Cancel Defaults Help
bx_config.n	35
d by sys_coning.in	30 1 37 Hindef HW ECC PRESENT
Project (Books {) Func 0, Temp	<



C. 在弹出的对话框中选择[Flash Download]

Cortex JLink/JTrace Target Driver Setup					>
Debug Trace Flash Download					
J-Link / J-Trace Adapter	-SW Devic	:e			
SN:		IDCODE	Device Name		Move
Device: J-Link	SWDI	e	ARM CoreSig	nt SW-DP	Up
HW : V9.50 dll : V6.44f					Down
FW : J-Link V9 compiled Dec 13 2					
Port: Max	🕼 Auto	matic Detection	ID COD	E:	
SW 💌 5 MHz 💌	C Man	ual Configuratio	n Device Nam	e:	
Auto Clk	Add	Delete l	Jpdate IR le	n:	
Connect & Reset Options Connect: Normal ✔ Reset: No	rmal	Ca V V	che Options Cache <u>C</u> ode Cache <u>M</u> emory	Download Op	otions de Download I to <u>F</u> lash
Interface TCP/IP				M	lisc
USB O TCP/IP Network Se	ettings	Dort	(Autor Aut	odetect	JLink Info
Scan 127 .	0.0	. 1 :	0		
State: ready		ļ		Ping	JLink Cmd
			确定	取迷	应 田(A)
			WEAL	-4X/F	A

D. 然后根据下图配置 keil 下载:

RF01/RF03/RF04/BX2416/BX2417/BX2418 选择 [Apollo_00_3V3] RF08 选择[Apollo_00_1V8]

Cortex JLink/JTrace Target Driver Setup	<
Debug Trace Flash Download	
Download Function C Erase Full Chip C Erase Sectors C Do not Erase Programming Algorithm Programming Algorithm	
Description Device Size Device Type Address Range	
APOLLO_00_1V8 8M Ext. Flash SPI 00800000H - 00FFFFFFH	
RF08 选择 APOLLO_00_1v8 RF01/RF03/RF04/BX2416/BX2417/BX2418 选择 APOLLO_00_3v3 _{Start:} 0x00800000 Size: 0x00800000	
Add Remove	
确定 取消 应用(A)	

E. 注意在使用 keil 下载之前要先使用 J-Flash 下载一个带用 bootloader 的 hex 文件, 如 SDK 3.x 工程编译后, 工程目录下生成的[template_with_bootloader.hex]



F. 使用 J-Link 连接电路板后, 即可通过点击下图箭头指向的[Load] 按钮下载程序到电路板





4.2 擦除 Flash 中的程序

(1) 与程序烧录相同, 使用 J-Flash 连接电路板成功后, 依次点击以下按钮 Target -> Manual Programming -> Erase Chip 或在下载页面直接按[F4] 擦除 Flash

