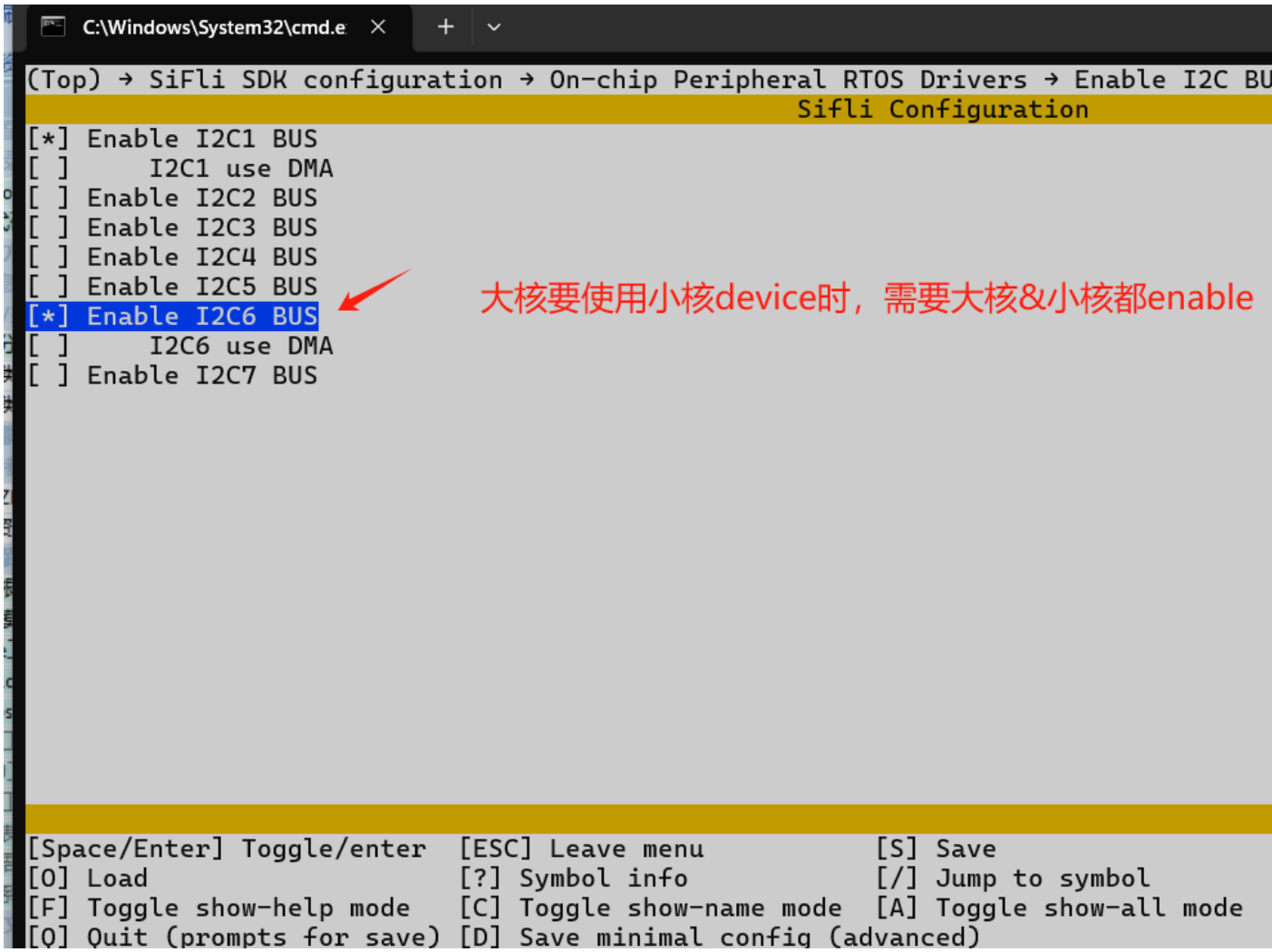
设备应用问题

1.大小核 device 配置注意事项

1.1. 大核要用小核上的 device 设备时，大小核必须都配置为打开。不然初始化后，小核会关掉 device。

例：spi3/pwm5 都是小核的设备，在大核使用时，需要同步在小核也配置为打开。

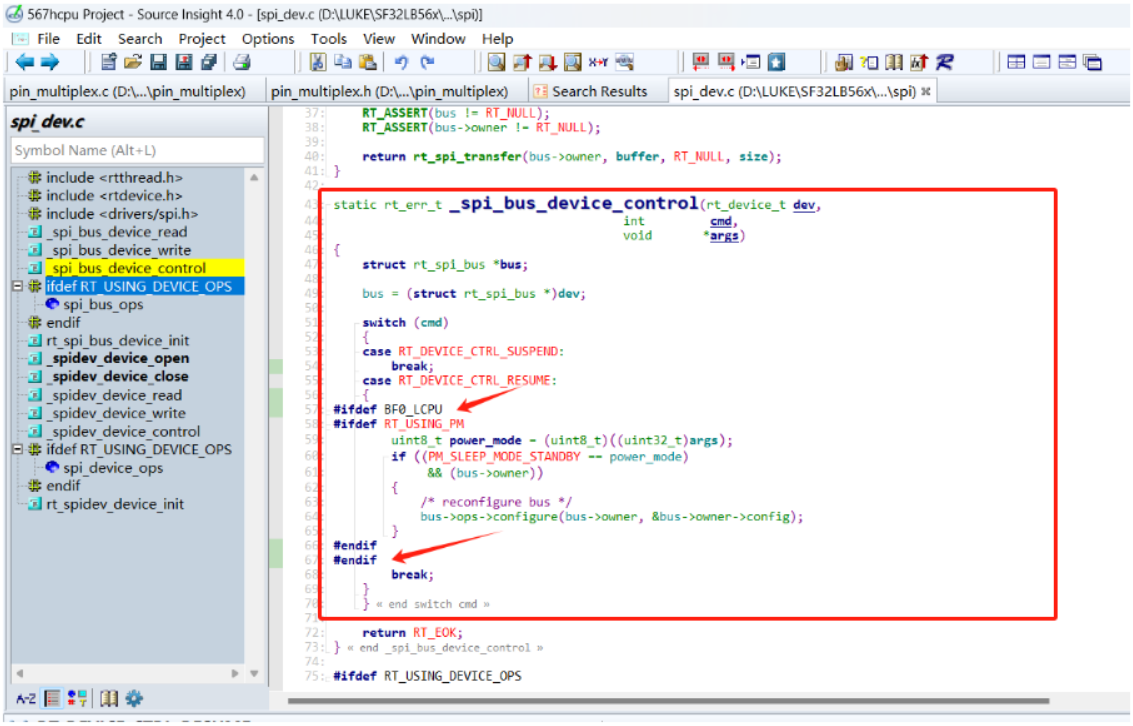
lcpu 目前代码是看该模块是不是开了，如果没开回去关掉这个模块电进行省电。但是 lcpu 跟 hcpu 又分开的，所以会导致 lcpu 把挂载在自身的 device 关掉了。大核来使用出现异常。



2.同一个 device 在大小核都要应用时，大核睡眠后，小核唤醒会读写失败。

例：spi3/I2C5

验证了几个方式，大核不用是就反注册，但是再用的时候要在注册一次。
最后还是采用了resume只小核跑的时候在配置。56x只存在大核睡了小核在跑的情景。不存在小核睡了，大核是唤醒状态。



