

A stylized green plant with several leaves is shown against a teal background. The plant's stem and leaves are rendered with a glowing, digital, wireframe-like texture, suggesting a connection between nature and technology.

英飞凌AURIX™ TC3xx车规级 单片机和典型模块技巧及运用 (CAN-FD、SMU、ADC)



**AURIX™ TC3xx 单片机系统架构介绍；
CAN-FD通信模块介绍
英飞凌主任工程师**



王迪

负责AURIX™单片机技术解答，拥有12年嵌入式电子经验，曾经在Atmel, Micronas, Sigmatel等半导体公司工作



Aurix TC3xx 单片机简介

Presented by: Di Wang



1: Aurix TC3xx 简介

- AURIX TC3xx 32位单片机适合于传统及现代汽车的安全关键的应用
 - 六核300MHz高性能Tricore架构
 - 传统应用：如制动、动力转向、发动机控制等。
 - 现代应用：混动、雷达、激光雷达、自动驾驶等。
- 具有：
 - 高性能、多样的连接性
 - 最高级别的信息与功能安全性
- 电源芯片：
 - 配合OPTIREG PMIC，比如TLE35584，可获得安全可靠的供电。
- 认证：
 - TC39x、TC37x、TC33x、TC32x、TC38x、TC36X产品系列已获莱茵萨尔（TÜV Saar）认证。
 - 开发过程和技术参数完全符合安全完整性等级ASIL D (ISO 26262)和SIL3 (IEC 61508)。

Flash	Package	TQFP-80	TQFP-100	T/LQFP-144	BGA-180	LQFP-176	BGA-216	LFBGA-292	LFBGA-516
6x 300 MHz	9xA Series 16 MB	Control & Actuate Sense & Compute						TC397XA 300 MHz	
6x 300 MHz	9x Series 16 MB							TC397X 300 MHz	TC399X 300 MHz
4x 300 MHz	8x Series 10 MB							TC387Q 300 MHz	TC389Q 300 MHz
3x 300 MHz	7xX Series 6 MB							TC377TX 300 MHz	
3x 300 MHz	7x Series 6 MB					TC375T 300 MHz		TC377T 300 MHz	
2x 300 MHz	6x Series 4 MB		TC364D 300 MHz		TC366D 300 MHz	TC365D 300 MHz		TC367D 300 MHz	
3x 300 MHz	5xA Series 4 MB				TC356TA 300 MHz			TC357TA 300 MHz	
2x 300 MHz	3xA Series 2 MB				TC336DA 300 MHz			TC337DA 300 MHz	
1x 200 MHz *	3x Series 2 MB	TC332L 200MHz *	TC333L 200MHz *	TC334L 200 MHz *	TC336L 200 MHz *			TC337L 200 MHz *	
1x 160 MHz	2x Series 1 MB	TC322L 160 MHz	TC323L 160 MHz	TC324L 160 MHz				TC327L 160 MHz	

2: Aurix 在汽车电子行业中的应用

- 动力:
 - 发动机控制单元 (ECU)
 - 变速箱控制单元
 - 电机控制单元
 - 混合动力系统控制单元
- 底盘:
 - 制动系统控制
 - 转向系统控制
 - 悬挂系统控制
- 车体:
 - 中控系统
 - 门窗控制
 - 安全气囊系统
 - ADAS高级驾驶辅助系统
 - 其他功能模块

3: Aurix 的历史

- <https://www.infineon.com/cms/en/product/promopages/historyoftricore/>
- 1999 AUDO (**A**utomotive **u**nified **p**rocessor)
 - TriCore (MCU RISC DSP)
- 2011 AUDO MAX
- 2014 AURIX TC2xx (**A**utomotive **R**ealtime **I**ntegrated **N**ext **G**eneration Architecture)
- 2018 AURIX TC3xx
- 2024 AURIX TC4xx
-

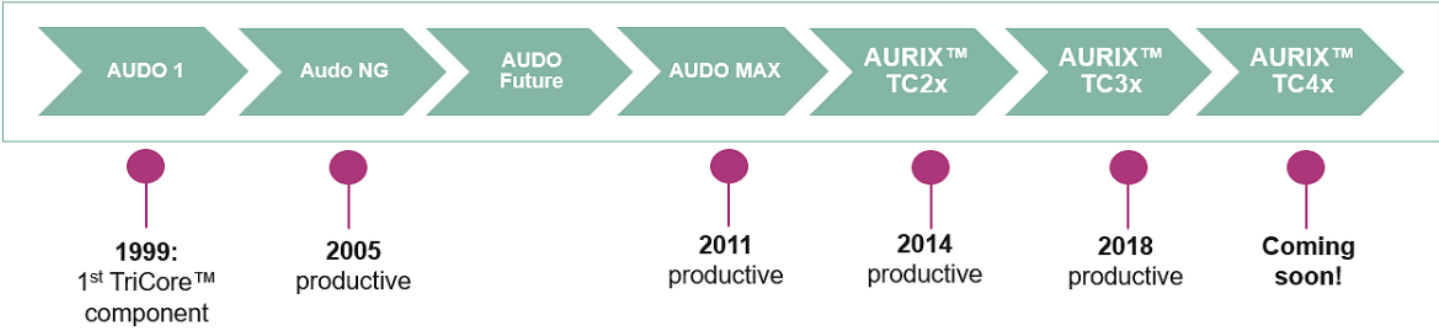
TriCore 累计销售数量 > 5 亿颗



The TriCore™ concept was born in 1999

TriCore™ integrates three functions: DSP, RISC & MCU

1st Target Market: Engine / Transmission Management Electrical Valve Train Starter/Alternator



第一目标市场：
发动机/变速器
电动阀门
启动器/发电机



TriCore™ enables the broadest family offering for automotive

4: Aurix 架构简介

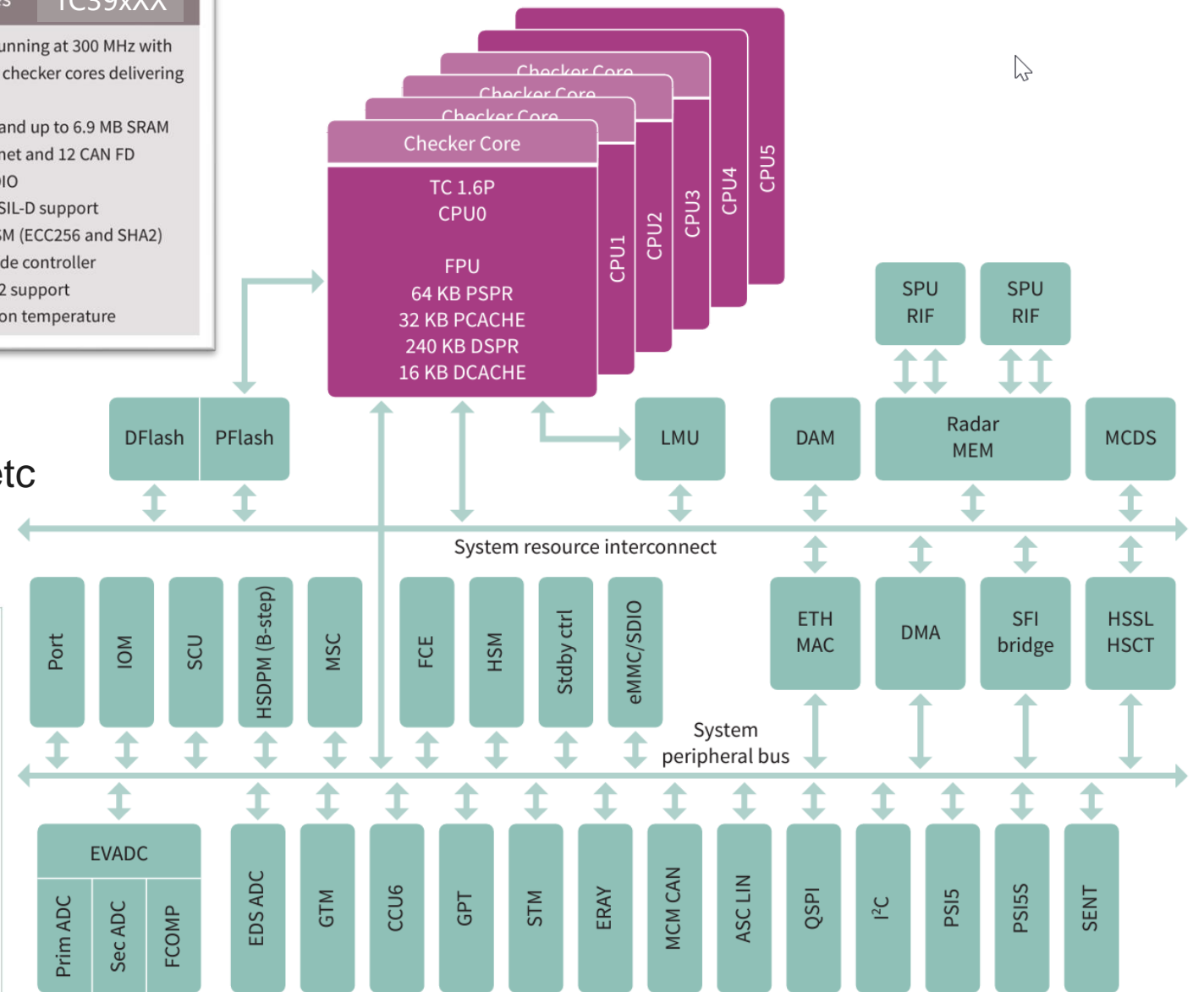
- TriCore: 高达6核, 300MHz主频
- SRAM: 高达6972KB
- Flash: 高达16MB
- 外设: GTM / CCU6 / EVADC / DSADC
- 通信及传感器接口: CAN/LIN/Flexray/Eth/etc
- 产品对比手册: [链接](#)

Key features TC39XX

- > 6 TriCore™ running at 300 MHz with 4 additional checker cores delivering 4000 DMIPS
- > 16 MB flash and up to 6.9 MB SRAM
- > 1 Gbit Ethernet and 12 CAN FD
- > 1x eMMC/SDIO
- > ISO 26262 ASIL-D support
- > eVita Full HSM (ECC256 and SHA2)
- > Standby mode controller
- > AUTOSAR 4.2 support
- > 165°C junction temperature

#All #Temperature Range #Series #Core Architecture #Feature packages #Flash size code #Frequency #Package type code

Brand		Device			Primary Option		Secondary Option					
SA	K	TC	3	7	5	T	P	-	96	F	300	W
Infinion product identifier	Temperature range	TriCore	Architecture	Series	Package	Core Architecture	Feature Package		Memory Size	Memory Type	Frequency	Package Type



5: Aurix 开发环境

– 免费的开发环境:

– 免费的ADS (Aurix Development Studio)

- 包含免费的编译器, 可以用来编译例子代码
- <https://www.infineon.com/cms/en/product/promopages/aurix-development-studio/>
- 用于评估, 并不预期用于量产

– 免费的Biface (早期的一个开发环境, 需自行设定编译器路径, 仍在一些项目中使用)

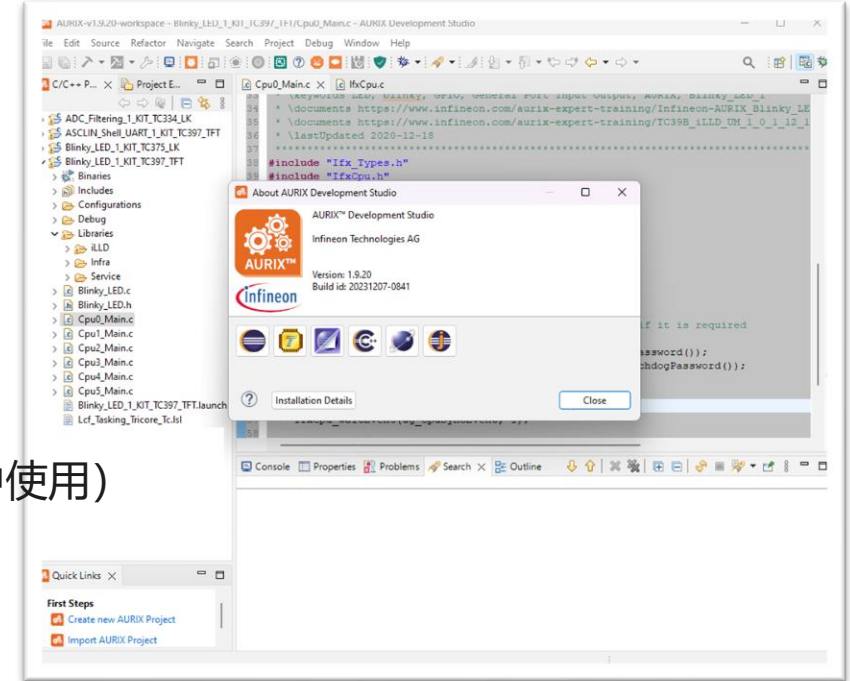
– 可以设置以下4种编译器的路径:

- Tasking
- Hightec
- Windriver
- Greenhills

1. 注意: Bifaces只是一个开发环境, 具体的代码仍然依赖iLLD和例子工程。当Bifaces搭配商用编译器时, 可以根据代码的级别来决定是否可以量产, 比如使用MCAL驱动和AUTOSAR架构, 还是使用iLLD底层驱动。对于iLLD来说, 并不预期用于量产。

– 商业版本的开发环境

- 客户可以通过联系上述编译器厂商, 了解商业版的开发环境。



6: Github仓库和例子代码

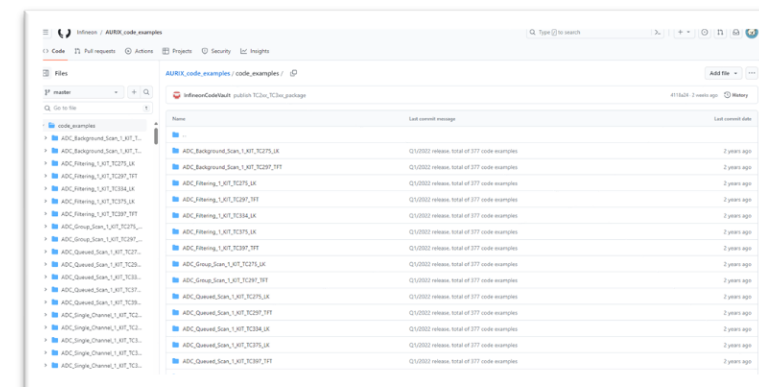
- 包含所有公开的Aurix例子代码
 - 链接: https://github.com/Infineon/AURIX_code_examples/tree/master

- 使用方法:

1. 可以从Github上下载, 在ADS上打开并编译, 或
2. 在ADS开发环境的File菜单中直接Import Project (导入工程)。
3. 在Cpu0_Main.c的顶部注释中, 有chm和pdf格式的《例子工程介绍》。
4. 《例子工程介绍》的内容与Aurix官方主页Training栏目中的TC3xx Quick Training和TC3xx Expert Training类同。
5. <https://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-microcontroller>

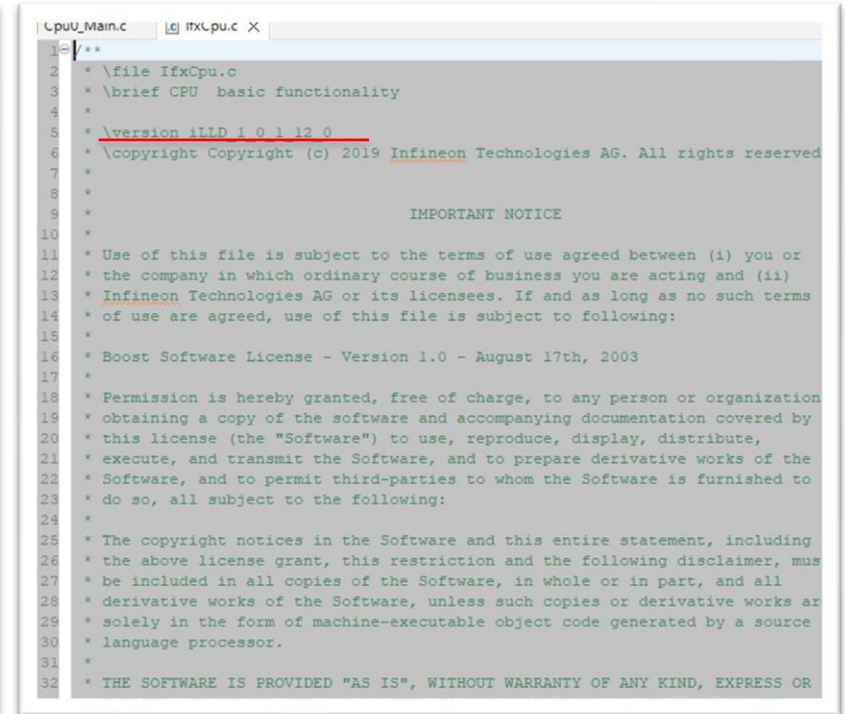
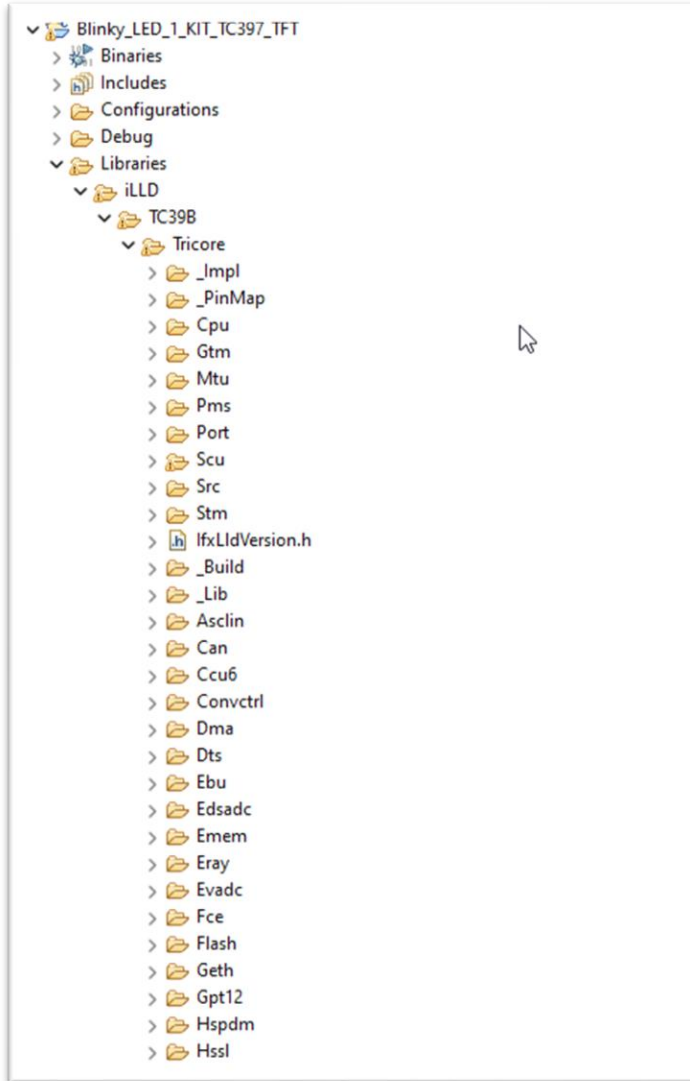
Training

