**S32K3 常见问题检查列表   
Check List**

版本 V1.0

2025年8月29日

目录

[**1.** **介绍** 11](#_Toc207354699)

[**1.1** **目的** 11](#_Toc207354700)

[**1.2** **免责声明** 11](#_Toc207354701)

[**2.** **硬件设计部分** 12](#_Toc207354702)

[**2.1** **必须遵循数据手册和硬件设计指南** 12](#_Toc207354703)

[**2.2** **必须遵循参考手册** 12](#_Toc207354704)

[**2.3** **必须参考应用笔记** 13](#_Toc207354705)

[**2.4** **必须参考EVB** 14](#_Toc207354706)

[**2.5** **高速通讯引脚建议选择Fast或Medimu类型引脚** 14](#_Toc207354707)

[**2.6** **高驱动能力引脚建议选择Fast或Medimu类型引脚** 14](#_Toc207354708)

[**2.7** **注意eMIOS模块不同类型通道的选择** 15](#_Toc207354709)

[**2.8** **注意只能做输入的引脚** 15](#_Toc207354710)

[**2.9** **不允许将高压信号通过串电阻接入IO引脚** 15](#_Toc207354711)

[**2.10** **ECC预防电路设计** 15](#_Toc207354712)

[**2.11** **ADC硬件设计注意事项** 15](#_Toc207354713)

[**2.11.1** P/S/X三种通道类型的选择 15](#_Toc207354714)

[**2.11.2** **不允许将高压信号通过串电阻接入ADC引脚** 16](#_Toc207354715)

[**2.11.3** **输入阻抗过大会导致采样时间长和通道间干扰** 16](#_Toc207354716)

[**2.11.4** **ADC参考电压低不能通过磁珠接地** 17](#_Toc207354717)

[**2.11.5** **ADC参考电压不能超过VDD\_HV\_A** 17](#_Toc207354718)

[**2.11.6** **需遵守K3 数据手册的要求** 17](#_Toc207354719)

[**2.11.7** **需遵守K3 硬件设计指南的要求** 17](#_Toc207354720)

[**2.11.8** **需参考硬件应用笔记的要求** 17](#_Toc207354721)

[**2.11.9** **ADC引脚上的高压脉冲可能损坏芯片** 17](#_Toc207354722)

[**2.12** **K3X8+FS26唤醒注意事项** 17](#_Toc207354723)

[**2.13** **遵守EMC相关要求** 19](#_Toc207354724)

[**2.13.1** **推荐使用6层板** 19](#_Toc207354725)

[**2.13.2** **产品金属外壳会提升EMC性能** 19](#_Toc207354726)

[**2.13.3** **增加产品外壳厚度可以提高MCU抗干扰能力** 19](#_Toc207354727)

[**2.13.4** **增加稳压二极管解决EMC问题** 19](#_Toc207354728)

[**3.** **软件开发部分** 20](#_Toc207354729)

[**3.1** **电源和低功耗（休眠唤醒）部分** 20](#_Toc207354730)

[**3.1.1** **Last Mile的配置** 20](#_Toc207354731)

[**3.1.2** **Pad keeping** 21](#_Toc207354732)

[**3.1.3** **建议使能低电压检测并修复清标志位Bug** 22](#_Toc207354733)

[**3.1.4** **进入休眠过程需要关闭外设并关闭中断** 22](#_Toc207354734)

[**3.1.5** **建议配置唤醒引脚滤波使能** 23](#_Toc207354735)

[**3.1.6** **快速唤醒后擦写FLASH失败问题** 23](#_Toc207354736)

[**3.1.7** **需参考NXP开发的低功耗示例工程** 24](#_Toc207354737)

[**3.2** **时钟** 25](#_Toc207354738)

[**3.2.1** **需严格遵守参考手册列出的几种时钟配置** 25](#_Toc207354739)

[**3.2.2** **使用的外设门控时钟要配置为使能** 27](#_Toc207354740)

[**3.2.3** **快速外部晶振FXOSC的自动增益设置** 28](#_Toc207354741)

[**3.2.4** **RTC同时用作计时和唤醒的注意事项** 29](#_Toc207354742)

[**3.2.5** **IAR编译器设置FLASH延时等待问题** 29](#_Toc207354743)

[**3.3** **M7内核** 30](#_Toc207354744)

[**3.3.1** **MPU设置** 30](#_Toc207354745)

[**3.3.2** **ARM推测性访问（**ERR052460**）的问题** 31](#_Toc207354746)

[**3.3.3** **Hardfault问题排查** 32](#_Toc207354747)

[**3.3.4** **浮点运算异常问题** 32](#_Toc207354748)

[**3.4** **存储器** 33](#_Toc207354749)

[**3.4.1** **需要具备ECC恢复机制** 33](#_Toc207354750)

[**3.4.2** **建议使能低电压监测来尽量避免ECC的发生** 33](#_Toc207354751)

[**3.4.3** **需要使用8字节写的方式对RAM进行ECC初始化** 33](#_Toc207354752)

[**3.4.4** **RAM操作等待和FLASH操作等待设置不当导致的问题** 34](#_Toc207354753)

[**3.4.5** **FLASH同一个块读同时写导致Hardfault问题** 34](#_Toc207354754)

[**3.4.6** **M7核和HSE核同时对一个FLASH块读写问题** 36](#_Toc207354755)

[**3.4.7** **栈溢出问题** 36](#_Toc207354756)

[**3.4.8** **Fee问题** 36](#_Toc207354757)

[**3.4.9** **RTD3.0.0-4.0.0虚拟扇区大小错误问题** 37](#_Toc207354758)

[**3.4.10** **链接文件中TCM大小与实际芯片需一致** 37](#_Toc207354759)

[**3.4.11** **TCM后门访问问题** 37](#_Toc207354760)

[**3.4.12** **RTD4.0.0擦写最后一个扇区问题** 38](#_Toc207354761)

[**3.4.13** **不主动打开Prefetch** 38](#_Toc207354762)

[**3.4.14** **对齐访问问题** 38](#_Toc207354763)

[**3.4.15** **注意UTEST的只写一次特性** 39](#_Toc207354764)

[**3.5** **复位相关问题** 39](#_Toc207354765)

[**3.5.1** **APP中无法获取复位源问题** 39](#_Toc207354766)

[**3.5.2** **恢复模式Recovery mode相关问题** 39](#_Toc207354767)

[**3.6** **ADC** 40](#_Toc207354768)

[**3.6.1** **需遵守K3 参考手册的要求** 40](#_Toc207354769)

[**3.6.2** **温度通道采样时间的要求** 40](#_Toc207354770)

[**3.6.3** **ADC采样时钟的要求** 41](#_Toc207354771)

[**3.6.4** **建议使能预采样** 42](#_Toc207354772)

[**3.6.5** **分辨率建议设置为12位** 42](#_Toc207354773)

[**3.6.6** **K310和K311 ADC使用注意事项** 43](#_Toc207354774)

[**3.6.7** **检查最新版本RTD的发布说明中提到的Bug** 43](#_Toc207354775)

[**3.6.8** **建议参考NXP的示例工程** 43](#_Toc207354776)

[**3.6.9** **ADC校准建议** 44](#_Toc207354777)

[**3.6.10** **ADC超时时间设置** 44](#_Toc207354778)

[**3.6.11** **ADC引脚上的高压脉冲可能损坏芯片** 44](#_Toc207354779)

[**3.7** **CAN** 44](#_Toc207354780)

[**3.7.1** **使能Enhance FIFO后，部分邮箱丢帧问题** 44](#_Toc207354781)

[**3.7.2** **CAN模块处于start状态下反初始化失败** 45](#_Toc207354782)

[**3.7.3** **CAN帧发送乱序问题** 45](#_Toc207354783)

[**3.7.4** **建议CAN FD使能传输延时补偿TDC** 45](#_Toc207354784)

[**3.7.5** **CAN采样点不满足要求问题** 45](#_Toc207354785)

[**3.7.6** **时钟配置错误导致的问题** 46](#_Toc207354786)

[**3.8** **SPI** 46](#_Toc207354787)

[**3.8.1** **使用GPIO做片选信号时CLK信号异常脉冲问题** 46](#_Toc207354788)

[**3.8.2** **配置参数需合理** 46](#_Toc207354789)

[**3.8.3** **SPI同步传输时间长问题** 46](#_Toc207354790)

[**3.8.4** **时钟配置错误导致的问题** 46](#_Toc207354791)

[**3.9** **I2C** 47](#_Toc207354792)

[**3.9.1** **采样保持时间无法调整问题** 47](#_Toc207354793)

[**3.9.2** **时钟配置错误导致的问题** 47](#_Toc207354794)

[**3.10** **以太网** 47](#_Toc207354795)

[**3.10.1** **以太网初始化前外部时钟需稳定** 47](#_Toc207354796)

[**3.10.2** **千兆以太网使用注意事项** 47](#_Toc207354797)

[**3.11** **eMIOS** 47](#_Toc207354798)

[**3.11.1** **PWM从100%占空比切换到0%占空比偶发失败** 47](#_Toc207354799)

[**3.12** **端口引脚设置** 48](#_Toc207354800)

[**3.12.1** **建议将JTAG信号配置安全模式** 48](#_Toc207354801)

[**3.12.2** **建议将复位信号配置成滤波使能** 49](#_Toc207354802)

[**3.12.3** **合理设置未使用引脚** 49](#_Toc207354803)

[**3.13** **中断相关** 50](#_Toc207354804)

[**3.13.1** **中断服务程序执行时间长问题** 50](#_Toc207354805)

[**3.13.2** **尽量避免长时间关闭全局中断** 50](#_Toc207354806)

[**3.13.3** **中断向量表需要128字节对齐** 50](#_Toc207354807)

[**3.13.4** **中断优先级设置要合理** 50](#_Toc207354808)

[**3.14** **RTD相关** 51](#_Toc207354809)

[**3.14.1** **推荐使用新版本RTD** 51](#_Toc207354810)

[**3.14.2** **参考RTD示例工程定位问题** 51](#_Toc207354811)

[**3.14.3** **RTD中EB配置示例工程移植到S32DS** 51](#_Toc207354812)

[**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** 52](#_Toc207354813)

[**3.14.5** **RTD中S32DS CT配置示例工程移植到K312** 52](#_Toc207354814)

[**3.14.6** **使用新版本RTD验证复现问题** 53](#_Toc207354815)

[**3.14.7** **根据最新版本RTD的发布说明排查使用RTD版本的Bug** 53](#_Toc207354816)

[**3.14.8** **链接文件问题** 54](#_Toc207354817)

[**3.14.9** **RTD1.0.0和RTD2.0.0 启动代码耗时长问题** 54](#_Toc207354818)

[**3.15** **编译器相关** 55](#_Toc207354819)

[**3.15.1** **优化等级设置不合理会出现问题** 55](#_Toc207354820)

[**3.15.2** **操作系统的优化等级需合理设置** 57](#_Toc207354821)

[**3.15.3** **IAR编译器** 57](#_Toc207354822)

[**3.16** **调试器相关** 58](#_Toc207354823)

[**3.16.1** **当用某种调试器无法连接板卡时，尝试其它调试器** 58](#_Toc207354824)

[**3.16.2** **尝试不带调试器后板卡运行是否正常** 58](#_Toc207354825)

[**3.16.3** **勾选调试器异常捕获选项造成的异常** 58](#_Toc207354826)

[**3.16.4** **调试器对RAM区进行ECC初始化** 59](#_Toc207354827)

[**3.16.5** **通过Attach 定位问题** 59](#_Toc207354828)

[**3.17** **EMC** 60](#_Toc207354829)

[**3.17.1** **请遵循硬件设计指南中关于EMC的建议** 60](#_Toc207354830)

[**3.17.2** **使用PLLFM，可解决EMC 1.6GHz左右不通过问题** 60](#_Toc207354831)

[**3.17.3** **使能Reset引脚的滤波功能** 60](#_Toc207354832)

[**3.17.4** **通过写一条DCF禁用Reset引脚的复位功能** 61](#_Toc207354833)

[**3.17.5** **禁止FXOSC的自动增益** 61](#_Toc207354834)

[**3.17.6** **解决ESD不通过的常用措施** 61](#_Toc207354835)

[**3.17.7** **非MCU原因引起EMC不通过情况** 61](#_Toc207354836)

[**3.18** **需检查勘误表ERRATA** 62](#_Toc207354837)

[**3.19** **性能提升** 62](#_Toc207354838)

[**3.19.1** **使能CACHE会显著提升代码执行速度** 62](#_Toc207354839)

[**3.19.2** **D-CACHE使能注意事项** 63](#_Toc207354840)

[**3.19.3** **将栈放入DTCM** 64](#_Toc207354841)

[**3.19.4** **使能FLASH Prefetch** 65](#_Toc207354842)

[**3.19.5** **使用TCM** 65](#_Toc207354843)

[**3.19.6** **使用编译器优化** 65](#_Toc207354844)

[**3.19.7** **使用GHS或IAR编译器** 66](#_Toc207354845)

[**3.19.8** **Fee性能提升** 66](#_Toc207354846)

[**3.20** **其它** 67](#_Toc207354847)

[**3.20.1** **初始化时间长问题** 67](#_Toc207354848)

[**3.20.2** **合理设置超时时间** 67](#_Toc207354849)

[**3.20.3** **通过DSB和ISB解决代码非顺序执行问题** 68](#_Toc207354850)

[**3.20.4** **OS使用注意事项** 68](#_Toc207354851)

[**3.20.5** **建议每个项目都配备EVB** 68](#_Toc207354852)

[**3.20.6** **建议使用新版本手册** 69](#_Toc207354853)

[**3.20.7** **焊接问题** 69](#_Toc207354854)

[**3.20.8** **测量MCU引脚输出电平是否正常** 69](#_Toc207354855)

[**3.20.9** **看门狗模块调试** 69](#_Toc207354856)

[**3.20.10** **逻辑分析仪测试造成的误导** 69](#_Toc207354857)

[**3.20.11** **周期任务和中断都操作某个外设的问题** 70](#_Toc207354858)

[**3.20.12** **Bootloader跳转APP失败问题** 70](#_Toc207354859)

[**3.20.13** **通过NXP社区获取示例工程和问题解决方案** 71](#_Toc207354860)

[**4.** **信息安全(HSE)部分** 71](#_Toc207354861)

[**4.1** **建议使用最新版本HSE固件并升级sBAF** 71](#_Toc207354862)

[**4.2** **HSE固件安装注意事项** 71](#_Toc207354863)

[**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** 73](#_Toc207354864)

[**4.4** **建议参考NXP中国应用工程师团队开发的HSE库2.0** 73](#_Toc207354865)

[**4.5** **未安装HSE固件情况下，时钟初始化前必须检查WFI标志** 74](#_Toc207354866)

