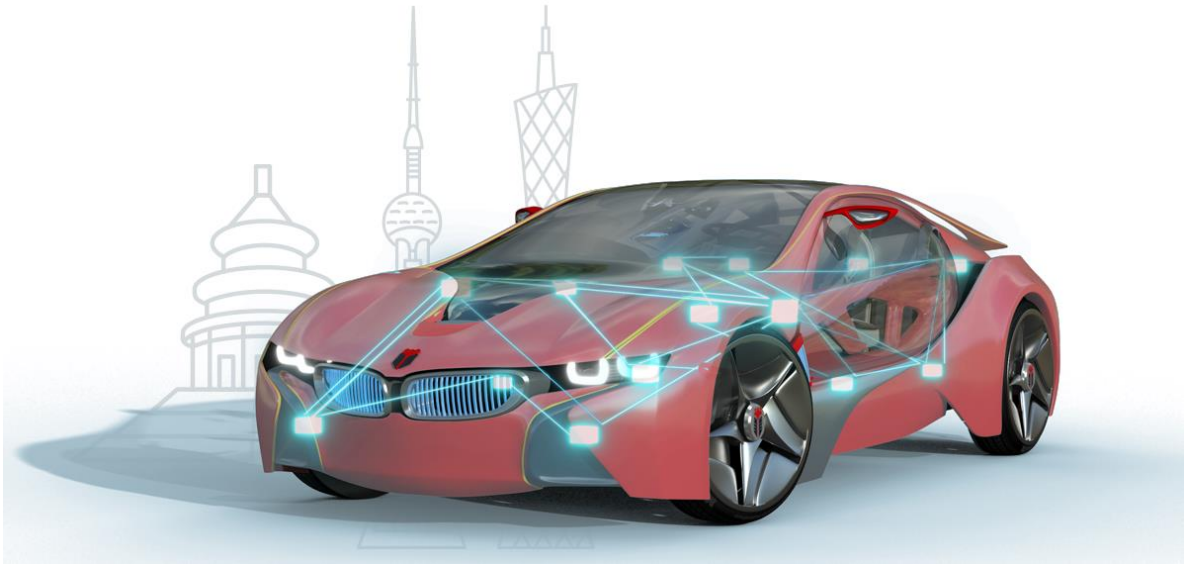
旗芯微 FC7300 EIM ERM 模块的典型应用

Typical Applications of FlagChip FC7300 EIM ERM Module

知从木牛基础软件平台功能安全库

ZC.MuNiu Basic Software Platform SafetyLibrary



旗芯微 FC7300 EIM ERM 模块的典型应用

TYPICAL APPLICATIONS OF FLAGCHIP FC7300 EIM ERM MODULE

知从木牛基础软件平台功能安全库

ZC.MuNiu Basic Software Platform SafetyLibrary

1 概述 OVERVIEW

旗芯微 FC7300 系列是基于多核 Arm® Cortex-M7 的高性能车规级超融合处理器（HPU），满足 ISO 26262 ASIL-D 最高功能安全等级要求。为实现这一严苛的安全目标，FC7300 芯片内部集成了多项关键安全机制，其中 EIM（错误注入模块） 和 ERM（错误报告模块） 扮演着至关重要的角色。

The Flagship FC7300 series is a high performance automotive grade hyper-converged processor (HPU) based on the multi-core Arm® Cortex-M7, which meets the requirements of ISO 26262 ASIL-D, the highest level of functional safety. To achieve this stringent safety goal, the FC7300 chips integrate several key safety mechanisms inside the chip, of which EIM (Error Injection Module) and ERM (Error Reporting Module) play a crucial role.

- EIM：主要以通过模拟 ECC 故障注入，进行上电、周期性的 ECC 功能自检。
  EIM：The main purpose is to perform power-up and periodic self-test of ECC function by simulating ECC fault injection.
- ERM：主要用于响应 ECC 故障，并触发中断进行故障处理，防止故障导致系统功能失效。
  ERM: It is mainly used to respond to ECC faults and trigger interrupts for troubleshooting to prevent the faults from causing the system functions to fail.

