S32K系列MCU应用开发详解

Rev 1.0, Initial version completed, May 8th,2019.

Rev 1.1, Fixed some known typo, Updated July 8th,2019.

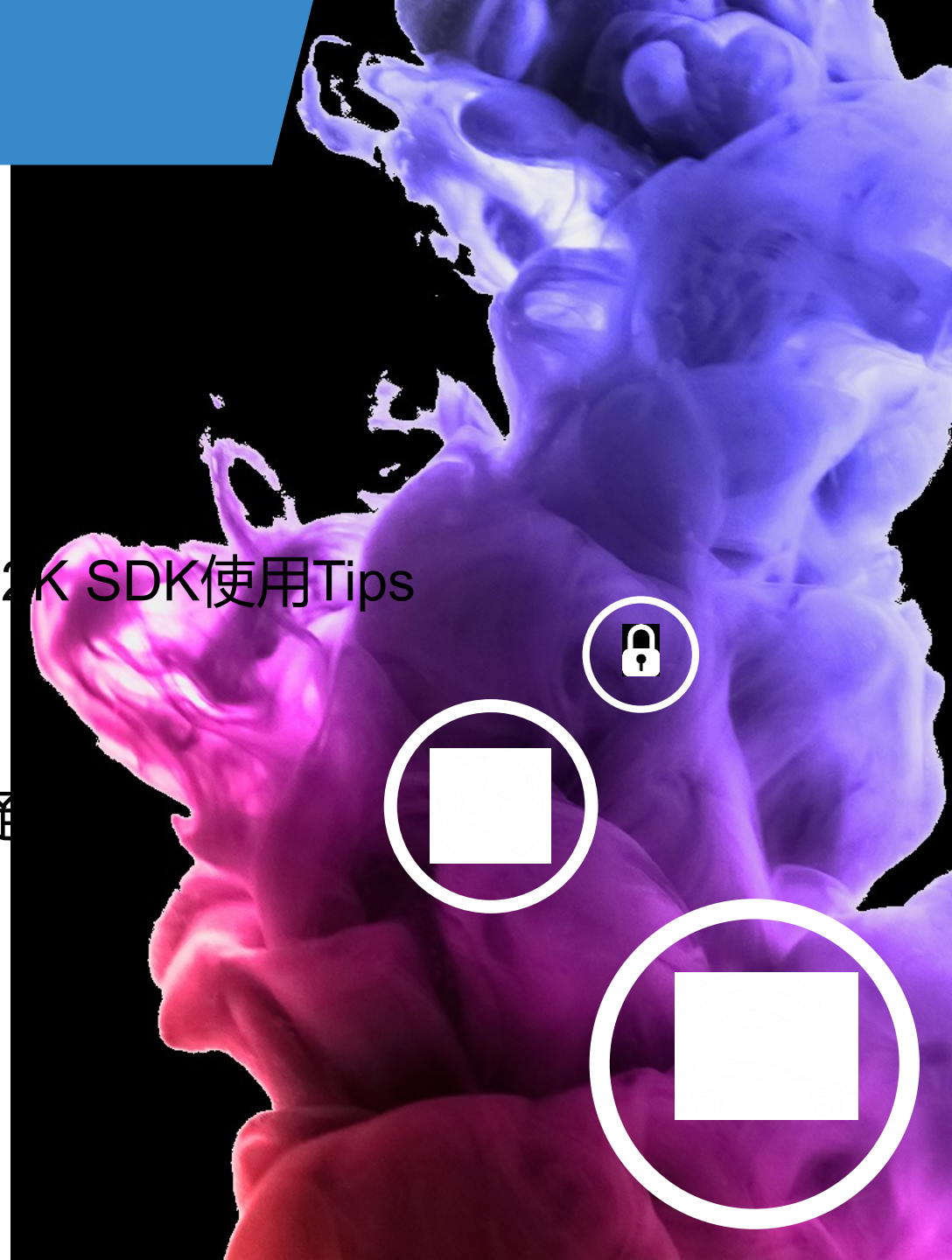
Rev 1.2, Added MTB and ITM/ETM trace features usage, Updated October 21th,2019.

Rev 1.3, add Chapter 5 S32K1xx App Guide in Chinese introduction, December,13th,2019.

Enwei Hu(胡恩伟)
NXP 汽车电子资深应用工程师
AMP-GPIS Team
2019年12月13日

内容提要

1. S32K系列MCU简介
2. S32K系列MCU内核和外设使用详解
3. S32K系列MCU硬件设计要点
4. S32K系列MCU软件开发之S32DS IDE和S32K SDK使用Tips
5. S32K系列MCU应用指南介绍
6. S32K系列MCU应用开发常见问题(FAQ)答疑
7. 总结与问答



NXP | SECURE CONNECTIONS FOR A SMARTER WORLD

1. NXP S32K系列MCU简介

- ✓*S32K系列MCU—加速汽车电子软件开发*
- ✓S32K14x 与 S32K11x 系列MCU功能模块概述
- ✓S32K系列MCU软件开发工具链选择

S32K系列MCU—加速汽车电子软件开发

高性能与高集成度

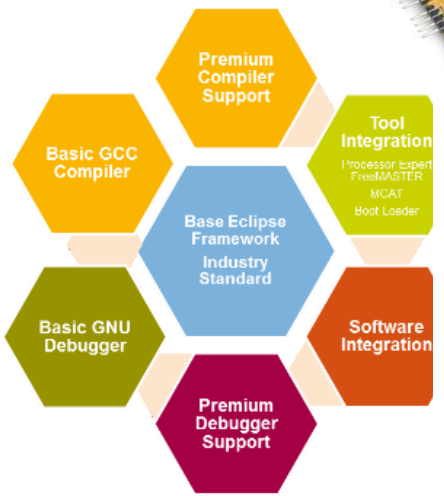
Future proof designs

- ARM Cortex M4F and M0+ cores
- ISO CAN-FD, CSEc hardware security, ISO26262 ASIL-B functional safety
- Ultra low power

汽车级软件开发套件(SDK)

Minimized complexity

- S32 Design Studio IDE
- Automotive-grade Software Development Kit (SDK)
- Autosar MCAL & OS, 3rd party ecosystem



S32 SDK

宽广的存储器和外设兼容特性

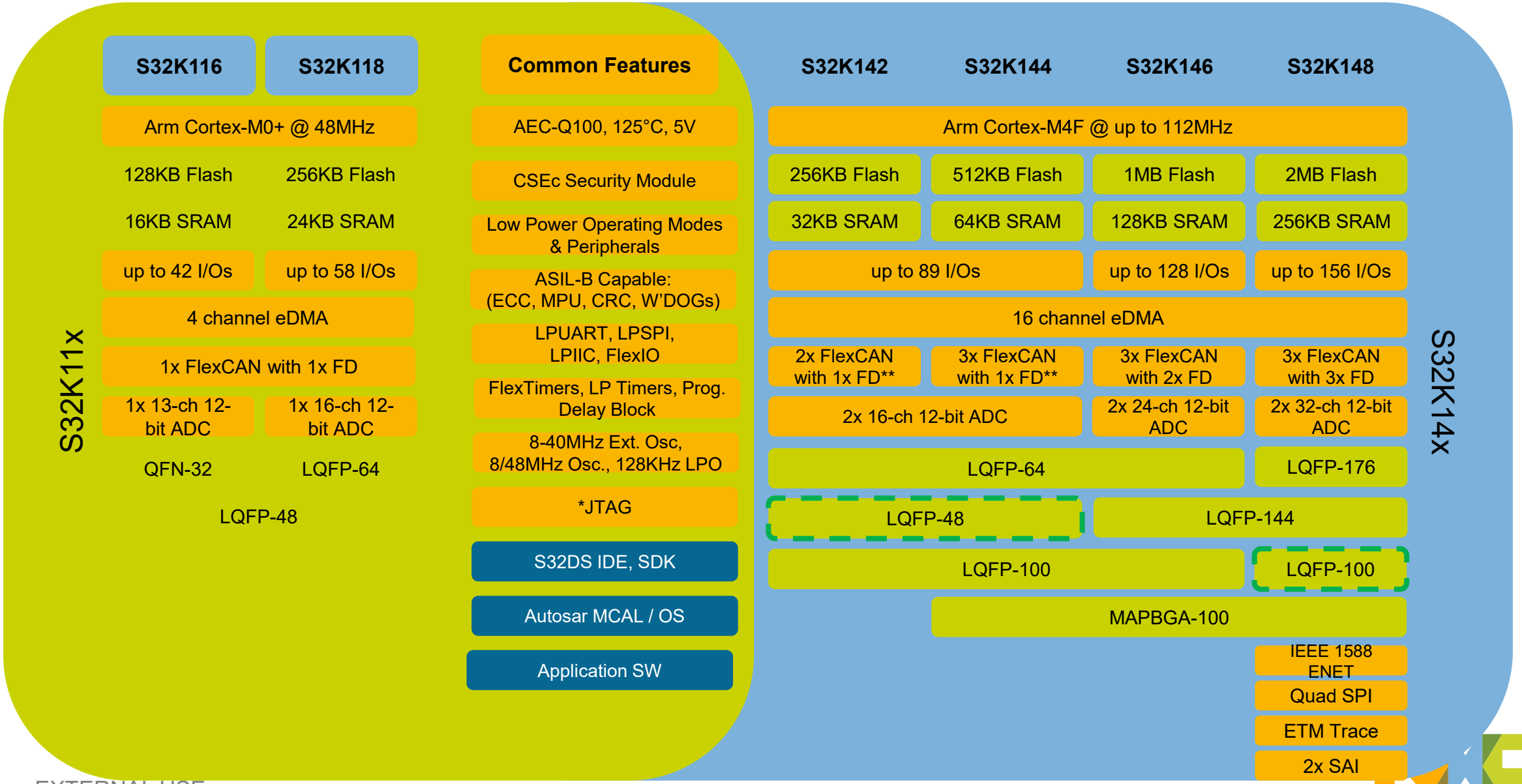
Maximised reuse

- **128KB** to **2MB**, **32** to **176 pins**
- H/w and S/w compatibility
- AEC Q100 grade 1 qualified (125°C), min. 15 year longevity

Product Longevity

S32K14x 与 S32K11x 系列MCU功能模块概述

Production | Development



*S32K14x only

** 2x CAN FD in S32K14xW (Grade 0)

S32K系列MCU软件开发工具链选择之集成开发环境(IDE)比较与选择

- 以下四种IDE可以支持S32K系列MCU的应用程序开发:

IDE	Compile/Optimization performance	Function Safety Certified	SDK support and integration	Debugger support	cost
S32DS for ARM v2018.R1	Based GNU toolchain, optimization is guaranteed by user	NO	Yes, and integrated the latest version SDK	P&E Multilink/OpenSDA IAR i-jet Segger J-Link Lauterbach iSYSTEM iC5700	free
IAR Embedded Workbench V8.30	Very good	yes	Yes, and can use with S32DS via Eclipse plug-in	P&E Multilink/OpenSDA IAR i-jet Segger J-Link Lauterbach iSYSTEM iC5700	medium
ARM MDK Keil 5	Good	yes	Yes, with pack but not the latest version	P&E Multilink/OpenSDA Segger J-Link ARM U-LINK/Pro Lauterbach iSYSTEM iC5700	medium
Green Hills MULTI IDE	Very good	yes	Yes, and can use with S32DS via Eclipse plug-in	Segger J-Link Lauterbach iSYSTEM iC5700	high

S32K1软件开发工具链选择之调试器(Debugger)比较与选择

- There are the following 6 debuggers can support S32K1xx serial MCU debug:

Features	IAR I-jet Trace	ARM ULINK/Pro	Segger J-Link/Pro	iSYSTEM iC5700	Lauterbach µTrace® for Cortex-M®	P&E Micro Multilink/Cyclone
Communication speed	fast	medium	medium	fast	fast	Max 20MB/s
Trace support	yes	yes	yes	yes	yes	no
breakpoints	No limit	No limit	No limit	No limit	No limit	Max 4 HW breakpoints
RTOS debug support	yes	yes	yes	yes	yes	yes
Dedicate debug SW	have	Have	have	have	Trace 32	No
Cost/price	medium	medium	medium	high	high	low

2. S32K系列MCU内核和外设使用详解

- ✓ARM Cortex-M0+ / M4: 内核寄存器、中断异常、存储器映射
- ✓S32K系列MCU的内部互联交换矩阵(Crossbar)、存储器(P-Flash /FlexNVM/EEEPROM和SRAM)
- ✓S32K系列MCU的复位启动过程、时钟源与时钟树、功耗模式
- ✓S32K系列MCU的外设—DMA、ADC、Timer、FlexCAN、ENET和FlexIO

S32K148 功能框图简介

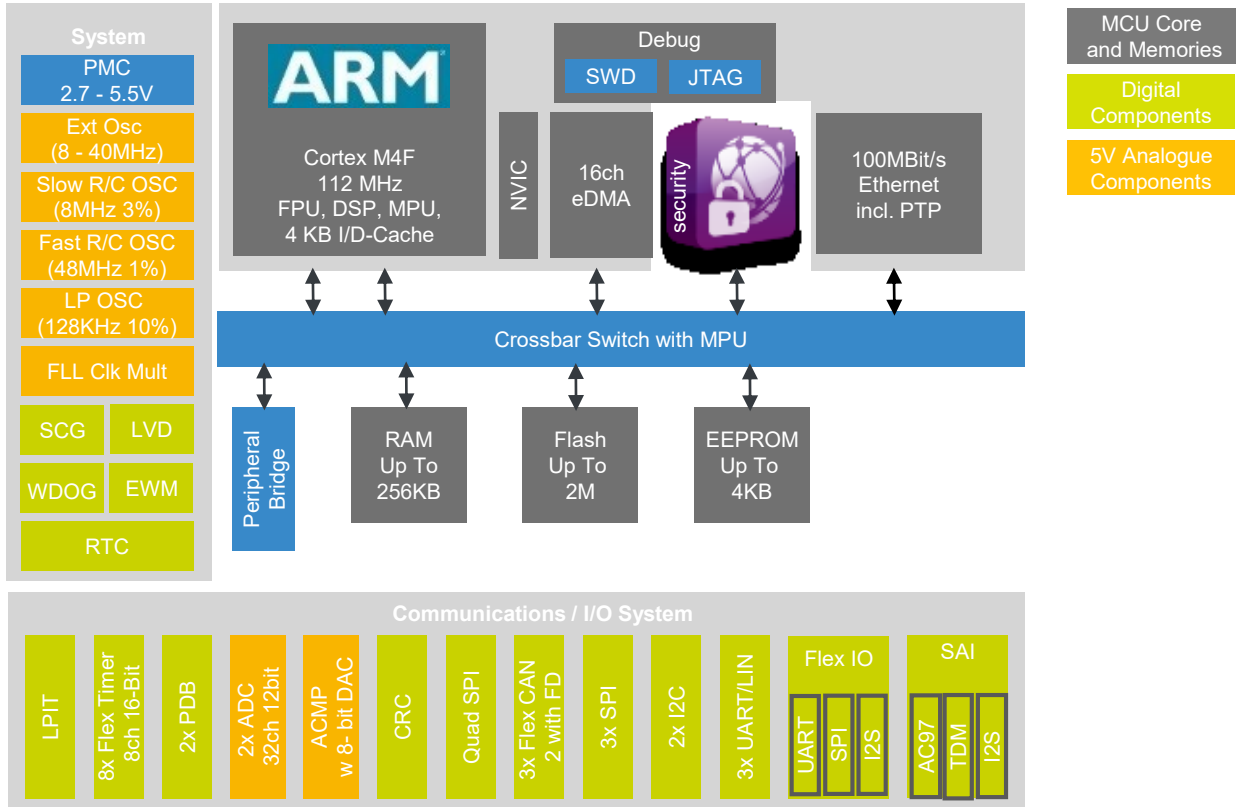
High performance
- ARM Cortex M4F up to 112MHz w FPU
- eDMA from 57xxx family

Software Friendly Architecture
- High RAM to Flash ratio
- Independent CPU and peripheral clocking
- 48MHz 1% IRC – no PLL init required in LP
- Registers maintained in all modes
- Programmable triggers for ADC no SW delay counters or extra interrupts

Functional safety
- ISO26262 support for ASIL B or higher
- Memory Protection Unit, ECC on P-Flash/FlexNVM and RAM
- Independent internal OSC for Watchdog
- Diversity between ADC and ACMP, SPI/SCI and FlexIO
- Core self test libraries
- Scalable LVD protection, CRC

Low power
- Low leakage technology
- Multiple VLP modes and IRC combos
- Wake-up on analog thresholds

Security
- CSEc (SHE-spec)



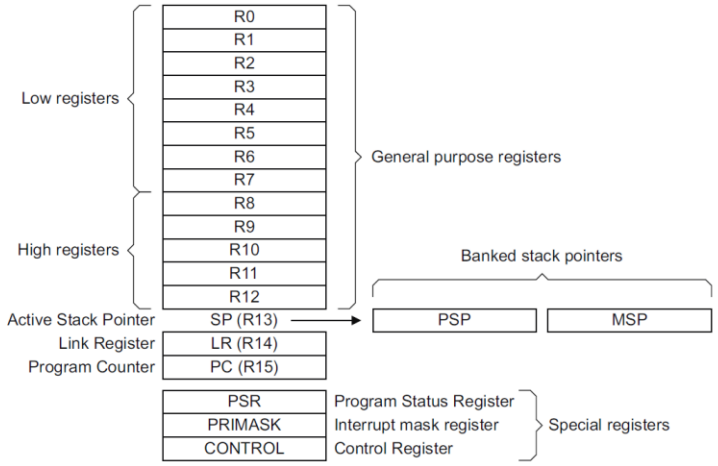
Operating Characteristics
- Voltage range: 2.7V to 5.5V
- Temperature (ambient): -40°C to +125°C

Packages & IO
- Open-drain for 3.3 V and hi-drive pins
- Powered ESD protection
- Packages: 100 BGA, 144 LQFP, 176 LQFP

ARM Cortex-M0+ / M4: 内核寄存器

S32K11x：ARM Cortex-M0+

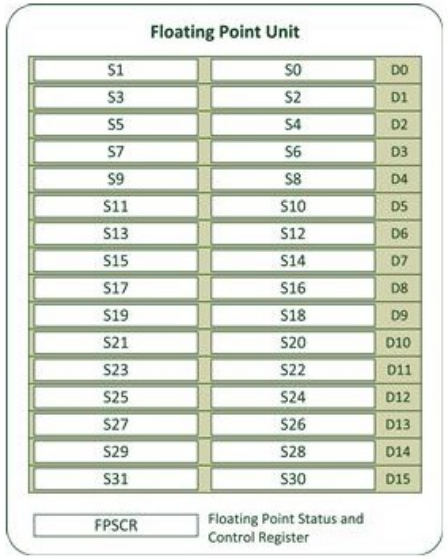
ARMv6-M，16-bit Thumb指令集

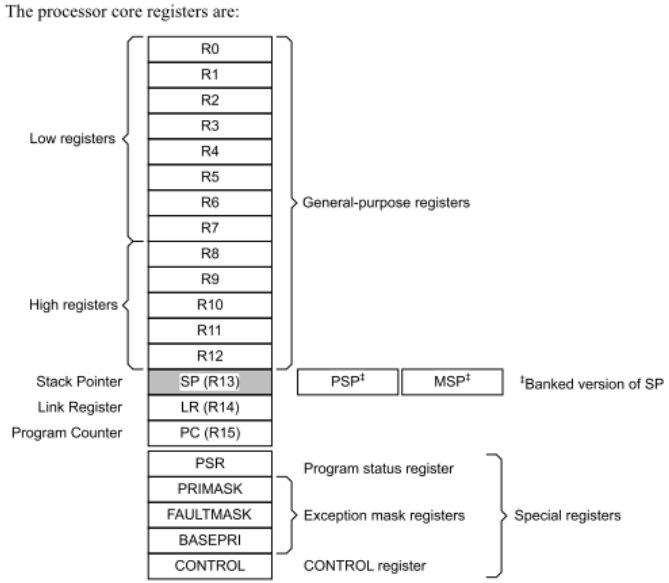


- CPU内核寄存器用于储存内核数据处理的中间结果，其访问速度是最快的，CPU所有的数据(SRAM/外设寄存器)操作都需要使用load指令先加载到CPU通用寄存器中，处理完成后，在使用store指令存回SRAM或者外设寄存器；

S32K14x：ARM Cortex-M4F

ARMv7-M， 16-bit + 32bitThumb2指令集





- CM4F内核有专门的32-bit FPU协处理器寄存器S0~S31及状态控制寄存器(FPSCR)；
- CM4F内核FPU有浮点数指令集；工作时不占用CPU通用寄存器和ALU资源，更高效；且中断发生时，可以通过lazy interrupt处理自动判断是否压栈这些FPU寄存器

ARM Cortex-M0+ / M4: 中断和异常

- ARM Cortex M系列CPU内核的所有系统异常和外设中断(IRQ中断)都由NVIC统一管理；
- 相较于CM4F，ARM Cortex M0+没有MemManage、BusFault和UsageFault异常；
- Reset(-3)、NMI(-2)和HardFault(-1)异常是默认使能的，且优先级固定，其余异常和IRQ中断的优先级可配置，数字越小优先级越高，默认优先级为0—可配置最高优先级；
- 默认复位后中断向量表基地址在**0x0000**地址，通过修改**NVIC->VTOR**寄存器可以将中断向量表重映射到其他地址(比如SRAM中)；
- ARM Cortex M系列CPU内核都集成了24-bit **SysTick Timer**定时器和**SCV**及**PendSV**异常，前者用于产生RTOS的内核(**Kernel**)心跳，后者用于实现RTOS的任务上下文(**context**)切换，从而方便不同ARM Cortex M系列CPU内核带RTOS的应用层代码移植；
- **一个外设IRQ中断要被CPU响应，需要配置**：
  - ①配置外设使能外设功能中断
  - ②NVIC中相应的IRQn使能；
  - ③配置该IRQn中断优先级(可选)
  - ④将该IRQn的中断ISR注册到中断向量表中；
- 在CM4F内核中，默认配置MemManage、BusFault和UsageFault异常关闭，相应的异常都会触发HardFault异常；通过**SCB**模块相关寄存器可以查找到具体的异常源；

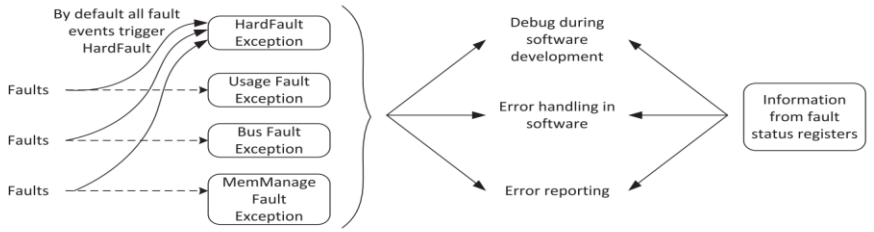


FIGURE 4.27
Fault exceptions usages

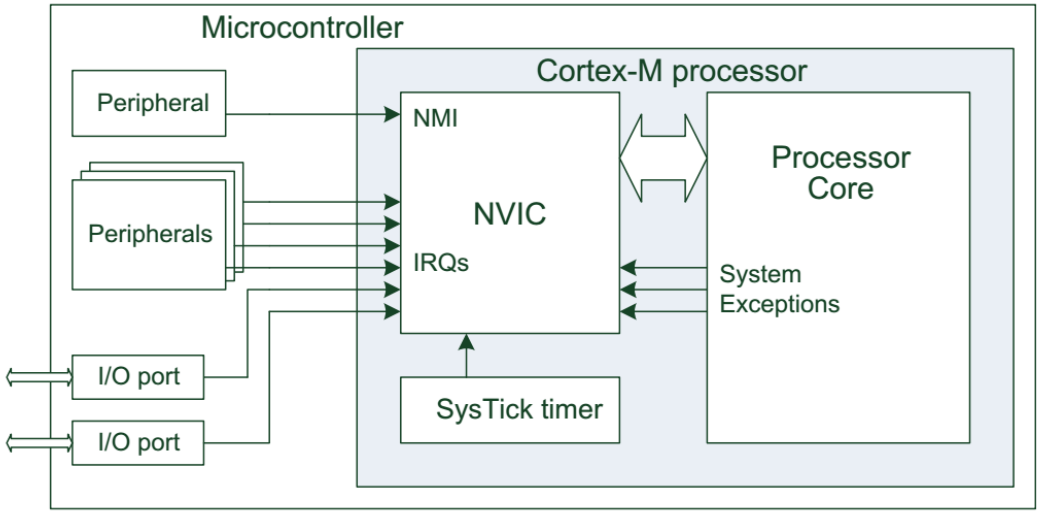


Table 4.9 Exception Types

Exception Number	CMSIS Interrupt Number	Exception Type	Priority	Function	Address Offset	Vectors
1	—	Reset	–3 (Highest)	Reset	0x48 – 0x3FF	IRQ #2 - #239
2	–14	NMI	–2	Non-Maskable interrupt	0x44	IRQ #1
3	–13	HardFault	–1	All classes of fault, when the corresponding fault handler cannot be activated because it is currently disabled or masked by exception masking	0x40	IRQ #0
4	–12	MemManage	Settable	Memory Management fault; caused by MPU violation or invalid accesses (such as an instruction fetch from a non-executable region)	0x3C	SysTick
5	–11	BusFault	Settable	Error response received from the bus system; caused by an instruction prefetch abort or data access error	0x38	PendSV
6	–10	Usage fault	Settable	Usage fault; typical causes are invalid instructions or invalid state transition attempts (such as trying to switch to ARM state in the Cortex-M3)	0x34	Reserved
7–10	—	—	—	Reserved	0x30	Debug Monitor
11	–5	SVC	Settable	Supervisor Call via SVC instruction	0x2C	SVC
12	–4	Debug monitor	Settable	Debug monitor – for software based debug (often not used)	0x28	Reserved
13	—	—	—	Reserved	0x24	Reserved
14	–2	PendSV	Settable	Pendable request for System Service	0x20	Reserved
15	–1	SYSTICK	Settable	System Tick Timer	0x1C	Reserved
16–255	0–239	IRQ	Settable	IRQ input #0–239	0x18	Usage fault
					0x14	Bus Fault
					0x10	MemManage Fault
					0x0C	HardFault
					0x08	NMI
					0x04	Reset
					0x00	Initial value of MPS

ARM Cortex M4F内核中断和异常类型　　**CM4F内核中断向量表**

ARM Cortex-M0+ / M4: 中断和异常

- ARM Cortex M0+内核支持最多**64**个中断管理；
- ARM Cortex M4F内核支持最多**256**个中断管理；
- 在ARM Cortex M0+内核中未实现MemManage(#4)、BusFault(#5)、UsageFault(#6)和DebugMonitor(#12)异常

Table B1-3 Exception numbers

Exception number	Exception
1	Reset
2	NMI
3	HardFault
4-10	Reserved
11	SVCall
12-13	Reserved
14	PendSV
15	SysTick, optional
16	External Interrupt(0)
…	…
16 + N	External Interrupt(N)

ARM Cortex M0+内核中断和异常类型

Table B1-4 Exception numbers

Exception number	Exception
1	Reset
2	NMI
3	HardFault
4	MemManage
5	BusFault
6	UsageFault
7-10	Reserved
11	SVCall
12	DebugMonitor
13	Reserved
14	PendSV
15	SysTick
16	External interrupt 0
.	.
.	.
.	.
16+N	External interrupt N

ARM Cortex M4F内核中断和异常类型

ARM Cortex M4 存储器映射

- 由ARM公司预定义的Cortex M系列CPU内核存储器地址映射；
- P-Flash、D-Flash/ FlexNVM和FlexRAM/EEE映射到代码区，SRAM和外设映射到SRAM和外设区，这样程序运行时，可实现真正的**哈弗架构**；
- 统一的PPB和复位中断向量表(0地址)映射方便应用程序代码在不同的ARM Cortex M系列内核之间移植；

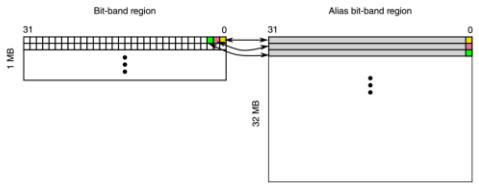
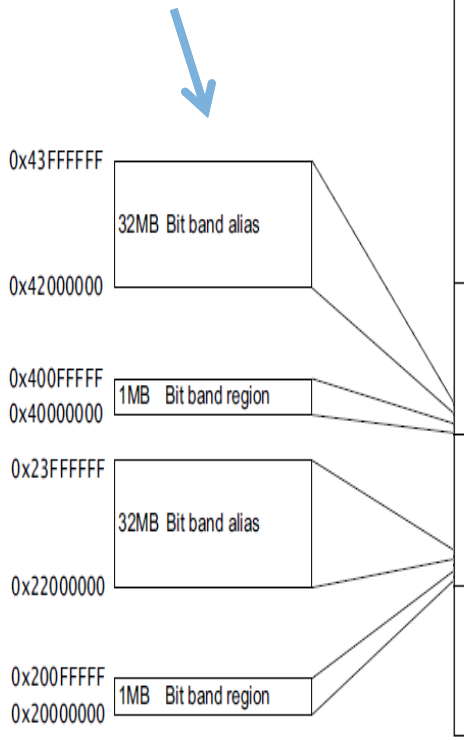
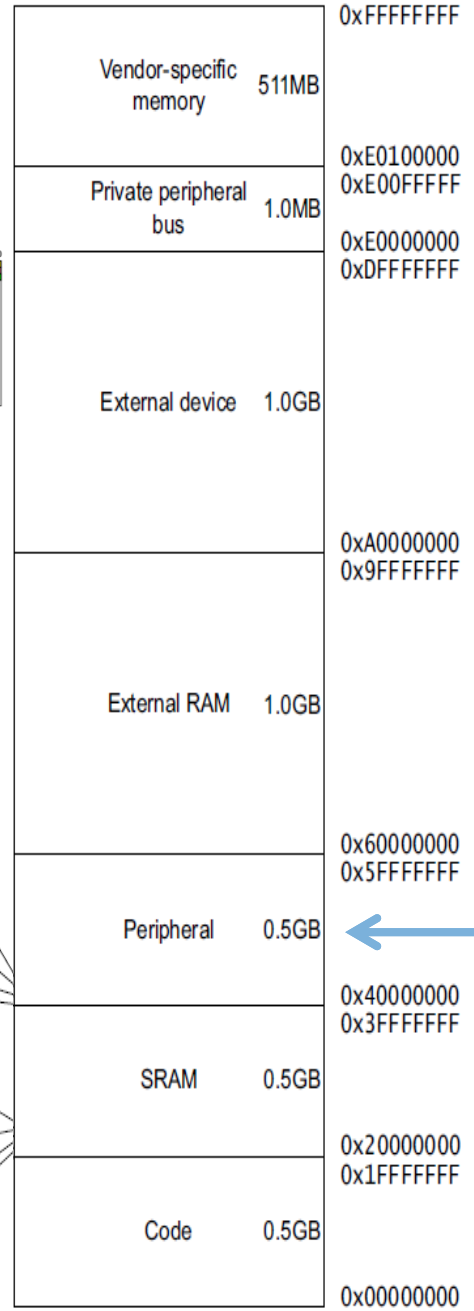