- + AURIX™ TC2xx Quick Trainings
- + AURIX™ TC2xx Expert Trainings
- + AURIX™ TC3xx Quick Trainings
- + AURIX™ TC3xx Expert Trainings



7: 免费的底层驱动iLLD

- iLLD (底层设备驱动程序)
 - 底层驱动并不用于商业及量产
 - 所有的例子代码都是基于iLLD
 - 可以从iLLD开始尝试各个例子代码
 - 注意它的版本, 有时需要更新
 - 在myICP中获取不同版本的压缩包
 - 同时阅读不同版本的Release Note



8: 编译器

– Tasking

1. 与ADS开发环境集成的免费编译器 (无需设置)
2. 带有Tasking IDE的免费编译器(FREE TASKING VX-TOOLSET FOR TRICORE/AURIX)
 - <https://www.tasking.com/landing/Free-TASKING-TriCore>
3. 商业许可证的版本
 - <https://www.infineon.com/cms/en/tools/aurix-tools/Compilers/TASKING/>

– Hightec

1. 带有Hightec IDE的免费编译器 (HighTec Free TriCore™ Entry Tool Chain)
 - <https://free-entry-toolchain.hightec-rt.com/>
2. 商业许可证的版本
 - <https://www.infineon.com/cms/en/tools/aurix-tools/Compilers/Hightec/>

– Windriver

– Greenhills

9: 调试器

– USB Miniwiggler

1. 使用经济的入门产品
2. 集成在Triboard开发板中,或
3. 单独的具有DAP/JTAG接口的USB Miniwiggler
4. 配合免费的ADS开发环境使用, 或
5. 配合免费的UDE调试软件使用

– https://www.infineon.com/cms/en/product/evaluation-boards/kit_dap_miniwiggler_usb/

– <https://www.pls-mc.com/products/ude-starterkits/>

– <https://www.infineon.com/cms/en/tools/aurix-tools/free-tools/hightec/>

– PLS UAD2pro

- 商业版本的调试器
- 调试软件: 商用版UDE Software

– <https://www.pls-mc.com/products/universal-access-devices-uad-via-jtag-cjtag-dap-swd-once-lpd-dxcpl-ca>

– Lauterbach PowerDebug Module USB3.0

- 商业版本的调试器
- 调试软件: PowerView Software (TRACE32)
- <https://www.lauterbach.com/products/LA-3500>



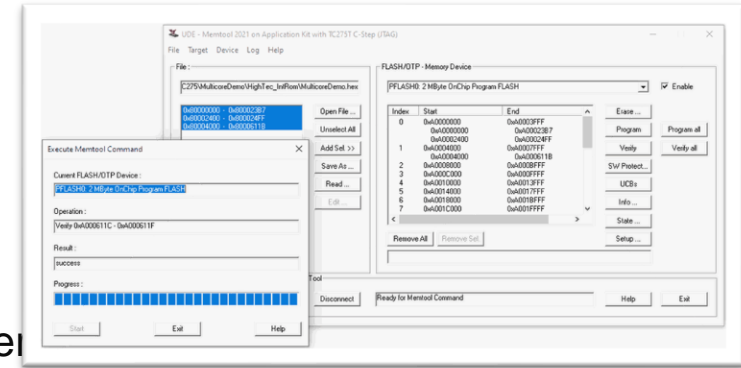
免费的UDE调试软件集成在AURIX™ Free Tool Chain 套装软件中, 由Hightec提供免费版本的编译器和IDE, 由PLS提供UDE。



10: 编程烧录工具

– 免费的英飞凌MemTool

- <https://www.infineon.com/cms/en/tools/aurix-tools/free-tools/infineon/>
- 可用于擦除、编程、验证和保护可选的闪存部分。
- 在Windows系统上通过DAS软件和UART接口连接调试器，比如Miniwiggler
- Miniwiggler使用DAP或JTAG接口连接到目标系统上。



– 商业化的PLS UDE MemTool

- <https://www.pls-mc.com/products/flash-programming-ude-memtool-flash-programmer-memory-device/>
- 具有高级功能，比如
 - 支持多个设备同时烧录（用于量产）
 - 对UCB（用户配置块）进行操作，包括HSM的部分
 - 可以使用更多的接口来连接目标系统，比如ASC、CAN、JTAG、RS232、K-Line.....
- 可联系PLS来获取进一步的技术支持

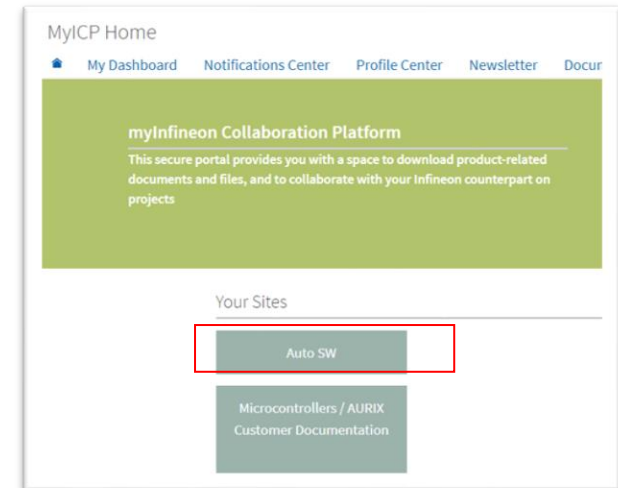
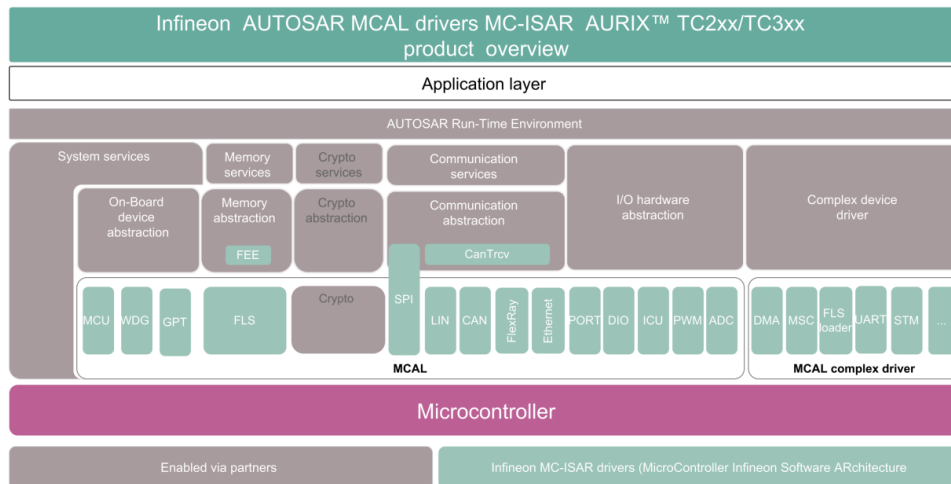
11: 信息安全和功能安全

- 注册myICP(我的英飞凌协作平台)
 - 获取更多TC3xx资料:
 - <https://www.infineon.com/cms/en/product/promopages/MyICP-platform-for-Microcontroller/>
- HSM
 - 硬件安全模块 (信息安全)
 - 需要专门申请权限以获取资料 (权限独立于TC3xx)
 - 在myICP中的路径: **AURIX TC3xx/10 AURIX TC3xx HSM**
- Safety Manual
 - 安全手册 (功能安全)
 - 可在myICP中下载 (权限与TC3xx相同)
 - 在myICP中的路径: **AURIX TC3xx/00 AURIX TC3xx General/02 Safety Documentation**

12: AUTOSAR架构

- MCAL:
 - MCU抽象层: 符合ISO26262标准, ASIL D等级的商业框架
 - 包含经过验证的驱动程序, 使用配置工具EB Tresos
 - 需专门购买, 或申请试用
 - 具有权限之后, 可以在myICP平台看到Auto SW的文件夹
 - 使用MCAL遇到的问题请联系我们的技术支持或合作伙伴
 - <https://www.infineon.com/cms/en/tools/aurix-embedded-sw/AUTOSAR/Infineon/>

TC3xx MCAL支持AUTOSAR V4.4.0标准, 提供符合ASIL D等级 (ISO 26262) 的驱动程序, 并符合SIL-2(IEC 61508)标准



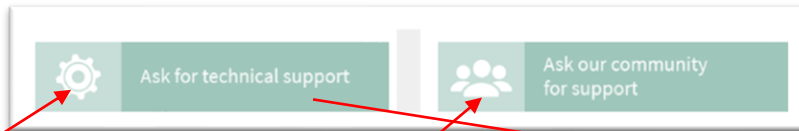
13: Aurix 官方网页

1. Aurix资料的类别有：产品汇总手册、产品简介、产品选择指南、User Manual、Datasheet、Application Notes、Errata等。
2. myICP上的资料比官网更全面，比如Tech Spec等。

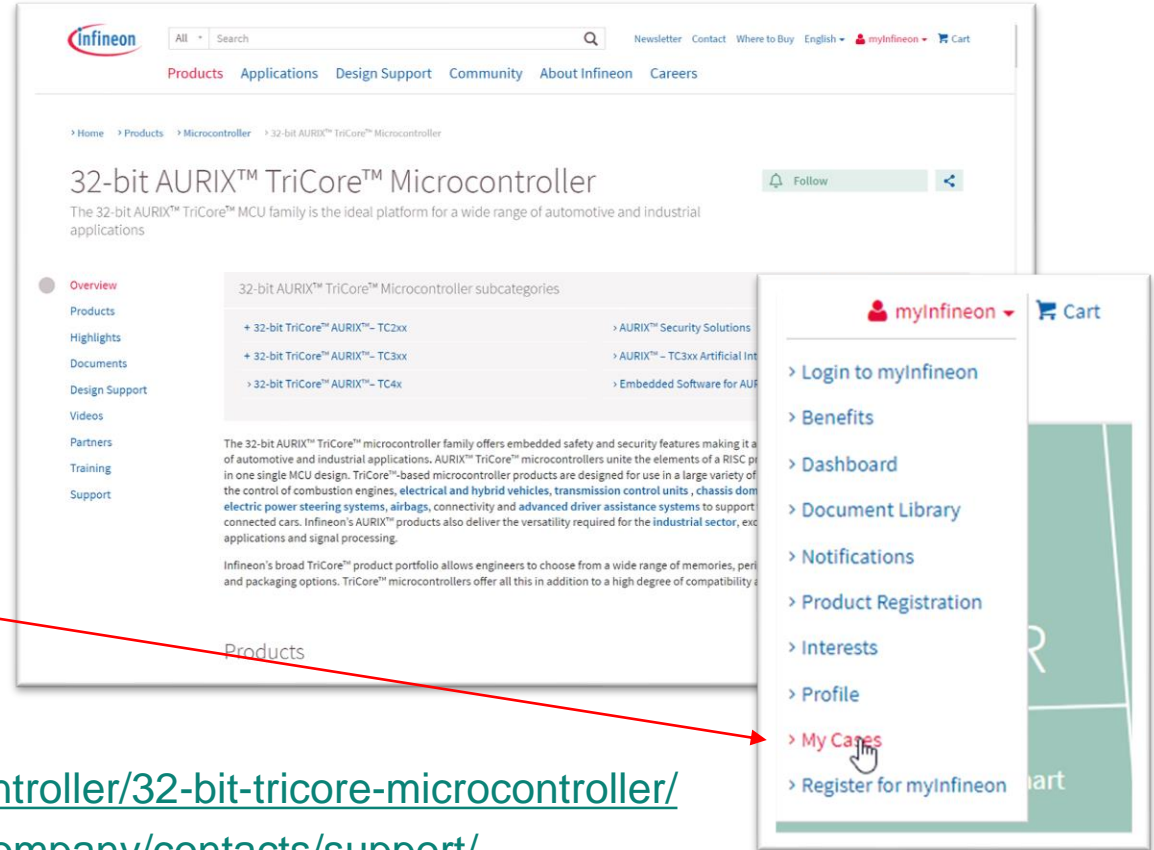
— 可以下载开放和公开的数据表和用户手册

— 主要包含以下栏目：

- 产品型号查询
- 文档
- 开发板及套件
- 合作伙伴
- 培训资料
- 支持



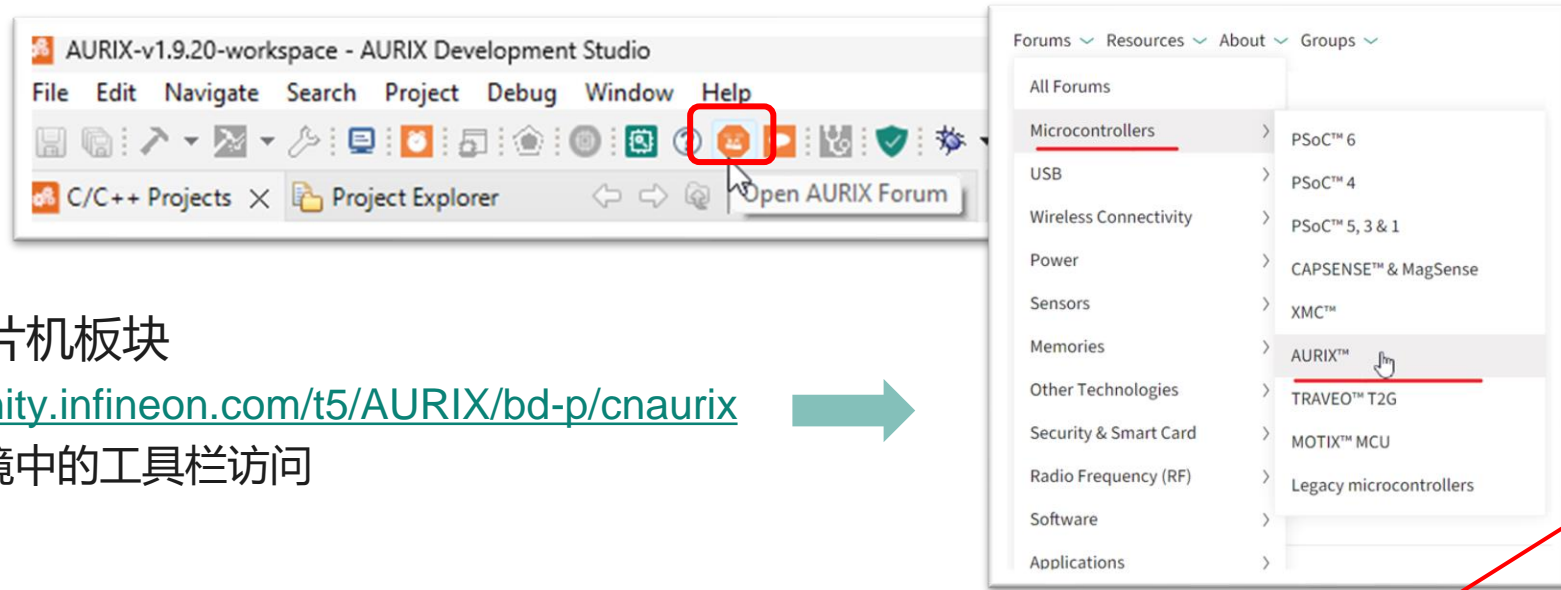
- 通过MyCases系统提问(需要授权)
- 通过论坛提问 (公开的)