这两个模块紧密协作，是 FC7300 达到 ASIL-D 认证、确保汽车动力、底盘、域控制等安全关键应用可靠运行的核心硬件基础

Working closely together, these two modules are the core hardware foundation for the FC7300 to achieve ASIL-D certification and ensure the reliable operation of safety-critical applications such as automotive power, chassis, and domain control.

2 ECC 机制介绍 INTRODUCTION TO THE ECC MECHANISM

对于内存有两种失效模式

1. 非法访问，通常使用 MPU 进行内存管理进行保护

2. 内存损坏，一般通过 ECC 来进行诊断，ECC 的机制可由软件或硬件实现。

There are two failure modes for memory

1. Illegal access, which is usually protected by memory management using MPUs.

2. Memory corruption, which is usually diagnosed by ECC, the mechanism of which can be implemented by software or hardware.

ECC 是芯片用于保障信息安全的一种常用机制，主要是为了避免存储设备中存放的数据因为硬件干扰而被篡改的情况。

ECC is a common mechanism used by chips to secure information, mainly to avoid situations where data stored in storage devices is tampered with due to hardware interference.

2.1 ECC 基本概念 ECC Basic Concepts

ECC 全称 Error Checking and Correcting，属于一种错误检查和纠正算法。在数字电路中，最小的数据单位就是叫“比特（bit）”，也叫数据“位”，“比特”也是内存中的最小单位，它是通过“1”和“0”表示数据高、低电平信号的。空间中的无线电磁干扰、电路噪声会导致内存与 CPU 在进行数据交互的时候发生 比特翻转（“0”变为“1”,“1”变为“0”），典型的 ECC 算法一般可以做到 纠正单比特错误 和 检查 2 比特错误 。

ECC full name Error Checking and Correcting, belongs to an error checking and correcting algorithm. In digital circuits, the smallest unit of data is called “bit (bit)”, also called data ‘bit’, “bit” is also the smallest unit of memory, which is signaled by the “1” and “0” indicate the data high and low level signals. Wireless electromagnetic interference and circuit noise in the space can cause bit flipping (“0” to “1”, ‘1’ to “0”) when the memory interacts with the CPU. "Typical ECC algorithms can correct single-bit errors and check for 2-bit errors.

2.2 ECC 数据的构成 Composition of ECC Data

ECC 需要的 bit 位数量由数据段的比特数决定，8 位的数据需要 5 位的 ECC 位进行校验，数据位每增加一倍，ECC 只增加一位校验位。例如，32 位数据的 ECC 校验位数量位 7 位。

The number of bit bits required for ECC is determined by the number of bits in the data segment. 8-bit data requires 5 ECC bits for checksum, and for every doubling of the data bits,

ECC increases by only one checksum bit. For example, the number of ECC parity bits for 32-bit data is 7 bits.

2.3 ECC 带来的潜伏故障问题 Latent Failure Problems from ECC

潜伏故障是 ECC 机制面临的一个常见问题，在检查冗余校验位时可能会由于一些原因导致无法检测到读取数据的损坏。当硬件无法解决自身带来问题时，就需要软件协助覆盖相应的故障。

Latent faults are a common problem faced by ECC mechanisms, where corruption of read data may not be detected for a number of reasons when checking for redundant parity bits. When the hardware is unable to solve the problem brought about by itself, software is required to assist in covering the corresponding faults.

以 FC7300 芯片为例，该芯片提供了 EIM 外设，专门提供了通过总线翻转数据比特位的功能，它在存储器和 EDC/ECC 功能之间的读取路径中插入了该模块的错误注入通道来达到注入 ECC 故障的手段。软件可以在上电时主动注入 ECC 故障，然后检查 ECC 监控机制的功能是否正常，从而降低潜伏故障率。

Taking the FC7300 chip as an example, the chip provides the EIM peripheral, which is specialized in providing the function of flipping the data bits through the bus, it inserts the error injection channel of the module in the read path between the memory and the EDC/ECC function to achieve the means of injecting ECC faults. The software can actively inject ECC faults at power-up and then check that the ECC monitoring mechanism is functioning correctly, thus reducing the latent fault rate.