[**4.6** **未安装HSE固件情况下，FLASH擦写前必须检查WFI标志** 75](#_Toc207354867)

[**4.7** **安装HSE固件情下，时钟初始化前必须检查HSE\_STATUS\_INIT\_OK** 75](#_Toc207354868)

[**4.8** **安装HSE固件情下，FLASH擦写前必须检查GPR3寄存器** 76](#_Toc207354869)

[**4.9** **功能/破坏性复位后HSE\_STATUS\_INIT\_OK置位时间长问题** 77](#_Toc207354870)

[**4.10** **随机数服务调用前检查随机数初始化OK标志** 77](#_Toc207354871)

[**4.11** **随机数服务首次调用耗时长问题** 77](#_Toc207354872)

[**4.12** **ADKP和生命周期演进注意问题** 78](#_Toc207354873)

[**4.13** **D-CACHE打开导致HSE服务失败问题** 78](#_Toc207354874)

[**4.14** **局部变量传递给HSE的注意事项** 79](#_Toc207354875)

[**4.15** **生命周期演进前需格式化Key目录** 79](#_Toc207354876)

[**4.16** **通过寄存器定位问题** 79](#_Toc207354877)

[**4.17** **使用空闲 的MU Channel** 80](#_Toc207354878)

[**4.18** **建议申请超级用户权限** 80](#_Toc207354879)

[**4.19** **生命周期演进后无法AB面切换（OTA）问题** 80](#_Toc207354880)

[**4.20** **HSE固件损坏或被擦除的原因** 81](#_Toc207354881)

[**4.21** **HSE信息安全标准认证情况** 81](#_Toc207354882)

[**4.22** **RTD3.0.0和RTD4.0.0的IVT错误问题** 81](#_Toc207354883)

[**4.23** **安装HSE固件后时钟初始化进入Hardfault问题** 82](#_Toc207354884)

[**4.24** **使用上电复位实现HSE功能导致的问题** 83](#_Toc207354885)

[**4.25** **OTA过程喂狗问题** 83](#_Toc207354886)

[**4.26** **0.2.40及之前版本存在极小概率HSE服务无响应问题** 84](#_Toc207354887)

[**4.27** **建议AB SWAP后尽快进行功能复位** 85](#_Toc207354888)

[**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** 85](#_Toc207354889)

[**4.29** **错误设置IVT导致芯片可能锁死问题** 85](#_Toc207354890)

[**4.30** **VKMS固件DATA FLASH占用多问题** 86](#_Toc207354891)

[**4.31** **K3X4 使用0.2.55版本AB SWAP固件读密钥信息问题** 86](#_Toc207354892)

[**4.32** **编译器优化设置不当导致HSE服务失败问题** 86](#_Toc207354893)

[**4.33** **参考应用笔记** 86](#_Toc207354894)

[**4.34** **建议做一些最小功能板用于HSE开发** 86](#_Toc207354895)

[**5.** **多核部分部分** 87](#_Toc207354896)

[**5.1** **共享存储器CACHE问题** 87](#_Toc207354897)

[**5.2** **PE调试器mac文件问题导致无法多核调试** 87](#_Toc207354898)

[**5.3** **多核相关宏定义设置问题** 88](#_Toc207354899)

[**5.4** **多核对同一个FLASH块同时读写问题** 88](#_Toc207354900)

[**5.5** **共享RAM shareable段设置问题** 88](#_Toc207354901)

[**5.6** **XRDC使用注意事项** 89](#_Toc207354902)

[**5.7** **参考应用笔记** 89](#_Toc207354903)

[**5.8** **示例工程** 89](#_Toc207354904)

[**5.8.1** **基于K324和K338的多核示例工程** 89](#_Toc207354905)

[**5.8.2** **基于RTD1.0.0 S32DS CT配置的K324工程** 90](#_Toc207354906)

[**5.8.3** **基于RTD4.0.0P24 S32DS3.5 CT配置的K358工程** 90](#_Toc207354907)

[**5.8.4** **基于RTD2.0.1 S32DS CT配置的K324 FreeRTOS工程** 90](#_Toc207354908)

[**5.8.5** **基于RTD3.0.0P07 EB配置的K338 GHS工程** 90](#_Toc207354909)

[**6.** **功能安全部分** 91](#_Toc207354910)

[**6.1** **FCCU相关** 91](#_Toc207354911)

[**6.1.1** **FCCU故障状态下外设引脚无输入输出功能问题** 91](#_Toc207354912)

[**6.2** **BIST相关** 91](#_Toc207354913)

[**6.2.1** **Trace时钟和CLKOUT需要设置成FIRC** 91](#_Toc207354914)

[**6.3** **故障注入相关** 91](#_Toc207354915)

[**6.3.1** **故障注入后再进行HSE服务调用** 91](#_Toc207354916)

[**6.4** **SAF相关** 91](#_Toc207354917)

[**6.4.1** **检查使用版本的已知Bug** 91](#_Toc207354918)

[**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** 92](#_Toc207354919)

[**6.4.3** **非锁步核不需要进行RCCUTEST\_RCCU\_CM7测试** 92](#_Toc207354920)

[**6.4.4** **SAF1.0.5之前版本可能出现看门狗sCheck失败的问题** 92](#_Toc207354921)

[**6.4.5** **SAF1.0.4版本Dcache和Flash EDMA sCheck失败问题** 92](#_Toc207354922)

[**6.4.6** **SAF1.0.4版本AB Swap固件下sCHECK FLASH检测失败问题** 92](#_Toc207354923)

[**6.4.7** **xRDC配置错误可能导致sCheck失败** 92](#_Toc207354924)

[**6.4.8** **Gmac模块进行sBoot前需完成初始化** 93](#_Toc207354925)

[**6.4.9** **其它** 93](#_Toc207354926)

[**6.5** **SAF开发license说明** 93](#_Toc207354927)

[**6.6** **应用笔记** 93](#_Toc207354928)

[**6.6.1** **S32K3xx\_Functional\_Safety\_Application\_Note** 93](#_Toc207354929)

[**6.7** **示例工程** 94](#_Toc207354930)

[**6.7.1** **基于K324的功能安全和信息安全集成Demo** 94](#_Toc207354931)

[**6.7.2** **基于K344和K358的功能安全Demo** 94](#_Toc207354932)

[**附录 A 文档中使用的缩略语** 95](#_Toc207354933)

1. **介绍**
   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. **免责声明**

以下内容为个人总结，可能存在理解有误的地方，仅供参考。

1. **硬件设计部分**
   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。

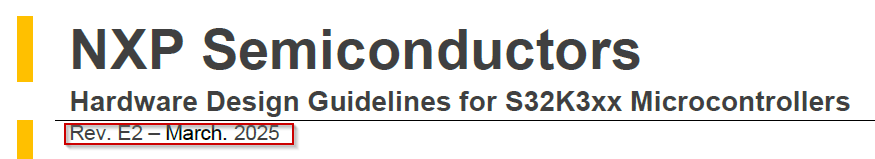
截至2025年8月26日：

1、最新版本数据手册为2025年7月发布的版本12，如下：



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：



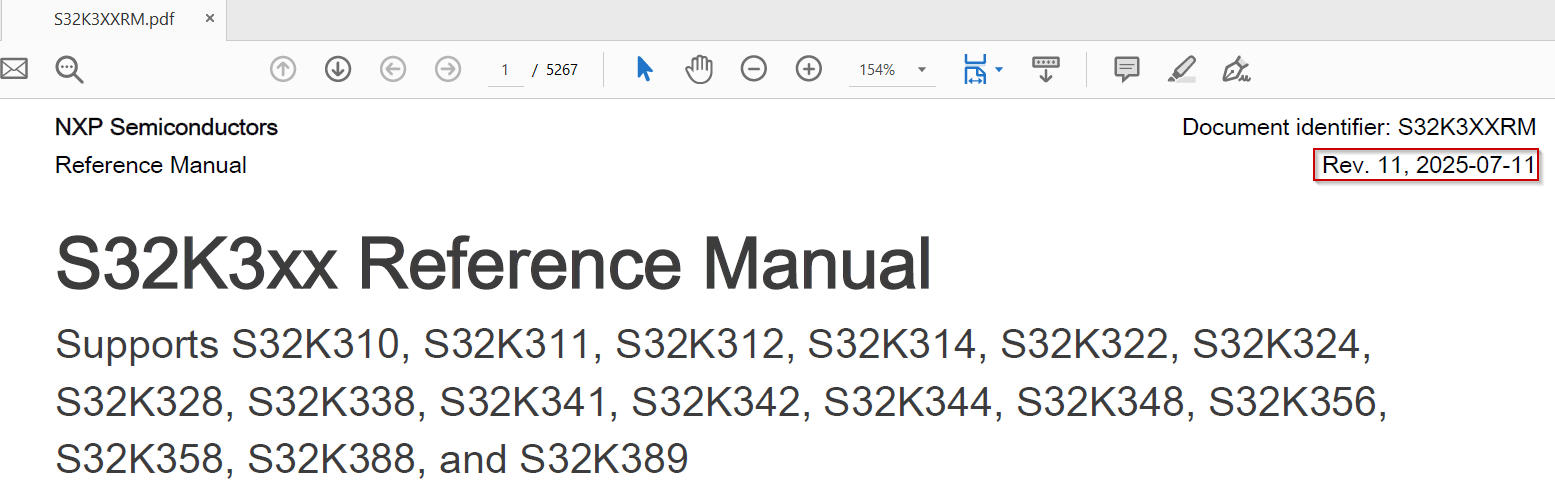
有些客户未遵循这两份文件，出现了问题。

* 1. **必须遵循参考手册**

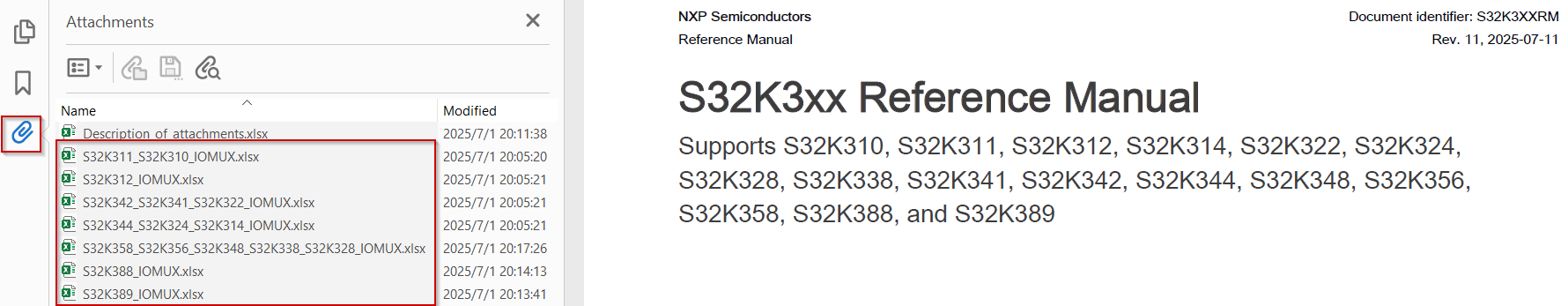
使用最新版本的参考手册，因为新版本会修复旧版本的Bug。

截至2025年8月26日：

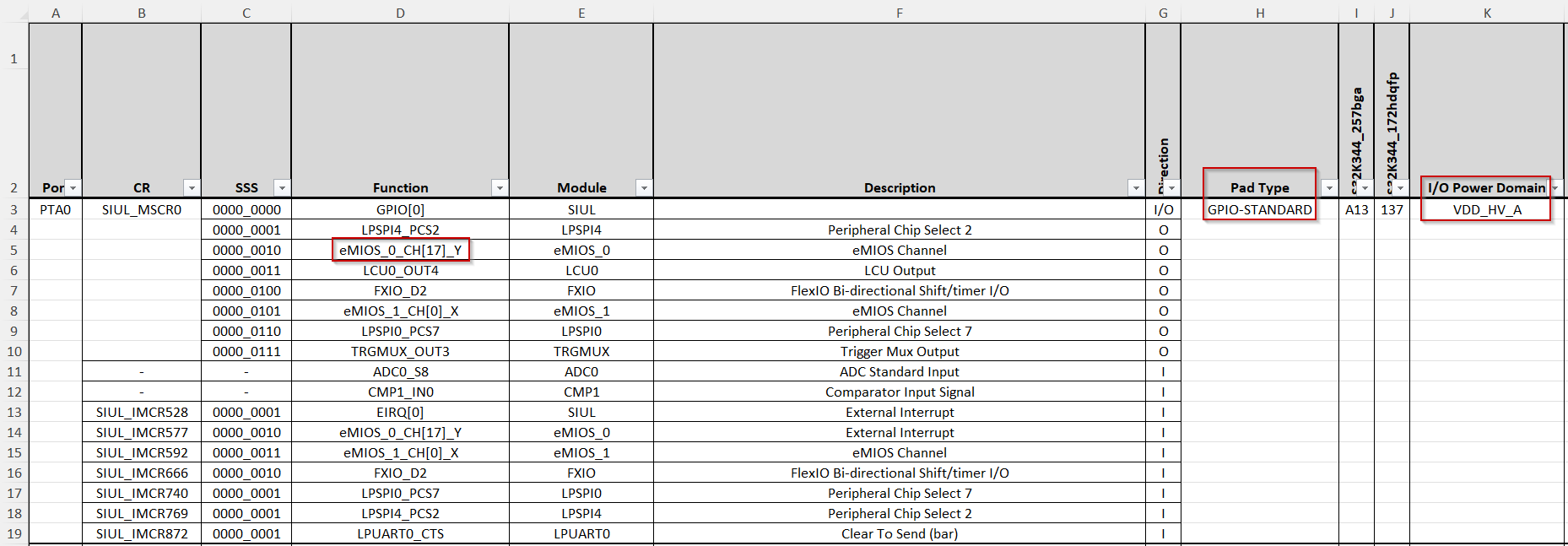
最新版本参考手册为2025年7月发布的版本11，如下：



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：



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。

* 1. **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。

* 1. **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。

* 1. **高速通讯引脚建议选择Fast或Medimu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）。

* 1. **高驱动能力引脚建议选择Fast或Medimu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。

* 1. **注意eMIOS模块不同类型通道的选择**

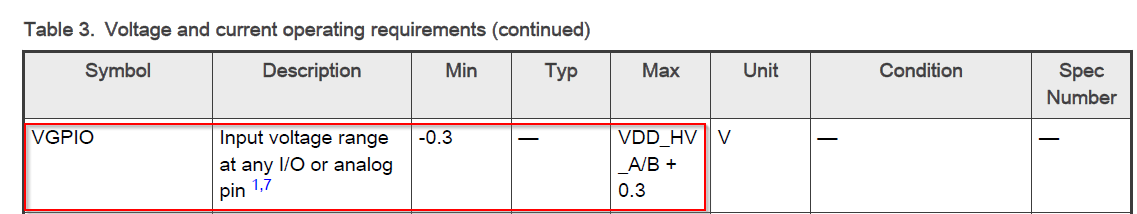
eMIOS有4种类型的通道，某些类型的通道不能实现特定的功能，在硬件设计上要充分考虑不同类型通道的差异。