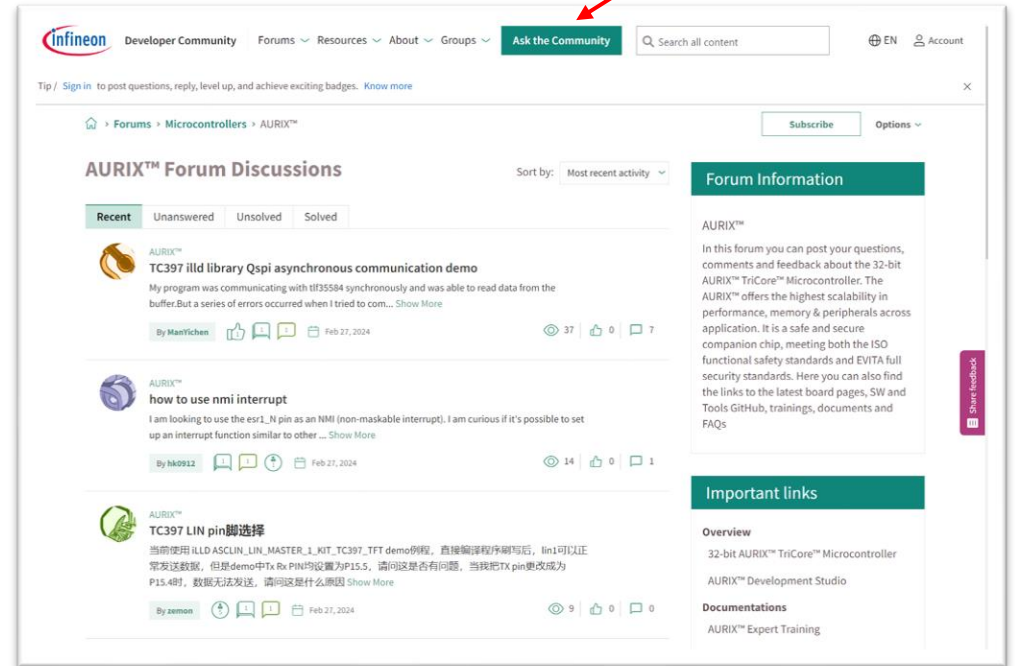
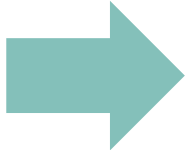
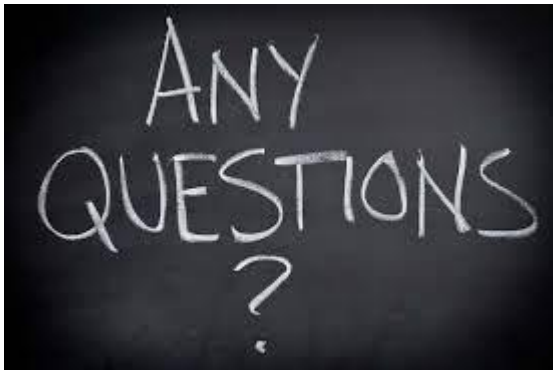
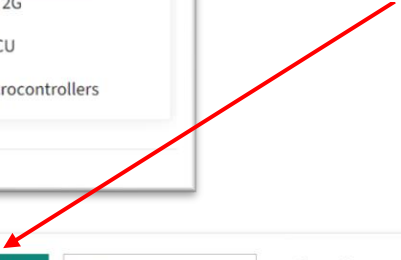
- 链接：<https://www.infineon.com/cms/en/product/microcontroller/32-bit-tricore-microcontroller/>
- 支持：<https://www.infineon.com/cms/en/about-infineon/company/contacts/support/>
- 获取更多TC3xx的资料，请注册myICP (我的英飞凌协作平台)：
- <https://www.infineon.com/cms/en/product/promopages/MyICP-platform-for-Microcontroller/>

14: Aurix 社区

- 英飞凌社区的Aurix单片机板块
 - 链接: <https://community.infineon.com/t5/AURIX/bd-p/cnaurix>
 - 亦可通过ADS开发环境中的工具栏访问
- 在社区上发帖提问
 - 每年有超过1400个有关Aurix单片机的帖子得到回复
 - 除了其他热心用户
 - 英飞凌的专家团队也会及时回答您的问题



Ask the Community



英飞凌开发者社区- 英飞凌技术支持首选平台

Infineon Developer Community—First Option for Infineon Technical Support



欢迎来到英飞凌开发者社区

76606 讨论

49607 成员

45988 解决方案



在英飞凌开发者社区，工程师和专家7/24 快速解决您任何关于产品和运用问题。平台总计拥有76,606个讨论，108,607名成员，85%的问题将在2天内完美解决



拥有海量技术资料(14,000个技术博客，7,200个知识库文章/培训/项目)，同时您可以搜索你想要解决的问题和信息



扫描右边的二维码，注册并创建您的社区账户，解锁所有会员福利，或使用您的 MyInfineon凭证进行登录。根据您的贡献获得对应奖章和等级升级



您可以在全中文内容的界面上自由浏览，同时可以尽情浏览其他语言界面，使您能够与全球工程师自由互动

向英飞凌工程师团队提问



扫码与全球工程师交流



Ask



Answer



Share



Connect



Collaborate

restricted

Questions Categories in Community

Error Solving during Products Application

#An example

请问下我在使用PSoC Creator4.4打开电脑文件路径“C:\Program Files (x86)\Cypress\EZ-PD CCGx Host SDK\CCGx\Firmware\projects\CYPD4226-40LQXI_notebook\CYPD4226-40LQXI_notebook.cywrk的这个工程文件的时候，刚开始我是可以打开看的，后面关闭后，再打开就打不开了，然后下面提示了如下的这些信息，请问这是怎么回事呢？我应该怎么解决？



Manual Finding

Example

请问XMC1302的Keil例程在哪里可以找到完全的版本，有没有模块化的例程？

您好，英飞凌社区的工程师们

我目前正在学习XMC1302，请问在哪里可以找到完整的，打包好的基于Keil的XMC底层驱动库的LLD例程
就算是我目前找到的例程，其中大多写在一个.c文件中，逻辑混杂，我几乎看不懂

有没有类似于模块化的例程可以提供，就是每一个模块都是各自独立的.c文件，比如ADC的操作在ADC.c中，Uart的操作在Uart.c中，这样逻辑比较清晰我比较容易看懂。



Principles Confusion

#Example

在输出电流频率>1kHz时，相电流出现畸变，是由于它不能工作在这么高的输出频率下？还是设置的外围电路不正确？

15: Aurix 生态系统

- AURIX™ Preferred Design Houses (PDH)
 - 英飞凌AURIX™首选设计合作伙伴
 - PDH为客户提供定制化的技术解决方案和扩展支持
 - PDH从英飞凌获得关于芯片的培训
 - PDH拥有更贴近客户应用的技术解决方案和专业知识
- 链接: <https://www.infineon.com/cms/en/partners/partner-ecosystems/aurix-preferred-design-houses-offerings/>

- PDH的等级

- 基础级:
 - 为客户提供基本原则和初级技能, 为设计团队提供基础培训
- 高级:
 - 提供高水平的项目特定应用支持和咨询服务
- 专家级:
 - 具备广泛的知识和能力, 能够全面支持项目的发展

AURIX™ Preferred Design Houses offerings

	GLOBAL	EMEA	AMERICAS	CHINA	KOREA
Author	+	+	+	+	+
Motor Control	•	•	•	•	•
Inverter Power	•	•	•	•	•
Lighting	•	•	•	•	•
PFC Power conversion	•	•	•	•	•
AURIX™ general support	•	•	•	•	•
AURIX™ general support ECU	•	•	•	•	•
Safety Support	•	•	•	•	•
Safety Support (ISO26262)	•	•	•	•	•
Security support (SHE+)	•	•	•	•	•
XMC™ general support	•	•	•	•	•
XMC™ general support ECU	•	•	•	•	•
CAN & classification	•	•	•	•	•
CANopen	•	•	•	•	•
Safety tool for XMC™	•	•	•	•	•
24 GHz radar	•	•	•	•	•
77 GHz radar	•	•	•	•	•
Brochure	•	•	•	•	•
Email	•	•	•	•	•
Phone	•	•	•	•	•

CAN-FD标准

- 2015年CAN-FD由博世开发，对经典CAN总线ISO11898-1标准的扩展
- 由英飞凌、恩智浦、戴姆勒、通用汽车推广，定名为：ISO11898-1:2015(E)
- CAN是“控制器局域网”的缩写
- FD指“灵活速率”
- 帧格式、数据速率有所不同
- 仲裁期-数据期
 - 仲裁期 up to 1Mbit/s，数据期速率up to 8Mbit/s
- 其他细节请参考经典CAN总线的常识：
 - 位填充
 - 终端电阻
 - 错误检测
 - 不归零编码
 - 差分信号
 - 逐位仲裁

CAN和CAN-FD的速率区别

- 经典CAN：10kHz – 1MHz，固定速率
- 常用的：500kbit/s或1Mbit/s，取决于整个系统，固定速率，8字节每帧
- 有竞争、仲裁机制

- CAN-FD：可达8Mbit/s，可达64字节每帧
- 常用的：仲裁期的名义速率为500kbit/s，数据期可以达到4Mbps

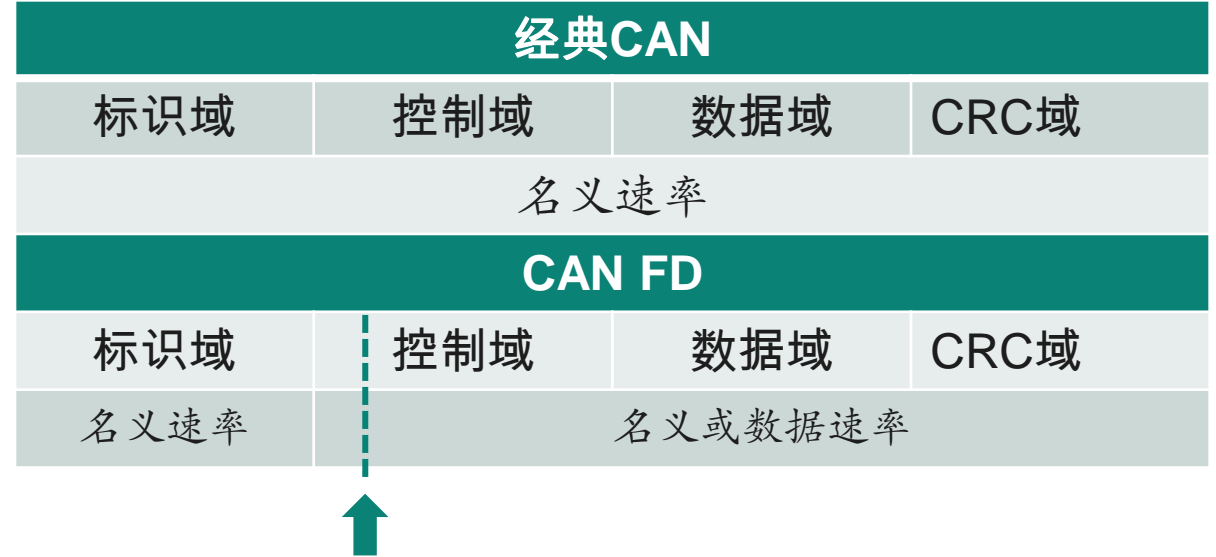
- 仲裁过程：
 - 发0（显性）的继续发，发1（隐性）的竞争不过，放弃发
 - 1逻辑-隐性：CAN_H CAN_L都为2.5V
 - 0逻辑-显性：CAN_H和CAN_L为3.5V、1.5V

- 确认过程：
 - ACK位是帧最后一位，传输方默认发隐性位。
 - 当总线上任一设备发出显性信号时，将告知总线，至少有一个设备已经确认接收

CAN和CAN-FD帧格式区别

- 经典CAN总线每帧0-8字节，CAN-FD则是每帧0-64字节
- 经典CAN总线CRC为15bit，CAN-FD的CRC为17/21bit
- DLC标志位：数据长度编码
- BRS控制位，速率切换（数据阶段以更高速率传输）
- 不改变速率，仍可享有*数据有效载荷和CRC覆盖范围
- 标准CAN没有专门的位表明速率，所有设备需要统一速率
- CAN-FD也同样没有专门的位表明速率，系统的速率是统一的
- CAN-FD若BRS有效（隐性‘1’），则需改变比特率（数据期正式开始）

- * CAN FD的有效载荷和CRC覆盖范围都增加了



CAN-FD配置

- 具有16个中断线，中断可以触发CPU、DMA、GTM
- 可配置：波特率、引脚
- 可配置：Message RAM
- 采样点及Bit Timing计算器
- 等等

帧格式

- 帧格式有4种：
- CBFF 11bit ID 8 byte data(经典基本帧格式)
- CEFF 29bit ID 8 byte data (经典扩展帧格式)
- FBFF 11bit ID 64 byte data (FD基本帧格式)
- FEFF 29bit ID 64 byte data (FD扩展帧格式)

TTCAN和CAN ISO-11898

- TTCAN：ISO-11898-4，基于时间触发，有预定的时间表，特定的时间间隔。
- CAN：ISO-11898-1，基于事件的，非确定的，有碰撞和仲裁



用于KIT_AURIX_TC397_TFT开发板的MCMCAN_FD_1例程

MCMCAN FD 数据传输



例程的主要功能

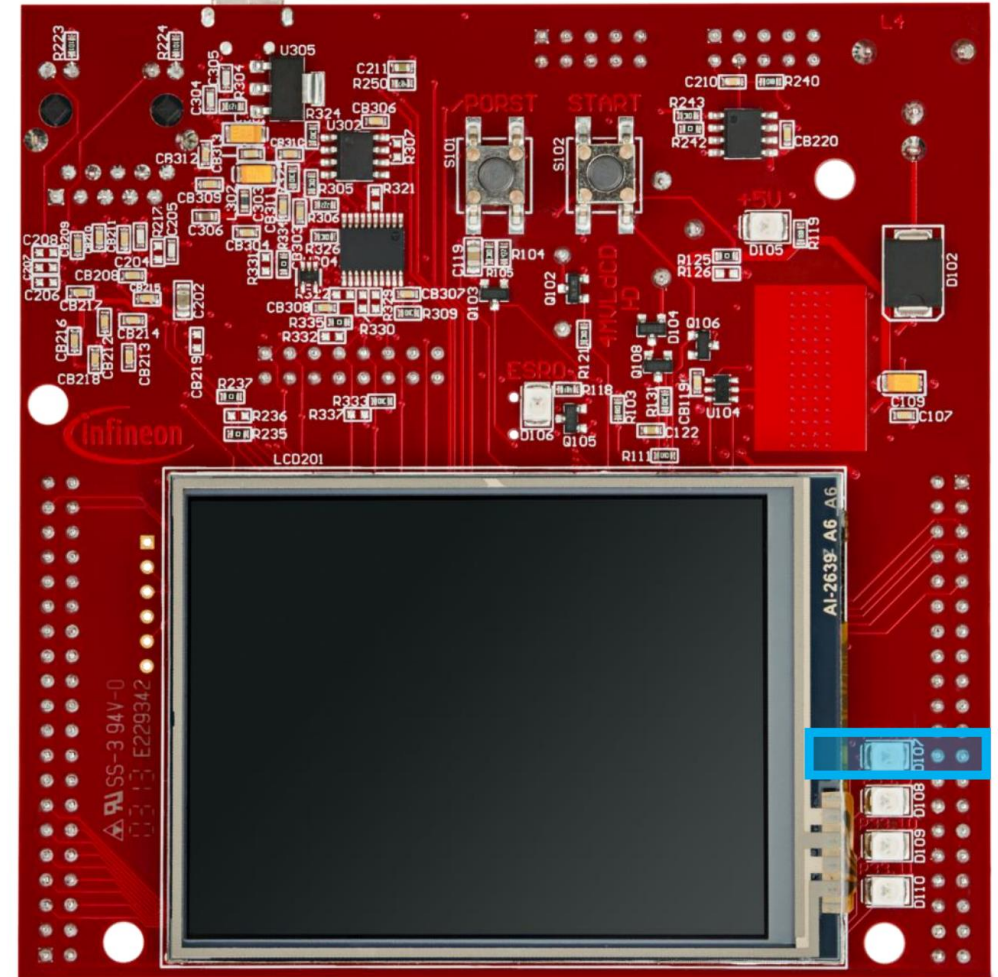
- 在灵活数据速率模式下，MCMCAN用于在同一设备中使用环回模式交换两个节点之间的数据。
- CAN消息从CAN节点0通过环回模式发送到CAN节点1。两个CAN节点都设置为CAN灵活数据速率模式（CAN FD）。每次CAN消息传输并成功接收后，都会生成一个中断。在中断服务例程中，会读取接收到的CAN消息的内容。在成功读取操作的情况下，接收到的数据将与传输的数据进行比较。如果所有消息都在没有检测到任何错误的情况下接收到，LED将点亮以确认成功接收消息。
- 例程的代码下载：
- https://github.com/Infineon/AURIX_code_examples/tree/master/code_examples/MCMCAN_FD_1_KIT_TC397_TFT

介绍

- › MCMCAN是取代AURIX™ TC2xx系列中MultiCAN+模块的新CAN接口。
- › MCMCAN模块支持ISO 11898-1标准下的经典CAN和CAN FD，以及ISO 11898-4标准下的基于时间触发的CAN (TTCAN)。
- › MCMCAN模块由M_CAN作为CAN节点组成（在AURIX™ TC39x设备中为4个节点），这些节点支持CAN FD。每个CAN节点通过两个引脚（TXD和RXD）进行通信。此外，还提供了用于测试目的的**内部环回模式**功能。
- › **可配置的消息RAM**用于存储要传输或接收的消息。消息RAM由MCMCAN模块内的所有CAN节点共享。

硬件设置

- 这个代码示例是为KIT_A2G_TC397_5V_TFT开发的。
- 在这个示例中，使用了LED D107。



实现

CCCRi.FDOE	CCCRi.BRSE	TxMsgk_T1.FDF	TxMsgk_T1.BRS	TxMsgk_T0.XTD	XTD Transmit Behavior
1	1	0	0	0	Classical CAN frames with standard ID (ISO 11898-1)
1	1	0	0	1	Classical CAN frames with extended ID (ISO 11898-1)
1	1	1	0	0	Long frame with standard ID
1	1	1	1	1	Long + Fast frame with extended ID

