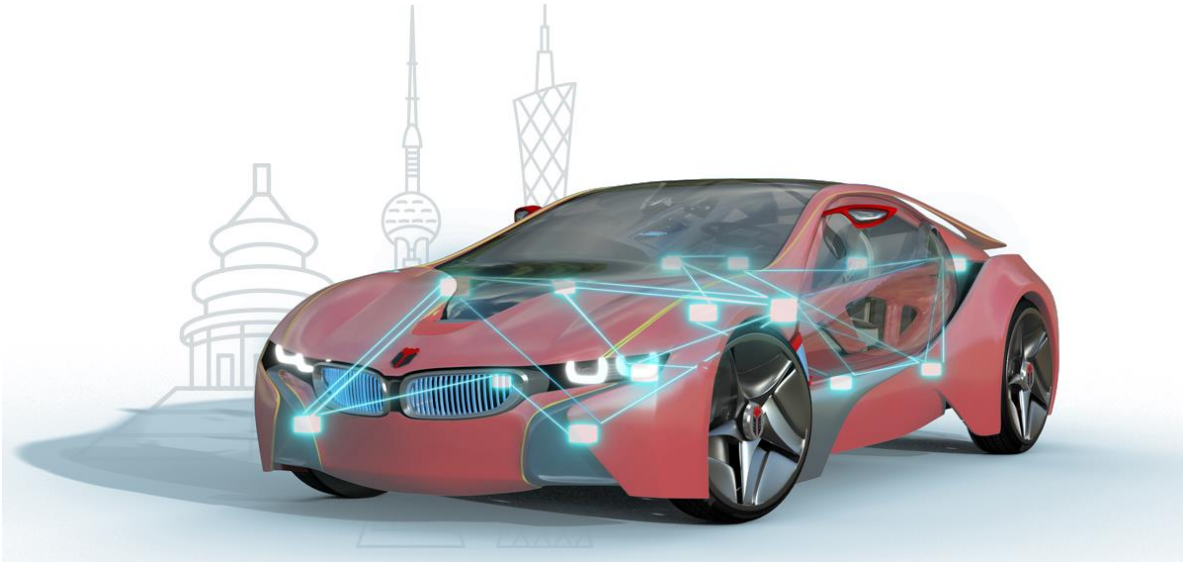
知从木牛-恩智浦 FS26 功能安全产品介绍

ZC.MUNIU FUNCTIONAL SAFETY PRODUCT MANUAL BASED ON NXP FS26

知从木牛功能安全库

ZC MuNiu Functional Safety Library



知从木牛-恩智浦 FS26 功能安全产品介绍

ZC.MUNIU FUNCTIONAL SAFETY PRODUCT MANUAL BASED ON NXP FS26

知从木牛功能安全库

ZC MuNiu Functional Safety Library

1 概述 OVERVIEW

知从木牛功能安全 SBC 系列软件旨在打造知从科技自主研发的满足客户功能安全要求的 System Basis Chip（SBC）平台化软件产品，适配不同厂家的 SBC 芯片。本手册说明了基于恩智浦 FS26 系列 SBC 实现的功能安全应用方案、软件架构等内容。本软件产品可帮助系统工程师和软件工程师能够快速地应用到客户产品中，满足功能安全需求。

The ZC.MuNiu Functional Safety SBC Series Software is designed to establish ZC Technology's independently developed System Basis Chip (SBC) platform software products that meet customer functional safety requirements, compatible with SBC chips from various manufacturers. This manual details the functional safety application solutions and software architecture based on NXP's FS26 series SBC. This software product enables system engineers and software engineers to rapidly integrate it into customer products, fulfilling functional safety requirements.

NXP FS26 是一款集成了多个开关式稳压器（Switchers）、线性稳压器（LDOs）、高边驱动（HSDs）、低边驱动（LSDs）以及完整安全功能的车规级电源管理芯片。其设计符合 ISO 26262 ASIL D 等级标准，是面向下一代汽车域控制器、ADAS、网关等关键应用的理想选择。

The NXP FS26 is an automotive-grade power management IC that integrates multiple switchers, LDOs, high-side drivers (HSDs), low-side drivers (LSDs), and comprehensive safety features. Designed to meet the ISO 26262 ASIL D standard, it is an ideal choice for next-generation critical applications such as automotive domain controllers, ADAS systems, and gateways.