2.睡眠唤醒注意事项

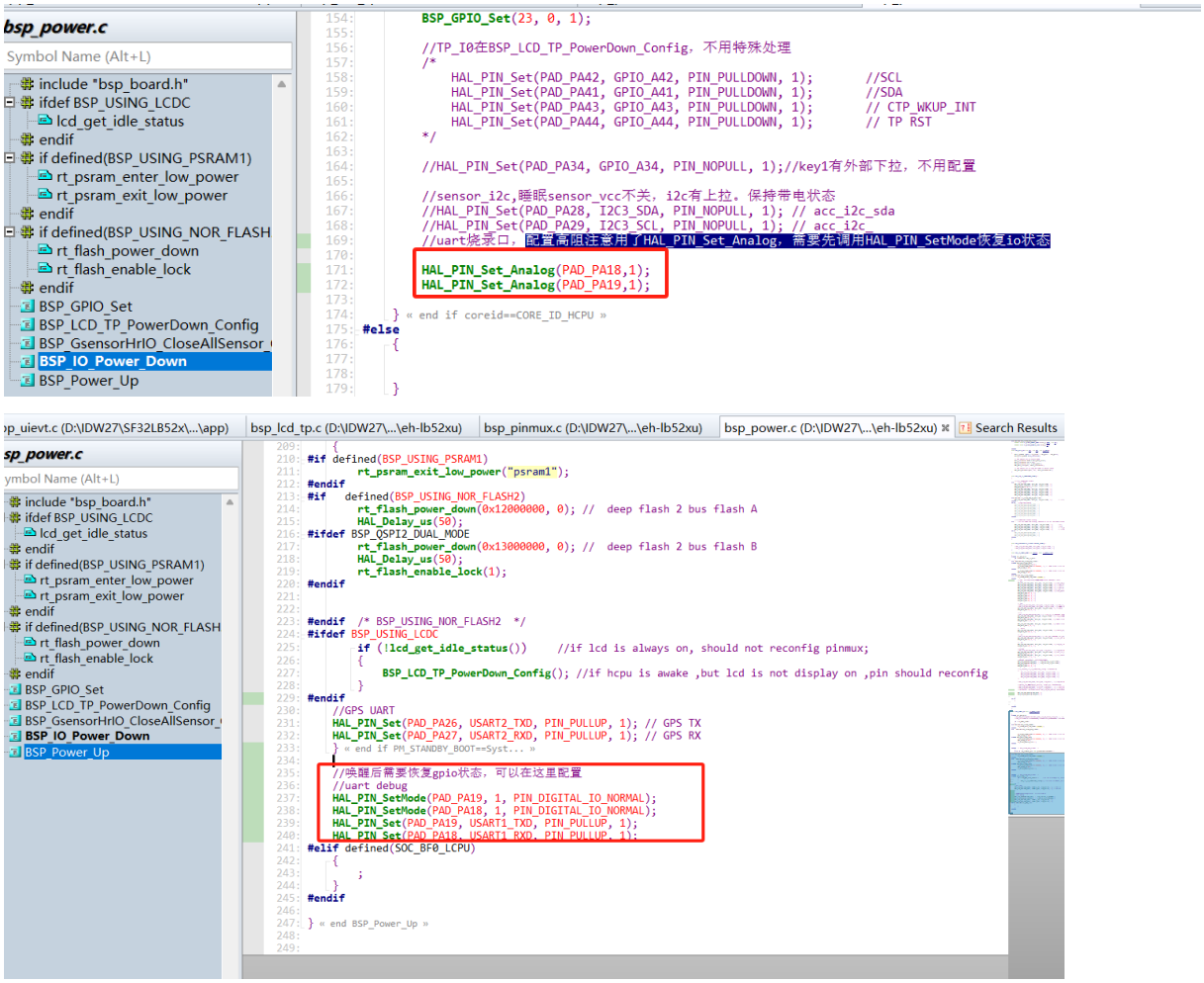
2.1. 52x 系列

2.1.1. 52x 配置高阻注意用了 HAL_PIN_Set_Analog，需要先调用 HAL_PIN_SetMode 恢复 io 状态

52x 睡眠唤醒后不会去恢复 bsp_pinmux 里的 pin 脚状态。需要在代码里主动恢复

统一管理可以睡眠前可以在 BSP_IO_Power_Down 配置睡眠的 pin 脚状态。

唤醒后可以在 BSP_Power_Up 里恢复。

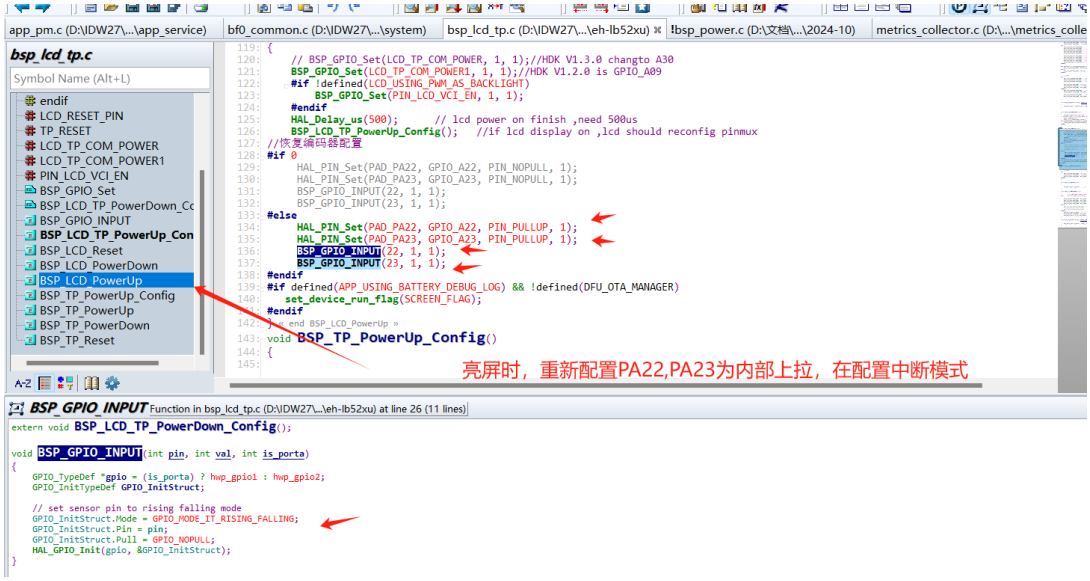


2.1.2. 52x 中断脚在睡眠前改了状态，要在唤醒前恢复中断模式。不然无法使用。

例:

无感编码器睡眠后必须关闭上拉电源。一般客户为了节省成本，使用的内部上拉。

睡眠时就会配置为内部下拉，输出 0。避免漏电。用 HAL_PIN_Set 配置下拉后，改变了中断状态。需要在唤醒后配置。



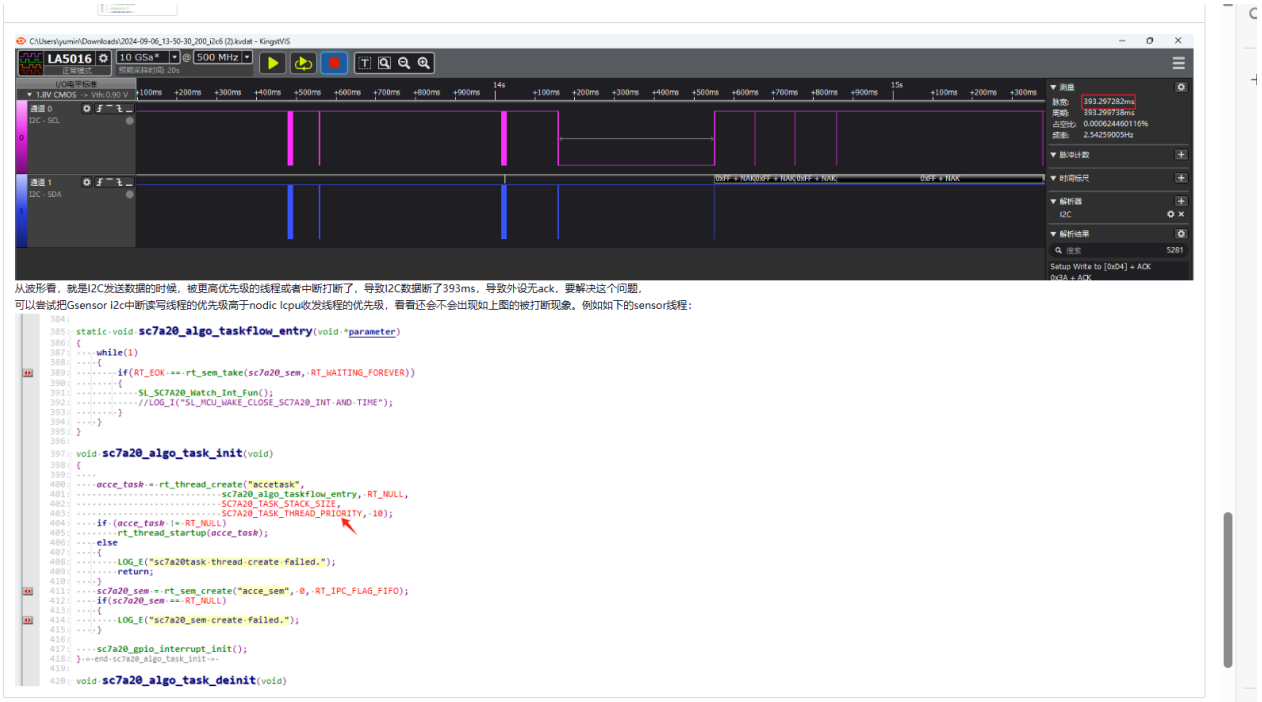
3.I2C 使用遇到过的问题

3.1. 线程优先级导致的 I2C 数据 err

问题原因：i2c 读写的线程优先级被更高的优先级打断，导致数据异常

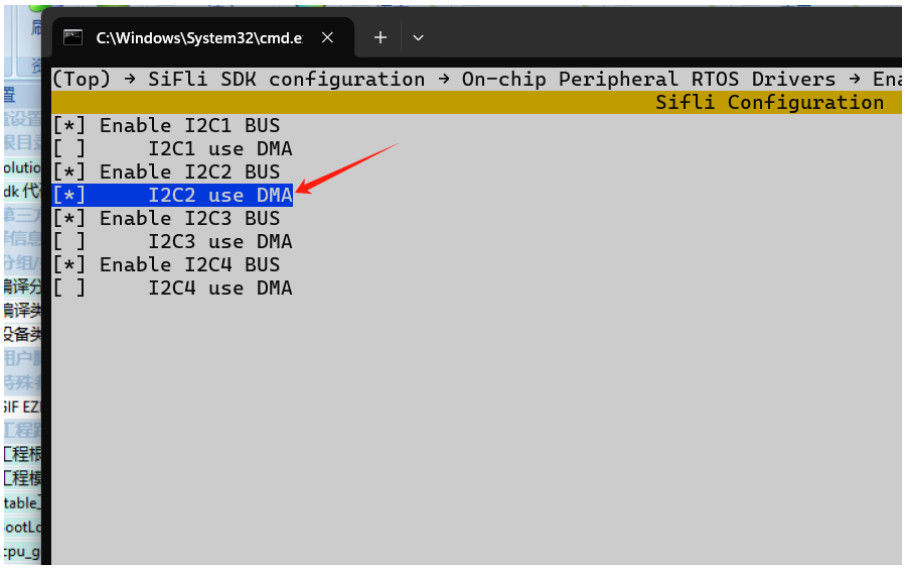
解决方法：

1. 把 I2C 读写有线程优先级提高
2. 配置 DMA 方式读写也能规避改问题，但是 DMA 通道有限。要看下有没有其他设备在用

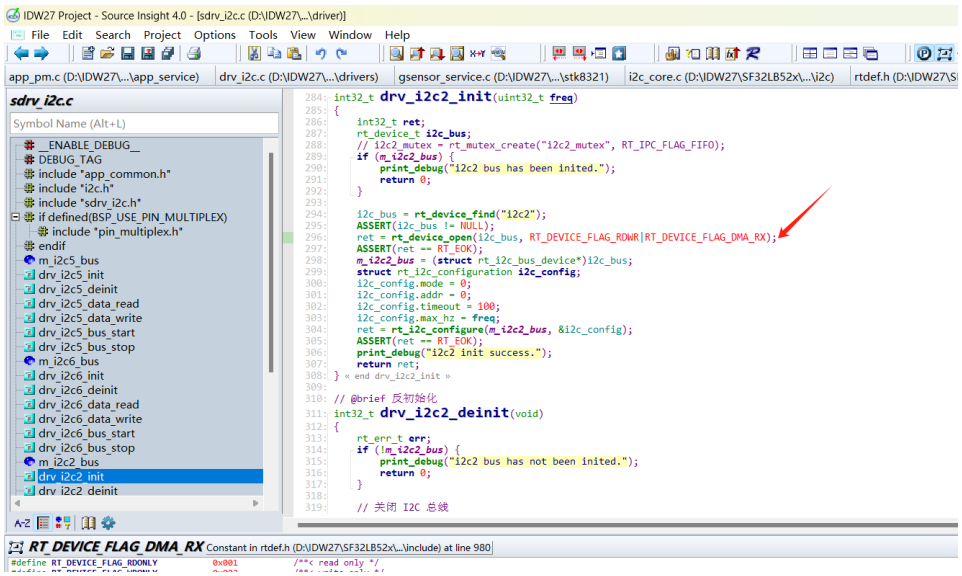


3.2. I2C 设备配置 dma

1）menuconfig 打开



2）注册 I2C 设备 open 时需要|上 dma

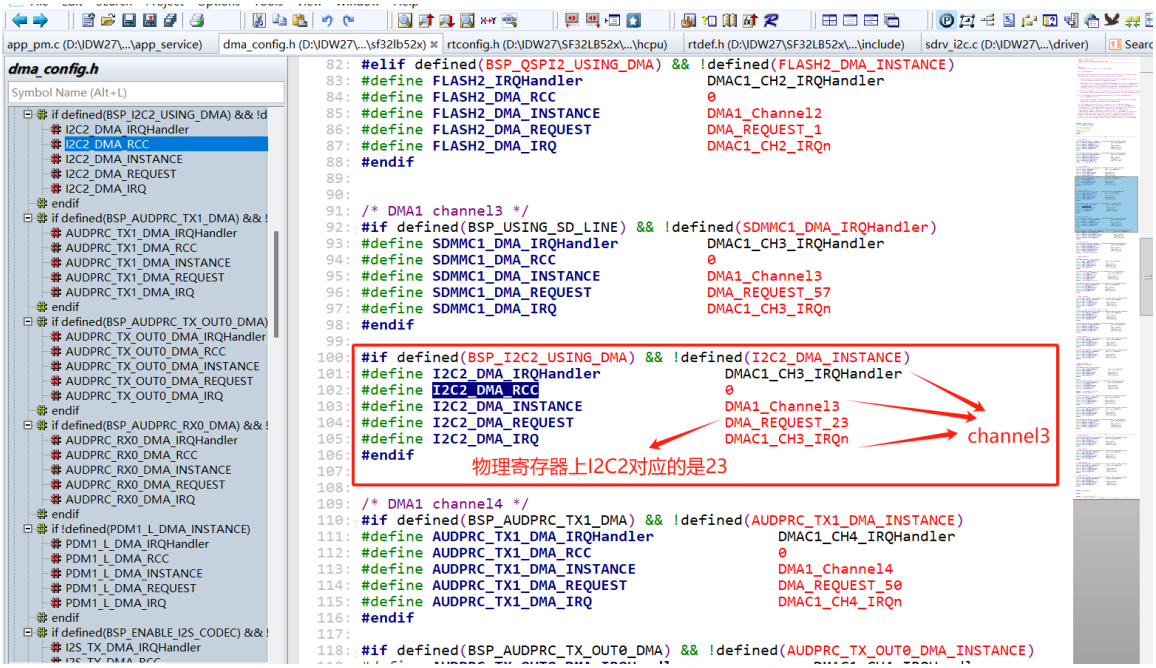


3）检查对应 I2C 的 dma 底层是否有配置 dma 通道，检查 dma_config.h

像是 52x 的 solution 代码，默认只给 I2C2 配置了 dma 通道。其他 I2C 通道没有配置。

配置新增是需要注意，I2C_DMA_REQUEST 是对应的物理寄存器地址，在芯片上是固定的。

每个外设不一样，需要看芯片手册。其他的几个参数都是对应的 channel，这个只要没被其他的设备使用都可以配置。



下图是 52x 的 DMA_REQUEST 对应 REQUEST

52x DMA req table

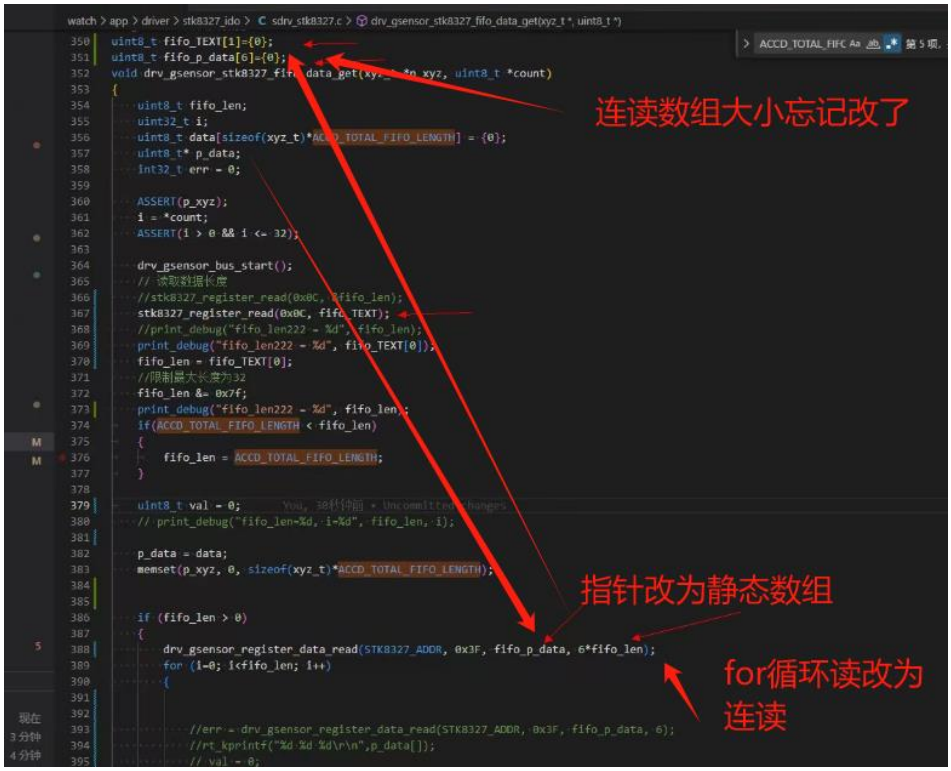
req_sel	DMAC1	DMAC2
0	mpi1	usart4_tx
1	mpi2	usart4_rx
2	/	usart5_tx
3	i2c4	usart5_rx
4	usart1_tx	/
5	usart1_rx	/
6	usart2_tx	btim3
7	usart2_rx	btim4
8	gptim1_update	/
9	gptim1_trigger	/
10	gptim1_cc1	/
11	gptim1_cc2	/
12	gptim1_cc3	/
13	gptim1_cc4	/
14	btim1	/
15	btim2	/
16	atim1_update	/
17	atim1_trigger	/
18	atim1_cc1	/
19	atim1_cc2	/
20	atim1_cc3	/
21	atim1_cc4	/
22	i2c1	/
23	i2c2	/
24	i2c3	/
25	atim1_com	/