									_		_	_	_	_			_			_	
SEGGER J-Fla	ash V6.44f - [new projec	t *]																	-		×
File Edit View	Target Options W	indow Help																			
Project - nev	Connect				四、软件和	l¥/SI)K\BX	Mi	cro-9	SDK:	3-rel	ease	-v3.2	2-202	2101	04\5		example.			x
Ne reject her	Disconnect				- (atin													, en			
Name				D0		x1	<u>×2 ×4</u>	1													
Host connection	Test		>	ЬT	2 3	4	5	6	2	8	9	0	B	C	n	F	F	ASCII			
Target interface	Production Prog	ramming F	7	0	22 24	00	90 1	2	00	10	14	90	00	10	92	12	00	BY24			
Init SWD speed	Froduction Frog	ranning r	<u> </u>	10	J6 J7	00	1 90		00	10	17	20	60	25	4.4	16	20	D064			
SWD speed	. Manual Program	iming	>		Secure	e Chip					00	38	60	35	11	01	29			5)	
HOU	DL	000020	00		Unsec	ure Ch	nip				06	92	ØF	D3	10	DB	68	iFBx			
MCU Coro	BlueX Apollo_UU_1V8	800030	DB								46	00	78	01	FØ	8E	F9	F	iF.	.×	
Endian	Little	800040	07		Check	Blank			F2	2	07	1B	ØF	DA	10	ØB	40	.t. kx	1F	e	
Check core ID	No	800050	98		Erase	Sector	s		ES	3	1E	98	41	CØ	B2	BØ	BD	.e.\.F	.ec	.A	
Use target RAM	192 KB @ 0x100000	800060	F8			cl. 1	_				06	A4	ØF	E3	1D	DB	08	jFD×			
-		800070	DB	L	Erase	Chip			F4	ŧ.	00	00	21	68	46	D9	F6	F		. ! hF	
Flash memory	Internal bank 0	800080	4E		Progra	am			F	5	07	1B	ØF	D9	10	13	40	N''kx	nF	e	
Base address	0x800000 9192 KB	800090	86		riogn						46	22	ดด	Ø1	FØ	69	F9		PT iF	·i.	
1 10011 0120	0132 10	800000	BD		Progra	am &	Verity		Fe	0	AR	90	42	00	D 3	10	BD	F	T. W	B	
		0000h0	60		Varifi						PD	20	24	D4	50	20	40			4 F	
		800080	62		verny				FC	•	PD	08	34	F4	Er	00	40	bn n.1		.4F	
		800000	04		Read	back				>	82	05	00	65	48	ØF	99		p	· · · H · ·	
		800000	14								00	29	00	40	42	00	FØ		. 2.)).CB	
		8000E0 :	11		Start A	Арриса	ation		PS	,	B2	70	BD	B7	1D	C1	04	CE	XA	····	
		8000F0	80	21	FØ B5	3F	4B 4	10	4C	1B	69	85	BØ	22	00	18	00	? 8	CL.i.	"	
		800100	49	00	00 93	01	FØ 8	14	F8	3C	4B	3D	4D	22	68	2B	60	I	<к	•M"h+`	
		800110	3C	4B	9A 42	4C	D1 3	BC	4B	3C	48	2B	60	00	9B	62	68	<k.bl.< th=""><th>< K<h +<="" th=""><th>•`bh</th><th>-</th></h></th></k.bl.<>	< K <h +<="" th=""><th>•`bh</th><th>-</th></h>	•`bh	-
1		000100	0.1	۲0	10 10	DD	ז רים	NA.	DD	99	20	EE	ħΘ	00	22	20	AD	L		V 011	-
LOG																					83
- 2048 * 4 KB (- Start of preg - End of prega - Start of det - End of detern - CPU speed con - Start of era: - End of erasis - Start of resto - Erad of resto - Erad of resto	8 0x00800000 paring flash programming ining dirty areas in sining dirty areas ining dirty areas ind cho be measured. sing chip to chip toring ing ing completed successful	flash cache Lly - Completed	afte	er 1.	109 sec																^
																					~
<																					>
Erase entire chip											C	Conn	ected	Ы	Cor	e Id	: 0×0	BC11477	Spee	ed: 4000	kH

(2) 擦除成功, 如下图:

SEGGER J-Fl	ash V6.44f - [new proje	ct *]	- 🗆 ×
File Edit View	v Target Options	Nindow Help	
Project - ne	w p 🗖 🗖 🔀	LevBULEX/2_文档资料\软件相关\SDK\BXMicro-SDK3-release-v3.2-20210104\SDK3\e	example 🗖 🔳 🖾
Name	Value	Address: 0x800000 x1 x2 x4	
Host connection	USB [Device 0]	0ddness 0 1 2 3 4 5 6 7 8 9 0 B C D F F	ASCII
Target interface	SWD	800000 42 58 32 34 00 80 12 00 1C 14 00 00 19 82 12 00	BX24
Init SWD speed	4000 kHz	800010 00 00 04 00 00 00 07 00 08 00 3B 60 35 11 01 29	::`5>
SWD speed	4000 KH2	800020 B0 B5 69 46 42 78 00 AF 92 06 92 0F D3 1D DB 08	iFBx
мси	BlueX Apollo 00 1V8	800030 DB 00 CB 10 9D 46 05 00 69 46 00 78 01 F0 8F F9	F. iF.x
Core	Cortex-M0	800040 07 21 01 20 6B 78 6C 46 1B 07 1B 0F DA 10 0B 40	t kylF
Endian	Little	9000E0 99 40 02 EC DD 46 10 40 42 1E 99 41 C0 D2 D0 DD	
Line brack DAM	No 193 KB (\$100000		47D
Use target hAM	132 NB @ 0X100000	800060 F8 55 6H 46 44 78 00 HF H4 06 H4 0F E5 1D DB 06	
Flash memory	Internal bank 0		·····F··· ···/nF···
Base address	0x800000	J-Flash V6.44f X 13 40	N"KXNF
Flash size	8192 KB	69 F9	
		10 BD	.FL.K.B
		Frase operation completed successfully Completed after 1.051 sec CO 46	bh h.!24F
		0E 00	рН
		00 F0	2.>.@B
		C1 04	CBXAp
		确定 18 00	.!?K@L.i"
		2B 60	IK=M"h+`
		800110 3C 4B 9A 42 4C D1 3C 4B 3C 48 2B 60 00 9B 62 68	<k.bl.<k<h+`bh< td=""></k.bl.<k<h+`bh<>
		0001100 01 40 10 10 EE ET NO EE OO 00 00 00 00 00 00 00 00 00 00 00 00	L /^ UOV
LOG			
- 2048 * 4 KB	@ 0x00800000		^
- Start of pre	paring flash programmi	ng	
- Start of det	ermining dirty areas i	n flash cache	
- End of deter	mining dirty areas		
- CrU speed co - Start of era	uia not de measurea. sing chip		
- End of erasi	ng chip		
- Start of res - End of resto	toring		_
- Frase operat	ion completed successf	ally - Completed after 1.051 sec	
			Ŷ
			·



4.3 Debug 与查看 Log

- (1) 使用 Keil Debug
 - A. 在 Keil 界面下, 点击[Start Debug], 按钮进入调试(调试过程中确认 J-link 连接正常)

Eile Edit View Project Flash Debu	ig Perjohenals Tools SVCS Window Help	
n 🖬 🖬 🖉 🕹 🖦 🖉 🔊 າ	← → 門 抱 抱 洗 <i>示 示 正 压</i> (2) → soluta □ 承 ● ○ ◇ ● (2) ● ○ ◇ ● (2) ● ○ ◇ ●	
* R.O. 000 + 0 + 1		
Reimer/停止 单步调试• D		a 🖬
Terister Value	00x000035C DDFC 8E0 0x00000355	^
Core	0x00000052 68E1 LDR rl,[r4,#0x0C] 0x0000036 4788 87X rl	
R1 0x00000000	0x00000342 E7F9 B 0x00000355	
R2 0x00000000 R3 0x0000000	0x00000364 6866 LDR rl,[r5,5034] 0x00001364 6811 0285 rl, r2	
84 0x00122004		~
B6 0x0012F000		- ×
B7 OxF7F7F7F7 B0 OxF7F7F7F7	οιζοτζοπήση ανήμετατήση σείζοστήση <u>ανήμετατήση</u> οιε <u>ζοτικέ</u> αυτητικέ αυήσης ατοχίτες αυτήσης του ματοχισικήση ματητικός συκά	• •
89 0x77777777 810 0x777777777	2	
R11 OATTTTTTT	3 * Stile : .h	
R13 (SP) 0x0012PC38	5 * Øsuthor :	
E14 (LR) 0x00000345 E15 (FC) 0x0000035C	6 * Brief :	
E 2753 0x6100000	8 * Østention	
B System	9 * dbl/center/Aconvi Convrinti(c) - NUTX Microelectronics.	
Mede Thread	11 * All rights reserved.	
Stark MSP		0
	14	
	15 */	
	17 /* Define to prevent recursive inclusion*/	
	16 Diffield BX_SYS_CONFIG.H	
	21 Carden optasplus	
	23 Feeding	
	24 -	
	26 / Includes	
	27 /* exported define*/	
	29 Bindef HM BX VERSION	
	30 Madrine HW EX VERSION 00	
寄存器	32 FEDIL	
	33 Bifndef BLE USER CONN NUM	
	31 FORLING DLG_UDDLg_UDDlg_UDd	~
Registers		,
Command	California Parale VII. DEVICE VII. CONSERVENT	0
* JLink Info: Reset: Reset det	e arter remervia usina.vv_vommans. vice via XIGR.SVSRSSTREQ. Name LocationV Type	
BS \\template\////	components/blues/ble/controller/drives/flash_cache.cl.36, 1 单键	
B3 ((template)/////	components/sluew/sle/vontolier/streer/st	
<.	>	
2		
ASSIGN BreakDisable BreakEnabl	is Breakkill Breaklet Breaklet Breaklet Breaklet Drink Dir Display Enter Evaluate Galammak-baak and Amenyi	
	J-LINK / J-TRACE Cortex t1: 0.0000000 sec L12 C4 CAP NUM S0	JKL OVR R/W