本产品实现了的 FS26 芯片软件驱动功能包含：

- 多路电源输出管理；

- SBC 状态机控制、低功耗控制与唤醒管理;
- 输出电压诊断管理;
- MCU 与 SBC 的 SPI 通信处理;
- SBC 片内 ABIST/LBIST 自检管理;
- 看门狗管理;
- MCU 错误监控管理;
- SBC 片外安全关断路径处理。

The FS26 chip software driver functionality implemented in this product includes:

- Multi-channel power supply output management;
- SBC state machine control, low-power control, and wake-up management;
- Output voltage diagnostic management;
- SPI communication processing between MCU and SBC;
- SBC on-chip ABIST/LBIST self-test management;
- Watchdog management;
- MCU error monitoring management;
- SBC off-chip secure shutdown path processing.

2 应用领域 APPLICATION FIELD

SafetyLibrary 可应用于有功能安全等级需求的控制器。例如:

The SafetyLibrary can be applied to controllers that require functional safety levels.

For example:

- 电机控制器
  Motor Controller
- 电池管理系统(BMS)
  Battery Management System

- 底盘系统应用
  Chassis System Applications
- 电气稳定控制(ESC)
  Electronic Stability Control
- 电动助力转向(EPS)
  Electric Power Steering
- 安全气囊和传感器集成应用
  Chassis Domain Line Control System Applications
- 雷达的应用
  Radar Applications

3 工具链支持 TOOLCHAIN SUPPORT

配置环境 Configuration Environment	
Hardware (Chip)	S32K3-FS26
Compilers Supported	S32 Design Studio for ARM(2018.R1) 、IAR v8.40.1
Debugger	Lauterbach (Trace32 R.2018.02) Isystem (IC5700)
Configuration Tools	知从木牛.配置工具V5.1.0
Configuration Environment	Win10 64bit

4 开发背景 DEVELOPMENT BACKGROUND

目前，汽车上的电子电气架构越来越复杂，对汽车电子的安全性要求也越来越高，为了满足汽车的安全性需求，汽车功能安全越来越受到重视。提到功能安全，大家首先想到的是功能安全的标准 ISO26262。其中，ISO 26262-5(2018) Clause 8 中介绍了 2 个度量：Single-point fault metric(单点故障度量)和 Latent-fault metric(潜伏故障度量)。根据不同的 ASIL 等级要求，单点故障度量和潜伏故障度量需要达到相应的等级。

Currently, the electronic and electrical architecture of automobiles is becoming increasingly complex, and the safety requirements for automotive electronics are also rising. To meet the safety requirements of automobiles, functional safety is gaining more attention. When it comes to functional safety, the first thing that comes to mind is the functional safety standard ISO 26262. In particular, ISO 26262-5(2018) Clause 8 introduces two metrics: Single-point fault metric (single-point fault metric) and Latent-fault metric (latent fault metric). Depending on the required ASIL level, the single-point fault metric and latent fault metric must meet the corresponding levels.

对于微控制器(MCU，以下简称 MCU)，在电子电气系统中，作为 SEooC(safety element out of context)进行设计开发。MCU 为了满足以上提到的 2 个度量要求，需要实现相应的安全机制。而安全机制可以分配到硬件和软件模块中。MCU 的 SafLib 就是实现分配到软件上的安全机制。

For microcontrollers (MCU, referred to as MCU below), within the electronic and electrical system, they are designed and developed as SEooC (safety element out of context). To meet the aforementioned metric requirements, MCUs need to implement corresponding safety

mechanisms. These safety mechanisms can be allocated to both hardware and software modules. The SafLib for MCUs is the implementation of safety mechanisms allocated to software.