* 1. **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。

* 1. **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 V的电压（包括通过串接电阻）接入到IO引脚，如下：



若违背，可能出现引脚内部电路损坏，或MCU工作异常。

* 1. **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。

* 1. **ADC硬件设计注意事项**
     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。

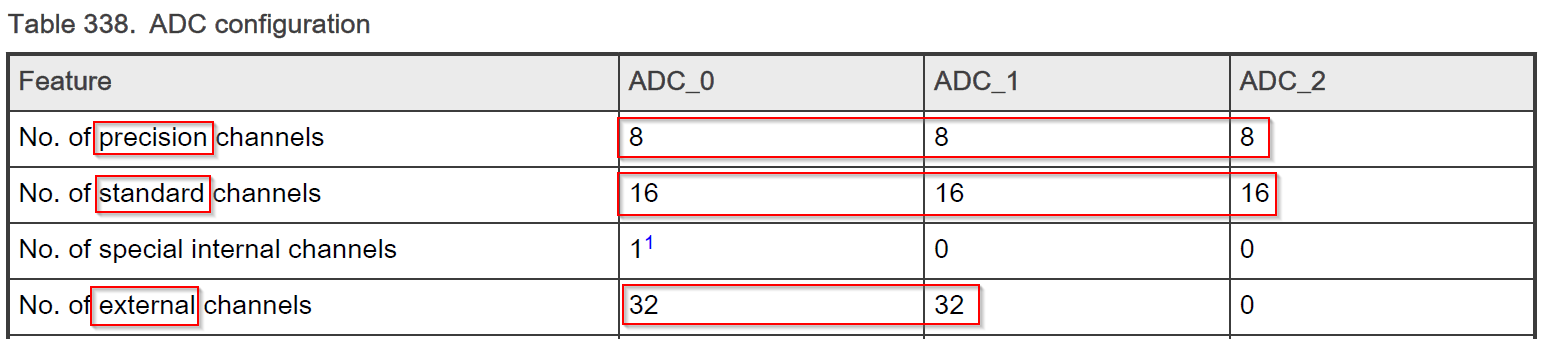


图 1 ADC的3种通道数目

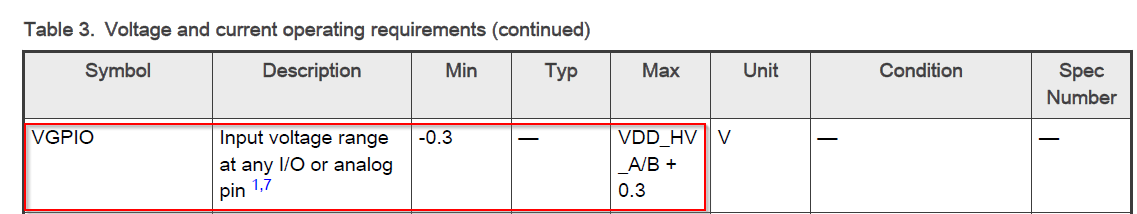
精度通道和标准通道的内部电气参数不同，如下是参数差异：



图 2 精度通道和标准通道内部参数对比图

* + 1. **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 V的电压（包括通过串接电阻）接入到ADC引脚，如下：



若违背，可能出现引脚内部电路损坏，或MCU工作异常。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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.

* + 1. **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。

* + 1. **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

* + 1. **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。

* 1. **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：

A screenshot of a computer error

AI-generated content may be incorrect.

A diagram of a computer

AI-generated content may be incorrect.

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”

* 1. **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。

* + 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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

1. **软件开发部分**
   1. **电源和低功耗（休眠唤醒）部分**
      1. **Last Mile的配置**

配置页面中Last Mile Regulator Enable需使能。

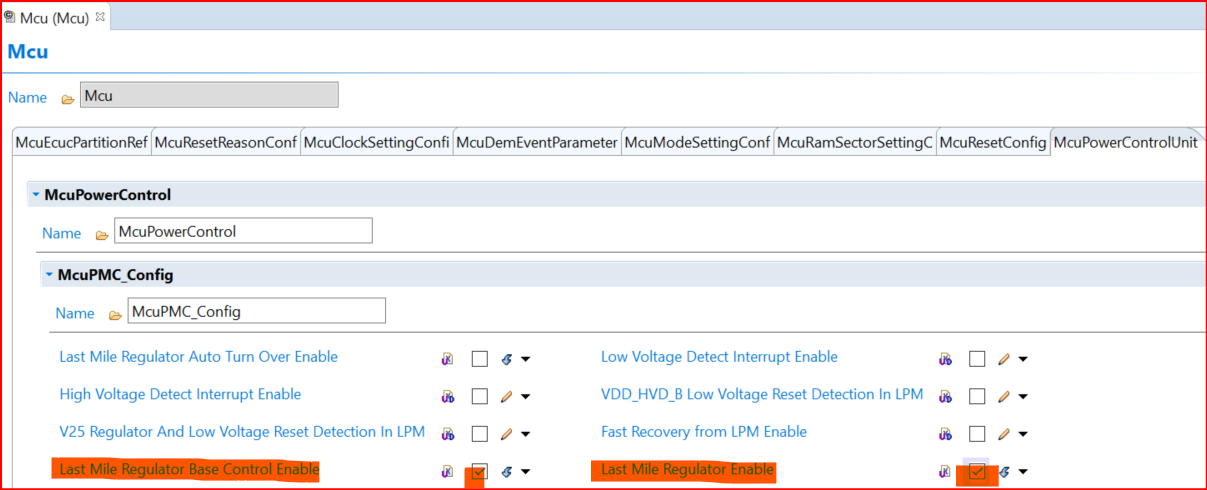
若使用外部三级管提供1.5V，配置页面还需使能Last Mile Regulator Base Control选项。

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，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：



* + 1. **Pad keeping**

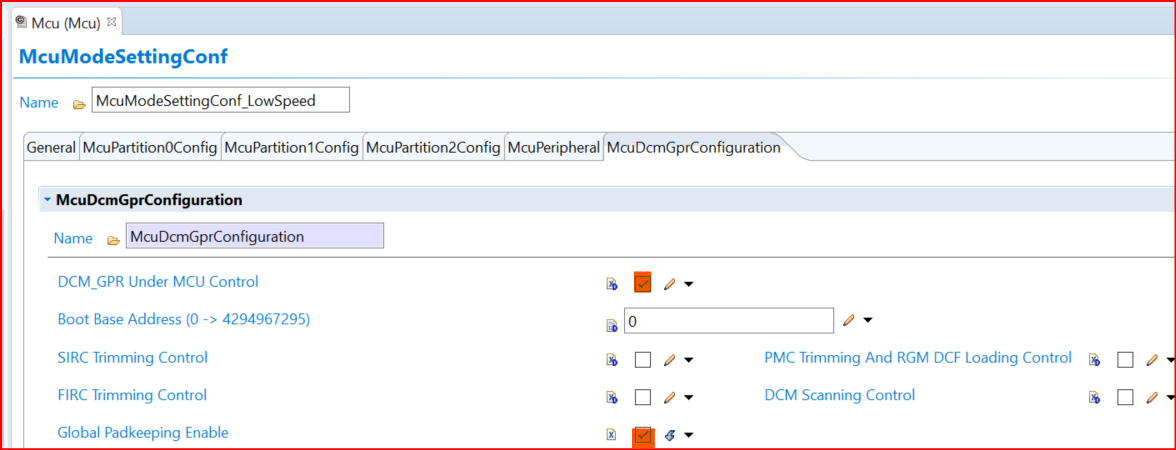
一、建议使能Pad Keeping功能。

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。

* + 1. **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

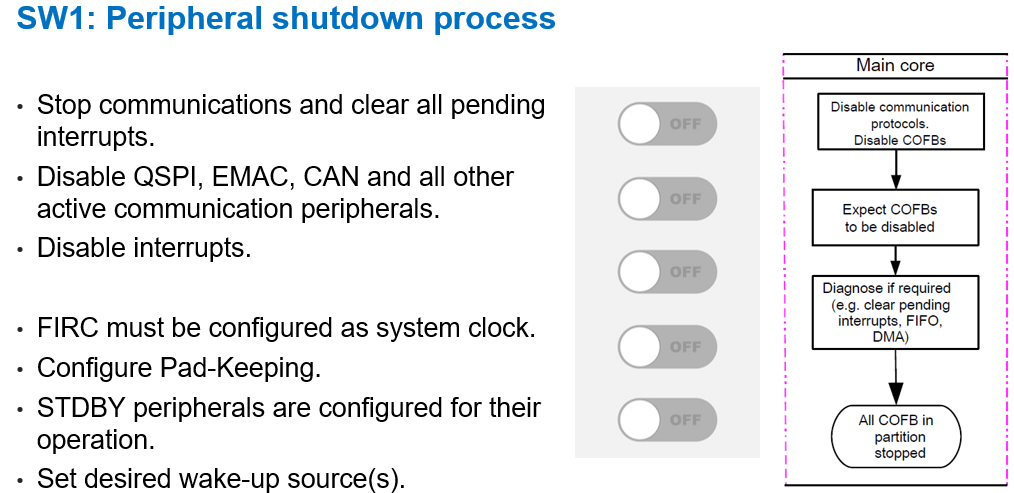
* + 1. **进入休眠过程需要关闭外设并关闭中断**

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。

A screenshot of a document

AI-generated content may be incorrect.



* + 1. **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。

* + 1. **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. 等待新版sBAF发布，使用出厂烧录新版sBAF的芯片（预计2026年下半年），或通过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版本预计2025年8月底发布（AB SWAP类型修复bug的sBAF还不确定发布时间）。若未在官网发现，可联系NXP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。