3 EIM 与 ERM 模块的应用 APPLICATION OF EIM AND ERM MODULES

3.1 上电检测 Startup Detection

针对 ECC 监控功能的验证通常在上电过程中进行检测，使用 EIM 模块对特定区域（例如 SRAM、FLASH、TCM 等）进行 ECC 故障注入，然后主动读取故障地址触发 ECC 故障，最后再观察故障是否能被 ERM 模块响应（如中断是否触发，故障地址是否被记录等）。

Verification for the ECC monitoring function is usually performed during the power-up process, using the EIM module to inject ECC faults into specific areas (e.g., SRAM, FLASH, TCM, etc.), then actively reading the fault address to trigger an ECC fault, and finally observing whether or not the fault can be responded to by the ERM module (e.g., whether or not an interrupt is triggered, whether or not the fault address is logged, etc.).

3.2 周期监控 Runtime Moniting

当上电自检通过之后，通常认为 ERM 的 ECC 监控功能是可以正常运行的，随后会开启 ERM 对应的监控功能，并开启中断响应用于 ECC 的故障处理（例如主动尝试清除故障、记录 DTC 信息，引发复位等动作）。

After the power-on self-test has passed, the ECC monitoring function of the ERM is usually considered to be operational, and the corresponding monitoring function of the ERM will then be turned on, and interrupt response will be turned on for ECC troubleshooting (e.g., active attempts to clear faults, record DTC information, trigger reset, and other actions).

3.3 下电检测 Shutdown Detection

下电检测的过程与上电检测类似，但需要注意的是，由于下电检测完毕后 ECU 会进入休眠或直接掉电，检测结果将无法通过 RAM 区进行保存。所以下电检测完毕后额外的步骤是通过 FLASH 存储的形式将检测结果保留，在下次上电时读取结果信息，并确认故障区域的故障是否仍然存在。

The process of power-down detection is similar to power-up detection, but it should be noted that since the ECU will go into hibernation or directly power down after the power-down detection is completed, the detection results will not be saved through the RAM area. Therefore, the additional step after the power-down test is to retain the test results in the form of FLASH storage, read the result information at the next power-up, and confirm whether the fault area still exists.

3.4 注意事项 Precautions

在自检过程中会往目标区域注入 ECC 故障，这个会导致这块区域的数据都产生异常。影响场景如下：

During the self-test process, an ECC fault is injected into the target area, which causes anomalies in the data in this area. The impact scenarios are as follows:

- 若多核共享的数据处于故障注入目标区域，则在核 0 做自检的时候，核 1 会由于读取了这个共享数据而触发异常。由于核 1 没有做自检而是在正常执行程序，因此核 1 会认为当前芯片存在了 ECC 故障，从而挂起。

If the data shared by multiple cores is in the fault injection target area, core 1 will trigger an exception due to reading this shared data while core 0 is doing self-test. Since core 1 is not doing self-test but executing the program normally, core 1 will think that there is an ECC fault on the current chip and hang.

- 若堆栈数据存放在了故障注入目标区域，则在注入故障后进行一次函数的调用会导致读写堆栈触发异常，并且由于堆栈数据故障，这还会导致异常调转错误，从而程序跑飞。

If the stack data is stored in the target area of the fault injection, a function call after the fault is injected will result in an exception triggered by reading or writing to the stack, and due to the faulty stack data, this will also result in an exception transfer error, which will result in the program running away.

