

# 算法模块烧录 key 到芯片 efuse

## 1. 硬件要求

只要是 micro (52x) 系列。PA30 都要外供，且 AVDDANA 脚要大于等于 efuseldo (2.5V) 电压；只有写 efuse PA30 才要外供 2.5V，且 AVDDANA 电压大于等于 2.5V。其他系列 (55x /56x /58x) 没有这个供电要求。

## 2. 软件要求

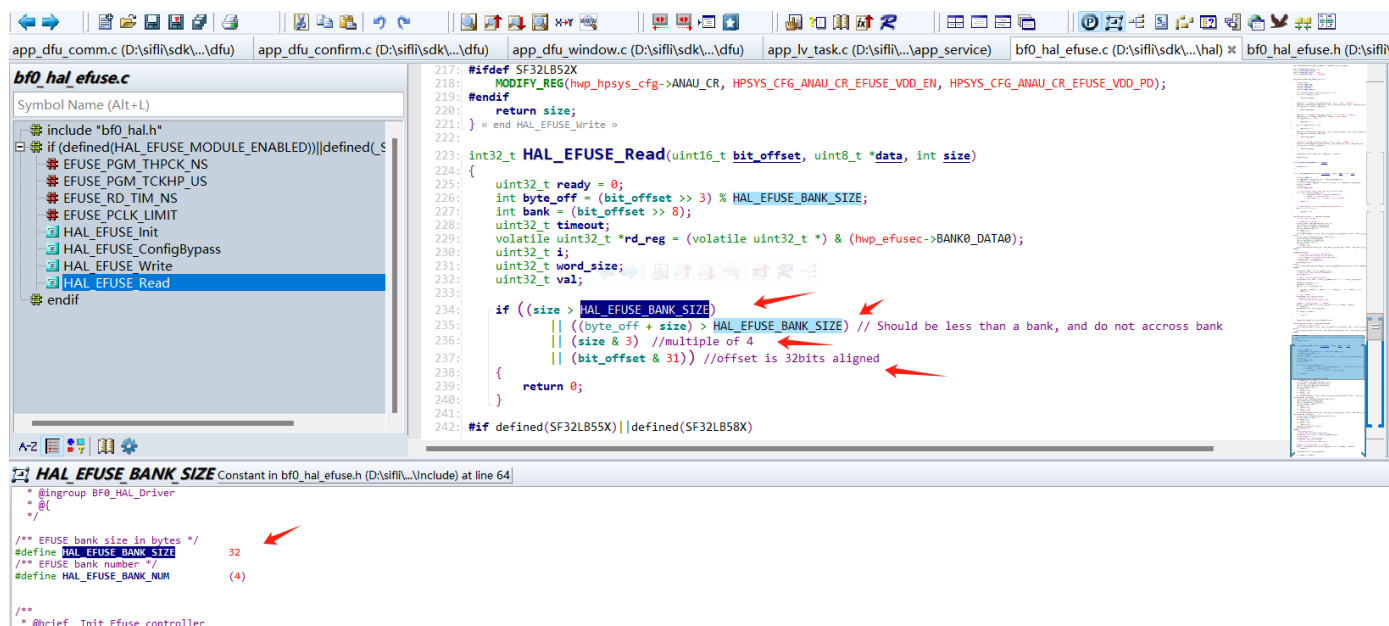
- 客户可以烧录读写的 EFUSE 区域，为 back2。back2 起始地址为 32\*2\*8.

```
int32_t ret = HAL_EFUSE_Write(32 * 2 * 8, data, 32);  
return;  
ret = HAL_EFUSE_Read(32 * 2 * 8, &score[0], 32);
```