* + 1. **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。

* 1. **时钟**
     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。

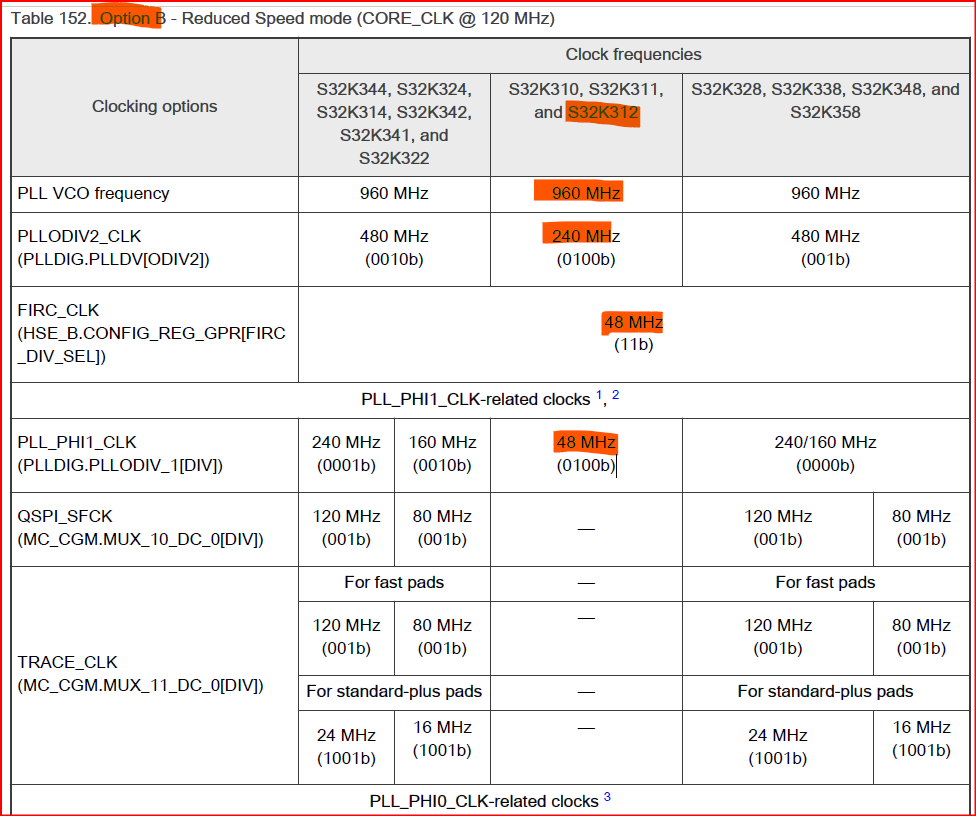
包括参考手册该选项的各个时钟部分设置，不仅仅是系统时钟。

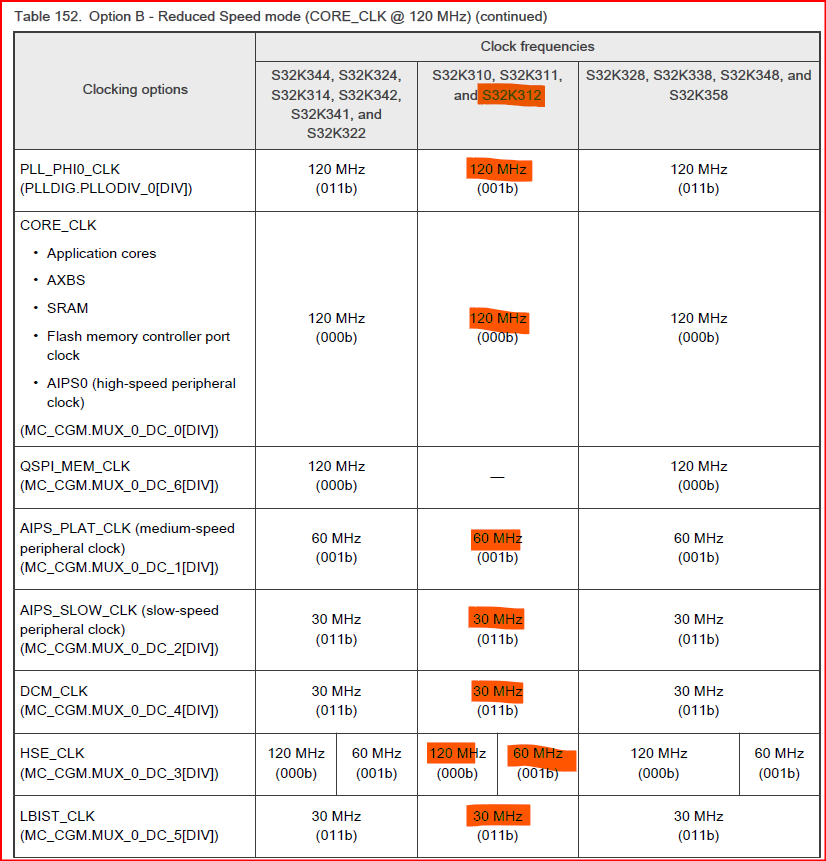
不遵守可能出现各种异常问题。例如，Fee\_write不能写数据到FLASH。

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问题。



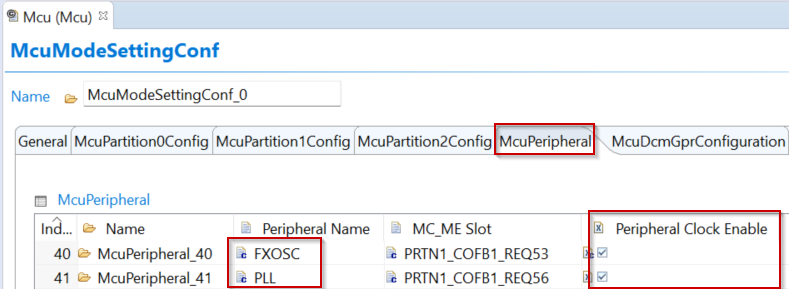


* + 1. **使用的外设门控时钟要配置为使能**

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。

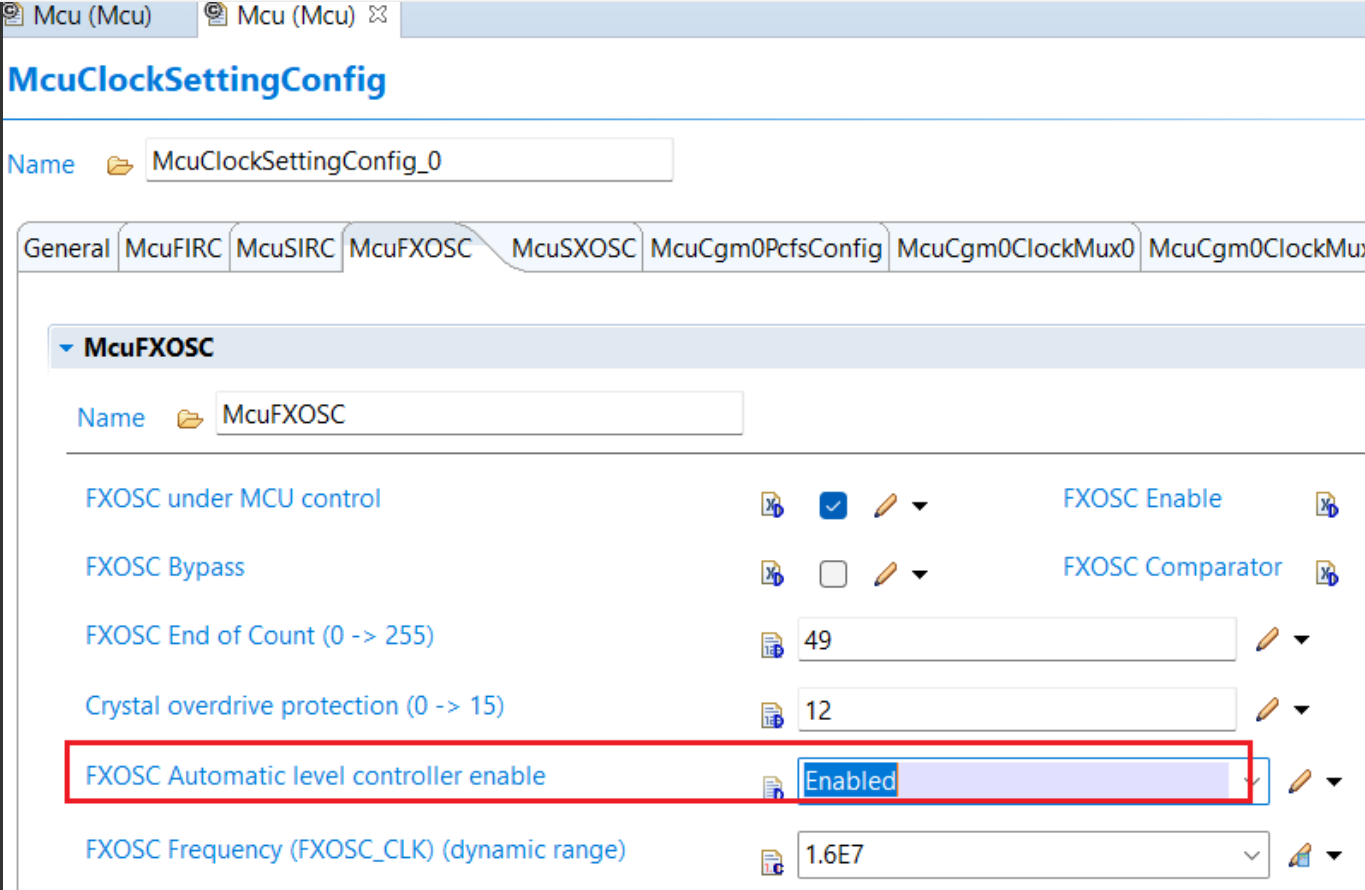
* + 1. **快速外部晶振FXOSC的自动增益设置**

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：

A screenshot of a computer program

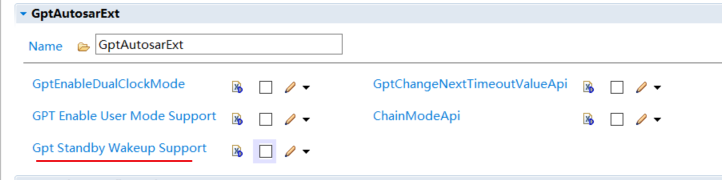
AI-generated content may be incorrect.

1. 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。
   * 1. **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= current RTCCNT+Wake Period

此时应用上要考虑CNT寄存器0xFFFFFFF->0X0的情况。



* + 1. **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。

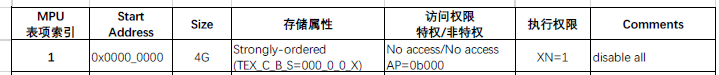
因此，若使用IAR编译器，该措施需添加。

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

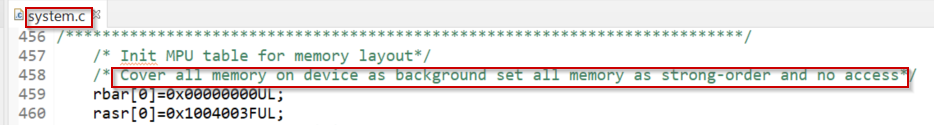
建议使用S32DS的GCC编译器或GreenHills的客户也在合适的机会按照RTD4.0.0添加该措施。因为不能保证使用这两款编译器就一定不出问题。

* 1. **M7内核**
     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。



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。



1. 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。

1. MPU各个区域设置读/写/访问权限和大小设置是否合适。

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

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。

1. MPU重叠区域以高区域（Region）号设置为准。
2. 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）。

1. 若MPU中将一段SRAM属性设置为CACHEABLE，同时也设置了SHAREABLE属性，则CACHABLE实际未启用，需去掉SHAREABLE属性实现CACHEABLE特性。
2. 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：



在RTD2.0.3及之前版本，未实现该建议，如下：



1. 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。
2. MPU多次设置的注意事项。
3. 设置MPU前建议关闭全局中断，设置完成后再打开全局中。
4. 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。
5. AUTOSAR OS也会设置MPU（需确认OS设置的MPU正确）。FreeRTOS不会设置MPU。
6. MPU依据后设置为准（生效）。
7. 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。

* + 1. **ARM推测性访问（**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（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UTEST区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。

* + 1. **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0xE000ED28），大小为四字节。

2、内存管理故障地址寄存器MMFAR（地址为0xE000ED34）

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0xE000ED38）

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。

* + 1. **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。

* 1. **存储器**
     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。

* + 1. **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。

* + 1. **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。

* + 1. **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：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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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)

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* 1. **复位相关问题**
     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。