26	usart3_tx	/
27	usart3_rx	/
28	spi1_tx	/
29	spi1_rx	/
30	spi2_tx	/
31	spi2_rx	/
32	i2s1_tx	
33	i2s1_rx	
34	/	
35	/	
36	pdm1_rx_l	
37	pdm1_rx_r	
38	gpadc	
39	adc0	

56x DMA req table

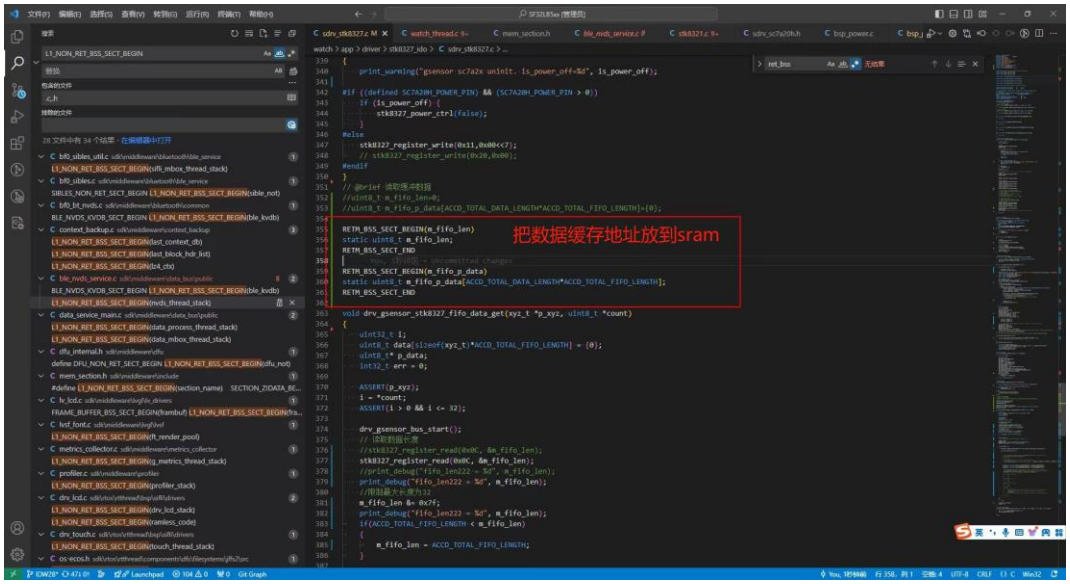
req_sel	DMAC1	DMAC2
0	mpi1	usart4_tx
1	mpi2	usart4_rx
2	mpi3	usart5_tx
3	i2c4	usart5_rx
4	usart1_tx	usart6_tx
5	usart1_rx	usart6_rx
6	usart2_tx	btim3
7	usart2_rx	btim4
8	gptim1_update	gptim3_update
9	gptim1_trigger	gptim3_trigger
10	gptim1_cc1	gptim3_cc1
11	gptim1_cc2	gptim3_cc2
12	gptim1_cc3	gptim3_cc3
13	gptim1_cc4	gptim3_cc4
14	btim1	gptim5_update
15	btim2	gptim5_trigger
16	atim1_update	spi3_tx
17	atim1_trigger	spi3_rx
18	atim1_cc1	spi4_tx
19	atim1_cc2	spi4_rx
20	atim1_cc3	mpi5
21	atim1_cc4	i2c5
22	i2c1	i2c6
23	i2c2	i2c7
24	i2c3	gptim4_update
25	atim1_com	gptim4_trigger
26	usart3_tx	gptim4_cc1
27	usart3_rx	gptim4_cc2
28	spi1_tx	audadc_ch0/gptim4_cc3
29	spi1_rx	audadc_ch1/gptim4_cc4
30	spi2_tx	gpadc
31	spi2_rx	/
32	i2s1_tx	
33	i2s1_rx	
34	sci_tx	
35	sci_rx	
36	pdm1_rx_l	
37	pdm1_rx_r	
38	pdm2_rx_l	
39	pdm2_rx_r	
40	/	
41	dac0	
42	dac1	
43	gptim2_update	
44	gptim2_trigger	
45	gptim2_cc1	
46	audprc_tx_out_ch1	
47	audprc_tx_out_ch0	
48	audprc_tx_ch3	
49	audprc_tx_ch2	

