



S32K3 常见问题检查列表

Check List

版本 V1.3

2026年1月4日

目录

1.	介绍	12
1.1	目的	12
1.2	免责声明	13
2.	硬件设计部分	13
2.1	必须遵循数据手册和硬件设计指南	13
2.2	必须遵循参考手册	14
2.3	必须参考应用笔记	15
2.4	必须参考 EVB	15
2.5	高速通讯引脚建议选择 Fast 或 Medimu 类型引脚	16
2.6	高驱动能力引脚建议选择 Fast 或 Medimu 类型引脚	16
2.7	注意 eMIOS 模块不同类型通道的选择	16
2.8	注意只能做输入的引脚	16
2.9	不允许将高压信号通过串电阻接入 IO 引脚	16
2.10	ECC 预防电路设计	17
2.11	ADC 硬件设计注意事项	17
2.11.1	P/S/X 三种通道类型的选择	17
2.11.2	不允许将高压信号通过串电阻接入 ADC 引脚	18
2.11.3	输入阻抗过大会导致采样时间长和通道间干扰	18
2.11.4	ADC 参考电压低不能通过磁珠接地	18
2.11.5	ADC 参考电压不能超过 VDD_HV_A	18
2.11.6	需遵守 K3 数据手册的要求	19
2.11.7	需遵守 K3 硬件设计指南的要求	19
2.11.8	需参考硬件应用笔记的要求	19
2.11.9	ADC 引脚上的高压脉冲可能损坏芯片	19
2.12	K3X8+FS26 唤醒注意事项	19
2.13	遵守 EMC 相关要求	20

2.13.1	推荐使用 6 层板	20
2.13.2	产品金属外壳会提升 EMC 性能.....	21
2.13.3	增加产品外壳厚度可以提高 MCU 抗干扰能力	21
2.13.4	增加稳压二极管解决 EMC 问题.....	21
2.13.5	EXTAL 附件的 V15/V25 滤波电容建议放在背面	21
2.13.6	<u>有源和无源晶体具体设计细节</u>	
2.14	使用 K3x4 时推荐使用 SPI 模块自身的片选信号	23
3.	软件开发部分	24
3.1	电源和低功耗（休眠唤醒）部分	24
3.1.1	Last Mile 的配置	24
3.1.2	Pad keeping	25
3.1.3	建议使能低电压检测并修复清标志位 Bug	26
3.1.4	进入休眠过程需要关闭外设并关闭中断	26
3.1.5	建议配置唤醒引脚滤波使能	27
3.1.6	快速唤醒后擦写 FLASH 失败问题	27
3.1.7	需参考 NXP 开发的低功耗示例工程	28
3.1.8	不要旁通 SIRC 的唤醒后调校值加载	28
3.2	时钟	29
3.2.1	需严格遵守参考手册列出的几种时钟配置	29
3.2.2	使用的外设门控时钟要配置为使能	32
3.2.3	快速外部晶振 FXOSC 的自动增益设置	33
3.2.4	RTC 同时用作计时和唤醒的注意事项	34
3.2.5	IAR 编译器设置 FLASH 延时等待问题	34
3.2.6	RTD5.0.0 中 K358 EIM1/2 时钟无法打开问题	35
3.2.7	有源晶振使用注意事项	35
3.3	M7 内核	35
3.3.1	MPU 设置	35

3.3.2	ARM 推测性访问 (ERR011573/ERR052460) 的问题	37
3.3.3	Hardfault 问题排查	38
3.3.4	浮点运算异常问题	38
3.4	存储器	39
3.4.1	需要具备 ECC 恢复机制	39
3.4.2	建议使能低电压监测来尽量避免 ECC 的发生	39
3.4.3	需要使用 8 字节写的方式对 RAM 进行 ECC 初始化	39
3.4.4	RAM 操作等待和 FLASH 操作等待设置不当导致的问题	40
3.4.5	FLASH 同一个块读同时写导致 Hardfault 问题	40
3.4.6	M7 核和 HSE 核同时对一个 FLASH 块读写问题	41
3.4.7	栈溢出问题	42
3.4.8	Fee 问题	42
3.4.9	RTD3.0.0-4.0.0 虚拟扇区大小错误问题	42
3.4.10	链接文件中 TCM 大小与实际芯片需一致	43
3.4.11	TCM 后门访问问题	43
3.4.12	RTD4.0.0 擦写最后一个扇区问题	43
3.4.13	不主动打开 Prefetch	44
3.4.14	对齐访问问题	44
3.4.15	注意 UTEST 的只写一次特性	44
3.4.16	K328/338/348/358 读写 SRAM_2 出错问题	44
3.4.17	Cache 无效化异常问题	45
3.5	复位相关问题	45
3.5.1	APP 中无法获取复位源问题	45
3.5.2	恢复模式 Recovery mode 相关问题	46
3.6	ADC	46
3.6.1	需遵守 K3 参考手册的要求	46
3.6.2	温度通道采样时间的要求	46

3.6.3	ADC 采样时钟的要求.....	47
3.6.4	建议使能预采样.....	49
3.6.5	分辨率建议设置为 12 位.....	49
3.6.6	K310 和 K311 ADC 使用注意事项.....	49
3.6.7	检查最新版本 RTD 的发布说明中提到的 Bug.....	49
3.6.8	建议参考 NXP 的示例工程.....	50
3.6.9	ADC 校准建议.....	50
3.6.10	ADC 超时时间设置.....	50
3.6.11	ADC 引脚上的高压脉冲可能损坏芯片.....	51
3.6.12	ADC 自检失败问题.....	51
3.6.13	ADC Read Group 偶发失败问题.....	51
3.7	CAN.....	52
3.7.1	使能 Enhance FIFO 后，部分邮箱丢帧问题.....	52
3.7.2	CAN 模块处于 start 状态下反初始化失败.....	52
3.7.3	CAN 帧发送乱序问题.....	53
3.7.4	建议 CAN FD 使能传输延时补偿 TDC.....	53
3.7.5	CAN 采样点不满足要求问题.....	53
3.7.6	时钟配置错误导致的问题.....	53
3.8	SPI.....	54
3.8.1	使用 GPIO 做片选信号时 CLK 信号异常脉冲问题.....	54
3.8.2	配置参数需合理.....	54
3.8.3	SPI 同步传输时间长问题.....	54
3.8.4	时钟配置错误导致的问题.....	54
3.9	I2C.....	54
3.9.1	采样保持时间无法调整问题.....	54
3.9.2	时钟配置错误导致的问题.....	55
3.9.3	I2C 从模式在快速模式下的问题.....	55

3.9.4	I2C 上拉电阻设置建议	55
3.10	以太网	56
3.10.1	以太网初始化前外部时钟需稳定	56
3.10.2	千兆以太网使用注意事项	56
3.11	eMIOS	56
3.11.1	PWM 从 100% 占空比切换到 0% 占空比偶发失败	56
3.12	端口引脚设置	56
3.12.1	建议将 JTAG 信号配置安全模式	56
3.12.2	建议将复位信号配置成滤波使能	57
3.12.3	合理设置未使用引脚	58
3.12.4	PTE13 引脚配置	58
3.13	中断相关	58
3.13.1	中断服务程序执行时间长问题	59
3.13.2	尽量避免长时间关闭全局中断	59
3.13.3	中断向量表需要 128 字节对齐	59
3.13.4	中断优先级设置要合理	59
3.14	RTD 相关	59
3.14.1	推荐使用新版本 RTD	59
3.14.2	参考 RTD 示例工程定位问题	60
3.14.3	RTD 中 EB 配置示例工程移植到 S32DS	60
3.14.4	RTD 中 EB 配置示例工程移植到 K312 等其它型号的 S32DS 工程	61
3.14.5	RTD 中 S32DS CT 配置示例工程移植到 K312	61
3.14.6	使用新版本 RTD 验证复现问题	61
3.14.7	根据最新版本 RTD 的发布说明排查使用 RTD 版本的 Bug	62
3.14.8	链接文件问题	62
3.14.9	RTD1.0.0 和 RTD2.0.0 启动代码耗时长问题	63
3.15	编译器相关	63

3.15.1	优化等级设置不合理会出现问题	63
3.15.2	操作系统的优化等级需合理设置	66
3.15.3	IAR 编译器.....	66
3.16	调试器相关	66
3.16.1	当用某种调试器无法连接板卡时，尝试其它调试器	66
3.16.2	尝试不带调试器后板卡运行是否正常	66
3.16.3	勾选调试器异常捕获选项造成的异常	66
3.16.4	调试器对 RAM 区进行 ECC 初始化.....	67
3.16.5	通过 Attach 定位问题.....	67
3.17	EMC.....	68
3.17.1	请遵循硬件设计指南中关于 EMC 的建议	68
3.17.2	使用 PLLFM，可解决 EMC 1.6GHz 左右不通过问题	68
3.17.3	使能 Reset 引脚的滤波功能.....	68
3.17.4	通过写一条 DCF 禁用 Reset 引脚的复位功能	69
3.17.5	禁止 FXOSC 的自动增益.....	69
3.17.6	解决 ESD 不通过及 PLL 失锁的常用措施.....	69
3.17.7	非 MCU 原因引起 EMC 不通过情况	70
3.18	需检查勘误表 ERRATA.....	70
3.19	性能提升	71
3.19.1	使能 CACHE 会显著提升代码执行速度	71
3.19.2	D-CACHE 使能注意事项.....	71
3.19.3	将栈放入 DTCM.....	72
3.19.4	使能 FLASH Prefetch	73
3.19.5	使用 TCM.....	73
3.19.6	使用编译器优化	73
3.19.7	使用 GHS 或 IAR 编译器.....	74
3.19.8	Fee 性能提升.....	74

3.19.9	减小启动时间	74
3.20	其它	75
3.20.1	初始化时间长问题	75
3.20.2	合理设置超时时间	75
3.20.3	通过 DSB 和 ISB 解决代码非顺序执行问题	76
3.20.4	OS 使用注意事项.....	76
3.20.5	建议每个项目都配备 EVB	76
3.20.6	建议使用新版本手册	77
3.20.7	焊接问题	77
3.20.8	测量 MCU 引脚输出电平是否正常	77
3.20.9	看门狗模块调试	77
3.20.10	逻辑分析仪测试造成的误导	77
3.20.11	周期任务和中断都操作某个外设的问题	78
3.20.12	Bootloader 跳转 APP 失败问题	78
3.20.13	通过 NXP 社区获取示例工程和问题解决方案	78
3.20.14	看门狗的影响	79
3.20.15	EB 软件由于多个 link 文件导致生成配置文件问题	79
4.	信息安全(HSE)部分.....	80
4.1	建议使用最新版本 HSE 固件并升级 sBAF	80
4.2	HSE 固件安装注意事项	80
4.3	HSE 固件安装后，S32DS 烧录程序时需选择对应的 FLASH 算法文件	81
4.4	建议参考 NXP 中国应用工程师团队开发的 HSE 库 2.0	81
4.5	未安装 HSE 固件情况下，时钟初始化前必须检查 WFI 标志	83
4.6	未安装 HSE 固件情况下，FLASH 擦写前必须检查 WFI 标志	83
4.7	安装 HSE 固件情下，时钟初始化前必须检查 HSE_STATUS_INIT_OK.....	84
4.8	安装 HSE 固件情下，FLASH 擦写前必须检查 GPR3 寄存器	85
4.9	功能/破坏性复位后 HSE_STATUS_INIT_OK 置位时间长问题	85

4.10	随机数服务调用前检查随机数初始化 OK 标志	86
4.11	随机数服务首次调用耗时长问题	86
4.12	ADKP 和生命周期演进注意问题	86
4.13	D-CACHE 打开导致 HSE 服务失败问题	87
4.14	局部变量传递给 HSE 的注意事项	87
4.15	生命周期演进前需格式化 Key 目录	88
4.16	通过寄存器定位问题	88
4.17	使用空闲的 MU Channel	88
4.18	建议申请超级用户权限	88
4.19	生命周期演进后无法 AB 面切换 (OTA) 问题	89
4.20	HSE 固件损坏或被擦除的原因	89
4.21	HSE 信息安全标准认证情况	90
4.22	RTD3.0.0 和 RTD4.0.0 的 IVT 错误问题	90
4.23	安装 HSE 固件后时钟初始化进入 Hardfault 问题	91
4.24	使用上电复位实现 HSE 功能导致的问题	91
4.25	OTA 过程喂狗问题	92
4.26	0.2.40 及之前版本存在极小概率 HSE 服务无响应问题	92
4.27	建议 AB SWAP 后尽快进行功能复位	93
4.28	避免通过上电复位实现固件安装、生命周期演进和 AB SWAP	94
4.29	错误设置 IVT 导致芯片可能锁死问题	94
4.30	VKMS 固件 DATA FLASH 占用多问题	94
4.31	K3X4 使用 0.2.55 版本 AB SWAP 固件读密钥信息问题	94
4.32	编译器优化设置不当导致 HSE 服务失败问题	95
4.33	参考应用笔记	95
4.34	建议做一些最小功能板用于 HSE 开发	95
4.35	HSE 服务未成功的常见原因	95
4.36	建议在密钥导入后调用服务检查加密 NVM 的完整性	96

4.37	K311 0.2.55 版本 HSE 固件被擦除后无法安装问题	96
4.38	0.2.55 版本 HSE 固件与 0.2.40 版本的主要区别	96
4.39	建议做故障追踪记录	97
5.	多核部分部分	97
5.1	共享存储器 CACHE 问题	97
5.2	PE 调试器 mac 文件问题导致无法多核调试	97
5.3	多核相关宏定义设置问题	98
5.4	多核对同一个 FLASH 块同时读写问题	98
5.5	共享 RAM shareable 段设置问题	99
5.6	XRDC 使用注意事项	99
5.7	参考应用笔记	99
5.8	示例工程	100
5.8.1	基于 K324 和 K338 的多核示例工程	100
5.8.2	基于 RTD1.0.0 S32DS CT 配置的 K324 工程	100
5.8.3	基于 RTD4.0.0P24 S32DS3.5 CT 配置的 K358 工程	101
5.8.4	基于 RTD2.0.1 S32DS CT 配置的 K324 FreeRTOS 工程	101
5.8.5	基于 RTD3.0.0P07 EB 配置的 K338 GHS 工程	101
5.9	OS 配置不当导致多核启动异常	101
5.10	单 ELF 工程不能完全使用 TCM 的问题	101
6.	功能安全部分	102
6.1	FCCU 相关	102
6.1.1	FCCU 故障状态下外设引脚无输入输出功能问题	102
6.2	BIST 相关	102
6.2.1	Trace 时钟和 CLKOUT 需要设置成 FIRC	102
6.3	故障注入相关	102
6.3.1	故障注入后再进行 HSE 服务调用	102
6.4	SAF 相关	102

6.4.1	检查使用版本的已知 Bug.....	102
6.4.2	将部分 sCheck 检测项放在 shutdown 阶段用以减小启动阶段时间	102
6.4.3	非锁步核不需要进行 RCCUTEST_RCCU_CM7 测试.....	103
6.4.4	SAF1.0.5 之前版本可能出现看门狗 sCheck 失败的问题	103
6.4.5	SAF1.0.4 版本 Dcache 和 Flash EDMA sCheck 失败问题.....	103
6.4.6	SAF1.0.4 版本 AB Swap 固件下 sCHECK FLASH 检测失败问题	103
6.4.7	xRDC 配置错误可能导致 sCheck 失败	103
6.4.8	Gmac 模块进行 sBoot 前需完成初始化.....	103
6.4.9	其它	103
6.5	SAF license 说明	104
6.6	应用笔记	104
6.6.1	S32K3xx_Functional_Safety_Application_Note	104
6.7	示例工程	104
6.7.1	基于 K324 的功能安全和信息安全集成 Demo.....	104
6.7.2	基于 K344 和 K358 的功能安全 Demo	104
附录 A	文档中使用的缩略语	105
附录 B	文档变更说明	106
	增加 3.9.4 节。I2C 上拉电阻设置建议。	107
	修改 3.4.16 节。K328/338/348/358 读写 SRAM_2 出错问题。 <i>这个问题不容易在研发阶段发现，需认真检查该条内容。</i>	107
	修改 4.35 节。加入 HSE 无响应的常见原因。	107



1. 介绍

1.1 目的

本文总结了 S32K3 应用中的常见问题和注意事项。分享了 S32K3 使用的经验和心得体会。希望客户在遇到问题前阅读，提前规避问题，少走弯路。

若客户未能提前阅读，在遇到问题时，也可查看该文档，能快速找到部分问题的解决方案。

强烈建议客户在重要里程碑节点前阅读该文档。例如，A 样、B 样、C 样、DV、量产前都阅读该文档的最新版本，及时发现问题。

1.2 免责声明

以下内容个人总结，可能存在理解有误的地方，仅供参考。

2. 硬件设计部分

2.1 必须遵循数据手册和硬件设计指南

使用最新版本的数据手册和硬件设计指南，因为新版本会修复旧版本的 Bug。

截至 2025 年 8 月 26 日：

1、最新版本数据手册为 2025 年 7 月发布的版本 12，如下：

S32K3XX

S32K3xx Data Sheet

Rev. 12 — 10 July 2025

Product Data Sheet

2、硬件设计指南在 NXP 官网的硬件设计包《S32K3_HW-DesignPackage.zip》中。

最新版本数据手册为 2025 年 3 月发布的版本 E2，如下：

NXP Semiconductors

Hardware Design Guidelines for S32K3xx Microcontrollers

Rev. E2 – March. 2025

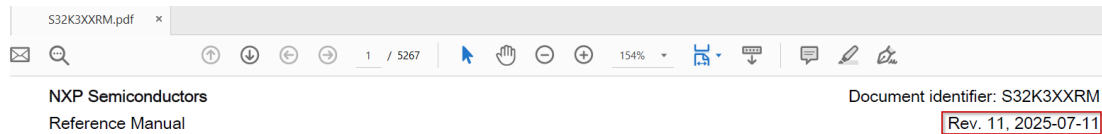
有些客户未遵循这两份文件，出现了问题。

2.2 必须遵循参考手册

使用最新版本的参考手册，因为新版本会修复旧版本的 Bug。

截至 2025 年 8 月 26 日：

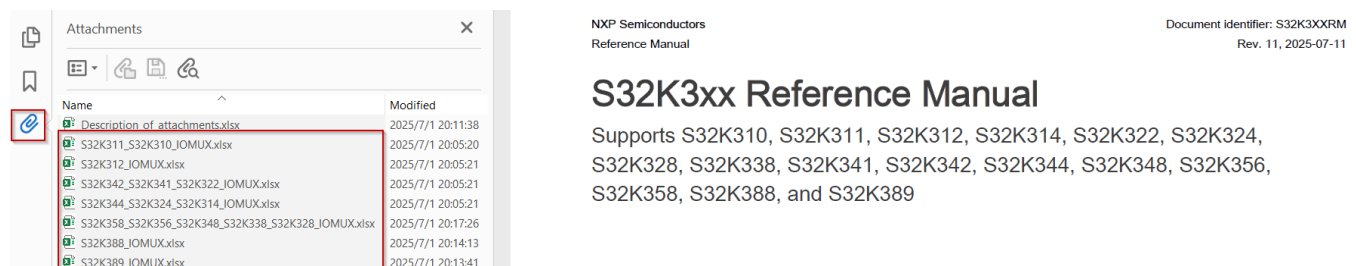
最新版本参考手册为 2025 年 7 月发布的版本 11，如下：



S32K3xx Reference Manual

Supports S32K310, S32K311, S32K312, S32K314, S32K322, S32K324, S32K328, S32K338, S32K341, S32K342, S32K344, S32K348, S32K356, S32K358, S32K388, and S32K389

在参考手册的附件有引脚定义，如下：



包括引脚功能、Pad 类型（标准、中等、Fast）和供电域等信息，以 PTA 为例，如下：

Port	CR	SSS	Function	Module	Description	Function	Pad Type	Pin	Package	I/O Power Domain
PTA0	SIUL_MSCRO	0000_0000	GPIO[0]	SIUL		I/O	GPIO-STANDARD	A13	K344	VDD_HV_A
		0000_0001	LPSP14_PCS2	LPSP14	Peripheral Chip Select 2	O				
		0000_0010	eMIOS_0_CH[17]_Y	eMIOS_0	eMIOS Channel	O				
		0000_0011	LCU0_OUT4	LCU0	LCU Output	O				
		0000_0100	FXIO_D2	FXIO	FlexIO Bi-directional Shift/timer I/O	O				
		0000_0101	eMIOS_1_CH[0]_X	eMIOS_1	eMIOS Channel	O				
		0000_0110	LPSP10_PCS7	LPSP10	Peripheral Chip Select 7	O				
		0000_0111	TRGMUX_OUT3	TRGMUX	Trigger Mux Output	O				
		-	ADC0_S8	ADC0	ADC Standard Input	I				
		-	CMP1_IN0	CMP1	Comparator Input Signal	I				
	SIUL_IMCR528	0000_0001	EIRQ[0]	SIUL	External Interrupt	I				
	SIUL_IMCR577	0000_0010	eMIOS_0_CH[17]_Y	eMIOS_0	External Interrupt	I				
	SIUL_IMCR592	0000_0011	eMIOS_1_CH[0]_X	eMIOS_1	eMIOS Channel	I				
	SIUL_IMCR666	0000_0010	FXIO_D2	FXIO	FlexIO Bi-directional Shift/timer I/O	I				
	SIUL_IMCR740	0000_0001	LPSP10_PCS7	LPSP10	Peripheral Chip Select 7	I				
	SIUL_IMCR769	0000_0001	LPSP14_PCS2	LPSP14	Peripheral Chip Select 2	I				
	SIUL_IMCR872	0000_0001	LPUART0_CTS	LPUART0	Clear To Send (bar)	I				

参考手册的部分章节也建议参考，例如通过电源章节了解电源原理，对于硬件设计有帮助。

2.3 必须参考应用笔记

需要参考与硬件设计相关的应用笔记。

例如：

- 1、FS26 相关《AN13850.pdf》。用于指导 FS26+K3 MCU 的硬件设计。
- 2、FS23 相关《AN14068.pdf》。用于指导 FS23+K3 MCU 的硬件设计。

可联系代理商 FAE 获取 AN13850，官网可以下载 AN14068。

2.4 必须参考 EVB

请参考 EVB 原理图设计。实际板卡与 EVB 设计不同的地方务必慎重。

NXP 应用工程师团队开发了 K312/K344/K358 核心板（172 引脚封装）。板卡内含有电源电路，USB 供电，所有 IO 引脚都已引出，方便调试。特别是 HSE 开发，使用该板卡很方便。

若客户需要，可通过代理商 FAE 获取原理图，Gerber 加工文件和 Bom。

2.5 高速通讯引脚建议选择 Fast 或 Medium 类型引脚

若某个通讯信号用作高速通讯，建议选择该信号对应的引脚类型为 Fast 或 Medium。

因为同一个信号可以选择不同的引脚。在原理图设计中需要注意。不注意的话，可能需要改版设计。

例如，LPSPI4 需要工作在 10M，那么，LPSPI4 的时钟和数据信号就建议选择引脚类型的 Fast 的引脚。如果选择引脚类型为 STANDARD，信号上升时间会较长。

在 K3 参考手册的附件中有每个引脚 PAD 类型的说明（STANDARD、Medium、Fast 等）。

2.6 高驱动能力引脚建议选择 Fast 或 Medium 类型引脚

若某个引脚需要较大的驱动电流，建议选择该信号对应的引脚类型为 Fast 或 Medium。

在 K3 参考手册的附件中有每个引脚 PAD 类型的说明（STANDARD、Medium、Fast 等）。

在 K3 数据手册有不同类型引脚驱动电流的说明。

2.7 注意 eMIOS 模块不同类型通道的选择

eMIOS 有 4 种类型的通道，某些类型的通道不能实现特定的功能，在硬件设计上要充分考虑不同类型通道的差异。

2.8 注意只能做输入的引脚

检查引脚功能是否具备。例如，PTA24、PTA25 引脚只能做输入，不能做输出。

2.9 不允许将高压信号通过串电阻接入 IO 引脚

在数据手册有描述，不能将大于 $VDD_HV_A/B + 0.3\text{ V}$ 的电压（包括通过串接电阻）接入到 IO 引脚，如下：

Table 3. Voltage and current operating requirements (continued)

Symbol	Description	Min	Typ	Max	Unit	Condition	Spec Number
VGPI0	Input voltage range at any I/O or analog pin ^{1,7}	-0.3	—	VDD_HV_A/B + 0.3	V	—	—

若违背，可能出现引脚内部电路损坏，或 MCU 工作异常。

2.10 ECC 预防电路设计

在 FLASH 擦除/写入过程出现电压跌落，可能产生 ECC。

建议硬件设计上增加 ECC 预防措施。

例如，采集电池电压，当电池电压低于某个值后，不进行 FLASH 的擦写操作。

增大电源的储能电容，能在掉电时给 MCU 维持供电一段时间，增加软件处理，让掉电时不进行 FLASH 擦写操作。

2.11 ADC 硬件设计注意事项

2.11.1 P/S/X 三种通道类型的选择

ADC 有 3 种类型的通道：

P：高精度通道；

S：标准通道

X：外部通道

推荐优先使用 P 通道，然后 S 通道，最后 X 通道。

Table 338. ADC configuration

Feature	ADC_0	ADC_1	ADC_2
No. of precision channels	8	8	8
No. of standard channels	16	16	16
No. of special internal channels	1 ¹	0	0
No. of external channels	32	32	0

图 1 ADC 的 3 种通道数目

精度通道和标准通道的内部电气参数不同，如下是参数差异：

CP1	Pin capacitance	1.42	—	4.38	pF	Precision channels	—
CP1	Pin capacitance	1.61	—	5.30	pF	Standard channels	—
CP2	Analog Bus Capacitance	0.32	—	4.18	pF	all channels	—
CP2	Analog Bus Capacitance	0.32	—	1.42	pF	Precision channels	—
CP2	Analog Bus Capacitance	0.497	—	4.18	pF	Standard channels	—
RSW1	Channel selection Switch impedance	65.9	—	1410	Ohm	all channels	—
RSW1	Channel selection Switch impedance	65.9	—	712	Ohm	Precision channels	—
RSW1	Channel selection Switch impedance	65.9	—	1410	Ohm	Standard channels	—

图 2 精度通道和标准通道内部参数对比图

2.11.2 不允许将高压信号通过串电阻接入 ADC 引脚

在数据手册有描述，不能将大于 $VDD_HV_A/B + 0.3\text{ V}$ 的电压（包括通过串接电阻）接入到 ADC 引脚，如下：

Table 3. Voltage and current operating requirements (continued)

Symbol	Description	Min	Typ	Max	Unit	Condition	Spec Number
VGPIO	Input voltage range at any I/O or analog pin ^{1,7}	-0.3	—	$VDD_HV_A/B + 0.3$	V	—	—

若违背，可能出现引脚内部电路损坏，或 MCU 工作异常。

2.11.3 输入阻抗过大会导致采样时间长和通道间干扰

输入阻抗大会导致需要长的采样时间。

虽然手册没有规定最大的输入阻抗，

但建议输入阻抗为 5K 左右，不建议输入阻抗超过 10K。

有客户遇到输入阻抗较大时，采样值受其它通道影响的问题。将阻抗改小后，问题消失。

2.11.4 ADC 参考电压低不能通过磁珠接地

在数据手册和硬件设计指南中，都要求 ADC 参考电压低（VREFL）需直接接入大地。

请不要将其通过磁珠接入大地。

在实际案例中，通过磁珠接地，导致多次校准不成功问题，以及 ADC 码值不连续问题。

2.11.5 ADC 参考电压不能超过 VDD_HV_A

典型案例是 VDD_HV_A 由 3.3V 供电，ADC 参考电压接 5V，就会出现问题。

2.11.6 需遵守 K3 数据手册的要求

见数据手册《S32K3xx Data Sheet》10.1~10.2 节。

请使用最新版本手册。目前最新版本是 2025 年 7 月发布的第 12 版。

很多应用问题都可用在数据手册上找到答案，建议遇到问题前/后都认真阅读。

2.11.7 需遵守 K3 硬件设计指南的要求

见硬件设计指南《S32K3xx - Hardware Design Guidelines -- Rev_E2.pdf》2.3，2.8 节。

请使用最新版本手册。目前最新版本是 2025 年 3 月发布的 E2 版。

2.11.8 需参考硬件应用笔记的要求

AN12217: S32K1xx ADC guidelines, spec and configuration

AN4373: Cookbook for SAR ADC Measurements

2.11.9 ADC 引脚上的高压脉冲可能损坏芯片

若由于外部芯片原因，导致一个高压脉冲（例如，20V，300us，具体见数据手册的说明）施加到 ADC 引脚，可能会导致 MCU 损坏，建议增加 TVS 保护。

2.12 K3X8+FS26 唤醒注意事项

请参考应用笔记 AN13850。可从 NXP FAE 或代理商 FAE 获取。

如下是该 AN 中 FS26 与 K396/K358 的连接说明及框图：

11 FS26 connection to S32K396 and S32K358

For the S32K396 and S32K358 NXP MCUs, a V_{CORE}_PGOOD signal is available through the GPIO2 pin. When GPIO2 is configured as V_{CORE}_PGOOD using the GPIO2_V_{CORE}_PGOOD_OTP bit, GPIO2 is connected to the HS switch (connected internally to V_{PRE}).