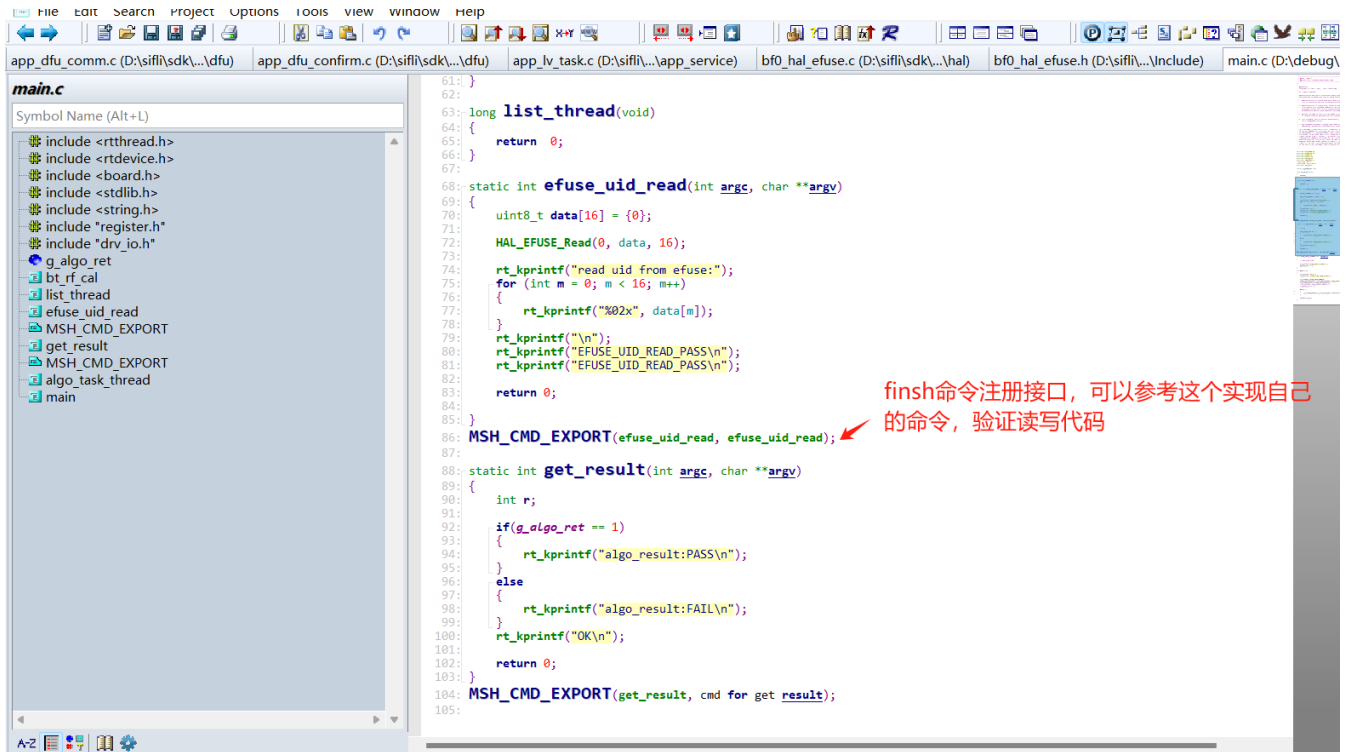
代码提供了标准的读，写接口。注意读写规则，有大小限制，要 4 字节对齐。

HAL\_EFUSE\_Write

HAL\_EFUSE\_Read



- b. 算法公司外发烧录时，对于 52x 系列（芯片没有内部合封的 flash）。我们提供了一个跑在 sram 的代码工程，在附件 [算法调试工程中](#)。



```
main.c
Symbol Name (Alt+L)
#include <rtthread.h>
#include <rtdevice.h>
#include <board.h>
#include <stdlib.h>
#include <string.h>
#include "register.h"
#include "drv_io.h"
g_algo_ret
bt_rf_cal
list_thread
efuse_uid_read
MSH_CMD_EXPORT
get_result
MSH_CMD_EXPORT
algo_task_thread
main

61: }
62:
63: long list_thread(void)
64: {
65:     return 0;
66: }
67:
68: static int efuse_uid_read(int argc, char **argv)
69: {
70:     uint8_t data[16] = {0};
71:
72:     HAL_EFUSE_Read(0, data, 16);
73:
74:     rt_kprintf("read uid from efuse:");
75:     for (int m = 0; m < 16; m++)
76:     {
77:         rt_kprintf("%02x", data[m]);
78:     }
79:     rt_kprintf("\n");
80:     rt_kprintf("EFUSE_UID_READ_PASS\n");
81:     rt_kprintf("EFUSE_UID_READ_PASS\n");
82:
83:     return 0;
84: }
85:
86: MSH_CMD_EXPORT(efuse_uid_read, efuse_uid_read);
87:
88: static int get_result(int argc, char **argv)
89: {
90:     int r;
91:
92:     if(g_algo_ret == 1)
93:     {
94:         rt_kprintf("algo_result:PASS\n");
95:     }
96:     else
97:     {
98:         rt_kprintf("algo_result:FAIL\n");
99:     }
100:     rt_kprintf("OK\n");
101:
102:     return 0;
103: }
104: MSH_CMD_EXPORT(get_result, cmd for get result);
105:
```

finsh命令注册接口，可以参考这个实现自己的命令，验证读写代码

### 3. 附件

算法调试工程.7z