* + 1. **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。

* 1. **ADC**
     1. **需遵守K3 参考手册的要求**

见参考手册《S32K3xx Reference Manual》第60章。

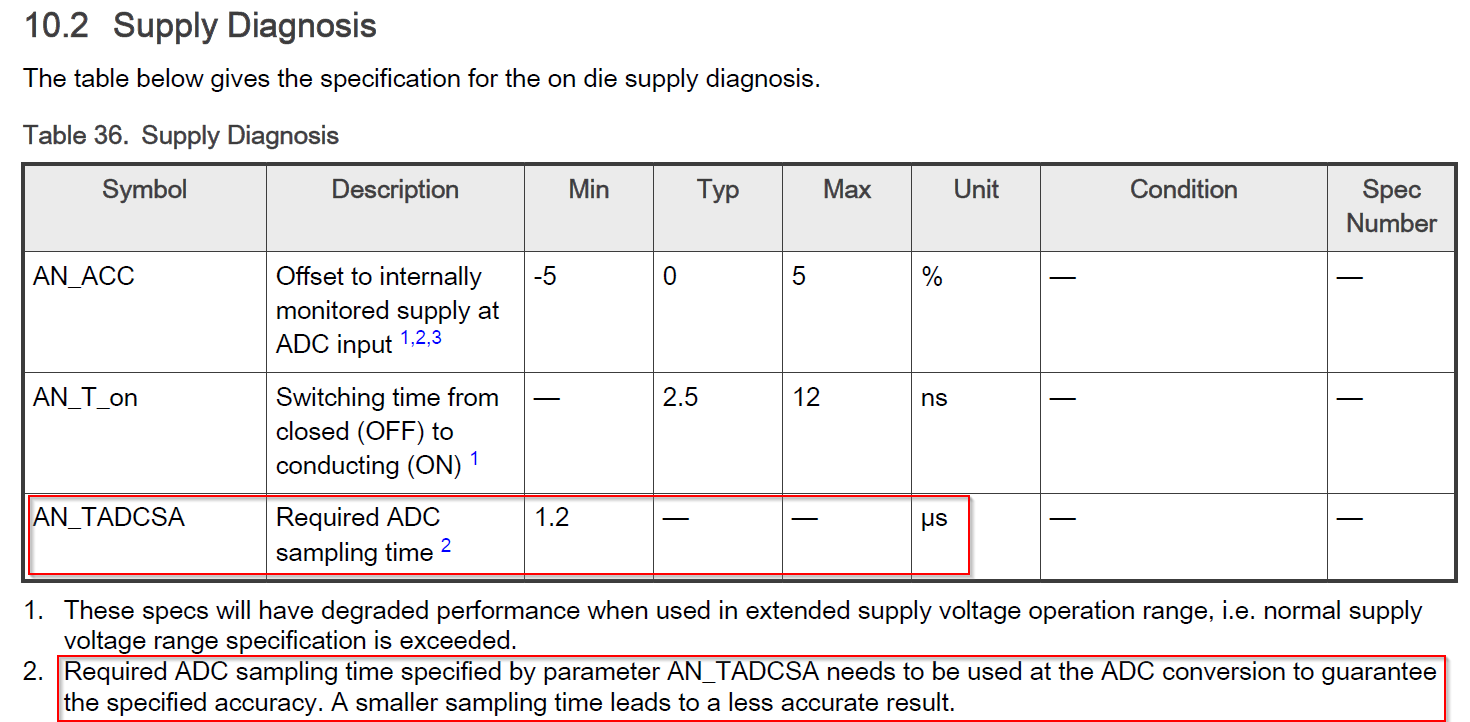
请使用最新版本手册。目前最新版本是2025年7月发布的第11版。

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。

* + 1. **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：

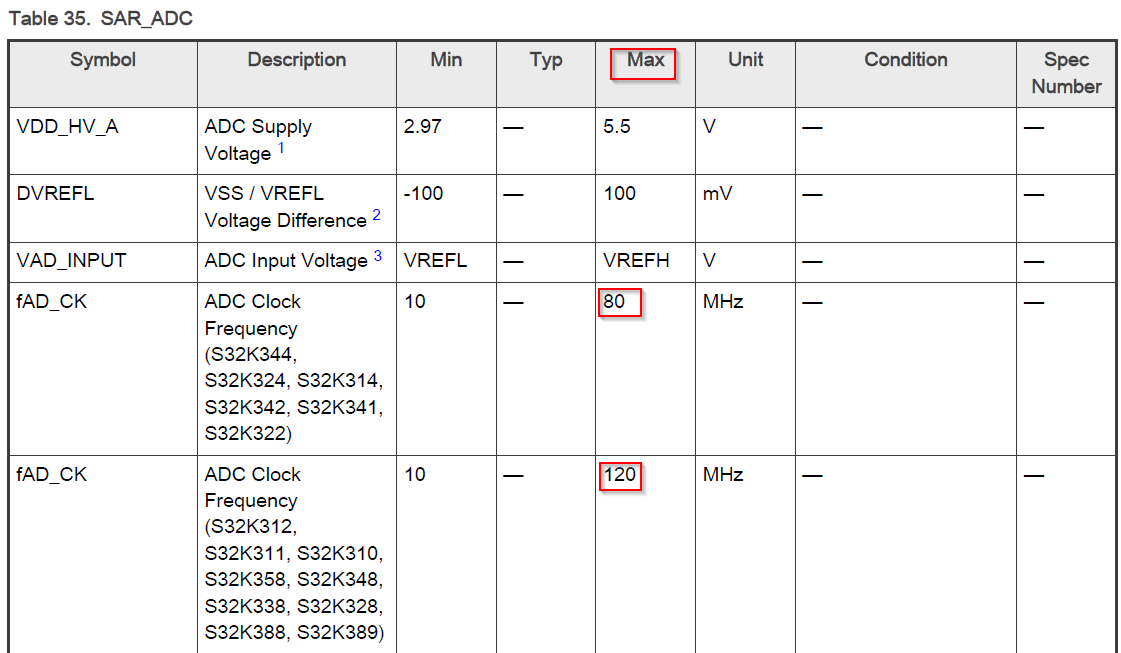


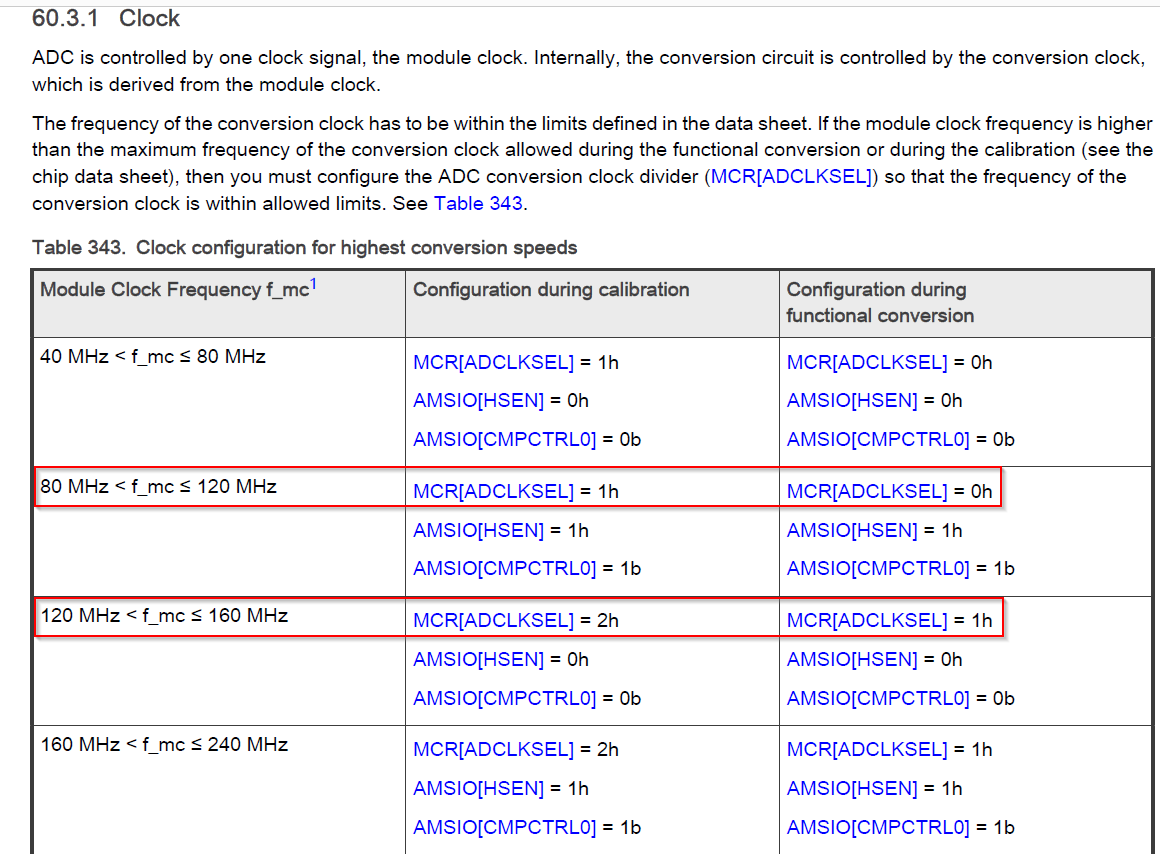
建议根据板卡实测情况，留适当余量。

温度通道建议使能预采样。

* + 1. **ADC采样时钟的要求**

见数据手册表格35的要求：

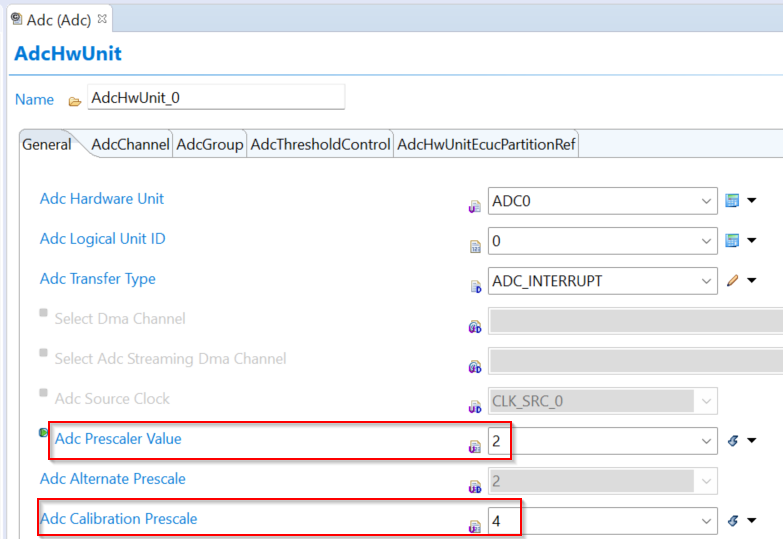




例如，K324,ADC转化时钟最高80M，校准时钟最高40M。

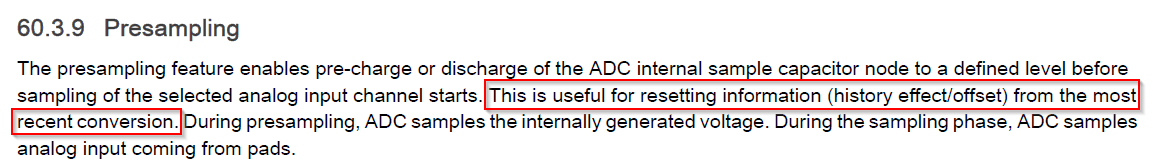
超过如上时钟频率，可能导致校准失败，数据采集异常，甚至损坏MCU。

如下是K324的最高转换频率设置：



* + 1. **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：



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。

* + 1. **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。

* + 1. **K310和K311 ADC使用注意事项**

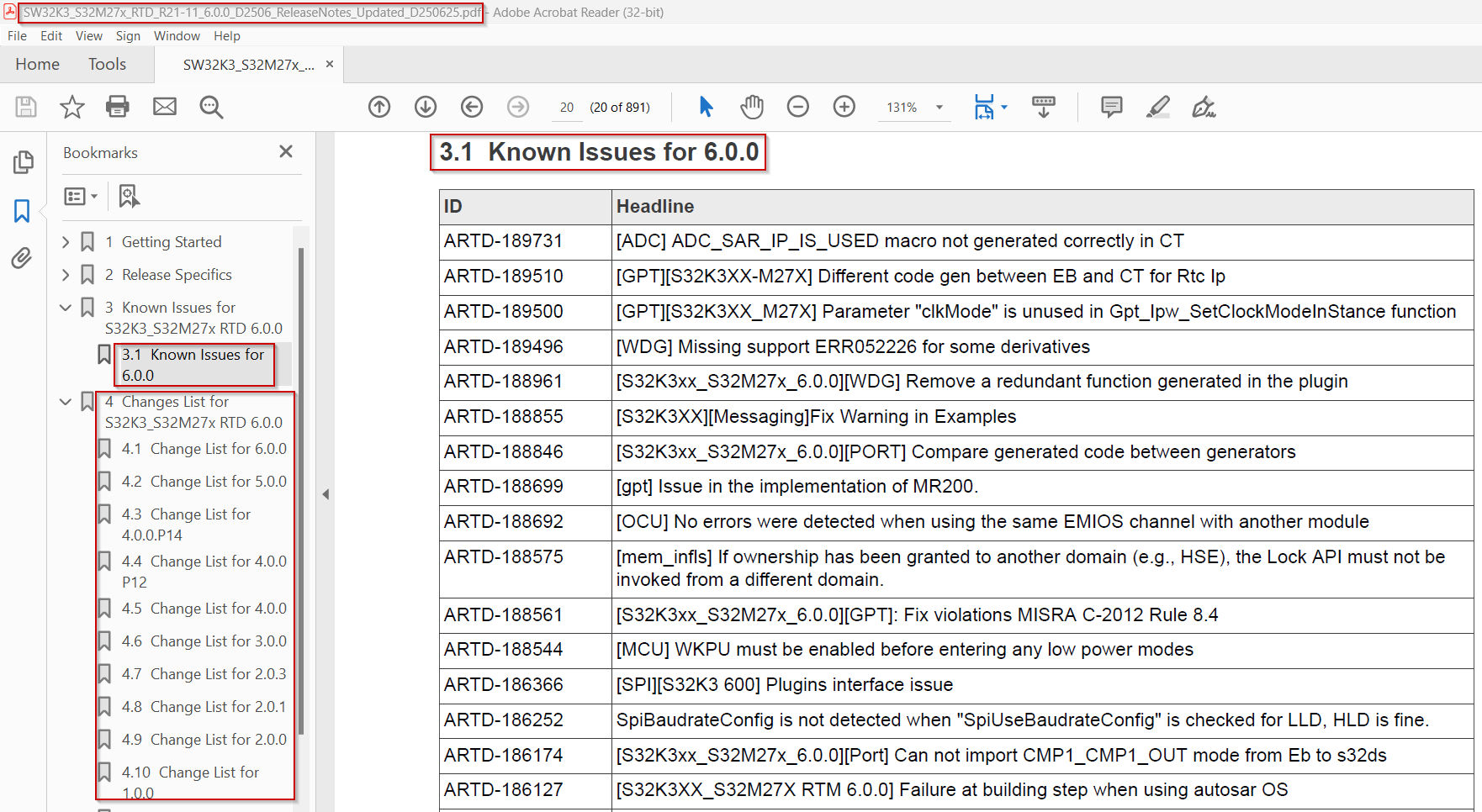
由于RTD4.0.0和RTD5.0.0的Bug，使用K310/K311的部分ADC通道时，若不是默认GPIO映射，需要修改DCMRWF4的对应位，否则无法采集这些通道。

* + 1. **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。

****

* + 1. **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。

* + 1. **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。
   * 1. **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。

* + 1. **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。

* 1. **CAN**
     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，并且RTD版本为5.0.0及之前版本，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，如上邮箱可正常使用，因为施加了软件措施，不会丢帧。

详见勘误表（ERRATA）的条目**ERR052438**。

* + 1. **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。

* + 1. **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。

* + 1. **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)。

* + 1. **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。

* + 1. **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。

* 1. **SPI**
     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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* 1. **I2C**
     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。

* + 1. **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。

* 1. **以太网**
     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。

* + 1. **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。

* 1. **eMIOS**
     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**》

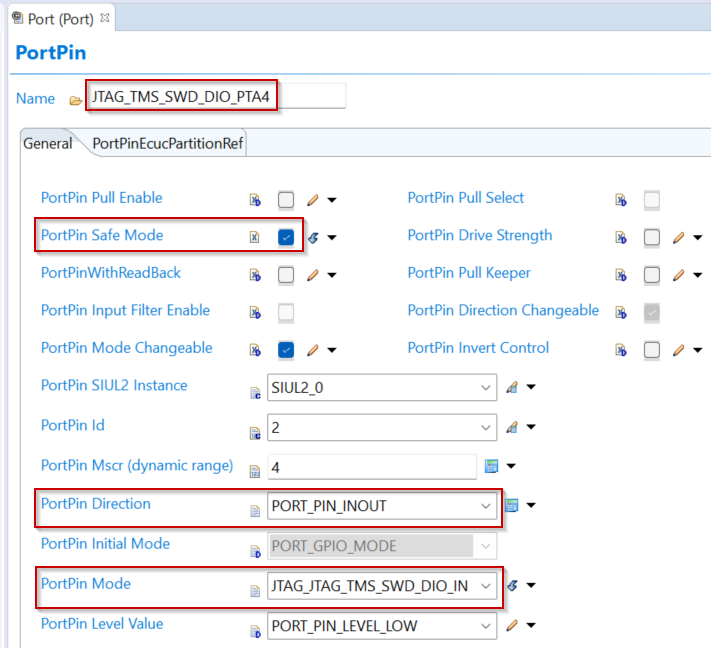
可联系NXP FAE获取修复代码和如上文档。

* 1. **端口引脚设置**
     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。

A screenshot of a computer

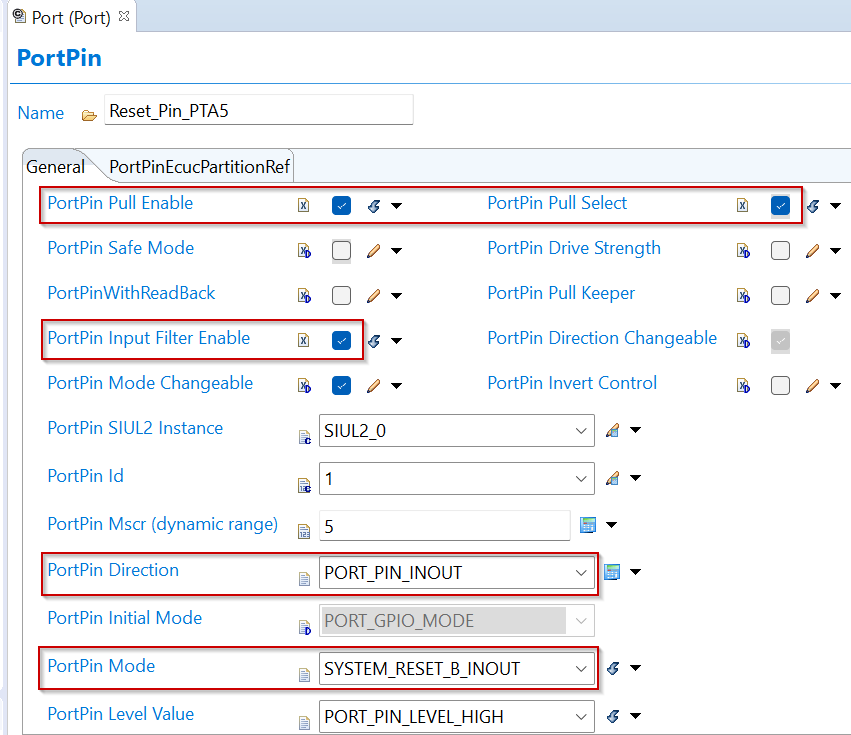
AI-generated content may be incorrect.

* + 1. **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。



* + 1. **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。

* 1. **中断相关**
     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的函数）放入ITCM中执行。

* + 1. **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。

* + 1. **中断向量表需要128字节对齐**

中断矢量表在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。

* + 1. **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。

* 1. **RTD相关**
     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，增加了CAN Enhanced FIFO和部分MB 共用的勘误表（ERRATA）的规避代码。

推荐使用新版本RTD。目前最新版本RTD是RTD6.0.0。

* + 1. **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。

* + 1. **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)

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>

* + 1. **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》。

* + 1. **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。

* + 1. **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。通过该方法能快速找到问题，加快问题解决时间。

* + 1. **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为了，

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：

A close up of text

AI-generated content may be incorrect.