V_{CORE}_PGOOD signal may be connected to the WKPU signal of the MCU, which requires a 3.3 V level. A resistor divider is needed to adapt the voltage level.

- When V_{CORE} is in the good voltage range, V_{CORE}_PGOOD is high-logic level.
- When V_{CORE} is not in the good voltage range or is disabled, V_{CORE}_PGOOD is low-logic level.

During Standby mode, in case of a reset from the MCU, its RESET pin must be connected to the FS26 RSTB pin, but also to the FS26 WAKE1 pin, which is configured as a low-level wake-up source, to wake up the SBC.

The total capacitance on the RSTB line should not exceed 1 nF.

The MCU EXTWKPU pin must be connected to the FS26 WAKE2 pin, configured as a high-level wake-up source.

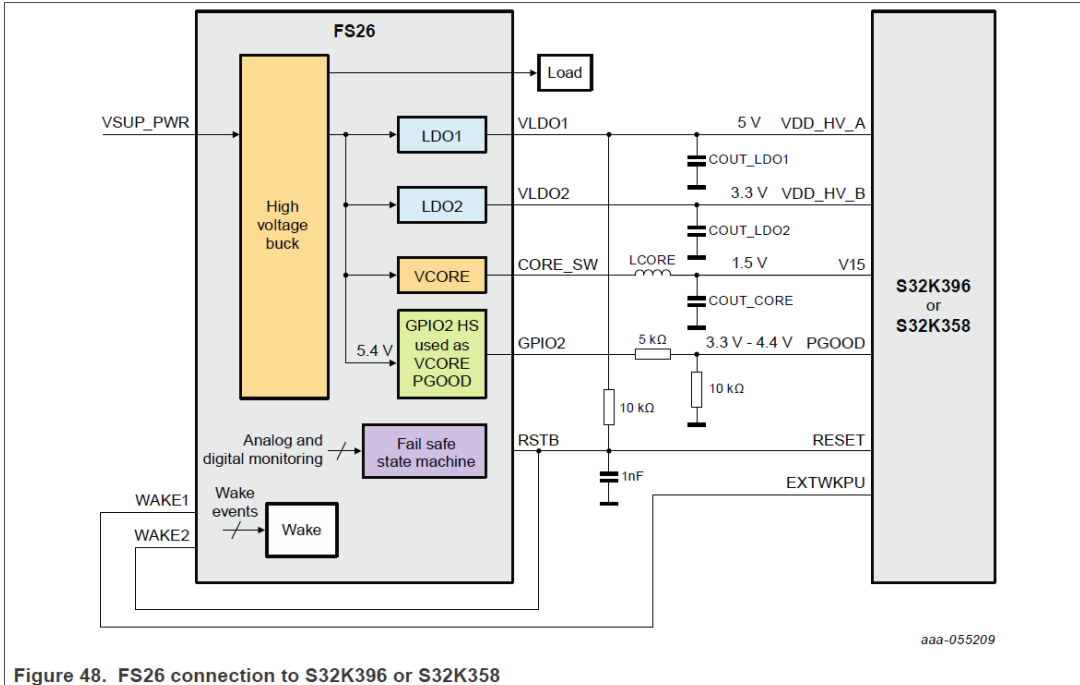


Figure 48. FS26 connection to S32K396 or S32K358

SBC Pgood 连到 MCU Pgood，同时 SBC 和 MCU PGood 都使能；

然后 MCU POR_WDG 必须 Disable(防止产生 Por Wdg reset), MCU External Wake 依然需要连接到 SBC, 同时 MCU Reset 也需要连到 SBC 的 Wake pin (主要是解决如果 SBC 进入 Standby，MCU 没有进入 Standby 特殊工况唤醒异常问题。通过两者的连接，MCU 会由于 Vcore 断电拉低 RSTB，RSTB 再拉 wake 引脚立即唤醒 FS26，解决此工况的唤醒问题)。

也可网上搜索如下文章参考：

“NXP S32K324 与 S32K328 硬件兼容设计优化指南”

2.13 遵守 EMC 相关要求

参考硬件设计指南中 EMC 相关章节。

硬件设计（例如 PCB 布局）遵守硬件设计指南，对解决 EMC 问题（例如 EMC 试验出现的锁相环失锁）有帮助。

2.13.1 推荐使用 6 层板

建议设计成 6 层板，提高 EMC 抗干扰能力。

实际案例，客户使用 4 层板未通过 EMC 测试，修改成 6 层板通过了 EMC 测试。

2.13.2 产品金属外壳会提升 EMC 性能

使用金属壳的 EMC 性能好于塑料壳

由于降本考虑，必须使用塑料壳时，考虑塑料壳喷金属粉的方案。

2.13.3 增加产品外壳厚度可以提高 MCU 抗干扰能力

通过增大外壳厚度（外壳与 MCU 间距离）可解决部分 EMC 问题。

原理是干扰与 MCU 距离变大。

2.13.4 增加稳压二极管解决 EMC 问题

MCU 供电电源若增加稳压二极管，可能会解决部分 EMC 问题。

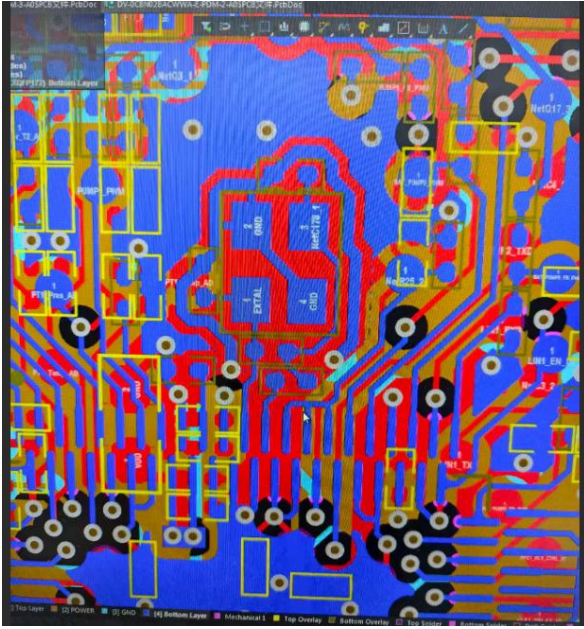
2.13.5 EXTAL 附件的 V15/V25 滤波电容建议放在背面

经过实测，EXTAL 信号和其附近的 V15/V25 的滤波电容若放在不同的面时，会提升抗 ESD 能力。

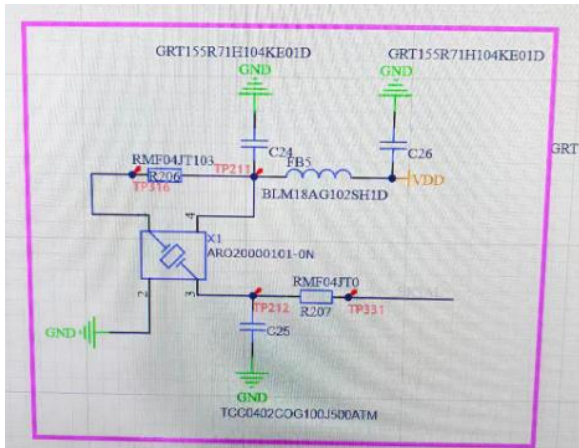
2.13.6

经过测试采用无源晶体最高可以通过 18KV 静电不出错；采用有源晶体可以通过 25KV 静电测试，有源晶体抗静电能力还可以往上

无源晶体需要优先保证 EXTAL 引脚走线最短，如下图；同时将 ALC_D=1



有源晶体原理图如下图



有源晶体 PCB 布局如下图，同样需要保证 EXTAL 走线最短

3. 软件开发部分

3.1 电源和低功耗（休眠唤醒）部分

3.1.1 Last Mile 的配置

配置页面中 Last Mile Regulator Enable 需使能。

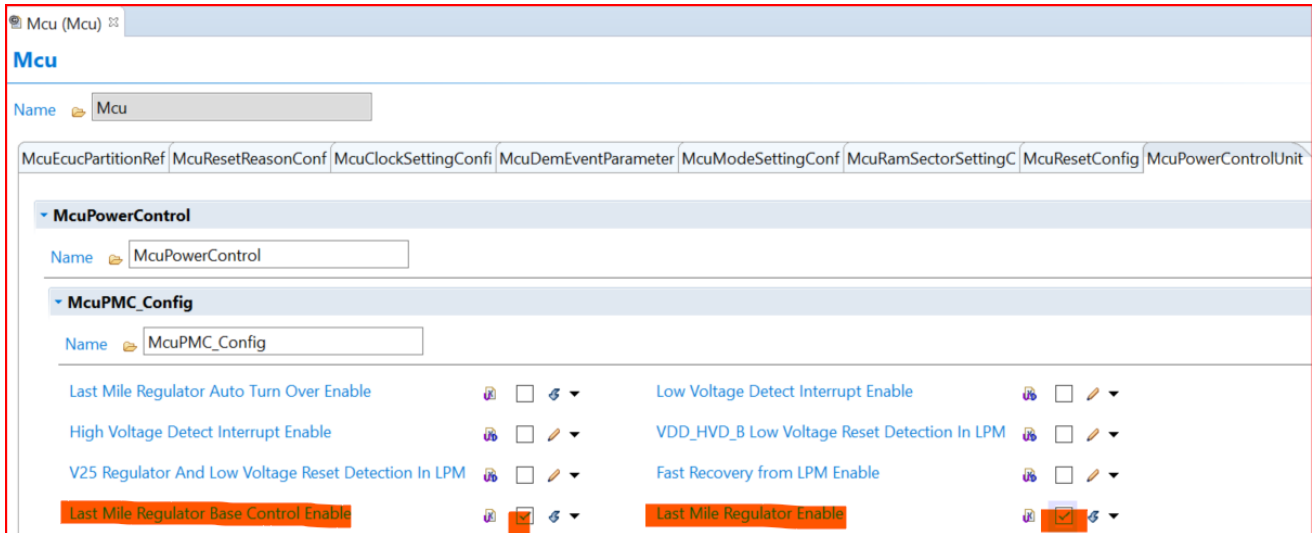
若使用外部三级管提供 1.5V，配置页面还**必需使能** Last Mile Regulator Base Control 选项。

若 1.5V 由外部电源芯片提供，不要勾选 Last Mile Regulator Base Control 选项。

若不使能 Last Mile Regulator 选项（下图右下角），芯片会发热，PLL 开启后程序可能跑飞，影响芯片寿命。

需要实测 V15 是否正常，以及 MCU 发热是否正常。

在 EB 中的配置页面如下：



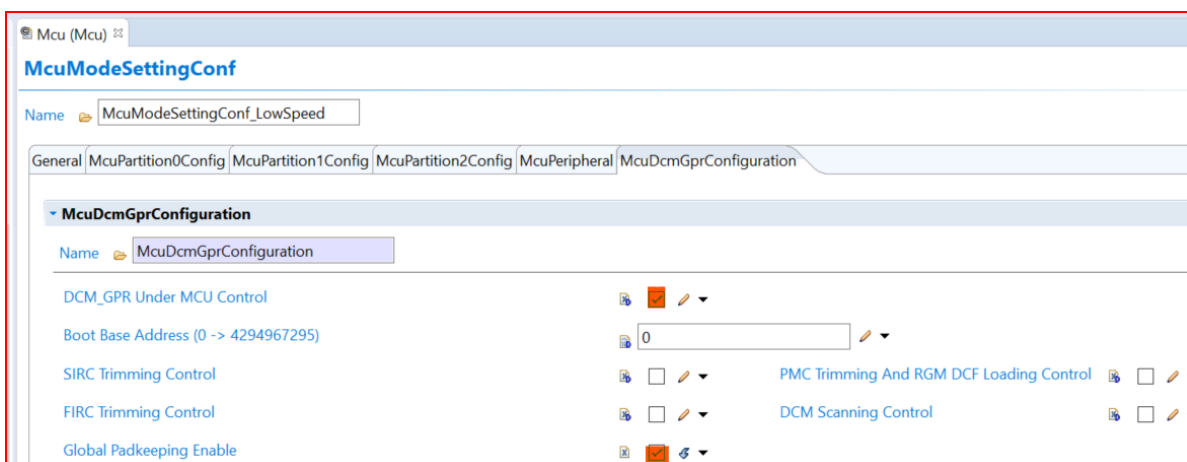
3.1.2 Pad keeping

一、建议使能 Pad Keeping 功能。

- 1、若需要引脚在休眠期间以及唤醒后保持希望的电平，需使能该功能。
- 2、若未使能，外部唤醒信号到来后，可能产生多个唤醒源标志置位。

需要进行 EB 配置或手写代码对寄存器 DCMRWF1 设置。

在 EB 配置中，低功耗模式下需勾选“DCM_GPR Under MCU Control”和“Global Padkeeping Enable”，见下图：



二、建议把需要 Pad Keeping 的引脚都配置勾选“PortPin Pull Keeper”。

因为并非所有的引脚 Pad Keeping 都由“Global Padkeeping Enable”控制。还存在少量引脚必须通过引脚页面勾选“PortPin Pull Keeper”使能 Pad Keeping 功能。这类引脚是“Active pins in Standby mode”的引脚。在 K3 RM 中有对“Active pins in Standby mode”引脚的说明。

因此，最简单的方法是：对所有需要 Pad Keeping 的引脚都在引脚配置中勾选“PortPin Pull Keeper”。同时，全局的 Pad keeping 使能与之前设置方法一致，即仍由 DCMRWF1 设置，或通过 EB 配置。

如果没有在引脚配置中使能“PortPin Pull Keeper”，也可以检查使用 Pad Keeping 的引脚是否不是“Active pins in Standby mode”。若不是，也可以不用设置。

三、建议在 Pad Keeping 禁止前，调用引脚（Port）初始化。

某些 API 函数会调用 Pad Keeping 禁止语句，例如 Mcu_Init()、Power_Ip_Init();若 Pad Keeping 禁止语句调用前，未进行引脚初始化，则可能出现引脚电平短时间跳变。

3.1.3 建议使能低电压检测并修复清标志位 Bug

确保 FLASH 擦写操作时电压稳定。启用低电压检测功能，当检测到低电压，进入中断，不要再启动新的 FLASH 擦写操作，以免由于电压低时擦写 FLASH 导致 ECC 问题。

在各个版本的 RTD 代码中，清低电压标志位存在 BUG。

若使能了低电压检查中断，需要修改 RTD 代码。请参考 RTD5.0.0 的代码实现，并且将

```
IP_PMC->LVSC &= (uint32)(~PMC_LVSC_UV_IRQ_FLAGS_MASK32);
```

修改为：

```
IP_PMC->LVSC |= (uint32)(PMC_LVSC_UV_IRQ_FLAGS_MASK32);
```

因为该标志位是写 1 清零。

该问题代码位置：Power_Ip_PMC.c-> Power_Ip_PMC_VoltageErrorIsr

3.1.4 进入休眠过程需要关闭外设并关闭中断

按照参考手册进入休眠的流程，程序开始执行进入休眠的相关代码时，需要禁用外设(特别是通讯外设，例如 CAN。通过外设反初始化或寄存器操作方式禁止外设)，关闭中断，以免休眠过程被中断打断造成异常。

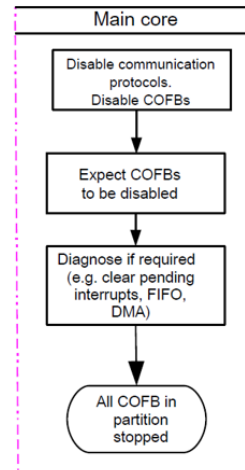
实际案例：休眠过程有中断产生，导致休眠异常。

STANDBY entry sequence

- **Software sequence**
 - **SW1**: Peripheral shutdown process
 - **SW2**: Application core shutdown process.
 - **SW3**: Flash low power handshake and Last Mile Regulator disable process.
 - **SW4**: Main core shutdown process.
- **Hardware sequence**
 - **HW1**: System clock gating.
 - **HW2**: Isolation assertion
 - **HW3**: Asserting reset to powered-down domain
 - **HW4**: Request to PMC for standby entry

SW1: Peripheral shutdown process

- Stop communications and clear all pending interrupts.
- Disable QSPI, EMAC, CAN and all other active communication peripherals.
- Disable interrupts.
- FIRC must be configured as system clock.
- Configure Pad-Keeping.
- STDBY peripherals are configured for their operation.
- Set desired wake-up source(s).



3.1.5 建议配置唤醒引脚滤波使能

建议在 EB/S32DS CT 配置页面中，将唤醒引脚滤波使能勾选，用于滤除干扰。

3.1.6 快速唤醒后擦写 FLASH 失败问题

检查对 FLASH 进行擦除/写操作前，是否是在快速唤醒（Fast wakeup），并且没有经历复位。

如果是的话，则需采取措施，否则可能擦写失败。

解决办法：

- 1、使用正常唤醒，不使用快速唤醒。
- 2、快速唤醒后调用复位 API，或等待看门狗复位。目的是有复位过程，则不会出现问题。
- 3、通过 HSE 服务升级 sBAF。
目前 K314/K324/K344 的修复 bug 的 Full memory 类型 sBAF 版本已发布（AB SWAP 类型修复 bug 的 sBAF 还未发布），K312 的修复 bug 的 Full memory 类型 sBAF 版本已发布（AB SWAP 类型修复 bug 的 sBAF 还不确定发布时间）。这些修复 Bug 的 sBAF 不在官网显示，需联系 NXP FAE 或销售获取。
- 4、在 FLASH 擦除和编写的建立（Setup）阶段关闭中断。关闭中断约 10us。可大大降低擦写失败的概率，有客户测试几十万次未发现出错，但不能保证擦写过程不失败。可联系代理商获取具体实现代码。

不遵循可能出现擦写失败。

3.1.7 需参考 NXP 开发的低功耗示例工程

需参考 NXP 开发示例工程的 EB/S32DS CT 配置和休眠相关代码。

注意细节差异，例如：低功耗模式选择 STANDBY 和 SOC_STANDBY 是不同的。

对于正常唤醒，建议配置中选择 STANDBY 低功耗模式。

对于快速唤醒，建议配置中选择 FAST_STANDBY 低功耗模式。

对示例工程有疑问的地方需要 NXP FAE 澄清。

因为不能保证这些示例工程的配置和代码调用一定是对的。

示例工程组 1:

工程文件夹内包括一篇应用工程师编写的休眠唤醒应用笔记和多个典型休眠唤醒示例工程。

文件夹名称：《S32K3_low_power_management》

工程包括基于 EB 配置的工程和基于 S32DS 配置的工程。基于 RTD2.0.0。

示例工程组 2:

工程包括 K312 工程和 K344 工程。

RTD 版本 4.0.0 和 5.0.0.

工程名称：《S32K3 Low Power Management RTD 4.0.0 & 5.0.0.zip》

工程包括基于 EB 配置的工程和基于 S32DS 配置的工程。

请联系代理商 FAE 获取。

3.1.8 不要旁通 SIRC 的唤醒后调校值加载

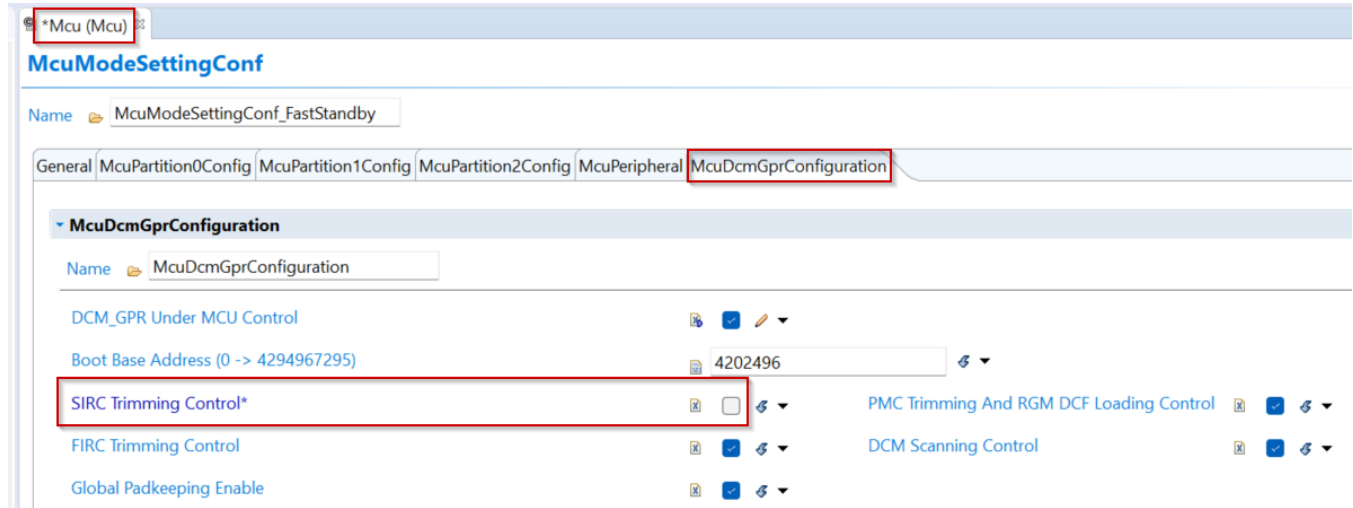
在新版本的参考手册中将 DCMRWF2 寄存器的第 6 位（SIRC_TRIM_BYP_STDBY_EXT）做为预留功能。

在之前版本的参考手册中将 DCMRWF2 寄存器的第 6 位做为唤醒后 SIRC 的调校值是否加载的配置位。

修改的原因是：若将该位设置为旁通，ADC 校准调校（trim）值不会载入 ADC 的 CALBISTREG 寄存器，进而会对 ADC 校准和转换精度产生一些影响。

将该位做保留位，则该位不要设置，保留默认值 0，即不旁通 SIRC 调校值载入。

因此，在休眠模式的配置中，不要勾选如下选项：



注意：不旁通该位设置，会导致 SIRC 在休眠唤醒过程中停止计时 3ms。如果不希望停止计时 3ms，也可以旁通该位（即勾选上图红框），但需要：

- 1、在上电复位或破坏性复位后，将 ADC 的 CALBISTREG[23:16]和 CALBISTREG[13:8]保存。
- 2、在唤醒后，将保存值写入 CALBISTREG[23:16]和 CALBISTREG[13:8]，然后进行 ADC 初始化和校准。
- 3、由于该位是预留位，芯片硬件升版后，可能将该位做其它功能，则该解决方案失效。虽然可能性很小，但有这种风险。

3.2 时钟

3.2.1 需严格遵守参考手册列出的几种时钟配置

检查时钟配置是否**严格遵守**参考手册的几种推荐选项。

若使用非参考手册推荐的时钟选项，可能出现异常。

包括参考手册该选项的**各个时钟部分**设置，不仅仅是系统时钟。

不遵守可能出现各种异常问题。例如，Fee_write 不能写数据到 FLASH。

以 K312 设置为选项 B (OPTION B) 为例, 参见下图红色标注部分:

注意: 使用 K312, 未完全遵循 (见图中标红部分) 时钟选项 Option B, 可能出现异常。

其它型号芯片未遵循手册的几个推荐设置选项同样可能出现异常。

Clocking options	Clock frequencies				
	S32K344, S32K324, S32K314, S32K342, S32K341, and S32K322		S32K310, S32K311, and S32K312	S32K328, S32K338, S32K348, and S32K358	
PLL VCO frequency	960 MHz		960 MHz	960 MHz	
PLLODIV2_CLK (PLLDIG.PLLDV[ODIV2])	480 MHz (0010b)		240 MHz (0100b)	480 MHz (001b)	
FIRC_CLK (HSE_B.CONFIG_REG_GPR[FIRC_DIV_SEL])	48 MHz (11b)				
PLL_PHI1_CLK-related clocks ^{1, 2}					
PLL_PHI1_CLK (PLLDIG.PLLODIV_1[DIV])	240 MHz (0001b)	160 MHz (0010b)	48 MHz (0100b)	240/160 MHz (0000b)	
QSPI_SFCK (MC_CGM.MUX_10_DC_0[DIV])	120 MHz (001b)	80 MHz (001b)	—	120 MHz (001b)	80 MHz (001b)
TRACE_CLK (MC_CGM.MUX_11_DC_0[DIV])	For fast pads		—	For fast pads	
	120 MHz (001b)	80 MHz (001b)	—	120 MHz (001b)	80 MHz (001b)
	For standard-plus pads		—	For standard-plus pads	
	24 MHz (1001b)	16 MHz (1001b)	—	24 MHz (1001b)	16 MHz (1001b)
PLL_PHI0_CLK-related clocks ³					

Table 152. Option B - Reduced Speed mode (CORE_CLK @ 120 MHz) (continued)

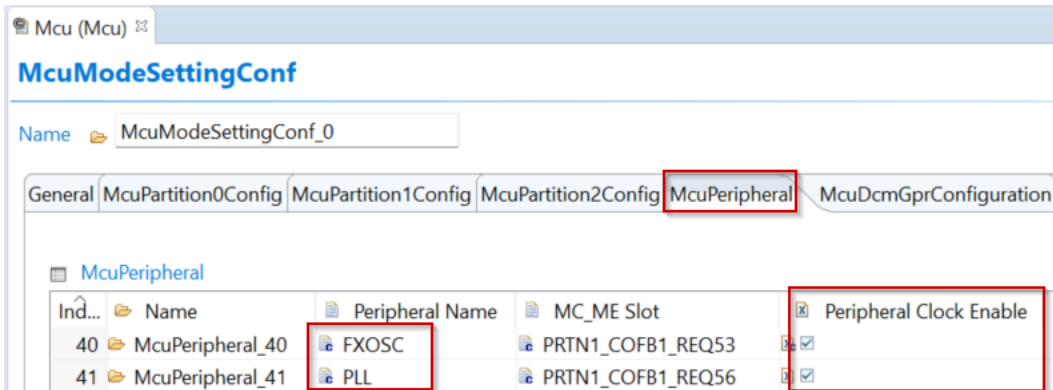
Clocking options	Clock frequencies					
	S32K344, S32K324, S32K314, S32K342, S32K341, and S32K322		S32K310, S32K311, and S32K312		S32K328, S32K338, S32K348, and S32K358	
PLL_PHI0_CLK (PLLDIG.PLLDIV_0[DIV])	120 MHz (011b)		120 MHz (001b)		120 MHz (011b)	
CORE_CLK <ul style="list-style-type: none"> • Application cores • AXBS • SRAM • Flash memory controller port clock • AIPS0 (high-speed peripheral clock) (MC_CGM.MUX_0_DC_0[DIV])	120 MHz (000b)		120 MHz (000b)		120 MHz (000b)	
QSPI_MEM_CLK (MC_CGM.MUX_0_DC_6[DIV])	120 MHz (000b)		—		120 MHz (000b)	
AIPS_PLAT_CLK (medium-speed peripheral clock) (MC_CGM.MUX_0_DC_1[DIV])	60 MHz (001b)		60 MHz (001b)		60 MHz (001b)	
AIPS_SLOW_CLK (slow-speed peripheral clock) (MC_CGM.MUX_0_DC_2[DIV])	30 MHz (011b)		30 MHz (011b)		30 MHz (011b)	
DCM_CLK (MC_CGM.MUX_0_DC_4[DIV])	30 MHz (011b)		30 MHz (011b)		30 MHz (011b)	
HSE_CLK (MC_CGM.MUX_0_DC_3[DIV])	120 MHz (000b)	60 MHz (001b)	120 MHz (000b)	60 MHz (001b)	120 MHz (000b)	60 MHz (001b)
LBIST_CLK (MC_CGM.MUX_0_DC_5[DIV])	30 MHz (011b)		30 MHz (011b)		30 MHz (011b)	

3.2.2 使用的外设门控时钟要配置为使能

外设门控时钟未使能时，对外设初始化会进入 Hardfault。

在 RUN 模式的外设使能配置页面中，需要勾选 FXOSC,PLL， PLLAUX (K3x8)时钟。

在低功耗模式的外设使能配置页面中，不需要勾选 FXOSC,PLL， PLLAUX (K3x8)时钟。



未在 EB/S32DS CT 配置中勾选 FXOSC,PLL， PLLAUX (K3x8)时钟，并且在调用 Mcu_SetMode 函数后调用功能复位，该复位会变成破坏性复位， Standby RAM 数据丢失，复位源是上电复 POR。