- 这个代码示例涵盖了四种不同的CAN FD使用情况。这些使用情况的定义在“g_useCaseConf”表中提供。

实现

- 应用程序代码可以分为四个部分：
 - › 使用initMcmcan()函数初始化MCMCAN模块以及相应节点和消息对象的初始化
 - › 使用initLed()函数初始化连接到LED（板上的D107）的端口引脚。LED用于验证CAN消息接收的成功。这在initLed()函数中完成
 - › 使用transmitCanMessage()函数传输配置的CAN消息
 - › 使用verifyCanMessage()函数验证接收到的CAN消息
- 另外，还实现了一个额外的中断服务例程（ISR）：
 - › 在RX中断时，ISR读取接收到的CAN消息，并在没有错误的情况下递增计数器，表示成功接收的CAN消息数量（在canIsrRxHandler()函数中实现）。

实现

- MCMCAN模块初始化
- 初始化分为三个阶段：
 - › 使用函数IfxCan_Can_initModuleConfig()将默认的CAN模块配置加载到配置结构中。
- 然后，使用函数IfxCan_Can_initModule()对CAN模块进行用户配置的初始化。
 - › 使用函数IfxCan_Can_initNodeConfig()将默认的CAN节点配置加载到配置结构中。使用函数IfxCan_Can_initNode()对CAN节点（CAN节点0和1）进行初始化，设置不同的CAN节点ID值，并定义了环回模式的使用。CAN节点0被定义为“源节点”，而CAN节点1表示“目的节点”。两个节点都设置为CAN FD长+快速帧模式。专用的TX缓冲区用于传输CAN消息，而接收则由RX FIFO 0结构处理。最后，在接收到CAN消息时会触发中断。
 - › 所有用于MCMCAN模块和节点初始化的函数都在iLLD头文件IfxCan_Can.h中声明。

实现

- LED连接的引脚初始化
- LED用于验证CAN消息接收的成功。在使用LED之前，必须配置LED连接的端口引脚。
 - › 第一步是将端口引脚设置为“高电平”；这会使LED保持关闭状态作为默认状态（IfxPort_setPinHigh()函数）。
 - › 第二步是使用IfxPort_setPinModeOutput()函数将端口引脚设置为推挽输出模式。
 - › 最后，通过IfxPort_setPinPadDriver()函数定义了引脚驱动器强度。
- 所有函数均在iLLD头文件IfxPort.h中声明。

实现

- CAN消息的传输
- 在传输CAN消息之前，需要初始化两个CAN消息（TX和RX）。基于当前的CAN FD使用情况配置TX消息（有关详细信息，请参见前页）。TX消息数据内容（即传输的数据内容）使用以下格式初始化，由当前数据负载字节和当前CAN消息值的组合：
- RX消息（存储接收到的CAN消息的消息）使用默认配置进行初始化（成功传输CAN后，值将被替换为有效内容）。此外，TX和RX消息数据内容都需要无效化。直到中断服务例程读取接收到的数据之前，不会传输任何额外的CAN消息。

bit	7	6	5	4	3	2	1	0
data content	<i>g_currentCan Message</i> range: 0 - 3		<i>currentDataPayloadByte</i> range: 0 - 63					

实现

- CAN消息的传输
 - › 使用IfxCan_Can_initMessage()函数初始化TX和RX消息。
 - › 使用memset()函数使TX和RX消息数据内容无效。
 - › 使用IfxCan_Can_sendMessage()函数传输CAN消息。只要返回的状态是IfxCan_Status_notSentBusy（如果存在待处理的发送请求，则会出现此状态），CAN消息将持续传输。
- 函数IfxCan_Can_initMessage()和IfxCan_Can_sendMessage()在iLLD头文件IfxCan_Can.h中声明。函数memset()在标准C库头文件string.h中声明。

实现

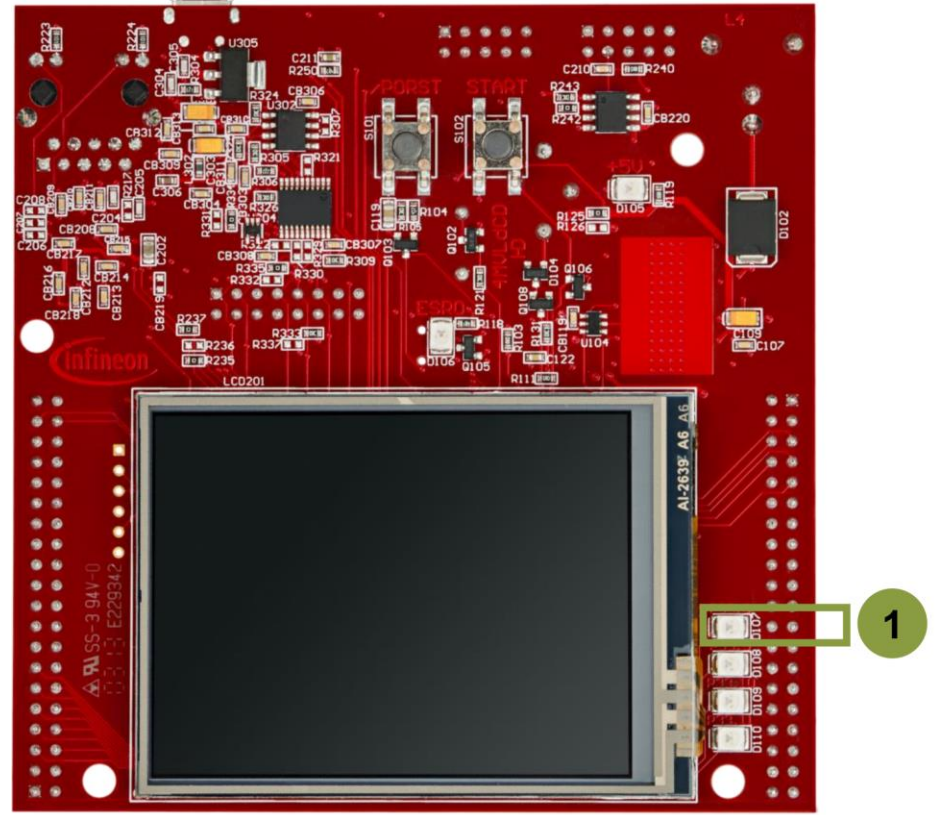
- CAN消息的验证
- 在成功接收每个CAN消息后，执行几项检查：
 - 1. 消息ID检查（检查接收到的消息ID是否与发送的消息ID匹配）。验证已接收到标准ID和扩展ID。
 - 2. 消息长度检查（检查接收到的消息长度是否与发送的消息长度匹配）。该检查涵盖了经典CAN和CAN FD帧大小。
 - 3. 帧模式检查（检查接收到的FD格式（FDF）和比特率切换（BRS）位字段值是否与发送的值匹配）。
 - 4. 预期有效数据检查（检查接收到的数据是否与预期数据匹配）。涵盖了经典CAN和CAN FD数据内容。
 - 5. 无效数据检查（检查无效数据是否在CAN传输中未被修改）。
- 如果没有观察到任何错误，在verifyCanMessage()函数返回时，g_status变量将保存CanCommunicationStatus_Success值。

实现

- 中断服务例程 (ISR)
- 成功接收CAN消息时会触发中断服务例程。
 - > RX中断服务例程通过使用IfxCan_Node_clearInterruptFlag()函数清除挂起的中断标志, 并使用IfxCan_Can_readMessage()函数读取接收到的CAN消息。一旦接收到的CAN消息被读取, 全局变量g_isrRxCount将被递增。该变量用作计数器, 指示成功接收的CAN消息数量。
- 函数IfxCan_Node_clearInterruptFlag()在iLLD头文件IfxCan.h中声明, 而函数IfxCan_Can_readMessage()在iLLD头文件IfxCan_Can.h中声明。

运行并测试

- 在编译代码并将设备刷新后，请观察以下行为：
 - 检查LED D107 (1) 是否亮起（所有CAN消息已成功接收并通过了所有检查）。



参考



- AURIX™ Development Studio 可在线获取：
 - › <https://www.infineon.com/aurixdevelopmentstudio>
 - › 使用“导入...”功能以获取更多代码示例。
- 更多代码示例可以在GIT存储库中找到：
 - › https://github.com/Infineon/AURIX_code_examples
- 如需额外培训，请访问我们的网页：
 - › <https://www.infineon.com/aurix-expert-training>
- 如有问题和需要支持，请使用AURIX™论坛：
 - › <https://www.infineonforums.com/forums/13-Aurix-Forum>



CAN FD Reference 参考资料

- Introduction to TC3xx MCMCAN Module:
 - https://www.infineon.com/dgdl/Infineon-AURIX_TC3xx_CAN_Interface-Training-v01_00-EN.pdf?fileId=5546d46274cf54d50174da4ee8cb226c
 - TC3xx MCMCAN 模块介绍
- Overview of CAN FD Protocol:
 - <https://www.kvaser.com/about-can/can-fd/>
 - <https://www.kvaser.com/can-fd-protocol-tutorial/>
 - CAN FD 协议介绍
- Helpful Tips for CAN Module Pin Settings:
 - <https://fastupload.io/5hDtDQDkTQre6ux/preview>
 - CAN 模块引脚设置技巧

SMU 安全管理单元模块 介绍

英飞凌资深工程师



Jeremy Zhou

拥有10年MCU产品支持经验



Safety Management Unit (SMU) 介绍

Jeremy Zhou



Agenda

1

SMU 故障管理功能介绍

2

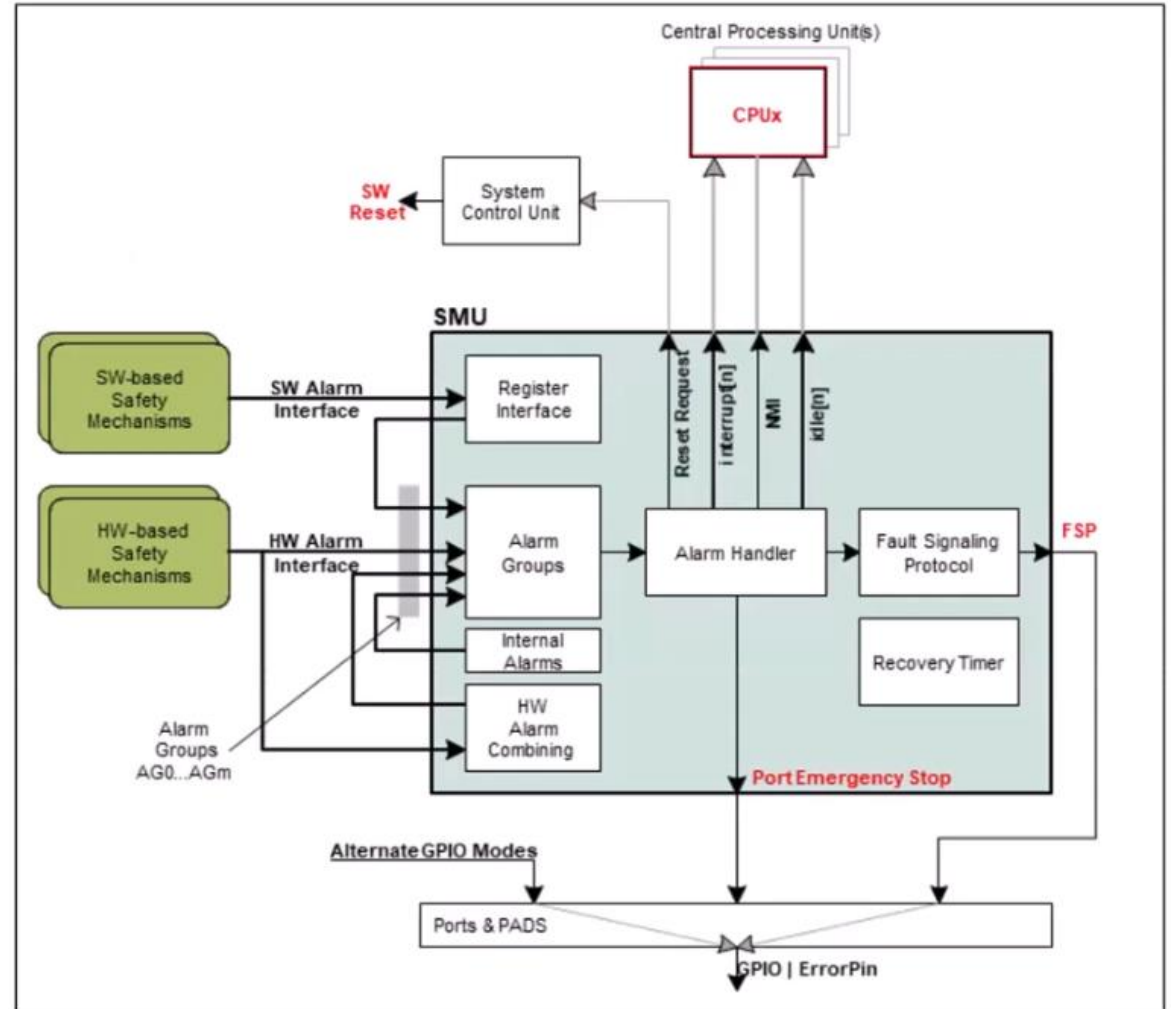
SMU_Core 架构

3

SMU_Standby 架构

Aurix 故障处理

- SMU 是故障处理架构的核心组件
- 在大多数情况下，可以诊断出影响 MCU 的故障，并执行不同的操作



SMU 功能介绍

- SMU 是安全架构的核心组件，集中所有与不同的硬件和软件安全机制相关的警报信号，提供通用接口，用于管理存在故障（bus error, ...）的微控制器的行为。
- 可为每个警报单独配置预设反应 (reaction)

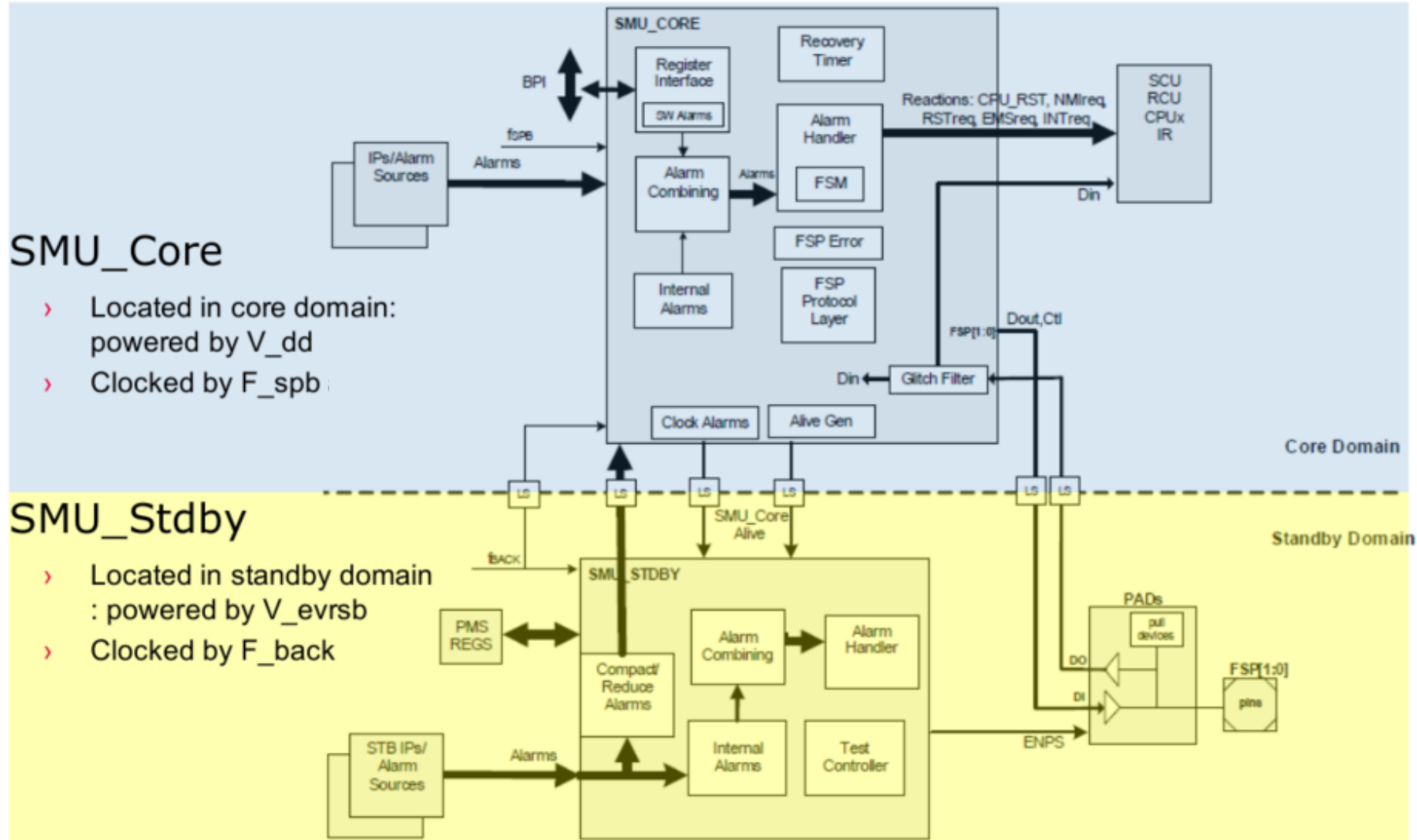
External reaction

- Transition **Fail Safe Protocol** on the error pin (P33.8) to “fault state”
- Optional: active **Port Emergency Stop** (switch selected port outputs to inputs)