Figure 3-2. Alias bit-band mapping

Bit-Band 区域
可用于原子位操作





通过私有外设总线(PPB Bus)访问，内核外设同样拥有同一的存储器映射(Memory Map)，从而方便访问

通过系统总线(System Bus)访问

此区域为S32K系列MCU片上外设寄存器地址映射

通过代码总线(CODE Bus)访问

S32K系列MCU的内部互联总线交换矩阵(Crossbar Switch)

- S32K系列MCU的Crossbar Switch(**AXBS-Lite**)是改进的ARM AMBA总线矩阵, 用于互联CPU内核与系统存储器及外设模块的访问(读写);
- AXBS-Lite的Master可以主动发起数据访问请求，而Slave则只能被动接受访问;
  - 每个Crossbar Switch的端口都有自己的控制总线、地址总线和数据总线;
  - Master的工作频率高于Salve，eg. CM4F core @max 112MHz vs. Flash @max 28MHz in HSRUN模式;
  - 所有的片上外设模块(比如 GPIO、ADC、定时器和通信外设)都是统一挂在外设桥上的(S2—**AIPS-Lite**)
  - 两个Crossbar的master可以同时访问不同的salve，从而保证MCU系统工作带宽;
  - Master对salve的访问受**MPU**的保护;
- **S32K11x**
  - 2个master和3个salve
  - 单周期GPIO不经过Crossbar Switch
- **S32K14x**
  - 3个master和3个salve(S32K142/4/6)
  - 4个master和4个salve(S32K148)

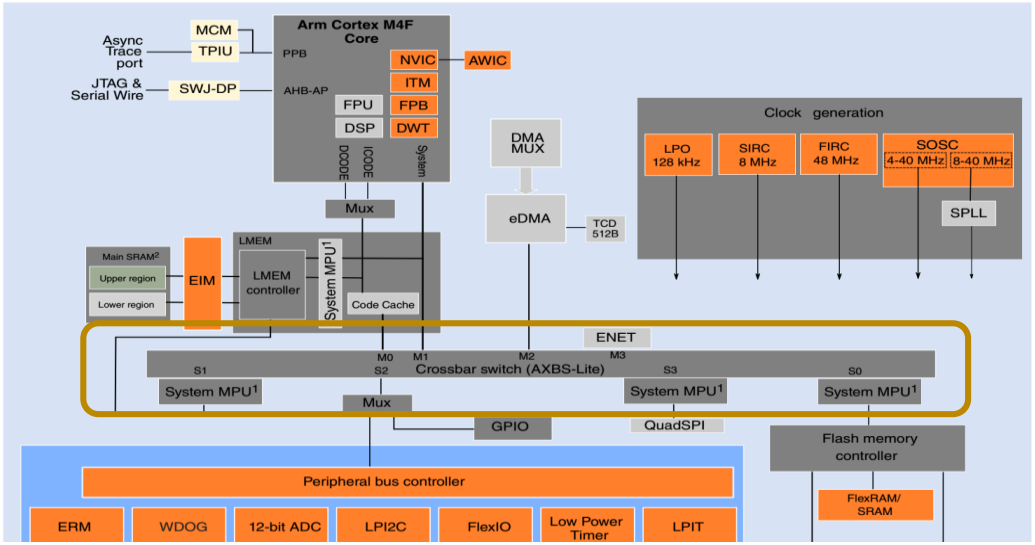


Table 14-1. Master port assignments and priority order for fixed-priority arbitration

Chips	Master port number 0	Master port number 1	Master port number 2	Master port number 3
	Priority 3 [1]	Priority 2[1]	Priority 1[1]	Priority 4[1]
S32K148	Arm core code bus	Arm core system bus	DMA	ENET
S32K146	Arm core code bus	Arm core system bus	DMA	-
S32K144	Arm core code bus	Arm core system bus	DMA	
S32K142	Arm core code bus	Arm core system bus	DMA	
S32K116	Arm core master bus	-	DMA	
S32K118	Arm core master bus	-	DMA	

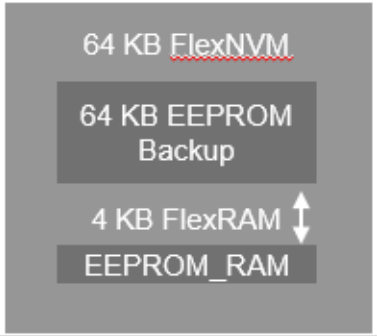
Table 14-2. Slave port assignments and system MPU protection

Chips	Slave port number 0	Slave port number 1	Slave port number 2	Slave port number 3
S32K148	Flash memory controller[1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO [2]	QuadSPI [1]
S32K146	Flash memory controller[1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO [2]	-
S32K144	Flash memory controller [1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO [2]	
S32K142	Flash memory controller [1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO [2]	
S32K116	Flash memory controller [1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO [2]	
S32K118	Flash memory controller [1]	SRAM controllers [1]	Peripheral Bridge 0/ GPIO[2]	

ion. For fixed priority, set setting applies to all slave ports.

S32K FlexNVM使用--Data Flash vs. 模拟EEPROM

- S32K系列MCU的FlexNVM 可以被分区用作Data Flash 或者 EEPROM备份(backup)Flash
- Data Flash 用于保存更新频率不高的数据.
  - e.g. 数据查找表或者配置信息
  - 最新2 KB 独立可擦除扇区(Sector)
- 模拟EEPROM则用于保存频繁更新的数据记录
  - e.g. 行车数据记录，里程数.
  - 2/4KB FlexRAM(用作EEPROM_RAM)将被硬件状态机自动备份到FlexNVM中
  - 用户像访问普通RAM一样即可访问EEPROM，无需传统EEPROM那样复杂的驱动代码
- 模拟EEPROM 使用Tips:
  - 每次写EEPROM时，必须检查FCNFG[EEERDY] 确认上一次写入数据已经被写入EEPROM backup FlexNVM中，否则将导致CPU内核HardFault异常；
  - 模拟EEPROM使能通过调用Flash分区命令(partition command)实现，分区信息保存在D-Flash IFR中；重新分区，将丢失之前的EEPROM数据；
  - 更多细节，请参考应用笔记-- AN11983, Using the S32K1xx EEPROM Functionality (REV 1)

EEPROM Only	Data Flash Only(Default)
	
• No Data Flash storage.	• 64 KB Data Flash with 2 KB individually erasable sectors and 1K P/E cycles each.
• EEPROM with 64 KB FlexNVM backup. - Roughly 100K record updates for a full 4 KB image, or 800K updates for a 512 B image, etc.	• No EEPROM. - 4 KB SRAM available for other uses.

S32K系列MCU的复位和启动(Reset & Boot)过程

Table 23-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low voltage detect (LVD) • Watchdog reset • Loss-of-clock (LOC) reset • Loss-of-lock (LOL) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • MDM DAP system reset
Debug reset	• JTAG reset

This MCU supports cold booting from internal flash memory.

When the MCU boots from internal flash memory, the reset vectors are located at address 0x0 (initial SP_main) and 0x4 (initial PC).

The MCU also supports boot from RAM by relocating the exception vector table to RAM. This is implemented through a programmable Vector Table Offset Register (VTOR) in NVIC module.

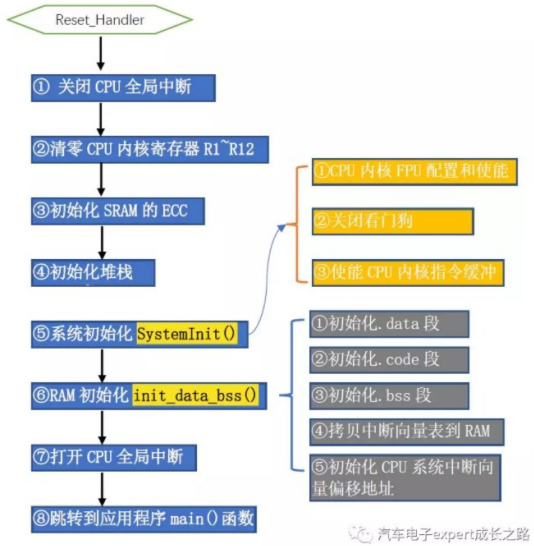
The MCU supports serial code download via
- CAN
- LIN
- SPI etc.

- CPU从复位向量(Reset_Handler)到运行至main()的startup过程，请参考如下公众号文章(点击文章标题即可跳转阅读)：
  - *《浅谈嵌入式MCU软件开发之S32K系列MCU启动过程及重映射代码到RAM中运行方法详解》；*

- 硬件复位和启动(boot)序列:

1. A system reset is held on internal logic, the RESET_B pin is driven out low, and the SCG is enabled in its default clocking mode.
2. Required clocks are enabled (core clock, system clock, flash clock, and any bus clocks that do not have clock gate control reset to disabled).
3. The system reset on internal logic continues to be held, but the flash controller is released from reset and begins initialization operation while the reset control logic continues to drive the RESET_B pin low.
4. Early in reset sequencing, the NVM option byte is read and stored to the flash memory module's FOPT register. If the LPBOOT is programmed for an alternate clock divider reset value, the system/core clock is switched to a slower clock speed.
5. When flash memory initialization completes, the RESET_B pin is released. If RESET_B continues to be asserted (an indication of a slow rise time on the RESET_B pin or external drive in low), the system continues to be held in reset. Once the RESET_B pin is detected high, the core clock is enabled and the system is released from reset.
6. When the system exits reset, the processor sets up the stack, program counter (PC), and link register (LR). The processor reads the start SP (SP_main) from vector-table offset 0. The core reads the start PC from vector-table offset 4. LR is written to 0xFFFF_FFFF.
7. If FlexNVM is enabled, the flash memory controller continues to restore the FlexNVM data. This data is not available immediately out of reset and the system should not access this data until the flash controller completes this initialization step as indicated by the EEERDY flag.

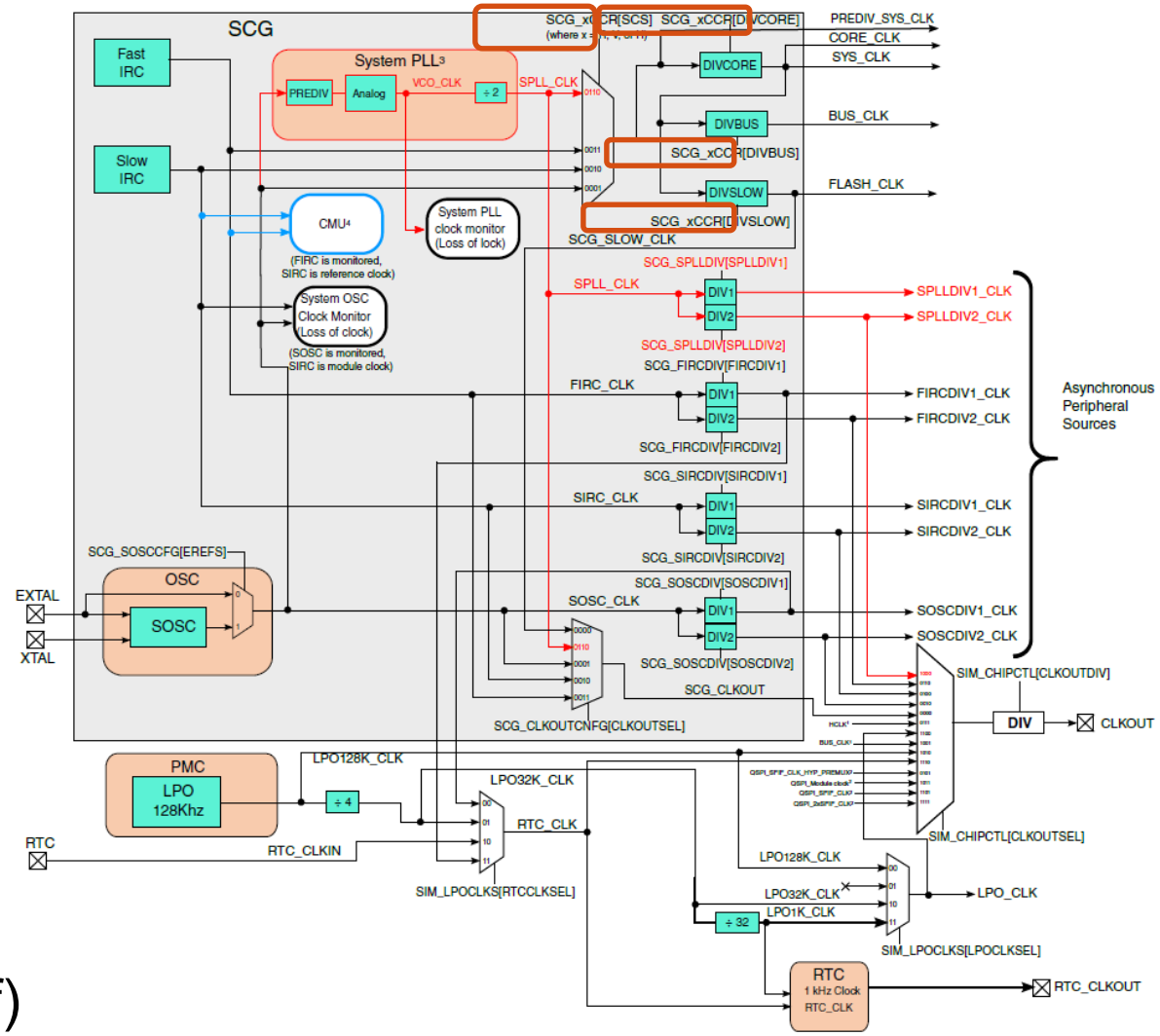
- 软件启动(Startup)流程:



S32K14x系列MCU的时钟树介绍

- S32K14x系列MCU有多种参考时钟可以选择
  - 48MHz FIRC(复位后的默认时钟源)
  - 8MHz SIRC
  - 4~40MHz XOSC(支持有源时钟和无源晶振)
  - SPLL(倍频时钟)
  - 128KHz LPO(低功耗时钟,可做RTC和Watchdog参考时钟)

- 各时钟源有多个分频器(DIV)输出，从而可以产生多种时钟频率供外设使用；
- HSRUN、RUN和VLPR模式有各自独立的内核时钟(CORE_CLK)、总线时钟(BUS_CLK)和Flash工作时钟(FLASH_CLK)配置寄存器；从而可以实现不同工作模式间的快速无缝切换；
- 仅外部XOSC可作为SPLL倍频参考时钟；

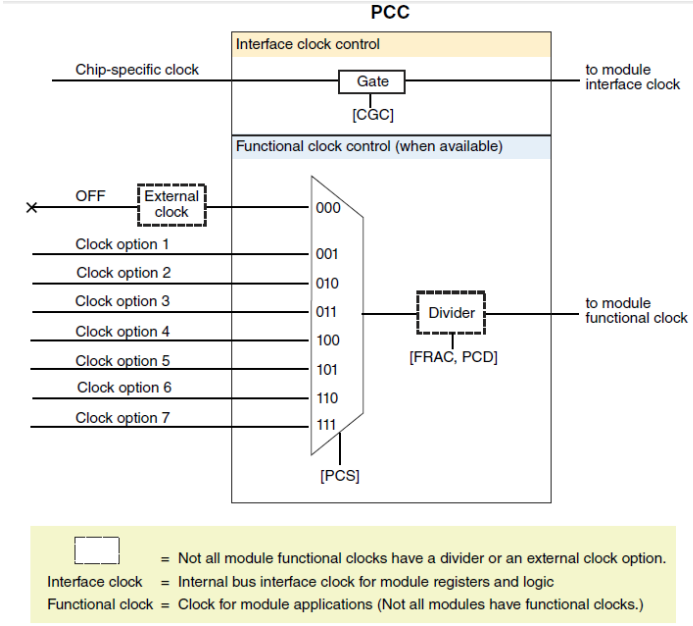


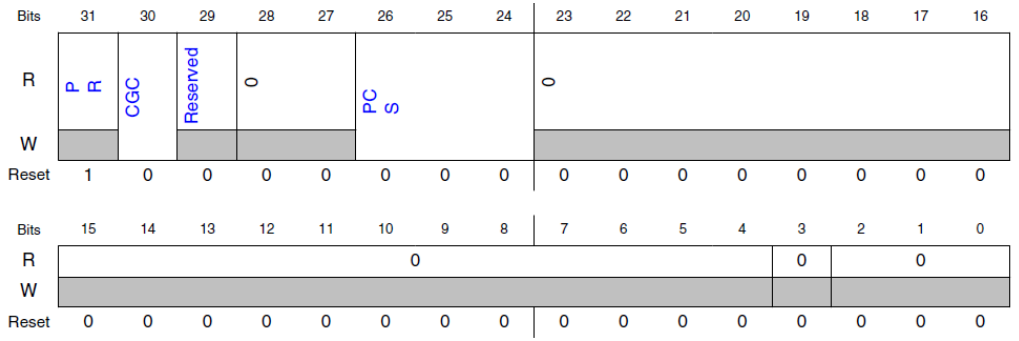
For more details:

- Reference Manual(S32K-RM_Rev9.pdf)
- AN5408, S32K14x Clock Calculator Guide - Application notes (REV1)

外设门控时钟(clock gate)和功能时钟(Function Clock)选择

- S32K系列MCU的所有外设模块都有各自的门控时钟，由PCC模块统一管理可以单独关闭或开启以优化系统功耗；
  - MCU复位后，默认外设时钟是关闭的(disable)，所以在配置某一外设模块(读写访问其控制状态寄存器)之前，请务必确认其门控时钟已经使能，否则将产生bus error，进入CPU内核的HardFault异常；
- 每个外设模块的功能时钟(Function Clock)有多种选择，从而可以灵活配置；
  - 不同的外设可以选择不同的功能时钟，从而可以实现功耗模式的快速无缝切换和产生不同的外设工作频率从而进一步降低系统功耗；
  - Max 115.2Kbit/s UART .VS max 20Mbit/s SPI





31 PR	Present This bit shows whether the peripheral is present on this device. 0b - Peripheral is not present. 1b - Peripheral is present.
30 CGC	Clock Gate Control This read/write bit enables the clock for the peripheral. 0b - Clock disabled 1b - Clock enabled. The current clock selection and divider options are locked.
26-24 PCS	Peripheral Clock Source Select This read/write bit field is used for peripherals that support various clock selections. This field can be written only when the clock is disabled (CGC = 0). 000b - Clock is off. 001b - Clock option 1 010b - Clock option 2 011b - Clock option 3 100b - Clock option 4 101b - Clock option 5 110b - Clock option 6 111b - Clock option 7

S32K14x系列MCU的功耗模式和时钟要求

- **S32K14x系列MCU的功耗模式切换状态机如右图，S32K11x系列没有HSRUN模式；**
- **HSRUN模式**
  - **高速运行模式**，通过SPLL将XOSC倍频输出作为系统参考时钟，可以CPU运行到112MHz，从而提高CPU计算性能；
  - HSRUN模式下，不支持CSEc和EEPROM擦除和写操作；
- **RUN 模式**
  - **正常运行模式**，其时钟源可以来自SPLL、SIRC或者FIRC，最大工作频率为80MHz；
  - MCU复位后的默认工作模式，使用48MHz FIRC作为系统参考时钟
- **VLPR** 模式
  - **低功耗运行模式**，此模式下SPLL和FIRC及外部XOSC必须关闭，所以只能使用SIRC作为系统参考时钟；
  - 此模式下，也不支持CSEc和EEPROM擦除和写操作；
- **VLPS模式**
  - **低功耗停止模式**，S32K系列MCU的最低功耗模式；
  - 此模式可以选择是否关闭SIRC，若要低功耗外设—比如LPUART、LPSPI继续工作，则需要使能SIRC并选择SIRC做其参考时钟；
- **STOP1模式**
  - **停止模式1**， FIRC, SIRC 和XOSC 时钟可用
  - 内核时钟、Flash时钟及系统和总线时钟关闭
- **STOP2模式**
  - **停止模式1**， FIRC, SIRC 和XOSC 时钟可用
  - 内核时钟和系统时钟关闭
- **功耗大小**
  - **HSRUN > RUN > STOP2 > STOP1 > VLPR > VLPS(~uA)**

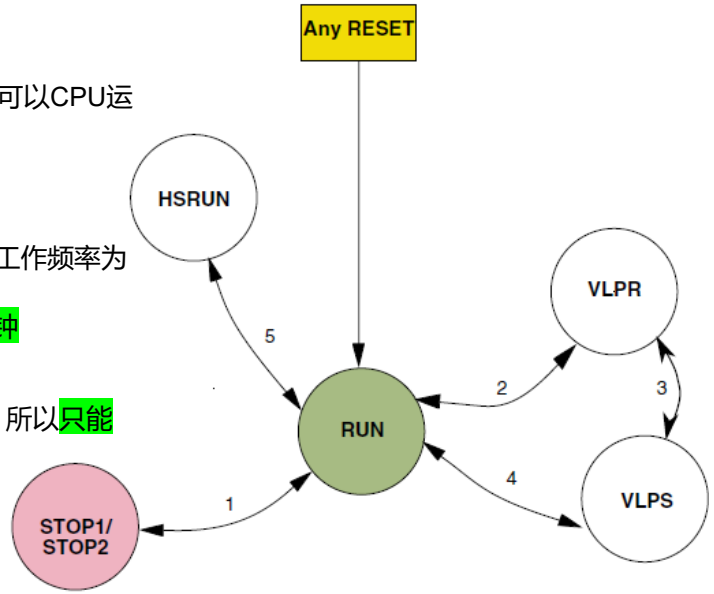


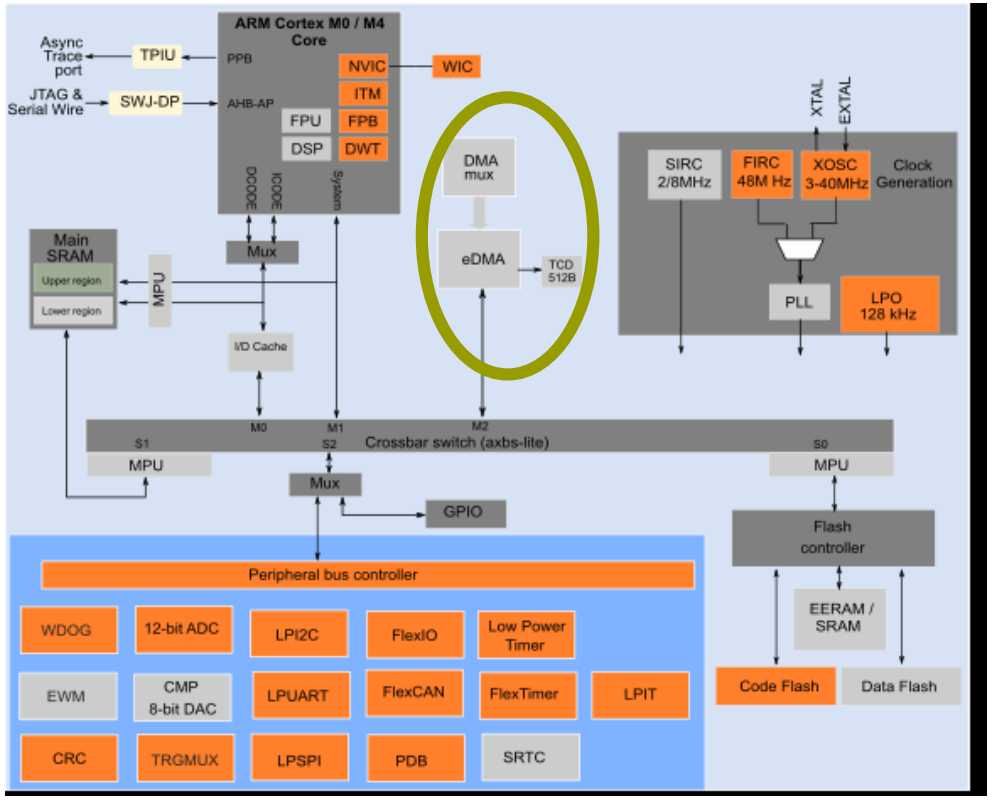
Table 38-4. Module operation in available power modes (continued)

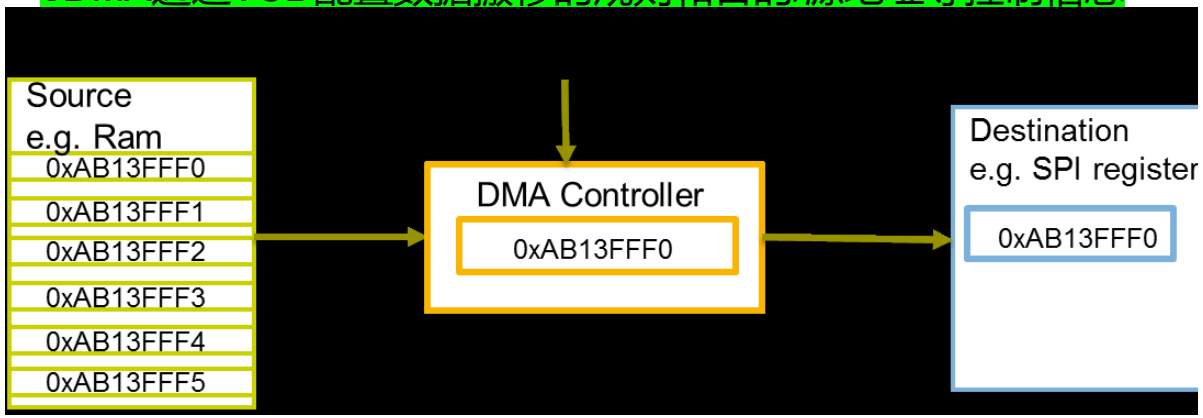
Modules	VLPR[1]	STOP(STOP1/ STOP2)	VLPS[1]	RUN	HSRUN
		needs to be set) STOP1, FF in STOP2			
EWM	FF (LPO Clock as source)	Static in STOP1, FF in STOP2	OFF	FF	FF
Clocks					
128 kHz LPO	FF	FF	FF	FF	FF
PLL	OFF [2]	FF	OFF [2]	FF	FF
SIRC	FF	FF	FF	FF	FF
FIRC	OFF [2]	FF	OFF [2]	FF	FF
System oscillator (OSC)	OFF [2]	FF	OFF [2]	FF	FF
SCG	Only SIRC is available	All clock sources are available	Only SIRC is available	All clock sources are available	All clock sources are available
Core clock	4 MHz max (SIRC as a source)	OFF	OFF	80 MHz max (PLL as a source)/48 MHz max (FIRC as a source)	112 MHz max (PLL as a source)
Flash clock	1 MHz max (SIRC as a source)	OFF in STOP1, Available in STOP2	OFF	26.67 MHz max (PLL as a source)/24 MHz max (FIRC as a source)	28 MHz max (PLL as a source)
System clock	4 MHz max (SIRC as a source)	OFF	OFF	80 MHz max (PLL as a source)/48 MHz max (FIRC as a source)	112 MHz max (PLL as a source)
Bus clock	4 MHz max (SIRC as a source)	OFF in STOP1, Available in STOP2	OFF	48 MHz max (FIRC as a source)/40 MHz max (PLL as a source)	56 MHz max (PLL as a source)

CSEc (Security) or EEPROM writes/erase will trigger error flags in HSRUN mode (112 MHz) because this use case is not allowed to execute simultaneously. The device need to switch to RUN mode (80 MHz) to execute CSEc (Security) or EEPROM writes/erase.