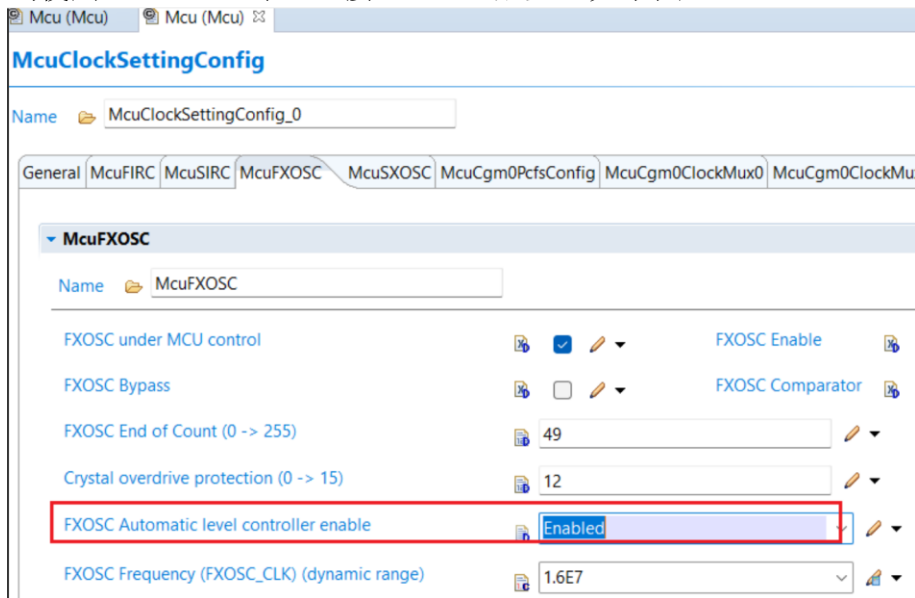
3.2.3 快速外部晶振 FXOSC 的自动增益设置

检查是否禁止快速外部晶振 FXOSC 的自动增益。

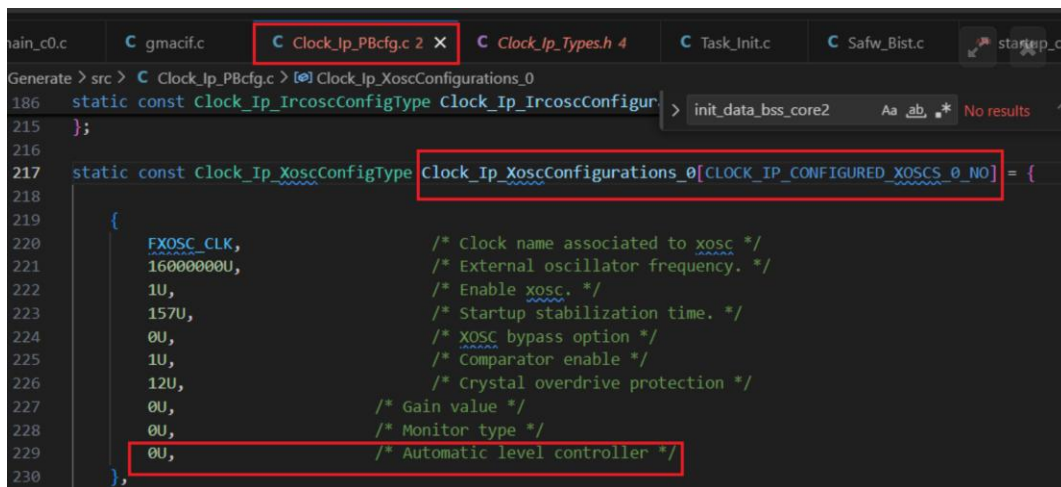
如果 EMC 试验（例如静电试验），出现锁相环失锁导致 MCU 复位，可禁止 FXOSC 的自动增益。

禁止自动增益有 2 种方法：

1、当使用 RTD5.0.0 时，直接通过 EB 配置，见下图：



当使用 RTD5.0.0 之前版本，可直接修改 EB 生成的配置文件，见下图：



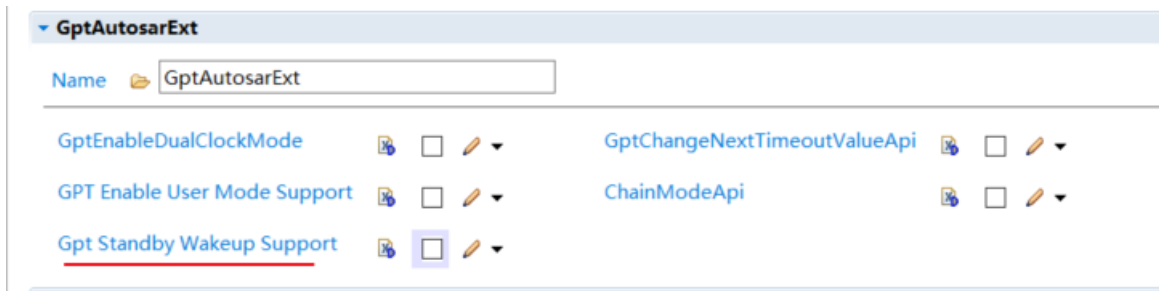
2、参考 K3 参考手册，找到设置该位的寄存器，设置寄存器中的增益值为禁止。

3.2.4 RTC 同时用作计时和唤醒的注意事项

若是，使能 Gpt Standby Wakeup Support（见下图），避免 Gpt_Init clear CNT。

同时，每次休眠前获取当前 RTCCNT 的值，设置 $RTCVAL = \text{current RTCCNT} + \text{Wake Period}$

此时应用上要考虑 CNT 寄存器 $0xFFFFFFFF \rightarrow 0X0$ 的情况。



3.2.5 IAR 编译器设置 FLASH 延时等待问题

若 RTD 使用 4.0.0 之前的版本，并且通过配置页面使能 FLASH 操作等待选项（例如：EB 配置中 MCU 配置页面不勾选 Mcu Disable Flash Config from RAM（RTD2.0.3 及之前版本）或 Mcu Disable Flash Wait States Config（RTD3.0.0 及以后版本）），推荐按照 RTD4.0.0（或 RTD5.0.0）的方法修改 FLASH 控制器的 CTL 寄存器。

即修改 Clock_Ip_Specific.c 中的 Clock_Ip_CodeInRamSetFlashWaitStates 函数。

在 RTD4.0.0 的发布说明（Release note）有相关说明。

有客户使用 IAR 编译器，遇到了该问题，并通过 RTD4.0.0 的措施解决了问题。

因此，若使用 IAR 编译器，该措施需添加。

目前暂无使用 S32DS 的 GCC 编译器或 GreenHills 的客户反映此问题。

建议使用 S32DS 的 GCC 编译器或 GreenHills 的客户也在合适的机会按照 RTD4.0.0 添加该措施。因为不能保证使用这两款编译器就一定不出问题。

3.2.6 RTD5.0.0 中 K358 EIM1/2 时钟无法打开问题

这是 RTD5.0.0 的 Bug，预计在 RTD6.0.0 中修复。K3X8 都存在此问题。

需要按照参考手册修改 Power_Ip_PBcfg.c 中 Partition2 COFB2 来使能 EIM1/2 的时钟。

如下是解决问题的代码：

```
IP_MC_ME->PRTN2_COFB2_CLKEN |= MC_ME_PRTN2_COFB2_CLKEN_REQ68(1);
IP_MC_ME->PRTN2_COFB2_CLKEN |= MC_ME_PRTN2_COFB2_CLKEN_REQ69(1);

/* Prepare partition update */

IP_MC_ME->PRTN2_PUPD |= MC_ME_PRTN2_PUPD_PCUD_MASK;

/* Trigger partition update by writing CTL_KEYS */

IP_MC_ME->CTL_KEY = 0x5AF0;

IP_MC_ME->CTL_KEY = 0xA50F;
```

3.2.7 有源晶振使用注意事项

- 1、根据 K3 参考手册描述，FXOSC 使能前，有源晶振不能通过 EXTAL 引脚输入，EXTAL 引脚应为低电平。例如，可通过 MCU 的 GPIO 控制有源晶振的输出。
- 2、需要判断 Clock Mux 3 Select Status Register (MUX_3_CSS)是否已成功切换到时钟。若未成功，需要再次操作 Clock Mux 3 Select Control Register (MUX_3_CSC)进行时钟切换，直至 MUX_3_CSS 指示时钟切换完成，再进行后面时钟配置。
因为使能 FXOSC 后，才能使能有源晶振的输出，有源晶振输出是否稳定需要判断，可以使用 MUX_3_CSS。若有源晶振输出未稳定，进行时钟切换可能会失败。

3.3 M7 内核

3.3.1 MPU 设置

- 1、为解决随机性访问带来的问题。必须使能 MPU。将背景 MPU 区域 0 设置为不可访问。

MPU 表项索引	Start Address	Size	存储属性	访问权限 特权/非特权	执行权限	Comments
1	0x0000_0000	4G	Strongly-ordered (TEX.C.B.S=000.0.0.X)	No access/No access AP=0b000	XN=1	disable all

在 S32DS 工程属性中加入使能 MPU 宏定义，则会进行 RTD 示例的 MPU 设置，该设置已实现了该条建议。

```
system.c
456 /*****
457  /* Init MPU table for memory layout*/
458  /* Cover all memory on device as background set all memory as strong-order and no access*/
459  rbar[0]=0x00000000UL;
460  rasr[0]=0x1004003FUL;
```

2、MPU 基地址必须对齐到区域大小的整数倍。

基地址即起始地址。

例如，设置基地址为 0x2040_8000（非 64K 对齐），区域大小为 64K，则会出现异常。

3、MPU 各个区域设置读/写/访问权限和大小设置是否合适。

例如：1、未设置 UTEST 区的读写访问权限，此时读写 UTEST 区就会 Hardfault。

2、将代码放入 RAM 运行，但 RAM 未设置可执行属性。

区域大小为 2 的 N 次幂，并且至少为 32 字节。

大小设置错误也会导致异常问题。

4、MPU 重叠区域以高区域（Region）号设置为准。

5、使用子区域（sub region）可实现灵活的区域属性设置。

设置寄存器中子区域字段（SubRegion）可实现期望的大小（例如，设置某个区域的大小是 6M，而寄存器中 SIZE 部分只能设置成 8M，则通过 SubRegion，设置 2M 空间不遵守本项设置实现，具体可参见启动代码 system.c 中 MPU 设置）。

6、若 MPU 中将一段 SRAM 属性设置为 CACHEABLE，同时也设置了 SHAREABLE 属性，则 CACHEABLE 实际未启用，需去掉 SHAREABLE 属性实现 CACHEABLE 特性。

7、建议在 MPU 控制寄存器中使能 HFNMIENA 位。

如果不使能，当 MCU 进入 Hardfault 或 NMI 中断时，会使能默认的 MPU 配置，不会使用工程设置的 MPU，进而可能导致数据不一致问题（如果在 Hardfault 或 NMI 中断不去写项目中 MPU 配置为 NON-CACHEABLE 的 RAM 区，则无此不一致问题）。例如，默认的 MPU 配置中 RAM 是 CACHEABLE 的，而项目中可能将某段 RAM 设置位 NON-CACHEABLE。

在 RTD4.0.0 的 RTD 启动代码中已经实现了该建议，如下：

```
system.c
600
601 /* Enable MPU, enables the MPU during the HardFault handler */
602 S32_MPU->CTRL |= (S32_MPU_CTRL_ENABLE_MASK | S32_MPU_CTRL_HFNMIENA_MASK);
```

在 RTD2.0.3 及之前版本，未实现该建议，如下：

```
578 /* Enable MPU */
579 S32_MPU->CTRL |= S32_MPU_CTRL_ENABLE_MASK;
```

8、当需要从一种 MPU 配置切换到另一种 MPU 配置时（例如 2 个任务使用不同的 MPU 配置），切换配置前需要关闭中，切换后再使能中断。因为 MPU 切换过程会先禁止 MPU（即：使用 ARM 默认的 MPU 设置），此时若进入中断，可能带来数据不一致问题。

9、MPU 多次设置的注意事项。

- (1) 设置 MPU 前建议关闭全局中断，设置完成后再打开全局中。
- (2) 通常，在启动代码和初始化代码中只选择一处进行设置即可。若两处都进行了设置，后设置处生效。建议只在一处设置，或两者设置相同。
- (3) AUTOSAR OS 也会设置 MPU（需确认 OS 设置的 MPU 正确）。FreeRTOS 不会设置 MPU。
- (4) MPU 依据后设置为准（生效）。

10、 检查是否使用 Vector 配置工具进行 MPU 设置，若是，检查工具产生的配置是否正确写入 ARM 的寄存器（RNR, RBAR 和 RASR）。

通过查看/调试代码是否正确写入各个区域的设置确定。

有实际案例，使用 Vector 配置工具进行 MPU 设置，16 个区域只有一个区域设置成功。

3.3.2 ARM 推测性访问（ERR011573/ERR052460）的问题

ARM 具备推测性访问的特性，会提升系统性能。

但需要做一些处理，防止 ARM 推测性访问带来的问题。

方法一、按照上一节内容设置 MPU

需考虑 ERRATA（ERR011573/ERR052460），即需要设置 MPU，让区域 0 为所有空间禁止访问，与 RTD 系统初始化函数中 MPU 初始化代码（system.c->SystemInit()）的默认区域 0 设置相同即可。防止预留地址 0x1B1000000~0x1B101FFF 被预取导致内核卡死。

典型问题案例：客户未将区域 0 设置为禁止访问，运行一段时间后内核挂死。

方法二、用 XRDC 设置内存和外设的访问权限和属性。

方法三、ERR052460 的推荐方法，即在启动代码的 SystemInit()中加入如下代码：

```
void utest0X1B_control(void)
{
#define S32_UTEST0X1B10_CTRL 0X402AC0F0 // MPU control register
volatile uint32_t *reg_utest1b10_ctl = (uint32_t *)S32_UTEST0X1B10_CTRL;
*reg_utest1b10_ctl = 0x1CB0499D;
*reg_utest1b10_ctl = 0xB9920D38;
```

```
}
```

可以将该 C 代码用汇编代码实现，然后，将其加入到启动代码 **Reset_Handler** 中。加入的位置越靠前越好。因为从理论上说，加入越早，越能避免随机性访问带来的该 ERRATA 问题。

3.3.3 Hardfault 问题排查

如下寄存器便于定位问题：

1、配置故障状态寄存器（CFSR）

该寄存器用于判断故障原因，CFSR（地址为 0xE00ED28），大小为四字节。

2、内存管理故障地址寄存器 MMFAR（地址为 0xE00ED34）

建议判断 CFSR 是否有 MMFAR 故障，并且故障地址是否有效。

3、总线故障地址寄存器 BFAR（地址为 0xE00ED38）

建议判断 CFSR 是否有 MMFAR 故障，并且故障地址是否有效。

4、程序计数器 PC

发生异常后，程序进入 Hardfault，会将运行异常的指令地址 PC 保存在堆栈中。

根据该 PC 值，可以找到出错的程序地址，对分析问题很有帮助。

首先，找到堆栈指针（MSP/PSP），然后，找到 PC。

将 PC 保存，根据反汇编代码，就可以找到异常地址，锁定问题。

5、链接寄存器 LR

发生异常后，程序进入 Hardfault，会将运行异常的指令地址 LR 保存在堆栈中。

根据该 LR 值，可以找到出错的程序地址，对分析问题很有帮助。

3.3.4 浮点运算异常问题

若出现浮点运算异常问题，请检查是否因为操作系统（OS）对 ARM 浮点相关寄存器进行了设置。有客户遇到过此问题。

3.4 存储器

3.4.1 需要具备 ECC 恢复机制

ECC 是 FLASH 应用的常见问题。已有很多客户遇到 ECC 问题。

若不施加 ECC 恢复机制，板卡将失去功能。

常用的恢复机制：

- 1、将出现 ECC 的 FLASH 扇区擦掉。
- 2、启用 ECC 抑制功能，配置页面勾选该选项。

可参考指导文件《MCU Ecc issue and solution answer standard_S32K3 S32K1 KEA.docx》。

可联系代理商 FAE 获取。

3.4.2 建议使能低电压监测来尽量避免 ECC 的发生

检查是否有 ECC 预防措施。

确保 FLASH 擦写操作时电压稳定。启用低电压检测功能，当检测到低电压，不再进行 FLASH 擦写操作，以免由于电压低时擦写 FLASH 导致 ECC 问题。

3.4.3 需要使用 8 字节写的方式对 RAM 进行 ECC 初始化

检查 RAM（特别注意 Standby RAM）进行 ECC 初始化时是否按照 8 字节写的方式。

例如：`*(uint64 *)0x20400000 = 0;`

不遵守可能 ECC 错误。

在 NXP 做的 BOOTLOADER 示例工程中使用了 4 字节清零的 ECC 初始化方式，

`*(uint32 *)0x20400000 = 0;`

是错误的，不要参照。

3.4.4 RAM 操作等待和 FLASH 操作等待设置不当导致的问题

检查 EB 配置中 RAM 操作等待和 FLASH 操作等待

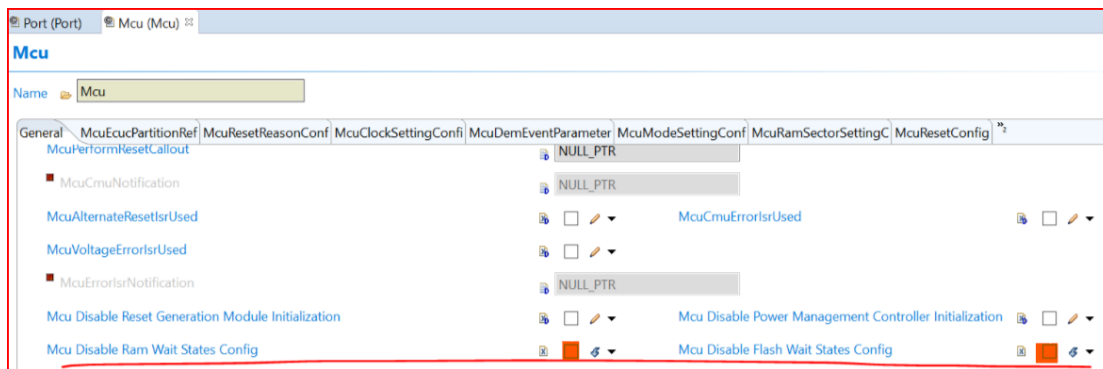
1、对于 K322, K341, K342, K3X4, K3X8, **不要**勾选。

2、对于 K310, K311 和 K312:

RAM 操作等待设置（EB 中 MCU 设置的 Mcu Disable Ram Wait States Config）建议勾选。不勾选性能略有下降，但不会有问题。

FLASH 操作等待设置（EB 中 MCU 设置的 Mcu Disable Flash Config from RAM（RTD2.0.3 及之前版本）或 Mcu Disable Flash Wait States Config（RTD3.0.0 及以后版本））建议不勾选，此时会根据内核时钟设置等待延时。若勾选，会按照寄存器默认值（RWSC=6）设置等待时延，性能略有下降，但不会有问题。

EB 配置见参照图 1。



实际案例：已有几个客户遇到此问题。由于不遵守时只是少量芯片会出现异常，故可能在研发阶段无法发现，问题很严重。建议检查生成的配置文件已按照 EB/CT 配置进行了更新。不遵守可能出现读写数据错误，也可能出现 ECC 错误。

特别是低温下，可能因为禁止了 RAM 等待（勾选了该选项），导致 ECC 问题。

在 K3 参考手册有说明该设置要求的说明。

3.4.5 FLASH 同一个块读同时写导致 Hardfault 问题

检查是否存在对 FLASH 的一个块 BLOCK 同时进行读和写。

例如，代码在 BLOCK0 上执行（读指令，读操作），同时，对 BLOCK0 擦除扇区或编程。

若使用 FLS API，可勾选“Fls Load Access Code On Job Start”，实现将 FLASH 底层驱动拷贝到 RAM 中执行的目的，具体可参见 RTD 代码实现，可解决同一个 BLOCK 的读同时写问题，见图 3。

注意：UTEST 区和 BLOCK0 同属一个块 BLOCK。所以，对 UTEST 区进行写操作时，FLASH 最底层驱动需在 RAM 中执行。但是勾选“Fls Load Access Code On Job Start”后，对 UTEST 区写操作时，RTD 代码并不会将 FLASH 底层驱动代码搬运到 RAM，故需要用 `__attribute__((section(".name")))` 方式或 `#pragma GCC section (text|data|bss|rodata) ".name"` 的方式将 FLASH 底层代码搬运到 RAM。

例如：

```
__attribute__((section("new_section"))) func_to_be_remapped();
```

或

```
#define FLS_START_SEC_RAMCODE
#include "Fls_MemMap.h"
hseSrvResponse_t EraseFlash(uint32_t dstAddr, uint32_t len);
#define FLS_STOP_SEC_RAMCODE
#include "Fls_MemMap.h"
```

具体实现请参考 HSE LIB2.0 或 HSE 固件安装工程对 UTEST 区写数据（例如写 HSE 固件安装特征字）的方法。需要注意如果一个函数内又调用一个函数，那么，只将上层函数放入 RAM 或 ITCM 是不行的，还需将调用的函数放入 RAM/ITCM。保证需放入 RAM/ITCM 的函数及其调用的函数（若调用函数内又调用了函数，那么被调用函数也需放入）全部放入。

不遵循可能出现擦写失败，Hardfault。

3.4.6 M7 核和 HSE 核同时对一个 FLASH 块读写问题

检查未安装 HSE 固件情况下，对 FLASH（包括 UTEST）进行擦除/写操作前是否等待 HSE 核进入 WFI。详见 4.6 节。

若已**安装了**HSE 固件，对 FLASH（包括 UTEST）进行擦除/写操作前判断 HSE 是否占用该块 BLOCK。即通过读取 HSE GPR Register 3（地址 0x4039C028）判断，详见 HSE 参考手册。

注意：BLOCK0 和 UTEST 同属一个 BLOCK。若 HSE 对 UTEST 读操作，那么不仅 M7 核不要对 UTEST 区进行写操作，也不要对 BLOCK0 进行擦写操作。

RTD6.0.0 代码加入了信号量机制，可以不用判断 HSE GPR3 寄存器。

3.4.7 栈溢出问题

局部变量存放在栈中。

若栈分配空间较小，局部变量较多（例如，函数内定义一个 4K 大小的数组），则可能出现栈溢出问题，进而程序运行溢出，甚至跑飞。

建议栈设置合适的大小。

出现异常问题时，考虑是否由于栈溢出导致，可增大栈大小，测试问题是否消失。

在 RTD 代码中，若使能了 CAN FIFO 功能，会在函数内定义一个 2K 大小的数组，需要考虑此工况栈空间是否足够。

3.4.8 Fee 问题

在 RTD4.0.0（RTD4.0.0HF01/02 除外）及之前版本，Fee 存在一些 Bug，另外，Fee 性能也较差。

RTD4.0.0HF01/02、RTD5.0.0 和 RTD6.0.0 修复了 Bug，并提升了性能。

实际案例：RTD3.0.0P01 一个 Cluster 未写完满后，就进行换页。

Fee 性能提升详见 3.19.8 节。

在多核芯片使用单 ELF 工程的情况，只支持一个核操作 Fee。

3.4.9 RTD3.0.0-4.0.0 虚拟扇区大小错误问题

检查 EB 配置生成的 C40_IP_MAX_VIRTUAL_SECTOR 或 FLS_MAX_VIRTUAL_SECTOR 是否正确。RTD3.0.0 之前版本无此问题。

对应 K358，RTD3.0.0 和 RTD4.0.0 生成的该值是 1039，是错误的，应该为 1040。

其它 K3 芯片同理，如下是正确定义，若生成的代码有误，请直接修改即可。

注意：每次新生成配置文件，该定义都会更新，需再次修改，如下：

K312: #define C40_IP_MAX_VIRTUAL_SECTOR (272U)

K3x4: #define C40_IP_MAX_VIRTUAL_SECTOR (528U)

K3x8: #define C40_IP_MAX_VIRTUAL_SECTOR (1040U)

3.4.10 链接文件中 TCM 大小与实际芯片需一致

若使用芯片与示例工程中芯片类型不同，检查是否在示例工程中修改了链接文件，保证链接文件中 ITCM,DTCM,RAM 是否与实际使用芯片相同。

例如，实际使用 K324，使用 K344 的示例工程，需要将链接文件中 ITCM 和 DTCM 的大小修改为实际值，否则无法调试。

3.4.11 TCM 后门访问问题

检查是否存在后门访问 TCM 问题。

每个核都分配一定大小的 ITCM 和 DTCM，若某个核访问其它核的 ITCM 或 DTCM，需要使用后门地址（Back Door，在参考手册附件有后面地址的说明）。

例如，HSE 核访问 M7 核的 DTCM，就需要使用后门地址。

核 0 访问核 1 的 DTCM，需要使用后门地址。

若需要使用 Backdoor 方式访问 ITCM，需要注意：

- 1、ITCM Backdoor 地址初始化不能使用 main 函数 8 字节写的方式，需使用启动代码中汇编代码方式。
- 2、开启 MPU 后，需将 Backdoor 地址加入 16 个区间中。

3.4.12 RTD4.0.0 擦写最后一个扇区问题

在 RTD4.0.0 中，对 FLASH 块（BLOCK）的最后一个扇区（Sector）进行擦写操作可能会失败，这是软件的 Bug，在 RTD5.0.0 中已修复。

若不需要对 FLASH 驱动（Access code）所在 BLCOK 的最后 sector 擦写，则无此问题。

例如，FLASH 驱动（Access code）在 BLCOK0，那么操作 BLOCK1 的最后一个扇区是没问题的，不涉及同一个 BLOCK 读同时写的问题。

若需要对 FLASH 驱动（Access code）所在 BLCOK 的最后 sector 擦写，可能会出现错误，不是一定出错。但存在隐患，需修改软件解决。

若需要修复，可参照 RTD5.0.0 修复。

见 RTD5.0.0 Release note 的 ARTD-117973。

3.4.13 不主动打开 Prefetch

检测是否主动打开了 Prefetch。推荐不要主动打开，可能引起问题。

RTD 代码默认不会打开 Prefetch。

若项目在 CACHE 和 TCM 等常规措施使用的情况下 CPU 负荷仍很高，也可以打开 Prefetch。建议做大量测试确保没问题。

Prefetch 的打开和关闭操作建议在 RAM 或 ITCM 中进行（即：将开/关操作函数分配到 RAM 或 ITCM），否则可能产生问题。

3.4.14 对齐访问问题

检查寄存器读取是否 4 字节对齐。

例如，`a = *(uint8_t*)(0x4039c021)` 会进入硬件错误。

3.4.15 注意 UTEST 的只写一次特性

注意 UTEST 区只能写一次，不能擦除，因此，对 UTEST 区写操作务必小心，确认写入数值是正确的。例如，对 0x1B000050 写入晶振频率时，必须与实际使用晶振频率一致。

3.4.16 K328/338/348/358 读写 SRAM_2 出错问题

在 RTD6.0.0 之前的版本，未设置 SRAM_2 的读写等待，所以，部分芯片会出现读写 SRAM_2 出错问题。

在 RTD6.0.0 的 Release note 中有修复该 Bug 的说明，见 ARTD-189942。

该问题只会在部分芯片出现（特别是低温情况），所以，不容易在研发阶段发现。

若使用 RTD6.0.0 之前的版本，建议升级 RTD 到 6.0.0 或之后版本。

若不想升级版本，也可以参考 RTD6.0.0 的函数 `Clock_Ip_PRAMCSetRamIWS` 进行修改。由于修改了 RTD 代码，需要自行承担风险。

若使用安全启动，并且安全启动使能了 PLL，那么，即使使用 RTD6.0.0，也可能在启动代码进行 SRAM_2 读写操作时出现读写出错问题。因为 RTD6.0.0 只解决时钟初始化后读写 SRAM_2 出错的问题。因此，该工况下，需要在启动代码进行 SRAM_2 操作前，设置 SRAM_2 的 PRCR1 寄存器的 FT_DIS 为 1。

3.4.17 Cache 无效化异常问题

在 RTD5.0.0~RTD7.0.0 的发布说明（Release note），搜索 ARTD-105430，有问题描述。

简单说，就是 RTD（RTD4.0.0 及之前版本，不包括 RTD4.0.0HF01/02）软件的一个 Bug，在 Cache 无效化（Invalidate）时没有把 Cache 内全部数据无效化。

该 Bug 导致了某些客户 Fee 模块出现异常。理论上说，该 Bug 不仅仅影响 Fee，与 Cache 无效化相关的应用都可能受到影响。

解决措施：

- 1、升级 RTD。使用 RTD4.0.0HF01/02、RTD5.0.0 及之后版本。
- 2、联系 NXP 销售，走 NXP 软件收费修复流程，修复 Bug。
- 3、自行修复。对比 RTD4.0.0 和 RTD4.0.0HF01 的差异，参照 RTD4.0.0HF01 的代码修改客户使用版本代码。该方法不推荐。

3.5 复位相关问题

3.5.1 APP 中无法获取复位源问题

检查是否 Bootloader 调用了读取复位源 API。

由于调用读取复位源 API 后会将复位源清掉，因此 APP 中无法再读到复位源。

若希望 APP 中也能读到复位源，有 3 种建议：

- 1: Bootloader 中不读取复位源，只在 APP 中读取。
- 2: Bootloader 中读取复位源后将复位源存在 ram 中，APP 从 ram 中读取复位源。

3: 修改 Bootloader 工程中读取复位源的 RTD 代码，屏蔽掉清复位源把代码。这样 APP 中可以读取到。

3.5.2 恢复模式 Recovery mode 相关问题

若使用 0.15.0 版本之前的 sBAF，那么，芯片默认是开启恢复模式（Recovery mode）的。

如果，想禁止恢复模式，需要修改 DCM 寄存器 DCMRWP1，见 HSE 参考手册。

若使用 0.15.0 版本的 sBAF，IVT 的 BCW（Bit7）和如上 DCM 的 DCMRWP1 寄存器共同确定是否使能恢复模式。

DCMRWP1 的禁止恢复模式位 BIT22 和 BIT23 上电默认值是 0，即使能恢复模式。

而 RTD 启动代码的 IVT 中，BCW 的 BIT7（如上的复位恢复模式位）默认是 0，故若使用默认 RTD 启动代码，则默认禁止恢复模式。

若需使能，需修改 BCW 的 BIT7。

3.6 ADC

3.6.1 需遵守 K3 参考手册的要求

见参考手册《S32K3xx Reference Manual》第 60 章。

请使用最新版本手册。目前最新版本是 2025 年 7 月发布的第 11 版。

很多应用问题都可用在参考手册上找到答案，建议遇到问题前/后都认真阅读。

3.6.2 温度通道采样时间的要求

温度通道需要的采样时间较长，需要的最小值是 1.2 微秒。

在数据手册有该项指标的描述：

10.2 Supply Diagnosis

The table below gives the specification for the on die supply diagnosis.