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。

* + 1. **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。

* + 1. **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采是否采用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。

* 1. **编译器相关**
     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。

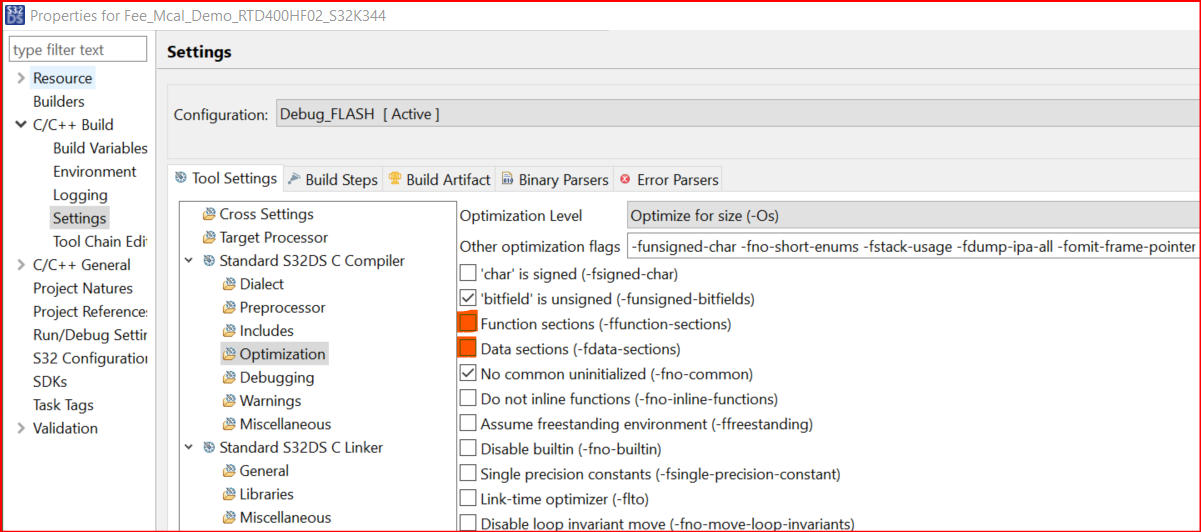
推荐按照该文件中的说明设置。

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。

A screenshot of a computer

AI-generated content may be incorrect.

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。

A screenshot of a computer

AI-generated content may be incorrect.

* + 1. **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。

* + 1. **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。

* 1. **调试器相关**
     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。

* + 1. **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。

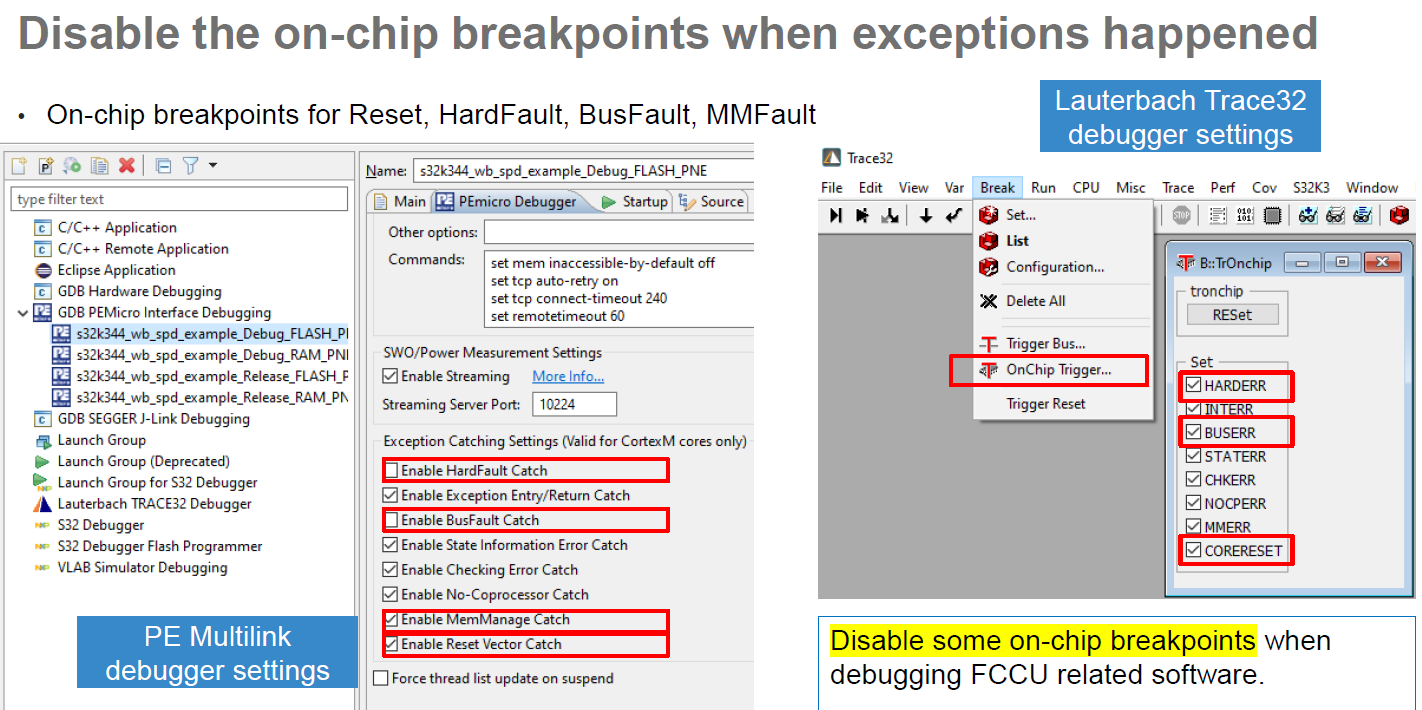
例如通过attach，或点灯，翻转IO等方式确认。

* + 1. **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。



* + 1. **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。

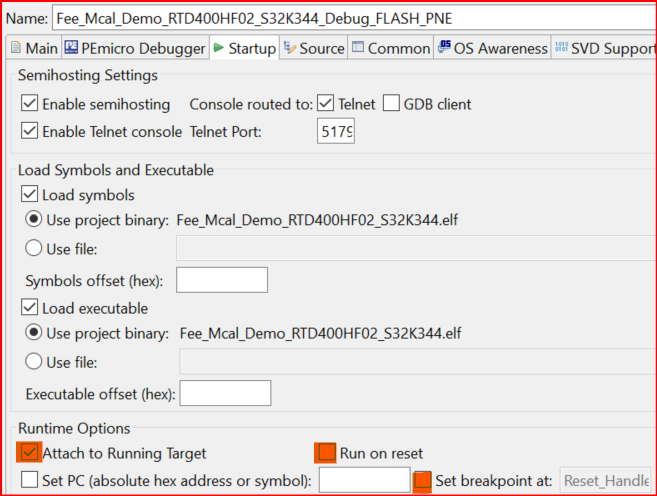
解决办法是检查程序中是否对RAM按照8字节写的方式进行了ECC初始化。

* + 1. **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：



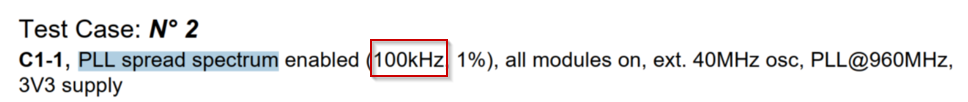
* 1. **EMC**
     1. **请遵循硬件设计指南中关于EMC的建议**

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。

* + 1. **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：

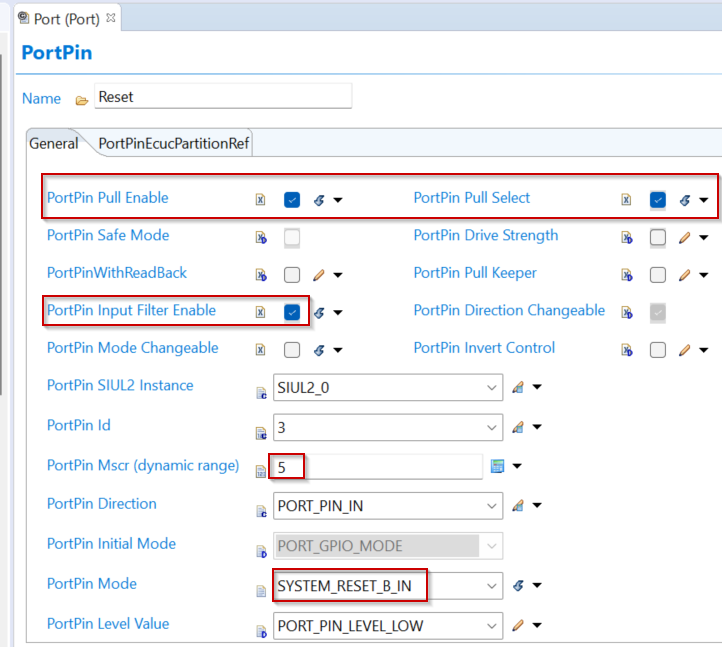


* + 1. **使能Reset引脚的滤波功能**

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。



* + 1. **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。

* + 1. **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。

* + 1. **解决ESD不通过的常用措施**

1. 禁止FXOSC的自动增益（ALC\_D）。
2. RESET引脚增加RC或者软件配置输入加Filter。
3. 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。
4. 配置PLL中ULKCTL的Unlock Range；手动修改ULKCTL寄存器，尝试不同的配置值，观测能否通过ESD试验。
5. 晶振布局需要遵守硬件设计指南的要求。
6. 使用6层板设计。
7. 产品使用金属外壳。
   * 1. **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。