- 若中断向量表存放在了故障注入目标区域，则在注入故障后，若触发了中断（例如定时中断、CAN 中断等），由于向量表数据异常则会导致最终中断跳转异常，从而程序跑飞。

If the interrupt vector table is stored in the fault injection target area, if an interrupt (e.g., timing interrupt, CAN interrupt, etc.) is triggered after the fault is injected, an exception in the vector table data will result in an abnormal final interrupt jump, and the program will fly.

因此在使用 EIM 进行故障注入测试的时候，应当小心谨慎，需要考虑的故障注入涉及到的各个方面。例如将避免将多核共享的变量防止在 ECC 故障注入影响的范围内；如果堆栈放置在 TCM 区域的话，那么在执行 TCM 相关测试时加入堆栈的切换工作；在做 FLASH 自检的时候，为了避免代码数据异常，应当考虑将代码放置在 RAM 中执行。

Therefore, when using EIM for fault injection testing, care should be taken to consider all aspects of fault injection. For example, the variables shared by multiple cores should be prevented from being affected by the ECC fault injection; if the stack is placed in the TCM area, the stack switching work should be added when performing TCM related tests; when doing the FLASH self-test, in order to avoid code data anomalies, the code should be considered to be placed in the RAM for execution.

4 知从木牛介绍 INTRODUCTION TO ZC MUNIU

知从木牛配置工具基于最新 ARTOP 架构，支持最新 AUTOSAR R21-11 标准所提供的基础平台上，根据 AUTOSAR 开发方法中定义的 ECU 配置步骤，实现了从配置、验证到代码生成的 ECU 配置全流程的功能。主要优势可以总结为以下几个方面：配置、验证和代码生成全流程功能的实现，完整的实现了 AUTOSAR 开发方法中 ECU 配置阶段的开发要求。

Based on the latest ARTOP architecture and the basic platform provided by the latest AUTOSAR R21-11 standard, the MNU Configuration Tool realizes the whole process of ECU configuration from configuration, validation to code generation according to the ECU configuration steps defined in the AUTOSAR development methodology. The main advantages can be summarized in the following aspects: the realization of the whole process of configuration, verification and code generation has completely realized the development requirements of the ECU configuration stage in the AUTOSAR development method.

在 ZC MuNiu SafetyLibrary 软件中实现了 EimErmTst 模块，该模块运用 Eim 和 Erm 外设实现了上电自检功能，下面将简单介绍一下使用 ZC MuNiu 进行 EimErmTst 模块配置的过程：

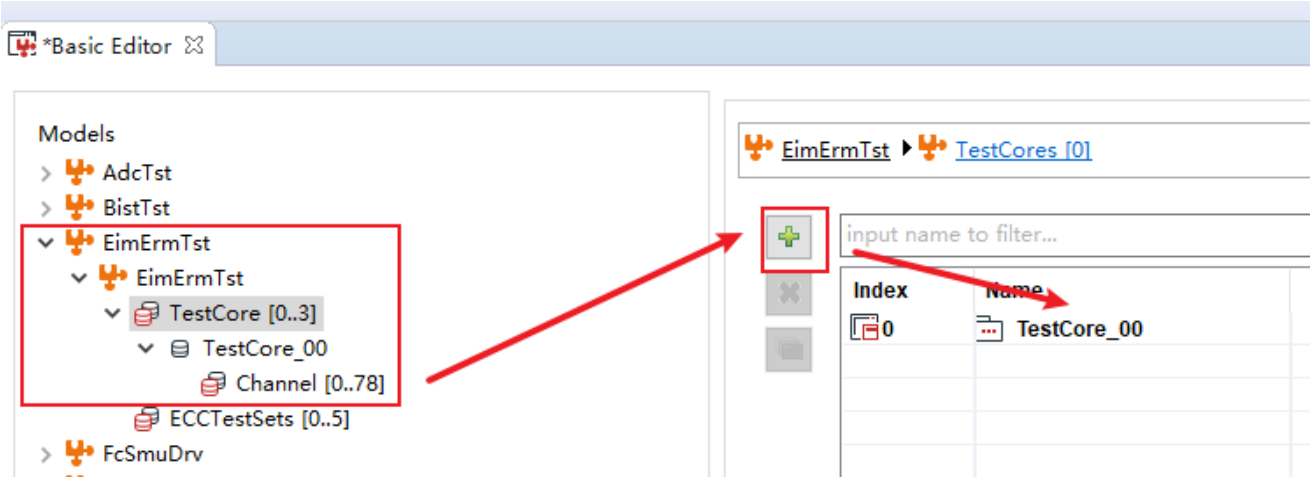
The EimErmTst module is implemented in the ZC MuNiu SafetyLibrary software, which utilizes the Eim and Erm peripherals to achieve the power-on self-test function. The following will briefly describe the process of configuring the EimErmTst module using ZC MuNiu:

4.1 EimErmTst 配置简介 Introduction to EimErmTst Configuration

4.1.1 添加测试集合 ADDING TEST SET

通过在 TestCore 界面可以为指定的 CPU 核添加测试集合。

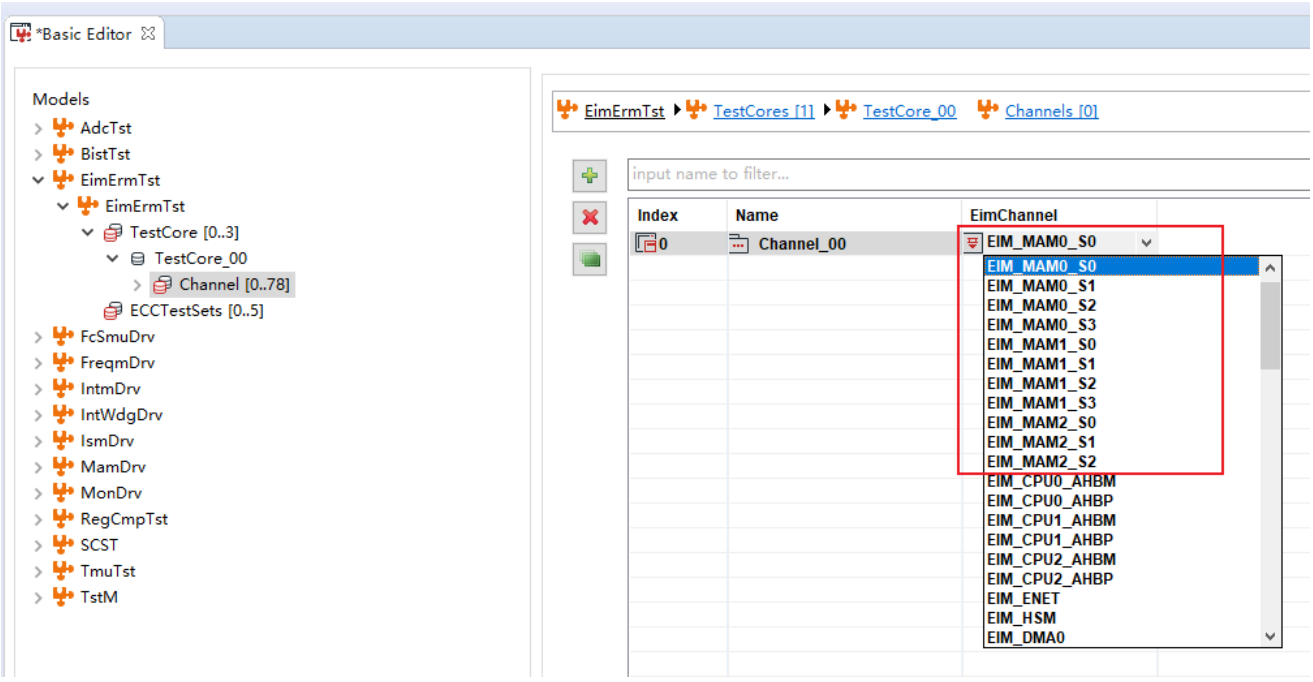
Test collections can be added for specific CPU cores through the TestCore interface.