Table 36. Supply Diagnosis

Symbol	Description	Min	Typ	Max	Unit	Condition	Spec Number
AN_ACC	Offset to internally monitored supply at ADC input ^{1,2,3}	-5	0	5	%	—	—
AN_T_on	Switching time from closed (OFF) to conducting (ON) ¹	—	2.5	12	ns	—	—
AN_TADCSA	Required ADC sampling time ²	1.2	—	—	μs	—	—

1. These specs will have degraded performance when used in extended supply voltage operation range, i.e. normal supply voltage range specification is exceeded.
2. Required ADC sampling time specified by parameter AN_TADCSA needs to be used at the ADC conversion to guarantee the specified accuracy. A smaller sampling time leads to a less accurate result.

建议根据板卡实测情况，留适当余量。

温度通道建议使能预采样。

3.6.3 ADC 采样时钟的要求

见数据手册表格 35 的要求：

Table 35. SAR_ADC

Symbol	Description	Min	Typ	Max	Unit	Condition	Spec Number
VDD_HV_A	ADC Supply Voltage ¹	2.97	—	5.5	V	—	—
DVREFL	VSS / VREFL Voltage Difference ²	-100	—	100	mV	—	—
VAD_INPUT	ADC Input Voltage ³	VREFL	—	VREFH	V	—	—
fAD_CK	ADC Clock Frequency (S32K344, S32K324, S32K314, S32K342, S32K341, S32K322)	10	—	80	MHz	—	—
fAD_CK	ADC Clock Frequency (S32K312, S32K311, S32K310, S32K358, S32K348, S32K338, S32K328, S32K388, S32K389)	10	—	120	MHz	—	—

60.3.1 Clock

ADC is controlled by one clock signal, the module clock. Internally, the conversion circuit is controlled by the conversion clock, which is derived from the module clock.

The frequency of the conversion clock has to be within the limits defined in the data sheet. If the module clock frequency is higher than the maximum frequency of the conversion clock allowed during the functional conversion or during the calibration (see the chip data sheet), then you must configure the ADC conversion clock divider (`MCR[ADCLKSEL]`) so that the frequency of the conversion clock is within allowed limits. See [Table 343](#).

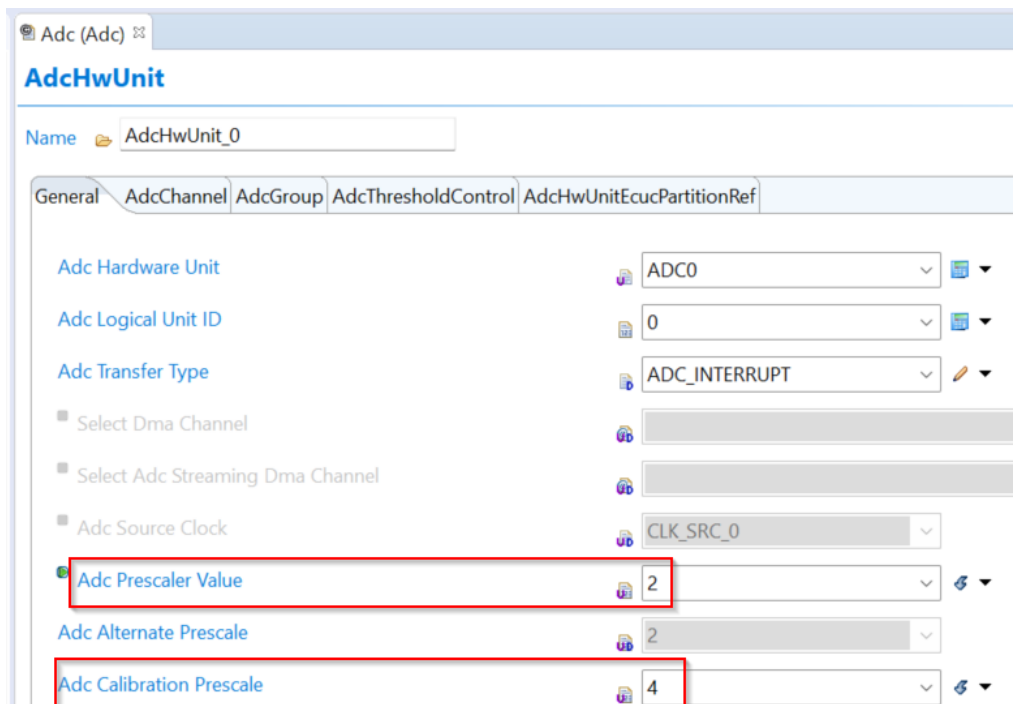
Table 343. Clock configuration for highest conversion speeds

Module Clock Frequency f_{mc}^1	Configuration during calibration	Configuration during functional conversion
40 MHz < $f_{mc} \leq$ 80 MHz	<code>MCR[ADCLKSEL] = 1h</code> <code>AMSIO[HSEN] = 0h</code> <code>AMSIO[CMPCTRL0] = 0b</code>	<code>MCR[ADCLKSEL] = 0h</code> <code>AMSIO[HSEN] = 0h</code> <code>AMSIO[CMPCTRL0] = 0b</code>
80 MHz < $f_{mc} \leq$ 120 MHz	<code>MCR[ADCLKSEL] = 1h</code> <code>AMSIO[HSEN] = 1h</code> <code>AMSIO[CMPCTRL0] = 1b</code>	<code>MCR[ADCLKSEL] = 0h</code> <code>AMSIO[HSEN] = 1h</code> <code>AMSIO[CMPCTRL0] = 1b</code>
120 MHz < $f_{mc} \leq$ 160 MHz	<code>MCR[ADCLKSEL] = 2h</code> <code>AMSIO[HSEN] = 0h</code> <code>AMSIO[CMPCTRL0] = 0b</code>	<code>MCR[ADCLKSEL] = 1h</code> <code>AMSIO[HSEN] = 0h</code> <code>AMSIO[CMPCTRL0] = 0b</code>
160 MHz < $f_{mc} \leq$ 240 MHz	<code>MCR[ADCLKSEL] = 2h</code> <code>AMSIO[HSEN] = 1h</code> <code>AMSIO[CMPCTRL0] = 1b</code>	<code>MCR[ADCLKSEL] = 1h</code> <code>AMSIO[HSEN] = 1h</code> <code>AMSIO[CMPCTRL0] = 1b</code>

例如，K324,ADC 转化时钟最高 80M，校准时钟最高 40M。

超过如上时钟频率，可能导致校准失败，数据采集异常，甚至损坏 MCU。

如下是 K324 的最高转换频率设置：



3.6.4 建议使能预采样

建议使能通道的预采样，有助于减少通道间的干扰，在参考手册有如下描述：

60.3.9 Presampling

The presampling feature enables pre-charge or discharge of the ADC internal sample capacitor node to a defined level before sampling of the selected analog input channel starts. This is useful for resetting information (history effect/offset) from the most recent conversion. During presampling, ADC samples the internally generated voltage. During the sampling phase, ADC samples analog input coming from pads.

预采样使能后，采样时间大约增加一倍，转换时间不变，这是唯一不利的地方。

3.6.5 分辨率建议设置为 12 位

推荐在 EB 或 CT 配置工具中，配置 12 位分辨率。

因为芯片出厂测试按照 12 位分辨率测试，能保证该分辨率下的精度。

设置 10 位分辨率情况下，信号精度没有 12 位好。

设置 14 位分辨率情况下，TUE 指标只是略好于 12 位分辨率，并非为 12 位分辨率时 TUE 指标的 1/4。

3.6.6 K310 和 K311 ADC 使用注意事项

由于 RTD4.0.0~RTD6.0.0 QLP04 的 Bug，使用 K310/K311 的部分 ADC 通道时，若不是默认 GPIO 映射，需要修改 DCMRWF4 的对应位，否则无法采集这些通道。

可以查看 Port 初始化后，DCMRWF4 寄存器值是否与预期一致。

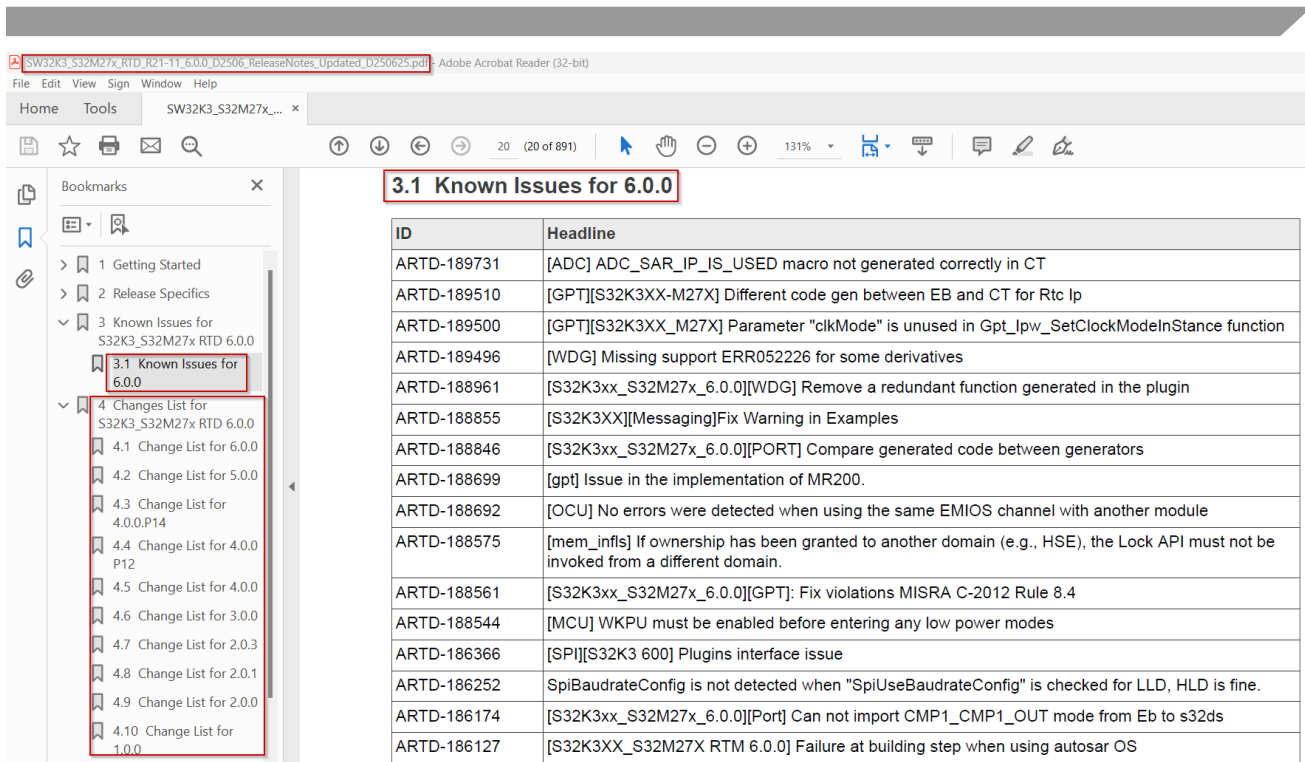
也可以通过修改 MCAL 代码和配置文件修改，可咨询 NXP FAE。

3.6.7 检查最新版本 RTD 的发布说明中提到的 Bug

请使用最新版本的 RTD 的发布说明（目前最新版本是 6.0.0）。

搜索关键字“ADC”，会看到 RTD 最新版板仍存在的 Bug，以及其它版本修复了之前版本的 Bug。

建议查看每一条是否与项目应用相关，及时发现问题。



3.6.8 建议参考 NXP 的示例工程

包括 RTD 自带的示例工程以及 NXP 重庆应用工程师开发的示例工程等。

3.6.9 ADC 校准建议

如果板卡电源或时钟不稳定，可能导致校准失败。

因此，推荐的校准策略如下：

- (1) 在 MCU 初始化过程中，对 ADC 校准的返回值进行判断。
- (2) 若返回值表明校准成功，程序正常运行。
- (3) 若返回值表明校准未成功，建议校准多次，若校准成功，程序继续运行。
- (4) 若多次校准均未成功，调用 ADC 自检函数 `Adc_SelfTest` ，
- (5) 若自检成功，再调用一次 ADC 校准函数。若校准成功，程序继续运行。若校准失败，调用复位 API（功能性复位或破坏性复位均可），MCU 复位。
- (6) 若自检失败，调用复位 API（功能性复位或破坏性复位均可），MCU 复位。
- (7) 当然，校准多次均失败，不进行 ADC 自检，直接进行复位也是可以的。

3.6.10 ADC 超时时间设置

超时时间设置太小，可能导致 ADC 校准失败，建议预留 2 倍以上余量。

若超时方法是基于 Dummy 计数，要考虑代码优化对 Dummy 计数的影响，建议实测 Dummy，并余量 2 倍以上余量。

3.6.11 ADC 引脚上的高压脉冲可能损坏芯片

若由于外部芯片原因，导致一个高压脉冲（例如，20V，300us，具体见数据手册的说明）施加到 ADC 引脚，可能会导致 MCU 损坏，建议增加 TVS 保护。

3.6.12 ADC 自检失败问题

有客户遇到低温下 ADC 自检的失败。

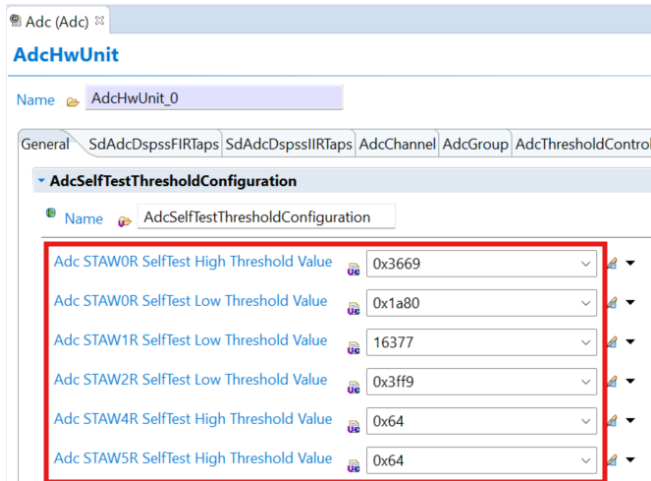
解决措施：

- 1、放大 ADC 自检阈值。

Table 351. Recommended self-test threshold values

Register field	Hex value	Decimal value	Comment
STAW0R[THRH]	3669h	13929	The threshold values depend on the stability of the supply and the reference voltage. When you see voltage variations, widen the limits.
STAW0R[THRL]	5 V reference: 1A7Fh 3 V reference: 2B27h	5 V reference: 6784 3 V reference: 11047	
STAW1R[THRL]	3FF9h	16377	
STAW2R[THRL]	3FF9h	16377	
STAW4R[THRH]	20h	32	
STAW4R[THRL]	7FE0h	-32	
STAW5R[THRH]	20h	32	
STAW5R[THRL]	7FE0h	-32	

在部分 RTD 版本（例如 RTD6.0.0）可以配置该阈值，部分 RTD 版本只能手动修改该阈值：



- 2、降低 ADC 模块时钟频率。

3.6.13 ADC Read Group 偶发失败问题

有客户使用 K341, RTD3.0.0 P07, 遇到调用 `Adc_ReadGroup`, 偶发返回值为 `E_NOT_OK` 的问题。

原因是 CDR 寄存器读取是非原子操作, 若读取该寄存器过程被中断 (例如被 CAN 中断打断), 可能会导致读取的 CDR 寄存器中 `VALID` 为 0, 进而 `Adc_ReadGroup` 的返回值可能为 `E_NOT_OK`。

建议解决措施是: 修改 RTD 代码, 在调用 CDR 函数前后关闭/打开全局中断, 如下:

```
SchM_Enter_Adc_ADC_EXCLUSIVE_AREA_10();
```

```
Cdr = CDR(AdcBasePtr, u32ChnIdx);
```

```
SchM_Exit_Adc_ADC_EXCLUSIVE_AREA_10();
```

RTD6.0.0 及之后版本实现 CDR 读取的方式不同, 修改方式如下:

```
SchM_Enter_Adc_ADC_EXCLUSIVE_AREA_10();
```

```
Cdr = CdrBaseRAddr[ChnIdx];
```

```
SchM_Exit_Adc_ADC_EXCLUSIVE_AREA_10();
```

3.7 CAN

3.7.1 使能 Enhance FIFO 后, 部分邮箱丢帧问题

检查使能 `Enhanced FIFO` 后, 如下邮箱是否已使用。

K311/K312 : 0,2,10,12,20,22,30,32,40,50,60

K314/ K324/K344/K358 : 0,8,18,28,38,48,58,66,68,76,78,86,88

其它型号受影响邮箱请查看对应型号的最新的勘误表。

在使能 `Enhanced FIFO` 后, K312 和 K3x4 的如上邮箱建议不要使用 (包括发送和接收), 否则可能导致丢帧。

使用 `RTD6.0.0` 时, 如上邮箱也不能使用。

详见勘误表 (ERRATA) 的条目 **ERR052438**。

3.7.2 CAN 模块处于 start 状态下反初始化失败

CAN 模块反初始化成功的前提是 CAN 模块已完成初始化, 并且未处于 start 状态。

如果 CAN 模块处于 start 状态, 请调用 API 令 CAN 处于 stop 状态, 然后再进行反初始化。

3.7.3 CAN 帧发送乱序问题

EB 配置默认按照 CAN 帧优先级进行发送。若希望按照 CAN 邮箱 ID 发送，可以手动设置 CAN 模块控制寄存器 1 的 LBUF 位。

3.7.4 建议 CAN FD 使能传输延时补偿 TDC

使用 CAN FD 时，建议使能传输延时补偿 TDC。

CANFD 允许通过 BRS 使能在数据段使用比仲裁段更高的波特率，最高速度能到达 8M。和 LIN 类似，CAN 为了检测发送错误，保证数据发送的可靠性，也会在 TX 引脚发送一个 bit 时同步在 RX 引脚接收外部 CAN 收发器 loop back 回来的 bit，然后再在 CAN 控制器内部进行比较，如果发送的 bit 和接收的 bit 不一致就会触发 bit error。当 BRS 使能时，CANFD 的 bit time 的时间 (200ns@5M) 可能会比外部 CAN 收发器的 loop delay 的时间更短，这就有可能导致 TX 引脚发送的 bit 和经过外部 CAN 收发器 loop back 回到 RX 引脚的 bit 不同，从而导致 CAN 总线错误。

Transceiver delay (TD) 传输节点延迟被定义为从控制器的 Tx 引脚 FlipFlop 到 Rx 引脚 FlipFlop 的信号传播延迟，包含芯片内部控制器和外部 CAN 收发器的延迟时间，对于 TJA1044 来说这个延迟在 50-230ns 之间，而 Second Sample Point(SSP) 是为了补偿节点的传输延迟，防止在一个 bit 时间段内无法正确采样到 Loop back 回来的 bit 位而重新定义的二次采样点，这个补偿会在 FlexCAN 初始化配置完成后由硬件自动触发，无需软件干预(ns 级别，软件来不及处理，而且需要频繁处理)。

3.7.5 CAN 采样点不满足要求问题

提高 CAN 协议时钟频率，可以改善 CAN 采样点。

需要联系 Vector 的 FAE 检查 VH6501 测试方案是否正确。包括测试脚本是否正确、测试配置是否合理、故障注入点是否正确（应注入到显性位，并且略早于采样点）等等。

实际案例：由于测试脚本问题导致采样点不能满足 OEM 要求。修改脚本后，问题解决。

3.7.6 时钟配置错误导致的问题

时钟配置必须严格遵循参考手册列出的几种配置，否则可能导致 CAN 通讯异常。

3.8 SPI

3.8.1 使用 GPIO 做片选信号时 CLK 信号异常脉冲问题

检查是否使用 GPIO 做片选 CS。若使用 SPI 模块的片选，异常脉冲不会产生问题。

若使用，请检查时钟信号是否有一个异常脉冲（针对 K3 部分芯片的 Bug，RTD 默认会复位 SPI 模块，产生一个短脉冲，一般不会对应用有影响）。

若存在，请根据 SPI 的从芯片/主芯片要求，确定是否会有影响。

若有影响，

请参考 RTD3.0.0 的第二个解决措施（定义 ERR_IPV_LPSPIV2_E050456_2ND_SOLUTION），或 RTD4.0.0P01（ERR_IPV_LPSPIV2_E050456）及以后版本的做法。

3.8.2 配置参数需合理

在配置页面可以设置 SPI 时钟到片选、片选到时钟和片选到片选等参数，需要合理设置，以免有过多时间的浪费。

3.8.3 SPI 同步传输时间长问题

通过开启 I CACHE 和 D CACHE，将 SPI 模块代码优化等级设置为 Os 等方式减小传输时间。

3.8.4 时钟配置错误导致的问题

时钟配置必须严格遵循参考手册列出的几种配置，否则可能导致 SPI 通讯异常。

3.9 I2C

3.9.1 采样保持时间无法调整问题

若遇到 I2C 采样保持时间时序问题无法通过 EB 配置解决时，可尝试手动使能 SCR->FILTEN bit。

3.9.2 时钟配置错误导致的问题

时钟配置必须严格遵循参考手册列出的几种配置，否则可能导致 I2C 通讯异常。

3.9.3 I2C 从模式在快速模式下的问题

若 MCU 的 I2C 模块工作在从模式，并且为 FAST 模式（400Kbps），并且是读操作。如果由于其它中断的影响，导致 I2C Repeat start 中断未及时响应，可能出现 I2C 时钟线拉低问题。

这是 RTD 代码的 Bug，可通过将函数 Lpi2c_Ip_SlaveCheckDataEvent 从 3309 行移到 3333 行的办法解决（以 RTD6.0.0 QLP04 为例），如下：

```
3295: static void Lpi2c_Ip_SlaveIRQHandlerInternal (uint8 Instance)
3296: {
3297:     LPI2C_Type *BaseAddr;
3298:     Lpi2c_Ip_SlaveStateType * slave;
3299:     boolean StopDetect = FALSE;
3300:     boolean RepeatStartDetect = FALSE;
3301:
3302:     BaseAddr = Lpi2c_Ip_pxBase[Instance];
3303:     Slave = Lpi2c_Ip_pxSlaveState[Instance];
3304:
3305:     StopDetect = LPI2C_Get_SlaveSTOPDetectEvent(BaseAddr);
3306:     RepeatStartDetect = LPI2C_Get_SlaveRepeatedStartEvent(BaseAddr);
3307:
3308:     /* Check address valid and tx/rx event */
3309:     //Lpi2c_Ip_SlaveCheckDataEvent(Instance, BaseAddr, Slave); //屏蔽该代码, 将其转移到3333行
3310:
3311:     if (RepeatStartDetect)
3312:     {
3313:         Slave->RepeatedStarts++;
3314:
3315:         if ((1U == Slave->RepeatedStarts) && (Slave->Is10bitAddress))
3316:         {
3317:             RepeatStartDetect = FALSE;
3318:             LPI2C_Clear_SlaveRepeatedStartEvent(BaseAddr);
3319:         }
3320:     }
3321:
3322:     if ((TRUE == StopDetect) || (TRUE == RepeatStartDetect))
3323:     {
3324:         /* Stop/repeated start detected */
3325:         Lpi2c_Ip_SlaveStopDetectHandler(BaseAddr, Slave);
3326:
3327:         if (TRUE == StopDetect)
3328:         {
3329:             /* reset repeated starts for a new transfer */
3330:             Slave->RepeatedStarts = 0U;
3331:         }
3332:     }
3333:     Lpi2c_Ip_SlaveCheckDataEvent(Instance, BaseAddr, Slave); //3309行代码转移到此, 以解决问题
3334:     /* Check for slave errors */
3335:     Lpi2c_Ip_SlaveCheckErrorEvent(BaseAddr, Slave);
3336: } /* end Lpi2c_Ip_SlaveIRQHandlerInternal */
```

3.9.4 I2C 上拉电阻设置建议

用户对 SCK/SDA 管脚不要设置使能内部上拉，需要直接使用外部上拉。

PC 总线的通信速率与上拉电阻的选择密切相关，选择不当会导致通信不稳定甚至完全失败。

核心原理是上拉电阻和总线等效电容组成的 RC 时间常数是会影响波形的上升时间，造成 I2C 速度的瓶颈。

所以外部上拉电阻数值在不同的速率下是不同的，NXP 推荐用户参照 I2C 的规范选择上拉电阻。

在产品阶段，用户可以这样选择：100KHz 速率时，上拉电阻可以选择 4.7K；400KHz 速率时，上拉电阻可以选择 2.2K。最终的上拉电阻参数需要用户在最终的压力测试后调整决定。

3.10 以太网

3.10.1 以太网初始化前外部时钟需稳定

以太网初始化前，若外部时钟未稳定，可能导致初始化失败。

3.10.2 千兆以太网使用注意事项

若使用千兆以太网 GMAC，并且使用发送引脚输出时钟（tx clock output on pad），需要手写 DCMRWF1 的"MAX_TX_RMII CLK loopback ENABLE"位。并且，将发送时钟配置成 250M。

3.11 eMIOS

3.11.1 PWM 从 100% 占空比切换到 0% 占空比偶发失败

若 PWM 通道工作在 OPWMCB 模式，占空比从 100% 切换到 0% 会出现偶发失败情况。

若 PWM 通道工作在 OPWMCB 模式，占空比从 0% 切换到 100% 也会出现偶发失败情况。

在 RTD4.0.0~RTD6.0.0 都存在此 Bug。

具体分析请见文档《EMIOS OPWMCB DutyCycle 0% update fails》

可联系 NXP FAE 获取修复代码和如上文档。

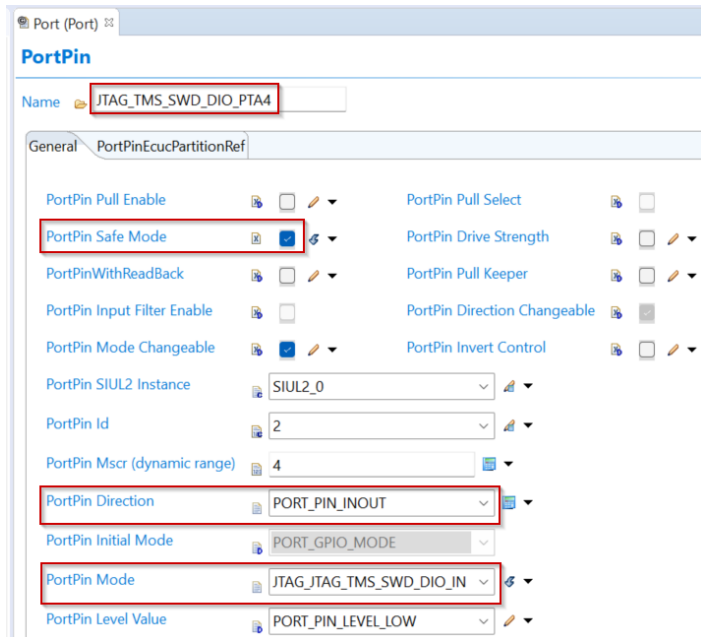
3.12 端口引脚设置

3.12.1 建议将 JTAG 信号配置安全模式

建议将 JTAG/SWD 引脚配置使能安全模式（SMC），好处是芯片进入安全模式后 JTAG 功能保存，便于连接调试器定位问题。

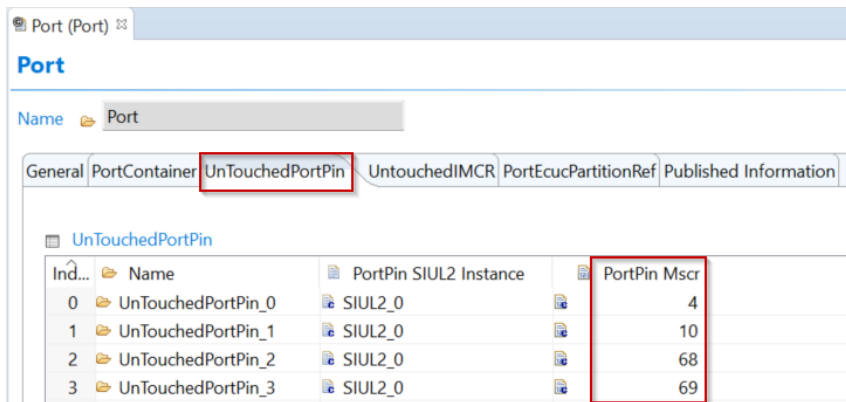
芯片只有在严重故障情况下才可能进入安全模式，概率非常低。

以 JTAG 中的 TMS 信号为例，配置如下：



将 JTAG/SWD 引脚放入 “UnTouchPortPin” 也可以，只是芯片进入安全模式下无法调试。

引脚放入 UnTouchPortPin 后，端口初始化时不对这些引脚配置，保持默认功能。

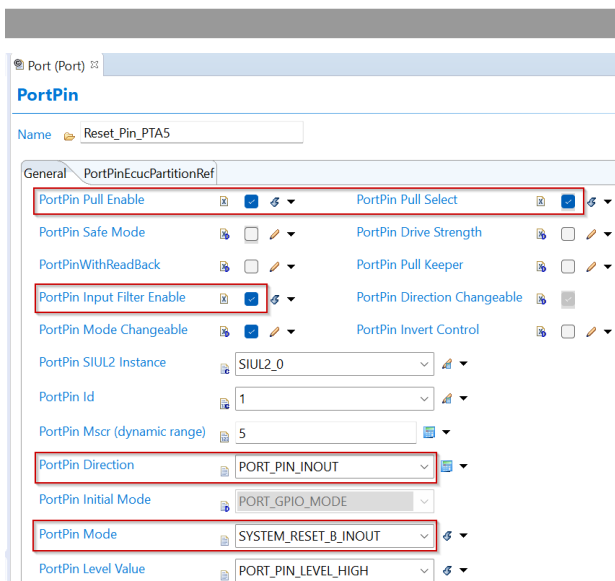


3.12.2 建议将复位信号配置成滤波使能

建议将复位引脚配置滤波使能，这会对通过 ESD 试验有帮助。

将复位引脚放入 “UnTouchPortPin” 也可以，这样不会使能复位滤波功能。

不要既不配置复位引脚，也不将其放入 “UnTouchPortPin”。那样的话，复位引脚失去复位功能。



3.12.3 合理设置未使用引脚

未在“使用引脚”和“UnTouchPortPin”中配置的引脚，会按照未使用引脚配置。

建议设置成禁止、输入且下拉使能，或设置成输出。

不建议设置成输入，并且未使能上/下拉。

不遵循可能低功耗电流超标。

3.12.4 PTE13 引脚配置

若 PTE13 用于三极管基极控制功能，推荐的做法是：在 PORT 模块中不配置该引脚，即保持默认。

1、若项目已量产，并且在 PORT 中将该引脚配置成了 PG_VRC_CTRL_OUT，由于功能是正常的，暂未发现问题，可暂不修改。建议在下一软件升级时按照上面提到的推荐做法修改。

