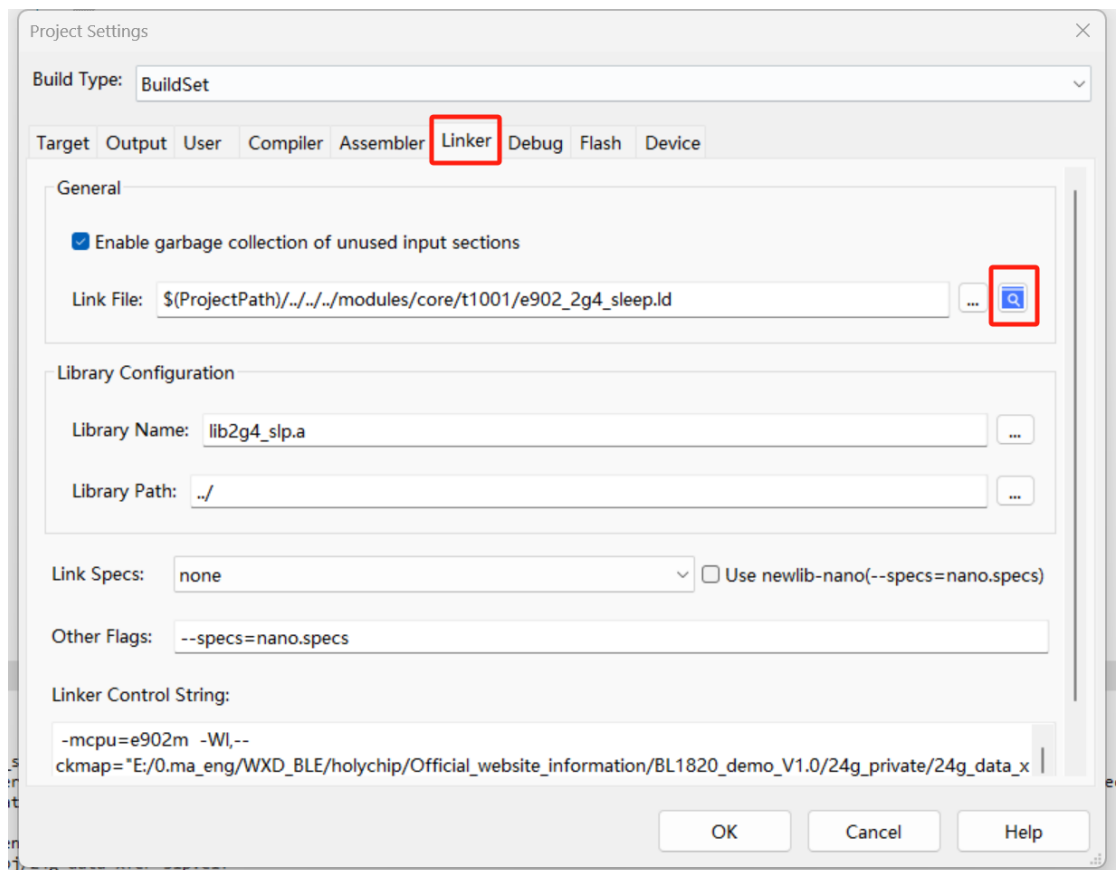
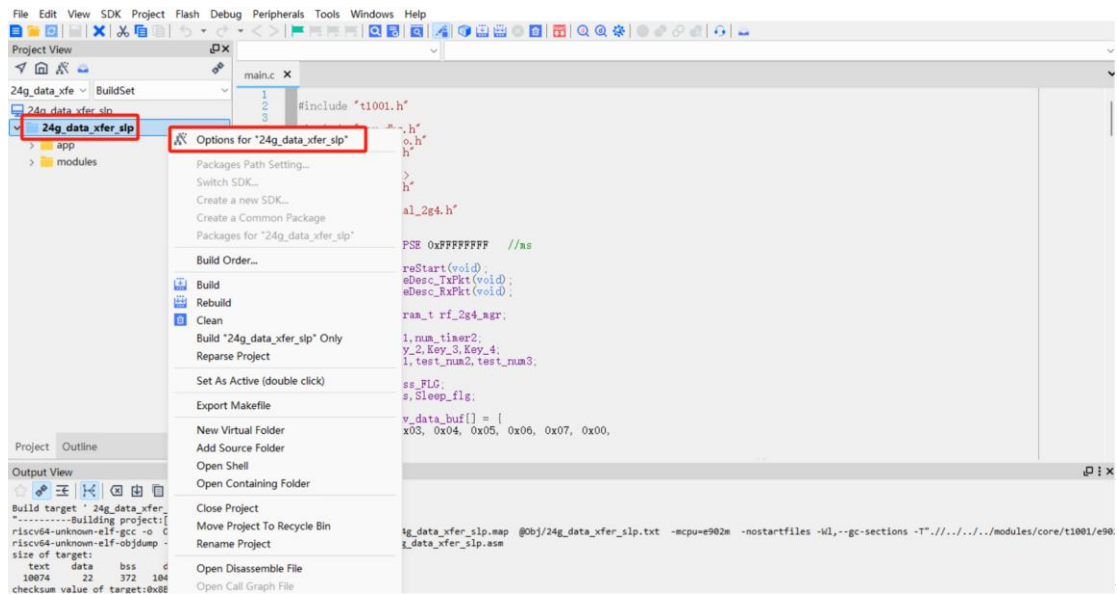