	ASIL B	ASIL C	ASIL D
Single-point fault metric	≥90 %	≥97 %	≥99 %

	ASIL B	ASIL C	ASIL D
Latent-fault metric	≥60 %	≥80 %	≥90 %

5 功能描述 FUNCTIONAL DESCRIPTION

5.1 产品特点 Product Feature

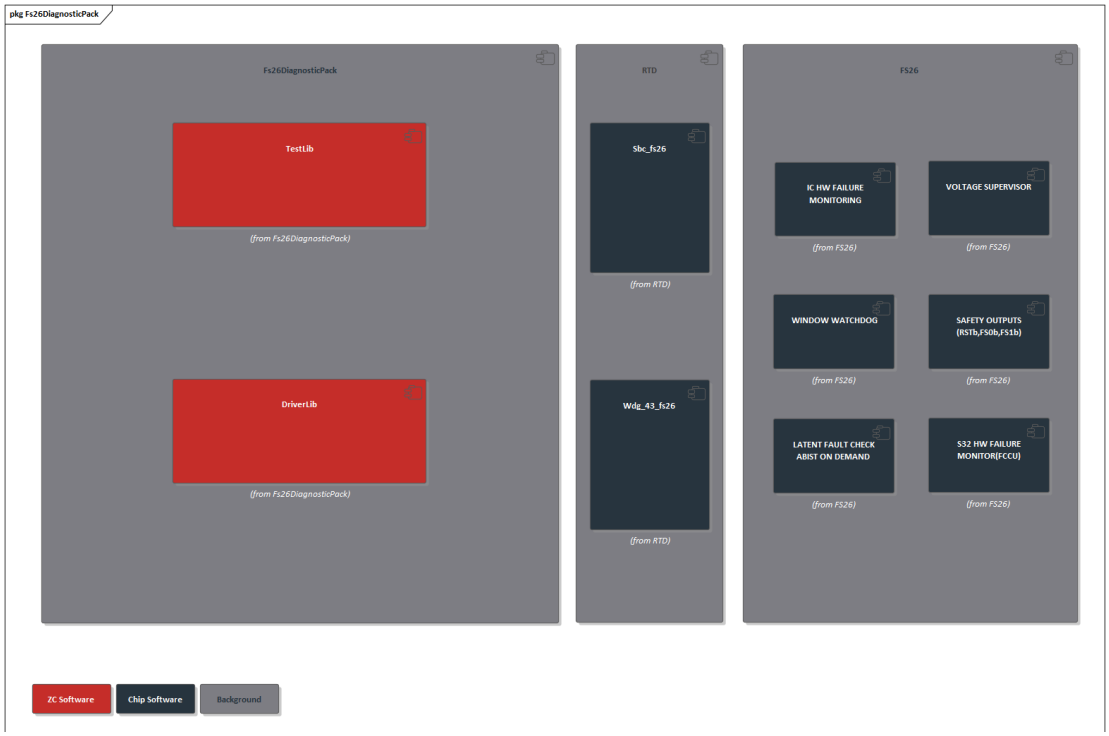


图 1：SAFETYLIBRARY-FS26 软件架构框图

FIGURE 1: SAFETYLIBRARY -FS26 SOFTWARE ARCHITECTURE DIAGRAM

- 可作为复杂驱动集成到 AUTOSAR 中

  Can be integrated as a complex driver into AUTOSAR .

- 可集成到非 AUTOSAR 软件架构中，灵活适配

  Can be integrated into non-AUTOSAR software architectures.

- 高安全性：搭配 NXP-SAF 软件可实现高达 ASIL-D 需求

  High Safety: It can achieve up to ASIL-D requirements when paired with SafLib.

5.2 SPI 周期性故障监控与响应 SPI Periodic Failure Monitoring and Response

本功能实现了 MCU 对 FS26 的主动、周期性健康检查。MCU 以固定时间基（如 10ms）通过 SPI 接口读取 FS26 的故障状态寄存器集，从而实时侦测电压监控异常、过热、看门狗错误、通信错误等故障，并根据故障严重级别立即执行相应的安全措施。

This feature enables the MCU to perform active, periodic health checks on the FS26. The MCU reads the FS26's fault status register set via the SPI interface at fixed time intervals (e.g., 10ms), thereby detecting faults such as voltage monitoring anomalies, overheating, watchdog

errors, and communication failures in real time. Based on the severity level of the fault, it immediately executes corresponding safety measures.

5.3 软件实施流程 Software Implementation Process

1. 初始化阶段：配置 FS26 的看门狗模式、故障报警引脚(FAILB)、各电源通道使能等。

   Initialization Phase: Configure FS26's watchdog mode, fault alarm pin (FAILB), and enable each power channel.

2. 定时任务触发：由 MCU 的实时操作系统（RTOS）定时器或中断服务程序（ISR）周期性触发监控任务。

   Scheduled Task Triggering: Periodically initiate monitoring tasks via the MCU's real-time operating system (RTOS) timer or interrupt service routine (ISR).

3. SPI 通信与读取：MCU 发起 SPI 读取序列，获取 FAIL_STAT_1, FAIL_STAT_2, DIAG_STAT 等关键寄存器值。

   SPI Communication and Reading: The MCU initiates an SPI read sequence to acquire critical register values such as FAIL_STAT_1, FAIL_STAT_2, and DIAG_STAT.

4. 数据校验与解析：对 SPI 数据进行 CRC 校验，确保通信完整性，随后解析寄存器标志位，确定故障源。

   Data Verification and Parsing: Perform CRC checks on SPI data to ensure communication integrity, then parse register flags to identify the fault source.

5. 故障响应：根据预设的故障响应策略，执行相应操作。

   Fault Response: Execute corresponding actions based on predefined fault response strategies.