2、若项目已量产，并且在 PORT 中将该引脚配置成了除 PG_VRC_CTRL_OUT 外的其它功能，建议测试 PTE13 输出是否正常（2.2V 左右），并尽早按照上面提到的推荐做法修改。

3、若项目未量产，请按照上面提到的推荐做法配置。

4、需要在 MCU 模块勾选 Last Mile Regulator Base Control Enable 和 Last Mile Regulator Enable。

以上配置正确后，MCU 周期运行时，测量 PTE13 引脚会是一个 2.2V 左右的电平。

3.13 中断相关

3.13.1 中断服务程序执行时间长问题

MCAL 执行中断服务程序（ISR）的时间在几微秒到十几微秒之间不等，耗时并不算长。
常见解决措施：

- 1、优化中断回调函数。ISR 内包括中断回调函数，减少回调函数执行时间可实现总的 ISR 执行时间变短。例如，将非必要的代码（例如计算）放在中断外的任务中执行。
- 2、启用 I CACHE 和 D CACHE。
- 3、将中断回调函数（或该函数内的耗时长函数）放入 ITCM 中执行。

3.13.2 尽量避免长时间关闭全局中断

长时间关闭全局中断，会导致其它中断无法得到响应，进而产生问题，尽量减少关闭全局中断的时间。例如，长时间关闭全局中断，导致 CAN/I2C/SPI/UART 等中断长时间无法进入，进而出现丢帧、数据溢出（OverRun）等问题。

3.13.3 中断向量表需要 128 字节对齐

中断向量表在 RAM 中起始地址需要 128 字节对齐，若不对齐，可能产生异常情况。

快速唤醒设置的中断向量表，如果将其拷贝到 Standby RAM，RAM 地址也需 128 字节对齐。

若不将其拷贝到 Standby RAM，那么向量表在 FLASH 中的地址需要 128 字节对齐。

3.13.4 中断优先级设置要合理

应评估好哪些中断的优先级高，哪些中断优先级低。不合理的设置会导致高优先级功能不能及时得到响应，进而引起问题。

DMA 各个通道的优先级也需合理设置，建议配置使能抢占，可以让高优先级 DMA 优先执行。

DMA 设置优先级数字越大表示优先级越高。

3.14 RTD 相关

3.14.1 推荐使用新版本 RTD

由于新版本 RTD 会修复旧版本 RTD 的 Bug，同时提升性能。

新版本的可维护性也更好，出现的 Bug，一般会在新版本中修复。

例如，RTD6.0.0 提升了 CAN 中断服务程序的执行时间。

推荐使用新版本 RTD。目前最新版本 RTD 是 RTD7.0.0。每个版本 RTD 的 Release note 中有 Bug 修复和新功能增加说明，建议认真阅读。

需要注意：

由于该版本 RTD 暂无示例工程，预计 2026 年 2 月的 7.0.0 版本才提供示例工程，这会带来一些不便。另外，该版本暂无 Crypto 模块，预计几个月后才会有包含该模块的版本。该版本支持 AUTOSAR R23-11。综合这些差异，客户最终确定选择 7.0.0，还是 6.0.0 中的最新量产版本。

3.14.2 参考 RTD 示例工程定位问题

若调试遇到问题，请运行 RTD 示例工程，或基于 RTD 示例工程稍作修改（适配实际板卡），验证示例工程是否功能正常。

若示例工程运行正常，对比与自己工程的差异。

若示例工程运行异常，排查硬件是否存在问题。

RTD 示例工程包括 MCAL 和非 MCAL 两种类型。

对于使用 EB 工具的客户，也建议安装 S32DS，并通过 S32DS 导入 RTD。

因为这样就可以直接使用 S32DS 导入 RTD 示例工程，调试 RTD 中 MCAL 和非 MCAL 示例工程，对解决问题很有帮助。

3.14.3 RTD 中 EB 配置示例工程移植到 S32DS

RTD 示例工程中 EBT 目录下含有 EB 配置工程和 MCAL 工程。

由于示例工程是采用命令行编译，有些客户习惯在 S32DS 上调试，所以，将这种示例工程移植到 S32DS 上很有用。

若有 NXP 社区账号，可访问如下链接或网址，实现工程移植：

[\[S32K3 Tools Part\]How to import RTD EB project into S32DS - NXP Community](#)

<https://community.nxp.com/t5/S32K-Knowledge-Base/S32K3-Tools-Part-How-to-import-RTD-EB-project-into-S32DS/ta-p/1966207>

若没有 NXP 账号，建议注册，访问 NXP 社区能解决很多工作中遇到的问题。也可联系代理商 FAE 帮忙将文字拷贝到 Word，并下载该链接的示例工程。

该链接内有移植过程的视频，很方便学习如何移植。

该链接参考的移植博客文章建议参考，链接如下：

<https://www.wpgdadatong.com.cn/blog/detail/74936>

3.14.4 RTD 中 EB 配置示例工程移植到 K312 等其它型号的 S32DS 工程

可参考 NXP 社区如下文章解决：

<https://community.nxp.com/t5/S32K-Knowledge-Base/S32K3-Tools-Part-How-to-port-RTD-s-existing-MCAL-demo-to-other/ta-p/1966315>

若没有 NXP 账号，建议注册，访问 NXP 社区能解决很多工作中遇到的问题。

也可联系代理商 FAE 获取《S32K3 工具篇 8：如何移植 RTD 现有 MCAL demo 到其他 K3 芯片.pdf》。

3.14.5 RTD 中 S32DS CT 配置示例工程移植到 K312

由于 RTD3.0.0~5.0.0 没有 K312 的示例工程，FAE 团队总结了制作 RTD3.0.0~5.0.0 K312 示例工程的方法（基于 K344 示例工程转换成 K312 示例工程）并且完成了很多模块的移植 Demo。

该方法也适用于制作其它型号（例如 K311）的示例工程。

指导文档《快速在同一个版本的 RTD 内，移植工程样例 same RTD version migration porting example project s32k3for_S32DS_Liek_li.docx》。

可联系代理商 FAE 获取该文档以及已移植完成的示例工程。

3.14.6 使用新版本 RTD 验证复现问题

由于新版本 RTD 会修复旧版本 RTD 的 Bug。

在使用某个版本的 RTD 进行调试遇到问题，尝试使用新版本，观测问题是否复现。

这个方法能解决部分问题。

新版本 RTD 可能未把修复的 Bug 都列全。

实际案例：客户遇到问题，查看新版本 RTD，发现问题相关函数有修改，然后，使用新版 RTD 示例工程（可能稍作修改）或参考新版 RTD 修改使用版本的代码，问题解决。

使用新版本 RTD 验证出现的问题是一个很典型和实用的方法。很多客户都通过该方法解决了问题。通过该方法能快速找到问题，加快问题解决时间。

3.14.7 根据最新版本 RTD 的发布说明排查使用 RTD 版本的 Bug

每个版本的 RTD 都有对应的发布说明（Release note），该文件在下载 RTD 安装文件时会看到。

以 RTD6.0.0 为例，

该文件名称为：

SW32K3_S32M27x_RTD_R21-11_6.0.0_D2506_ReleaseNotes_Updated_D250625.pdf

在该文件内包括该版本已知的 Bug，以及修复了之前版本的哪些 Bug。

建议下载使用 RTD 对应版本的发布说明，和遵循该 AUTOSAR 版本的最新版本的发布说明。

例如，使用 RTD2.0.1，则需要下载 RTD2.0.1 版本的发布说明，和遵循 AUTOSAR4.4 标准的最新的 RTD4.0.0P24 和 RTD2.0.3 版本的发布说明。

通过 RTD2.0.1 版本的发布说明，知道该版本有哪些已知 Bug。

通过遵循 AUTOSAR4.4 标准的最新的 RTD 版本（RTD4.0.0P24）和 RTD2.0.3 的发布说明，知道 RTD2.0.1 版本哪些 Bug 被修复。

例如，使用 RTD3.0.0，则需要下载 RTD3.0.0 版本的发布说明，和遵循 AUTOSAR4.7（R21-11）标准的最新的 RTD 版本（RTD6.0.0）的发布说明。

实际案例：客户遇到 FlexIO 做 SPI 主模块时，同一 Job 下两个 Channel 间片选信号并非预期的低电平。

经查找，在 RTD5.0.0 Release note 中已知问题中有该问题的说明，见下图：

3 Known Issues for S32K3 RTD 5.0.0

3.1 Known Issues for 5.0.0

ARTD-54102	[SPI][S32K][FLEXIO] Master CS is only continuous between datas in channel but not between channels in job
------------	---

对 NXP 内部同事，可查看 Jira 系统确认客户问题是否是已知问题。

3.14.8 链接文件问题

需检查链接文件内容设置是否正确。包括每个段的起始地址、大小（例如 `__RAM_CACHEABLE_SIZE`）、是否与其它段重叠等等。

可参考对比 RTD 自带链接文件。

可参考 NXP 重庆 AE 编写的链接文件应用笔记初稿《S32K3xx Linker File and Startup Code Application Note_draft.pdf》，文档详细介绍了链接文件的使用，可从代理商 FAE 获取。

3.14.9 RTD1.0.0 和 RTD2.0.0 启动代码耗时长问题

检查启动代码中对数据搬运是否采用 4 字节搬运方式。

若使用 RTD3.0.0 之前版本，在启动代码中对数据搬运采用每次搬运 1 字节的方式，数据量大时耗时较长，可参考 RTD3.0.0 版本启动代码，数据搬运采用每次搬运 4 字节的方式，会加快数据拷贝时间。若数据量特别大，耗时很长，也可考虑在搬运数据前打开 CACHE，加快数据搬运速度。

3.15 编译器相关

3.15.1 优化等级设置不合理会出现问题

检查 RTD 代码是否按照 RTD 发布说明 Release note 中优化选项设置。

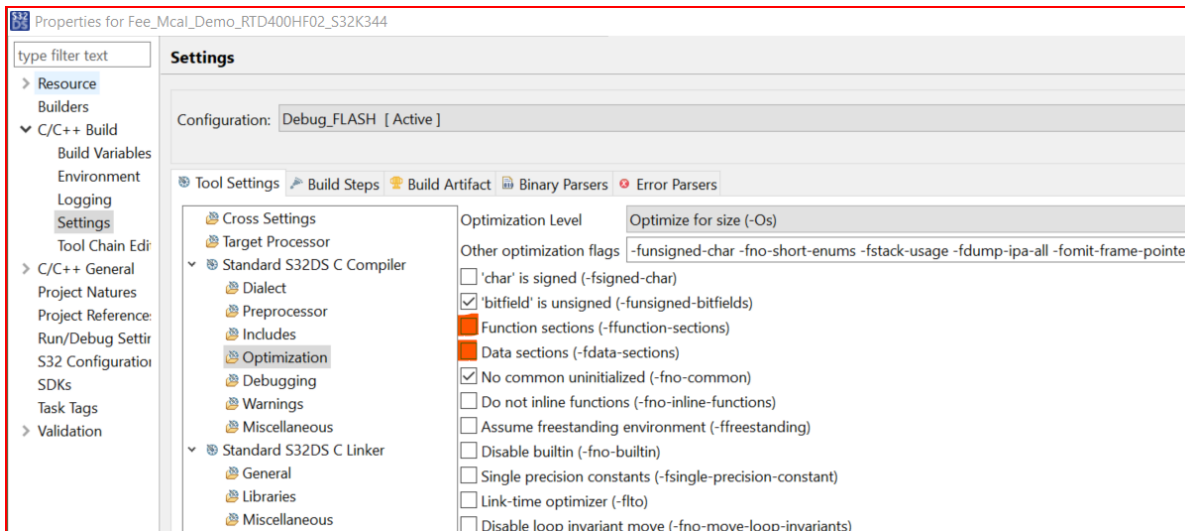
推荐按照该文件中的说明设置。

若未遵循，请通过充分的试验验证，以免由于优化设置产生问题。

RTD 代码和用户代码可设置不同的优化。

可按照文件夹/文件设置不同的优化。例如，用户代码设置成 O0，RTD 代码设置成 Os。

推荐按照下图设置：



RTD 优化等级设置为 Os。

Other optimization flags 推荐设置如下：

-funsigned-char -fno-short-enums -fstack-usage -fdump-ipa-all -fomit-frame-pointer

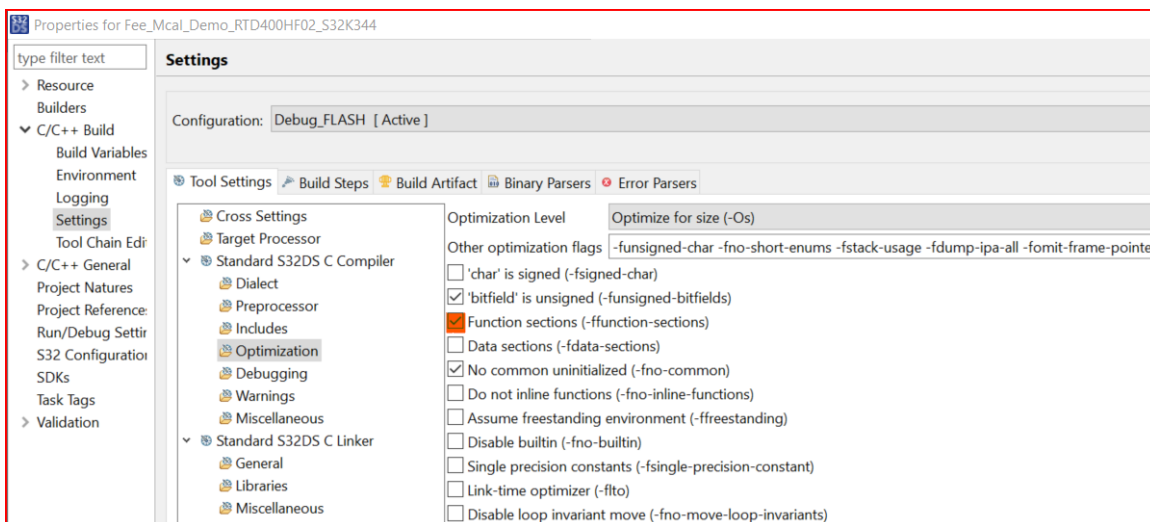
下面勾选参照图中设置。

注意：优化设置不当，是产生异常的一个主要可能。

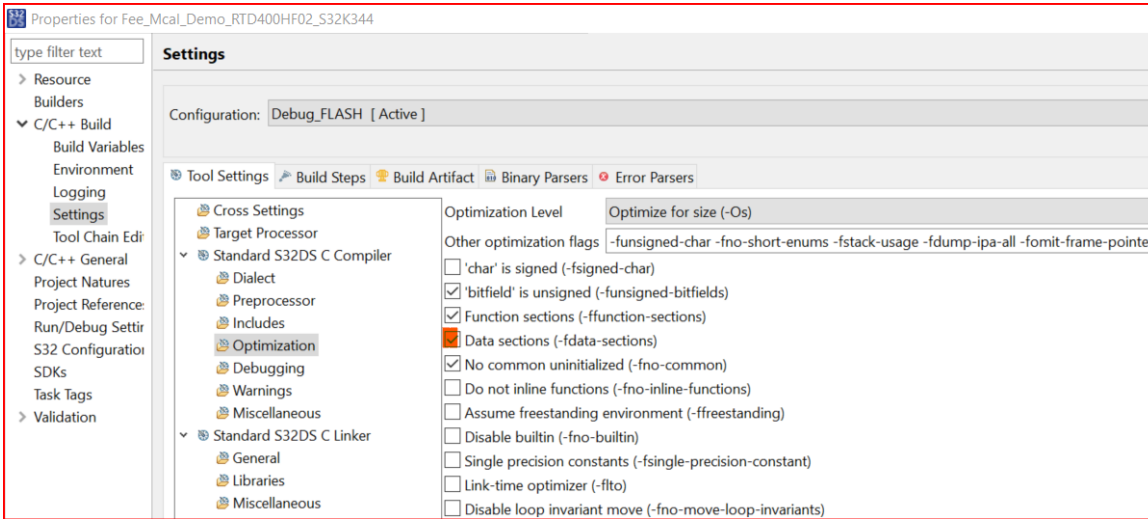
若出现问题，可尝试使用推荐的优化设置，观测问题是否消失。

建议将 Setting->Standard S32DS C Linker->Miscellaneous 下的“Removed unused sections”勾选。

若需提升代码执行性能，降低 CPU 负荷，下图中标红的选项也可勾选，但要**做充分验证**。



若需进一步提升代码执行性能，下图中标红的选项也可勾选，但要做充分验证。



因为，勾选“Data section”可能导致某些变量本打算分别到某个段，实际被分配到另一个段。

可通过.map 文件查看变量/函数是否编译到希望的段。

例如，使用 DMA 用的缓存 Buffer 需要放置在 Non cacheable 区。如果勾选“Data section”，并且使用下图所示的#include “XXXMemMap.h”的方式将 Buffer 放入 Non cacheable 区，则实际可能放入 cacheable 区，产生数据不一致问题。使用下图所示的 attribute 方法，不受这个勾选影响。

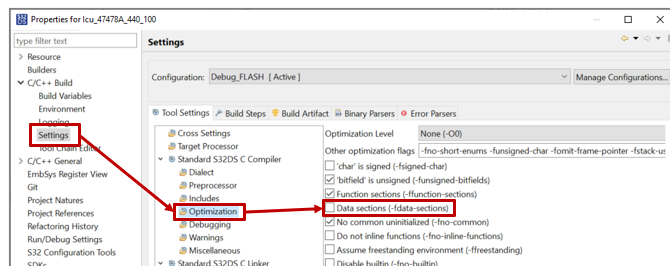
• Data Cache Restrictions:

- In the DMA transfer mode, DMA transfers may issue cache coherency problems. To avoid possible coherency issues when D-CACHE is enabled, the user shall ensure that the buffers used as TCD source and destination are allocated in the NON-CACHEABLE area (by means of `Icu_Memmap`).
- **Disable `-fdata-section`** to make `Icu_Memmap` effective.

```
#define ICU_START_SEC_VAR_CLEARED_UNSPECIFIED_NO_CACHEABLE
#include "Icu_Memmap.h"

#if (ICU_SIGNALMEASUREMENT_USES_DMA == STD_ON)
/** @brief array for saving value of DMA */
static volatile Icu_ValueType Icu_aDmaBuffer[ICU_MAX_CHANNEL][ICU_DMA_MAJORLOOP_COUNT];
/** @brief Array for saving the period */
static volatile Icu_ValueType Icu_aIsSecondInterrupt[ICU_MAX_CHANNEL];
/** @brief Array for saving the period */
static volatile Icu_ValueType Icu_aFirstEdgeTimeStamp[ICU_MAX_CHANNEL];
#endif /* ICU_SIGNALMEASUREMENT_USES_DMA */
#define ICU_STOP_SEC_VAR_CLEARED_UNSPECIFIED_NO_CACHEABLE
#include "Icu_Memmap.h"
```

`__attribute__((section("mcal_data_no_cacheable")))`



3.15.2 操作系统的优化等级需合理设置

检测操作系统（OS）优化等级设置是否合适。

有客户遇到操作系统（OS）和基础软件 BSW 都设置成 Os 后，OS 跑飞问题。然后，将操作系统（OS）优化等级设置为 O1，BSW 优化等级设置成 Os 问题消失。

因此，需确认操作系统的推荐的优化设置。

3.15.3 IAR 编译器

使用 IAR 编译器进行编译或调试时遇到问题，建议先与 IAR 技术支持联系。

有些客户遇到 IAR 编译器编译和调试的问题，通过联系 IAR 技术支持得到了解决。

NXP 技术支持一般都没有 IAR 编译器，对 IAR 不熟悉。

3.16 调试器相关

3.16.1 当用某种调试器无法连接板卡时，尝试其它调试器

例如，板卡周期复位，使用某种调试器无法连接，此时，可尝试劳德巴赫等其它调试器，可能连接上。

3.16.2 尝试不带调试器后板卡运行是否正常

若带调试器运行异常时，可尝试不带调试器，观测能否复现问题。若不能复现问题，可能是调试器的影响。

例如通过 attach，或点灯，翻转 IO 等方式确认。

打开 Memory、寄存器、变量观测页面，有时会对调试有影响，可尝试关闭这些调试页面。

3.16.3 勾选调试器异常捕获选项造成的异常

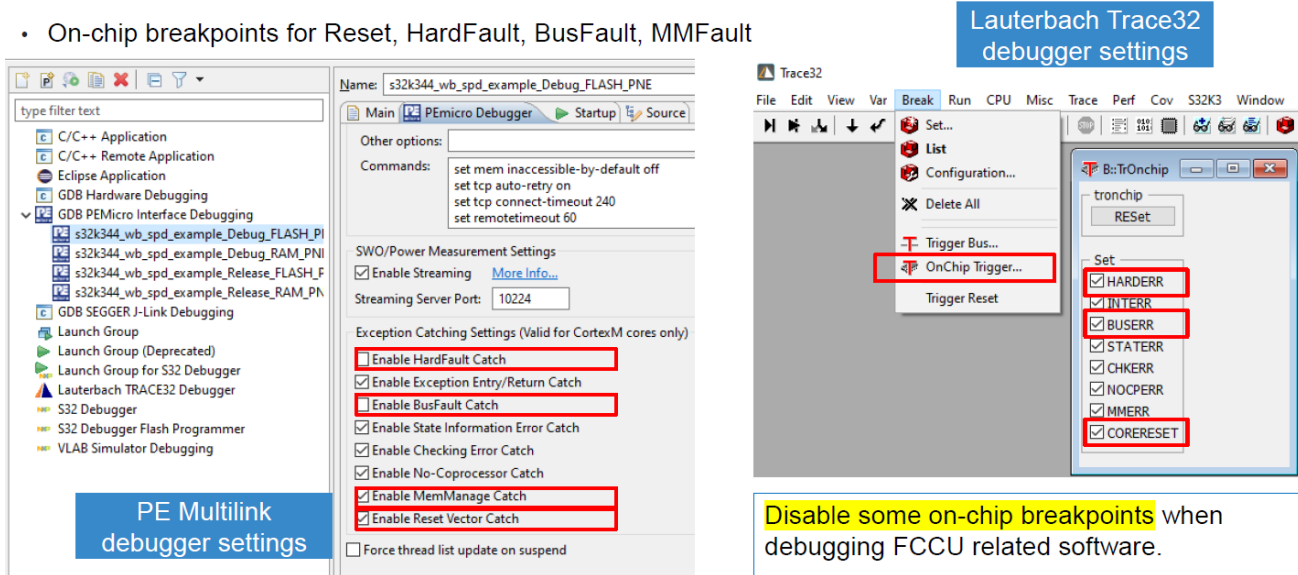
检查 FCCU 调试时，是否勾选了异常捕捉。

若勾选了一些异常捕捉，可能导致测试项失败。

见下图，FCCU 调试时，红框内选项都不要勾选。

Disable the on-chip breakpoints when exceptions happened

- On-chip breakpoints for Reset, HardFault, BusFault, MMFault



3.16.4 调试器对 RAM 区进行 ECC 初始化

检查是否调试器影响 RAM ECC 的产生。

调试器会对 RAM 进行 ECC 初始化，所以，可能出现带调试器时没用 RAM ECC 问题，不带会有。

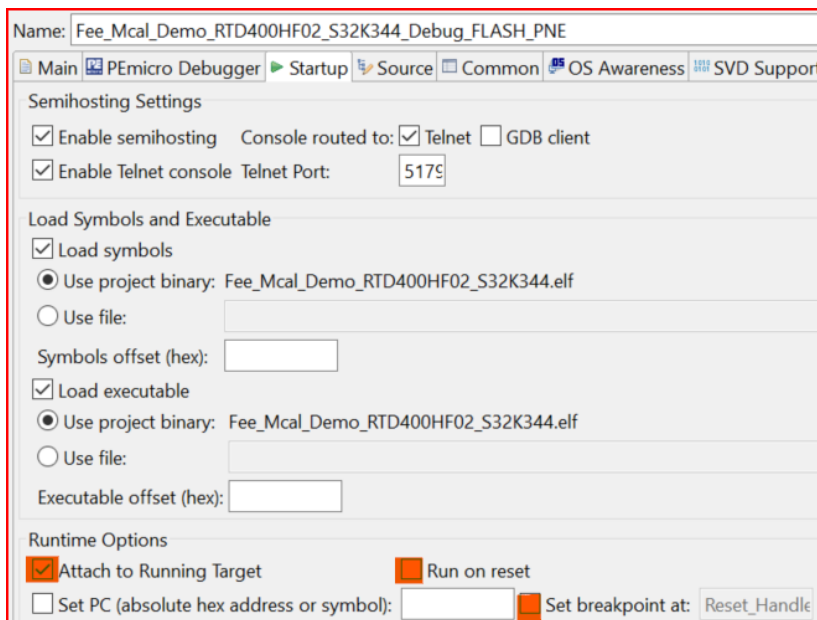
解决办法是检查程序中是否对 RAM 按照 8 字节写的方式进行了 ECC 初始化。

3.16.5 通过 Attach 定位问题

Attach 是排查问题的常用方法。

因为通过 Attach，不需要重新烧录代码，不会产生复位，可连接 MCU。

Attach 的 PE 下设置如下：



3.17 EMC

3.17.1 请遵循硬件设计指南中关于 EMC 的建议

请遵循硬件设计指南中关于 EMC 的建议，器件布局（例如 FXOSC 的布局）、接地（例如 FXOSC 的布局）等建议对改善 EMC 很重要。

3.17.2 使用 PLLFM，可解决 EMC 1.6GHz 左右不通过问题

实际案例：有客户遇到 1.6GHz EMC 未通过问题。通过加入 PLLFM 通过了 EMC 试验。

目前，S32K344 HDQFP EMC 报告在 PLL spread spectrum 频率上有笔误，应修改成 32kHz，以后会升级报告，如下：

Test Case: N° 2

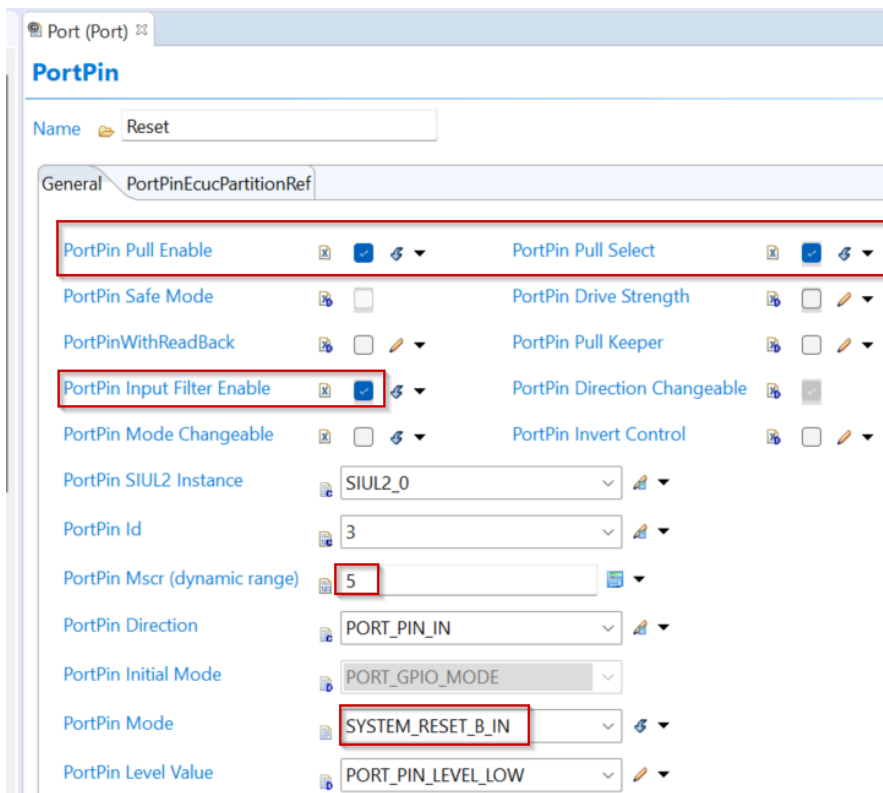
C1-1, PLL spread spectrum enabled (100kHz 1%), all modules on, ext. 40MHz osc, PLL@960MHz, 3V3 supply