DMAMUX & eDMA模块

- 作为Crossbar的Master(**M2**)，提供无CPU参与的数据搬移功能，减少CPU中断loading，提高MCU工作效率；
- 支持软件触发或者硬件外设(中断) 或者通道链接(linking)触发DMA请求
  - 支持8,16, 32-bit 位宽数据传输
  - 4(S32K11x)/16(S32K14x)DMA通道
  - 支持主次(minor & major)循环
  - 支持环形缓存(circular buffers)
- 每个DMA通道完成主循环数据搬移后可以配置使能独立的中断；
- 灵活的通道优先级机制：可编程固定优先级或者轮询
- DMA 作为**master**连接到crossbar switch.
- CPU and DMA 可以同时访问不同的crossbar slave，e.g. SRAM_U vs. SRAM_L
- **DMAMUX配置DMA通道的触发源**
- eDMA通过TCD配置数据搬移的规则和目的/源地址等控制信息





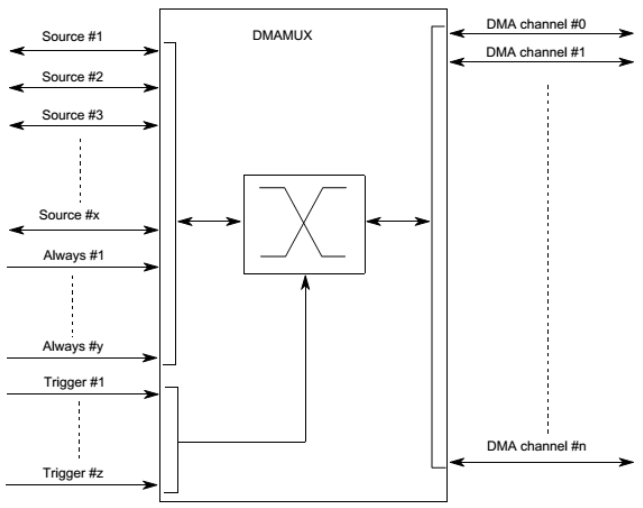


Figure 15-1. DMAMUX block diagram

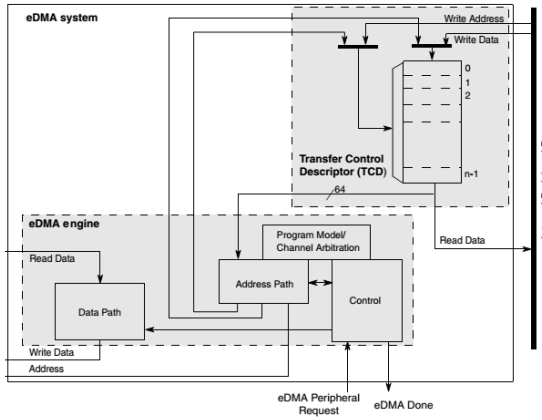
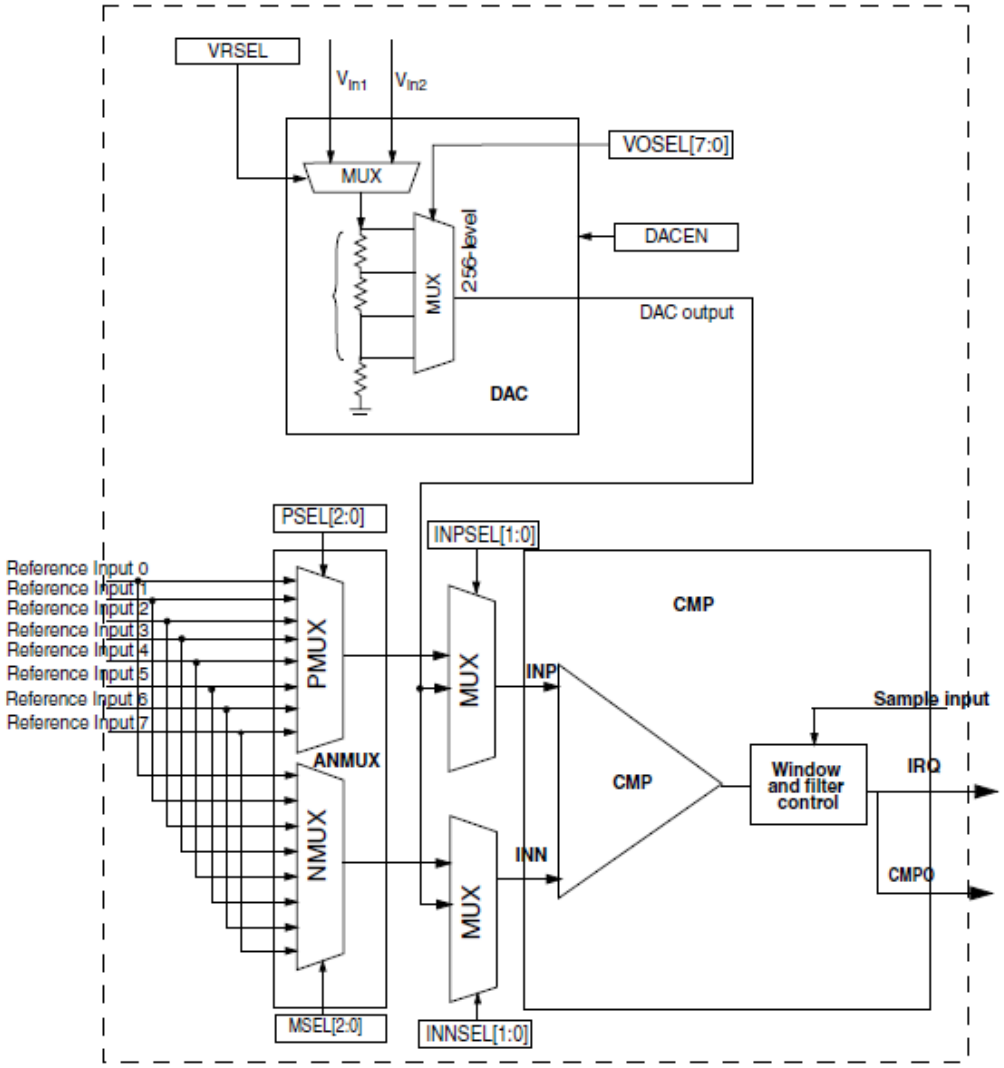


Figure 16-1. eDMA system block diagram

模拟比较器(ACMP)模块

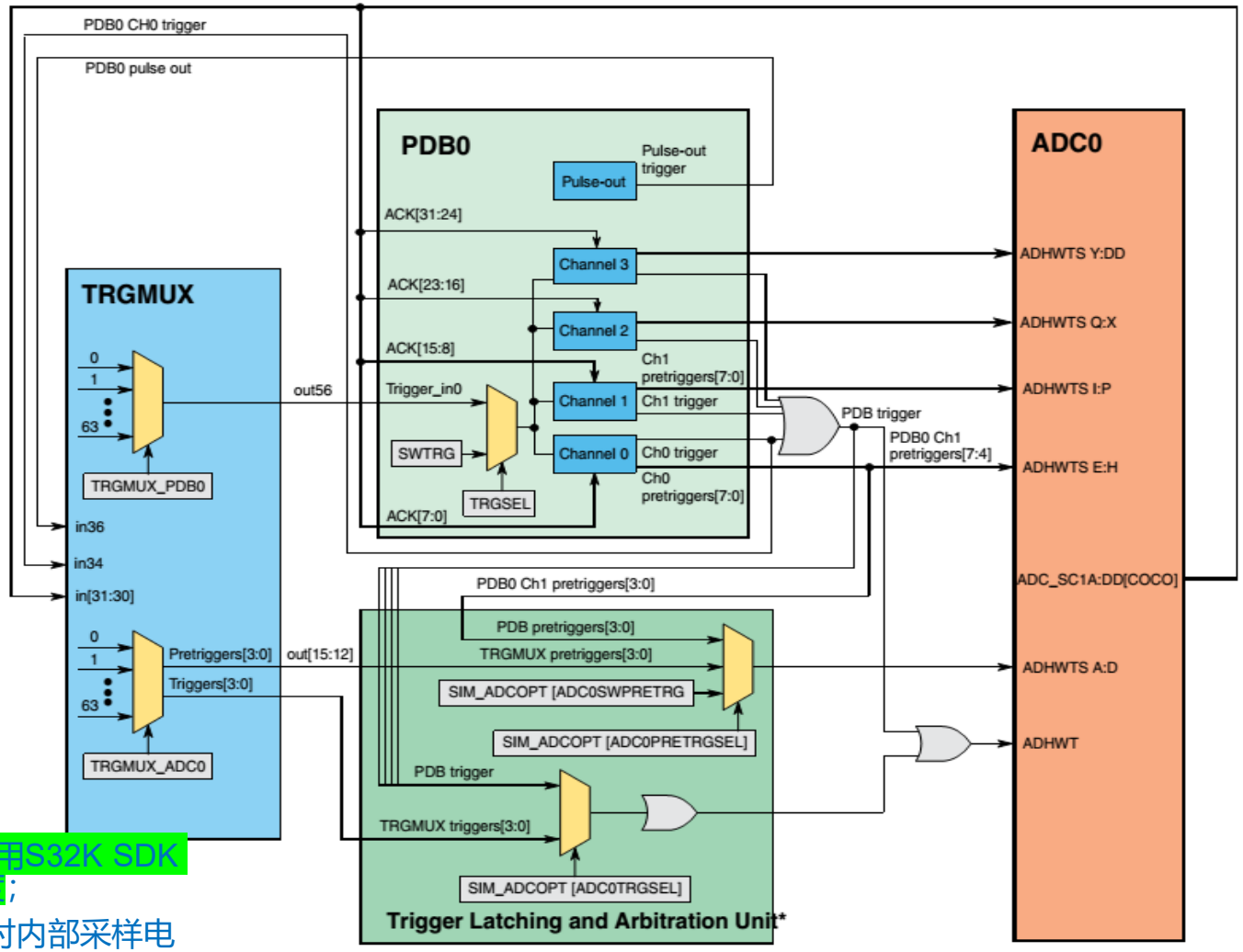
- 支持Rail to Rail 0V-5V模拟输入
- 8 个ACMP 通道
- 集成8-bit DAC产生比较参考电平
- 支持滤波模式
- 支持通过PDB & LPIT 脉冲实现窗口模式
- 支持LPTMR定时器触发比较
- 比较结果输出至FTM, TRGMUX 或者MCU引脚
- VLPS模式下还可以全功能工作，从而可作为低功耗唤醒源



S32K ADC & PDB模块

- 单端12-bit逐次逼近寄存器型(**SAR**) ADC，支持配置为10-bit或者8-bit分辨率以提高转换速度;
- 右对齐无符号(right-justified unsigned)结果输出;
- 支持单次或者连续转换模式;
- S32K14x双ADC模块支持交错(interleave)采样通道;
- 可配置采样时间和转换速度(sample time and conversion speed)
- 支持转换完成/硬件平均完成标志中断
- 可选软件触发转换/硬件触发AD转换(触发源和通道有相应的PDB模块和TRGMUX配置)
- 支持可编程自动比较(大于、小于、等于，范围内或超出范围)功能;
- 支持硬件2/4/8/16/32次采样平均输出功能;
- 外部可选参考电压VREFH/VREFL;
- 自校准功能，保证ADC采样精度;

- 使用Tips:
  - 在完成ADC模块初始化后开始ADC采样转换前，必须做自校准(调用S32K SDK ADC组件的相应API函数即可)，才能获得TUE=±4LSB 的转换精度;
  - **ADC的采样保持时间必须设置的足够长**，以满足ADC通道间切换时内部采样电容能够真实反映外部信号的最快变化;
  - 更多细节，请参考应用笔记-- AN12217, S32K1xx ADC guidelines, spec and configuration – Application note (REV 0)

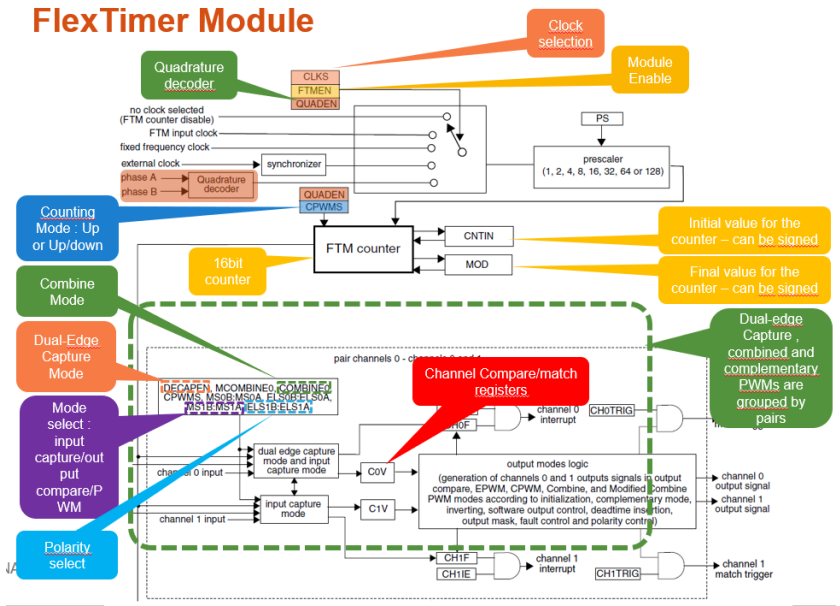


* The block depicts simplified schematic and doesn't show detailed latching. See section 'Trigger Latching and Arbitration' of ADC chapter for details

Figure 38-2. ADC0_PDB0 ADC Triggering scheme example

S32K定时器模块之FlexTimer

- **S32K系列MCU FlexTimer定时器功能特性**
  - One **16-bit** counter in each module
  - The counting can be up or up-down
  - Each channel can be configured for **input capture, output compare, or PWM generation**
  - PWM Modes: **Edge Aligned, Center Aligned, Combined, Complementary**
  - New combined mode to generate a PWM signal (with independent control of both edges of PWM signal)
  - **Complementary outputs, include the dead time insertion**
  - Up to 4 fault inputs for global fault control
  - **Dual edge capture for pulse and period width measurement**
  - FTM1 & FTM2 with **Quadrature decoder with input filters, relative position counting and interrupt on**
  - Dynamic synchronisation in HW or SW trigger mode



- **S32K系列MCU FlexTimer定时器数量**

Parameter	S32K11x		S32K14x			
	K116	K118	K142	K144	K146	K148
FlexTimer (16-bit counter) 8 channels	2x (16)		4x (32)		6x (48)	8x (64)

- **关于FlexTimer的使用，请参考以下应用笔记：**
  - AN5303 Features and Operation Modes of FlexTimer Module on S32K (REV 0)

Features and Operation Modes of FlexTimer Module on S32K

1. Introduction

This application note describes the multiple features and operation modes of the FlexTimer (FTM) module supported by code examples and oscilloscope snapshots. The FTM module is used on many Kinetis and Vybrid MCUs. However, this application note is primarily focused on the S32K product series of 32-bit automotive MCUs. The FTM module is an enhanced version of the Timer/PWM module (TPM) that provides new features for motor-control, lighting, and power-conversion applications. The key enhancements are:

- Signed-up counter
- Hardware dead-time insertion
- Fault-control inputs

Contents

S32K定时器模块之LPIT和LPTMR及RTC

- **S32K系列MCU 的LPIT定时器功能特性**
  - LPIT是一个**32-bit**的低功耗周期中断定时器
  - **4**个独立通道，**可设置独立的中断周期和中断ISR**
  - 支持DMA功能
  - 支持通道级联(chain)，组成64-bit定时器/计数器
  - 支持时间触发捕捉和触发输出到TRIGMUX模块
- **S32K系列MCU 的LPTMR定时器功能特性**
  - LPTMR是一个**16-bit**的低功耗通用定时器
  - **支持脉冲计数功能；**
  - 可选定时器比较异步唤醒中断—支持自由计数模式或者比较复位模式
  - 支持硬件触发输出
- **S32K系列MCU 的RTC定时器功能特性**
  - RTC是一个**32-bit**的秒计数器，带闹钟(alarm)功能；
  - 支持外部RTC_CLKIN输入32.768KHz有源时钟作为时钟源实现实时时间日历；
  - 支持RTC时钟信号输出(RTC_CLKOUT);
  - 通过其16-bit预分频器可以实现0.12ppm~3906ppm的时钟补偿；

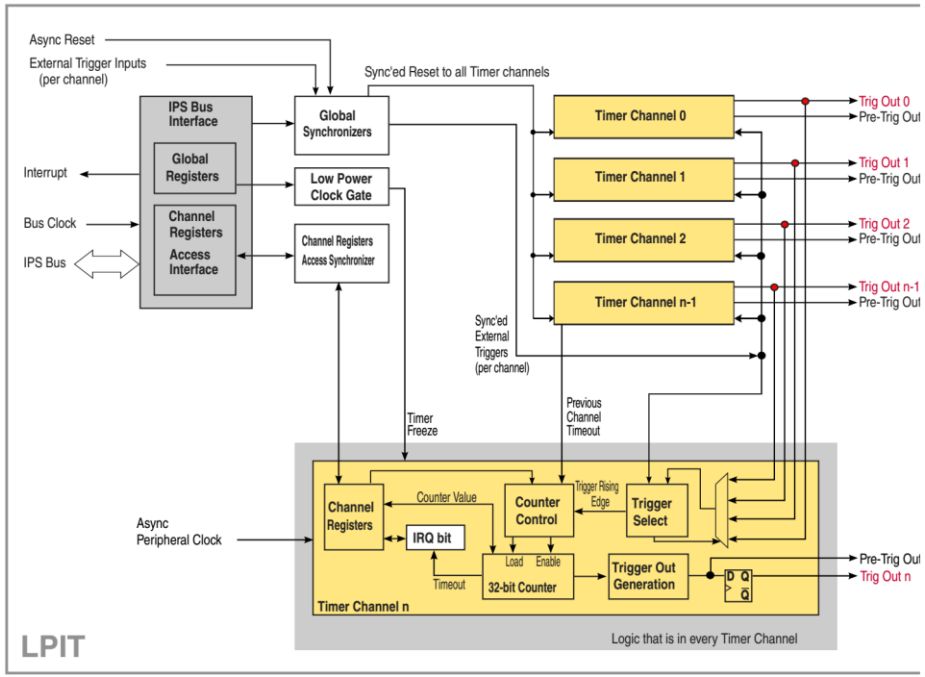


Table 47-1. Pulse counter input options

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	TRGMUX output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	LPTMR_ALT3 pin

48.2.3 RTC signal descriptions

Table 48-1. RTC signal descriptions

Signal	Description	I/O
RTC_CLKOUT	Prescaler square-wave output or RTC 32.768 kHz clock	O

48.2.3.1 RTC clock output

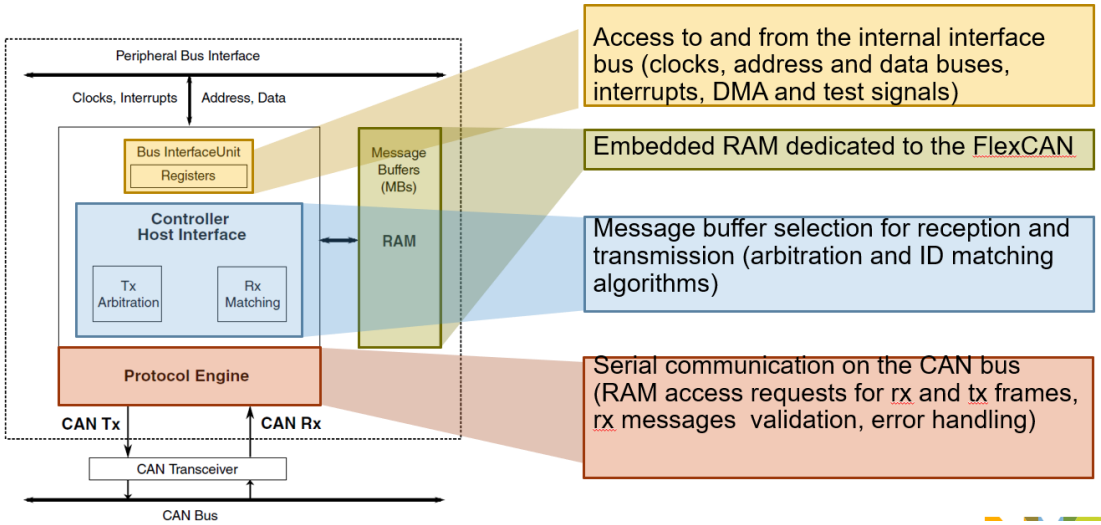
The RTC_CLKOUT signal can output either a square wave prescaler output (configurable to 1, 2, 4, 8, 16, 32, 64 or 128 Hz) or the RTC 32.768 kHz clock.

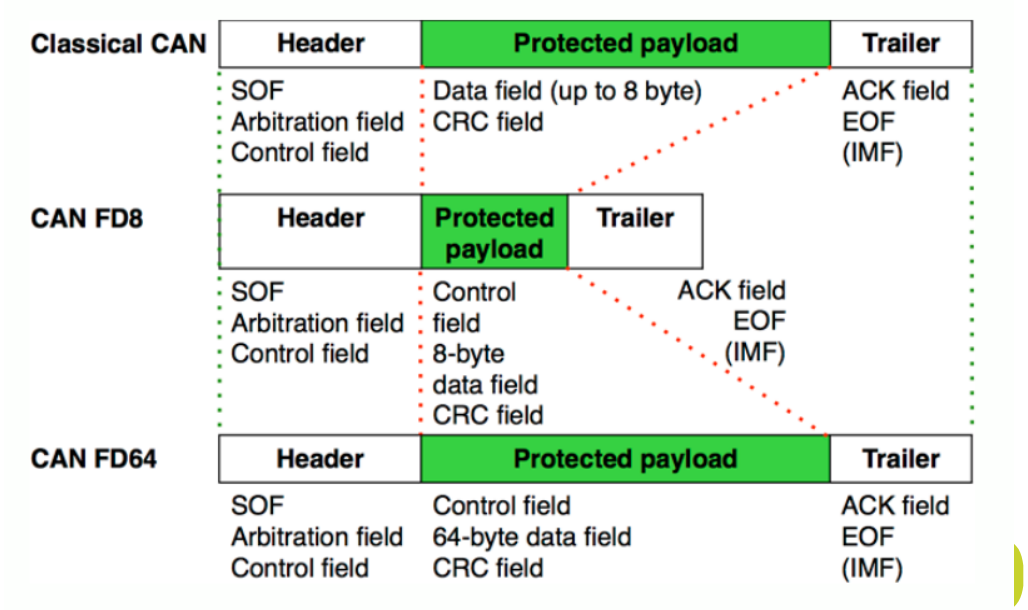
S32K系列MCU的FlexCAN 模块

- **S32K系列MCU的FlexCAN实现了CAN-FD和CAN 2.0A/B协议规范**
  - 标准数据帧;
  - 扩展数据帧;
  - 0~64B数据长度;
  - 可编程通信波特率，最高可达8Mbps(**CAN-FD**):
  - 内容相关地址;
  - 兼容ISO11898-1标准;
  - 灵活可配的消息信箱(MB-mailbox)以接收/发送0 to 8, 16, 32 or 64 字节数据;
  - 每个MB可以独立的配置为发送或者接收，支持标准帧和扩展帧;
  - 独立的可屏蔽TX/RX中断和RX ID滤波配置
  - **MB0~MB5可以配置为FIFO功能并支持DMA接收(仅CAN 2.0消息帧);**
  - 强大的RX FIFO ID滤波功能，可以支持多达完整的128个扩展帧/256个标准帧或者512个8-bit部分ID匹配滤波;
  - 支持STOP低功耗模式虚拟网唤醒功能(not VLPS);

Table 53-1. FlexCAN instances and features

Chip	Instances	Number of Message Buffers (MB)	ISO CAN FD feature [1]	PNET feature	External time tick	CTRL2 register [2] reset value
S32K116	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
S32K118	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
S32K142	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
	FlexCAN1	16 MBs	No	No	No	0x00B0_0000
S32K144	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
	FlexCAN1	16 MBs	No	No	No	0x00B0_0000
	FlexCAN2	16 MBs	No	No	No	0x00B0_0000
S32K146	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
	FlexCAN1	32 MBs	Yes	No	No	0x00A0_0000
	FlexCAN2	16 MBs	No	No	No	0x00B0_0000
S32K148	FlexCAN0	32 MBs	Yes	Yes	Yes	0x00A0_0000
	FlexCAN1	32 MBs	Yes	No	No	0x00A0_0000
	FlexCAN2	32 MBs	Yes	No	No	0x00A0_0000





Faster and longer: Larger payloads improve the protocol efficiency and lead to a higher throughput

S32K ENET—以太网模块

- **S32K148**集成了一个10/100-Mbit/s 以太网MAC, 实现3个网络层硬件加功能速：eg. IP, TCP, UDP, and ICMP；
  - 实现完整的IEEE 802.3协议规范，提供前导码生成、帧填充、CRC自动生成和检查；
  - 支持动态配置10/100M-Bits模式
  - 支持10/100 Mbit/s 全双工和半双工通信
  - 支持4-bit MII和2-bit RMII接口与以太网收发器连接
  - 支持IEEE 802.1Q VLAN-tagged 帧
  - 提供内部回环自测
  - MDIO支持IEEE 802.3 Clause 22和Clause 45)
- 在S32K SDK中集成了基于FreeRTOS和LwIP的TCP/IP协议栈可做学习参考；
- 汽车以太网物理层为100BASE-T1标准的非屏蔽双绞线，工业以太网的物理层为100BASE-TX标准，二者不能直接连接，需要进行物理层转换；仅支持点到点直连，多个节点互联需要以太网交换器；
  - 常见的汽车以太网收发器，eg. NXP **TJA1101**和**TJA1102**；
  - 常见的工业以太网收发器，eg. NS **DP83848C**；
  - 常见的汽车以太网交换机，eg. NXP **SJA1105**；

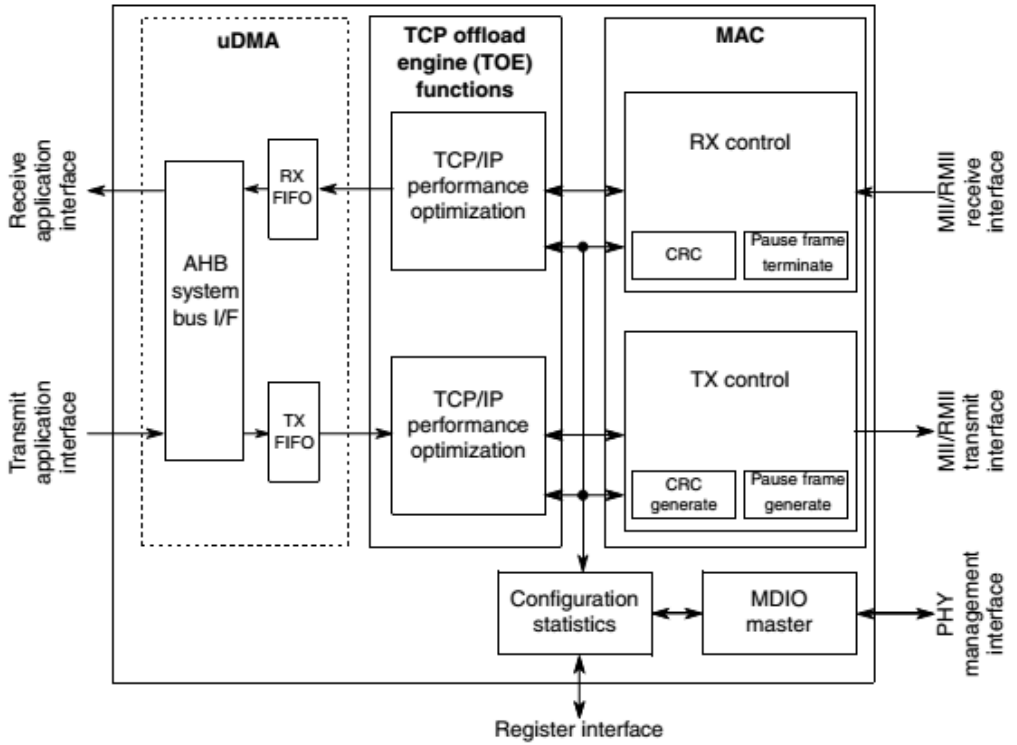
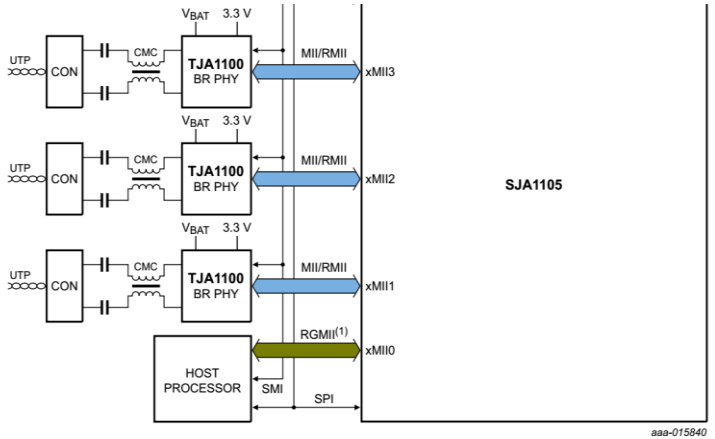
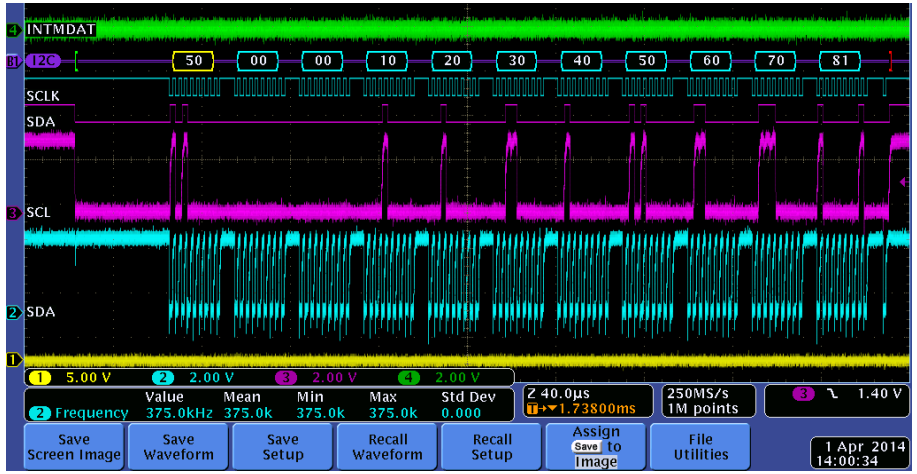


Figure 52-1. Ethernet MAC-NET core block diagram



S32K FlexIO 模块

- **FlexIO** = **Flex**ible **I**nput and **O**utput，集成若干**32-bit**串行移位寄存器(**shifter**)、**16-bit**定时器(**timer**)**并与若干MCU引脚I/O连接，从而**可以软件配置灵活实现不同的功能外设；
- 可编程逻辑产生复杂的波形输出
- 可以模拟标准的串行通信协议：eg. UART, SPI, I2C, I2S, LCD RGB, PWM, etc.
- 自动产生通信时序，无需CPU干预，从而减小CPU loading；
- 支持DMA；
- S32K SDK中集成现成的FlexIO模拟串行通信协议驱动程序；



- ***Tips***：关于FlexIO的使用，请参考以下应用笔记：**AN12174, Using FlexIO to emulate communications and timing peripherals** (REV 0)

52.1.1 FlexIO Configuration

Table 52-1. FlexIO Configuration

	Timers	Shifters	Pins
Number	4	4	8

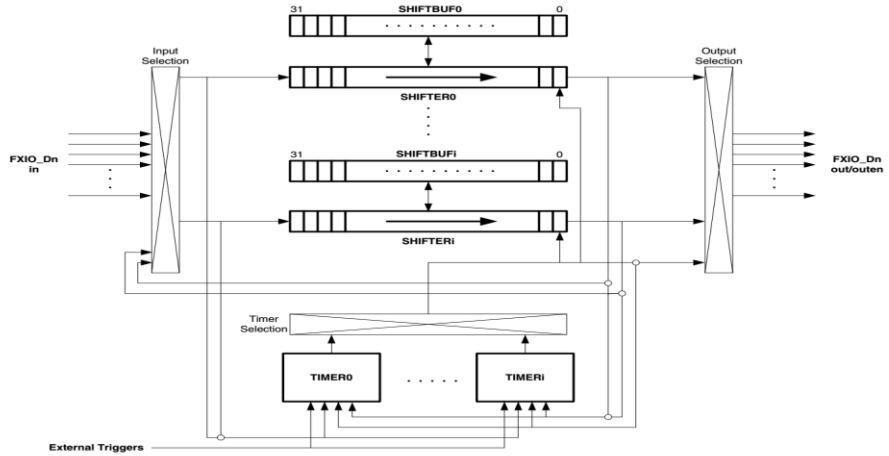


Figure 52-1. FlexIO block diagram

Use Case	Supported using	FlexIO usage	Comments
UART	Two independent parts **Transmit & Receive:** One Timer, One Shifter, One Pin **(TX)** One Timer, One Shifter, One Pin **(RX)**	50%	• Allows polling and interrupt mode • Configurable bit order (bit swapped buff MSB first) • Multiple transfers can be supported using DMA controller • Does not support automatic insertion of parity bits
SPI Master	Two timers, Two shifters, Four pins	50%	CPHA=0 and CPHA=1 supported
SPI Slave	One timer, Two shifters, Four pins	33%	CPHA=0 and CPHA=1 supported
I2C Master	Two timers, Two shifters, Two pins	50%	FlexIO inserts a stop bit after every word to generate/verify the ACK/NACK
I2S Master	Two timers, Two shifters, Four pins	50%	Data transfers can be supported using the DMA

3. S32K系列MCU硬件设计要点

- ✓S32K系列MCU的供电电路设计
- ✓S32K系列MCU的时钟电路设计
- ✓S32K系列MCU的调试接口电路设计
- ✓S32K系列MCU的复位和NMI引脚电路设计
- ✓S32K系列MCU的CAN和LPI2C通信电路设计

S32K系列MCU的供电电路设计

- S32K1系列MCU支持3.3V和5V系统供电(VDD)；
  - 每组VDD至少需要一个**0.1uF**的旁路(bypass)电容且需要就**近放**置;
- 系统DC-DC/LDO电源输出做VDD和VDDA时增加一个**10uF**的储能(bulk)电容,有利于电源的稳定；
- VDDA/VSS和VREFH/VREFL为ADC/ACMP参考电源，采样高精度LDO单独供电，可提高其工作性能；
  - 模拟电源VDDA&VREFH与数字电源VDD之间应当采样0Ω电阻或者磁阻进行隔离，且走线尽量远离干扰源(比如晶振电路、高速翻转的IO引脚等)