3.3. 软件算法调用 I2C dma 读数据

1）用 dma 读书时需要注意，需要配置一个全局的静态数组。如果缓存数据的地址不是固定的会导致 dma 搬数后，应用层返回的是 0 值。



2）软件运行频繁，速率高时对数组的加载位置也有影响。默认注册的静态数组是在 psram，如果读数据频繁，数据多。在 psram 搬运可能来不及，会出现上层拿到的数据部分为 0。解决方法，把注册的静态数组放在 sram 里。这么做会让 dma 搬运速度提升，避免速度不够出现的 dma 搬运没完成。



4. GPIO 配置 FAQ

4.1. gpio 中断怎么配置，中断唤醒怎么配置

solution 代码里可以参考 button_service.c 跟 battery_comm.c

```c
static void GpsEnableWake(void)
{
#ifdef BSP_USING_PM
    int8_t wakeup_pin;
    uint16_t gpio_pin;
    GPIO_TypeDef *gpio;
#endif /* BSP_USING_PM */
    log_debug("....");
    //加载中断
    rt_pin_mode(WAKE_PIN,PIN_MODE_INPUT);

#ifdef BSP_USING_PM
    gpio = GET_GPIO_INSTANCE(WAKE_PIN);
    gpio_pin = GET_GPIOx_PIN(WAKE_PIN);

#if(WAKE_PIN > 96)
    wakeup_pin = HAL_LPAON_QueryWakeupPin(gpio, gpio_pin);
#else
    wakeup_pin = HAL_HPAON_QueryWakeupPin(gpio, gpio_pin);
#endif
    RT_ASSERT(wakeup_pin >= 0);

    pm_enable_pin_wakeup(wakeup_pin, AON_PIN_MODE_DOUBLE_EDGE);
#endif /* BSP_USING_PM */

    rt_pin_attach_irq(WAKE_PIN,PIN_IRQ_MODE_RISING_FALLING,um_loda_wake_irq_handler,NULL);
    rt_pin_irq_enable(WAKE_PIN,1);
}
```

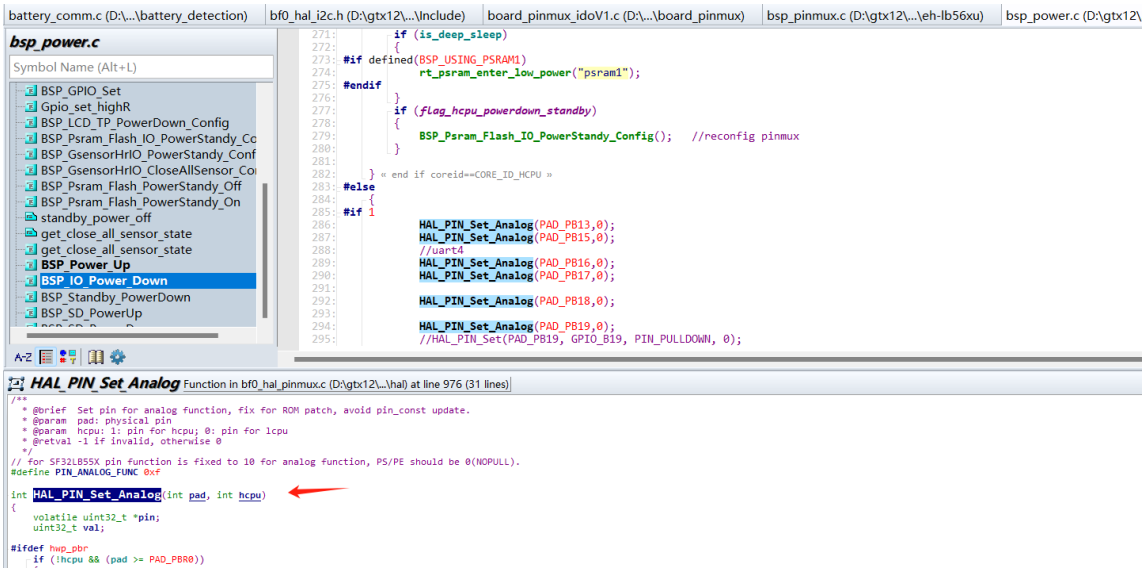
中断唤醒，PB脚在大核使用
可以直接唤醒大核。但是在小核使用
只能唤醒小核，要同步消息订阅去唤醒大核

pin脚注意区分大小核

唤醒使能

唤醒后，正常执行中断函数

4.2. 配置高阻



1）“pad”注意 PA,PB 不要搞错，PA 是 HCPU，PB 是 LCPU。

2）hcpu，这里是区分大小核的，PA 脚是大核的 PIN 脚，这里是 1. PB 脚是小核的 PIN 脚，要用 0.