Internal reaction

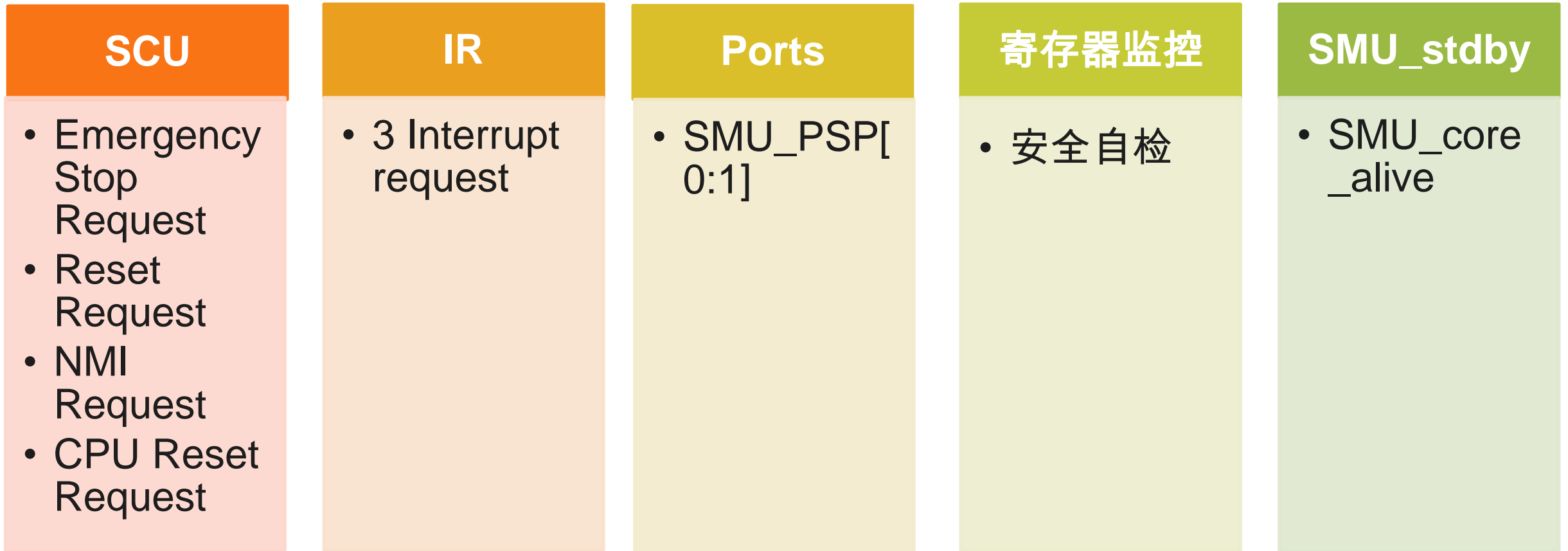
- Issue **NMI** to all CPUs
- Issue **interrupt** to a configurable set of CPUs
- Issue a **system reset** or **application reset**
- Issue a **RESET** to a configurable set of CPUs

SMU 架构概述



核心域 SMU (SMU_Core)

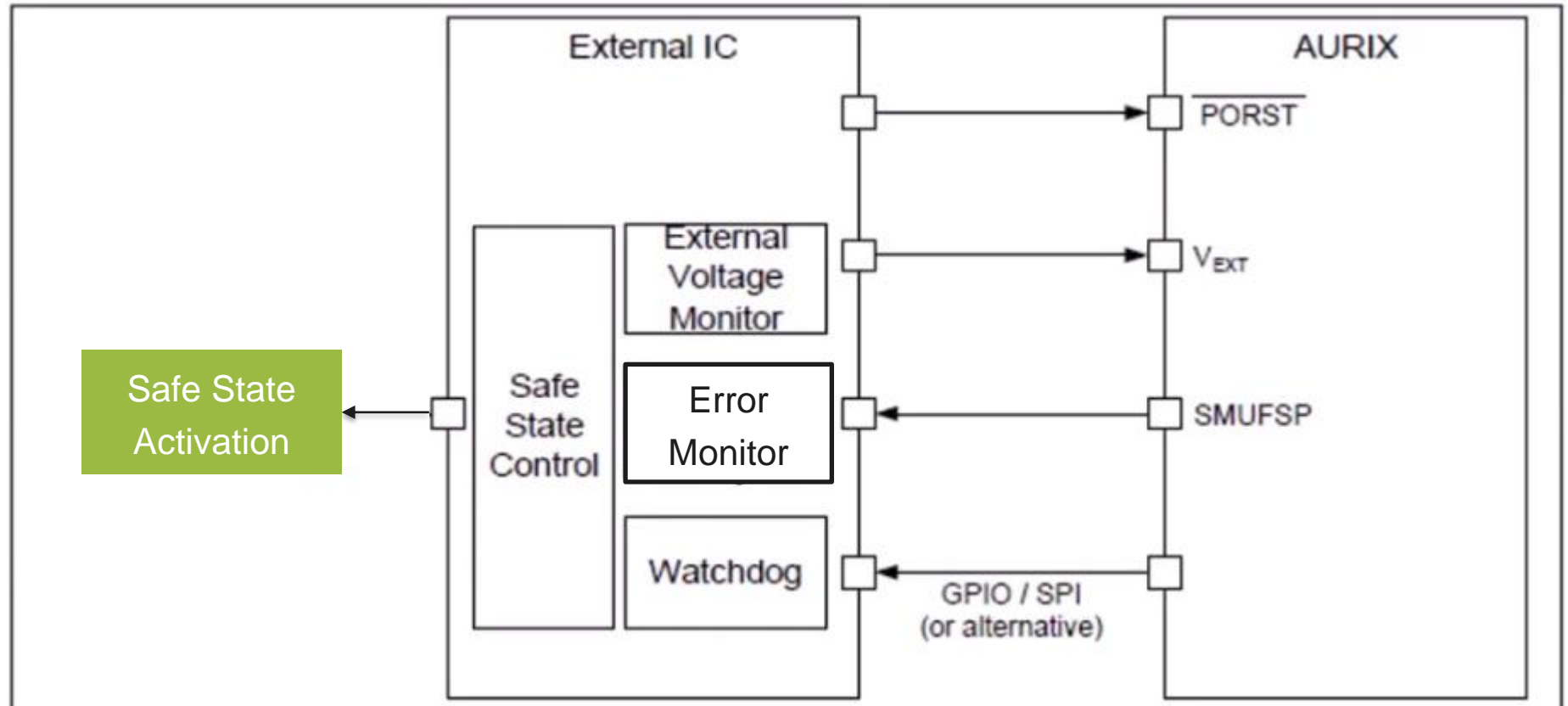
- 核心域 SMU (也称为 SMU_core) 作为安全概念定义的硬件监视器 (hardware monitors) 实施安全机制可收集大多数警报信号。
- 接口 (Interfaces) 概述



故障信号协议 (Fault Signaling Protocol, FSP)

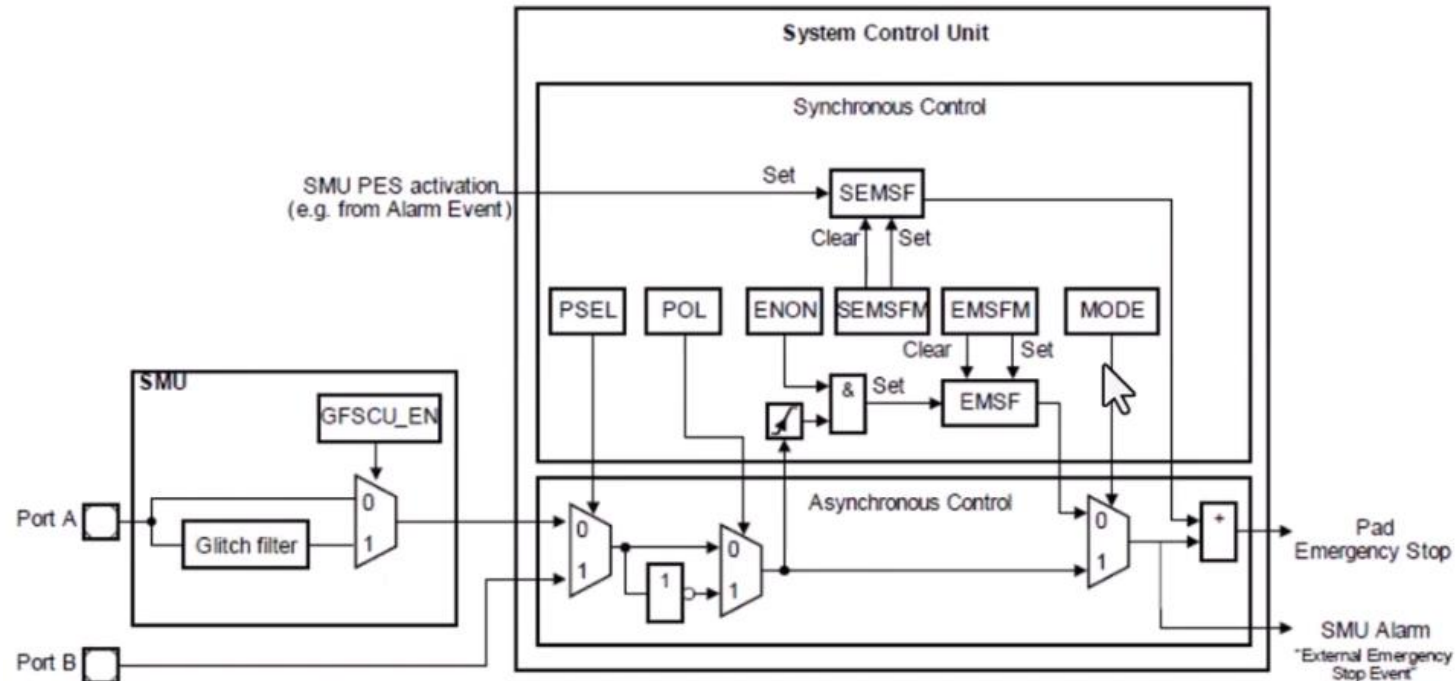
故障信号协议使微控制器能够向外部安全控制器设备报告紧急情况，以控制安全系统的安全状态。

- Bi-stable protocol (default)
- Dynamic dual-rail protocol
- Time-switching protocol



Port紧急停止 (Port Emergency Stop)

- Port紧急停止功能可强制Pad进入通用输入模式 (General Purpose Input Mode)。SCU的端口紧急停止请求可通过以下任何情况激活：
 - SMU_ActivatePES()软件命令
 - SMU_AG<x>FSP 启用且 FSP 启用的报警事件
 - 在 SMU_AG<x>CFX 寄存器中配置了内部操作且为该操作启用了 SMU_AGC.PES 的警报事件



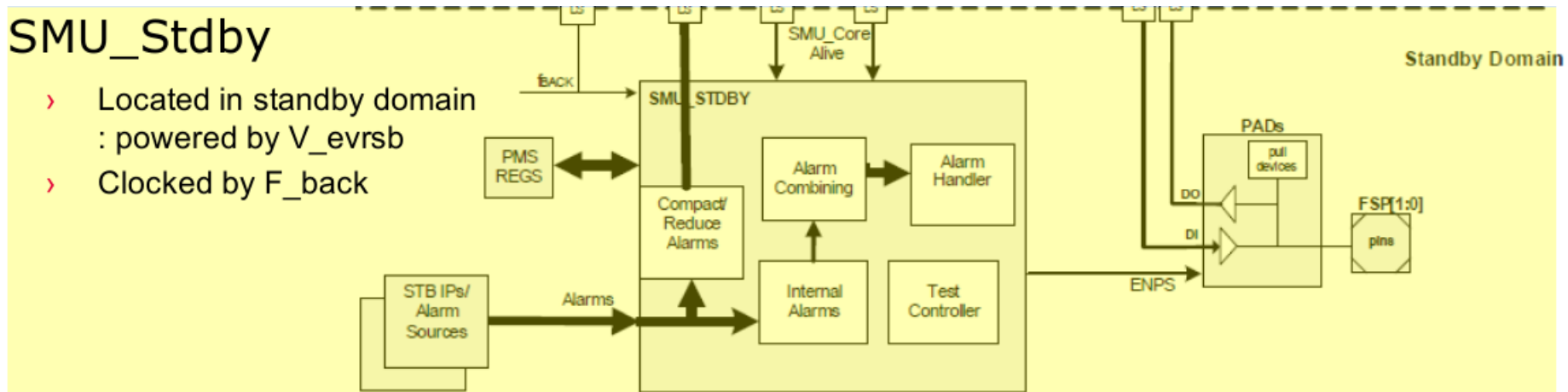
恢复计时器 (Recovery Timer, RT)

RT允许在超时前有时间对警报做出反应并尝试恢复，以控制安全系统的安全状态。

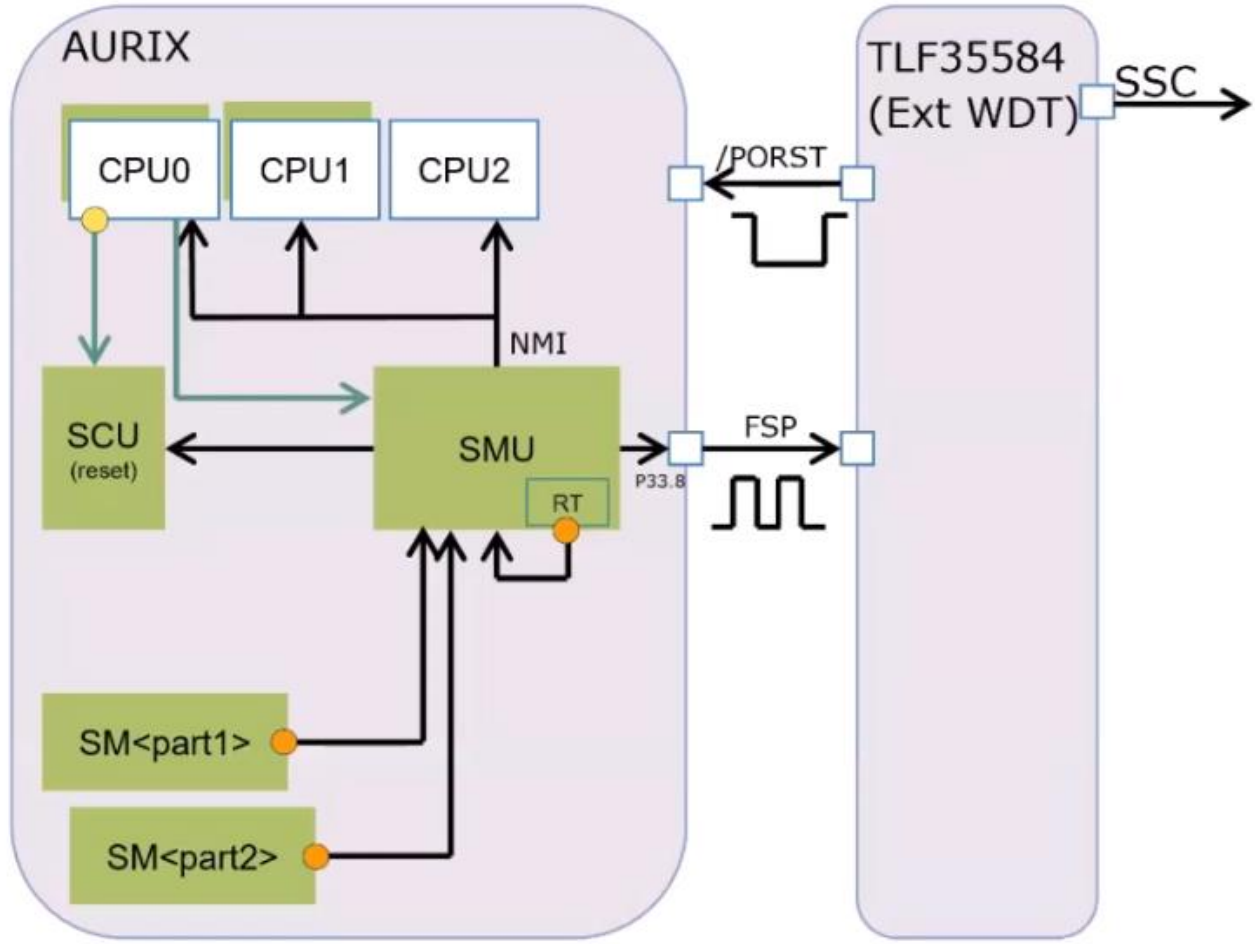
- 如果启用，警报会启动RT，同时，警报也触发 NMI 或中断，以启动错误处理程序
- 在错误处理程序中，可尝试RT停止计数，否则，计数超时后，会触发超时警报，导致reset发生

SMU_Stdby

- 备用域 SMU (也称为 SMU_Stdby) 从检测时钟 (无时钟), 电源 (电压过低 / 过高) 和温度故障 (温度过低 / 过高) 的模块收集警报。
- 收集 SMU_Alive 报警信号
- 实现了内置自检功能, 允许用户测试 SMU_Stdby 对报警信号的反应以及从次级电压监视器到 SMU_Stdby 的完整报警路径



故障处理案例



ADC 模数转换模块介绍

英飞凌高级资深工程师



李坤乔

拥有9年汽车电子工作经验，精通车规级嵌入式芯片软件开发和设计，熟悉BLDC/PMSM电机相关应用领域



Enhanced Versatile Analog-to-Digital Converter (EVADC)