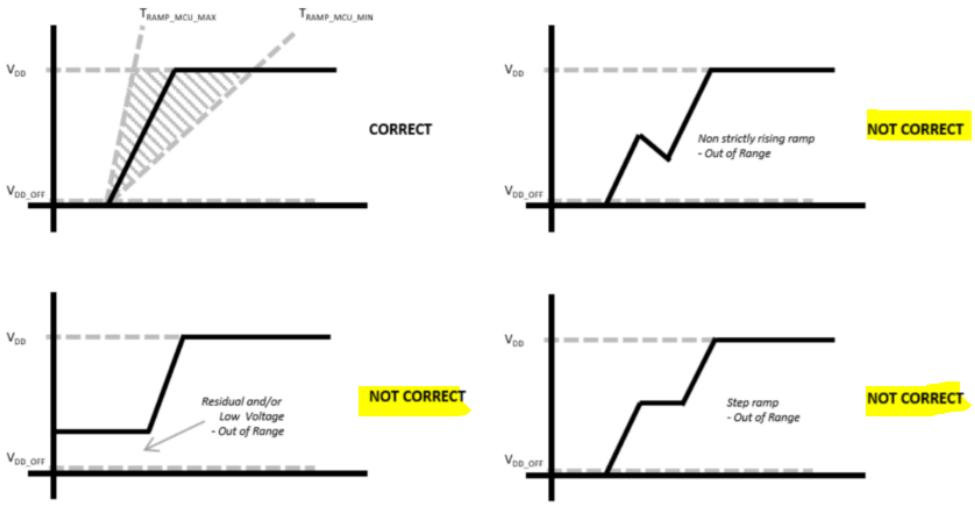
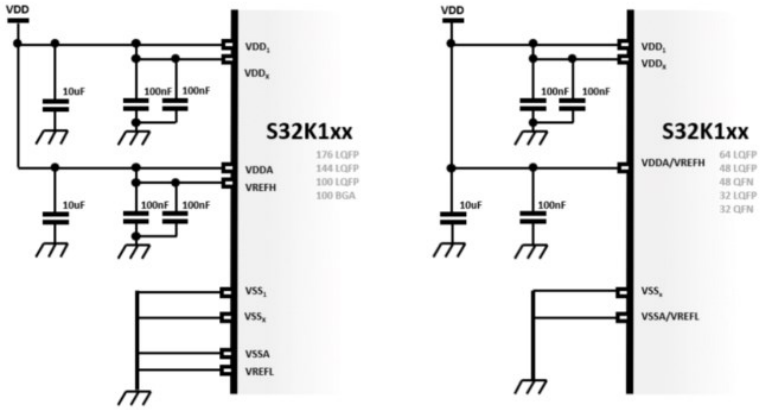


Figure 3. MCU power supply ramp rate



Power Domain	Description	Voltage	Bulk/Bypass Capacitor for domain						Decoupling Capacitor per pin	Characteristics
			Package							
			176 LQFP	144 LQFP	100 LQFP/ BGA	64 LQFP	48 LQFP/ QFN	32 LQFP/ QFN		
VDDX[1]	Supply voltage	5 V/3.3 V	10uF	10uF	10uF	10uF	10uF	10uF	0.1uF	X7R Ceramic
VDDA[1]	Analog supply voltage		10uF	10uF	10uF				0.1uF	X7R Ceramic
VREFH[1]	ADC reference voltage high								0.1uF	X7R Ceramic
VSS	Ground	GND	VSS, VSSA and VREFL must be shorted to GND at package level.							
VSSA	Analog Ground									
VREFL	ADC reference voltage low									

1. VDD and VDDA must be shorted to a common reference on PCB. Appropriate decoupling capacitors to be used to filter noise on the supplies

Table 1. Absolute maximum ratings

Symbol	Parameter	Conditions[1]	Min	Max	Unit
V_{DD}[2]	2.7 V - 5. 5V input supply voltage	—	-0.3	5.8[3]	V
V_{REFH}	3.3 V / 5.0 V ADC high reference voltage	—	-0.3	5.8[3]	V
$I_{INJPAD\_DC\_ABS}$[4]	Continuous DC input current (positive / negative) that can be injected into an I/O pin	—	-3	+3	mA
V_{IN_DC}	Continuous DC Voltage on any I/O pin with respect to V_{SS}	—	-0.8	5.8[5]	V
$I_{INJSUM\_DC\_ABS}$	Sum of absolute value of injected currents on all the pins (Continuous DC limit)	—	—	30	mA
T_{ramp}[6]	ECU supply ramp rate	—	0.5 V/min	500 V/ms	—
T_{ramp_MCU}[7]	MCU supply ramp rate	—	0.5 V/min	100 V/ms	—
T_A[8]	Ambient temperature	—	-40	125	°C

S32K系列MCU的时钟电路设计

- S32K系列MCU支持外部4~40MHz的无源石英晶振(Crystal)输入或者最大不超过50(S32K14x)/48MHz(S32K11x)的有源时钟输入作为系统参考时钟源；
  - **选择无源石英晶振时**，需要并联两个到地的pF级振荡电容，具体外设电路参数需要晶振厂家匹配决定；
  - **选择有源时钟作时**，EXTAL作为输入，XTAL引脚悬空即可；
  - 无高精度定时和高度通信总线(比如CAN总线和以太网)时，可以使用内部集成的8MHz SIRC或者48MHz FIRC时钟作为系统参考时钟，以降低系统功耗，同时提高系统EMC特性；
- 时钟电路应尽量靠近MCU引脚、且将其走线包地以提高其抗干扰能力；

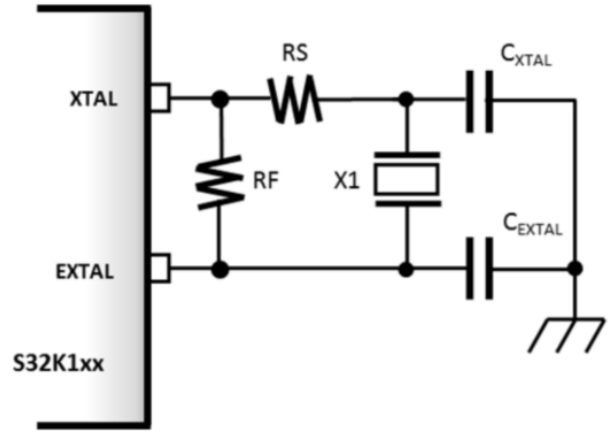


Figure 2. Reference oscillator circuit

Feedback Resistor

- When Low-gain is selected, internal RF will be selected, and external RF is not required.
- When High-gain is selected, external RF(1M Ohm) need to be connected for proper operation of crystal. For external resistor, up to 5% tolerance is allowed.

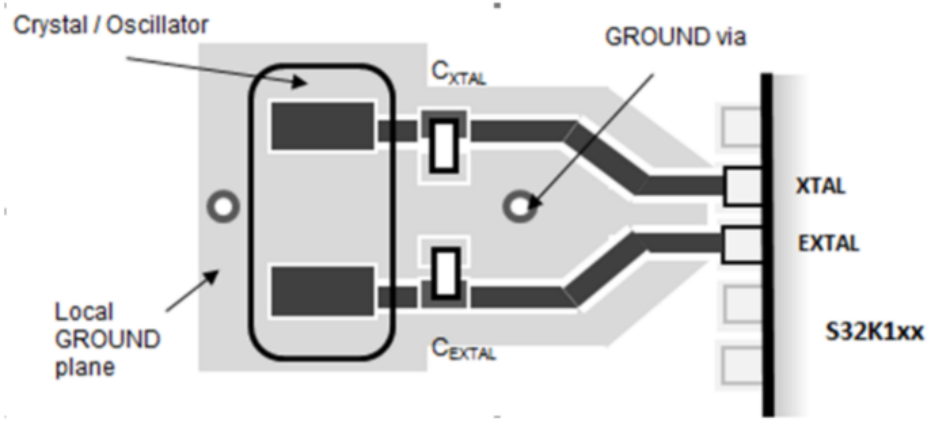


Figure 3. Suggested crystal oscillator layout

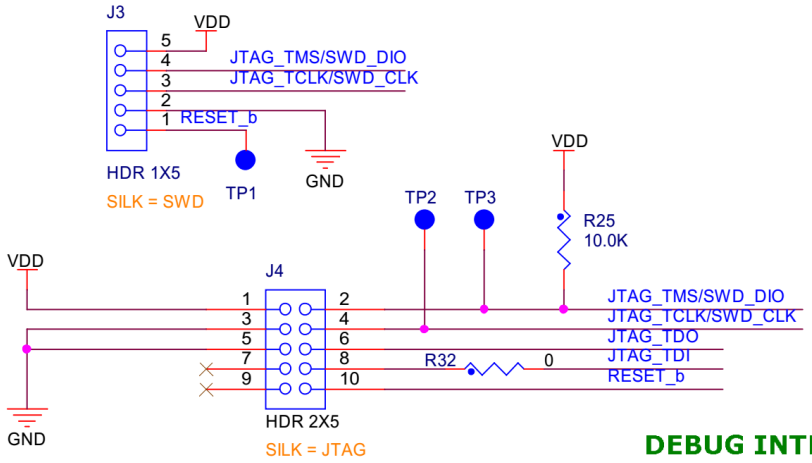
Table 18. External System Oscillator frequency specifications

Symbol	Description	Min.		Typ.		Max.		Unit	Notes
		S32K14x	S32K11x	S32K14x	S32K11x	S32K14x	S32K11x		
f_{osc_hi}	Oscillator crystal or resonator frequency	4		—		40 [1]		MHz	
f_{ec_extal}	Input clock frequency (external clock mode)	—		—		50	48	MHz	2
t_{dc_extal}	Input clock duty cycle (external clock mode)	48		50		52		%	2
t_{cst}	Crystal Start-up Time								
	8 MHz low-gain mode (HGO=0)	—		1.5		—		ms	3
	8 MHz high-gain mode (HGO=1)	—		2.5		—			
	40 MHz low-gain mode (HGO=0)	—		2		—			
	40 MHz high-gain mode (HGO=1)	—		2		—			

1. For an ideal clock of 40 MHz, if permitted by application requirements, an error of +/- 5% is supported with 50% duty cycle
2. Frequencies below 40 MHz can be used for degraded duty cycle upto 40-60%
3. Proper PC board layout procedures must be followed to achieve specifications.

S32K系列MCU的SWD/JTAG调试接口电路设计

- S32K11x系列MCU仅支持SWD调试接口；
- S32K14x系列MCU支持SWD和JTAG调试接口和ETM跟踪(trace S32K148 only)调试接口；
  - ETM跟踪调试功能需要高级的调试器支持(比如Lauterbach、J-LINK等)
  - SWD与JTAG信号复用(如下表)；
  - 使用SWD接口可以节省两个MCU引脚；
- 调试接口电路需要包含电压(**VDD**，用于调试器测量目标MCU供电情况)、地**GND**(为调试和控制信号通过电平参考)和复位信号(**RESET**，用于调试时硬件复位MCU)以及调试控制信号(SWD—**SWDIO**和**SWCLK**，JTAG—**TMS**、**TCK**、**TDI**和**TDO**)；
  - 典型的S32K14x系列MCU调试接口电路如下：



DEBUG INTERFACE

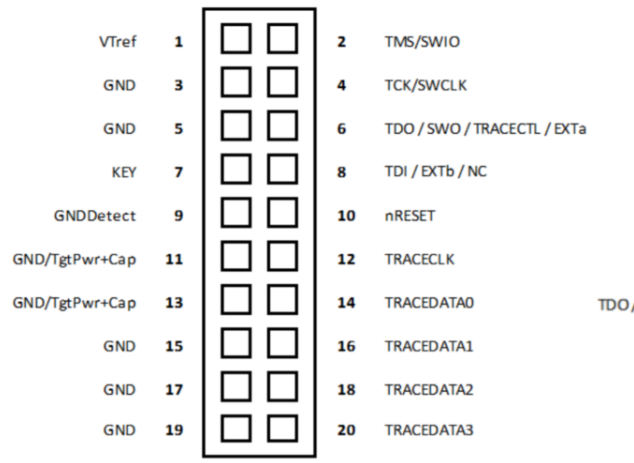


Figure 4. 20-pin Cortex Debug D ETM connector pin layout

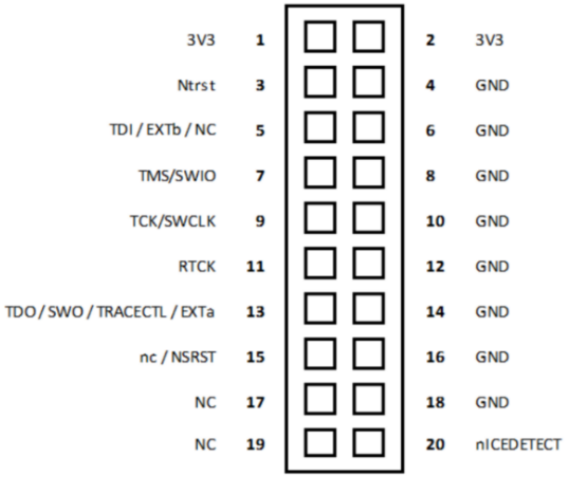


Figure 6. 20-pin IDC connector

Table 3. JTAG signal description

JTAG Mode	SWD Mode	Signal
TCK	SWCLK	Clock into the core
TDI	-	JTAG Test Data Input
TDO	SWV	JTAG Test Data Output / SWV trace data output (SWO)
TMS	SWDIO	JTAG Test Mode Select / SWD data in/out
GND	GND	-

The pull up/down resitors for the JTAG signals are included internally by the default pad configuration. See the device reference manual and datasheet.

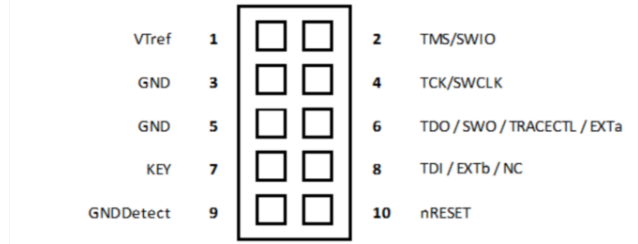


Figure 5. 10-pin Cortex Debug connector pin layout

Table 56-1. Supported debug interfaces

Chip	Supported debug interfaces			
	IEEE 1149.1 JTAG	Serial Wire Debug(SWD)	SWO	4-pin parallel trace port[1]
S32K116	No [2]	Yes	No	No[3]
S32K118	No[2]	Yes	No	No[3]
S32K142	Yes	Yes	Yes	No
S32K146	Yes	Yes	Yes	No
S32K144	Yes	Yes	Yes	No
S32K148	Yes	Yes	Yes	Yes

S32K系列MCU的复位和NMI引脚电路设计

- S32K系列MCU的**RESET**引脚(**PTA5**)是一个开漏且带内部内部弱上拉的低电平有效双向I/O
  - 所有的MCU复位源都会将RESET引脚拉低至少128个总线时钟周期，并等待Flash存储器初始化完成才会释放(拉高)；
  - 为了保证MCU系统可靠工作，推荐在RESET外部上拉一个4.7K/10KΩ电阻到VDD，同时并联一个0.1uF的滤波电容到GND(该滤波电容需要靠近MCU引脚就近放置)
- S32K系列MCU的**NMI**引脚(**PTD3**)是ARM Cortex M4F/M0+ CPU内核不可屏蔽中断(**NMI**—**N**on-**M**askable **I**nterrupt)异常的输入，地电平有效；
  - 默认NMI异常是使能的，且优先级仅次于RESET；

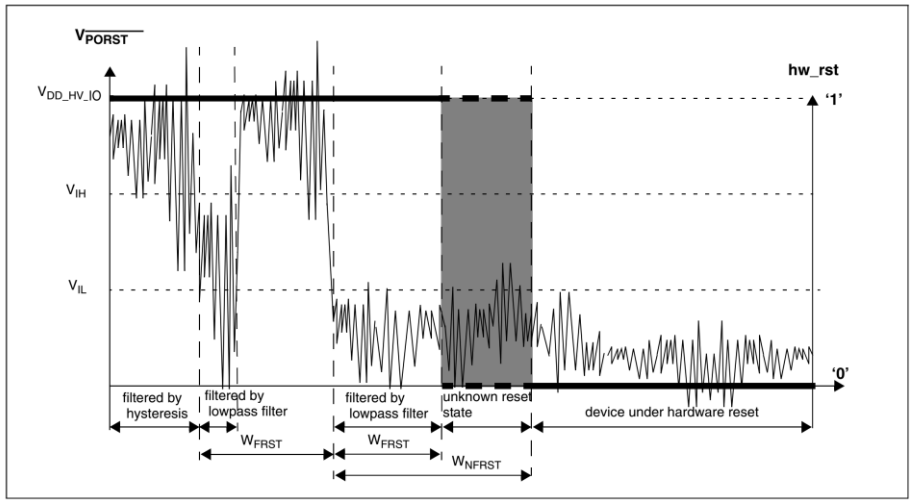


Figure 10. Noise filtering on reset signal

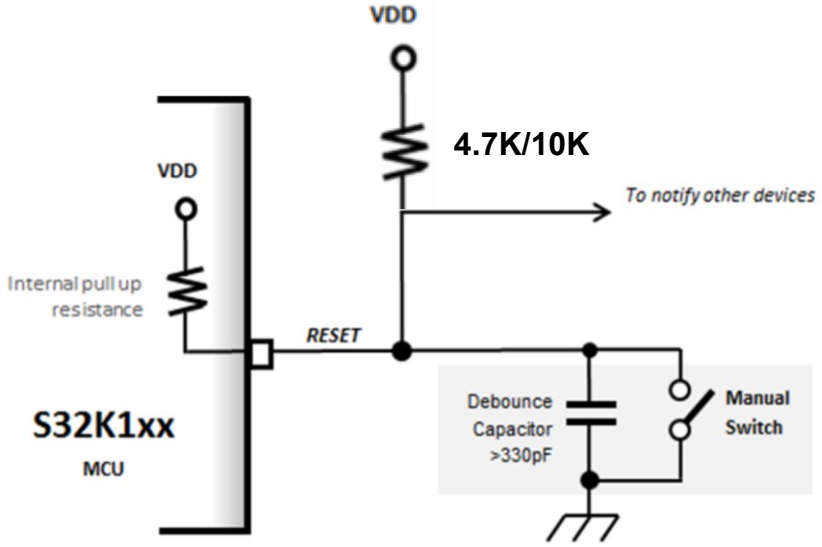


Figure 8. RESET circuit

Table 10. General switching specifications

Symbol	Description	Min.	Max.	Unit	Notes
	GPIO pin interrupt pulse width (digital glitch filter disabled) — Synchronous path	1.5	—	Bus clock cycles	1, 2
	GPIO pin interrupt pulse width (digital glitch filter disabled, passive filter disabled) — Asynchronous path	50	—	ns	3
WFRST	RESET input filtered pulse	—	10	ns	4
WNFRST	RESET input not filtered pulse	Maximum of (100 ns, bus clock period)	—	ns	5

1. This is the minimum pulse width that is guaranteed to pass through the pin synchronization circuitry. Shorter pulses may or may not be recognized. In Stop and VLPS modes, the synchronizer is bypassed so shorter pulses can be recognized in that case.
2. The greater of synchronous and asynchronous timing must be met.
3. These pins do not have a passive filter on the inputs. This is the shortest pulse width that is guaranteed to be recognized.
4. Maximum length of RESET pulse which will be the filtered by internal filter only if PCR_PTA5[PFE] is at its reset value of 1'b1.
5. Minimum length of RESET pulse, guaranteed not to be filtered by the internal filter only if PCR_PTA5[PFE] is at its reset value of 1'b1. This number depends on the bus clock period also. In this case, minimum pulse width which will cause reset is 250 ns. For faster clock frequencies which have clock period less than 100 ns, the minimum pulse width not filtered will

I/O parameters

be 100 ns. After this filtering mechanism, the software has an option to put additional filtering in addition to this, by means of PCM_RPC register and/or PORT_DFER register for PTA5.

S32K系列MCU的CAN通信接口电路设计

- CAN bus为汽车ECU间通信最常见的通信总线，其以差分信号通过半双工通信能力、具有抗干扰能力强、误码率低等特点；
- CAN总线需要物理层的收发器(Transceiver)完成MCU端TTL电平信号到CAN bus差分电平信号的转换；
  - 常见的CAN收发器如NXP的TJA1042/3/4；
- 在CAN总线的最远两端需要端接120Ω的终端匹配电阻，并联形成60Ω的总线传输阻抗，以避免信号反射和衰减；
  - 在CAN通信节点串联共模电感可以增加总线的抗干扰能力；

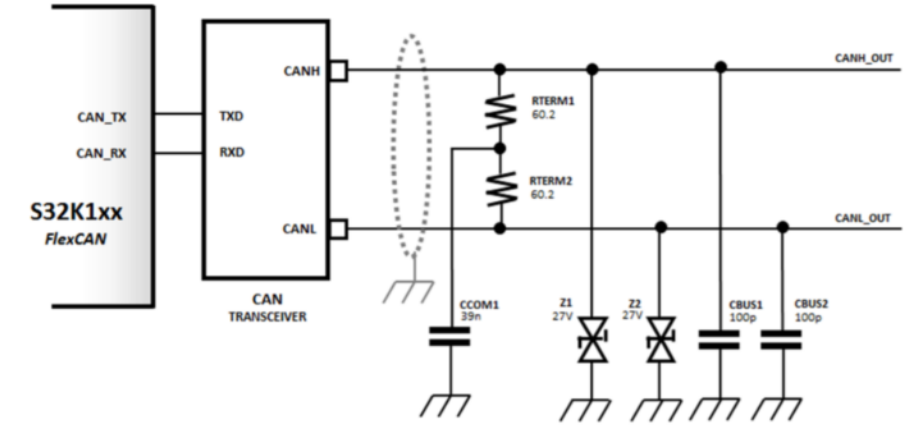


Figure 14. CAN physical transceiver circuit

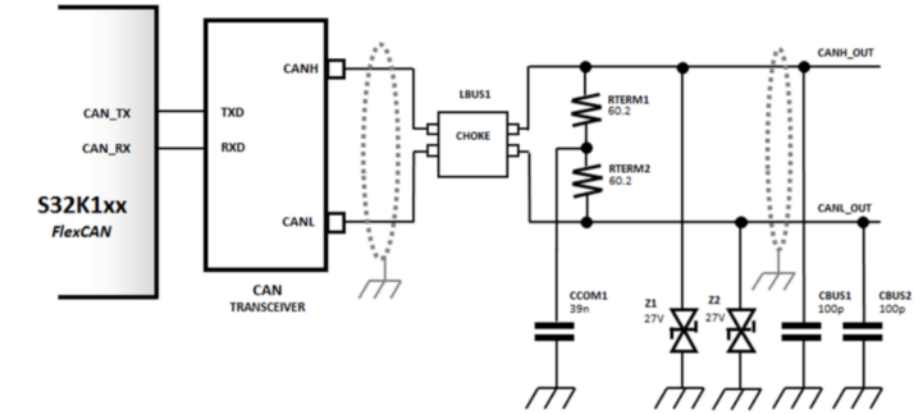


Figure 15. CAN physical transceiver circuit with common mode choke

S32K系列MCU的I2C通信接口电路设计

- I2C总线为一主(Master)多从(Slave)总线，支持标准模式(100Kbit/s)和高速模式(400Kbit/s)；
- I2C总线接口为**开漏结构(OD)**，外部需要接上拉电阻才能实现高电平输出；
  - 典型的上拉电阻为4.7KΩ(高速模式)和10KΩ(标准模式)
  - I2C总线空闲时SDL和SCL都是高电平
- I2C总线用于板级功能外设模块扩展，常用连接外部EEPROM存储器(比如AT24C16/32)或者传感器(比如光线传感器、加速度及磁力传感器等)；

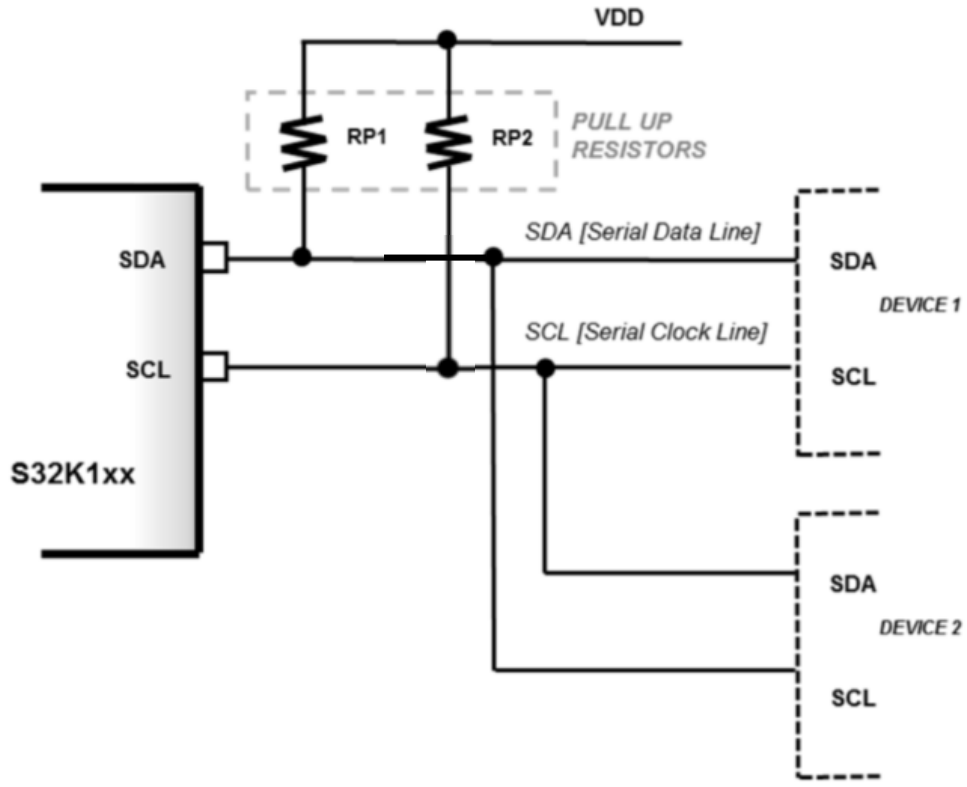
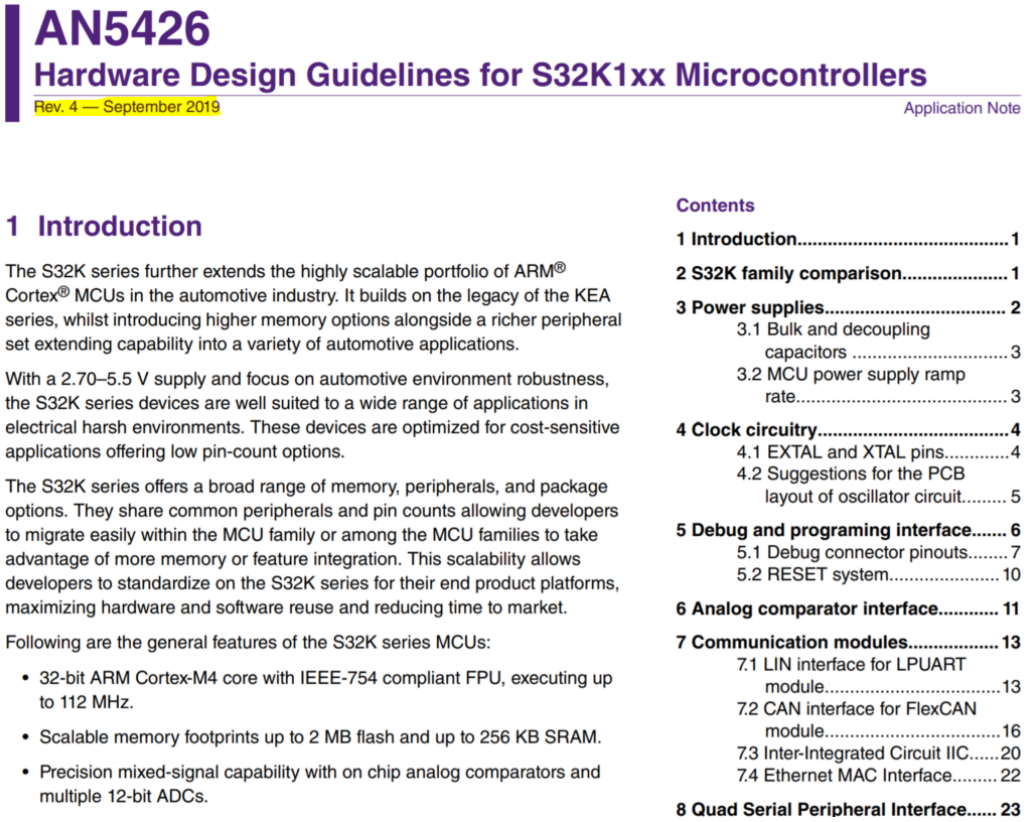


Figure 18. Connection of I2C-bus devices to the I2C-bus

S32K系列MCU的其他功能电路设计

- 此外，S32K系列MCU还提供了SPI、UART等通信接口，可以用于连接外部传感器和功能模块，比如压力传感器、GPS模块和蓝牙通信模块等；

- 更多S32K系列MCU的硬件电路设计，请参考以下官网应用笔记(点击跳转阅读或者下载)：
  - AN5426, Hardware Design Guidelines for S32K1xx Microcontrollers



AN5426

Hardware Design Guidelines for S32K1xx Microcontrollers

Rev. 4 — September 2019 Application Note

1 Introduction

The S32K series further extends the highly scalable portfolio of ARM® Cortex® MCUs in the automotive industry. It builds on the legacy of the KEA series, whilst introducing higher memory options alongside a richer peripheral set extending capability into a variety of automotive applications.

With a 2.70–5.5 V supply and focus on automotive environment robustness, the S32K series devices are well suited to a wide range of applications in electrical harsh environments. These devices are optimized for cost-sensitive applications offering low pin-count options.

The S32K series offers a broad range of memory, peripherals, and package options. They share common peripherals and pin counts allowing developers to migrate easily within the MCU family or among the MCU families to take advantage of more memory or feature integration. This scalability allows developers to standardize on the S32K series for their end product platforms, maximizing hardware and software reuse and reducing time to market.

Following are the general features of the S32K series MCUs:

- 32-bit ARM Cortex-M4 core with IEEE-754 compliant FPU, executing up to 112 MHz.
- Scalable memory footprints up to 2 MB flash and up to 256 KB SRAM.
- Precision mixed-signal capability with on chip analog comparators and multiple 12-bit ADCs.