4.1.2 配置 EIM 测试通道 CONFIGURING THE EIM TEST CHANNEL

在对应的 Channel 配置页面下可以添加多个 EIM 测试通道。

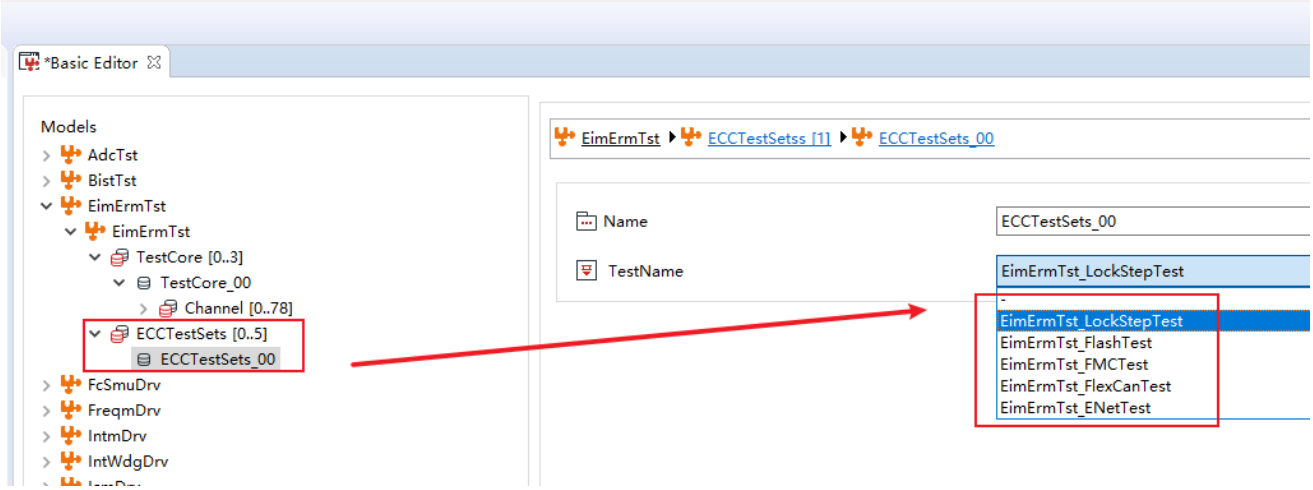
Multiple EIM test channels can be added under the corresponding Channel configuration page.



4.1.3 额外安全机制检测功能配置 CONFIGURATION OF ADDITIONAL SECURITY MECHANISM DETECTION FUNCTIONS

知从木牛功能安全具有高度的可扩展性，针对客户提出的额外安全机制需求，知从可以针对性地制作检测功能，并提供相应的配置项。在 ECCTestSets 页面中选择需要执行的检测项。