- (2) 使用 RTT Viewer 查看 LOG
 - A. 芯片能打印 LOG 的前提是芯片已经烧录了支持 LOG 的固件
 - B. 在 SDK 3.x 的工程中, [bx_sys_config.h] 文件内有两个宏定义用于设定 Log,按照下图设定, 可以把 RTT Log 打开





C. 另外, BlueX 芯片在开启休眠情况下,无法用 J-Link 连接上的开发者可以将 [bx_sys_config.h] 文件中的[DEBUGGER_ATTACHED] 定义为 1,如下图,當 [DEBUGGER_ATTACHED] 定义为 1 后,芯片在休眠状态下也可以连接 J-Link,方便调试操 作,需要注意的是,调试完成后需要再把[DEBUGGER_ATTACHED] 定义为[0],否则芯片 再休眠期间的功耗会增加

<u>File Edit View Project Flash Debug</u>	Peijpherals Iools SVCS Window Help
🗎 💕 🛃 🥔 🐰 🖦 🛍 🤊 🥲	🖛 => 陀 喪 喪 : [注 / 注 / / / / / / / / / / / / / / / /
🗇 🖾 🛍 🥔 🔜 🙀 template	🛛 🔊 📥 🖶 🗢 🗇 🎰
Project 🗴 🕱	🚺 bx_sid\$_config.h 📋 bx_sig_config.h 📋 bx_sig_config.h 🗋 bx_sig_config.h 📋 bx_sig_config.h 📋 bx_sig_config.h 📋 bx_sig_config.h 📋 bx_sig_config.h
Project: ble_base	167 #endif
🖮 💭 template	168 -
🗉 🧰 bx/core	169 #ifndef EXT NAKE UP ENABLE
😐 🧰 bx/ble	10 FORTINE EXI WARE UP ENABLE 1
🖨 🗁 bx/log	
nr_micro_shell.c	173 = #if (RC32K_USED 0)
ansi_port.c	174 Estindef LPCLK_DRIFT_MAX
ansi.c	175 #define LPCLK_DRIFT_MAX 20
bx_shell.c	1/0 - #engli 1/7 = #else
bxsh uart.c	178 #define LPCLK DRIFT MAX 500
🛞 📄 bx dba.c	179 #endif
B SEGGER RTLC	180 -
B SEGGER BTT printf.c	
by dbg asm s	133 HEADING DEBUGGER ATTACHED 1
by/drivers	184 #endif
	185 -
by the c	186 //#define DO_NOT_WAKEUP
urer/profiler	187 188 Hiifndef FDFFDTOS WARFID DFLAV
user/promes	199 #define FREENTOS WAKEUP DELAY (BLE WAKEUP TIME - 200)
B and user copice bloc	190 #endif
user_service_ble.c	191
user app	192 Hinder XIAL STARTUP TIME
user blo s	199 fendir in a since
user ble task s	195 -
user ble profile s	196 = #ifndef LDO_3V1_OUTPUT_SLEEP_RET
asei_bie_prome.c	197 #define LDO_3V1_OUTPUT_SLEEP_RET 1
user/drivers	190 -
in a cig	200 ⊟≢ifndef LDO 1V8 OUTPUT SLEEP RET
Bx_config.n	201 #define LDO_1V8_OUTPUT_SLEEP_RET 1
bx_sys_contig.h	202 fendif
bx_pcb_contig.n	203 F
bx_app_config.h	205 #define VDD 1V8 SLEEP LDO1 1
bx_ip_config.h	206 #endif
I hu sdb2 contra h	207
E Project Books {} Func 0, Temp	