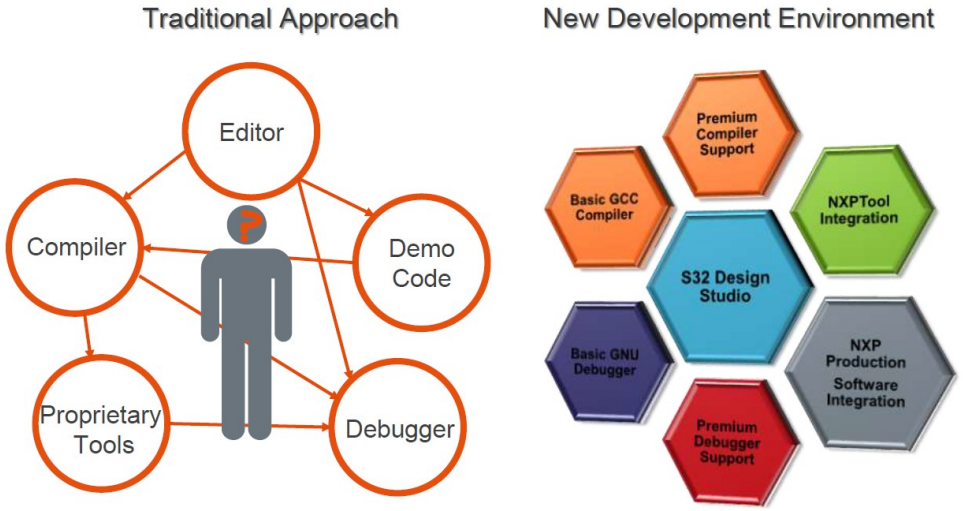
Contents

4. S32K系列MCU软件开发之 S32DS IDE和S32K SDK使用TIPS

- ✓NXP S32 Design Studio IDE功能概述
- ✓S32DS for ARM v2018.R1 IDE导出S32K SDK demo工程
- ✓S32K SDK软件架构和驱动组件API介绍
- ✓S32DS IDE及S32K SDK使用Tips

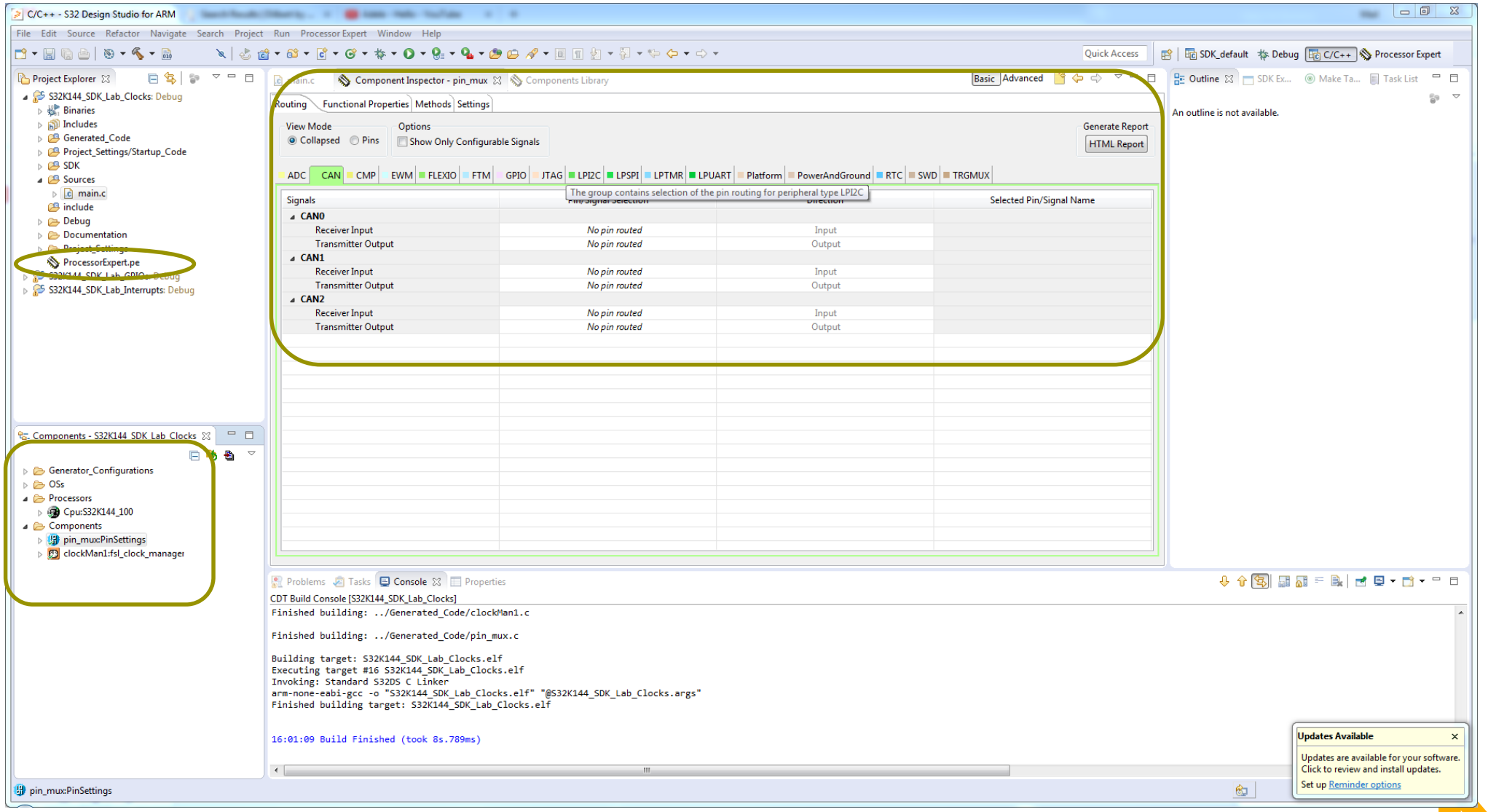
NXP S32 Design Studio IDE功能概述

- 免费下载安装使用；
- 无代码大小限制；
- Eclipse界面集成GNU编译器和调试工具链(GCC + **GDB**)
- 集成S32K SDK (以补丁包方式更新)及其图形化配置工具--Processor Expert；
- 支持第三方开发工具链集成，通过安装第三方工具链的Eclipse 插件方式；
- 统一的Eclipse界面支持NXP不同的汽车级MCU—eg.S32K, Qorivva MPC57xx, S32R
- S32DS for Arm v2018.R1 IDE下载地址：
  - S32 Design Studio for Arm 2018.R1 – Windows/Linux；
- S32DS IDE补丁及S32K最新RTM SDK下载链接：
  - S32 Design Studio for ARM v2018.R1 Update 9 with S32 SDK S32K1xx RTM 3.0.0 (REV UP9)
  - S32 Design Studio for ARM v2018.R1 Update 10 with S32 SDK S32K1xx SR 3.0.1 (REV UP10)



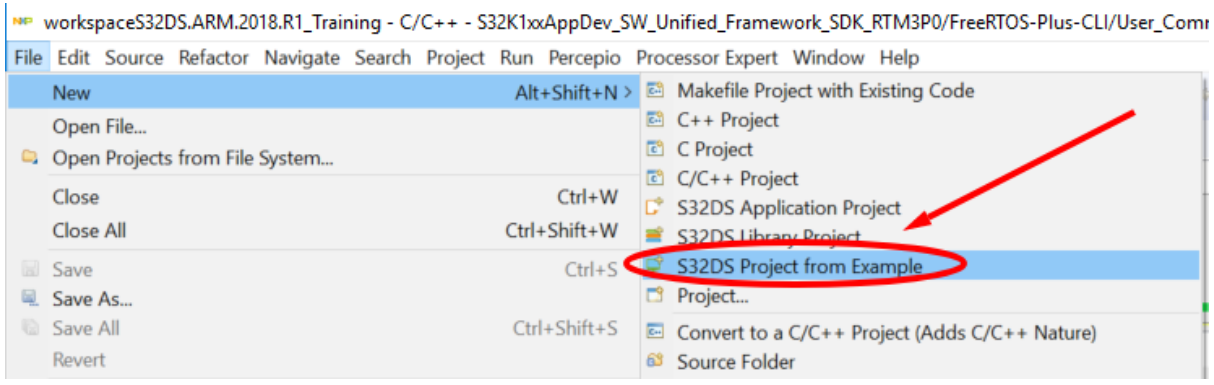
	Devices supported	Integrated NXP Tools	Integrated NXP Software	Graphical Tools	Compiler	Debugger	Built-in GDB interface
S32DS for ARM-based MCUs	KEA S32K14x MAC57D54H S32V234	FreeMASTER AMMCLib for KEA and S32K MCUs	S32K SDK KEA SDK MAC57xx SDK		GCC GreenHills IAR	IAR iSystem Lauterbach P&E and SEGGER	SEGGER J-Link P&E Multilink Cyclone OpenSDA
S32DS for Power Architecture based MCUs	MPC56xx MPC57xx S32R274	FreeMASTER AMMCLib for MPC56xx and MPC57xx MCUs	MPC5748G SDK MPC5746C SDK	SPT graph tools	GCC GreenHills Diab	iSystem Lauterbach P&E and PLS	P&E Multilink Cyclone OpenSDA
S32DS for vision processors	S32V234	NXP APU Compiler ISP assembler DDR Configuration tools	Vision SDK Linux BSP	ISP and APEX graph tools	GCC	Lauterbach P&E	P&E Multilink Cyclone OpenSDA

S32 Design Studio IDE的图形化SDK配置界面

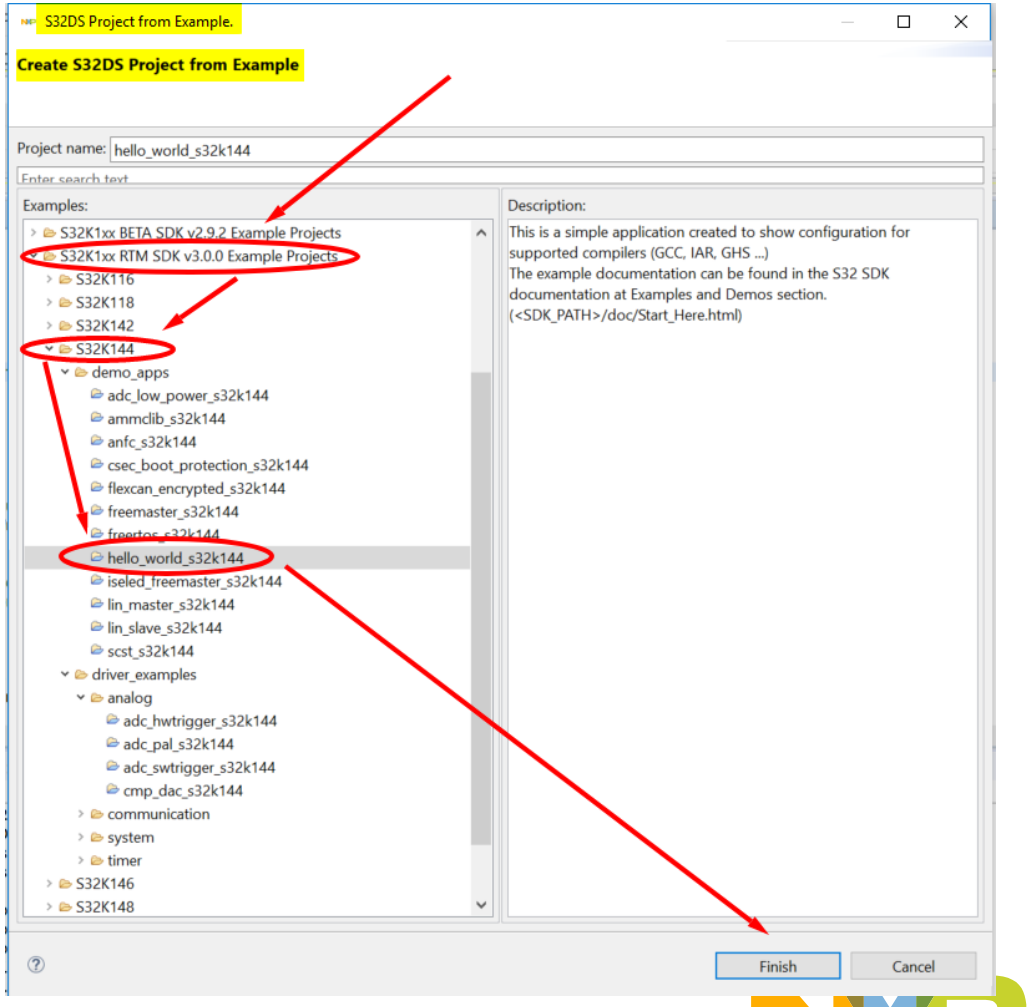


S32DS for ARM v2018.R1 IDE导出S32K SDK demo工程

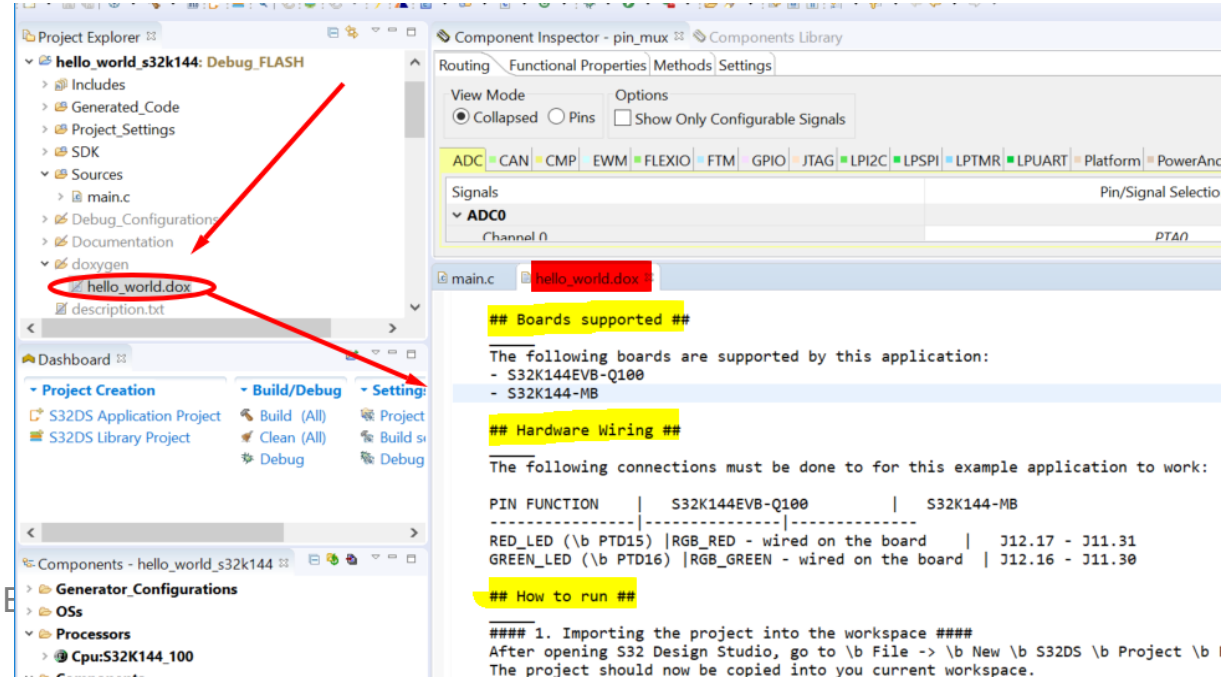
- ①File->S32DS Project from Example



- ②select SDK->MCU part->Demo project

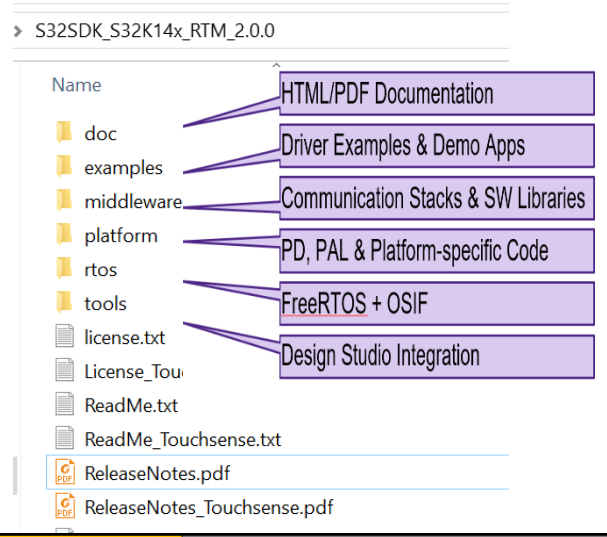


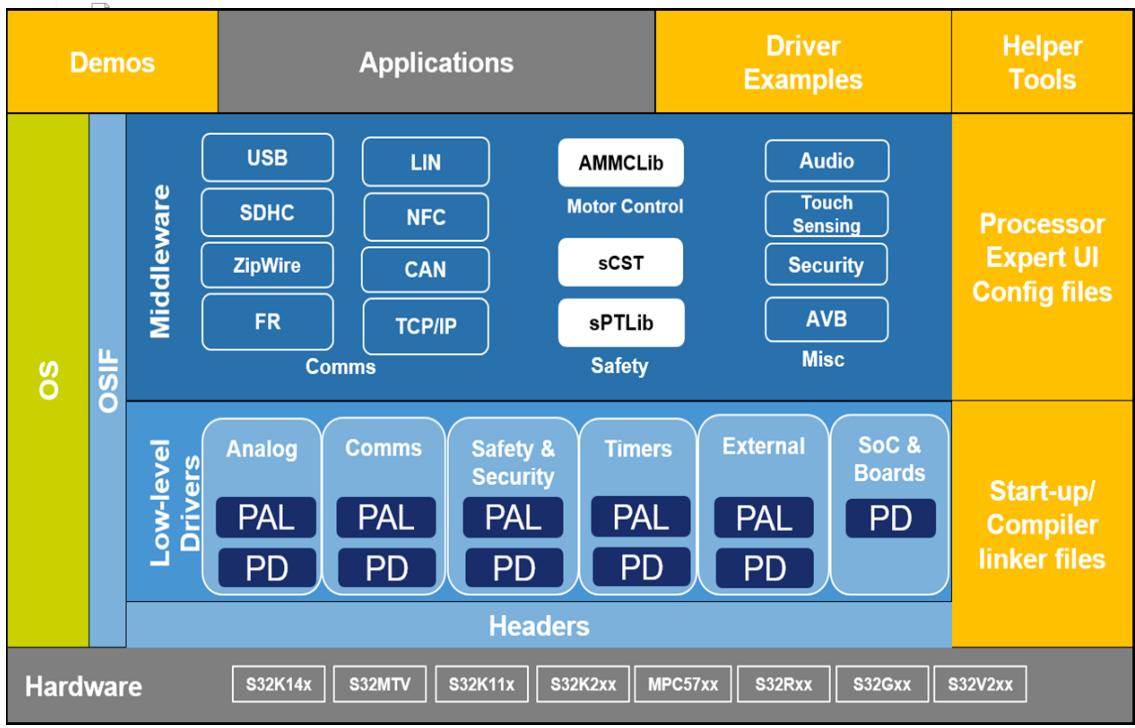
- ③read doxygen/<demo_name>.dox to know how to run:



S32K SDK – 软件架构与特性概述

- S32K SDK是NXP为S32K系列MCU开发的软件开发套件，其提供了S32K系列MCU的硬件操作接口，包括硬件抽象层、外设驱动、实时操作系统(RTOS)、各种通信协议栈、应用中间件(middleware)。旨在简化加速S32K系列MCU的应用程序开发；
- 通过S32DS IDE集成的处理器专家(Processor Expert)为SDK提供了一个可视化的图像配置界面和自动代码生成工具，简化了SDK的使用；
- S32K SDK是免费(permissive open-source license)以开放源代码方式提供给客户，同时也做了一定的测试，能够保证代码质量；推荐用户下载使用RTM版本的SDK；
- S32K SDK**支持多种编译器**，eg. GCC、IAR、Keil、GHS、DIAB等；
- S32K SDK除了通过**PD**和**PAL**组件提供底层驱动(**LLD**--Low Level **D**rivers)和中间件(Middleware)之外，还提供了不同工具链编译器的启动(Start-up)和链接文件(Linker filers)，移植好的FreeRTOS和RTOS接口层(OSIF)以及处理器专家(Processor Expert)配置GUI和丰富的demo程序和详细的帮助文档；
- ***Tips***：关于S32K SDK软件架构和编程思想，请参考阅读以下公众号文章(点击文章标题即可跳转阅读)：
  - 《S32K SDK使用详解之S32 SDK软件编程思想详解》；
  - 《S32K SDK使用详解之S32 SDK软件架构详解》；





S32K1xx SDK 驱动组件概况

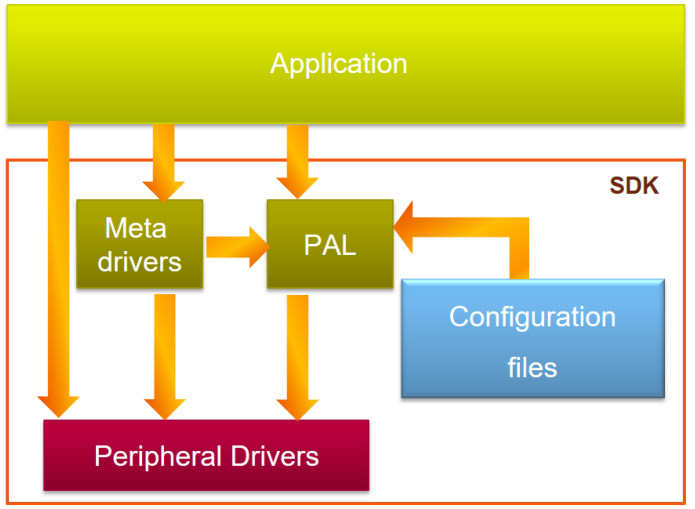
- SDK组件包含所有S32K系列MCU的功能外设模块驱动，和内核中断、系统时钟和功耗管理驱动代码；
- 提供LIN 和基于LwIP的TCP/IP 协议栈以及SBC 驱动;
- 集成电机控制库- AMMCLib和内核自测库--SCST;
- 提供移植好的 FreeRTOS操作系统;
- 所有S32K SDK组件可以通过处理器专家图形化配置和自动代码生成，且所有API在S32DS IDE中使用时支持拖拽(**drag & drop**)添加/调用;

3.1 Drivers

- ADC
- CMP
- CRC
- CSEc
- DMA
- EIM
- ENET
- ERM
- EWM
- FLASH
- FLASH_MX25L6433F
- FLEXCAN
- FLEXIO (I2C, SPI, I2S, UART profiles)
- FTM
- LIN
- LPI2C
- LPIT
- LPSPI
- LPTMR
- LPUART
- MCU (Clock Manager, Interrupt Manager, Power Ma
- MPU
- PINS
- PDB
- PHY_TJA110x
- QSPI
- RTC
- SAI
- TRGMUX
- WDOG

3.2 PAL

- ADC
- CAN
- I2C
- I2S
- IC
- MPU
- OC
- PWM
- SECURITY
- SPI
- TIMING
- UART
- WDG



3.3 Middleware

- LIN stack – provides support for LIN 2.1, LIN 2.2 and J2602 communication protocols
- TCP/IP stack – available for S32K148, for more details see TCP/IP stack release notes (in the SDK installation folder)
- SBC drivers – provides support for UJA1169 and UJA113x System Basis Chips

Note: *For Touch Sense, ISELED and NFC contact your Sales representative or FAE for more information.*

3.4 Libraries

- AMMCLib
- sCST – available for S32K144

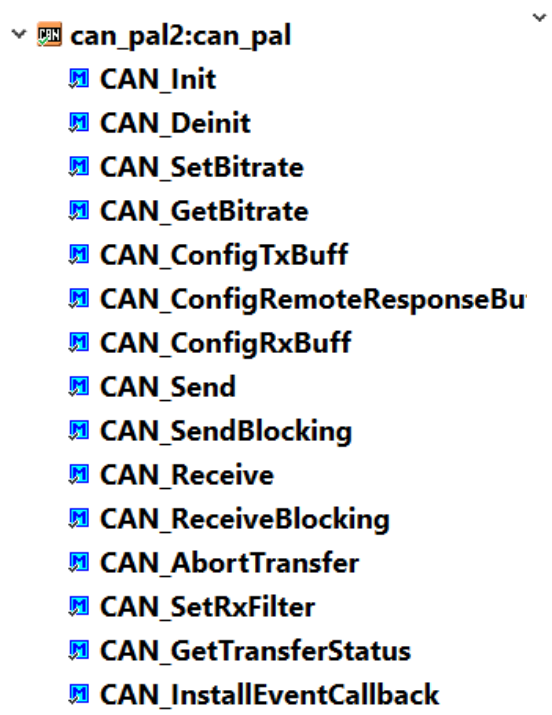
3.5 RTOS

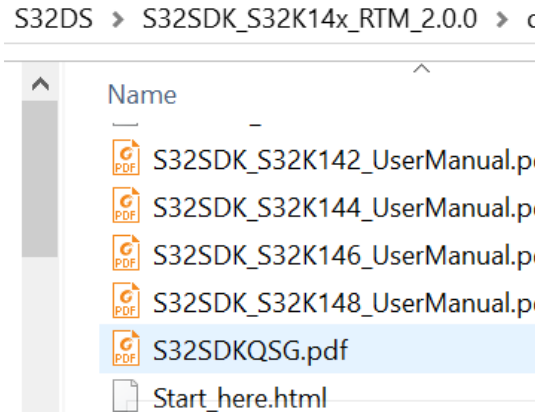
- FreeRTOS version 10.0.1

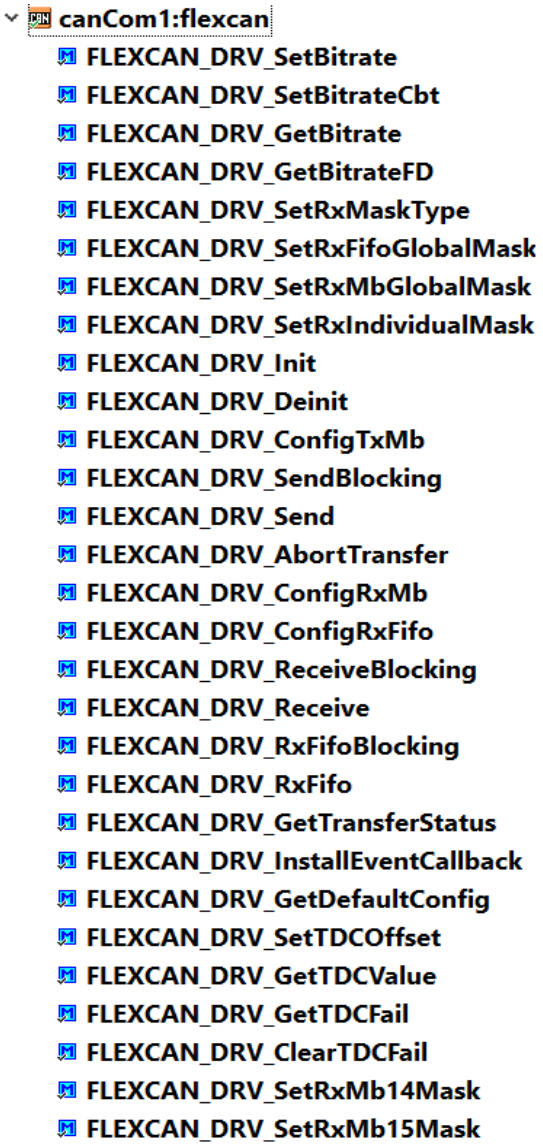
S32K SDK 外设驱动组件API 介绍

- PD层组件提供了丰富的API接口，支持对外设寄存器进行直接读写操作，其API函数以<peripheral name>_DRV_xxxx命名，提供对某一外设初始化(*_Init*)、反初始化(*_Deinit*)、配置(*_Config*)、数据发送(*_Send*)、数据接收(*_Receive*)、写寄存器(*_Set*)和读寄存器/状态(*_Get*)以及中断回调函数注册(*_InstallEventCallback*)等功能函数；
- PAL层组件对PD层组件功能进行了抽象和封装，提供了相对较少但功能强大的API接口，其API函数以<peripheral function name>_xxxx命名，提供对某一功能集外设初始化(*_Init*)、反初始化(*_Deinit*)、配置(*_Config*)、数据发送(*_Send*)、数据接收(*_Receive*)、写寄存器(*_Set*)和读寄存器/状态(*_Get*)以及中断回调函数注册(*_InstallEventCallback*)等功能函数；

- **For more details**: please refer to S32SDK_S32K1xx_UserManual.pdf under folder **S32DS\S32SDK_S32K14x_RTM_2.0.0\doc**







S32K SDK 使用要点

- 每个PAL/PD组件驱动都有一个定义为全局或者静态类型的状态结构体，其中保存着外设驱动的工作状态，在整个驱动代码工作时都将被用到；
- 驱动状态结构体的内容不允许被应用层代码使用或者修改；
- PD和PAL层驱动以及中间件软件都不包含用到的外设功能模块的时钟和引脚初始化，所以应用程序调用相应的API函数之前，必须调用以下3个API函数，完成MCU系统的时钟和引脚初始化；

```
/* Initialize and configure clocks */
CLOCK_SYS_Init(g_clockManConfigsArr, CLOCK_MANAGER_CONFIG_CNT,
               g_clockManCallbacksArr, CLOCK_MANAGER_CALLBACK_CNT);
CLOCK_SYS_UpdateConfiguration(0U, CLOCK_MANAGER_POLICY_AGREEMENT);

/* Initialize pins */
PINS_DRV_Init(NUM_OF_CONFIGURED_PINS, g_pin_mux_InitConfigArr);
```

- 在代码开发阶段，推荐通过配置编译器选项，增加DEV_ERROR_DETECT 全局宏定义(symbol)，从而打开SDK驱动代码的DEV_ASSERT断言工作机制，完成对API函数参数合法性检查，捕获应用代码错误；
- 将调试接口(**SWD/JTAG**)引脚复用为其他外设功能时，需要特别小心，尤其是MCU的RESET引脚，可能导致MCU无法连接调试和下载程序；

S32DS IDE及S32K SDK使用

《S32DS IDE使用tips--工程属性配置(编译选项和C编译器、汇编器及链接器设置)》

- 引言
- 1. 如何打开S32DS应用工程的属性设置
- 2. 设置Cross Settings
- 2.1 配置Create flash image
- 2.2 配置print size
- 3. 配置Target Processor
- 4. 配置Standard S32DS C Compiler(C编译器)
- 4.1预处理器设置(Preprocessor)
- 4.2 包含路径(Includes)
- 4.3设置优化等级(Optimization)
- 4.4 配置调试信息(debugging)
- 5. 配置Standard S32DS C Linker(C链接器)
- 5.1 添加/设置链接文件
- 5.2 添加用户库
- 6. 配置Standard S32DS Assembler (汇编器)
- 总结

- 《S32DS IDE使用Tips--应用程序开发实战实用技巧总结与详解(工欲善其事必先利其器)》

一、S32DS IDE工程管理篇

1. 打开/导入(Import)S32DS IDE工程
2. 备份/导出(Export)S32DSIDE工程
3. S32DS工程重命名(Rename)
4. 新建编译目标(Build Target)实现应用软件和SDK代码复用
5. 新建调试目标(Debug Target)实现个性化debug

二、S32DS IDE工程文件代码编辑阅读篇

1. 当前文件(File)/当前工程(Project)和当前工作空间(Workspace)快速文本搜索(Search Text)
2. 快速定位变量(Variable/函数(Function)/宏(Marco)定义
3. 快速定位/打开文件存储目录(Show In -> System Explorer)
4. 使用第三方编辑器打开工程文件(Open With -> System Editor)
5. 文本文件内容比较(Compare With -> Each Other)功能使用
6. 查看函数/变量调用层级(Call Hierarchy/Ctrl + Alt + H)