3.faq

1. I2C 报错怎么办

   1）检查外部上拉，或者内部上拉。确认上拉是否异常。外部上拉是否匹配，I2C 配置的速率，外设是否支持。

例：跑 400k 速率，外部上拉用的 4.7K 或者 10k。I2C 实际测量出来的速率会不对。

2）检查外设电源，是否异常。

例：经常又同学的外设电源是跟其他地方共用的，导致电源被关。外设断电，导致 I2C 访问异常。

3）上逻辑分析仪

逻辑分析仪，是调试驱动必备的。能分析具体数据。

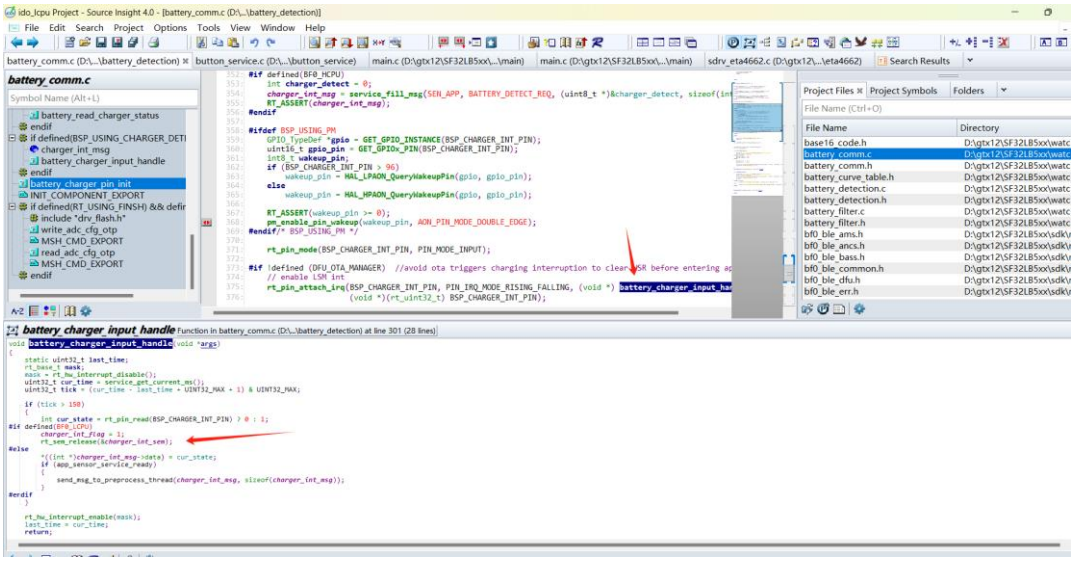
2. uart 报错怎么办

   由于 uart fifo 只有 1 字节。所有用 RX 中断方式会出现接收漏掉的情况。

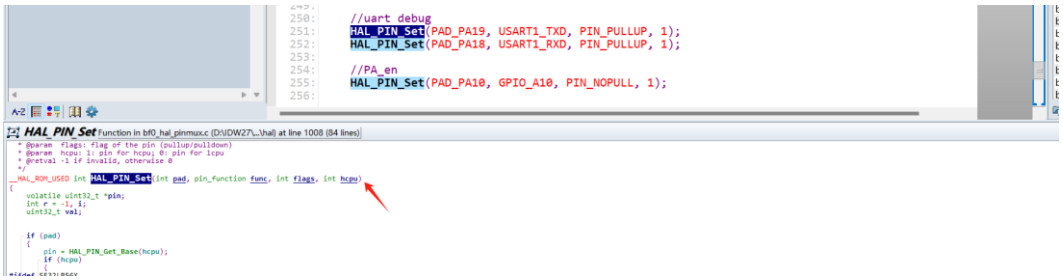
   需要用 RX_DMA 来接收。

3. 中断唤醒后，怎么马上又睡眠了。应该怎么唤醒后去做用户逻辑

   唤醒后要看中断处理逻辑，没事件需要处理。系统进入 idle 一段时间就会睡眠



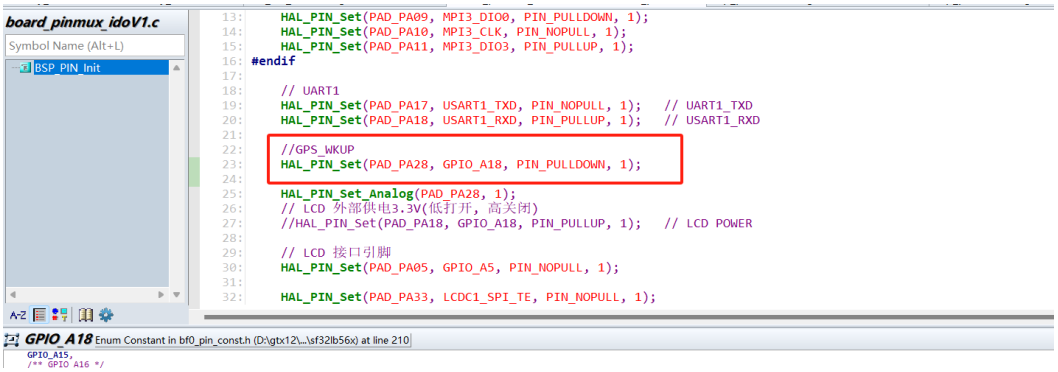
4. 为什么配置了 pin 脚不生效



1）“pad” 注意 PA,PB 不要搞错，PA 是 HCPU，PB 是 LCPU。经常有同学搞错。PB 能在 HCPU 直接调用。PA 不能直接在 LCPU 调用。

2）func，注意对照 pinmux，config 表看下是否支持，注意是否配置错了。

例：PA28,对应的是 GPIO_A28，不要复制，粘贴忘记改了。



3）flags，内部上下拉状态。有外部上下拉的，不需要再配置内部上下拉了。这里对于通讯接口（I2C/SPI）统一推荐使用外部上拉，芯片内部上拉不可调节。大概在 18K 左右，速率高的时候会出现数据异常

4）hcpu，这里是区分大小核的，PA 脚是大核的 PIN 脚，这里是 1. PB 脚是小核的 PIN 脚，要用 0.