3.17.3 使能 Reset 引脚的滤波功能

实际案例：有客户遇到 EMC（ESD）试验未通过问题，通过 EB 配置复位引脚的滤波使能，通过了试验。

通过减小 Reset 引脚的上拉电阻，也可以增强 Reset 引脚的抗干扰能力。

通过改变 Reset 引脚外的滤波电容容值，也会增强 Reset 引脚的抗干扰能力。



3.17.4 通过写一条 DCF 禁用 Reset 引脚的复位功能

实际案例：有客户遇到辐射试验未通过，试验过程发现 Reset 引脚翻转，并有干扰信号。通过写一条 DCF（查看参考手册附件 DCF 说明）禁用 Reset 引脚的复位功能，可通过试验。

在需要 Reset 引脚的复位功能时，可以通过通讯或 OTA，让程序再写入一条 DCF，使能 Reset 引脚的复位功能。

3.17.5 禁止 FXOSC 的自动增益

详见本文档时钟部分 3.2.3 节描述。

加入该措施有助于通过 ESD 等 EMC 试验。

3.17.6 解决 ESD 不通过及 PLL 失锁的常用措施

- 1、禁止 FXOSC 的自动增益（ALC_D）。
- 2、RESET 引脚增加 RC 并且软件配置输入加 Filter。
- 3、晶体品质因数 Q：高 Q 晶体抗扰性更强，更换晶振；
- 4、负载电容：增大电容可提升抗扰性，但会增加启动时间。

- 5、请晶振供应商做匹配，配置功率大的晶振、使用其它品牌的晶振。
- 6、配置 PLL 中 ULKCTL 的 Unlock Range；手动修改 ULKCTL 寄存器（PLLCAL2[ULKCTL]），设置为±33，观测能否通过 ESD 试验。
- 7、晶振布局布线需要遵守 K3 硬件设计指南 3.2 节和晶振厂家设计指南的要求。
例如，晶振 EXTAL 到 MCU 晶振引脚的距离尽量短，晶振到 MCU 的走线与静电泄放路径尽量垂直等等。
- 8、使用 6 层板设计。
- 9、产品使用金属外壳。
- 10、对于未使用 HSE 的 K3 产品，可以设置 DCM 模块的 DCMRWP3 寄存器的 Bit9（DEST_RST9_AS_IPI），将锁相环失锁从默认的复位改成中断。需要用户使能 PLL 失锁中断 SoC_PLL_IRQn (NVIC 212)，并添加 ISR（RTD 不包含该 ISR）。并且在 PLL 失锁中断中将系统时钟切换到 FIRC（通过 MUX_0_CSC[SAFE_SW]=1）。之后再通过时钟初始化或裸代码方式切回 PLL。建议在第一次（或非常有限次）PLL 失锁中断中将系统时钟切换到 FIRC，之后配置 DCMRWP3 为锁相环失锁复位，以免出现时钟丢失，MCU 无法运行的情况。
使用 HSE 的 K3 产品不能使用该方法，因为 HSE 有时钟监控，PLL 失锁后 HSE 会产生复位。
- 11、使用有源晶振。有客户从无源晶振改成有源晶振，通过 ESD 试验。
- 12、将复位源通过 CAN 等方式读出，确认是复位引脚或晶振等哪里产生的复位。
- 13、跨导（GM）值满足 K3 硬件设计指南 3.2 节中的公式（ $GM > 5 \times gmcri$ ），且尽量大。降低晶振频率，推荐使用 8M 晶振。

3.17.7 非 MCU 原因引起 EMC 不通过情况

检查是否因为板卡内其它电路导致的辐射超标。

有客户因为板卡内的防反驱动电路导致辐射超标，该电路后加入 RC 滤波解决了问题，并非 MCU 导致的问题。

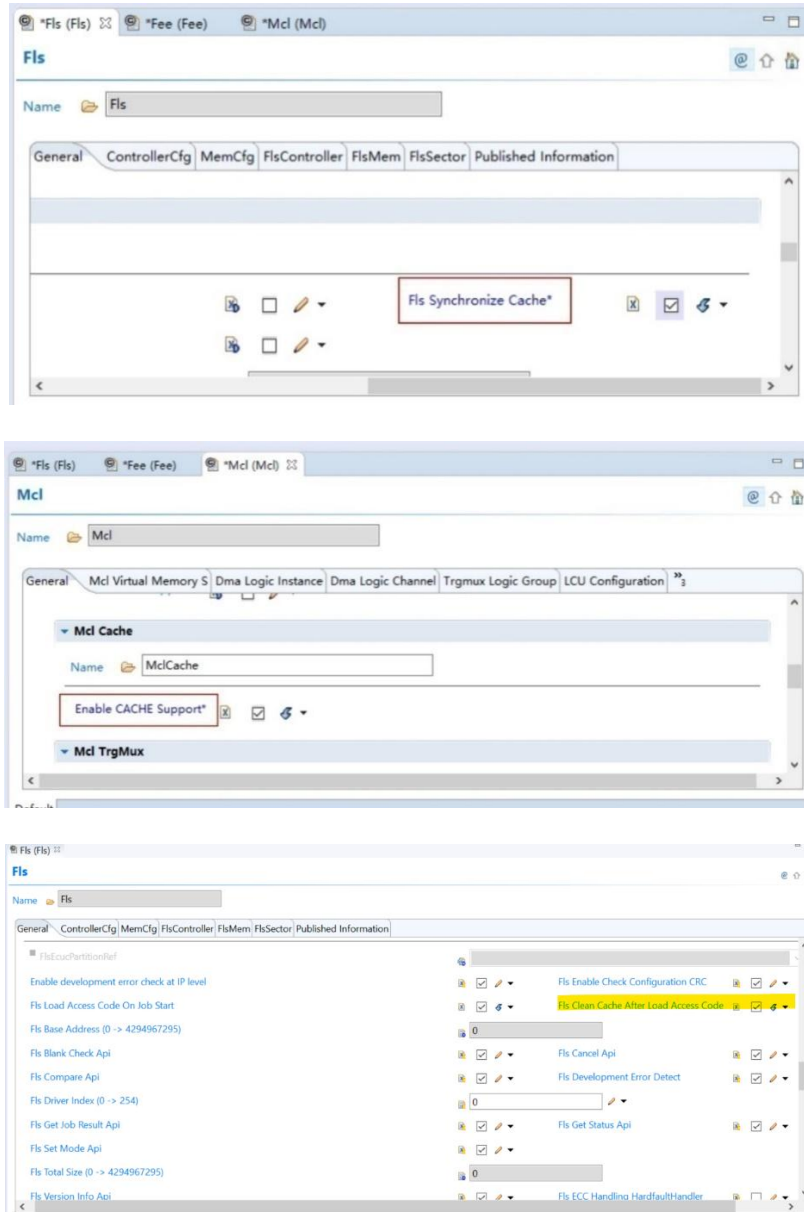
3.18 需检查勘误表 ERRATA

在软件开发过程中，建议遇到问题前阅读勘误表 ERRATA。

在产品量产前，务必检查勘误表。

产品开发中遇到的问题时，检查问题是否与勘误表 ERRATA 描述内容有关。

2、打开了 D-cache，使用 FLS 或模拟 EEPROM（FEE）driver 需要注意数据一致性的问题，这点在 RTD_FLS_UM 中有体现，并给出了三种方法，方法 3 更方便，只需要 EB 里勾选一些配置，见下图：



3、当栈分配在 CACHEABLE RAM，如果用户在自己的应用程序中，本来 CACHE 是关闭的，要打开的话，必须要把栈区空间 Invalidate，再使能 CACHE，否则可能产生数据不同步，产生 HardFault。

3.19.3 将栈放入 DTCM

将栈放入 DTCM 会提升性能。

如果将局部变量传递给 HSE 做参数，推荐将栈放入 DTCM，解决 M7 核和 HSE 核间数据不一致问题。

也可以把栈放入 non cacheable RAM 中，只是会降低一些性能。

若局部变量传递给 HSE 做参数，同时将栈分配到 cacheable RAM 区，

则需要先在调用局部变量做为参数的 HSE 服务前，调用 CACHE 刷新函数（调用 API 是 Mcl_CacheClean），解决数据不一致问题。

3.19.4 使能 FLASH Prefetch

通常不建议打开 Prefetch。

但必须提升性能时，也可以打开 Prefetch。

在开启指令和数据 CACHE 情况下，打开 FLASH Prefetch 可以提升性能。

使能 Prefetch 方法是加入手写代码使能相关寄存器，具体可查看参考手册 FLASH 寄存器。

在未开启指令和数据 CACHE 情况下，不建议打开 FLASH Prefetch，可能出现 Hardfault。

3.19.5 使用 TCM

ITCM 和 DTCM 的性能是最好的，M7 访问该地址零等待。

使用后会提升代码执行速度。建议充分使用 TCM。

ITCM 建议用于：中断矢量表和函数（中断服务程序、频繁使用的函数以及对执行时间要求高的函数等）。

DTCM 建议用于：变量和堆栈（包括 OS 任务栈）等，中断矢量表放入 DTCM 和 ITCM 都可以。注意 DMA 和 HSE 核访问 ITCM/DTCM 需使用后门地址 BACKDOOR。

请联系代理商 FAE 获取 TCM 使用的应用笔记和示例工程。

《S32K3xx Linker File and Startup Code Application Note》

《S32K344_Linkers_File_Modification_Example_Prj.zip》

3.19.6 使用编译器优化

编译器设置代码不优化时，性能会差一些。

建议 MCAL 部分按照 RTD 发布说明设置优化等级，即 GCC 编译器的 Os。

Os 编译的代码量最小，并非性能最好。O1 或 O2 的性能可能会好于 Os。

但谨慎使用 O1 或 O2，因为 RTD 开发验证时使用的是 Os，并未使用 O1/O2 进行验证。

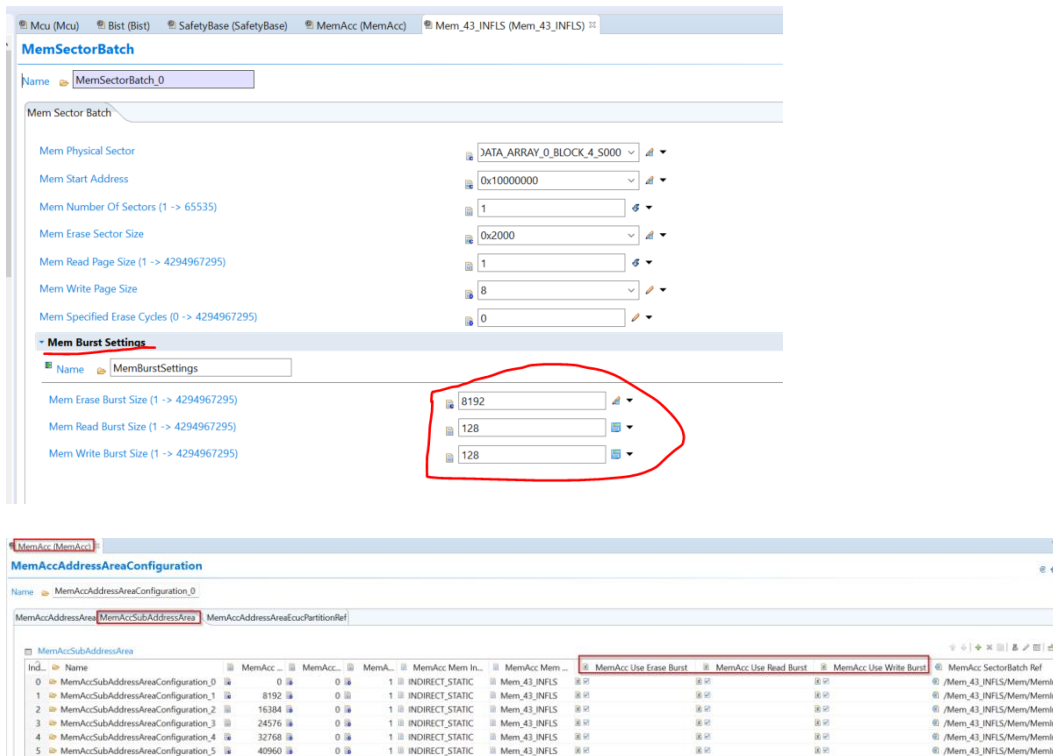
若客户对代码进行充分的测试未发现异常，也可以使用 O1 或 O2。

3.19.7 使用 GHS 或 IAR 编译器

商用编译器编译出的代码执行速度更快。GreenHill 和 IAR 是主流的编译器。如果想使用 S32DS 环境，也可以在 S32DS 上安装 GHS 或 IAR 的插件，实现在 S32DS 上使用 GHS 或 IAR 编译的目的。

3.19.8 Fee 性能提升

使用 RTD4.0.0 HF01/02，RTD5.0.0 以及 RTD6.0.0 版本的 FEE 模块，使能 ICACHE 和 DCACHE，同时设置 BURST SIZE 为 128，间下图：



3.19.9 减小启动时间

1、通过 IVT 使能 PLL，会缩短安全启动时间。

2、将部分或全部 sCheck 检测项放在关机（shutdown）阶段进行。

例如，将 FCCU 初始化，ADC 自检放在启动阶段，将 XRDC、FLASH 和看门狗测试放在关机阶段进行。

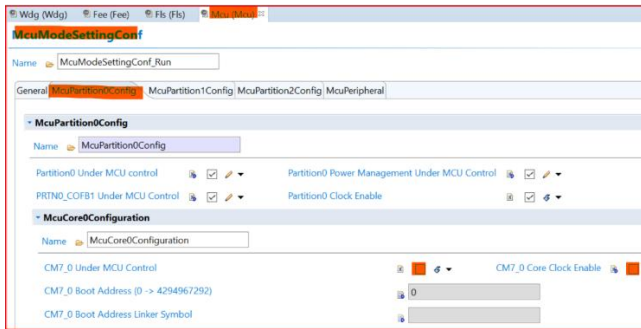
3.20 其它

3.20.1 初始化时间长问题

一、Mcu_SetMode()执行时间长问题

检查是否在 EB 配置或 S32DS 配置时在模式设置中勾选了“CM7_0 Under MCU Control”。

不要勾选，见下图：



若勾选图中的 2 个选项会导致 Mcu_SetMode()执行时间长，程序会执行不必要的超时等待。

单核（K310/K311/K312/K314/K344/K348 等），或多核中的主核（核 0 是主核的情况），都不要勾选“CM7_0 Under MCU Control”。

3.20.2 合理设置超时时间

很多实际案例都是超时时间设置不合理导致。

1、按照数据手册的指标设置超时时间。不能仅根据实测来设置。例如，FLASH 扇区擦除时间，假设典型值是 50ms，最大值是 100ms。此时，就要按照 100ms 或再预留余量来设置超时时间。又如，数据手册中 SIRC 精度是 10%，实测 5%，此时要按 10%或再预留余量来设置超时时间。

2、考虑各种可能的情况。例如，Fee 的换页需要大量时间。功能复位后 HSE_INIT_OK 的置位时间要长于上电复位是该位置位的时间。等等。

3、有些工况可能考虑不到，因此尽量多留余量。例如，看门狗动作时间是 200ms。根据评估，某项等待最大时间是 10ms，那么，可以考虑将超时时间设置为 20ms 或更大，避免考虑不到的工况导致超时。

3.20.3 通过 DSB 和 ISB 解决代码非顺序执行问题

由于 ARM 的特性，代码并非按照顺序执行。要充分意识到这可能带来的问题。

可通过在某条代码前加入 DSB 和 ISB 指令，来保证该条代码在前面代码执行后才会执行。

例如：第一条代码是对变量 a 赋值，第二条代码是对变量 b 赋值，第三条代码是操作寄存器产生软件复位。那么，实际的代码执行可能是第三条代码在第二条代码执行前完成执行，即变量 b 未赋值的情况下产生复位。通过在第三条代码前加入 DSB 和 ISB 指令，则保证第三条指令在第二条执行执行完成后执行。

实际案例：通过裸代码（寄存器操作方式，非 API 调用方式）实现软件功能复位，建议在软件功能复位代码前加入 DSB 和 ISB 指令，否则，可能出现软件功能复位代码前的代码未执行完成的情况下 MCU 已复位。如下是加入 DSB 和 ISB 解决了问题的代码：

```
146 RunStandbyCounter0 = 0x55AA00FF;
147 RunStandbyCounter1 = 0xAAAA5555;
148 *(uint32_t *) (0x20400000) = 0xFF0A00A;
149
150 ASM_KEYWORD("dsb");
151 ASM_KEYWORD("isb");
152
153 *(uint32_t *) (0x402DC004) = MC_ME_MODE_CONF(MC_ME_MODE_CONF_FUNC_RST_MASK);
154 *(uint32_t *) (0x402DC008) = MC_ME_MODE_UPD_MODE_UPD(MC_ME_MODE_UPD_MODE_UPD_MASK);
155
156 *(uint32_t *) (0x402DC000) = MC_ME_CTL_KEY_KEY(MC_ME_CTL_KEY_DIRECT_KEY_U32);
157 *(uint32_t *) (0x402DC000) = MC_ME_CTL_KEY_KEY(MC_ME_CTL_KEY_INVERTED_KEY_U32);
```

3.20.4 OS 使用注意事项

检查是否 OS 进行了中断初始化，同时应用初始化代码也调用了 Platform_Init()。

若是，建议保留 OS 进行的中断初始化，屏蔽掉应用初始化代码调用的 Platform_Init()。

3.20.5 建议每个项目都配备 EVB

无论项目处于研发阶段，还是量产阶段，都建议采购/申请一块 EVB。

配备 EVB 会有很多好处，例如：

- 1、排查产品硬件是否存在问题。
- 2、加速软件开发。
- 3、便于和 FAE 共同定位问题。因为和 FAE 使用相同硬件复现问题，能方便问题的解决。

3.20.6 建议使用新版本手册

检查是否使用新版手册。包括 K3 参考手册，数据手册，HSE 用户手册等等。

因为旧版本手册会有一个描述错误，或描述不清的问题。

3.20.7 焊接问题

检查是否存在虚焊、短路焊接等问题。

实际案例：客户反馈 SPI 通讯问题。现场测试 MCU 的供电引脚，发现没电压。用手按一下 MCU，电压正常。重新焊接，MCU 工作正常。

3.20.8 测量 MCU 引脚输出电平是否正常

检查 MCU 的各个输出引脚的电平是否在芯片数据手册的电压范围。

例如，使用 PTE13 控制 V15 的三极管基极，若测试 PTE13 的输出电压为 0 或 VCC，则说明有问题。又如，测得 V15 的电压低于 1.4V，也说明有问题。

3.20.9 看门狗模块调试

有 2 种调试方法，

- 1: 不带调试器调试，通过 LED 闪烁或 attache 方法观测看门狗功能是否正常。
- 2: 带调试器时，需使能冻结模式，并且也需要及时喂狗。若带调试器测试看门狗功能异常，可通过不带调试器的方法调试，不带调试器时看门狗功能正常即可。

3.20.10 逻辑分析仪测试造成的误导

当用逻辑分析仪测量信号异常时，建议使用示波器测试一下。排除仪器的影响。

因为有时逻辑分析仪测试异常，但示波器测试没问题，以示波器测试为准。

同理，若 MCU 通讯异常，而逻辑分析仪测试信号正常，也建议使用示波器测试一下，以示波器为准。

用示波器测量毛刺或时间短的脉冲，建议使用单次捕捉，否则由于时间窗口大，可能捕捉不到。

3.20.11 周期任务和中断都操作某个外设的问题

检查是否存在周期任务和中断都对某个外设进行启动操作。

若存在，需谨慎评估是否存在问题。

在出现问题时，尝试禁止中断中对该外设操作，观测问题是否消失。然后，采取合适措施解决问题。

3.20.12 Bootloader 跳转 APP 失败问题

检查 Bootloader 跳转 APP 地址是否正确。

NXP 提供的 Bootloader 示例工程是基于 RTD2.0.0 的。

该版本 RTD 的链接文件中".boot_header"（IVT）段是 4K（4096）对齐。

RTD3.0.0 及以上版本的链接文件中".boot_header"段是 8K（8192）、2K（2048）或其它大小对齐。因此，当 APP 使用了 RTD3.0.0 及以上版本的链接文件时，跳转到 APP 时需要从 RTD2.0.0 示例工程 APPStartAddr + 0x1000 修改为 APPStartAddr + 0x2000（8K 对齐情况）或其它值（例如 2K 对齐）。

也可以修改 APP 的链接文件，将".boot_header"（IVT）段改成 4K（4096）对齐。

还可以使用 IVT 的 CM7_0_VTOR_ADDR（地址偏移 0x0C）。

若 APP 使用 RTD2.0.~2.0.3，不存在该注意事项。

如下是链接文件中".boot_header"（IVT）段对齐的位置：

```
.pflash :  
{  
    KEEP*(.boot_header)  
    . = ALIGN(8192); /* The minimum erase size of C40 Flash IP is 8kb */  
    __text_start = .;
```

3.20.13 通过 NXP 社区获取示例工程和问题解决方案

建议注册 NXP 社区账号。NXP 社区上有很多常见问题的讨论，也有很多示例工程，可解决很多问题。

建议遇到问题时，先在社区搜索一下，可能会找到问题答案，快速解决问题。

通过访问如下链接或网址，获取典型示例工程：

[S32K Knowledge Base - NXP Community](#)

<https://community.nxp.com/t5/S32K-Knowledge-Base/tkb-p/S32K@tkb>

3.20.14 看门狗的影响

看门狗包括外部看门狗和芯片内部看门狗。

外部看门狗包括 SBC 芯片内部的看门狗，以及其它类型的外部看门狗。

在调试和产品实际运行中，都需要考虑看门狗的影响。

在调试时，建议关闭外部看门狗和内部看门狗。

产品程序上要及时喂狗，考虑异常情况下喂狗是否及时。

例如，Fee 初始化，或 Fee 写，会有换页的情况，换页时时间较长。

另外，烧录程序和调试时，SBC 需要设置为开发模式，以免 SBC 拉低复位信号。

3.20.15 EB 软件由于多个 link 文件导致生成配置文件问题

在 EB 的 links 文件夹内建议只保留一个版本 RTD 的 link 文件，否则生成的配置文件可能使用非预期的 RTD。例如，links 文件夹内含义 RTD4.0.0 和 RTD5.0.0 的 link 文件，那么生成配置文件不能保证一定使用 RTD5.0.0 的文件。

有客户遇到这种问题。

4. 信息安全(HSE)部分

4.1 建议使用最新版本 HSE 固件并升级 sBAF

建议使用最新的 0.2.55 版本，若未使用，建议升级。

sBAF 建议使用 0.15.0 版本，若未使用，建议升级。

旧版本存在一些 Bug（例如 4.26 节介绍的 HSE 无响应问题），新版本修改 Bug 的同时也提升了性能。详见 0.2.55 版本 HSE 固件的发布说明，以 K344 为例，文件为：

《HSE_FW_S32K344_0_2_55_0_ReleaseNotes.pdf》。

注意 sBAF 和 HSE 固件选用兼容版本，否则可能出现异常。

在 HSE 参考手册中有 sBAF 和 HSE 固件兼容性说明。若使用旧版 sBAF 芯片，同时使用新版 HSE 固件，可能存在不兼容问题。同理，使用新版 sBAF 芯片，同时使用旧版本 HSE 固件，也可能存在不兼容问题。

4.2 HSE 固件安装注意事项

一、HSE 固件安装的方法

1、传统的通过写 UTEST 区 0x1B000000 未非 FF 值的方法。

2、通过使能 IVT 中 BCW Bit9 的方法。

方法 2 不需写 UTEST 区 0x1B000000，但需要保证 sBAF 版本为 0.15.0 及以后版本。

二、HSE 固件安装工程建议使用 NXP 应用工程师开发的基于 RTD4.0.0 的 HSE lib2.0 的固件安装工程（请使用 2025 年 12 月 30 日版本 s32k3_hse_lib_rtd400hf01_20251230.7z）。

这个工程考虑了固件安装的注意事项，经过了大量验证。

三、HSE 固件安装工程在时钟初始化前需要等待 HSE 核进入休眠 WFI。

实现代码详见 4.5 节。

四、HSE 固件安装工程在 UTEST 区写入数据前需要等待 HSE 核进入休眠 WFI。

实现代码同三，若时钟初始化前已添加等待 WFI 代码，此处不需再重复添加。

五、写 UTEST 区需要注意 FLASH 读同时写（RWW）问题。UTEST 区与 BLOCK0 属于同一个 BLOCK。因此，需要将 FLASH 底层驱动代码放在 BLOCK1-3/RAM/TCM。

六、若 HSE 时钟设置为 Option A（80M,K314/324/344/322/341/342）或 Option B（60M,K310/311/312），**不需要**写 DCF 时钟配置。其它情况，需要写入一条 DCF 时钟配置。见 K3 参考手册附件的《S32K3xx_DCF_clients》，如下：

HSE_CLK_MODE_AND_GSKT_CTRL	30-29	hse Clock mode selection and Bypass HSE IAHB gasket control	These bits control the HSE clock control and HSE IAHB gasket operation in different clocking options as listed in field description.	The values in different clocking options are as listed below. 00 = Ratio of 1:2 in between HSE subsystem programming interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK). HSE IAHB gasket enabled. See the 'Clocking Overview' section in the device Reference Manual for the clock frequency ratios and the gasket configurations. 01 = Ratio of 1:2 in between HSE subsystem programming interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK). HSE IAHB gasket bypass. See the 'Clocking Overview' section in the device Reference Manual for the clock frequency ratios and the gasket configurations. 1x = Ratio of 1:4 in between HSE subsystem programming interface clock (AIPS_SLOW_CLK) and HSE module clock (HSE_CLK). HSE IAHB gasket enabled.
----------------------------	-------	---	--	---

图 1

具体说明：

1、当 K310/311/312 设置 HSE 时钟为 120M 时，需要写一条 DCF，将上图 30-29bit 写入 11 或 10。

具体实现可参考本节第二条提到的 HSE lib2.0 固件安装工程，或 0.2.6.0 版本 HSE 固件安装工程中的函数 ChangeDcfClockOption()。若不写 DCF，可能会导致 Hardfault。

2、当 K310/311/312 设置 HSE 时钟为 60M 时，不需要写 DCF。

3、**K328/338/348/358 需要写 DCF（无论哪种时钟配置）**，将上图 30-29bit 写入 11 或 10（OPTIONA+配置）。若不写，可能出现 HSE 固件被擦除问题。

写入后，并且在 IVT 中使能了 PLL，安全启动时 HSE 时钟为 120M，安全启动后，M7 核时钟为 240M，即 OPTIONA+配置。详见 HSE 参考手册。图 1（现有最新版参考手册的描述）描述有误。现有参考手册容易造成误导，让我们误以为不写 DCF（或 DCF 中上图 30-29bit 写入 00）时，是 OPTIONA+配置。

七、通过功能复位实现固件安装，不要通过上电复位实现，否则可能固件安装失败。

八、固件安装过程需保证电源文档，无异常复位打断固件安装。

4.3 HSE 固件安装后，S32DS 烧录程序时需选择对应的 FLASH 算法文件

使用 PE Micro 调试器，安装完 Full memory 类型的 HSE 固件后，S32DS 程序烧录页面 ->PEmicroDebugger->Advanced Options 页面下，需选择 xxx_hse_enabled.arp，安装完 AB SWAP 类型的 HSE 固件后，需选择 xxx_ab_swap.arp。

4.4 建议参考 NXP 中国应用工程师团队开发的 HSE 库 2.0

HSE Lib2.0 包括基于 RTD3.0.0P07 版本和 RTD4.0.0HF01 版本的工程，提供了大量的实用的 HSE 例程，方便进行 HSE 开发。

建议使用**最新版本**的基于 RTD4.0.0HF01 的工程，因为该工程在不断更新完善。

目前最新版本是 2025 年 12 月 30 日版本，文件名称如下：

[s32k3_hse_lib_rtd400hf01_20251230.7z](#)

工程中也包括很多**根据实际经验得到的提示**，很多问题从提示中能得到解答，请认真阅读工程中的注释说明。

工程中会将某些函数放入 RAM/ITCM，某些变量放入 non cacheable RAM/DTCM。需参考遵循，否则会有问题。

例如，写 UTEST 区（或调用 HSE 进行生命周期演进），就需要将几个函数放入 RAM，否则存在对同一个 FLASH BLOCK 读同时写的异常问题。

在 HSE LIB2.0 文件夹内含有

- 1、《Doc》。介绍如何使用以及 ADKP 调试。使用前需阅读。
- 2、《RTD_LLD_NON_AUTOSAR_PATCH_400》。这是与 RTD 源代码不同的地方，基于源代码进行了修改，解决一些 Bug，将 FLASH 驱动代码映射到 TCM。客户实际项目也需使用这些代码文件，而不是 RTD 默认代码文件。
- 3、HseLib_GenProj_K312_Rtd400/ HseLib_GenProj_K344_Rtd400/ HseLib_GenProj_K358_Rtd400