三、S32DS IDE工程程序下载与调试篇

1. 正常debug与Attach的区别与配置
2. 设置调试默认断点
3. 查看全局变量和局部变量(Variables & Expressions)
4. 查看和保存存储器内容(Memory)
5. 查看和修改CPU内核通用寄存器(Registers)
6. 查看和修改CPU内核特殊寄存器(SPR Registers)
7. 查看和修改外设寄存器(EmbSys Registers)
8. 查看函数调用栈(Call Stack)，分析程序调用过程
9. 查看汇编代码和进行汇编指令级调试(Disassembly)
10. 使用RTOS调试插件

S32DS IDE及S32K SDK使用Tips

- 关于S32DS IDE使用Tips和S32K SDK使用细节，请参考右侧公众号文章(**点击文档标题即可跳转阅读**)：

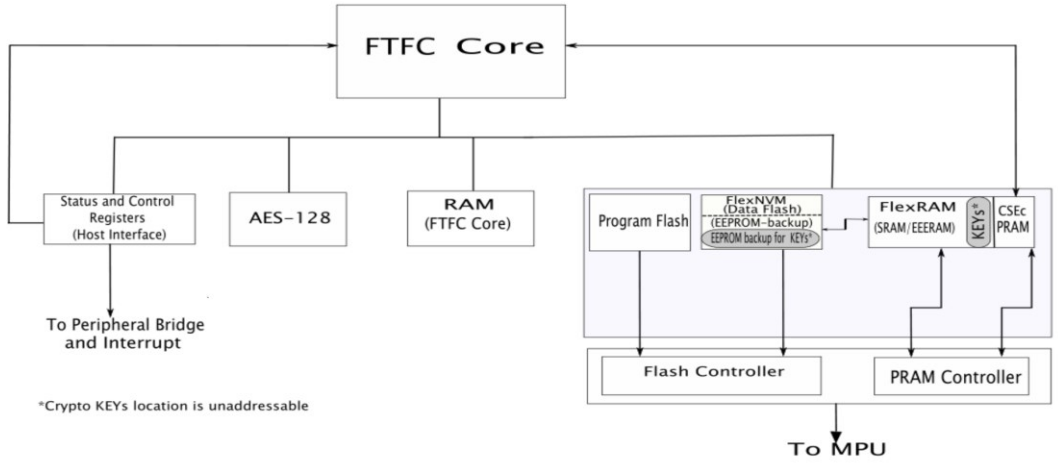
- 《S32DS使用Tips--S32DS for Power V1.2 链接文件和启动过程详解》；
- 《S32K系列MCU使用Tips之SDK软件架构和使用详解》；
- 《S32DS使用Tips--SDK使用常见问题(FAQ)答疑》；
- 《S32DS IDE使用Tips--应用工程调试常见问题(FAQ)答疑》；
- 《 S32DS 使用Tips之S32DS for Power不同版本之间的GNU工具链差异与外设寄存器位域访问问题总结》；
- 《 S32DS使用Tips之S32DS for Power v1.1应用工程升级到v1.2重新编译运行程序跑飞问题解决》；
- 《S32DS 使用tips--S32DS for ARM v1.3工程到S32DS for ARM V2.0迁移升级方法和注意事项》；
- 《 S32DS 使用 tips--工程属性配置(编译选项和C编译器、汇编器及链接器设置)》；
- 《 S32DS使用Tips--如何编译生成和调用静态库》；
- 《S32DS使用Tips--如何通过创建新的编译目标(Build Target)在同一个S32DS工程中同时编译静态库和应用程序》；
- 《 S32DS使用Tips--如何配置和使能Attach功能定位软件程序bug和完成bootloader与应用程序工程的联合调试》；
- 《 CodeWarrior与S32DS IDE使用 Tips之如何在应用工程中保留定义但未使用的全局常量、变量(用于参数标定)》；
- 《 S32DS 使用 tips--使用Flash from file下载S19或elf文件》；
- 《S32DS for ARM v2018.R1安装IAR Eclipse插件调用IAR工具链开发S32K系列MCU应用程序详解》；
- 《浅谈嵌入式MCU软件开发之条件断点的设置与使用详解(以S32DS IDE + U-Multink debugger为例介绍)》；
- 《S32DS IDE使用Tips--应用程序开发实战实用技巧总结与详解(工欲善其事必先利其器)》；

- 《S32DS使用Tips--SDK使用常见问题(FAQ)答疑》；
- 《S32K SDK使用详解之S32 SDK软件编程思想详解》；
- 《S32K SDK使用详解之S32 SDK软件架构详解》；
- 《S32K SDK使用详解之Keil MDK开发S32K系列MCU应用程序(使用Processor Expert配置SDK)》；
- 《S32K SDK使用详解之GHS Multi(Eclipse插件)开发S32K1xx系列MCU应用程序(使用PE配置SDK)》

5. S32K系列MCU应用指南介绍

- ✓ CSEc 硬件加密模块使用详解
- ✓ EEPROM模块使用详解
- ✓ FlexIO模块使用详解
- ✓ RTC模块使用详解
- ✓ 存储器ECC功能使用详解
- ✓ 芯片锁死(lockup)复位原因分析与恢复方法详解

CSEc 硬件加密模块使用详解

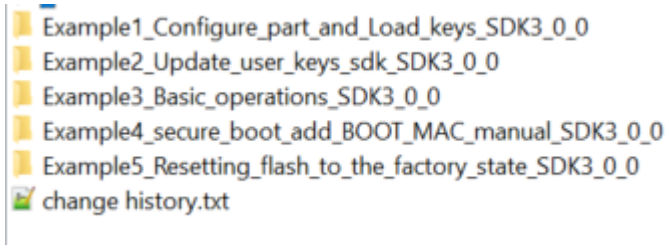


上图中的各个模块功能如下：

- FTFC Core：FTFC 模块的核心，包含硬件核心和 FTFC 的软件算法。
- Host Interface：系统与 FTFC 模块的通信接口。
- AES-128：AES-128 为一个硬件子模块，提供 AES-128 加密、解密算法和 CMAC 生成、验证算法。
- RAM(FTFC Core)：一块对用户不可见的 RAM 区域，用于提高 FTFC Core 的算法性能。
- Flash Controller：提供访问 Flash 的接口，由于 CSEc 的密钥是存储在 E-Flash（模拟 EEPROM）中，并且使用安全引导功能的时候，会对 Program Flash 中的数据进行 CMAC 验证，所以需要 Flash 控制器提供访问 Flash 的接口。
- PRAM(Parameter space Random Access Memory) Controller：应用程序与 CSEc 模块的通信接口，应用程序加载命令和数据至 FTFC 模块、FTFC 模块返回操作结果和数据至应用程序均通过 PRAM 接口实现。
- Program Flash：程序存储器，用于存储应用程序代码。
- FlexNVM：可配置的 Flash 存储器，可被配置为正常的 Flash 存储器用于存储数据或应用代码，也可被配置为模拟 EEPROM 存储器，作为 EEPROM 的数据保存区域。
- FlexRAM：可灵活配置的 RAM，可被配置为正常的 RAM，在程序运行中和 SRAM(System Random Access Memory)一样的被使用，也可被配置为模拟 EEPROM 存储器。由于 RAM 掉电数据即会被丢失，所以当 FlexRAM 被配置为模拟 EEPROM 时，需要 FlexNVM 配置出一块区域作为模拟 EEPROM 的数据保存区域与 FlexRAM 共同配合才能完成模拟 EEPROM 的功能，当写入和读出数据时，只需要按照操作 RAM 的方式，硬件会自动同步 FlexRAM 和 FlexNVM 中的模拟 EEPROM 数据。

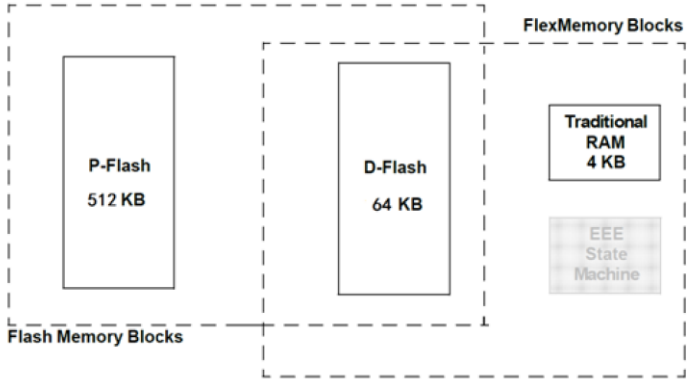
目录

- 配套软件demo工程

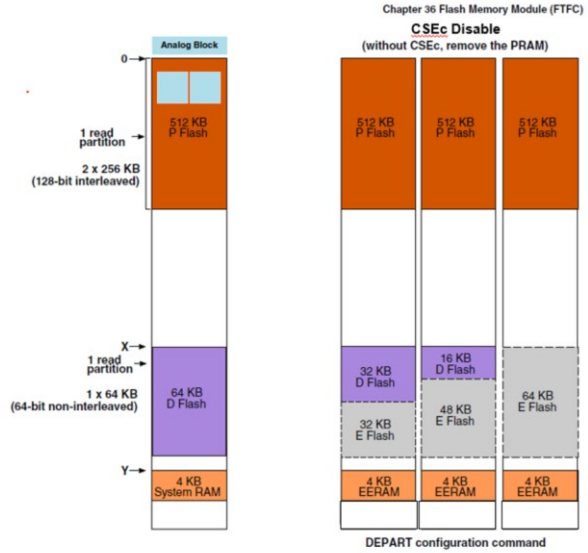


EEPROM模块使用详解

下图是S32K144的EEPROM和CSEc全部禁止即不使用EEPROM功能的内存分布框图。



下图是只使能S32K144的EEPROM的内存分布。



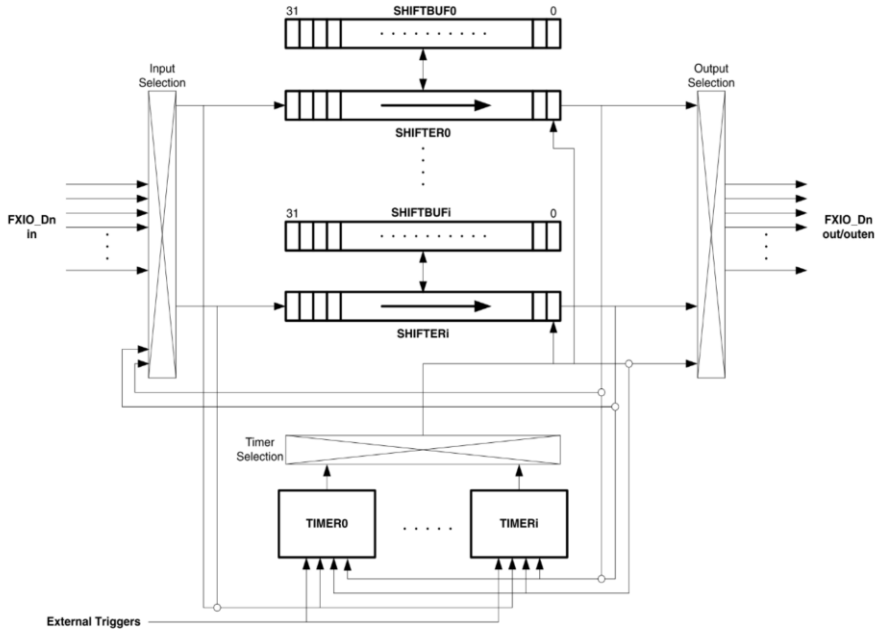
S32K1xx 系列 MCU 应用指南之 EEPROM 模块使用详解

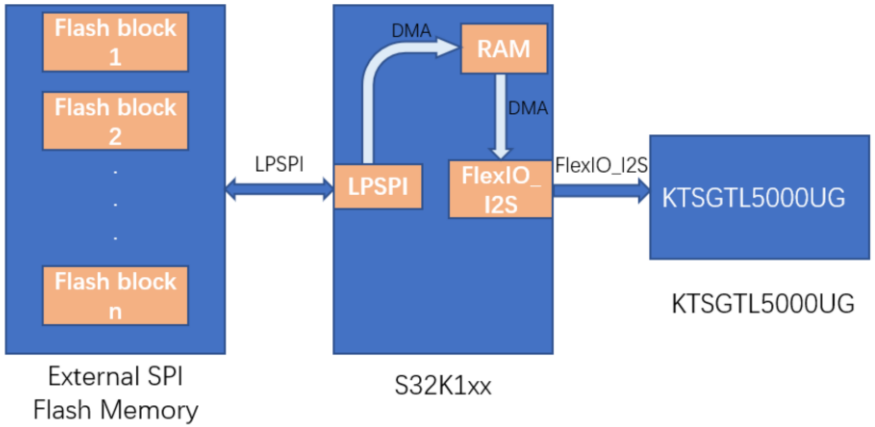
- 配套软件demo工程

S32K1xx_SDK_RTM3_0_FlexNVMPartition.zip
S32K1xx_SDK_RTM3_0_FlexNVMPEUsage.zip
S32K1xx_SDK_RTM3_0_FlexNVMQuickWrite.zip
S32K144_SDK_RTM3_0_EEPROMStartup.zip

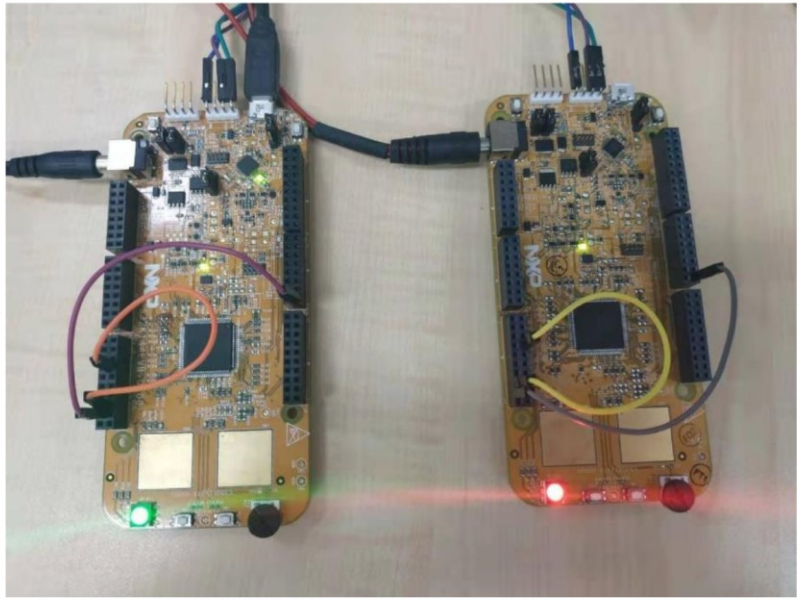
FlexIO模块使用详解

FlexIO 移位器和定时器的配置的模块框图如图所示：





内容提要



- 配套软件demo工程

S32K1xx_SDK_RTM3_0_SPIFlash_lpSPI_DMA_FlexIO_i2s.zip

S32K1xxAppUserGuide_FLEXLIO_LIN_SDK_RTM3P0.zip

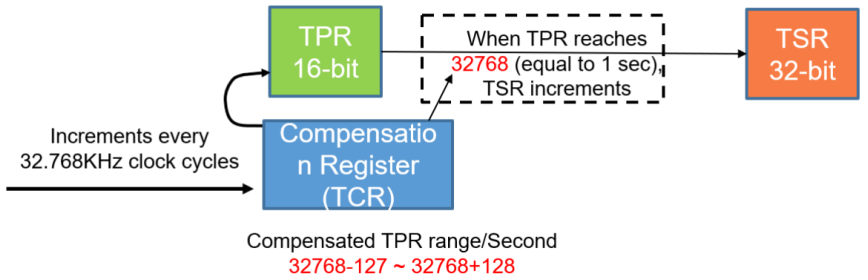
RTC模块使用详解

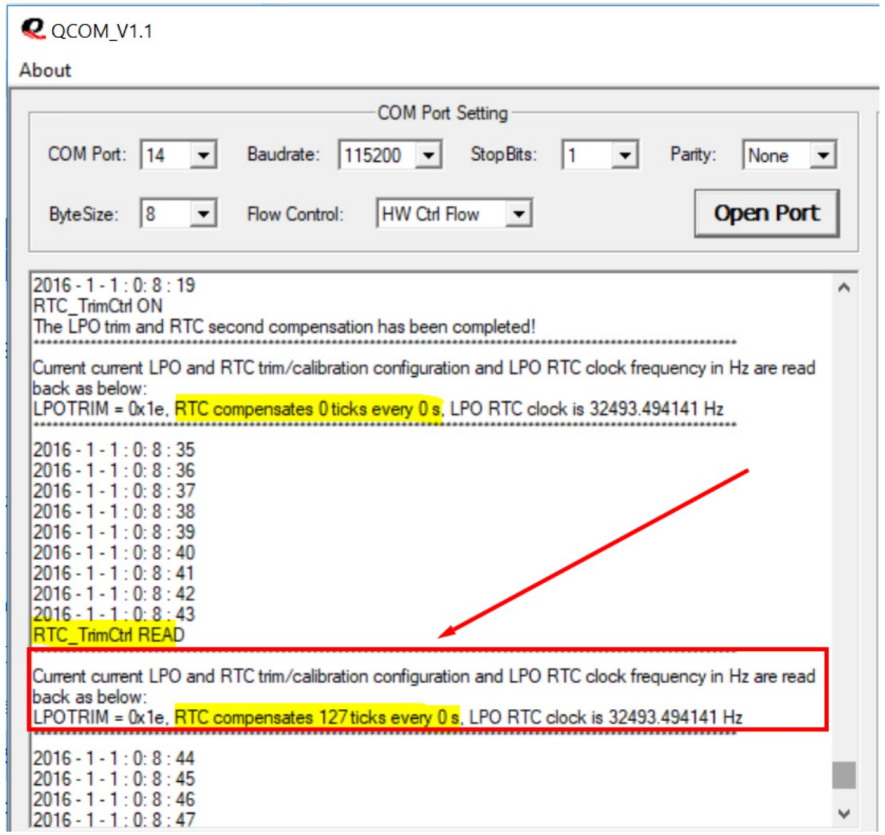
S32K1xx 系列 MCU 应用指南之 RTC 模块使用详解

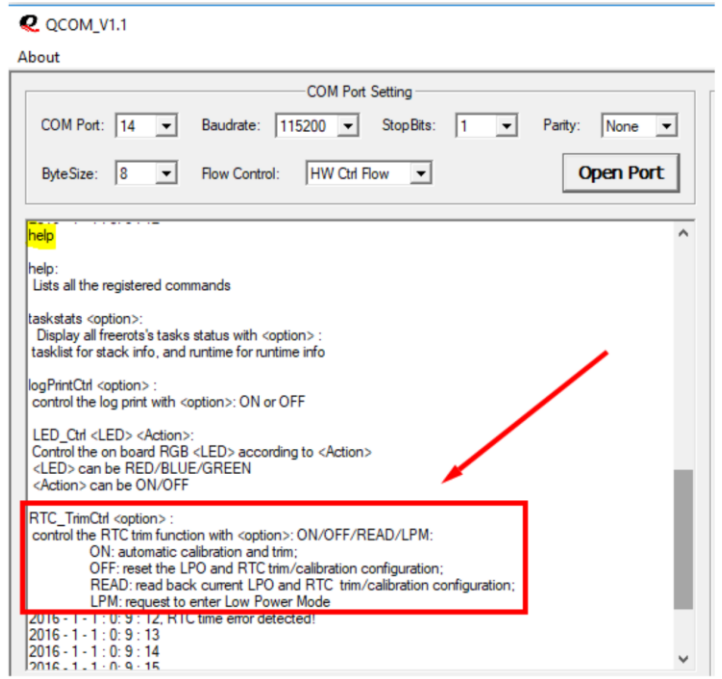
内容提要

2.1 RTC 时钟计时补偿工作原理

根据以上时间补偿寄存器 **TCR** 的介绍，可知 S32K1xx 系列 MCU 的 RTC 模块的时钟补偿工作原理如下：







- 配套软件demo工程

Example_S32K144_RTC_VLPS_S32DSR1_v1.zip
S32K1xxAppDev_RTC_SDK_RTM3P0.zip

存储器ECC功能使用详解

S32K1xx 系列 MCU 的存储器资源及其是否具有 ECC 功能及类型如下表：

S32K1xx 系列 MCU 的存储器资源	是否有 ECC 功能	ECC 类型
SRAM	有	32 + 7
FlexRAM(2/4KB)	无	-
MTB(1 KB, S32K116/8 only)	无	-
CPU Cache(S32K142/4/6/8 only)	无(有奇偶校验)	-
P-Flash	有	64 + 8
FlexNVM	有	64 + 8

2. S32K1xx 系列 MCU 的 SRAM ECC 实现

S32K1xx 系列 MCU SRAM 的 39-bit ECC 解码器功能框图如下：

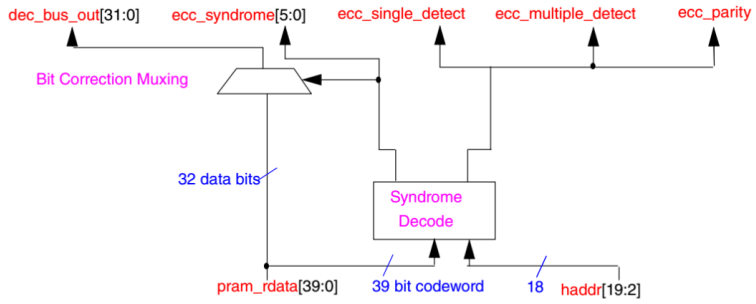
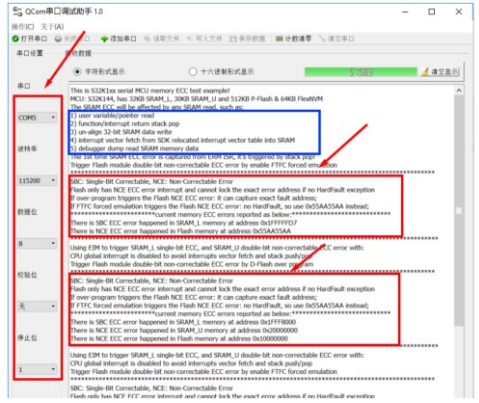


Figure 32-2. Block Diagram: 39-bit ECC Decode

内容提要

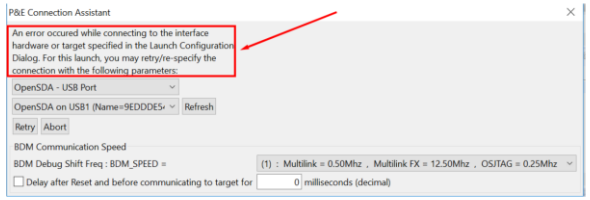
- 配套软件demo工程

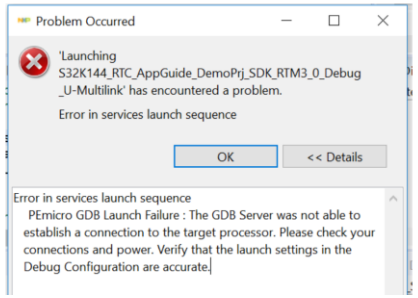
S32K1xxAppDev_MemoryECC_SDK_RTM3P0.zip



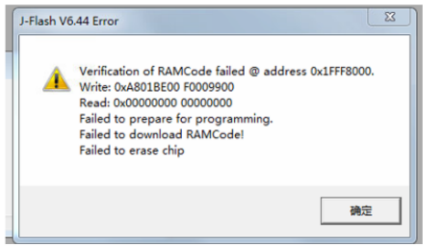
芯片锁死(lockup)复位原因分析与恢复方法详解

使用 PEMicro debugger(OpenSDA/U-Multilink)和 Flash Programmer(U-Cyclone)时会提示如下错误，调试无法建立与目标 MCU 的连接：

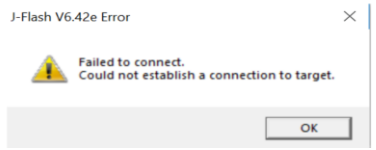


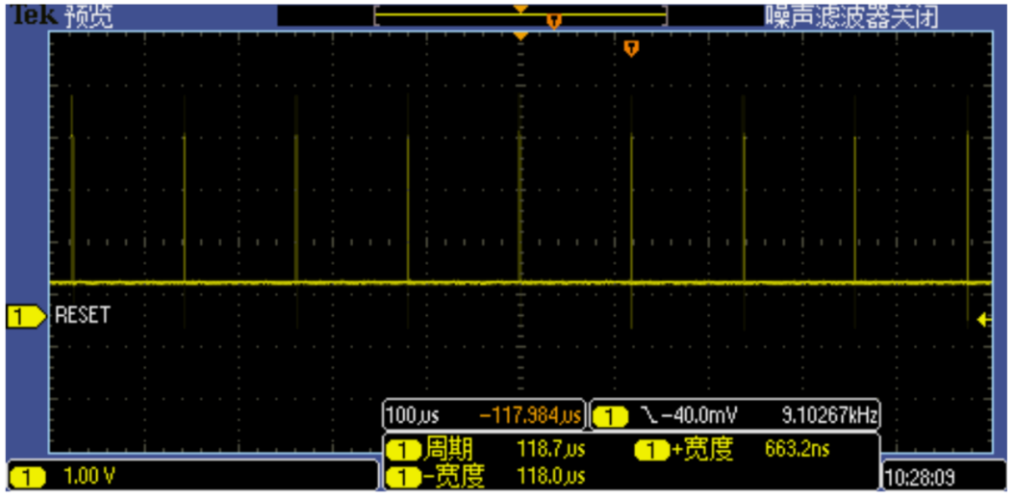


使用 J-LINK 调试或者下载程序时会弹出窗口提示“ Failed to RAMCode”错误



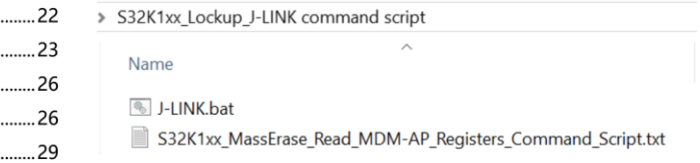
连接失败：





内容提要

- 配套软件demo工程

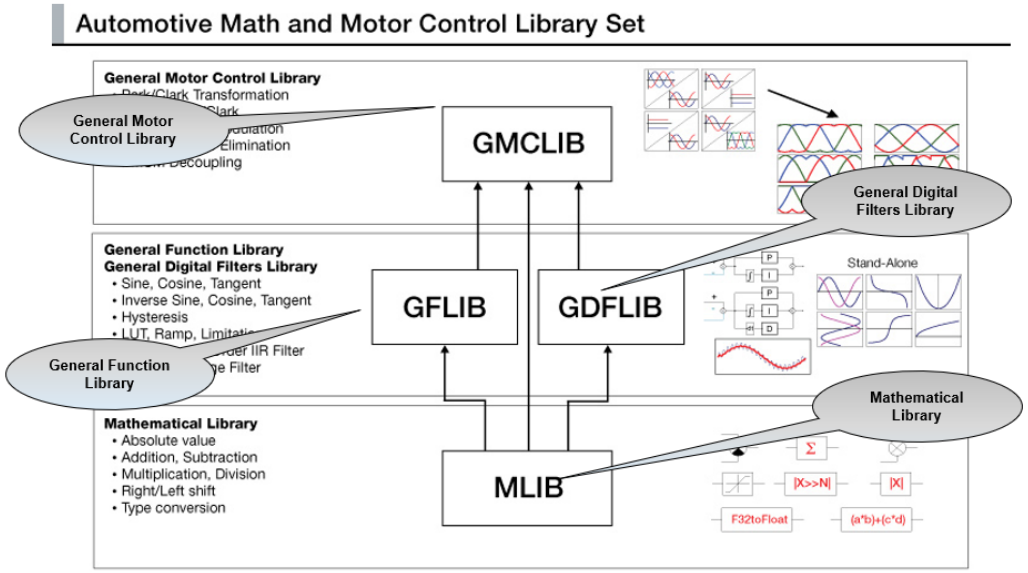


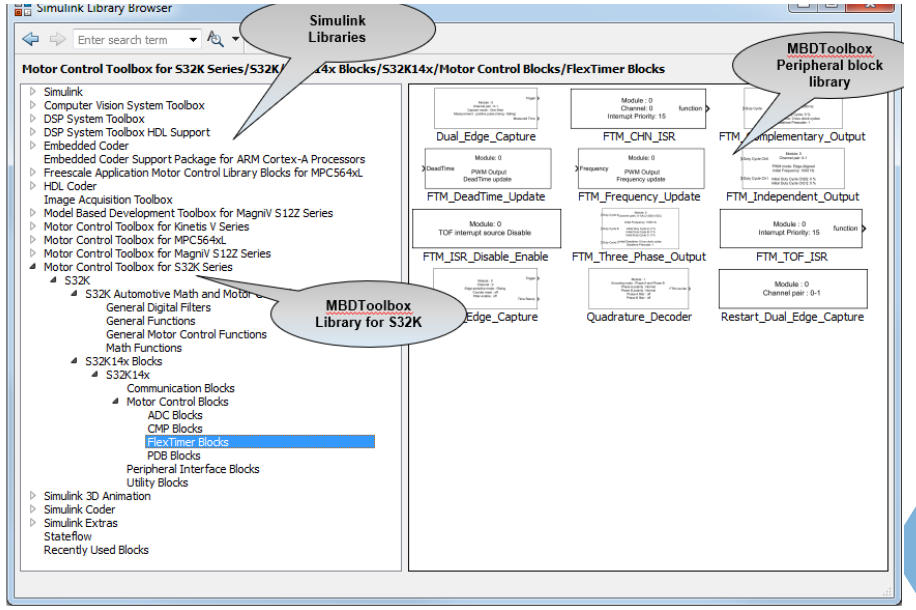
6. S32K系列MCU应用开发常见问题(FAQ)答疑

- ✓S32K系列MCU的电机控制库(AMMCLIB)介绍
- ✓基于Matlab Simulink的模型设计S32K1xx应用程序开发 (MBDT)
- ✓S32K系列MCU MBDT下载及使用学习资源链接
- ✓使用Keil MDK和IAR开发S32K系列MCU的应用程序
- ✓S32K系列MCU的CPU性能优化
- ✓S32K1xx系列MCU的跟踪调试(Trace)功能使用

S32K系列MCU的电机控制库(AMMCLIB)介绍

- 针对电机控制，NXP提供了专门高度优化的电极数学控制库-**AMMCLIB**，以方便用户设计和开发自己的电机控制算法，其主要包括以下函数库：
  - MLIB：数学运算库， e.g. 加减乘除和移位；
  - GFLIB：通用功能算法库， e.g. 三角函数运行；
  - GDFLIB：通用数字滤波算法库，e.g. FIR，IIR滤波；
  - GMLIB：通用电机控制库，e.g. Park, Clark变换，SVM；
- 在AMMCLIB的基础上，为了方便用户专注于应用层的电机控制软件开发，NXP还提供了S32K Simulink电机控制工具箱(Toolbox)
  - 支持模型开发和电机控制算法C代码自动生成
  - 集成点击控制相关的外设驱动配置代码自动生成，比如ADC模块、CMP模块、FlexTimer定时器和PDB模块等；