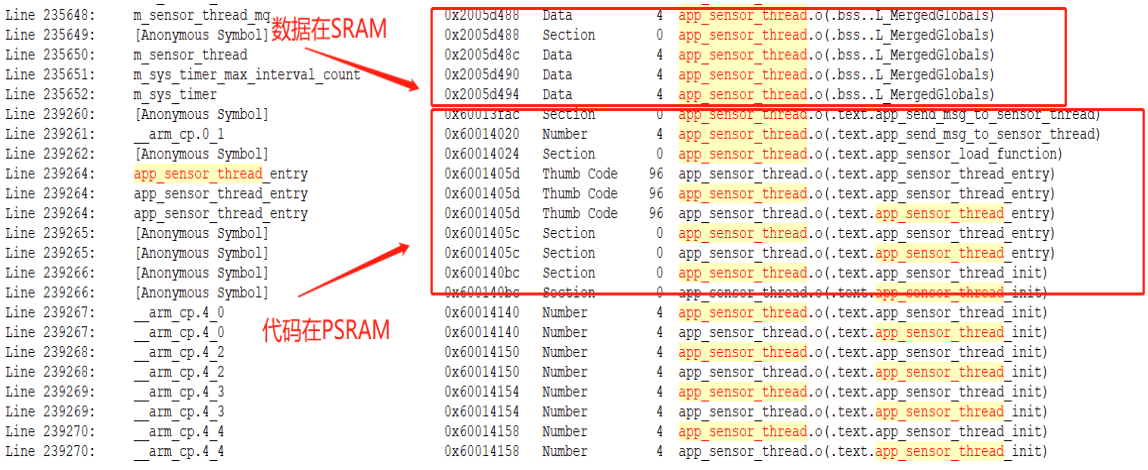
经常有同学复制，粘贴忘记改了。

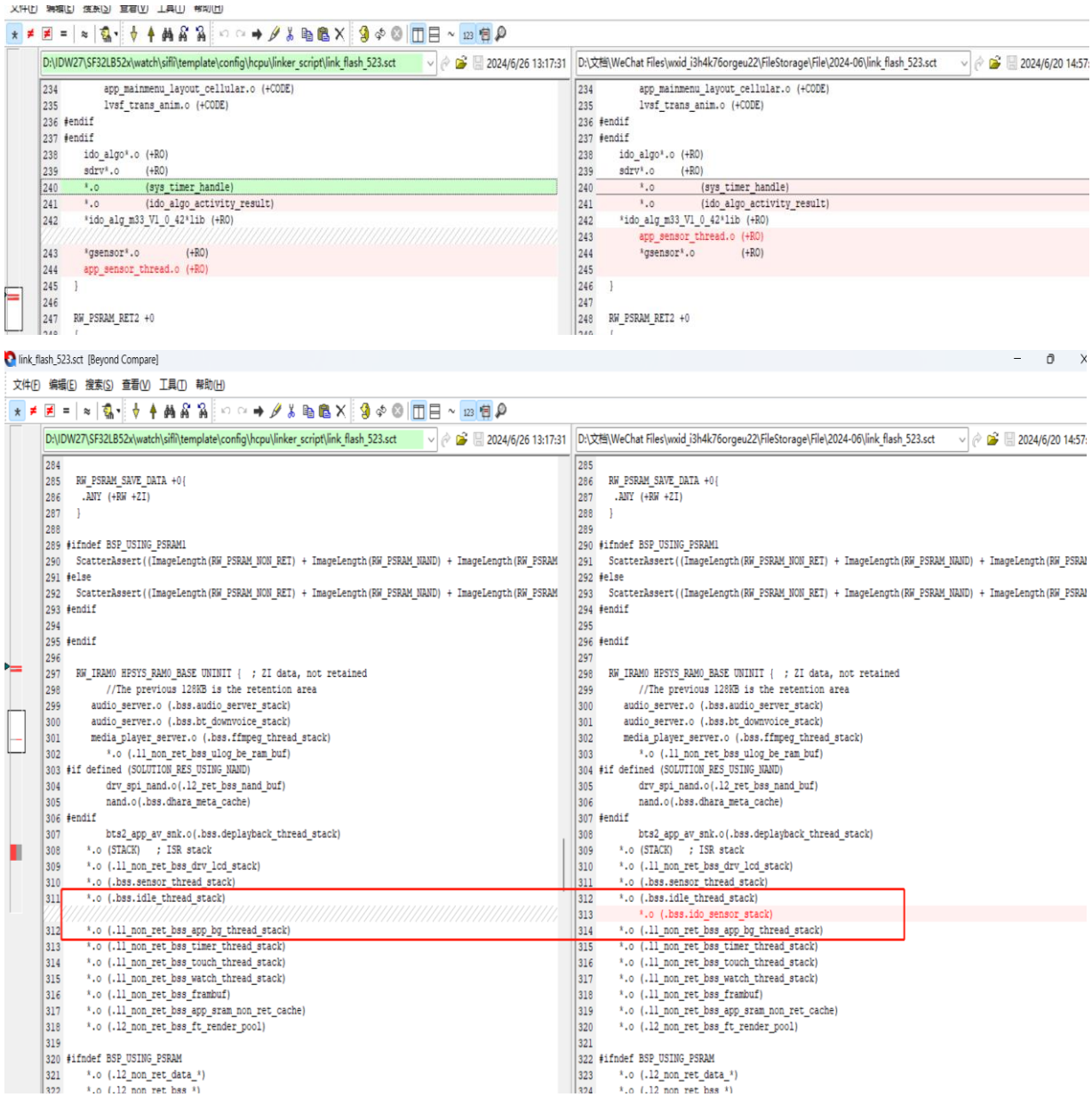
5.功耗优化（算法&数据搬运到 PSRAM&SRAM）

1）算法先从 nor_flash 移到 psram（+RO）。编译后，可以搜相关函数。看地址是不是变 0x6 开头

2）算法数据全移到 SRAM (+RW+ZI) 编译后，可以搜相关函数。看地址是不是变 0x2 开头

3）常用算法移动到 SRAM（+CODE）（验证驱动&算法相关代码搬运到 sram 处理，功耗有降低（爱都 IDW27/SIFLI523）平均功耗 30ua）.对于有功耗优化诉求的客户都可以尝试下





link_flash_523.sct [Beyond Compare]

文件(F) 编辑(E) 搜索(S) 查看(V) 工具(T) 帮助(H)

D:\JDW27\SF32LB52x\watch\sifli\template\config\hcpu\linker_script\link_flash_523.sct 2024/6/26 13:17:31

```
234        app_mainmenu_layout_cellular.o (+CODE)
235        lvsf_trans_anim.o (+CODE)
236  #endif
237  #endif
238      ido_algo*.o (+RO)
239      sdrv*.o      (+RO)
240      *.o          (sys_timer_handle)
241      *.o          (ido_algo_activity_result)
242      *ido_alg_m33_V1_0_42*lib (+RO)
243      *gsensor*.o          (+RO)
244      app_sensor_thread.o (+RO)
245    }
246
247    RW_PSRAM_RET2 +0
```

D:\文档\WeChat Files\wxid_i3h4k76orgeu22\FileStorage\File\2024-06\link_flash_523.sct 2024/6/20 14:57:

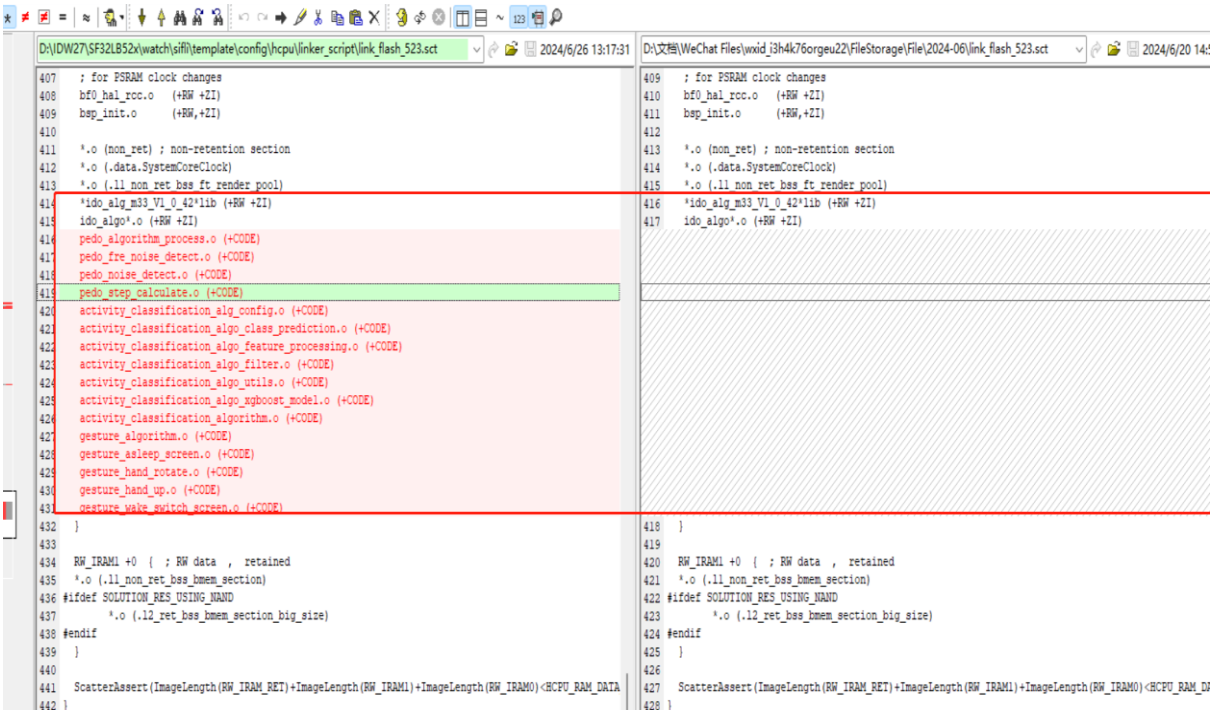
```
234        app_mainmenu_layout_cellular.o (+CODE)
235        lvsf_trans_anim.o (+CODE)
236  #endif
237  #endif
238      ido_algo*.o (+RO)
239      sdrv*.o      (+RO)
240        *.o          (sys_timer_handle)
241        *.o          (ido_algo_activity_result)
242      *ido_alg_m33_V1_0_42*lib (+RO)
243        app_sensor_thread.o (+RO)
244        *gsensor*.o          (+RO)
245
246    }
247
248    RW_PSRAM_RET2 +0
```