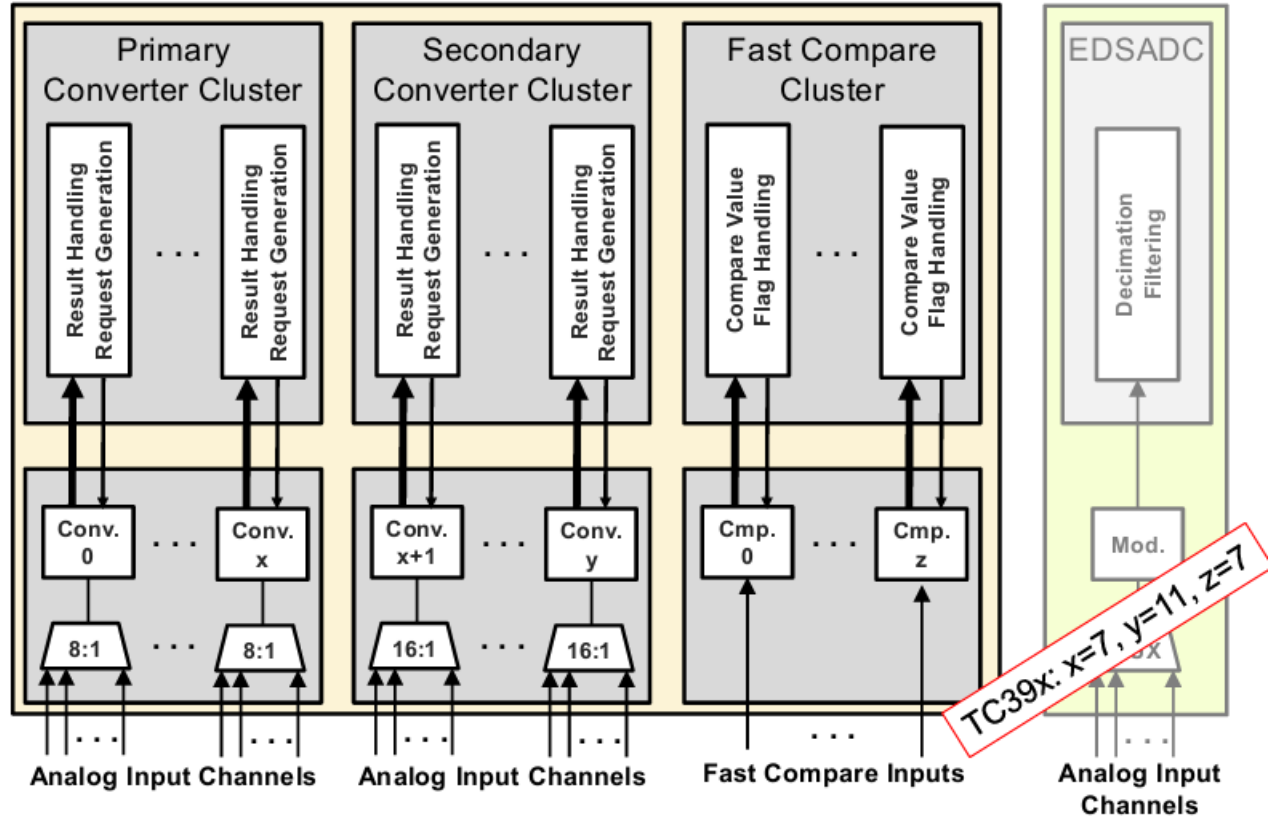
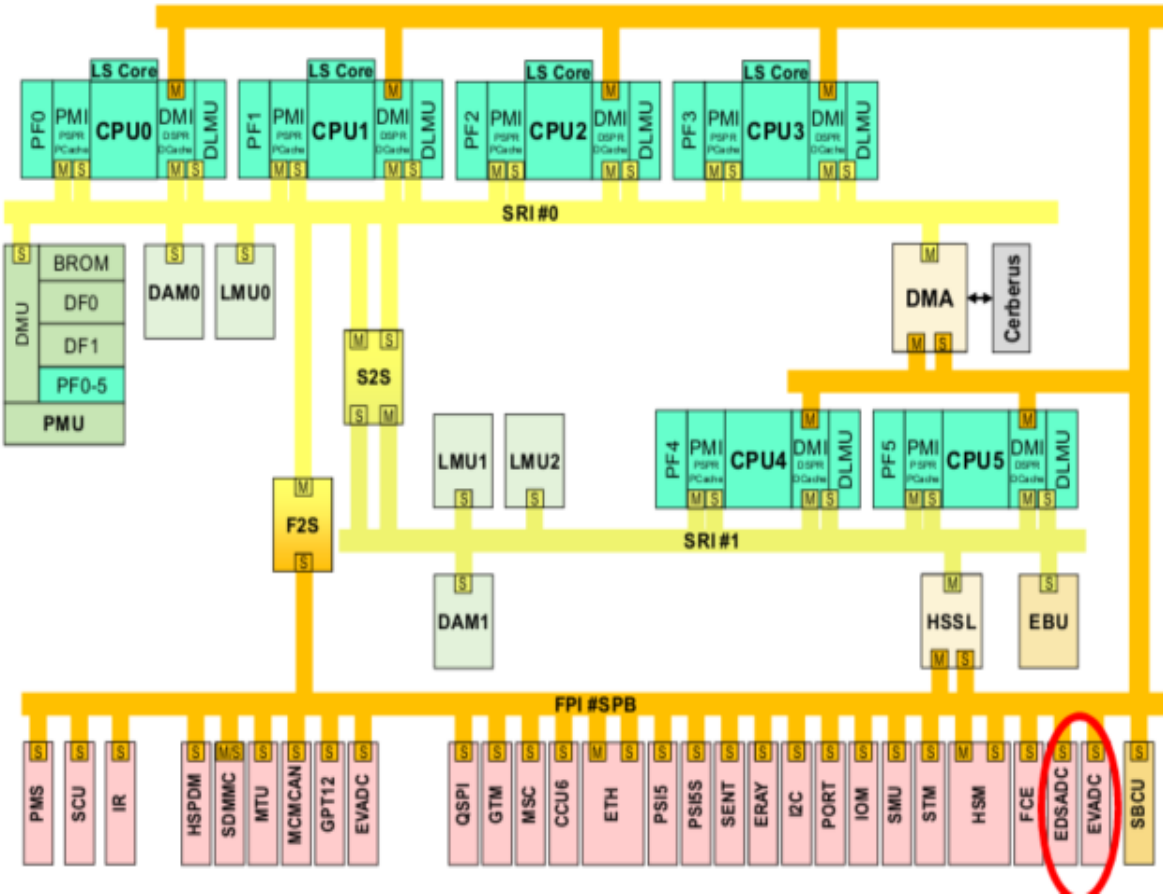
Kunqiao Li



Content

- › 基础介绍
- › 技术指标
- › 特性列表
- › 工作模式
- › 仲裁机制
- › 触发方式
- › 服务响应
- › 数据处理

基础介绍



技术指标

	TC39x (16 MB)	TC38x (10 MB)	TC37x (6 MB)	TC36x (4 MB)	TC33x (2 MB)
Total # of independent Sampling Stages	12	12	8	6	4
Secondary ADC Groups	4 x SAR	4 x SAR	4 x SAR	2 x SAR	2 x SAR
Primary ADC Groups	8 x SAR	8 x SAR	4 x SAR	4 x SAR	2 x SAR
# of Fast Compare Sub Channels	8	4	2	2	0
# of DSADC Converter	14	10	6	4	0
# of analog inputs AI + GPIO	74 + 28 (= 102)	72 + 28 (= 100)	48 + 28 (= 76)	48 + 12 (= 60)	24 + 10 (= 34)

Converter reference clock	f_{ADCl} SR	16	40	53.33	MHz	$4.5 \text{ V} \leq V_{\text{DDM}} \leq 5.5 \text{ V}$
		16	20	26.67	MHz	$2.97 \text{ V} \leq V_{\text{DDM}} < 4.5 \text{ V}$
Total Unadjusted Error ^{1) 2)}	TUE CC	-4	-	4	LSB	12-bit resolution for primary/secondary groups, 10-bit resolution for fast compare channels

Element	20.0 MHz (DIVA = 7)	26.7 MHz (DIVA = 5)	40.0 MHz (DIVA = 3)	53.3 MHz (DIVA = 2)
Sample time = 100 ns	2 x 50 ns = 100 ns	3 x 37.5 ns = 112.5 ns	4 x 25 ns = 100 ns	6 x 18.75 ns = 112.5 ns
Sample time = 500 ns	10 x 50 ns = 500 ns	14 x 37.5 ns = 525 ns	34 x 25 ns = 850 ns	34 x 18.75 ns = 637.5 ns
Result generation	13 x 50 ns = 650 ns	13 x 37.5 ns = 487.5 ns	13 x 25 ns = 325 ns	13 x 18.75 ns = 243.75 ns
Postcalibration	4 x 50 ns = 200 ns	4 x 37.5 ns = 150 ns	4 x 25 ns = 100 ns	6 x 18.75 ns = 112.5 ns
Sync postcalibration	5 x 6.25 ns = 31.25 ns	5 x 6.25 ns = 31.25 ns	5 x 6.25 ns = 31.25 ns	5 x 6.25 ns = 31.25 ns
Sync statemachine	3 x 6.25 ns - 18.75 ns	3 x 6.25 ns - 18.75 ns	3 x 6.25 ns - 18.75 ns	3 x 6.25 ns - 18.75 ns
Noise reduction step (0, 1, 3, 7)	218.75 ns	168.75 ns	118.75 ns	93.75 ns
Conversion with postcalibration	1000 ns	800 ns	575 ns	518.75 ns
Conversion with 3 noise red. steps (CMS = 10 _g) and postcalibration	1656.25 ns	1306.25 ns	931.25 ns	800 ns
Conversion without postcalibration, primary groups	768.75 ns	618.75 ns	443.75 ns	375 ns
Conversion without postcalibration, secondary groups	1168.75 ns	1031.25 ns	1193.75 ns	900 ns
Maximum conversion rate	1.3 MS/s	1.6 MS/s	2.2 MS/s	2.6 MS/s
Compare steps	2 x 50 ns = 100 ns	2 x 37.5 ns = 75 ns	2 x 25 ns = 50 ns	2 x 18.75 ns = 37.5 ns
Fast compare operation	218.75 ns	206.25 ns	168.75 ns	168.75 ns
Maximum fast compare rate	4.5 MS/s	4.8 MS/s	5.9 MS/s	5.9 MS/s

Details datasheet based on the product

[32-bit-tricore-aurix-tc3xx](#)

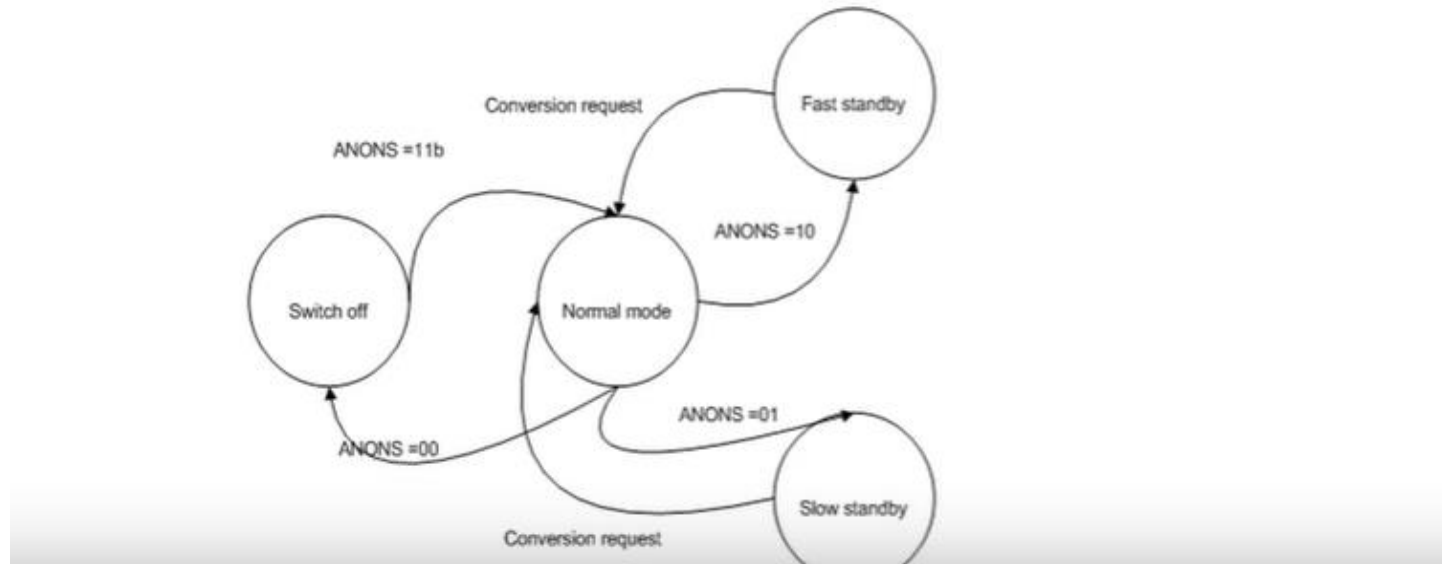
特性列表

Nominal Analog Supply Voltage	<ul style="list-style-type: none"> • 5.0 V • 3.3 V
Source Selection and Arbitration	<ul style="list-style-type: none"> • Programmable arbitrary conversion sequence (single or repeated) • Request source chaining to generate extended conversion sequences • Cancel-inject-restart mode
Trigger Sources	Software, timer (GTM) events, external events
Result Handling	Independent result registers
	Configurable limit checking against programmable border values
	Storage of maximum/minimum value
	Data rate reduction (adding conversion results)
	FIR/IIR filter with selectable coefficients
Fast Compare (5 Msamples/s)	<ul style="list-style-type: none"> • Automatic handling of flags and output signals • Compare value adjustable via SW, via conversion channel or via ramp
Service Request	Selectable events can trigger service requests
Built-In Safety	<ul style="list-style-type: none"> • Register access protection • Broken wire detection • Multiplexer test mode • Automatic execution of test sequences
Parallel Conversions	2 up to 4 ADCs
Suspend/Power Saving Modes	<ul style="list-style-type: none"> • Suspend Mode • Power down Mode

工作模式

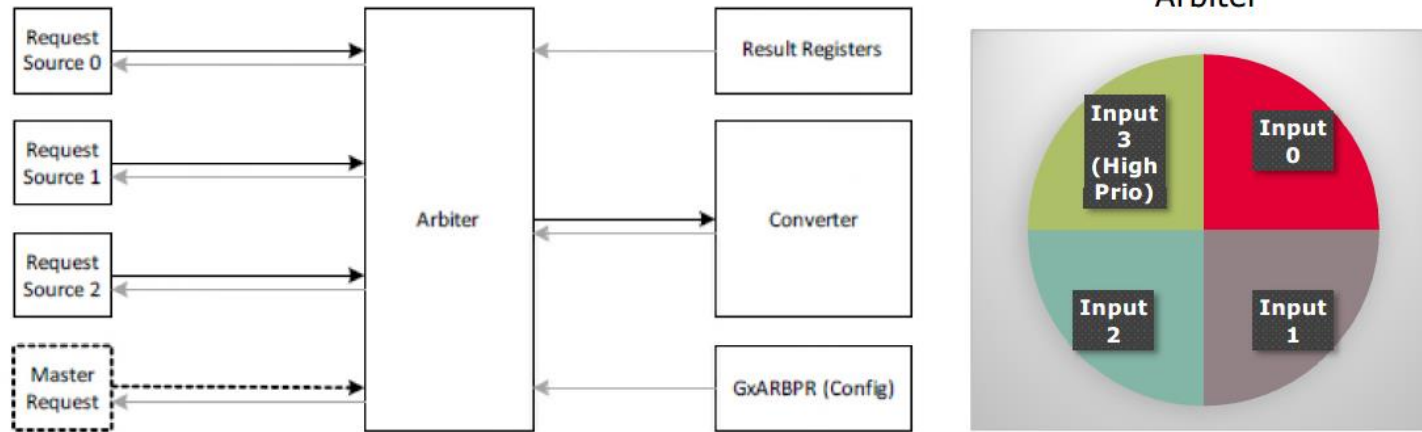
Configure the operating mode in **GxARBCFG.ANONS**

- 11b : Normal Operation
- 10b : Fast Standby mode
- 01b : Slow Standby mode
- 00b : Converter switched Off (default after reset)