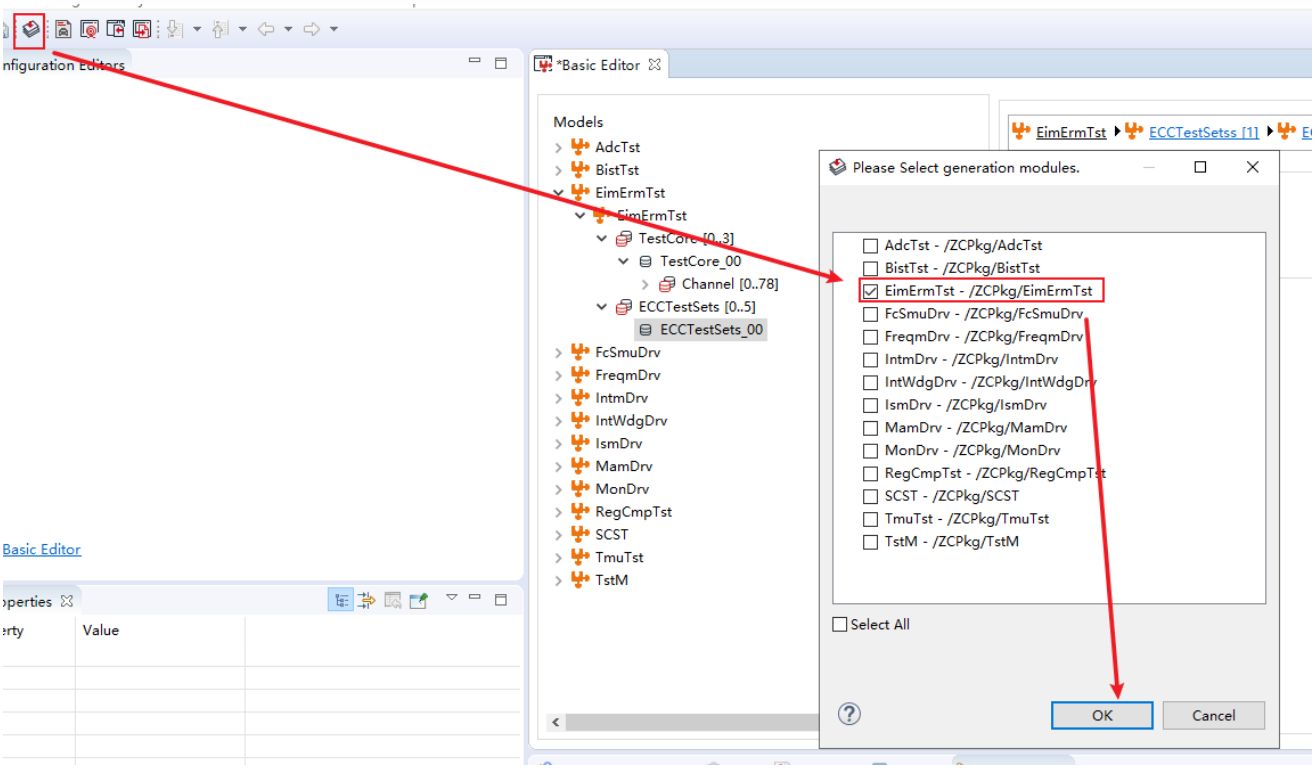
ZC MuNiu SafetyLibrary software is highly scalable. In response to the needs of customers for additional safety mechanisms, ZC can create targeted testing functions and provide the corresponding configuration items. In the ECCTestSets page, select the tests that need to be performed.



4.1.4 生成配置代码 GENERATING CONFIGURATION CODE

在配置完所有功能后即可生成 EimErmTst 相关的配置代码。

The EimErmTst related configuration code can be generated after all the functions are configured.