* 1. **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。

若是芯片本身Bug，在勘误表一般会给出解决方案（Workround）。

如果未检查勘误表，未采取软件措施规避问题，当产品量产后出现问题，会带来很严重的后果。

请使用最新版本的勘误表。

截至2025年8月27日，K3各芯片最新勘误表为2025年7月的版本。以K3X4为例，最新勘误表版本如下：



* 1. **性能提升**
     1. **使能CACHE会显著提升代码执行速度**

使能I CACHE和D CACHE会有较大的性能提升。

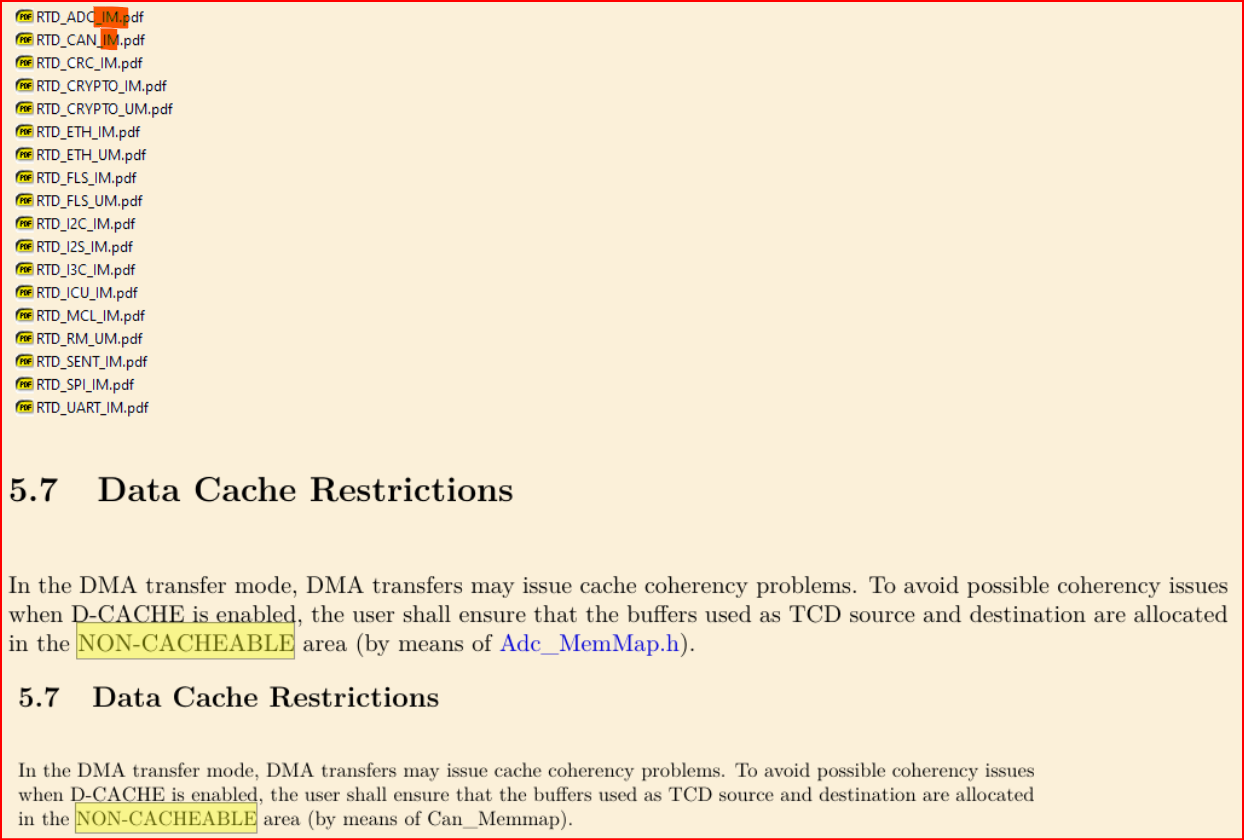
若出现MCU的CPU 负荷高问题，打开CACHE会明显降低CPU 负荷。

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

* + 1. **D-CACHE使能注意事项**

不必担心D-CACHE使能后带来的问题，只要遵循了D-CACHE使用的注意事项。

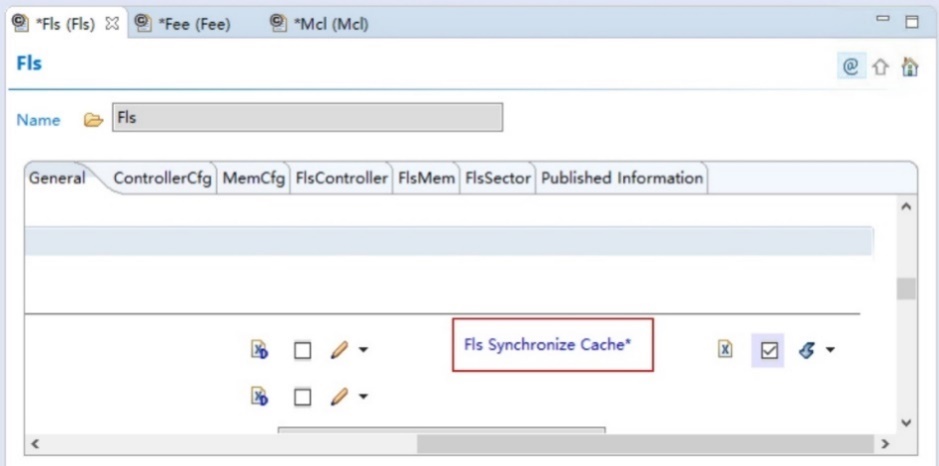
需要遵循RTD软件中每个模块集成手册（IM）有相关说明，见下图：

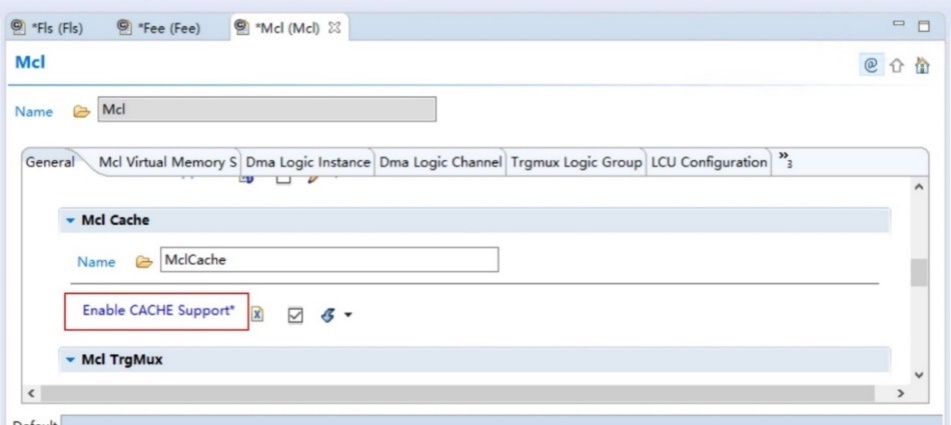


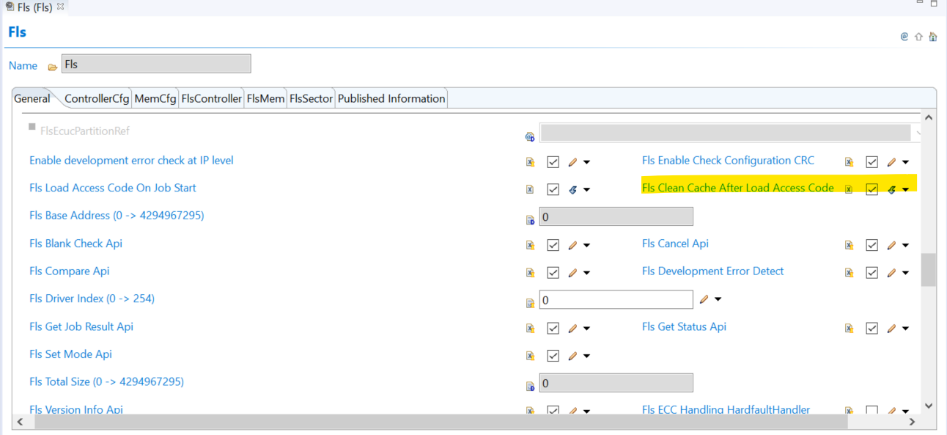
下面介绍几个主要注意事项：

1、使能CACHE后，在使用DMA搬运数据时需要注意。由于CACHE的使用，CACHEABLE特性的RAM数据内容不会实时更新，而DMA搬运的源/目的地址就不能选用CACHEABLE的RAM，需选用NON CHACHEABLE、TCM、SHAREABLE RAM，否则搬运的数据/M7核读取的数据并非期望的数据。

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。

* + 1. **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，

则需要在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。

* + 1. **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CACHE情况下，不建议打开FLASH Prefetch，可能出现Hardfault。

* + 1. **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\_Linker\_File\_Modification\_Example\_Prj.zip》

* + 1. **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。

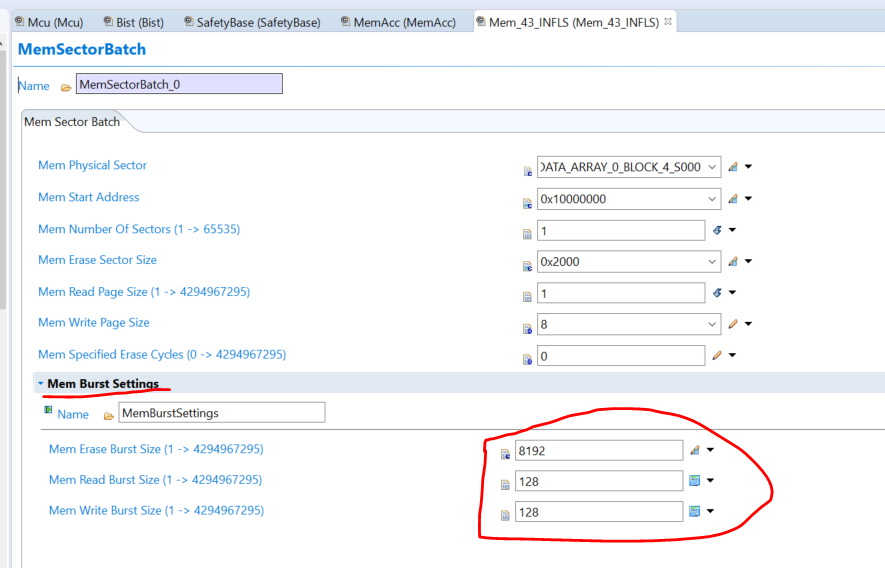
若客户对代码进行充分的测试未发现异常，也可以使用O1或O2。

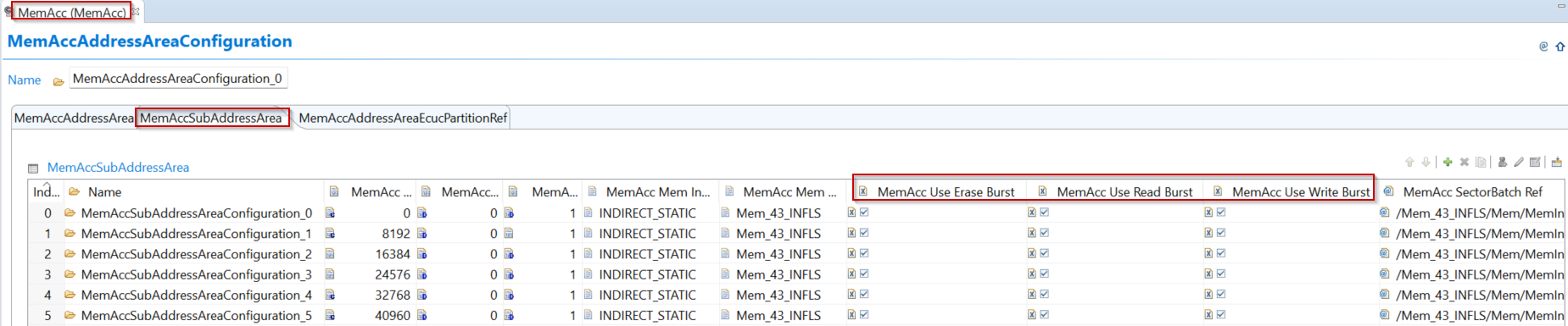
* + 1. **使用GHS或IAR编译器**

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。

* + 1. **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：



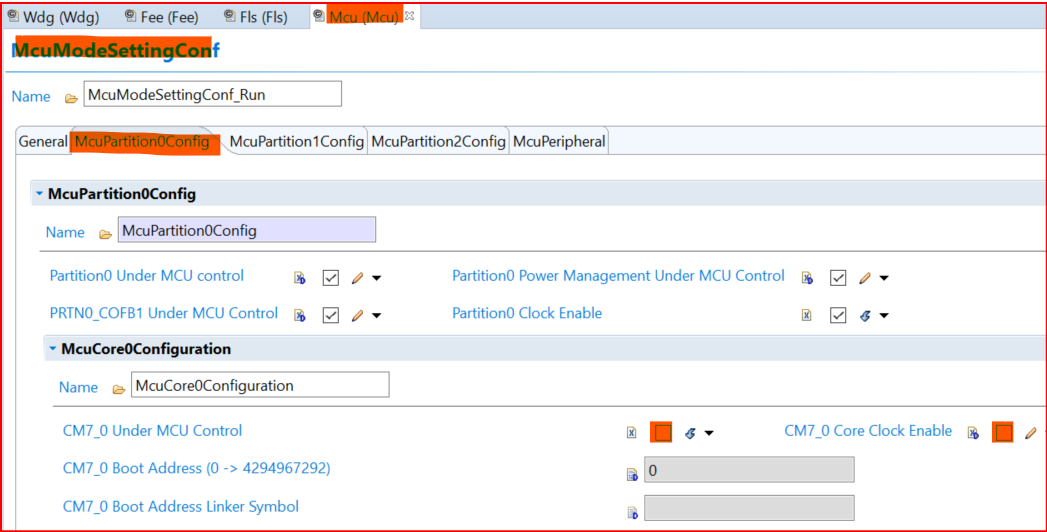


* 1. **其它**
     1. **初始化时间长问题**

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”。

* + 1. **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。

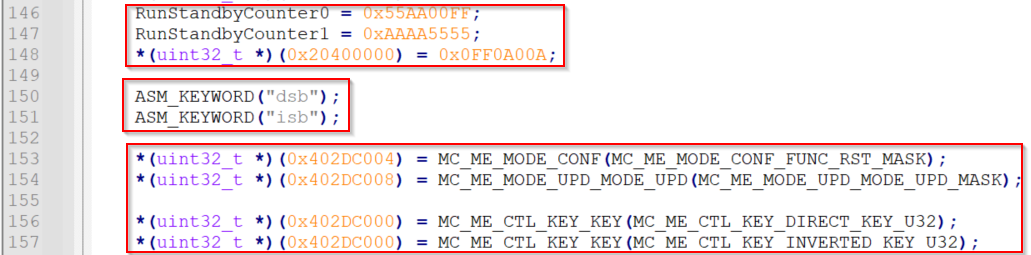
* + 1. **通过DSB和ISB解决代码非顺序执行问题**

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：



* + 1. **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().

* + 1. **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。
   * 1. **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。

* + 1. **焊接问题**

检查是否存在虚焊、短路焊接等问题。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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 = .;

* + 1. **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)

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

1. **信息安全(HSE)部分**
   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。

* 1. **HSE固件安装注意事项**

一、HSE固件安装的两种方法

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6月及之后版本）。

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，K388除外）或Option B（60M情况），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：

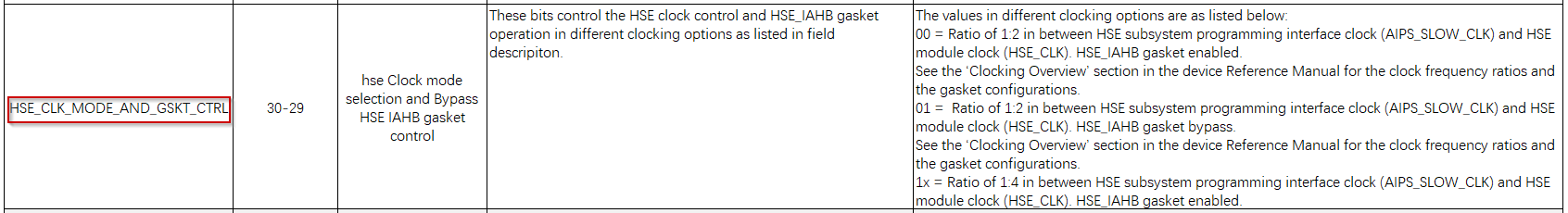


图1

具体说明：

1. 当K312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。

1. 当K312设置HSE时钟为60M时，不需要写DCF。
2. 当K328/338/348/358设置HSE时钟为120M时，需要写DCF，将上图30-29bit写入11或10。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。

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。

* 1. **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。

* 1. **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6月27日版本，文件名称如下：**s32k3\_hse\_lib\_rtd400hf01\_20250627.7z**

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。

* 1. **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。

* 1. **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) )

{

;

}

* 1. **安装HSE固件情下，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）。

* 1. **安装HSE固件情下，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。

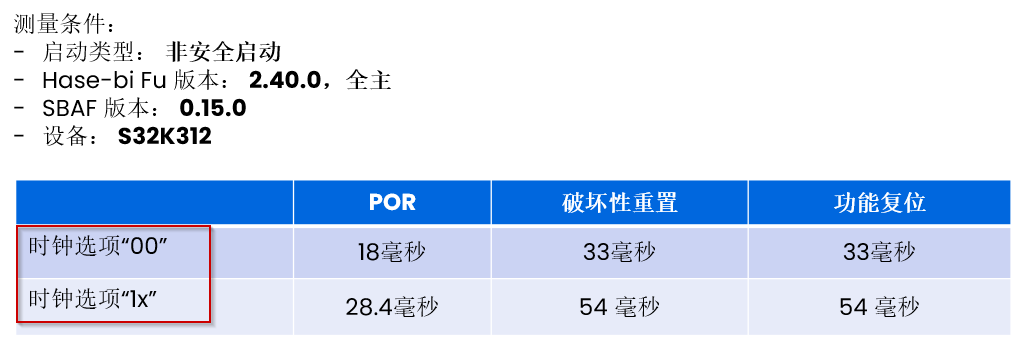
注意：BLOCK0和UTEST同属一个BLOCK。若HSE对UTEST读操作，那么不仅M7核不要对UTEST区进行写操作，也不要对BLOCK0进行擦写操作。

RTD6.0.0代码加入了信号量机制，可以不用判断HSE GPR3寄存器。

* 1. **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：



* 1. **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。

* 1. **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。

* 1. **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（0x00400000~0x004FFFFF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。

* 1. **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。

* 1. **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，

则需要在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。

* 1. **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。

* 1. **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

* 1. **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。

* 1. **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）。

* 1. **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。

* 1. **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。

1. 时钟配置未严格遵循K3参考手册的要求的几种配置。
   1. **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。

* 1. **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。

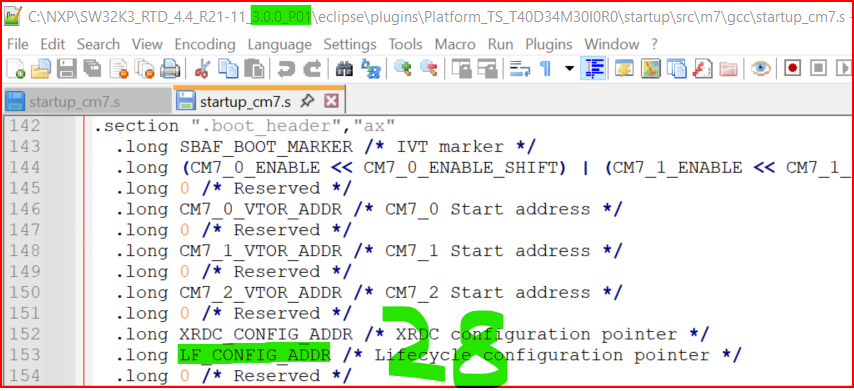


图1

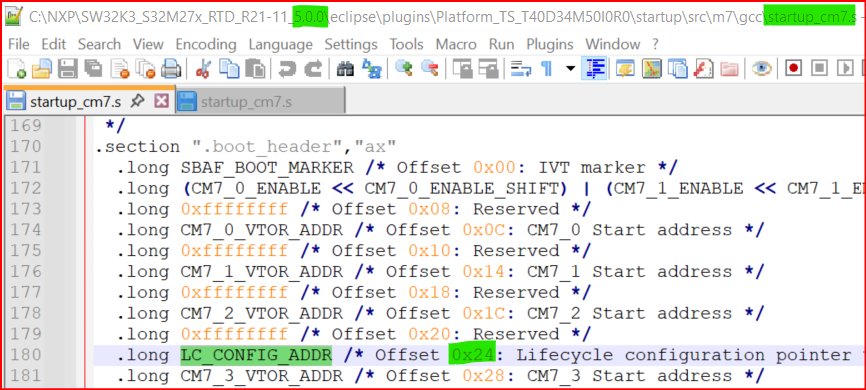


图2

* 1. **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。

* 1. **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。

* 1. **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，所以不方便SPI初始化，进而不能对FS26进行喂狗操作。