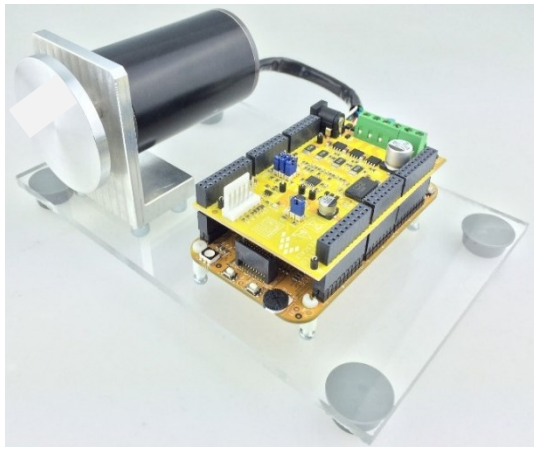




基于Matlab Simulink的模型设计的S32K1xx应用程序开发 (MBDT)

- 快速开发S32K系列MCU应用代码：
  - **支持硬件平台**：S32K14x EVB + 电机控制套件；
  - **支持工具链**：Matlab Simulink工具箱 + S32DS IDE + S32K SDK + AMMCLIB + RAppID Boot Loader utility自动代码生成和编译链接下载；
  - FreeMASTER GUI提供可视化实时PIL结果验证；

NXP S32K EVB Kit



Model Based DesingToolbox
with Simulink ™

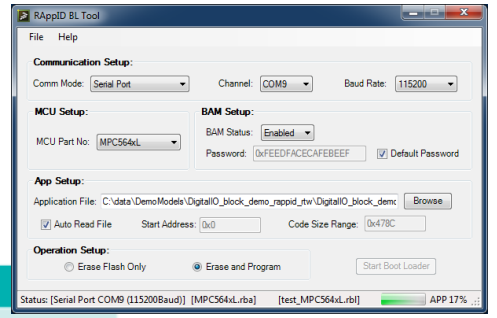
Model-based design
Driver configuration
Assignment to pins
Initialization setup

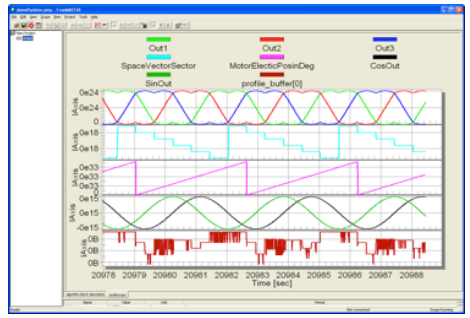
MATLAB SIMULINK

FREEMASTER

Signal Visualization and Data Acquisition Tool

RAppID Bootloader Utility





Summary of Devices Driver Blocks Supported

	Core and System				Communication					Motor Control					
MCUs	**GPIO**	**Timers**	**ISR**	**DMA**	**CAN**	**SPI**	**I²C**	**UART**	**FlexIO**	**ADC**	**PWM**	**PDB**	**CTU**	**GDU**	**AMMCLIB**
S32K1xx	x	x	x	x	x	x	x	x	x	x	x	x			x
MPC57xx (B/C/G/P)	x	x	x	x	x	x	x	x		x	x		x		x
MPC56xx (L/K)	x	x	x		x	x	x	x		x	x		x		x
S12 MagniV (ZVMx/ZVCx)	x	x	x		x	x	x	x		x	x			x	x
Kinetis V Series (1x/3x/4x/5x)	x	x	x		x	x	x	x		x	x	x			x

S32K系列MCU MBDT下载及使用学习资源链接

- **任意有效邮箱注册并登陆NXP账号后，访问以下链接即可免费下载**
  - https://www.nxp.com/support/developer-resources/run-time-software/automotive-software-and-tools/model-based-design-toolbox:MC_TOOLBOX?tab=Design_Tools_Tab#nogo

- **其他相关学习资料和文档(点击文档标题即可跳转下载阅读)**
  - Model-Based Design Toolbox for S32K1xx - QSG (REV 0)
  - Model-Based Design Toolbox for S32K1xx - Release Notes (REV 3.0.0
  - Model-Based Design Toolbox License Installation (REV 3.0.0)
  - Model-Based Design Toolbox Marketing Presentation (REV 0)
  - Model-Based Design Toolbox Enabling Motor Control Applications (REV 0)

- **在线培训视频和Community技术支持**
  - https://www.nxp.com/support/training-events/bldc-motor-control-with-mbdt-training-course:TIP-BLDC-MBDT-COURSE;
  - https://community.nxp.com/community/mbdt;

使用Keil MDK和IAR开发S32K系列MCU的应用程序

- **安装Keil 5， MDK 5.23及以上版本通过安装pack即可支持S32K系列MCU的软件开发，且可以使用最新的S32K SDK；**
  - 具体开发流程和经验分享请参考如下公众号文章(**点击文档标题即可跳转阅读**)：
  - 《**S32K SDK使用详解之Keil MDK开发S32K系列MCU应用程序(使用Processor Expert配置SDK)**》；
- **IAR 8.0及以上版本可以支持S32K系列MCU，且可以通过安装Eclipse Plugin的方式嵌套在S32DS IDE中使用，这样可以直接使用Processor Expert图形化配置和生成SDK代码，方便快捷；**
  - 具体开发流程和经验分享请参考如下公众号文章(**点击文档标题即可跳转阅读**)：
  - 《***S32DS for ARM v2018.R1安装IAR Eclipse插件调用IAR工具链开发S32K系列MCU应用程序详解***》

S32K SDK使用详解之Keil MDK开发S32K1xx系列MCU应用程序(使用Processor Expert配置SDK)

Original:Enwei Hu 汽车电子expert成长之路 4/17

内容提要

S32DS for ARM v2018.R1安装IAR Eclipse插件调用IAR工具链开发S32K系列MCU应用程序详解

Original:Enwei Hu 汽车电子expert成长之路 2018-11-08

内容提要

S32K系列MCU的CPU性能优化

S32K系列MCU的性能跟以下因素有关:

- **① CPU内核的指令和数据Cache是否使能(S32K14x系列only)**
  - S32K14x的CM4F内核有4KB的Cache，默认是关闭的;
  - 配置LMEM->LCCR寄存器可以使能和刷新(Flush) CPU cache;

```
// Flush and enable I cache and write buffer
LMEM->PCCCR = LMEM_PCCCR_ENCACHE_MASK | LMEM_PCCCR_INVW1_MASK
            | LMEM_PCCCR_INVW0_MASK   | LMEM_PCCCR_GO_MASK;
```

- **② Flash的工作频率是否配置到允许运行的最高频率**
  - Flash工作频率越高，CPU取指令越快
  - HSRUN @ max28MHz，RUN@max26.67MHz， VLPR@max1MHz

Following table summarizes the S32K14x/S32K11x maximum frequencies and example configurations for each internal clock

Clock	HSRUN	RUN		VLPR	Notes
	S32K14x only	S32K14x	S32K11x		
CORE_CLK SYS_CLK	112MHz	80MHz	48MHz	4MHz	Must be configured to be more than or equal to BUS_CLK.
BUS_CLK	56MHz	48MHz (40MHz when using PLL as system clock source)	48MHz	4MHz	Must be integer divide of the CORE_CLK.
FLASH_CLK	28MHz	26.67MHz	24MHz	1MHz	Must be integer divide of the CORE_CLK. The core clock to flash clock ratio is limited to a max value of 8.

- **③ Flash指令和数据预取功能是否使能**
  - Flash的buffer推测(speculation)和预取(prefetch)与CPU的Cache功能类似，只是前者可以优化的地址范围较小***32B***(**128-bit/16B** buffer size), 而后者的可以优化***4KB***的地址范围内的指令和数据读取;
  - 默认Flash指令和数据预取功能是使能的;
  - 运行**Flash Program**驱动时，需要关闭CPU Cache和Flash预取功能;

OCMDRn	Reset Value	On-Chip Memory Type
OCMDR0	0xDC089000	Program Flash
OCMDR1	0xCA08B000	Data Flash
OCMDR2	0xC304D000	EEERAM

- **④ 具体使用的编译器和优化配置，使用-Ospeed最大优化运行速度/性能**
  - Speed vs. Size二者不可兼得;
  - 无优化时，专业的Keil和IAR编译器编译结果明显优于S32DS使用的GCC;
  - S32DS提供的GCC优化选项十分粗略，具体的优化详解并不清楚，所以建议客户慎用。

```
    /* Disable cache to ensure that all flash operations will take effect instantly
     * this is device dependent and necessary for Flash program option*/
#ifdef S32K144_SERIES
    MSCM->OCMDR[0u] |= MSCM_OCMDR_OCM1(0xFu);
    MSCM->OCMDR[1u] |= MSCM_OCMDR_OCM1(0xFu);
    MSCM->OCMDR[2u] |= MSCM_OCMDR_OCM1(0xFu);
#endif /* S32K144_SERIES */
```

- **⑤将频繁访问的数据(.bss段、.data段和堆栈(heap & stack))放置在SRAM_U区(新建工程默认配置)**
  - 这样程序运行时，代码(code bus)和数据(system bus)可以并行访问，为真正的**哈弗架构**;

- **⑥使用硬件FPU可以将CPU浮点数计算性能提高1个数量级(S32K14x系列only)**
  - 更多关于S32K14x的FPU使用，请参考公众号文章：《*S32K14x系列MCU使用Tips之硬件FPU特性介绍和使用详解*》
  - ***新建工程时，默认配置未使用硬件FPU;***

```
speed_floatTest:
00001b95:   movs    r2, #0
00001b97:   vldr    s0, [pc, #100]  ; 0x1bfc <.text_6>
00001b9b:   vldr    s1, [pc, #100]  ; 0x1c00 <.text_7>
00001b9f:   movs    r1, r0
 51           while(cntTemp){
00001ba0:   cmp     r1, #0
00001ba2:   beq.n   0x1bb8 <speed_floatTest+36>
 52             cntTemp--;
00001ba4:   subs    r1, r1, #1
 53             z = (x + y) * y;
00001ba6:   vadd.f32        s4, s0, s1
00001baa:   vmul.f32        s4, s4, s1
00001bae:   vmov.f32        s2, s4
 54             k = z;
00001bb2:   vmov.f32        s3, s2
00001bb6:   b.n     0x1ba0 <speed_floatTest+12>
            }
```

S32K1xx 系列MCU的跟踪调试(Trace)功能概述

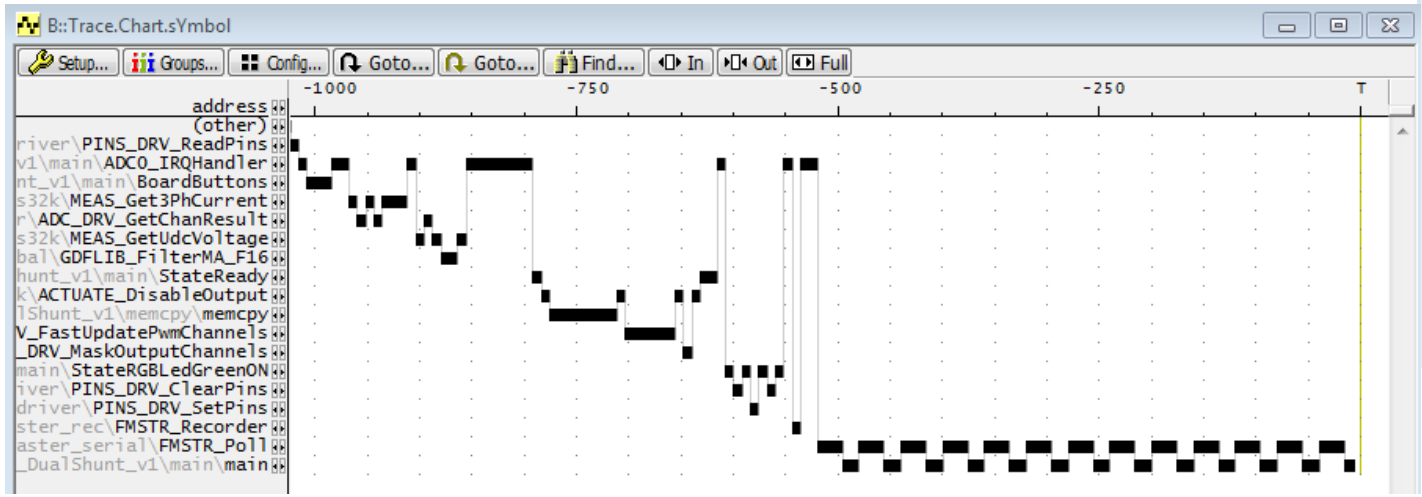
- **S32K116/S32K118**
  - Cortex M0+ core, OnChip trace, 1K SRAM as trace memory(**MTB**)
  - Program trace only, no data trace;
- **S32K142/S32K144/S32K146**
  - Cortex M4 core, no specific trace data outputs;
  - DataTrace available (max two variables because only 2 HW watchpoints(**DWT**))
  - Interrupt Trace (interrupts entry and exit)
  - PC Sampler (sample PC every 64 instructions, it's similar to program trace, but not 100% of software execution flow)
- **S32K148**
  - Cortex M4 core, 4 PIN Trace Data outputs + 1 PIN Trace Clock;
  - Both Program Trace and Data Trace are available

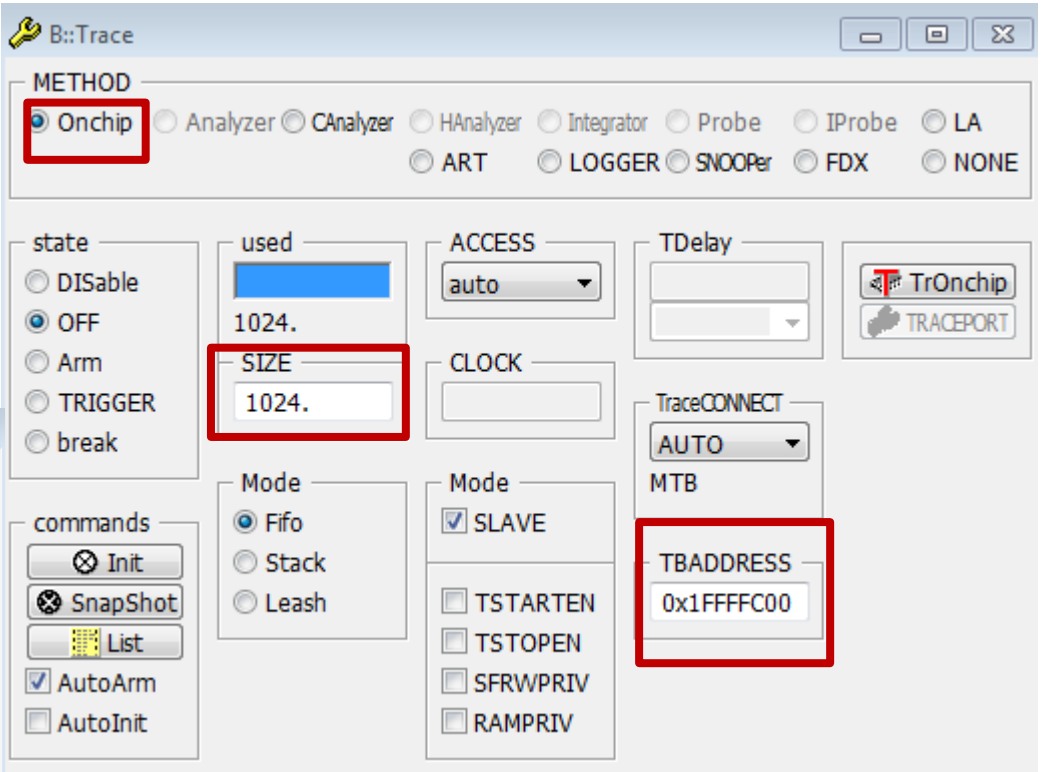
Table 56-11. Core Trace Connectivity

Chip	Feature
S32K116, S32K118	Trace data generated by the CM0+ core can be stored in MTB RAM.
S32K142, S32K144, S32K146	DWT and ITM trace data can traced out only through SWO interface.
S32K148	DWT, ETM and ITM trace data can be traced out through SWO or 4 pin parallel trace port.

Trace on S32K116/8 – Cortex M0+

- Lauterbach安装目录下有示例脚本，通常在” C:\T32\demo\arm\hardware\s32k\S32K116” 。
- S32K116的Trace脚本有以下关键点：
  - 1）选择CPU型号，sys.cpu S32K116；
    这样才能在后续命令中选择”OnChip” Trace Method; 这表示trace信息保存在芯片的SRAM，停下来之后在通过SWD接口传到电脑；
  - 2）Trace.TBADDRESS 0x1FFFFC00
    Trace.SIZE 0x400
    这两条命令是设定trace信息保存的地址，只能设定在这个范围，1K空间。
- Trace信息可以用trace.list或trace.chart来查看，如下图所示。





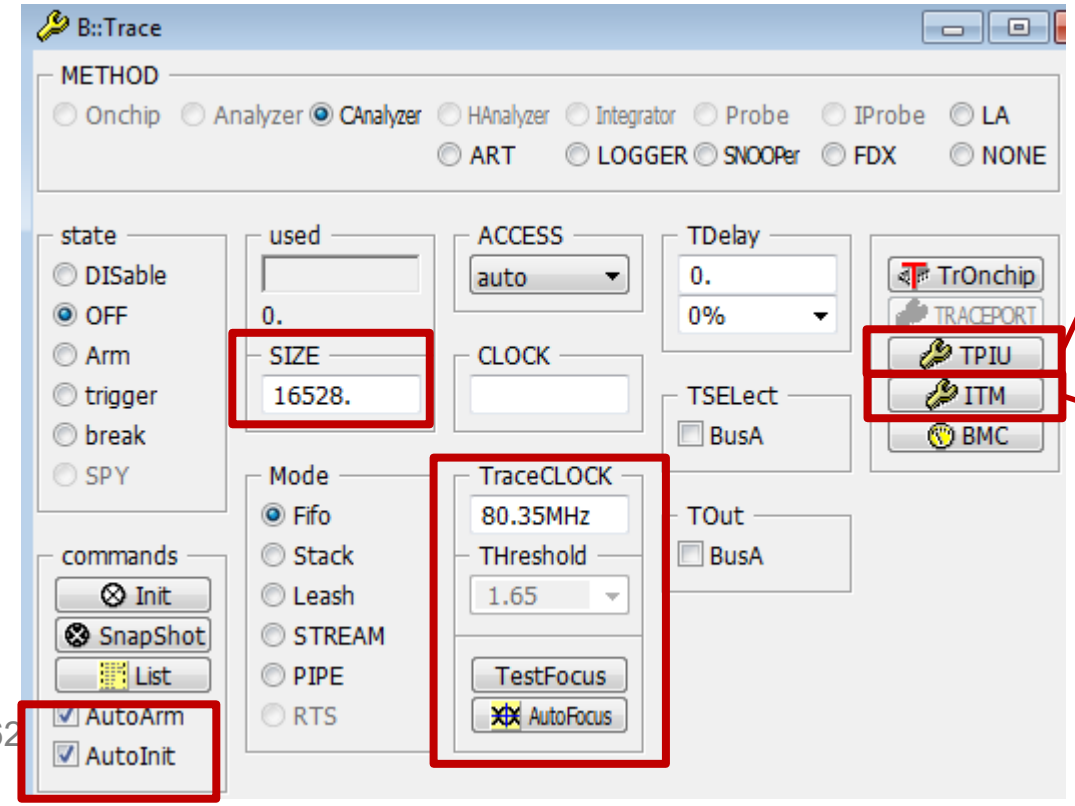
Trace on S32K142/4/6 – Cortex M4F

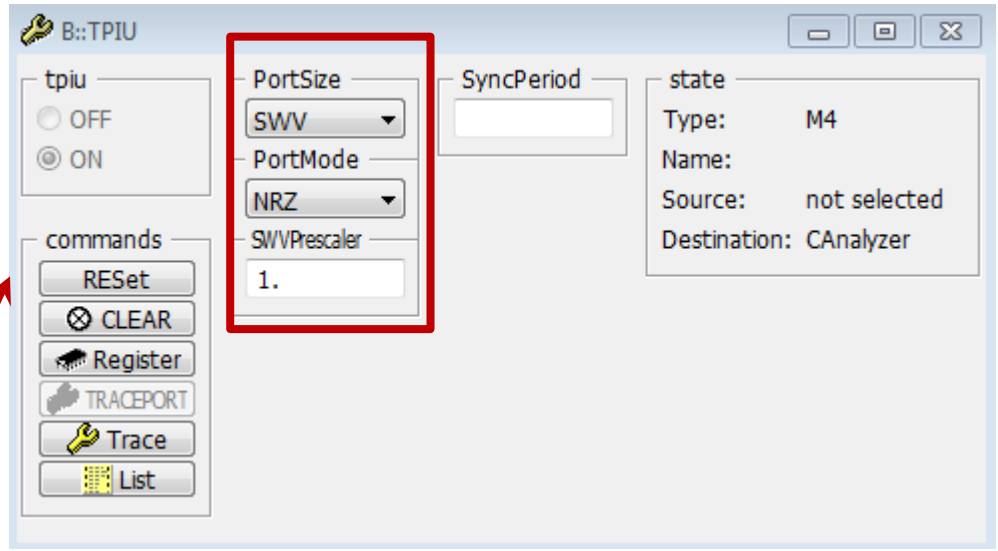
- Lauterbach软件安装目录下有示例脚本，通常在” C:\T32\demo\arm\hardware\s32k\S32K144” 。
- S32K144的Trace脚本有以下关键点：
  - 1) S32K144没有专门的trace信息输出关键，只需要调试接口SWD；因此不用配置PORT PCR；
  - 2) 以下命令使能trace功能。必须配置”TPIU.PortSize SWV”，因为没有专用的trace接口，只能通过SWD接口串行输出trace信息；
    - Trace.METHOD CAnalyzer
    - Trace.AutoInit ON
    - Trace.AutoArm ON
    - TPIU.PortSize SWV
    - ITM.DataTrace ON
    - ITM.ON
  - 3) 配置TPIU.SWVPrescaler
    - 运行现有脚本” DO ~~/demo/arm/etc/serial_wire_viewer/swv_autocalibration.cmm”
    - 或者直接设置 ”TPIU.SWVPrescaler 1”
  - 4) 设置断点，使其触发DataTrace；如下命令是在变量myVar被改写时触发DataTrace，这样可以记录下该变量的所有修改；
    - Var.Break.Set myVar /Write /TraceData
  - 5) 显示变量的变化曲线：
    - Trace.DRAW.Var %DEFault myVar

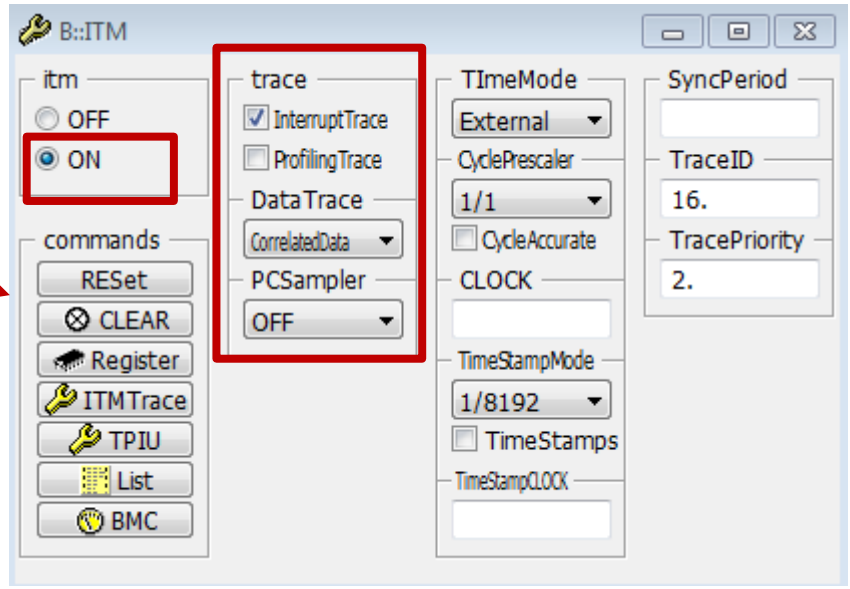
S32K142/4/6 Trace配置对话框

重要的配置选项在下列各对话框中用红色方框标出。

- AutoInit/AutoArm：程序运行时自动启动trace；
- TraceCLOCK：点击AutoFocus可自动获取合适的TraceCLOCK；
- SIZE: 该数值不宜过大，这里配置的FIFO模式，缓冲区一直保存最新一段时间的trace信息。
- TPIU：PortSize和SWVPrescaler按如图所示配置；
- ITM：根据需要进行配置，可用于DataTrace，InterruptTrace，PC Sampler；

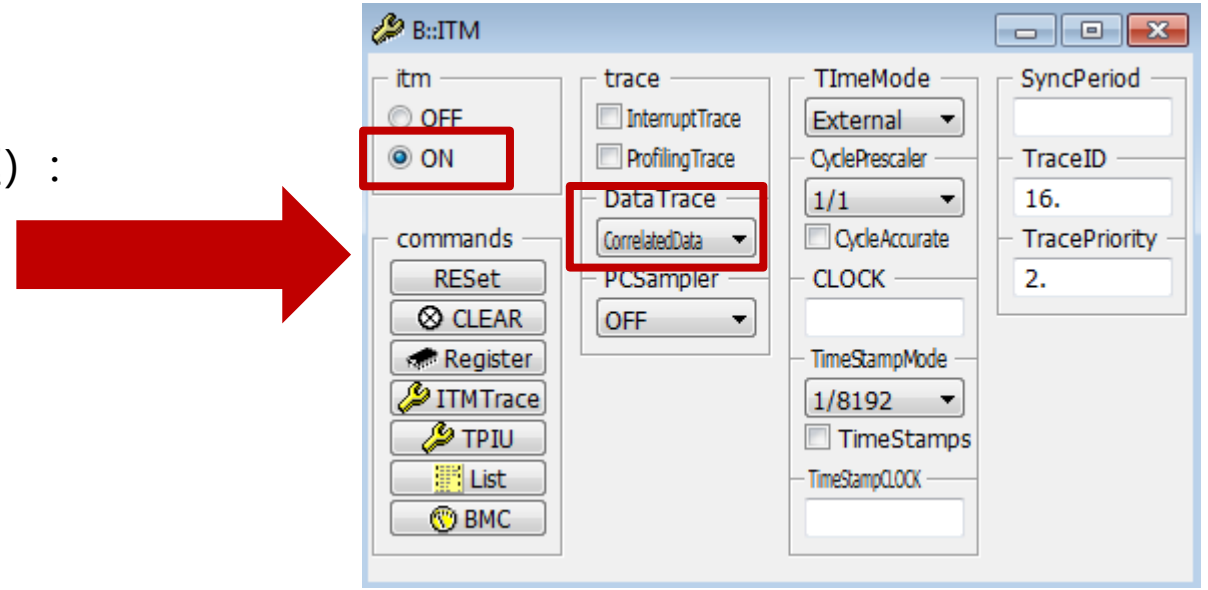




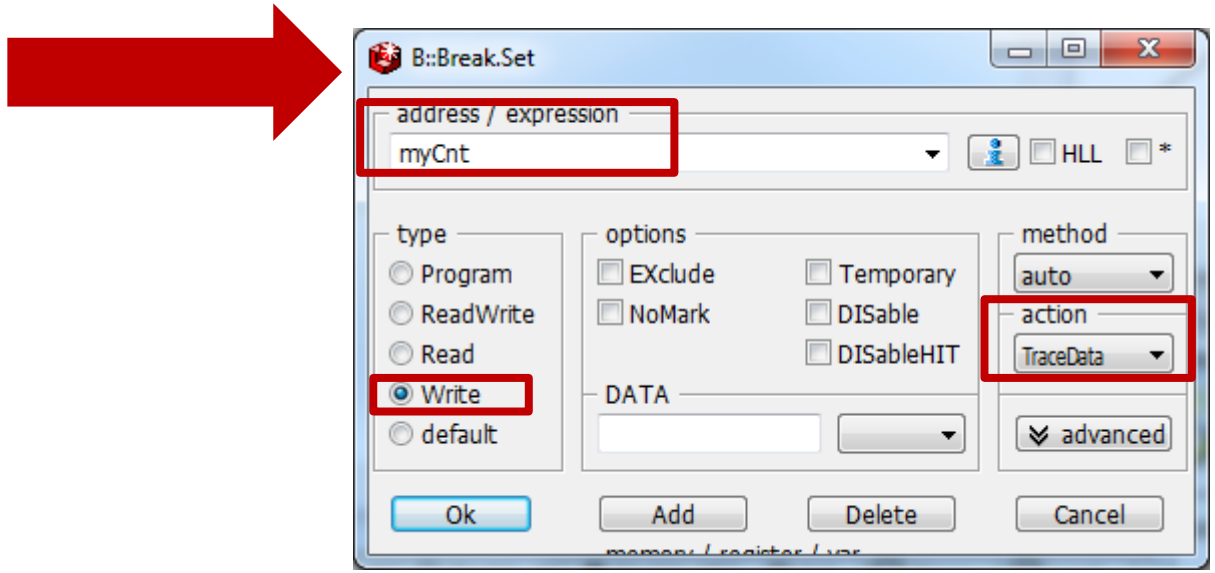


S32K142/4/6 – DataTrace

- TRACE和TPIU对话框按上页配置，无需改变。
- ITM对话框配置如图所示，或如下命令所示（等效于右图对话框）：
  - ITM.DataTrace.CorrelatedData
  - ITM.InterruptTrace.OFF
  - ITM.PcSampler OFF
  - ITM.ON



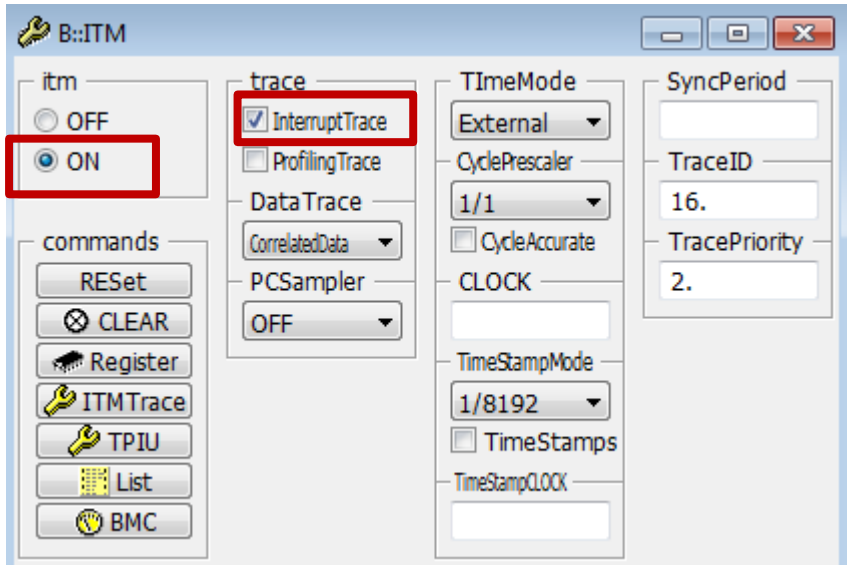
- 针对要监视的变量，设置断点：
  - Var.Break.Set myCnt /Write /TraceData
  - 等效于按右图对话框设置断点；