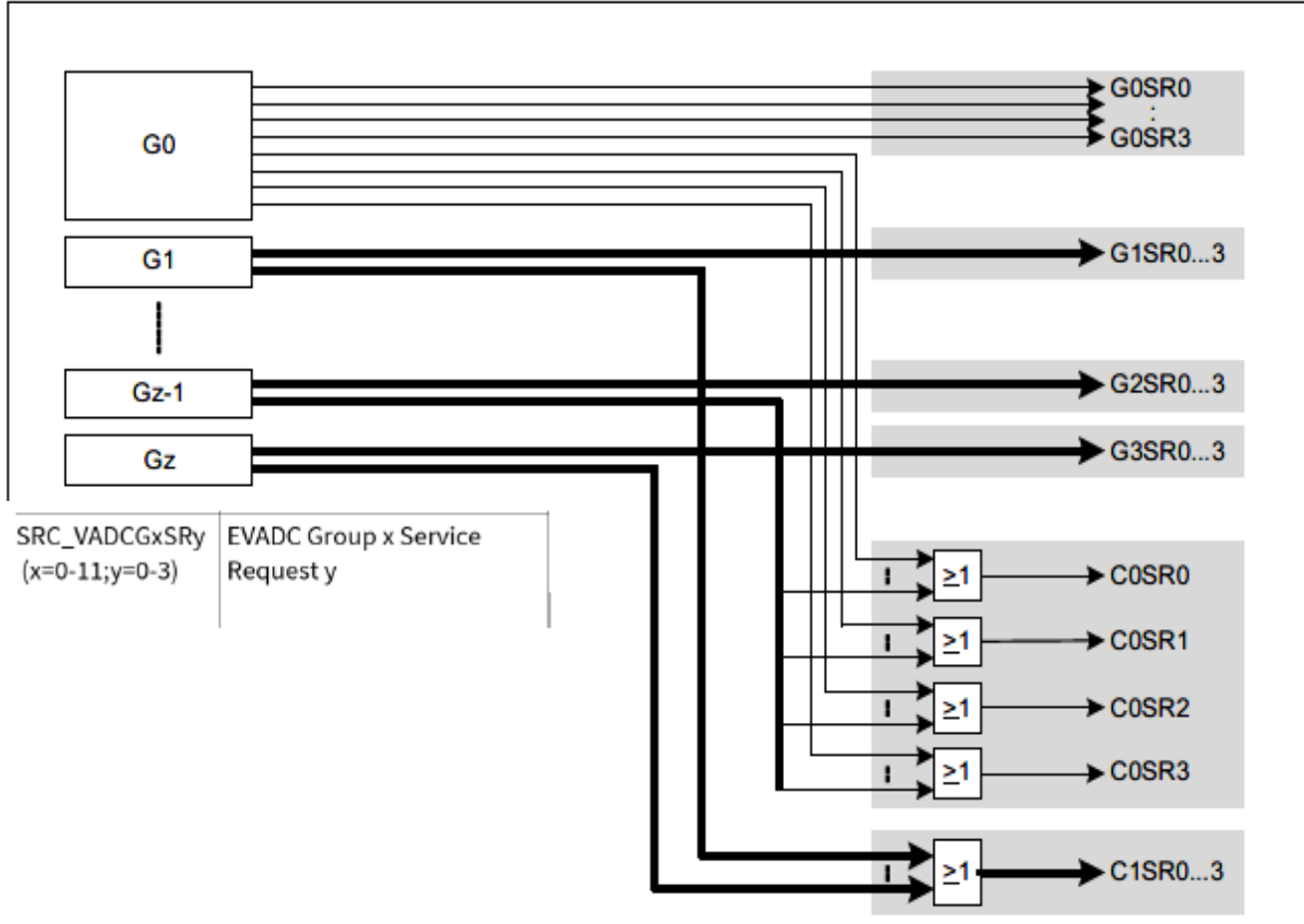
仲裁机制

- › The request source arbiter resolves the different request sources in a combinatorial way which means there is no sampling jitter anymore.
 - **Input 0:** Queued source **Q0**, 8/16-stage sequences in arbitrary order
 - **Input 1:** Queued source **Q1**, 8/16-stage sequences in arbitrary order
 - **Input 2:** Queued source **Q2**, 8/16-stage sequences in arbitrary order, intra-group concatenation, test
 - **Input 3:** Synchronization source, synchronized conversion requests from another ADC kernel (always handled with the highest priority in a synchronization slave kernel)



- 仲裁器会有一个非常小的时间去轮询每一个在queue中是否有等待转换的source。
- 每一个queue都可以配置优先级，仲裁器会根据最高优先级的等待转换的queue胜出进行转换。
- 每一个queue中的source不是转换申请发出就立刻可以执行，有等待的可能性。
- Master request的queue始终处在最高优先级。

服务响应

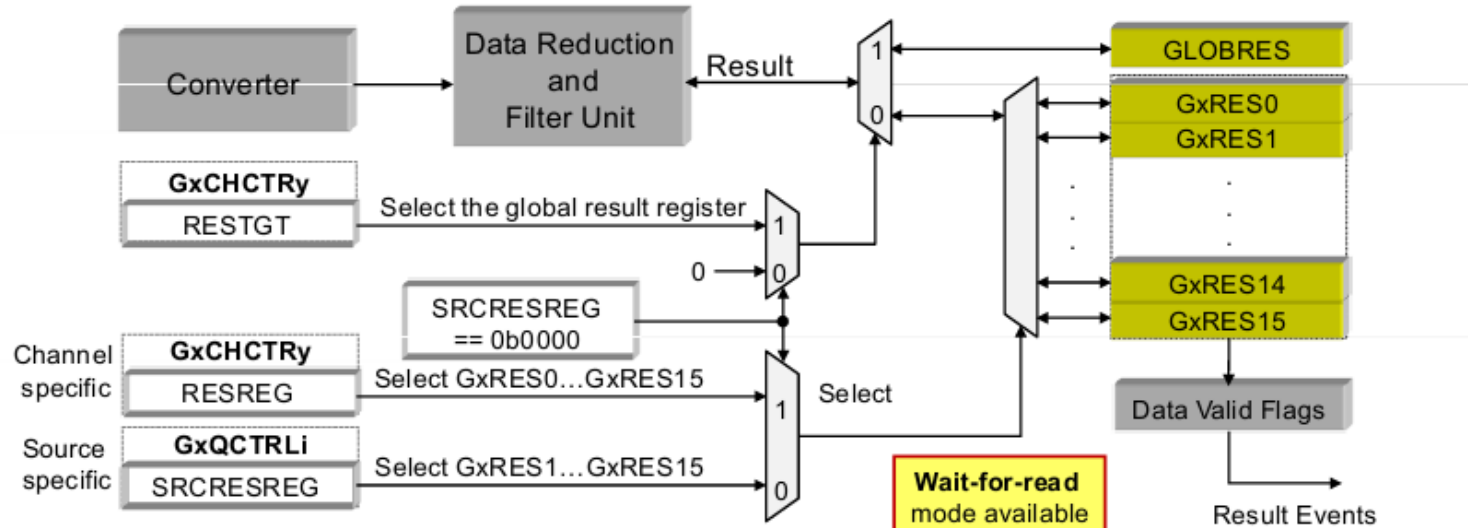
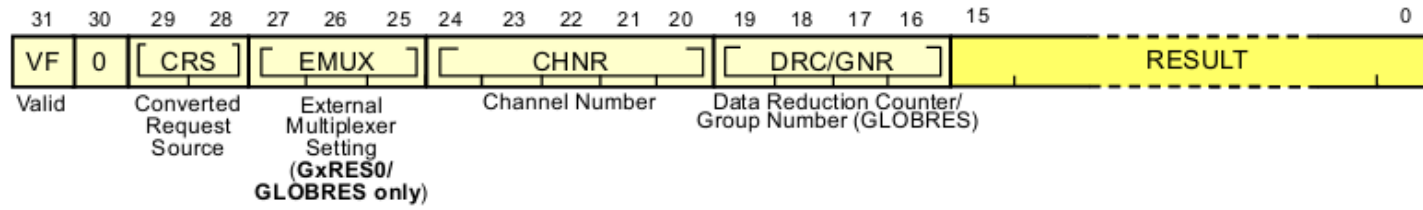


SRC_VADCGxSRy
(x=0-11;y=0-3) | EVADC Group x Service Request y

SRC_VADCCGxSRy
y (x=0-1;y=0-3) | EVADC Common Group x Service Request y

- > **Request source events:** 表明请求源已完成所请求的转换序列，应用软件可以启动进一步的操作
- > **Channel events:** 表明转换已完成。比如在可编程数值范围内的结果数值触发通道事件
- > **Result events:** 表明结果寄存器中有一个新的有效结果

GxRESn (with n = 0...15, x = 0...11), GLOBRES



Result registers:

- **Group** result register returns the result value and the channel number
- **Global** result register returns the result value and the channel number and the group number

Read the result register via the **application view**

实际案例介绍及运用 英飞凌主任工程师



胡强

主要负责AURIX™的车身应用技术支持，十余年MCU开发和
技术支持经验



实际案例介绍及应用

Qiang Hu



Agenda

1

CAN-FD通信模块应用

2

SMU模块的实际应用

3

如何用好ADC模块

CAN-FD通信模块应用1 – 数据地址

以TC39X为例：

- 包含三个CAN模块，每个模块的RAM地址均不相同；
- 当CAN模块访问了其他RAM地址会导致进入Trap。

```
void IfxCpu_Trap_busError(uint32 tin)
{
    volatile IfxCpu_Trap trapWatch;
    trapWatch = IfxCpu_Trap_extractTrapInfo(IfxCpu_Trap_Class_bus, tin);
    IFX_CFG_CPU_TRAP_BE_HOOK(trapWatch);
    IFX_CFG_CPU_TRAP_DEBUG;
    __asm("rslcx"); /* Restore lower context before returning. lower con
    __asm("rfe");
}
```

- 如代码所示，当CAN1模块访问了CAN0模块的RAM地址，会进入数据同步访问错误。

Table 1 MCMCAN RAM and Register Address

Device	Module	RAM
TC39x	CAN0	F020,0000 - F020,7FFF
	CAN1	F021,0000 - F021,3FFF
	CAN2	F022,0000 - F022,3FFF

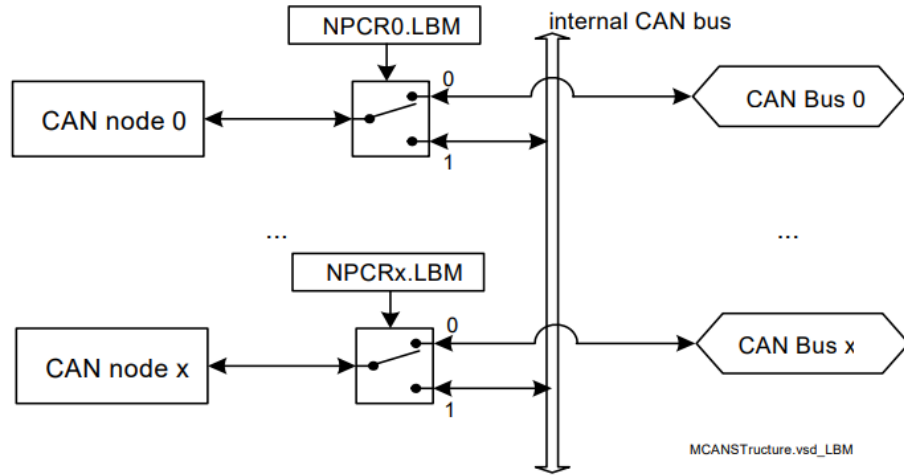
Class 4 – System Bus and Peripheral Errors

1	PSE	Synch.	HW	Program Fetch Synchronous Error.
2	DSE	Synch.	HW	Data Access Synchronous Error.
3	DAE	Asynch.	HW	Data Access Asynchronous Error.
4	CAE	Asynch	HW	Coprocessor Trap Asynchronous Error.
5	PIE	Synch	HW	Program Memory Integrity Error.
6	DIE	Asynch	HW	Data Memory Integrity Error.
7	TAE	Asynch	HW	Temporal Asynchronous Error

CAN-FD通信模块应用2 – 环回模式

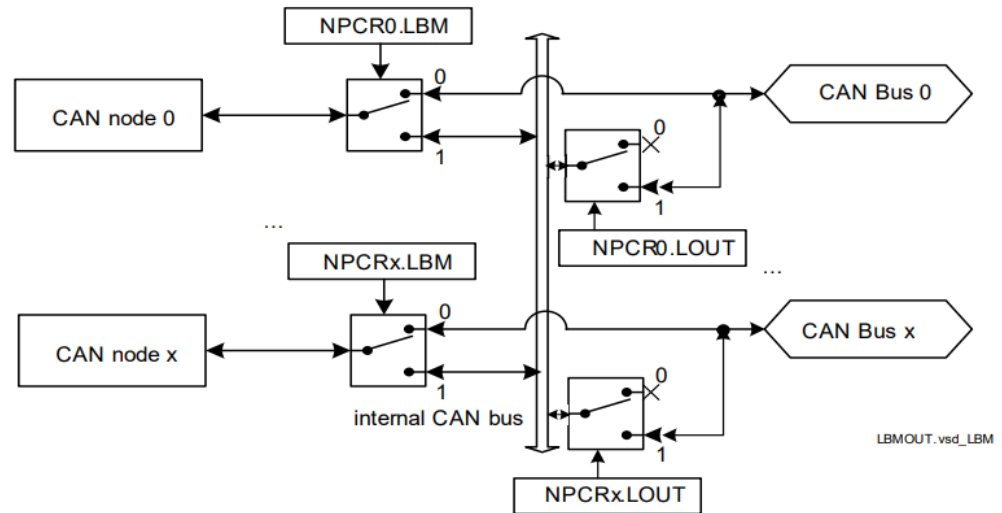
内部环回模式

- 同一个模块内实现四个CAN节点的任意收发；
- 可快速验证CAN节点配置和收发功能；
- 同时也能测试外部收发器是否正常。



外部环回模式

- 在内部环回模式基础上增加了信号输出功能；
- 可使用外部设备监测CAN总线报文；
- 请注意此模式仅适用于**B版本**芯片。



CAN-FD通信模块应用3 – 高优先级报文接收

在某些特殊应用中，需要快速响应特定的CAN报文

硬件机制：

- CAN模块通过滤波器设置，快速匹配高优先级报文；
- 多种处理方式：生成中断，或者叠加存储至FIFO里。

软件处理：

- 非常重要的报文存储地址分配至不同的接收缓存区；
- 提升高优先级报文中断的优先级；
- 注意高优先级报文的索引号与FIFO本身的索引号有区别。

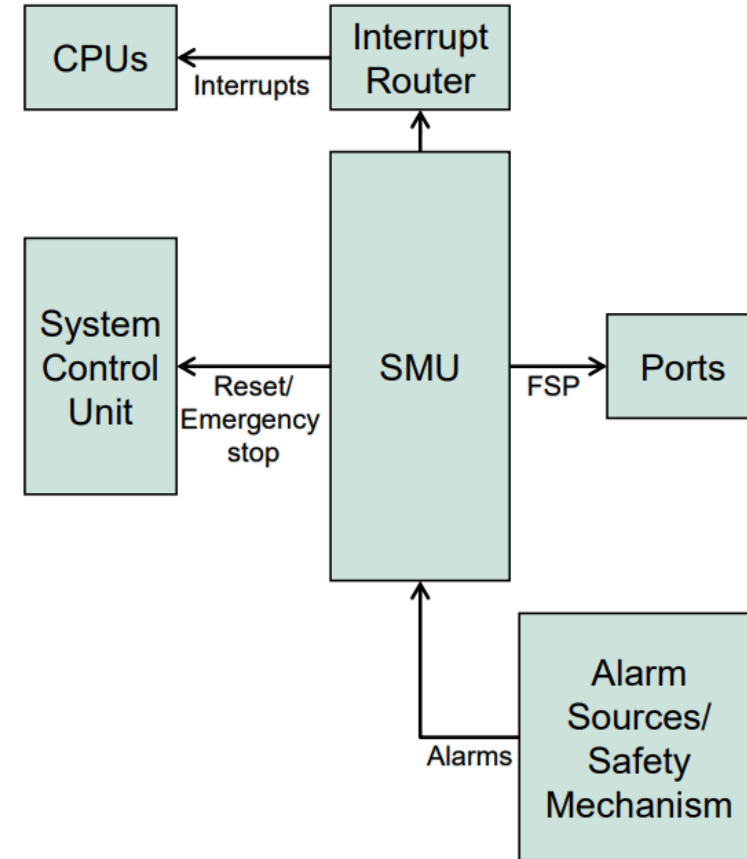
SFEC	29:27	rw	<p>Standard Filter Element Configuration</p> <p>All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = “100”, “101” or “110”, a match sets interrupt flag IR.HPM and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.</p> <p>000_B Disable filter element</p> <p>001_B Store in Rx FIFO 0 if filter matches</p> <p>010_B Store in Rx FIFO 1 if filter matches</p> <p>011_B Reject ID if filter matches</p> <p>100_B Set priority if filter matches</p> <p>101_B Set priority and store in FIFO 0 if filter matches</p> <p>110_B Set priority and store in FIFO 1 if filter matches</p> <p>111_B Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored</p>
------	-------	----	--

SMU模块的实际应用1 – Alarm系统

Alarm是来自于硬件安全机制的异常警报，可以根据系统需求设定异常警报的处理方式，包括：

- 生成中断请求至任意CPU；
- 生成不可屏蔽中断至SCU模块；
- 生成复位请求至SCU模块；
- 激活端口输出紧急停止控制；
- 生成复位请求至CPU。

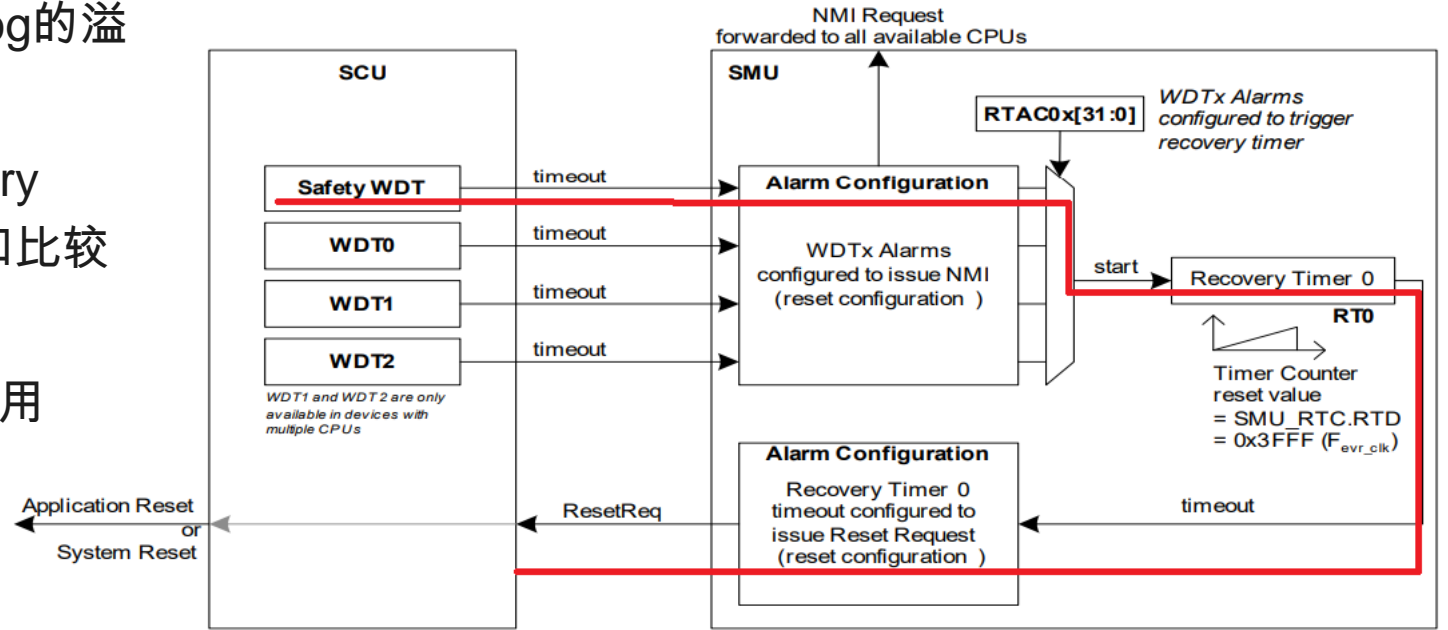
实际应用中比较多的设定是生成不可屏蔽中断至SCU模块，触发SCU的Trap系统，调试时需要仔细检查的。