解决方案：NXP 应用工程师会在2025年9月底前提供一个解决方案（即使用裸代码方式进行SPI初始化和FS26喂狗操作），解决这个问题。

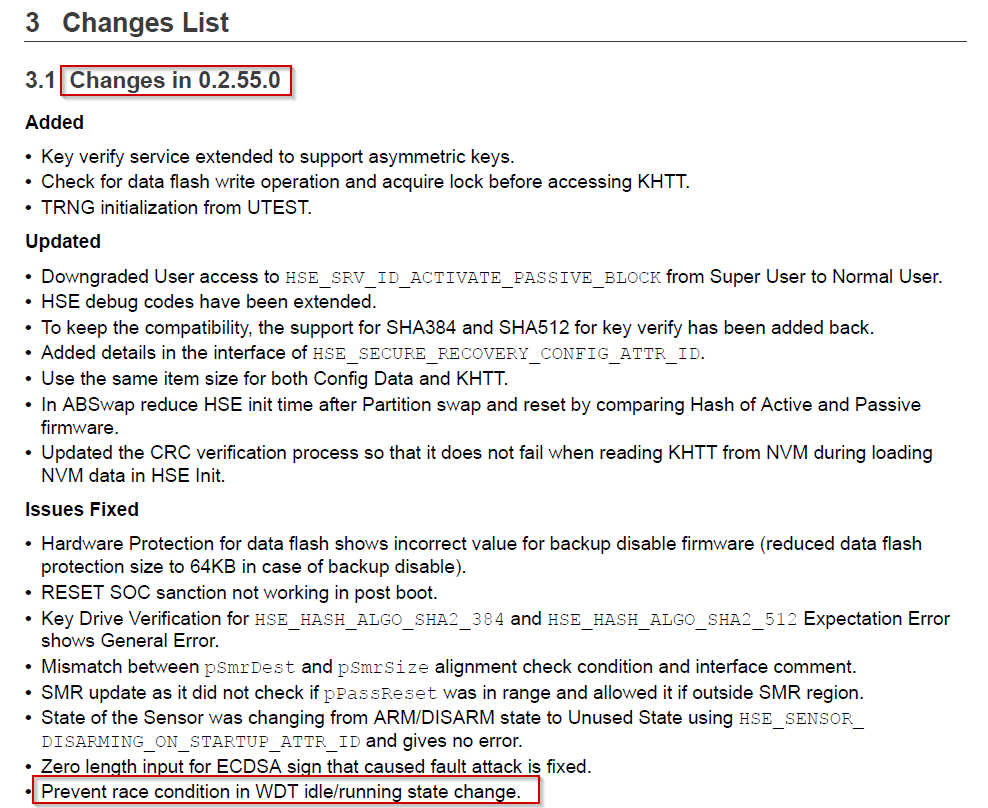
         若板卡使用其它SBC或外部看门狗，也需考虑喂狗问题。

         若HSE固件升级时，没有及时喂狗，可能导致HSE固件被擦除。

* 1. **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：



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。

* 1. **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。

* 1. **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。

* 1. **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。

* 1. **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。

* 1. **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。

* 1. **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。

* 1. **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》。

1. 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。

* 1. **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。

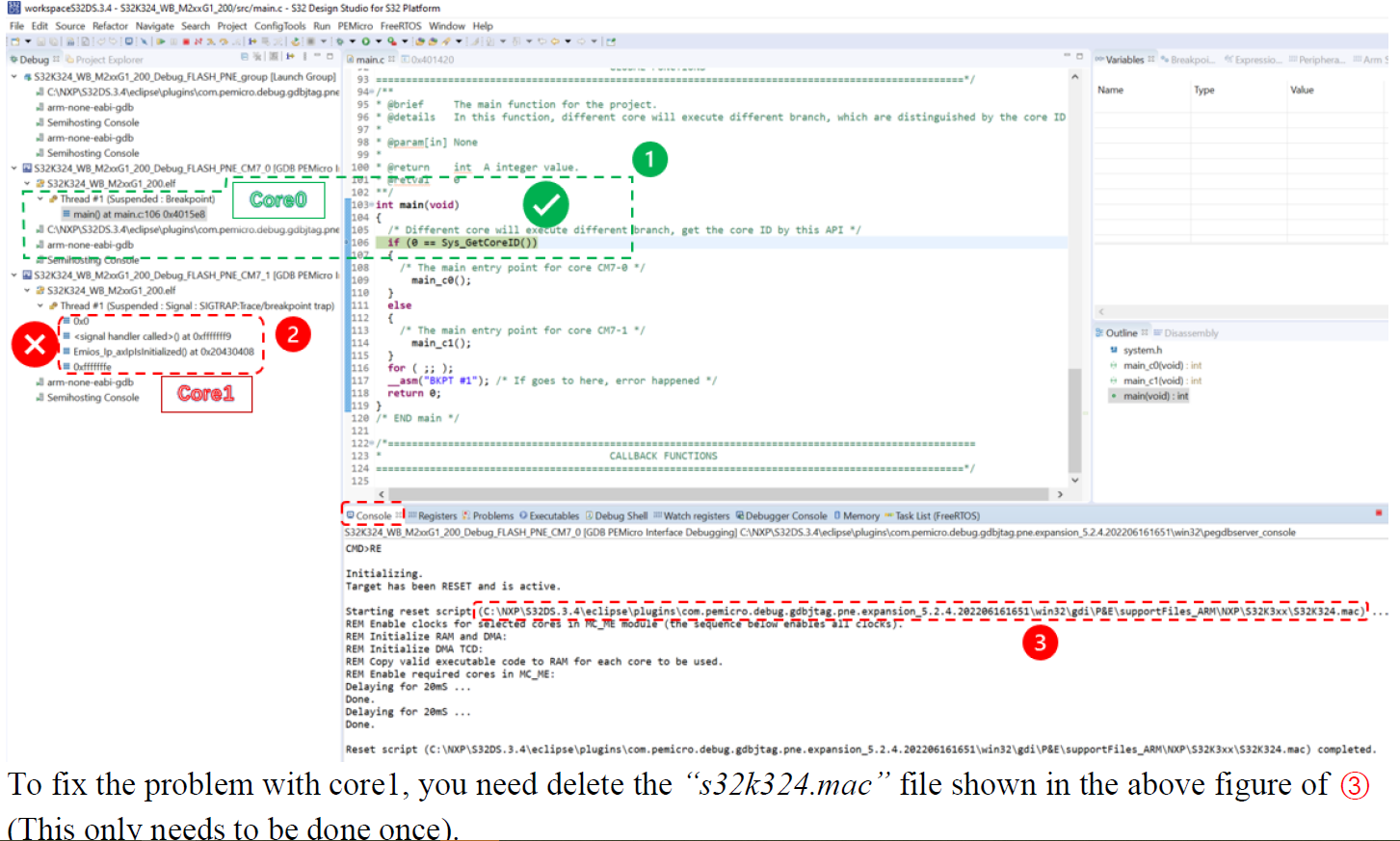
1. **多核部分部分**
   1. **共享存储器CACHE问题**

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。
   1. **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。



* 1. **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，否则出现单elf工程从核再次初始化RAM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

在其它版本RTD，没有此问题，可忽略此项检查。

* 1. **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。

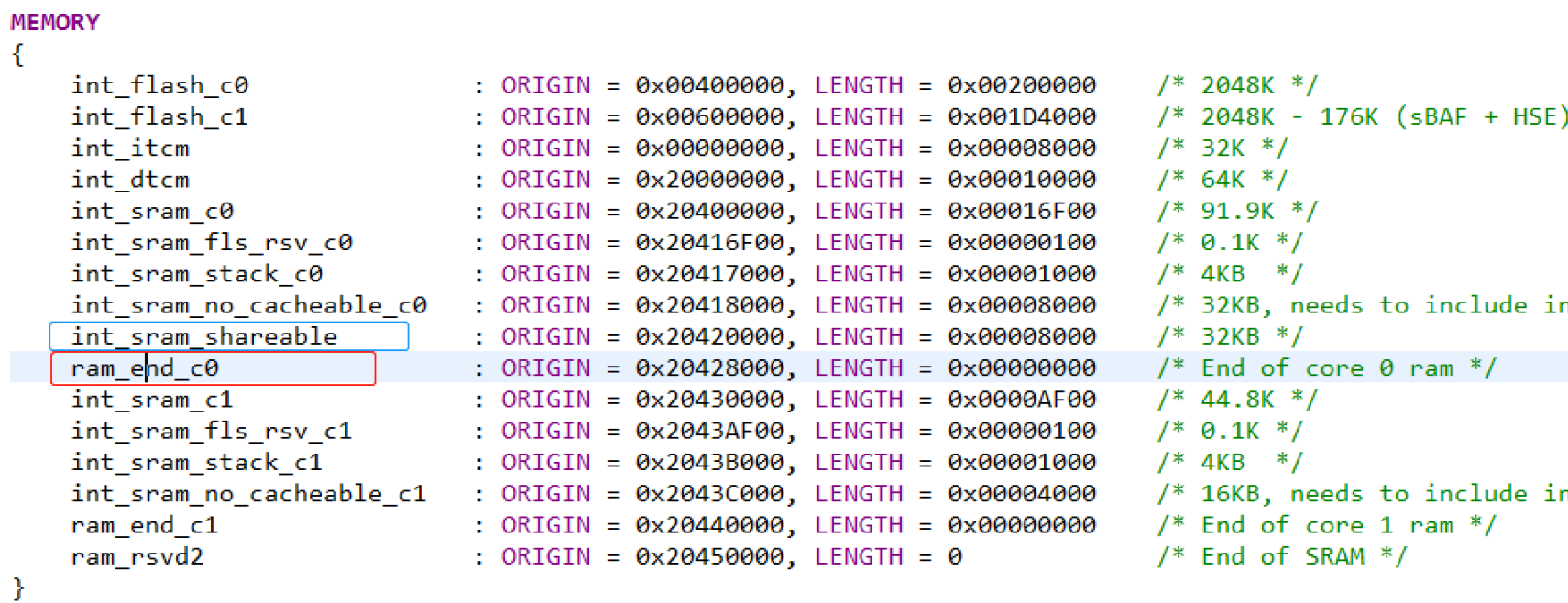
若存在，可能产生硬件错误。

解决办法可以通过EB/S32DS CT配置，将FLASH 操作的底层代码（Access code）搬运到RAM中运行。

* 1. **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默认的链接文件（或基于其修改）。

若工程中使用共享RAM，则需要修改默认的链接文件，把shareable放置在end\_c0之前，见下图：



RTD2.0.x不存在这个问题。

* 1. **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。

* 1. **参考应用笔记**

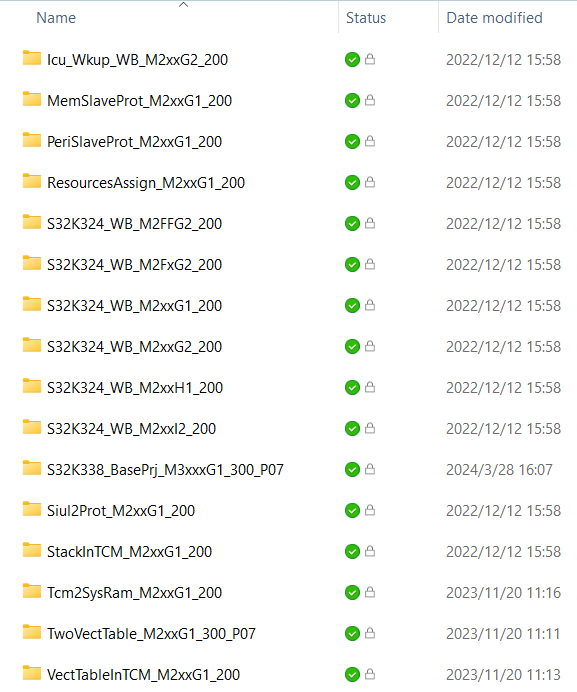
1、多核应用笔记《AN-Multi-Core\_Implementation\_in\_RTD\_of\_S32K3.pdf》。

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。

* 1. **示例工程**
     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：



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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

* + 1. **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。

1. **功能安全部分**
   1. **FCCU相关**
      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。

* 1. **BIST相关**
     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。

* 1. **故障注入相关**
     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。

* 1. **SAF相关**
     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（祥见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。

* + 1. **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。

* + 1. **非锁步核不需要进行RCCUTEST\_RCCU\_CM7测试**
    2. **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。

* + 1. **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。

* + 1. **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。

* + 1. **xRDC配置错误可能导致sCheck失败**

检查sCheck失败是否是否因为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。

* + 1. **Gmac模块进行sBoot前需完成初始化**
    2. **其它**

1. sCheck前，清除DCMGPRRWF1 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。
   1. **SAF开发license说明**

NXP提供SAF的开发License，一年内免费。开发License可获取新版本SAF全部源码。

开发License不提供质量保证，不提供或提供有限技术支持。

开发License是2025年8月开始提供的License，与之前提供的评估License不同。

开发License只用于开发评估，限制将其用于生产。

获取开发License需要用户先申请，然后NXP销售与客户沟通，填写表格，最后分享SAF代码。

* 1. **应用笔记**
     1. **S32K3xx\_Functional\_Safety\_Application\_Note**

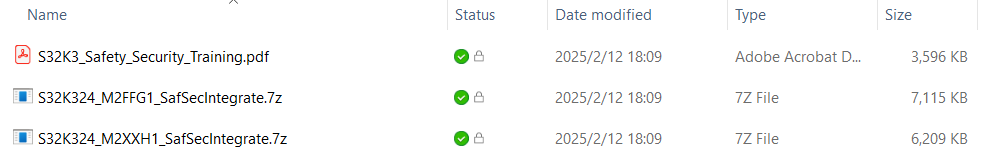
由NXP 应用工程师团队编写。

请联系代理商FAE获取。

* 1. **示例工程**
     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：



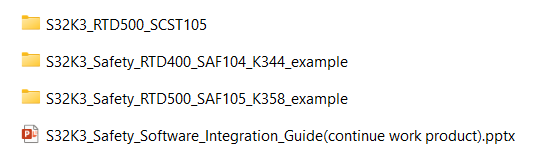
说明：由于SCST和SAF是收费软件，需要客户具备这些软件的license才能提供。

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。

* + 1. **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：



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 |