- D. 完成上述内容修改好代码后,使用 Keil 完成编译,并把固件烧录到电路板上,便可以使用 RTT Viewer 查看 LOG
- E. 搜索已经安装的 RTT Viewer 程序, 如图

J-Link RTT Viewer V6.46j 应用

F. 单击打开进入 RTT Viewer 配置界面,根据 Flash 电压可以選擇[APOLLO_00_1V8] or [APOLLO_00_3V3] 來設定[Specify Target Device],接著在 Target Interface& Speed 中选择 [SWD] 和[4000kHz],再從 RTT Control Block 中选择 [Auto Detection],配置完毕以后,点 击[OK] 就可以查看 RTT log

J-Link RTT Viewer V6.46j			
le Terminals Input Logging Help			
Log All Terminals Terminal 0			
🔝 J-Link RTT Viewer V6.46j Configuration	7	×	
Connection to J-Link		-	
● USB □ Serial No			
O ICb/Ib			
O Existing Session			
Specify Target Device			
APOLLO_00_1V8	~		
Script file (optional)			
Target Interface & Speed		_	
SWD	4000 kH	iz •	
RTT Control Block		-	
Auto Detection ○ Address ○ Se	arch <u>R</u> an	ige	
J-Link automatically detects the RTT contro	l block.		
OK	Can	ncel	



G. RTT log 如下图

🔜 J-Link RTT Viewer V6.46j
<u>File</u> <u>Terminals</u> Input Logging <u>H</u> elp
Log All Terminals Terminal 0
00> rc_calib loop count i=4436
00> rc_calib loop count i=6621
00> rc_calib loop count i=6705
00> rc_calib loop count i=6743
00> rc_calib loop count i=6766
00> rc_calib loop count i=6696
00> rc_calib loop count i=6798
00> rc_calib loop count i=6803
00> rc_calib loop count i=6714
00> I/ARCH INIT:main
00> RO=0x2A
00> I/osapp.utils:osapp utils set dev init,role=0xA
00> bond manage init
00> uart framing error
00> W/osapp.uart server:gapm cmp evt operation:0x1
00> soft wakeup
00> I/osapp.uart server:noti status:67
00> soft wakeup



5. 文档修改记录

版本	修改内容	日期	作者
1.0	创建初始版本	2020/02/04	姚琪
1.1	添加开发板部分内容	2020/02/05	淡明洁
1.2	调整 debug 部分内容	2020/03/03	夏杰
1.3	调整&添加开发板的部分内容,调整封面,增加目录	2020/07/17	简任锋
1.4	更新开发板部分内容,新增附录	2021/01/18	简任锋
1.5	细化第4章节内容,更新图片,修改部分开发板说明内容	2021/06/11	简任锋
1.6	更新英文版本	2021/06/16	彭格格
1.7	增加 BX2400-eRF01e-G1a 开发板说明与原理图	2021/08/11	简任锋

6. 附录

6.1 BX2400-dRF0xp-S1c 原理图



6.2 BX2400-eRF01e-G10 原理图



6.3 BX2400-eRF01e-G1a 原理图