- 运行程序，暂停，查看变量曲线：
  - Trace.draw.var %Default myCnt

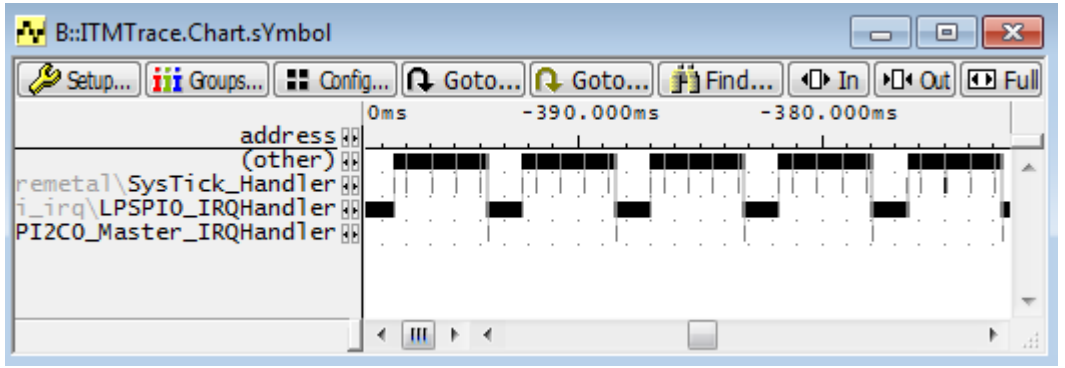
S32K142/4/6 – InterruptTrace

- TRACE和TPIU对话框按配置无变化。
- InterruptTrace可记录下所有的中断进入和退出；
- ITM对话框配置如图所示，或如下命令所示（等效于右图对话框）
  - ITM.DataTrace.CorrelatedData
  - ITM.InterruptTrace.ON
  - ITM.PcSampler OFF
  - ITM.ON



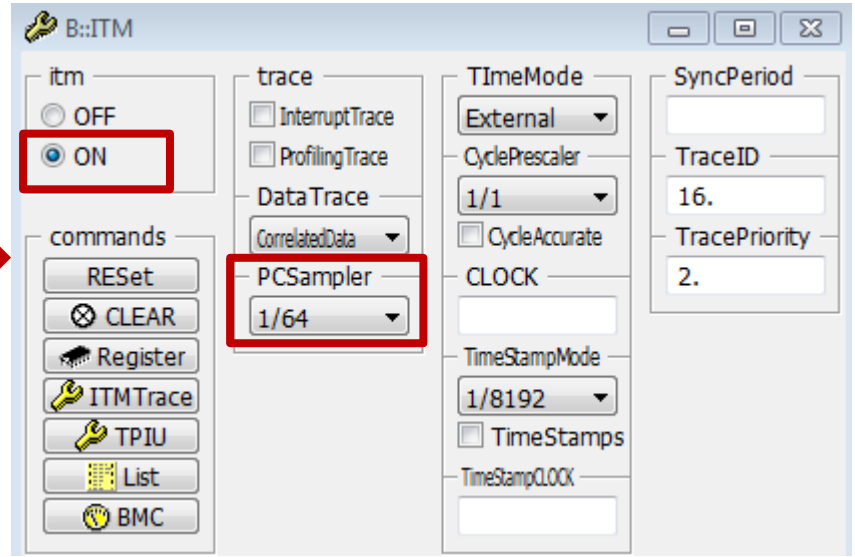
- 如果有用于DataTrace的断点，则需要将其删除或禁用，避免它影响InterruptTrace

- 运行程序，暂停，查看Trace信息：
  - Trace.Chart.Symbol



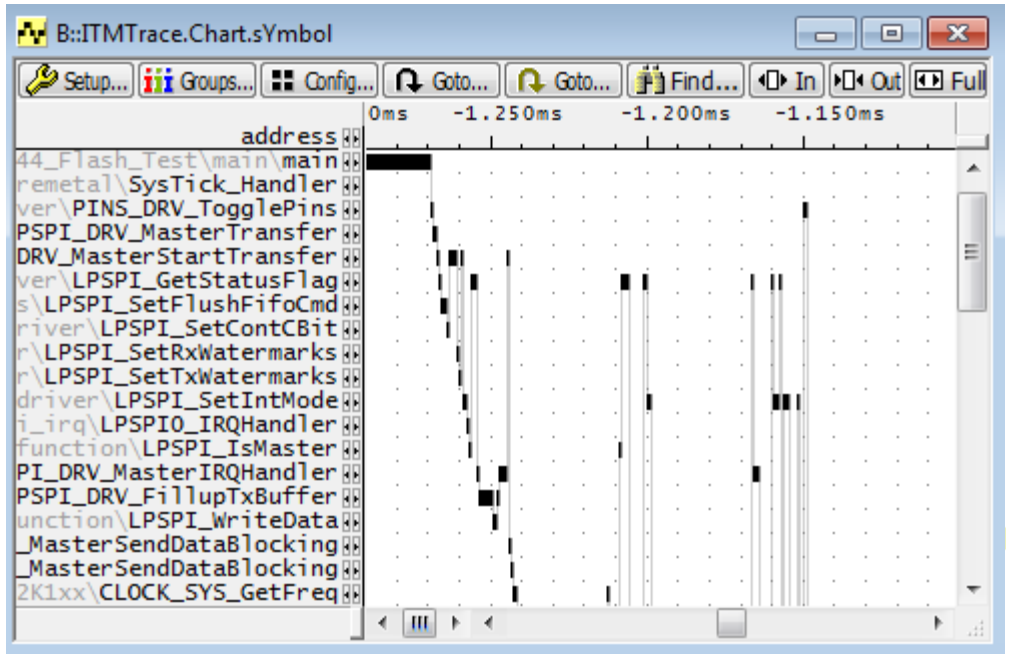
S32K142/4/6 – PC Sampler

- TRACE和TPIU对话框按配置无变化。
- PC Sampler最高以1/64的比率对PC采样，可间接获取程序运行流。
- ITM对话框配置如图所示，或如下命令所示（等效于右图对话框）
  - ITM.DataTrace.CorrelatedData
  - ITM.InterruptTrace.OFF
  - ITM.PcSampler 1/64
  - ITM.ON



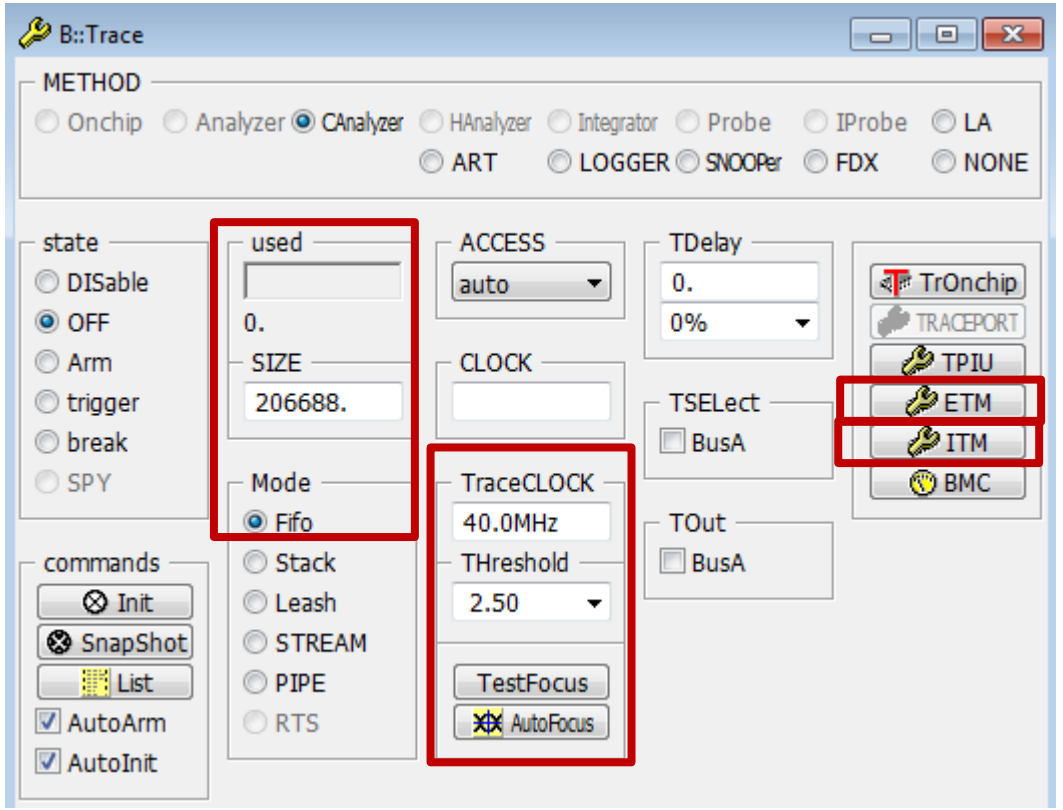
- 如果有用于DataTrace的断点，则需要将其删除或禁用，避免它影响PCSampler

- 运行程序，暂停，查看Trace信息：
  - Trace.Chart.Symbol



S32K148 – Cortex M4F trace with ETM

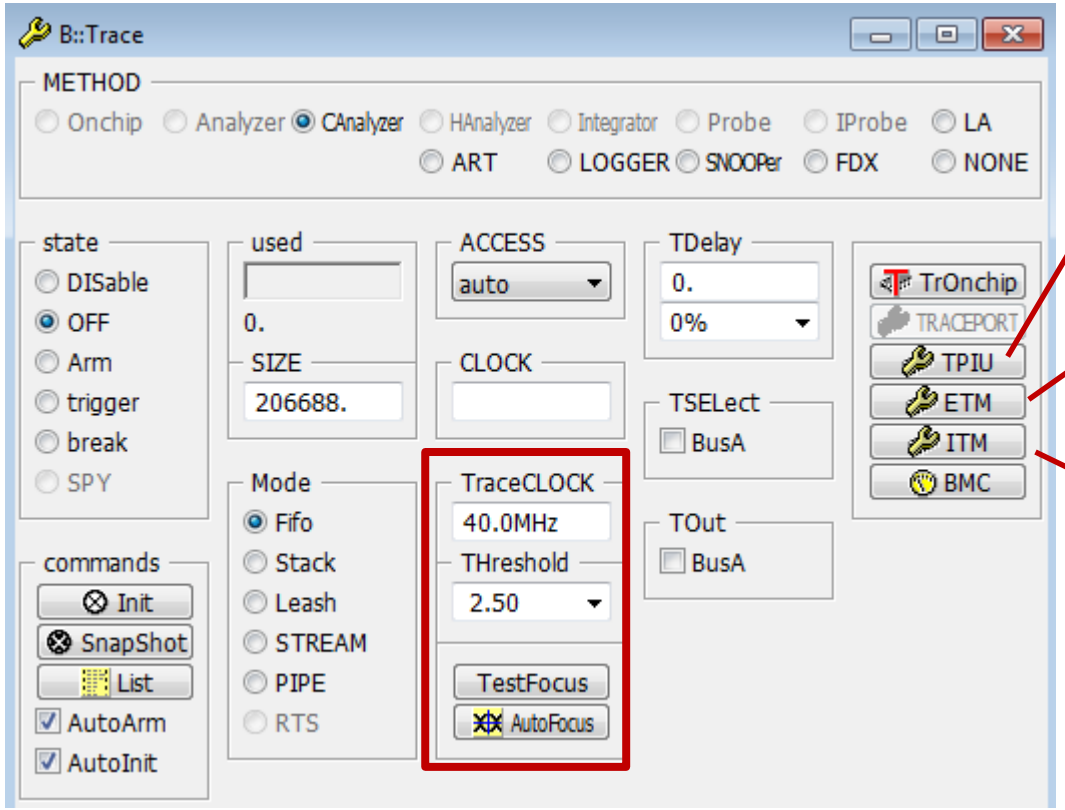
- Lauterbach安装目录下有示例脚本，通常在” C:\T32\demo\arm\hardware\s32k\S32K148” 。
- S32K148的Trace脚本有以下关键点：
  - 1) 需要配置Trace_D0~D3和Trace_CLK，共5个pin的PCR寄存器；此外该脚本还考虑了S32K148 EVB上的布线的因素，设置了这5个PCR寄存器的DSE（drive strength enable）；经过验证，必须有以下配置，否则trace通讯会出问题。
    - PTD16 DSE = LOW, TRACE_CLKOUT
    - PTD15 DSE = HIGH, TRACE_D3
    - PTE4 DSE = HIGH, TRACE_D1
  - 2) 必须合理配置Trace的时钟频率，TRACE模块使用core clock作为时钟源，可以配置core_clk为80MHz，trace clock分频得到40MHz；
    - SIM_CHIPCTL[TRACECLK_SEL] = 0;
      （复位后默认为该配置，trace模块时钟源为core_clk）
    - SIM_CLKDIV4 = 0x10000002; （trace clock divider enable, divide by 2）
  - 3) 上述配置可以由脚本实现，也可由软件代码来实现。
    如果是后者，那么trace功能必须在时钟和管脚配置的代码运行之后才能使用。
    此后可以在Trace => Configuration.. 窗口，
    点击”AutoFocus”来获取TraceCLOCK，结果应该是40MHz，且不报错；
  - 4) Program Trace和Data Trace可以分别运行，也可以同时运行。
    最好只运行其中一项，以确保trace信息的完整传输。
    - ITM用于Data Trace
    - ETM用于Program Trace



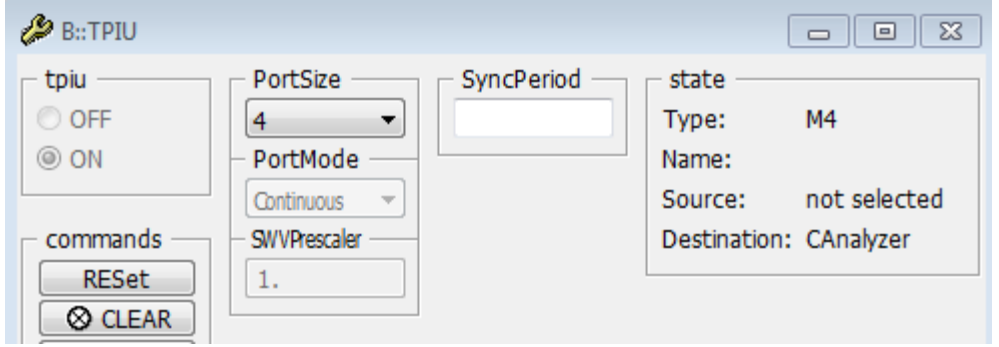
S32K148 Trace配置对话框

这些配置既可以在图形界面完成，也可以由脚本命令实现。

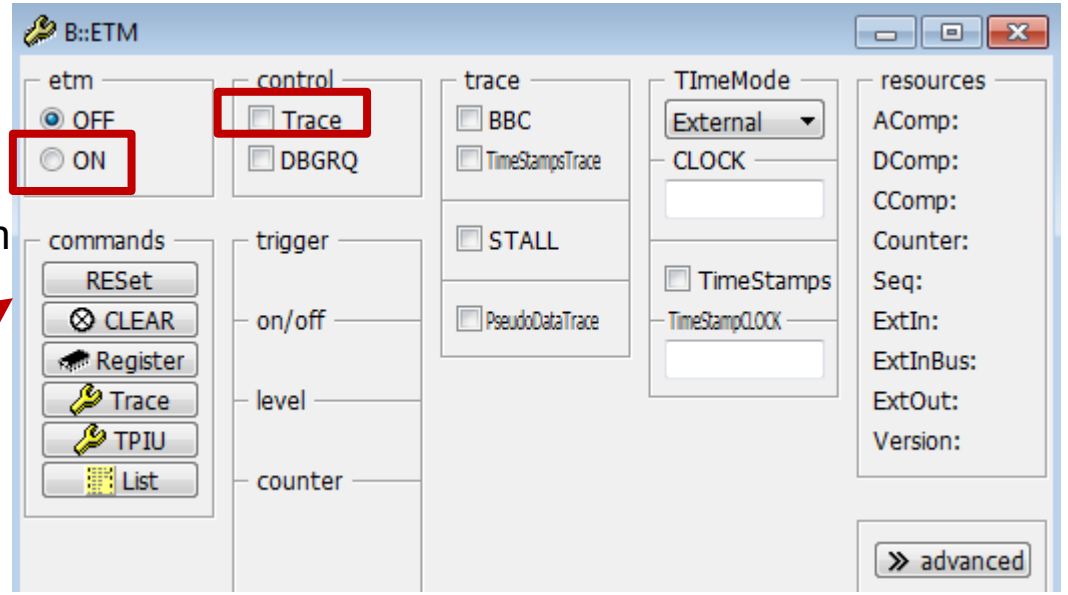
```
TPIU.PortSize 4
TPIU.PortMode Continuous
ITM.DataTrace CorrelatedData
ITM.ON
ETM.Trace ON
ETM.ON
```



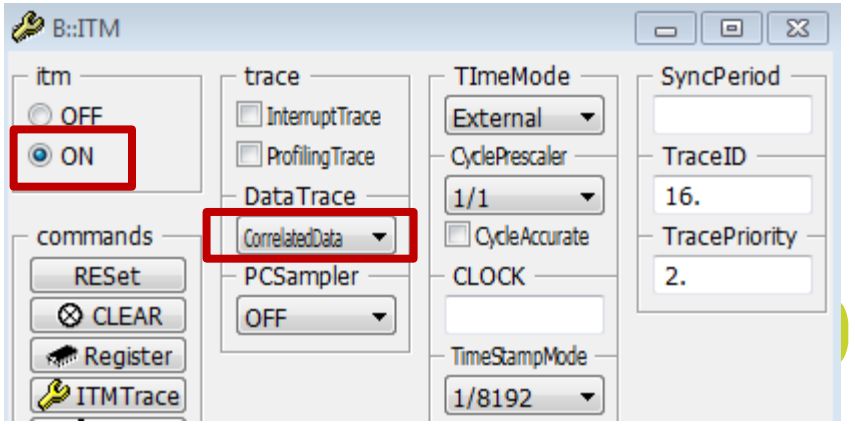
Trace信息传输通过4PIN



Program Trace



Data Trace



S32K148 – Program Trace

Trace基本配置，使能ETM，关闭 ITM

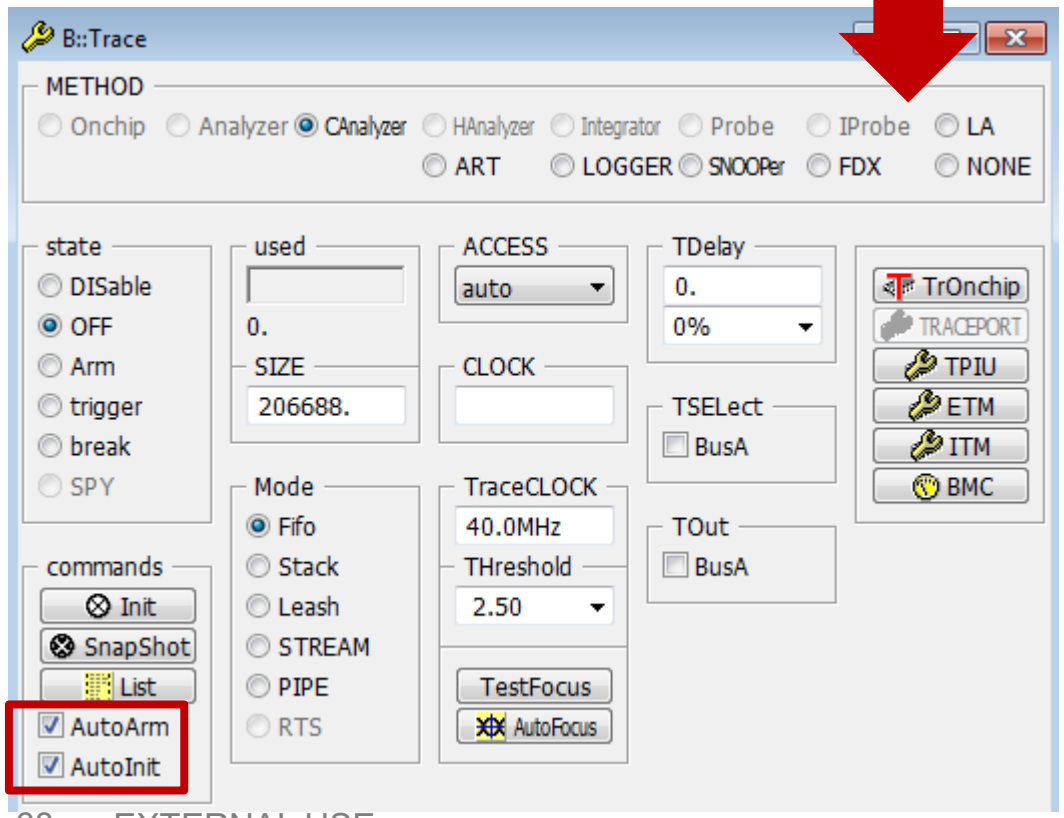
```
TPIU.PortSize 4
TPIU.PortMode Continuous
ITM.DataTrace CorrelatedData
ITM.OFF
ETM.Trace ON
ETM.ON
```

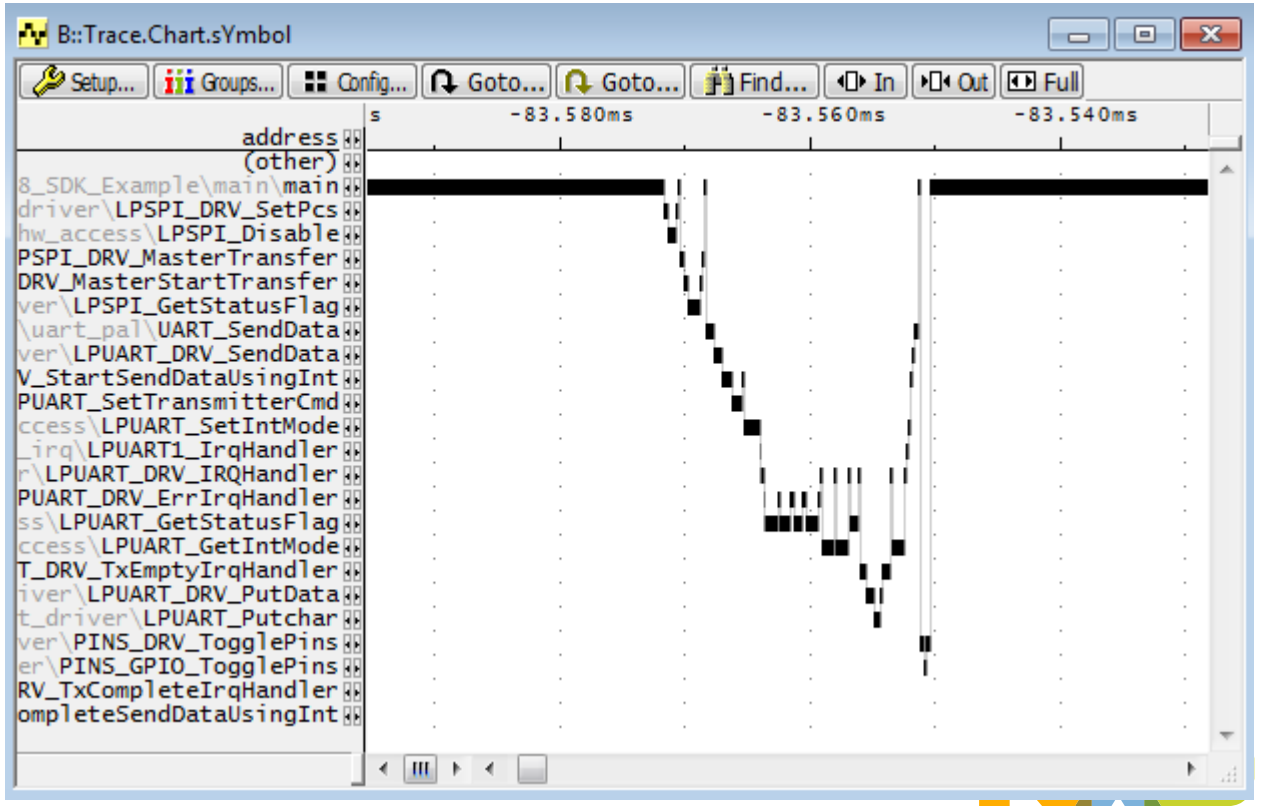
选择Trace方法，且使能自动初始化：

```
Trace.METHOD CAnalyzer
Trace.AutoInit ON
Trace.AutoArm ON
Trace.AutoFocus
```

运行程序，然后停止，查看trace信息：

```
Trace.List
Trace.Chart.Symbol
```





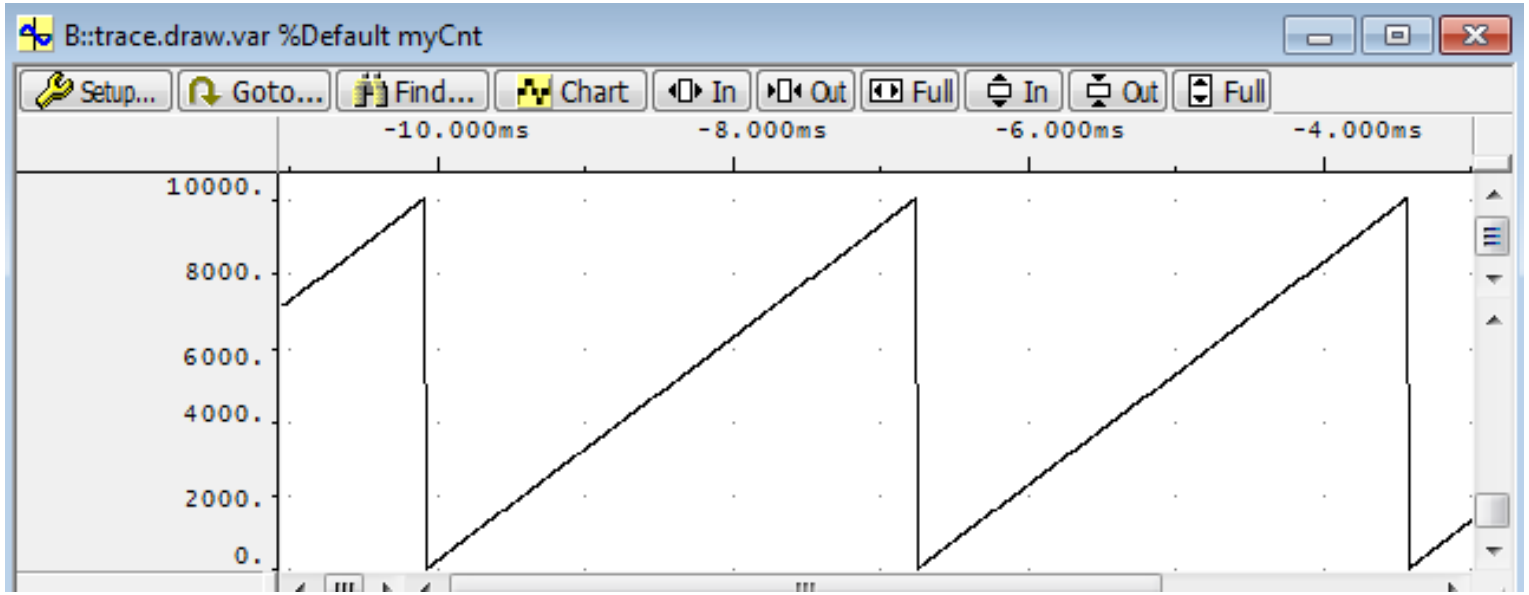
S32K148 – Data Trace

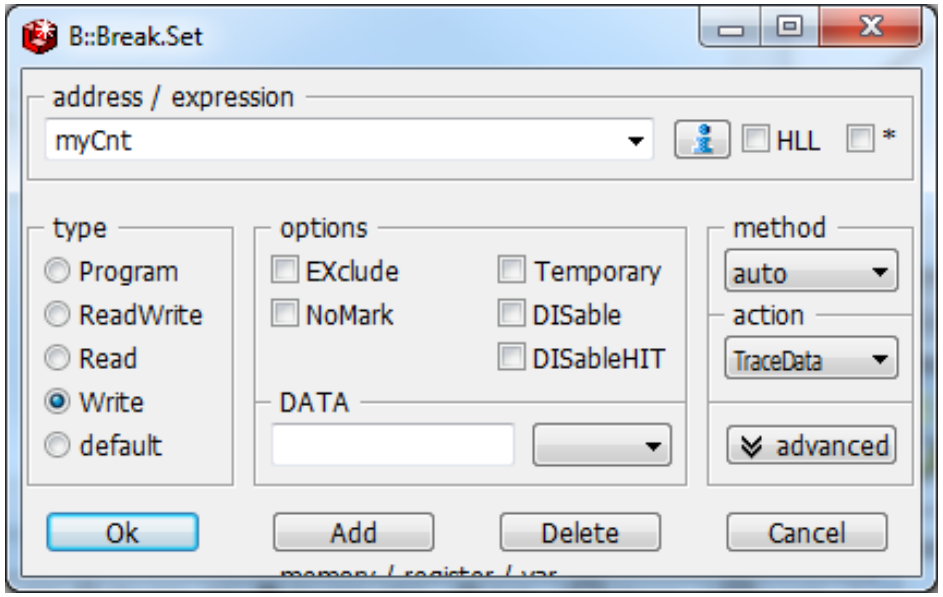
Trace基本配置，关闭ETM，使能 ITM
TPIU.PortSize 4
TPIU.PortMode Continuous
ITM.DataTrace CorrelatedData
ITM.ON
ETM.Trace ON
ETM.OFF

选择Trace方法，且使能自动初始化：
Trace.METHOD CAnalyzer
Trace.AutoInit ON
Trace.AutoArm ON
Trace.AutoFocus

在被监测变量上设置” 断点”
可使用如下命令，或下图对话框。
其含义是设置断点，当出现对该变量的写操作时，触发TraceData的动作，该命令是DataTrace的必要条件：
Var.Break.Set myCnt /Write /TraceData
最多支持2个变量因为只有2个硬件断点

运行程序，然后停止，查看trace信息：
Trace.draw.var %Default myCnt
可看到变量的变化曲线。





7. 总结与问答

- ✓ S32K系列MCU用户文档和技术支持
- ✓ 问答(Q&A)

S32K系列MCU用户文档和技术支持(NXP官网和微信公众号)

1. S32K官网产品网页
   - ✓ www.nxp.com/S32K;
   - ✓ 所有用户文档和软件demo工程均可以在此网页→**文档**一栏下载;
   - ✓ 参考手册(**RM**)、数据手册(**DS**)、勘误表(**Errata**)、应用笔记(**AN**)等等和评估板(**EVB**)及参考设计(**RDB**)链接;
2. NXP S32K 技术论坛(Community)
   - ✓ https://community.nxp.com/community/s32/s32k;
   - ✓ FAQ快速查找;
   - ✓ 与NXP原厂技术工程师互动QA，快速解决疑难问题;
   - ✓ 工程师技术交流与经验分享;
3. 微信公众号—**汽车电子expert成长之路(右侧二维码)**
   - ✓ S32DS IDE使用Tips
   - ✓ S32K1xx外设使用Tips
   - ✓ S32K SDK使用详解
   - ✓ 浅谈嵌入式MCU软件开发
   - ✓ 汽车ECU bootloader开发
   - ✓ 汽车ECU标定实现

SECURE CONNECTIONS
FOR A SMARTER WORLD