通常不需要对这些工程修改配置。如果修改了这些工程的配置，并且生成了代码，则需要将《RTD_LLD_NON_AUTOSAR_PATCH_400》中的文件替换到这个工程的《generate》和《RTD》文件夹内。

HSE LIB2.0 使用的常见问题：

- (1) 直接修改 HSE Lib2.0 示例工程中的宏定义，将 S32K344 切换至 S32K314，出现 Bus fault;

示例工程比较复杂，出现问题不宜排查，因此不建议使用这种方式直接进行移植。
推荐新建 S32K314 工程，并逐步移植代码。

- (2) 新建工程执行 Flash 操作时出现 Hard fault;

为避免对 Flash 操作时出现读写冲突，HSE Lib2.0 修改了 RTD 代码（C40_Ip.c 以及 C40_Ip.h），将部分函数重定位至 ITCM 中。此外，generate/include/C40_Ip_Cfg.h 中的 FLS_MAX_VIRTUAL_SECTOR 数量有变动。参考示例工程代码中这三个文件进行修改后，可以正常运行。

- (3) AB Swap 安全启动操作流程错误;

AB Swap 在执行 AB 面翻转之前要保证另一面有可执行代码。HSE Lib2.0 示例工程中有 AB 面翻转操作，在使能相关宏定义之前需要进行确认。

(4) CR 配置错误导致无法正常跳转至 APP;

如果对 Linker 文件进行了修改（增加了 APP），需要相应修改 CR 表的 APP 启动地址，否则会导致配置完安全启动后无法正常跳转。

4.5 未安装 HSE 固件情况下，时钟初始化前必须检查 WFI 标志

未安装 HSE 固件情况下，时钟初始化前需要等待 HSE 核进入 WFI。

WFI 标志判断加入代码：

```
while(0 == (IP_MC_ME->PRTN0_CORE2_STAT & MC_ME_PRTN0_CORE0_STAT_WFI_MASK))
```

或用裸代码方式：

```
#define WFI_BIT (1U << 31U)
```

```
while ( FALSE == ((* (uint32_t *) (0x402DC188)) & WFI_BIT) )
```

```
{
```

```
    ;
```

```
}
```

复位信号释放到 WFI 置位时间约为 18ms。

4.6 未安装 HSE 固件情况下，FLASH 擦写前必须检查 WFI 标志

未安装 HSE 固件情况下，时钟初始化前需要等待 HSE 核进入 WFI。

若时钟初始化前，对 FLASH（包括 UTEST）进行擦除/写操作，那么需要在操作 FLASH 前等待 HSE 核进入 WFI。

WFI 标志判断加入代码：

```
while(0 == (IP_MC_ME->PRTN0_CORE2_STAT & MC_ME_PRTN0_CORE0_STAT_WFI_MASK))
```

或用裸代码方式：

```
#define WFI_BIT (1U << 31U)
```

```
while ( FALSE == ((* (uint32_t *) (0x402DC188)) & WFI_BIT) )
```

```
{  
    ;  
}
```

4.7 安装 HSE 固件情况下，时钟初始化前必须检查 HSE_STATUS_INIT_OK

检查安装了 HSE 固件的工程，在时钟初始化前是否已等待 HSE_STATUS_INIT_OK 标志置位。

在 Bootloader 和 APP 的时钟初始化前都需要检查该标志。

加入代码：

```
#define HSE_STATUS_INIT_OK_BIT (1U << 24U)
```

```
while ( FALSE == HSE_CheckStatus(HSE_STATUS_INIT_OK) )
```

或用裸代码方式：

```
while ( FALSE == ((* (uint32_t *) (0x4038C104)) & HSE_STATUS_INIT_OK_BIT) )
```

```
{  
    ;  
}
```

因为 HSE_STATUS_INIT_OK 标志置位前，HSE 核正在进行初始化操作，此时若 M7 核初始化时钟，启用 PLL，会导致 HSE 核工作频率变化，从 24M 变成 80M，进而可能导致 HSE 初始化异常，HSE 看门狗动作，进而芯片复位。

不加入该等待不一定会出现问题。

在 sBAF (0.15.0) 和 HSE 固件 (0.2.40.0) 下 (对于 K344, DCF 保持默认, 即 Option A, HSE 时钟 80M) 测得 HSE_STATUS_INIT_OK 标志置位时间约为 18.05ms, WFI 标志置位时间约为 18.1ms。

对于 K312,

1、DCF 保持默认(即 Option B, HSE 时钟 60M) , 测得 HSE_STATUS_INIT_OK 标志置位时间为 18ms。

2、写入 DCF 非默认值(即 Option B, HSE 时钟 120M)，测得 HSE_STATUS_INIT_OK 标志置位时间为 28.7ms。

因此，对于 K312，若 HSE 时钟使用 120M,需要在 UTEST 区写入 DCF（参考 HSE LIB2.0 或固件安装过程，不写可能出现复位问题）。

若使用安全启动，并且在安全启动中使用 PLL，那么，需要写 0x1B000050 的 FXOSC 配置（注意示例工程是写入 16M 外部晶振对应值，若使用其它外部晶振，需修改为对应值，UTEST 区只能写一次，若写错误，无法修改）。

若 HSE 时钟使用 60M，不需要写 DCF。若写入后，HSE_STATUS_INIT_OK 会增加约 10ms。由于时钟初始化前需要等待 HSE_STATUS_INIT_OK，所以启动实际也可能增加 10ms（启动代码中变量初始化时间短），也可能增加较少（启动代码中变量初始化时间长）。

4.8 安装 HSE 固件情况下，FLASH 擦写前必须检查 GPR3 寄存器

若对 FLASH 擦除/写操作是在 MCU 启动后数百毫秒后，通常 HSE 核已进入休眠，但判断一下更可靠，建议判断一下。

若已**安装了**HSE 固件，对 FLASH（包括 UTEST）进行擦除/写操作前判断 HSE 是否占用该块 BLOCK。即通过读取 HSE GPR Register 3（地址 0x4039C028）判断，详见 HSE 参考手册。

注意：BLOCK0 和 UTEST 同属一个 BLOCK。若 HSE 对 UTEST 读操作，那么不仅 M7 核不要对 UTEST 区进行写操作，也不要对 BLOCK0 进行擦写操作。

RTD6.0.0 代码加入了信号量机制，可以不用判断 HSE GPR3 寄存器。

4.9 功能/破坏性复位后 HSE_STATUS_INIT_OK 置位时间长问题

上电复位 HSE_STATUS_INIT_OK 置位时间最短。

功能复位和破坏性复位后的 HSE_STATUS_INIT_OK 置位时间较长见下表：

测量条件:

- 启动类型: 非安全启动
- Hase-bi Fu 版本: **2.40.0**, 全主
- SBAF 版本: **0.15.0**
- 设备: **S32K312**

	POR	破坏性重置	功能复位
时钟选项“00”	18毫秒	33毫秒	33毫秒
时钟选项“1x”	28.4毫秒	54 毫秒	54 毫秒

4.10 随机数服务调用前检查随机数初始化 OK 标志

若随机数初始化未完成, 调用随机数服务可能会失败。

4.11 随机数服务首次调用耗时长问题

芯片启动后首次调用随机数获取服务耗时较长 (约 40ms)。

可初始化阶段异步模式调用一次随机数获取服务, 不需等待, 继续执行其它函数, 本次调用的随机数获取结果也不需使用, 这样, 在周期任务调用随机数服务已不是首次调用, 故耗时很短。

可参考 HSE LIB2.0 的示例做法。

4.12 ADKP 和生命周期演进注意问题

需要使用功能复位, 不要使用上电复位。

通过上电复位完成 ADKP 和生命周期演进时, 可能出现 UTEST 区擦写不完整问题。

需要将如下 4 个函数的**定义**和**声明**放入 RAM 或 ITCM 中 (**注意: 定义和声明都要放**)

- 1、Hse_Ip_ServiceRequest
- 2、OsIf_GetCounter
- 3、Mu_Ip_IsResponseReady
- 4、OsIf_GetElapsed

原因是避免对 Block0 (0x00400000~0x004FFFFFF 这 1M 空间+UTEST 区) 读同时写的问题。

调用 HSE 服务进行 ADKP 写入和进行生命周期演进时，HSE 核会写 UTEST 区，而此时 M7 核调用如上 4 个函数进行等待，故需将如上 4 个函数放入 RAM 或 ITCM 中，避免 RWW 问题。

同时，需要将 RTD 代码中 Hse_Ip.c 中的数组 Hse_Ip_apMuBase 前的 const 去掉。

否则可能出现对 FLASH 同一个 BLOCK 读同时写的问题。因为不去掉 const 时，该数组存放在 FLASH 中，对其访问会读 FLASH。

不能通过某些板卡未去掉 const 没问题，就认为其它板卡也没问题，是有隐患的。

有客户使用 K312 遇到该问题。

安装 HSE 和不安装 HSE 两种情况下，ADKP 写入到 UTEST 区的地址不同。

未安装 HSE 情况下，调试口加密可参考如下链接：<https://community.nxp.com/t5/S32K-Knowledge-Base/S32K3-Restrict-the-debug-access-with-a-password-when-HSE-is-not/ta-p/1735073>

4.13 D-CACHE 打开导致 HSE 服务失败问题

HSE 服务调用失败，特别是提示参数无效，很可能与 DCACHE 有关。

可通过关闭 DATA CACHE，观测问题是否消失，进而确认是否由于 RAM 的 CACHE 属性设置不当造成。

关闭 DATA CACHE 是确认 HSE 问题的常规方法。

关闭 DATA CACHE 可通过设置 S32DS 属性删掉“D_CACHE_ENABLE”或将其修改成其它名字的方法，也可以屏蔽 system.c 中 DATA CACHE 初始化代码，或调用 API 关闭 CACHE。

解决办法：若是 DCACHE 导致的问题，将传递给 HSE 的参数放入 NON-CACHEABLE RAM 或 DTCM 即可。可参考 HSE LIB2.0。

4.14 局部变量传递给 HSE 的注意事项

若局部变量传递给 HSE 做参数，同时将栈分配到 cacheable RAM 区，

则需要先在调用局部变量做为参数的 HSE 服务前，调用 CACHE 刷新函数（调用 API 是 Mcl_CacheClean），解决数据不一致问题。

建议将栈放入 DTCM，即规避了该问题，也提升代码执行速度。参考 HSE LIB2.0 的做法。

4.15 生命周期演进前需格式化 Key 目录

检查生命周期演进前，是否已格式化密钥目录（KEY catalog）。

若未格式化，会演进失败，演进服务返回值为 Not Allow。

4.16 通过寄存器定位问题

遇到 HSE 相关问题，请检查如下寄存器，用于定位问题。

结合手册，分析这些寄存器提示的信息。

可读取未出现问题板卡的如下寄存器，对比差异。

1. MU0_B FSR
2. MU0_B GSR
3. FLASH MCRS
4. DCMRWP1
5. MC_RGM
6. HSE_GPR_3 (0x4039C028)
7. CONFIGURATION_GPR
8. IVT content

4.17 使用空闲的 MU Channel

检查使用 MU 的通道前，是否通过调用 API 获取可用通道号。

若不查阅可用通道，而是用固定通道，可能出现问题。

HSE 服务使用的 MU Channel 需要处于 Free 状态。

实际案例：使用某个 MU Channel 进行某些 HSE 服务，该 Channel 未释放前，又使用该 Channel 申请超级用户权限，结果失败。

4.18 建议申请超级用户权限

部分 HSE 服务需要具备超级用户权限才能进行。建议在客户开发阶段（customer delivery）导入用于申请超级用户权限的密钥（Key）。

另外，要导入密钥的拥有者（Owner）属性与超级用户权限需一致。

实际案例：芯片的生命周期处于 OEM 阶段，想要导入一个 Key，导入失败。

此时芯片具备 OEM 超级用户权限，但这个需要导入的 Key handle 属性 owner 是 customer（HSE_KEY_OWNER_CUST），此时必须申请 customer 超级用户权限（SU=CUST_SUPER_USER）。

4.19 生命周期演进后无法 AB 面切换（OTA）问题

若使用 0.2.40.0 版本及之前版本的 HSE 固件，生命周期演进后，若需要 AB 面切换进行 OTA，必须先申请超级用户权限。

申请超级用户权限必须在 CUST_DEL 阶段导入授权验证密钥。

所以，当在 CUST_DEL 阶段未导入授权验证密钥，并且演进生命周期到 OEM/IN FIELD，则无法 AB 面 OTA。

使用 0.2.55.0 版本 HSE 固件时，AB 面切换 OTA 升级时，不需要超级用户权限。

但是，仍建议客户在 CUST_DEL 阶段导入授权验证密钥，这样就可以在 OEM/IN FIELD 阶段申请获取超级用户权限。因为部分 HSE 服务是需要超级用户权限的，例如导入 KEY。

可查看 HSE 参考手册了解哪些服务需要超级用户权限。

4.20 HSE 固件损坏或被擦除的原因

- 1、HSE 固件安装或更新时，电源异常（例如掉电，电源波动严重）、有内部或外部复位产生。
- 2、时钟初始化前未等待 HSE_STATUS_INIT_OK。
- 3、XRDC 初始化前未等待 HSE_STATUS_INIT_OK。
- 4、BIST 前未等待 HSE_STATUS_INIT_OK。

实际案例：BIST 前未等待 HSE_STATUS_INIT_OK，调用 AB SWAP，然后调用复位，之后 HSE 会进行 AB 面翻转和 HSE 固件备份（0.2.40 版本及之前版本），同时，M7 核进行 BIST，完成 BIST 后会产生复位，复位会打断 HSE 固件备份。

- 5、时钟配置未严格遵循 K3 参考手册的要求的几种配置。

4.21 HSE 信息安全标准认证情况

超过 EVIT FULL

遵循 ISO21434 并且被 SESIP 和 NIST CAVP 认证。

4.22 RTD3.0.0 和 RTD4.0.0 的 IVT 错误问题

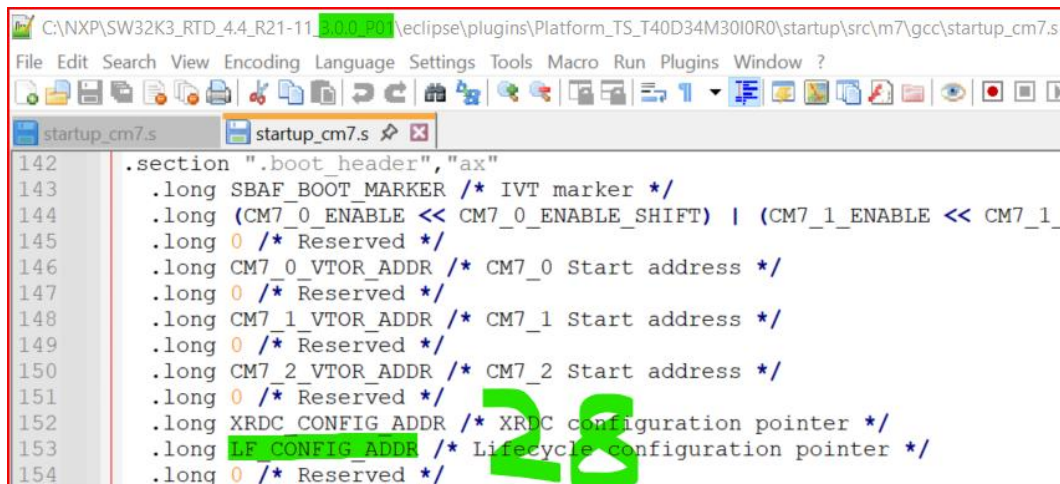
在这两个版本 RTD 的启动代码中，生命周期配置字地址（LF_CONFIG_ADDR）位于 IVT（boot header）的偏移地址是 0x28（见图 1）。这是错误的。

生命周期配置字地址应该位于 IVT 中的偏移地址是 0x24。在 K3 参考手册有说明。

RTD5.0.0 修复了这个问题，见图 2。RTD2.0.x 也没有这个问题。

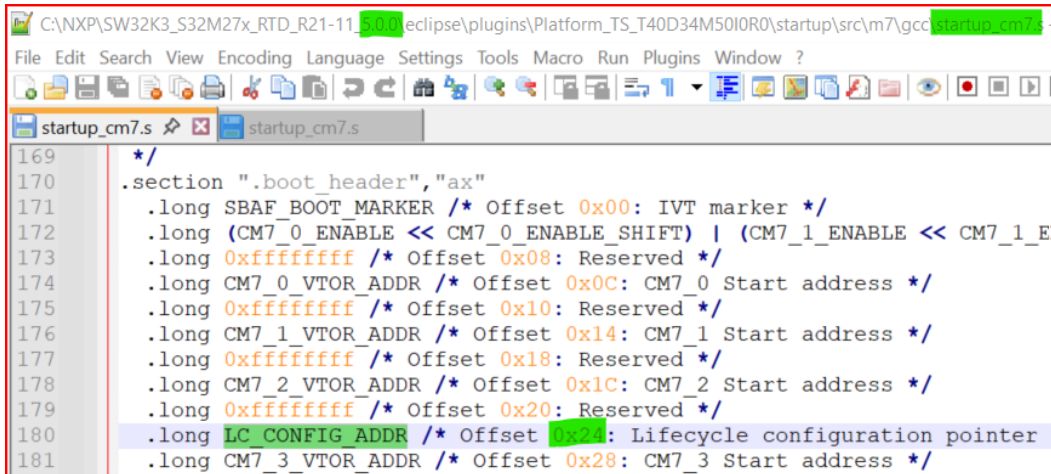
注意会对如下两个应用产生影响：

- 1、在未安装 HSE 固件，且需要进行调试口加密。
- 2、Basic secure boot。



```
142 .section ".boot_header", "ax"
143     .long SBAF_BOOT_MARKER /* IVT marker */
144     .long (CM7_0_ENABLE << CM7_0_ENABLE_SHIFT) | (CM7_1_ENABLE << CM7_1_
145     .long 0 /* Reserved */
146     .long CM7_0_VTOR_ADDR /* CM7_0 Start address */
147     .long 0 /* Reserved */
148     .long CM7_1_VTOR_ADDR /* CM7_1 Start address */
149     .long 0 /* Reserved */
150     .long CM7_2_VTOR_ADDR /* CM7_2 Start address */
151     .long 0 /* Reserved */
152     .long XRDC_CONFIG_ADDR /* XRDC configuration pointer */
153     .long LF_CONFIG_ADDR /* Lifecycle configuration pointer */
154     .long 0 /* Reserved */
```

图 1



```
169 */
170 .section ".boot_header", "ax"
171 .long SBAF_BOOT_MARKER /* Offset 0x00: IVT marker */
172 .long (CM7_0_ENABLE << CM7_0_ENABLE_SHIFT) | (CM7_1_ENABLE << CM7_1_E
173 .long 0xffffffff /* Offset 0x08: Reserved */
174 .long CM7_0_VTOR_ADDR /* Offset 0x0C: CM7_0 Start address */
175 .long 0xffffffff /* Offset 0x10: Reserved */
176 .long CM7_1_VTOR_ADDR /* Offset 0x14: CM7_1 Start address */
177 .long 0xffffffff /* Offset 0x18: Reserved */
178 .long CM7_2_VTOR_ADDR /* Offset 0x1C: CM7_2 Start address */
179 .long 0xffffffff /* Offset 0x20: Reserved */
180 .long LC_CONFIG_ADDR /* Offset 0x24: Lifecycle configuration pointer
181 .long CM7_3_VTOR_ADDR /* Offset 0x28: CM7_3 Start address */
```

图 2

4.23 安装 HSE 固件后时钟初始化进入 Hardfault 问题

若使用 RTD2.0.0~RTD2.0.3，在安装 HSE 固件后不要勾选 Mcu->时钟配置页面->McuFIRC 页面下的“FIRC under MCU control”。

RTD4.0.0 以后，可勾选。

4.24 使用上电复位实现 HSE 功能导致的问题

例如 HSE 固件安装、sBAF 升级、ADKP 写入、生命周期演进、OTA AB 面切换等，手册要求通过调用功能复位实现。

如果没有使用功能复位，而是通过上电复位实现，有可能出现问题，不推荐。

因为上电复位方式，在电源电压不稳定时 sBAF 就开始进行这些功能。这些功能需要擦写 FLASH，而擦写 FLASH 需要电流较大，有可能导致电源电压波动，进而 FLASH 擦写失败。

例如，通过上电复位实现生命周期演进，有可能 UTEST 区只写一部分生命周期特征字，进行生命周期异常。

4.25 OTA 过程喂狗问题

若使用 0.2.55.0 版本 HSE 固件，OTA 需要时间 100ms 左右，即调用 AB SWAP，然后调用软件功能复位，大约 100ms 后 HSE_STATUS_INIT_OK 会置位。

若使用 0.2.40.0 版本 HSE 固件，OTA 需要时间 1s 左右，即调用 AB SWAP，然后调用软件功能复位，大约 1s 后 HSE_STATUS_INIT_OK 会置位。

所以，推荐使用 0.2.55.0 版本 HSE 固件。

无论 0.2.55.0，还是 0.2.40.0 版本的 HSE AB SWAP 类型固件，进行 HSE 固件更新都需要 1s 左右。

若板卡使用 FS26，则有一个看门狗喂狗的问题。因为 FS26 需要 256ms 内喂狗，而 HSE 固件升级需要 1s 左右，0.2.40 版本固件 AB SWAP 需要 1s 左右，在这 1s 内进行时钟初始化可能出现问题。但在时钟初始化前，可以进行 SPI 初始化，进而对 FS26 进行喂狗操作。

NXP 应用工程师会在 2025 年 9 月底提供时钟初始化前 SPI 初始化和喂狗的示例代码，解决喂狗问题。可联系 NXP FAE 或代理商 FAE 获取参考代码。

若使用安全启动，并且：

- 1、安全启动后的时钟初始化不修改 HSE 时钟和 AIPS_SLOW_CLOCK，
- 2、在 HSE_STATUS_INIT_OK 置位前不操作 DATA FLASH，
- 3、修改 EB/S32DS CT 的一个勾选项配置，

那么，可以不等待 HSE_STATUS_INIT_OK 置位。

具体实现方案请联系 NXP FAE 或代理商 FAE。

最新版本的 HSE LIB2.0 包括该实现的 Demo。

若板卡使用其它 SBC 或外部看门狗，也需考虑喂狗问题。

若 HSE 固件升级时，没有及时喂狗，可能导致 HSE 固件被擦除。

4.26 0.2.40 及之前版本存在极小概率 HSE 服务无响应问题

0.2.40 及之前版本 HSE 固件存在一个 Bug，即：HSE 服务调用时刻刚好是 HSE 5 秒周期喂狗时刻，可能出现 HSE 服务无响应问题，只有通过 MCU 复位才能再次使用 HSE。

在 0.2.55.0 版本 HSE 固件的 Release note 有此说明，如下：

3 Changes List

3.1 Changes in 0.2.55.0

Added

- Key verify service extended to support asymmetric keys.
- Check for data flash write operation and acquire lock before accessing KHTT.
- TRNG initialization from UTEST.

Updated

- Downgraded User access to HSE_SRV_ID_ACTIVATE_PASSIVE_BLOCK from Super User to Normal User.
- HSE debug codes have been extended.
- To keep the compatibility, the support for SHA384 and SHA512 for key verify has been added back.
- Added details in the interface of HSE_SECURE_RECOVERY_CONFIG_ATTR_ID.
- Use the same item size for both Config Data and KHTT.
- In ABSwap reduce HSE init time after Partition swap and reset by comparing Hash of Active and Passive firmware.
- Updated the CRC verification process so that it does not fail when reading KHTT from NVM during loading NVM data in HSE init.

Issues Fixed

- Hardware Protection for data flash shows incorrect value for backup disable firmware (reduced data flash protection size to 64KB in case of backup disable).
- RESET SOC sanction not working in post boot.
- Key Drive Verification for HSE_HASH_ALGO_SHA2_384 and HSE_HASH_ALGO_SHA2_512 Expectation Error shows General Error.
- Mismatch between pSmrDest and pSmrSize alignment check condition and interface comment.
- SMR update as it did not check if pPassReset was in range and allowed it if outside SMR region.
- State of the Sensor was changing from ARM/DISARM state to Unused State using HSE_SENSOR_DISARMING_ON_STARTUP_ATTR_ID and gives no error.
- Zero length input for ECDSA sign that caused fault attack is fixed.
- Prevent race condition in WDT idle/running state change.

原因说明：

HSE 核每 5 秒进行一次喂狗操作，喂狗前没有关闭中断（0.2.40 及之前版本 HSE 固件），在喂狗过程的几个寄存器操作之间，若 M7 核调用 HSE 服务产生中断，可能会导致 HSE 核异常，只能通过复位恢复。

由于喂狗操作的完成时间非常短，所以，刚好 M7 核在这么短的时间里调用 HSE 服务的概率非常低，而且并非在喂狗操作内任意时刻被中断打断都会出问题。

有几个客户遇到了该问题。他们都是周期性调用 HSE 服务，刚好在某个周期调用 HSE 服务是 HSE 核 5 秒周期喂狗时刻，触发了该问题。

非周期性调用 HSE 服务的项目，暂无客户反馈遇到此问题。

建议使用 0.2.55 版本 HSE 固件。

4.27 建议 AB SWAP 后尽快进行功能复位

建议调用 HSE AB SWAP 服务后，尽快调用功能复位，实现 AB 面翻转。

不建议使用上电复位实现 AB 面翻转，因为上电复位后，电源还未稳定，进行 AB 面翻转需要操作 FLASH，耗电大，可能产生 FLASH 擦写失败问题。

在 AB 面翻转服务调用后，没有立即复位，而是进入休眠的情况：

- 1、正常休眠唤醒方式下可以正常唤醒的。
- 2、快速唤醒方式(Fast wakeup)可能不能唤醒。

建议使用快速唤醒方式，AB 面翻转后，先不要进入休眠，而是通过功能复位完成 AB 面翻转后再休眠。

4.28 避免通过上电复位实现固件安装、生命周期演进和 AB SWAP

要通过功能复位实现 HSE 固件安装、ADKP、生命周期演进、AB SWAP 等与 FLASH（包括 UTEST）擦写相关的 HSE 服务，不要通过上电复位实现，否则可能失败。

因为上电复位后，电源还未稳定，进行操作 FLASH，耗电大，可能产生 FLASH 擦写失败问题。

4.29 错误设置 IVT 导致芯片可能锁死问题

IVT 的修改一定要慎重。

检查 IVT 中内核使能是否设置正常。

例如：如果使用 K314，设置使能 2 个核，可能导致芯片被锁，提示 secure debug。

```
(CM7_0_ENABLE << CM7_0_ENABLE_SHIFT) | (CM7_1_ENABLE << CM7_1_ENABLE_SHIFT)
```

如果使用 K324，设置启动核 1，不使能核 0，可能导致芯片被锁，提示 secure debug。

芯片启动后，启动代码 sBAF 会检查 IVT，如果认为异常，则可能会锁芯片，提示 secure debug，因此，需要按照实际情况设置，务必谨慎修改 IVT。

4.30 VKMS 固件 DATA FLASH 占用多问题

若使用 VKMS HSE 固件（大众定制 HSE 固件），需要注意该 HSE 固件会占用更多的 DATA FLASH。因此，使用 FEE 操作 DATA FLASH 时，需要考虑 VKMS 实际使用 FLASH 大小。

4.31 K3X4 使用 0.2.55 版本 AB SWAP 固件读密钥信息问题

若 Key 为空，使用 K3X4 AB Swap，调用获取 Key 信息 API，返回值为参数无效（Parameter invalid）。Full memory 固件正常。K312 AB Swap 调用该功能也正常，即返回值为空（empty）。

请参考该版本 HSE 固件的 Release note，获取更详细的其它已知问题说明。

4.32 编译器优化设置不当导致 HSE 服务失败问题

请检查编译器优化设置。

有客户遇到使用我们提供的示例工程 HSE 服务正常，但将 HSE 相关代码移植到自己工程后异常。

原因是编译器优化设置不同，导致部分变量本应分配到 TCM，实际分配到 cacheable RAM。

4.33 参考应用笔记

1、安全启动应用笔记《S32K3xx Secure Boot Application note v0.1.1.0》

对应中文版：《AN13465-安全启动中文版.pdf》。

2、生命周期演进应用笔记《S32K3xx Lifecycle Management Application note v0.1.1.0.pdf》。

可联系代理商 FAE 获取。

4.34 建议做一些最小功能板用于 HSE 开发

建议制作一些 MCU 最小系统板卡，用于 HSE 功能调试。

特别是调试口加密功能调试，生命周期需要演进到 OEM 或 IN FIELD，演进后调试口加密，需要解密才能调试。可客户自己开发最小系统板，也可从代理商 FAE 获取我们 NXP 开发的最小系统板卡的原理图、Gerber 加工文件和 BOM 文件。