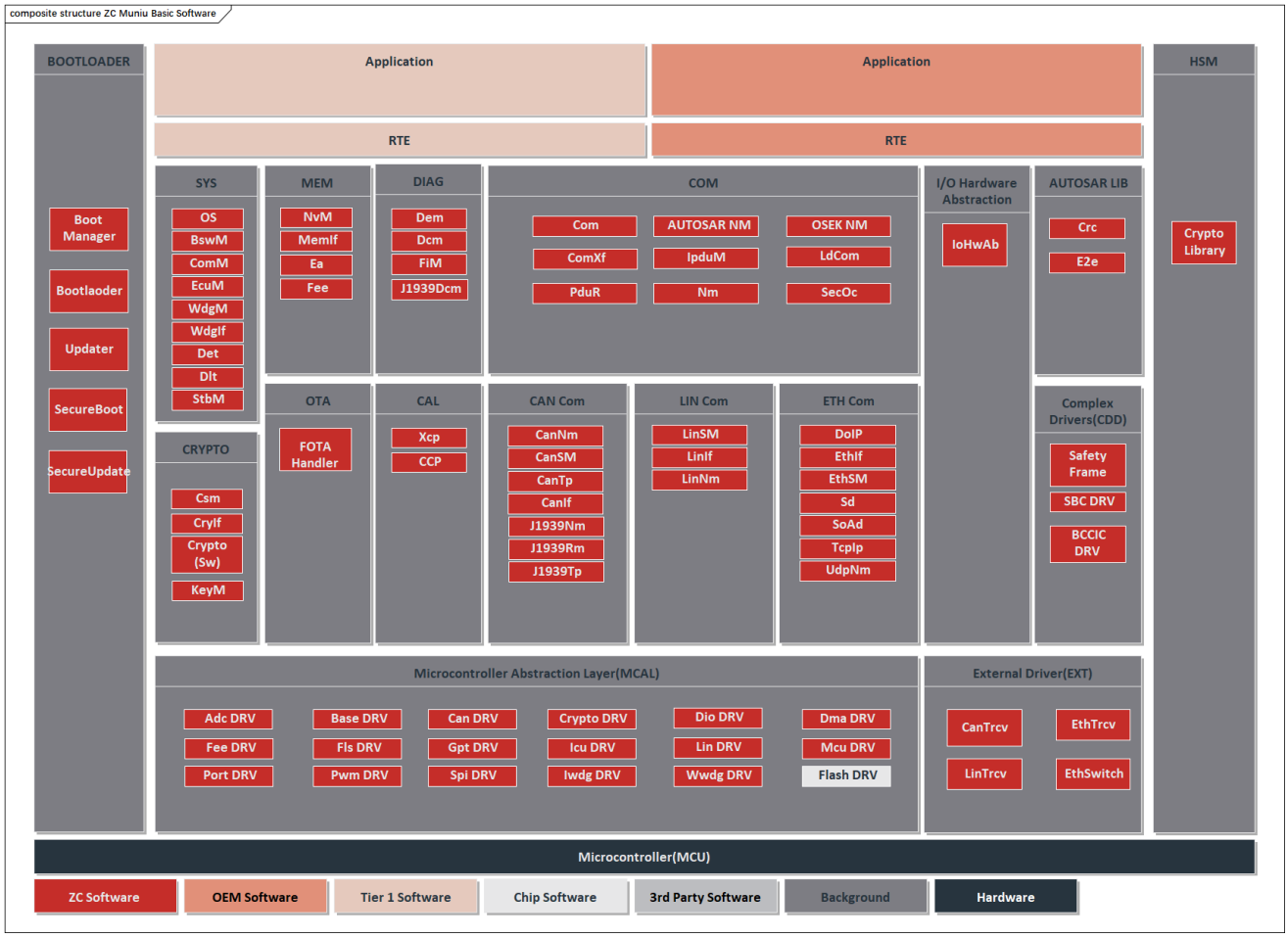


4.2 应用领域 Areas of application

知从木牛配置工具，给 ECU 控制器软件开发提供友好的人机界面。可以支持标准的 AUTOSAR 基础软件代码模块的配置以及复杂驱动的配置界面开发。目前主要应用于如下场景:

The ZC MuNiu configuration tool provides a user-friendly HMI for ECU controller software development. It supports the configuration of standard AUTOSAR basic software code modules

as well as the development of configuration interfaces for complex drivers. Currently, it is mainly used in the following scenarios:



- 知从木牛基础软件平台标准 AUTOSAR 模块配置
- 知从木牛基础软件平台复杂驱动模块配置
  - SAFETY FRAME
  - CRYPTO LIBRARY
  - BCCIC
  - SBC
- 同芯片企业合作，提供 MCU MCAL 的配置工具
- ZC MuNiu Basic Software Platform Standard AUTOSAR Module Configuration
- ZC MuNiu Basic Software Platform Complex Driver Module Configuration
  - SAFETY FRAME
  - CRYPTO LIBRARY
  - BCCIC
  - SBC
- Collaboration with chip companies to provide configuration tools for MCU MCALs