SMU模块的实际应用2 – WDT系统

在复位后，CPU watchdog或者Safety watchdog的溢出可以直接触发SMU reset：

- 当watchdog发生超时时，会首先触发recovery timer启动，然后当recovery timer的计数值和比较值匹配时，就会发送reset请求给SMM模块；
- ADS开发环境自动生成的main函数会主动禁用CPU watchdog和Safety watchdog；



```
void core0_main(void)
{
    unsigned int num;

    IfxCpu_enableInterrupts();

    /* !!WATCHDOG0 AND SAFETY WATCHDOG ARE DISABLED HERE!!
     * Enable the watchdogs and service them periodically if it is required
     */
    IfxScuWdt_disableCpuWatchdog(IfxScuWdt_getCpuWatchdogPassword());
    IfxScuWdt_disableSafetyWatchdog(IfxScuWdt_getSafetyWatchdogPassword());
}
```

RTAC00[31:0] Selects Alarms to be used by Recovery Timer 0

RES[6:0]	ALID 1[4:0]	GID1[3:0]	RES[6:0]	ALID 0[4:0]	GID0[3:0]
----------	-------------	-----------	----------	-------------	-----------

RTAC01[31:0] Selects Alarms to be used by Recovery Timer 0

RES[6:0]	ALID 3[4:0]	GID3[3:0]	RES[6:0]	ALID 2[4:0]	GID2[3:0]
----------	-------------	-----------	----------	-------------	-----------

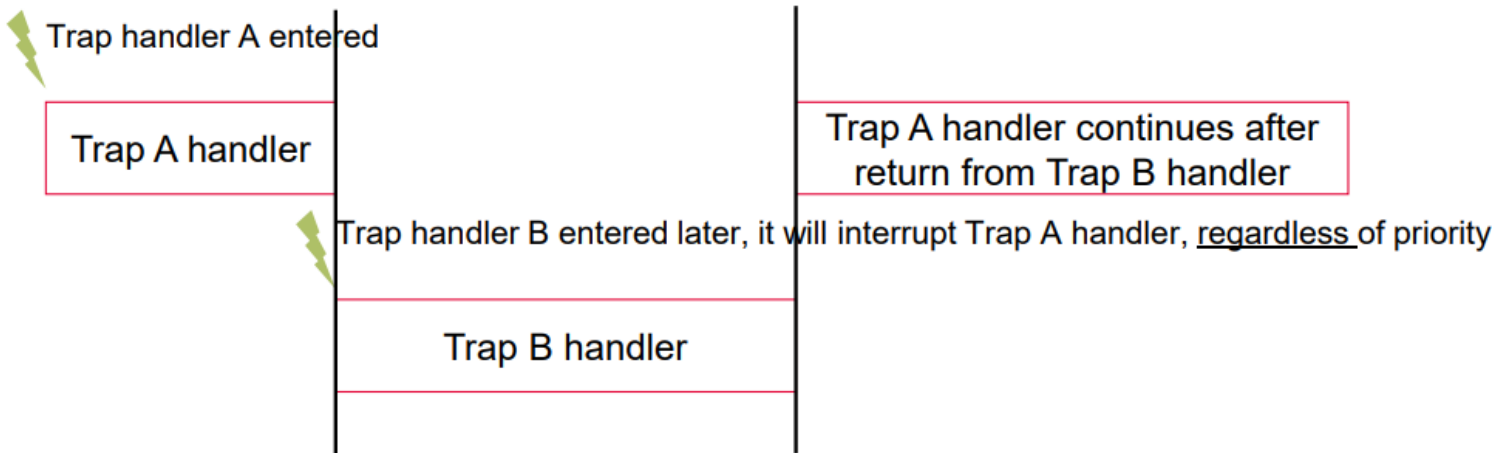
SMU模块的实际应用3 – Trap系统

Trap与中断的区别如右图所示：

- Trap的优先级高于中断；
- 异步Trap的优先级高于同步Trap;

Trap	Interrupt
不能被禁用	可被如下方式禁用： 1, 清除全局中断使能位ICR.IE; 2, 设置对应的中断优先级SRPN为0; 3, 清除对应的中断使能位SRE。
优先级固定，无法配置	优先级可由用户任意配置
只能被CPU处理	可被CPU或者DMA处理

- Trap的优先级只是在有多个Trap同时发生时有效，且低优先级Trap会被忽略；
- Trap是后到的先执行，与中断不一样。



SMU模块的实际应用4 – Trap实际调试

Trap调试时观察点：

- A11寄存器是Trap处理函数返回地址，也就是触发Trap的指令地址，适用于同步Trap;
- D15寄存器是Trap标识符，用来表示触发Trap的类型。

The screenshot displays three windows from an IDE:

- Core Registers:** A table showing the state of various registers. A11 is highlighted with a red box and contains the value 0x80000BB0. D15 is highlighted with a red box and contains the value 0x00000006. Other registers like A0-A15 and D0-D15 are also listed with their values.
- Source Code (IfxCpu_Trap.c):** Shows the implementation of trap handling functions. The function `IfxCpu_Trap_internalProtectionError` is highlighted in yellow. It includes code for restoring context and returning to the instruction that triggered the trap.
- Assembly Code (Ifx_Ssw_Tc0.c):** Shows the assembly instructions corresponding to the source code. A red box highlights the instruction `LD.A %a1, 0`, which is annotated with `a1,temp (0x0)`. This instruction is the return address for the trap handler.

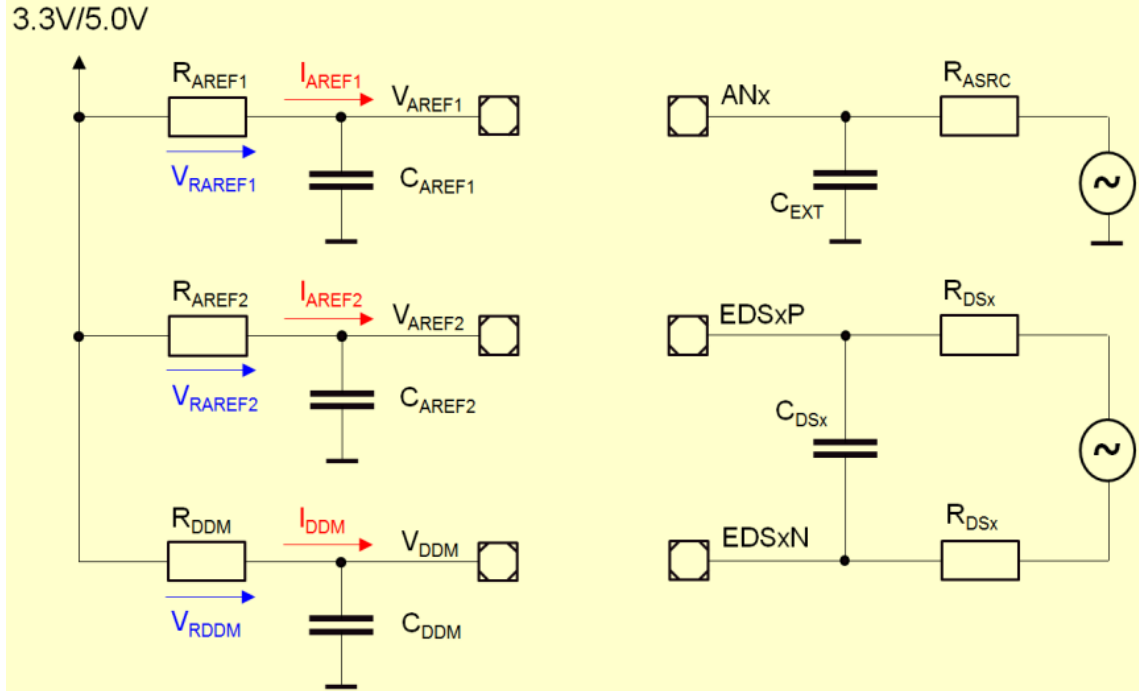
Green arrows indicate the flow of information: one arrow points from the A11 register value to the `LD.A` instruction in the assembly code, and another points from the D15 register value to the `__asm("rslcx");` instruction in the source code.

如何用好ADC模块1 – 输入滤波

滤波设计如右图所示，包括了模拟供电，参考源和ADC输入通道。其中各自的设计参考说明如下：

- VDDM和ADC数量有关，取决于芯片型号和封装；
- VAREF和需要同时转换的ADC通道有关；
- 模拟输入通道和ADC的输入缓冲器设置，预充电设置，转换时间，输入截止频率等相关。
- 常规应用推荐阻容值如下表所示。

管脚名称	推荐阻值	推荐容值	功能说明
VAREF1	4.7Ω	3.3uF	EDSADC参考电压源
VAREF2	8.2Ω	2.0uF	EVADC参考电压源
VDDM	1.0Ω	360nF	模拟电源供电
EVADC IN	10KΩ	27nF	EVADC输入通道/输入缓冲器禁用
EVADC IN	10KΩ	5.1nF	EVADC输入通道/输入缓冲器使能
EDSADC IN	10KΩ	1.6pF	EDSADC差分输入通道



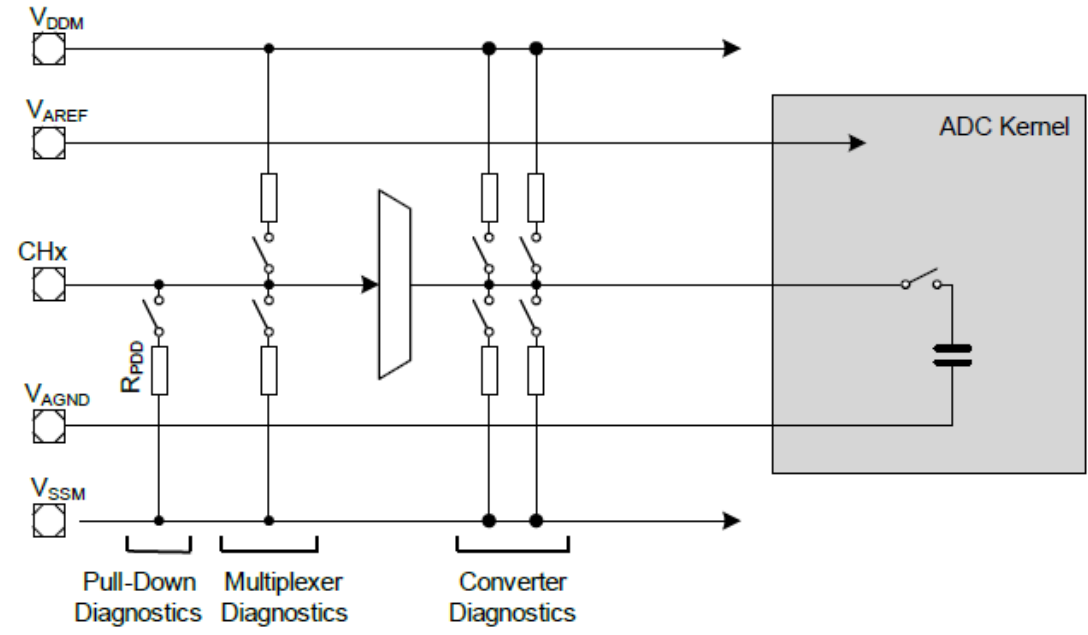
如何用好ADC模块2 – 输入通道选择

GROUP 0	GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP 5	GROUP 6	GROUP 7	GROUP 8	GROUP 9	GROUP 10	GROUP 11
AN0	AN8	AN16	AN24(P40.0)	AN40	AN48	AN56	AN64(P41.8)	AN32(P40.4)	P00.12	P33.7	AN4
AN1	AN9	AN17(P40.10)	AN25(P40.1)	AN41	AN49	AN57	AN65	AN33(P40.5)	P00.11	P33.6	AN5
AN2	AN10	AN18(P40.11)	AN26(P40.2)	AN42	AN50	AN58	AN66	AN34	P00.10	P33.5	AN6
AN3	AN11	AN19(P40.12)	AN27(P40.3)	AN43	AN51	AN59	AN67(P40.15)	AN35	P00.9	P33.4	AN7
AN4	AN12	AN20	AN28(P40.13)	AN28(P40.13)	AN52	AN60	AN68(P41.0)	AN36(P40.6)	P00.8	P33.3	AN8
AN5	AN13	AN21	AN29(P40.14)	AN29(P40.14)	AN53	AN61	AN69(P41.1)	AN37(P40.7)	P00.7	P33.2	AN9
AN6	AN14	AN22	AN30	AN30	AN54(P41.4)	AN62(P41.6)	AN70(P41.2)	AN38(P40.8)	P00.6	P33.1	AN10
AN7	AN15	AN23	AN31	AN31	AN55(P41.5)	AN63(P41.7)	AN71(P41.3)	AN39(P40.9)	P00.5	P33.0	AN11
								AN40	P00.4	P34.4	AN18(P40.11)
								AN41	P00.3	P34.3	AN19(P40.12)
								AN42	P00.2	P34.2	AN26(P40.2)
								AN43	P00.1	P34.1	AN27(P40.3)
ADC各个模块输入通道与管脚映射关系如图：								AN44	P01.5		AN32(P40.4)
选择顺序为黑色，蓝色，红色。								AN45	P01.4		AN33(P40.5)
								AN46	P01.3		AN34
								AN47	P02.11		AN35

如何用好ADC模块3 – 输入诊断

安全特性需要对ADC输入通道进行诊断：

- 通过在不同的位置施加不同阻值来进行诊断；
- 内部集成了强下拉电阻和弱上下拉电阻；
- 包括下拉诊断，复用器诊断，转换器诊断，断路诊断等；
- 可通过软件使能选择不同的诊断类型



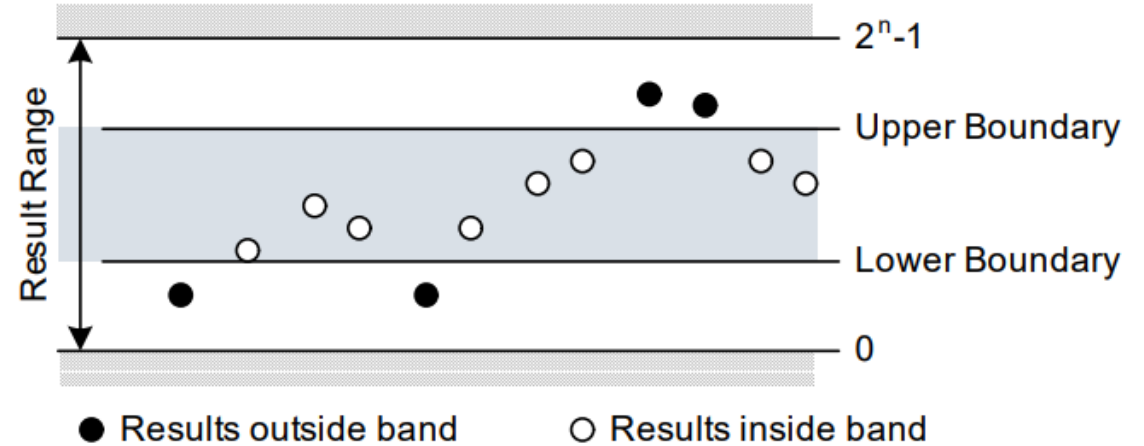
如下表所示。

诊断名称	电阻类型	最小值	最大值	功能说明
下拉诊断	R _{PDD}		0.3KΩ	强下拉电阻，检查与外部信号源的连接
复用器诊断	R _{MDU}	30KΩ	42KΩ	弱上拉电阻与弱下拉电阻 检查多路模拟复用器的功能
	R _{MDD}	43KΩ	58KΩ	
转换器诊断	R _{CSD}	45KΩ	75KΩ	检查ADC转换器是否正常，可设置 V _{DDM} , V _{SSEM} , 1/2*V _{DDM} , 2/3*V _{DDM}

如何用好ADC模块4 – 数字比较器

实现可生成中断的电压监测系统：

- 每个ADC模块包含一路组内比较器；
- 所有的ADC模块共享一路全局比较器；
- 硬件自动判别，可以节省CPU资源。



某实际应用中，需要对其中一路ADC通道的电压值进行监控，希望响应时间越短越好。

- 任务调度周期是1ms；
- 软件判断时会引入最多1ms延时；
- 使用ADC中断会让CPU频繁进入中断；
- 使用数字比较器，只取决于采样周期，延时固定可调。