# BL1820 串口烧录流程

使用 CKlink（串口通信方式，一些常用串口工具也可以，接线为 3.3V，TX，RX，GND）进行烧录。首先串口工具的 TX 连接芯片 GPIO20，RX 连接芯片 GPIO21，3.3V 连接芯片 VDD，GND 先不连接。然后选取“OTP”，点击“读取”，再连接 GND 线（原因：串口连接会直接上电，芯片内部有程序会直接上电驱动）。



右键选择“24g\_data\_xfer\_slp” → Options for “24g\_data\_xfer\_slp” → Linker → 点击放大镜图标打开相应文件。



在 e902\_2g4\_sleep.ld 文件中 \_\_ROM\_BASE = 0x10000000; \_\_ROM\_SIZE = 0x00040000; 为开发板外接flash的地址和块大小。

如需使用OTP芯片，需修改以下两点：

1. 将地址和块大小替换为 \_\_ROM\_BASE = 0x1f800000;  
\_\_ROM\_SIZE = 0x00040000;

```
main.c e902_2g4_sleep.ld X
1 __ROM_BASE = 0x10000000;
2 __ROM_SIZE = 0x00040000;
3
4 __RAM_BASE = 0x20000000;
5 __RAM_SIZE = 0x00040000;
6
7 __STACK_SIZE = 0x00000300;
8 __HEAP_SIZE = 0x00000000;
9
10 MEMORY
11 {
12     FLASH (rx) : ORIGIN = __ROM_BASE, LENGTH = __ROM_SIZE
13     RAM (rwx) : ORIGIN = __RAM_BASE, LENGTH = __RAM_SIZE
14 }
15
16 /* Linker script to place sections and symbol values. Should be used together
17 * with other linker script that defines memory regions FLASH and RAM.
18 * It references following symbols, which must be defined in code:
19 *   reset_Handler : Entry of reset handler
20 *
21 * It defines following symbols, which code can use without definition:
22 *   __exidx_start
23 *   __exidx_end
24 *   __copy_table_start__
25 *   __copy_table_end__
```

```
main.c omw_config.h *e902_2g4_sleep.ld X
1 __ROM_BASE = 0x1f800000;
2 __ROM_SIZE = 0x00040000;
3
4 __RAM_BASE = 0x20000000;
5 __RAM_SIZE = 0x00040000;
6
7 __STACK_SIZE = 0x00000300;
8 __HEAP_SIZE = 0x00000000;
9
10 MEMORY
11 {
12     FLASH (rx) : ORIGIN = __ROM_BASE, LENGTH = __ROM_SIZE
13     RAM (rwx) : ORIGIN = __RAM_BASE, LENGTH = __RAM_SIZE
14 }
15
16 /* Linker script to place sections and symbol values. Should be used together
17 * with other linker script that defines memory regions FLASH and RAM.
18 * It references following symbols, which must be defined in code:
19 *   reset_Handler : Entry of reset handler
20 *
21 * It defines following symbols, which code can use without definition:
22 *   __exidx_start
23 *   __exidx_end
24 *   __copy_table_start__
25 *   __copy_table_end__
```