D:\JDW27\SF32LB52x\watch\sifli\template\config\hcpu\linker_script\link_flash_523.sct 2024/6/26 13:17:31

```
284
285    RW_PSRAM_SAVE_DATA +0{
286      .ANY (+RW +ZI)
287    }
288
289  #ifndef BSP_USING_PSRAM1
290    ScatterAssert((ImageLength(RW_PSRAM_NON_RET) + ImageLength(RW_PSRAM_NAND) + ImageLength(RW_PSRAM
291  #else
292    ScatterAssert((ImageLength(RW_PSRAM_NON_RET) + ImageLength(RW_PSRAM_NAND) + ImageLength(RW_PSRAM
293  #endif
294
295  #endif
296
297    RW_IRAM0 HPSYS_RAM0_BASE UNINIT {  ; ZI data, not retained
298          //The previous 128KB is the retention area
299      audio_server.o (.bss.audio_server_stack)
300      audio_server.o (.bss.bt_downvoice_stack)
301      media_player_server.o (.bss.ffmpeg_thread_stack)
302        *.o (.ll_non_ret_bss_ulog_be_ram_buf)
303  #if defined (SOLUTION_RES_USING_NAND)
304        drv_spi_nand.o(.l2_ret_bss_nand_buf)
305        nand.o(.bss.dhara_meta_cache)
306  #endif
307        bts2_app_av_snk.o(.bss.deplayback_thread_stack)
308      *.o (STACK)   ; ISR stack
309      *.o (.ll_non_ret_bss_drv_lcd_stack)
310      *.o (.bss.sensor_thread_stack)
311      *.o (.bss.idle_thread_stack)
312      *.o (.ll_non_ret_bss_app_bg_thread_stack)
313      *.o (.ll_non_ret_bss_timer_thread_stack)
314      *.o (.ll_non_ret_bss_touch_thread_stack)
315      *.o (.ll_non_ret_bss_watch_thread_stack)
316      *.o (.ll_non_ret_bss_frambuf)
317      *.o (.ll_non_ret_bss_app_sram_non_ret_cache)
318      *.o (.l2_non_ret_bss_ft_render_pool)
319
320  #ifndef BSP_USING_PSRAM
321      *.o (.l2_non_ret_data_*)
```

D:\文档\WeChat Files\wxid_i3h4k76orgeu22\FileStorage\File\2024-06\link_flash_523.sct 2024/6/20 14:57:

```
285
286    RW_PSRAM_SAVE_DATA +0{
287      .ANY (+RW +ZI)
288    }
289
290  #ifndef BSP_USING_PSRAM1
291    ScatterAssert((ImageLength(RW_PSRAM_NON_RET) + ImageLength(RW_PSRAM_NAND) + ImageLength(RW_PSRAI
292  #else
293    ScatterAssert((ImageLength(RW_PSRAM_NON_RET) + ImageLength(RW_PSRAM_NAND) + ImageLength(RW_PSRAI
294  #endif
295
296  #endif
297
298    RW_IRAM0 HPSYS_RAM0_BASE UNINIT {  ; ZI data, not retained
299          //The previous 128KB is the retention area
300      audio_server.o (.bss.audio_server_stack)
301      audio_server.o (.bss.bt_downvoice_stack)
302      media_player_server.o (.bss.ffmpeg_thread_stack)
303        *.o (.ll_non_ret_bss_ulog_be_ram_buf)
304  #if defined (SOLUTION_RES_USING_NAND)
305        drv_spi_nand.o(.l2_ret_bss_nand_buf)
306        nand.o(.bss.dhara_meta_cache)
307  #endif
308        bts2_app_av_snk.o(.bss.deplayback_thread_stack)
309      *.o (STACK)   ; ISR stack
310      *.o (.ll_non_ret_bss_drv_lcd_stack)
311      *.o (.bss.sensor_thread_stack)
312      *.o (.bss.idle_thread_stack)
313        *.o (.bss.ido_sensor_stack)
314      *.o (.ll_non_ret_bss_app_bg_thread_stack)
315      *.o (.ll_non_ret_bss_timer_thread_stack)
316      *.o (.ll_non_ret_bss_touch_thread_stack)
317      *.o (.ll_non_ret_bss_watch_thread_stack)
318      *.o (.ll_non_ret_bss_frambuf)
319      *.o (.ll_non_ret_bss_app_sram_non_ret_cache)
320      *.o (.l2_non_ret_bss_ft_render_pool)
321
322  #ifndef BSP_USING_PSRAM
323      *.o (.l2_non_ret_data_*)
```



D:\IDW27\SF32LB52x\watch\sifli\template\config\hcpu\linker_script\link_flash_523.sct 2024/6/26 13:17:31

```
    ; for PSRAM clock changes
    bf0_hal_rcc.o    (+RW +ZI)
    bsp_init.o       (+RW,+ZI)

    *.o (non_ret) ; non-retention section
    *.o (.data.SystemCoreClock)
    *.o (.ll_non_ret_bss_ft_render_pool)
    *ido_alg_m33_V1_0_42*lib (+RW +ZI)
    ido_algo*.o (+RW +ZI)
    pedo_algorithm_process.o (+CODE)
    pedo_fre_noise_detect.o (+CODE)
    pedo_noise_detect.o (+CODE)
    pedo_step_calculate.o (+CODE)
    activity_classification_alg_config.o (+CODE)
    activity_classification_algo_class_prediction.o (+CODE)
    activity_classification_algo_feature_processing.o (+CODE)
    activity_classification_algo_filter.o (+CODE)
    activity_classification_algo_utils.o (+CODE)
    activity_classification_algo_xgboost_model.o (+CODE)
    activity_classification_algorithm.o (+CODE)
    gesture_algorithm.o (+CODE)
    gesture_asleep_screen.o (+CODE)
    gesture_hand_rotate.o (+CODE)
    gesture_hand_up.o (+CODE)
    gesture_wake_switch_screen.o (+CODE)
  }

  RW_IRAM1 +0  {  ; RW data  ,  retained
   *.o (.ll_non_ret_bss_bmem_section)
#ifdef SOLUTION_RES_USING_NAND
        *.o (.l2_ret_bss_bmem_section_big_size)
#endif
  }

  ScatterAssert(ImageLength(RW_IRAM_RET)+ImageLength(RW_IRAM1)+ImageLength(RW_IRAM0)<HCPU_RAM_DATA
}
```

D:\文档\WeChat Files\wxid_i3h4k76orgeu22\FileStorage\File\2024-06\link_flash_523.sct 2024/6/20 14:

```
    ; for PSRAM clock changes
    bf0_hal_rcc.o    (+RW +ZI)
    bsp_init.o       (+RW,+ZI)

    *.o (non_ret) ; non-retention section
    *.o (.data.SystemCoreClock)
    *.o (.ll_non_ret_bss_ft_render_pool)
    *ido_alg_m33_V1_0_42*lib (+RW +ZI)
    ido_algo*.o (+RW +ZI)
  }

  RW_IRAM1 +0  {  ; RW data  ,  retained
   *.o (.ll_non_ret_bss_bmem_section)
#ifdef SOLUTION_RES_USING_NAND
        *.o (.l2_ret_bss_bmem_section_big_size)
#endif
  }

  ScatterAssert(ImageLength(RW_IRAM_RET)+ImageLength(RW_IRAM1)+ImageLength(RW_IRAM0)<HCPU_RAM_DA
}
```