也可以在客户开发出的板卡上，只焊接电源和 MCU 等基本器件，用于 HSE 开发，**以免调试导致 MCU 无法使用或需要输入密码才能使用，造成板卡浪费。**

4.35 HSE 服务未成功的常见原因

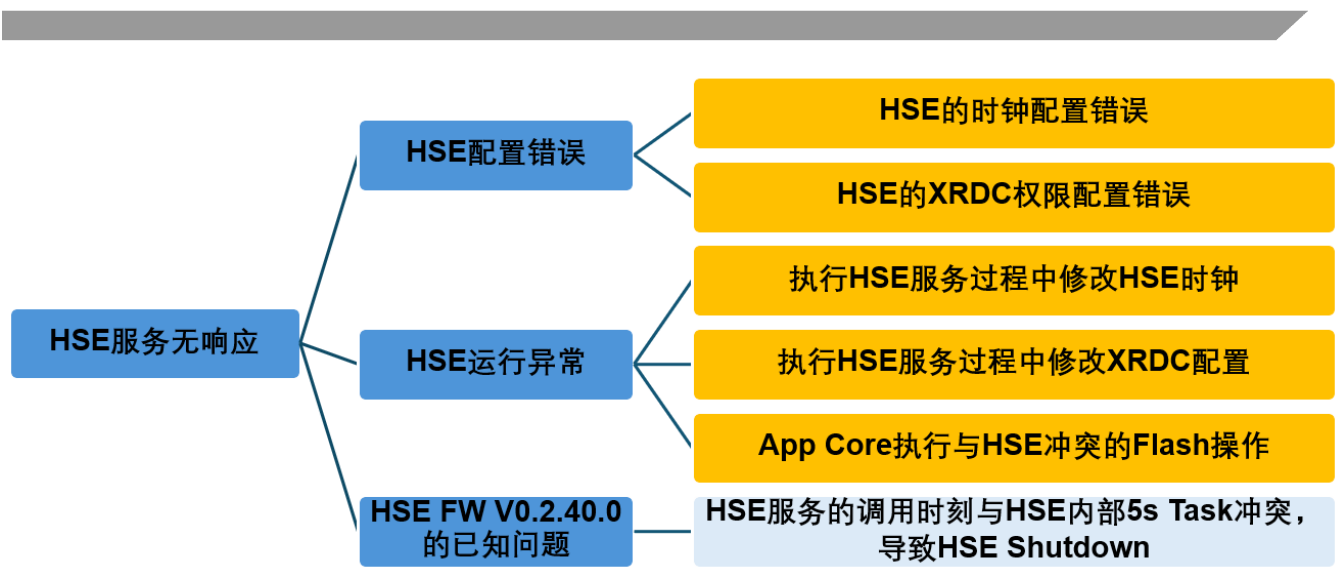
1、Data Cache 相关。例如，传递给 HSE 的参数在 cacheable ram 中。可关闭 Data Cache，观测问题是否消失。

2、MPU 设置错误。详见本文档 MPU 设置章节（3.3.1 节）。

3、HSE 服务描述符内容错误。例如，错误的 pSmrEntry 结构体参数（指针或值错误，例如：pSmrDest 和 smrSize）。

4、时钟配置未严格遵循参考手册的推荐配置。

5、HSE 服务无响应的常见原因，如下：



4.36 建议在密钥导入后调用服务检查加密 NVM 的完整性

如果在 HSE 固件写入密钥到 Secure NVM 过程中出现 ECC，会导致密钥导入失败。因此，建议调用密钥导入服务后，再调用 HSE 服务 *HSE_SRV_ID_FW_INTEGRITY_CHECK* 检查 NVM 完整性，进而确认密钥真正导入成功。

4.37 K311 0.2.55 版本 HSE 固件被擦除后无法安装问题

如果遇到 0.2.55 版本 HSE 固件被擦除后无法安装问题，可尝试安装 0.2.40 版本 HSE 固件，然后，再升级 HSE 固件。

在 K311 上遇到过此问题。若其它型号芯片遇到此问题，可尝试此方法。

4.38 0.2.55 版本 HSE 固件与 0.2.40 版本的主要区别

- 1、 权限变更：在 V0.2.55.0 中，Passive Block 激活不再需要超级用户权限。
- 2、 AB 分区切换优化：分区切换后的固件初始化时间显著缩短。
- 3、 SMR 更新修复：修正了当 pPassReset 超出 SMR 区域时的行为。
- 4、 消除 HSE 无响应问题： 详见本章 4.26 节。
- 5、 ECDSA 参数检查： V0.2.55.0 增加了对 bInputIsHashed 的验证。
- 6、 SHE 密钥目录对齐：新版本强制执行 RAM 中 SHE 密钥的槽位限制。

4.39 建议做故障追踪记录

建议使能 MU 的 GIER 中断，将 MUB 的 GIER[GIE0]位使能，在中断中将 GSR、FSR 和 GPR3 记录，便于故障（例如固件被擦除）定位问题。可通过存入故障信息到 FLASH 或通过 CAN 通讯传出等方式记录保存故障信息。

5. 多核部分部分

5.1 共享存储器 CACHE 问题

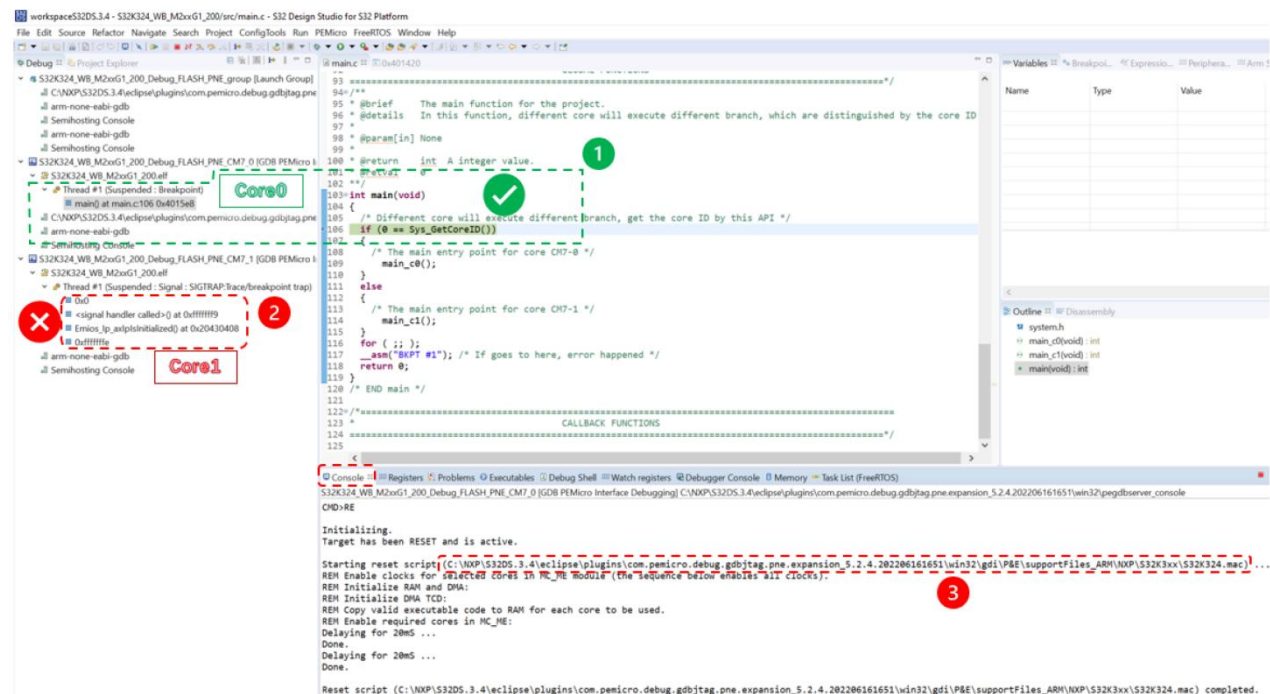
若多核间使用共享 memory 传递信息，那么可以选择如下几种方法之一：

- 1、将该 memroy 设置为 Non-cacheable（推荐）。
- 2、对该 memroy 读写操作前，调用 cache clean 来 clean 和 invalidata cache 数据。

5.2 PE 调试器 mac 文件问题导致无法多核调试

检查是否因为.mac 文件导致从核不能调试。

若使用 PE 调试器，参考下图检查。



To fix the problem with core1, you need delete the “s32k324.mac” file shown in the above figure of ③ (This only needs to be done once).

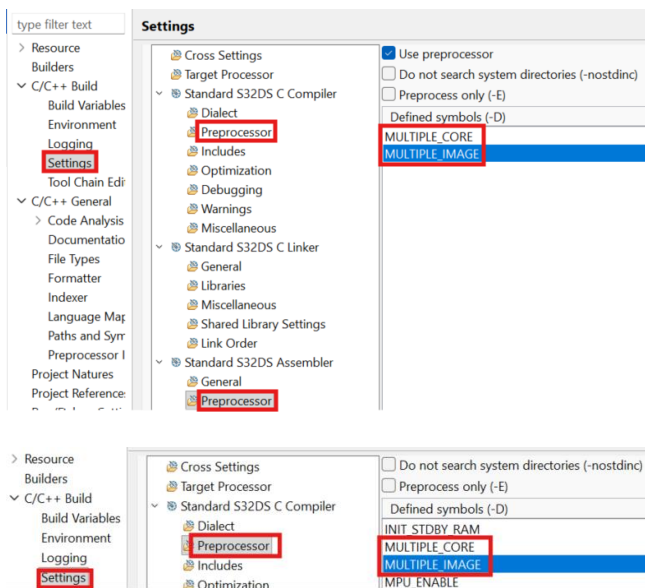
5.3 多核相关宏定义设置问题

单 elf 工程和多 elf 工程，注意启动代码中宏定义设置是否正确，否则出现从核初始化 RAM 不当的问题。

在 RTD4.0.0 和 RTD5.0.0 中，多核项目则需设置 `MULTIPLE_CORE` 宏定义，若不是单 elf 工程，还需要设置 `MULTIPLE_IMAGE` 宏定义，单 elf 工程不要设置该宏定义。

例如，有客户使用 RTD5.0.0, K324, 双 elf 工程，未设置 `MULTIPLE_IMAGE` 宏定义。出现的现象是调试时正常，不带调试器进入 **Hardfault**。原因是此时启动代码不会对核 1 的 RAM 进行 ECC 初始化。

设置方法：



在其它版本 RTD，没有此问题，可忽略此项检查。

5.4 多核对同一个 FLASH 块同时读写问题

检查是否存一个核对某个 FLASH 块（Block）擦除或写入操作，同时另一个核在该 FLASH 块上运行程序。

若存在，可能产生硬件错误。

以 K358 为例，可参考如下措施：

若核 0 对 FLASH 的 Block0 进行擦写操作时，核 2 会访问/读取 Block0 的程序或数据。

- 1、可以通过共享变量方式，让核 2 执行 WFI，然后核 0 禁止核 2 的时钟。
- 2、可以通过核间中断，让核 2 跳转到 RAM 中执行，直至核 0 对 Block0 擦写完成。
- 3、可以将 Block0 需要写数据地址改变到 DATA FLASH。例如，核 0 本来需要在 Block0 写入 Tag 信息，那么，可以考虑将 Tag 信息放入 DATA FLASH，就不需要对 Block0 写操作了。

5.5 共享 RAM shareable 段设置问题

检查双核工程是否使用 RTD3.0.0，RTD4.0.0 和 RTD5.0.0 默认的连接文件（或基于其修改）。

若工程中使用共享RAM，则需要修改默认的连接文件，把shareable放置在end_c0之前，见下图：

```
MEMORY
{
  int_flash_c0           : ORIGIN = 0x00400000, LENGTH = 0x00200000 /* 2048K */
  int_flash_c1           : ORIGIN = 0x00600000, LENGTH = 0x001D4000 /* 2048K - 176K (sBAF + HSE)
  int_itcm               : ORIGIN = 0x00000000, LENGTH = 0x00008000 /* 32K */
  int_dtcM               : ORIGIN = 0x20000000, LENGTH = 0x00010000 /* 64K */
  int_sram_c0            : ORIGIN = 0x20400000, LENGTH = 0x00016F00 /* 91.9K */
  int_sram_fls_rsv_c0    : ORIGIN = 0x20416F00, LENGTH = 0x00000100 /* 0.1K */
  int_sram_stack_c0     : ORIGIN = 0x20417000, LENGTH = 0x00001000 /* 4KB */
  int_sram_no_cacheable_c0 : ORIGIN = 0x20418000, LENGTH = 0x00008000 /* 32KB, needs to include ir
  int_sram_shareable     : ORIGIN = 0x20420000, LENGTH = 0x00008000 /* 32KB */
  ram_end_c0            : ORIGIN = 0x20428000, LENGTH = 0x00000000 /* End of core 0 ram */
  int_sram_c1            : ORIGIN = 0x20430000, LENGTH = 0x0000AF00 /* 44.8K */
  int_sram_fls_rsv_c1    : ORIGIN = 0x2043AF00, LENGTH = 0x00000100 /* 0.1K */
  int_sram_stack_c1     : ORIGIN = 0x2043B000, LENGTH = 0x00001000 /* 4KB */
  int_sram_no_cacheable_c1 : ORIGIN = 0x2043C000, LENGTH = 0x00004000 /* 16KB, needs to include ir
  ram_end_c1            : ORIGIN = 0x20440000, LENGTH = 0x00000000 /* End of core 1 ram */
  ram_rsvd2             : ORIGIN = 0x20450000, LENGTH = 0 /* End of SRAM */
}
```

RTD2.0.x不存在这个问题。

5.6 XRDC 使用注意事项

若配置 XRDC，并且使用 HSE，请检查 HSE Domain 是否加入。

5.7 参考应用笔记

- 1、多核应用笔记《AN-Multi-Core_Implementation_in_RTD_of_S32K3.pdf》。
- 2、XRDC 应用笔记《S32K3xx resources isolation and protection by Extended Resource Domain Controller(XRDC)》。请从代理商 FAE 获取。

5.8 示例工程

5.8.1 基于 K324 和 K338 的多核示例工程

由 NXP 应用工程师团队开发，包括 EB 配置工程和 S32DS 工程，如下：

Name	Status	Date modified
Icu_Wkup_WB_M2xxG2_200	✔ ◻	2022/12/12 15:58
MemSlaveProt_M2xxG1_200	✔ ◻	2022/12/12 15:58
PeriSlaveProt_M2xxG1_200	✔ ◻	2022/12/12 15:58
ResourcesAssign_M2xxG1_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2FFG2_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2FxG2_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2xxG1_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2xxG2_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2xxH1_200	✔ ◻	2022/12/12 15:58
S32K324_WB_M2xxI2_200	✔ ◻	2022/12/12 15:58
S32K338_BasePrj_M3xxxG1_300_P07	✔ ◻	2024/3/28 16:07
Siul2Prot_M2xxG1_200	✔ ◻	2022/12/12 15:58
StackInTCM_M2xxG1_200	✔ ◻	2022/12/12 15:58
Tcm2SysRam_M2xxG1_200	✔ ◻	2023/11/20 11:16
TwoVectTable_M2xxG1_300_P07	✔ ◻	2023/11/20 11:11
VectTableInTCM_M2xxG1_200	✔ ◻	2023/11/20 11:13

工程说明文件为：《S32K32X_Multicore_Project_Quick_Start_Guide.pdf》。该文件介绍了工程使用方法和注意事项。《ProjectNameFormat.pdf》是工程命名规则。

请从代理商 FAE 获取示例工程和工程说明文件。

5.8.2 基于 RTD1.0.0 S32DS CT 配置的 K324 工程

工程名称《s32k324_dualcore_demo_20220524.zip》

工程说明文件：《S32K324_Dual_Core_Basics_v1.2.pdf》

工程说明：该工程是详细介绍了信号量的使用，通过 XRDC 将共享 RAM 区与信号量绑定，当未获取信号量时对共享 RAM 访问，会触发 `hardfault`，示例工程有对应代码实现和测试。

工程包括核间中断的使用。通过核间中断实现双核对共享变量的访问。

该工程是学习信号量和核间中断的很好的示例。

请从代理商 FAE 获取示例工程和工程说明文件。

5.8.3 基于 RTD4.0.0P24 S32DS3.5 CT 配置的 K358 工程

工程名称《基于 RTD4.0.0P24 的 K358 LLD Demo》

工程说明：工程提供 2 种由核 0 启动核 2 的方法，一是通过调用 API Power_Ip_SetMode，一是通过裸写代码实现。

请从代理商 FAE 获取示例工程。

5.8.4 基于 RTD2.0.1 S32DS CT 配置的 K324 FreeRTOS 工程

工程名称《FreeRTOS_Toggle_Led_Example_S32K324_M7_ForPE.7z》

工程说明：使用 FreeRTOS，双 elf 工程，双核都翻转 LED 灯，PE 调试器。

请从代理商 FAE 获取示例工程。

5.8.5 基于 RTD3.0.0P07 EB 配置的 K338 GHS 工程

工程名称《S32K338_BasePrj_M3xxxG1_300_P07_GHS.zip》

实现单 elf 三核工程。该工程使用 Green Hills 编译器。

该工程是 NXP 重庆 AE 开发的多核示例工程。请从代理商 FAE 获取示例工程。

5.9 OS 配置不当导致多核启动异常

若出现多核芯片启动异常问题，需要检测是否与操作系统（OS）有关。

客户遇到过由于 OS 导致多核启动异常的问题。

5.10 单 ELF 工程不能完全使用 TCM 的问题

单 ELF 工程，可以让每个核的堆栈分配到各自的 DTCM，但是变量需要按顺序编译，导致 TCM 资源浪费，每个核都有部分 TCM 无法使用。

例如，希望将核 0 的 64K 变量和核 1 的 64K 变量编译到 DTCM 中，每个核的 DTCM 大小都是 128K。编译后，核 0 的变量地址被分配到 0x20000000~0x20007FFF，核 1 的变量地址被分配到 0x20008000~0x2000FFFF。核 0 的 DTCM 地址 0x20008000~0x2000FFFF 无法再使用，核 1 的 DTCM 地址 0x20000000~0x20007FFF 无法再使用。

多 ELF 工程没有这个限制，可以充分使用 TCM 资源。

6. 功能安全部分

6.1 FCCU 相关

6.1.1 FCCU 故障状态下外设引脚无输入输出功能问题

当使能了 FCCU，并且 FCCU 处于故障状态时，如果没有勾选外设引脚的 SMC 使能，则外设引脚无输入输出功能，因为 FCCU 关闭了 IO 的输出 buffer。

例如，进行故障注入，将 FCCU 的反应类型设置为复位，并且没有勾选外设引脚（例如 JTAG/SPI/CAN 等）的 SMC 使能，则由于 FCCU 产生的复位后，外设引脚无功能输入输出。

解决办法是使能外设引脚的 SMC 配置。

6.2 BIST 相关

6.2.1 Trace 时钟和 CLKOUT 需要设置成 FIRC

若 Trace 时钟和 CLKOUT 时钟未设置成 FIRC，进行 BIST 后，时钟初始化时间会增加。

6.3 故障注入相关

6.3.1 故障注入后再进行 HSE 服务调用

检查 EIM 错误注入与 HSE 服务调用是否冲突。

S32K3 EIM 在注入错误时，比如对 RAM 或者 FLASH 的错误注入，可能会和 HSE 的某些服务产生冲突，比如 HSE 对 FLASH 或 RAM key 的操作。建议先进行 EIM 错误注入的检测，之后再调用 HSE 服务进行 FLASH 或者 RAM 的 key 处理。

6.4 SAF 相关

6.4.1 检查使用版本的已知 Bug

目前 SAF 的最新版本是 1.0.6CD02，在该版本的 Release note 中列出了之前版本的 Bug 情况。

请申请 SAF 开发 License（详见 6.6 节），或从 NXP FAE，获取最新版本的 Release note，检查本项目使用的 SAF 版本存在的 Bug 是否在本项目中可能产生。

6.4.2 将部分 sCheck 检测项放在 shutdown 阶段用以减小启动阶段时间

若在启动阶段运行所有的 sCheck 检测项，启动时间增大很多。

可以考虑将部分或全部 sCheck 检测项放在关机（shutdown）阶段进行。

例如，将 FCCU 初始化，ADC 自检放在启动阶段，将 XRDC、FLASH 和看门狗测试放在关机阶段进行。

6.4.3 非锁步核不需要进行 RCCUTEST_RCCU_CM7 测试

6.4.4 SAF1.0.5 之前版本可能出现看门狗 sCheck 失败的问题

该问题与芯片 ERRATA 052226 有关。

1.0.5 版本 SAF 通过为 TO 值增加低门限的方法修复了这个问题。

6.4.5 SAF1.0.4 版本 Dcache 和 Flash EDMA sCheck 失败问题

使用 HSE ABSWAP 后，当 Active block 是高地址时，1.0.4 版本 SAF 会出现 Flash EDMA sCheck 失败问题。在 1.0.5 版本 SAF 修复了这个 Bug。

建议使用新版本 SAF。

建议使用旧版本 SAF 时，提前阅读最新版本 SAF 列出的旧版本已知问题。

6.4.6 SAF1.0.4 版本 AB Swap 固件下 sCHECK FLASH 检测失败问题

检查是否使用 SAF1.0.4 版本，同时，安装了 HSE 固件，通过 AB SWAP OTA 到物理高地址，此时进行 sCHECK FLASH 检测。

若是，可能检测失败。

SAF1.0.5 版本修复了该问题。

6.4.7 xRDC 配置错误可能导致 sCheck 失败

检查 sCheck 失败是否是因为 xRDC 配置错误。

对 TCM 检查时 K3X4 使用 MRC1，K3X8 使用 MRC3。

在启动中进行 sCheck，可以不配置 xRDC。

6.4.8 Gmac 模块进行 sBoot 前需完成初始化

6.4.9 其它

- 1、sCheck 前，清除 DCMGPRWF1 的 .mac 选择，即选择 MII 模式，否则，会导致 GMAC sCheck 不通过。
- 2、检查链接文件和 MPU 设置是否合理。
- 3、在 SAF1.0.3/4/5 集成到 Vector AUTOSAR 包后，sCheck 缺少 Core ID 信息。预计 SAF1.0.6 中修复该问题。目前可从 NXP FAE 获取修复代码。

6.5 SAF license 说明

NXP 不提供 SAF 的开发 License，只提供评估版本 License，和之前一样。

如有疑问，请联系 NXP 销售或 Marketing。

6.6 应用笔记

6.6.1 S32K3xx_Functional_Safety_Application_Note

由 NXP 应用工程师团队编写。










请联系代理商 FAE 获取。

6.7 示例工程

6.7.1 基于 K324 的功能安全和信息安全集成 Demo

由 NXP 应用工程师团队开发。

工程文件夹名称《S32K324_SafSecIntegrate》，内含基于 GCC 编译器和 GHS 编译器的示例工程，以及工程介绍文档，如下：

Name	Status	Date modified	Type	Size
 S32K3_Safety_Security_Training.pdf	 	2025/2/12 18:09	Adobe Acrobat D...	3,596 KB
 S32K324_M2FFG1_SafSecIntegrate.7z	 	2025/2/12 18:09	7Z File	7,115 KB
 S32K324_M2XXH1_SafSecIntegrate.7z	 	2025/2/12 18:09	7Z File	6,209 KB





说明：由于 SCST 和 SAF 是收费软件，需要客户具备这些软件的 license 才能提供。

SCST 提供评估 License。不能获取全部代码，只能对部分功能进行评估。

6.7.2 基于 K344 和 K358 的功能安全 Demo

由 NXP FAE 开发。

内含基于 GCC 编译器和 GHS 编译器的示例工程，以及工程介绍文档，如下：

-  S32K3_RTD500_SCST105
-  S32K3_Safety_RTD400_SAF104_K344_example
-  S32K3_Safety_RTD500_SAF105_K358_example
-  S32K3_Safety_Software_Integration_Guide(continue work product).pptx

说明：由于 SCST 和 SAF 是收费软件，需要客户具备这些软件的 license 才能提供。

附录 A 文档中使用的缩略语

HW	Hardware
SW	Software
POR	Power-On Reset
OS	Operation System
SCST	Structural Core Self-Test (Premium software)
SPD	Safety Peripheral Driver (Standard software and it's a sub-set of SAF)
SAF	Safety Software Framework (Premium software)
Bist	Built-in self-test
sCheck	Square Check (SW injects errors to verify HW safety mechanisms)
sBoot	Safety Boot (static registers check)
FCCU	Fault Collection and Control Unit
SWT	Software Watchdog Timer
MPU	Memory Protection Unit
CMU	Clock Monitor Unit
PMC	Power Management Controller
CRC	Cyclic Redundancy Check
ECC	Error Correction Code
ERM	Error Reporting Module
EIM	Error Injection Module
LVD	Low voltage detection
HVD	High voltage detection
LVR	Low voltage reset
XRDC	Extended Resource Domain Controller

附录 B 文档变更说明

2025 年 8 月 29 日发布 V1.0 版本

2025 年 10 月 9 日发布 V1.1 版本

V1.1 版本修改内容如下：

1	增加 5.9 节
2	完善 5.4 节
3	增加 3.20.14 节
4	增加 4.35 节
5	增加 3.2.6 节
6	增加 3.4.16 节
7	增加 3.1.8 节
8	完善 4.12 节内容
9	增加 5.10 节
10	增加 4.37 节
11	修改 6.5 节内容
12	修改 3.17.6 节内容
13	修改 4.25 节内容
14	修改 3.7.1 节内容

2025 年 11 月 13 日发布 V1.2 版本

V1.2 版本在 V1.1 的基础上修改内容如下：

1	修改 3.1.6 节。去掉使用烧录新版 sBAF 的解决方案。因为 NXP 暂无使用新版 sBAF 烧录到正常量产芯片（非特殊料号芯片）的计划。
2	修改 3.17.6 节。完善第 7 条，即：晶振 EXTAL 到 MCU 晶振引脚的距离尽量短。增加解决 ESD 问题的第 10 条建议，即：对于未使用 HSE 的产品，尝试将 PLL 失锁从复位改成中断。增加第 11 条建议，即：使用有源晶振。增加第 12 条建议，即：读取复位源。增加第 13 条建议，即：跨导值满足硬件设计指南要求，降低晶振频率。
3	修改 3.6.6 节。增加 RTD6.0.0 QLP04 存在的 K310/311 部分 ADC 通道无法采集数据问题。
4	增加 3.9.3 节。I2C 作从模式，并且为 FAST 模式可能遇到的问题。
5	修改 5.3 节。以实际案例为例，加入具体解决措施。
6	修改 3.3.2 节。加入规避 ERR052460 的第 3 种方法。近期仍有客户遇到此问题，而且是量产后发现该问题，问题严重，务必检查此项。
7	修改 3.4.4 节。RAM 等待设置问题。增加实际案例异常情况说明，近期仍有客户遇到此问题，而且是量产后发现该问题，问题严重，务必检查此项。
8	增加 4.38 节。说明 0.2.55.0 版本 HSE 固件与 0.2.40.0 版本的差异，建议更新至 0.2.55。
9	增加 3.12.4。说明 PTE13 作为三极管基极控制功能时的配置方法。

10	修改 6.4.9 节。增加一条，即：在 SAF1.0.3/4/5 集成到 Vector AUTOSAR 包后，sCheck 缺少 Core ID 信息。
11	修改 3.14.1 节。最新 RTD 版本为 7.0.0，推荐新项目 and 正在开发的项目使用该版本。该版本的 Release note 中有 Bug 修复和新功能增加的说明。由于该版本 RTD 暂无示例工程，预计 2026 年 2 月的 7.0.0 版本才提供示例工程，这会带来一些不便。另外，该版本暂无 Crypto 模块，预计几个月后才会有包含该模块的版本。该版本支持 AUTOSAR R23-11。综合这些差异，客户最终确定选择 7.0.0，还是 6.0.0 中的最新量产版本。
12	增加 3.9.4 节。I2C 上拉电阻设置建议。

2026 年 1 月 4 日发布 V1.3 版本

V1.2 版本在 V1.1 的基础上修改内容如下：

1	增加 3.20.15 节，EB 的 Links 文件夹内含有多个版本 RTD 的 link 文件可能出现问题。
2	增加 3.19.9 节，减小启动时间的常用方法。
3	增加 2.14 节，推荐使用 SPI 模块自身的片选信号。
4	增加 3.4.17 节，Cache 无效化问题，RTD4.0.0 及之前版本（不包括 RTD4.0.0HF01/02）软件的 Bug。
5	增加 3.6.12 节，ADC 自检失败问题。
6	修改 4.2 节。K328/338/348/358 需要写 DCF，否则 HSE 固件可能被擦除。 HSE LIB2.0 更新 ，版本为 2025 年 12 月 30 日 版本。该版本增加了更多的时钟选项，增加了密钥更新和交换的用例代码，增加了某些情况下旁通 HSE INIT OK 的代码。
7	修改 3.4.16 节。K328/338/348/358 读写 SRAM_2 出错问题。 <u>这个问题不容易在研发阶段发现，需认真检查该条内容。</u>
8	修改 4.35 节。加入 HSE 无响应的常见原因。
9	修改 3.4.13 节。Prefetch 的打开和关闭操作建议在 RAM 或 ITCM 中进行。
10	增加 4.39 节。建议做故障追踪记录。在产品出现固件被擦除等问题后，若能记录故障信息，则便于定位问题。
11	增加 3.2.7 节。有源晶振使用注意事项。
12	增加 2.13.5 节。建议 EXTAL 和其附近的 V15/V25 的滤波电容放置在不同的面。
13	修改 3.16.2 节。打开 Memory、寄存器、变量观测页面，有时会对调试有影响
14	增加 3.6.13 节。ADC 读组转换结果偶发失败问题。