2. omw\_dbg.h → omw\_config.h, omw\_config.h中修改为对应的宏定义

```
main.c X e902_2g4_sleep.ld omw_dbg.h omw_config.h
1
2 #include "t1001.h"
3
4 #include "omw_dbg.h"
5 #include "omw_gpio.h"
6 #include "Inital.h"
7
8 #include <stdio.h>
9 #include "string.h"
10
11 #include "t100x_hal_2g4.h"
12
13 //only for demo
14 #define SLEEP_ELAPSE 0xFFFFFFFF //ms
15
16 void RF_2G4_PrepareStart(void);
17 void RF_2G4_UpdateDesc_TxPkt(void);
18 void RF_2G4_UpdateDesc_RxPkt(void);
19
20 extern omw_2g4_param_t rf_2g4_mgr;
21
22 uint8_t num_timer1,num_timer2;
23 uint32_t Key_1,Key_2,Key_3,Key_4;
24 uint32_t test_num1,test_num2,test_num3;
25 uint8_t TX_FLG;
26 uint8_t Key_success_FLG;
27 uint8_t Key_status,Sleep_flg;
28 //only for demo
29 uint8_t tx_ble_adv_data_buf[] = {
30     0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x00,
31 };
```

```
main.c e902_2g4_sleep.ld omw_dbg.h omw_config.h X
1 #ifndef __OMW_CONFIG_H__
2 #define __OMW_CONFIG_H__
3
4 #ifdef __cplusplus
5 extern "C" {
6 #endif
7
8 #ifndef OMW_INCLUDE_IN_ASM
9 #include <stddef.h>
10 #include <stdarg.h>
11 #include <string.h>
12 #endif
13
14 #define OMW_CHIP_T1001
15 #define OMW_HAS_LOG_UART
16 #define OMW_BLE_CNTR_V0 //V2
17 #define OMW_BLE_HOST_V0 //V1, V2
18
19 #define OMW_TIMER
20
21 #define OMW_EN_DEEP_SLEEP
22 #ifdef OMW_EN_DEEP_SLEEP
23
24 #define CONFIG_OTP_PROGRAM 0
25 #define CONFIG_FLASH_PROGRAM 1
26
27 #if (CONFIG_FLASH_PROGRAM)
28 // #define QSPI_VDD_FUNC_REG_VAL 0x18180000 //PIN 10, 11
29 // #define QSPI_VDD_OB_REG_VAL 0x000000C0
30 // #define QSPI_VDD_O_REG_VAL 0x000000C0
31
```

编译后，obj文件夹中0x1f800000为OTP文件，0x10000000为flash文件。

1820\_car test > RX\_RC\_Car > app > 24g\_data\_xfer\_slp > cdK > Obj

排序查看

名称	修改日期	类型	大小
_rt_entry.S	2025/2/13 14:31	Assembler Source	0 KB
24g_data_xfer_slp.elf	2025/2/13 14:31	ELF 文件	243 KB
24g_data_xfer_slp.txt	2025/2/13 14:31	文本文档	1 KB
24g_data_xfer_slp_0x1f800000.bin	2025/2/13 14:20	BIN 文件	12 KB
24g_data_xfer_slp_0x10000000.bin	2025/2/13 14:31	BIN 文件	14 KB
24g_data_xfer_slp_main.d	2025/2/13 14:31	D 文件	2 KB
24g_data_xfer_slp_main.o	2025/2/13 14:31	O 文件	44 KB
24g_data_xfer_slp_motor_control.c.d	2025/2/13 14:31	D 文件	2 KB
24g_data_xfer_slp_motor_control.c.o	2025/2/13 14:31	O 文件	25 KB
24g_data_xfer_slp_user_init.d	2025/2/13 14:31	D 文件	2 KB
24g_data_xfer_slp_user_init.o	2025/2/13 14:31	O 文件	33 KB
BL_bootloader_0X1F803F80.bin	2024/12/3 09:17	BIN 文件	1 KB
drv_omw_gpio.d	2025/2/13 14:31	D 文件	1 KB
drv_omw_gpio.o	2025/2/13 14:31	O 文件	30 KB