5.4 故障分级响应策略 Fault Graded Response Strategy

故障级别 Fault Level	故障示例 Fault Examples	响应策略 Response Strategy
Level 1	瞬时性 UV/OV 事件、通信 CRC 错误 Transient UV/OV events, communication CRC errors	记录 DTC Record DTC
Level 2	长期/永久性故障 Long-term/Permanent Failure	记录 DTC、系统功能降级 Record DTCs, System Function Degradation

Level 3	关键电源失效，内核自检故障 Critical power supply failure, kernel self-test error	记录 DTC、停止喂狗 Record DTC, stop feeding the dog

注：停止喂狗（Stop Feeding WDG）机制：这是最高级别的软件安全响应。当 MCU 侦测到无法处理的致命故障时，将主动停止刷新 FS26 的看门狗。FS26 将在超时后自动执行其硬件安全响应（如触发芯片复位），确保系统强制进入已知的安全状态，这是实现安全机制独立性的关键。

Note: Stop Feeding WDG Mechanism: This represents the highest level of software safety response. When the MCU detects an unhandled critical fault, it proactively halts the refresh of the FS26 watchdog timer. Upon timeout, the FS26 automatically executes its hardware safety response (e.g., triggering a chip reset), ensuring the system forcibly transitions to a known safe state. This is crucial for achieving independence of the safety mechanism.

5.5 ABIST 自检功能驱动与诊断 ABIST Self-Test Function Driver and Diagnostics

5.5.1 ABIST 功能简介 ABIST FUNCTION OVERVIEW

ABIST（Analog Built-In Self-Test）是 FS26 内部的模拟模块自检功能，用于检测芯片内部的潜在缺陷。

ABIST (Analog Built-In Self-Test) is an internal self-test function for the FS26's analog module, designed to detect potential defects within the chip.

- ABIST1：快速自检，耗时短，覆盖核心模块。

  ABIST1: Rapid self-test with short duration, covering core modules.

- ABIST2（ABIST On Demand）：完整自检，耗时长，覆盖全部模块。

  ABIST2 (ABIST On Demand): Comprehensive self-test with extended duration, covering all modules.

5.5.2 自检触发策略 SELF-TEST TRIGGERING STRATEGY

- 启动自检（Ignition On）：车辆上电后，在执行主功能前，进行一次全面的 ABIST2 自检，确保 FS26 初始状态正常。

  Ignition On Self-Test: Upon vehicle power-up, a comprehensive ABIST2 self-test is performed before executing primary functions to ensure the FS26 is in a normal initial state.

- 周期性自检：在车辆运行期间，每隔一定时间（如 24 小时）或在特定条件下（如车辆静止）触发 ABIST1 快速自检，实现运行中的在线诊断。

Periodic Self-Test: During vehicle operation, an ABIST1 rapid self-test is triggered at regular intervals (e.g., every 24 hours) or under specific conditions (e.g., when the vehicle is stationary) to enable online diagnostics while the system is running.

5.5.3 自检执行与结果处理流程 SELF-TEST EXECUTION AND RESULT HANDLING PROCESS

- 请求自检：MCU 通过 SPI 向 ABIST 控制寄存器写入特定命令序列启动自检。

Requesting Self-Test: The MCU initiates the self-test by writing a specific command sequence to the ABIST control register via SPI.

- 等待完成：MCU 延迟等待 t_ABIST1 或 t_ABIST2 时间，或轮询 DIAG_STAT.ABIST_DONE 状态位。

Waiting for Completion: The MCU delays for either t_ABIST1 or t_ABIST2 duration, or polls the DIAG_STAT.ABIST_DONE status bit.

- 验证结果：自检完成后，读取 DIAG_STAT.ABIST_PASS 位。

Verify Result: After self-test completion, read the DIAG_STAT.ABIST_PASS bit.

- 结果处理：

Result Handling:

  - 通过（PASS）：系统继续正常运行。

    Pass: System continues normal operation.

  - 失败（FAIL）：此为严重硬件故障指示。MCU 将记录最高优先级的 DTC，并立即执行 Level 3 响应（停止喂狗），引导系统进入安全状态。

Fail: Indicates a critical hardware fault. The MCU logs the highest-priority DTC and immediately executes a Level 3 response (stop feeding the dog), transitioning the system to a safe state.

5.6 看门狗功能介绍 Watchdog Function Overview

为什么看门狗是必须的？原因是运行在硬件世界中的软件会受到各种外界因素的影响，比如：汽车上使用的诸多零部件，鉴于汽车环境的恶劣，各类 ECU 中的软件均有可能遭受如外部电磁干扰，高温等环境因素的影响，从而导致程序“跑飞”或者“死机”现象，此时如果有看门狗的存在，便可以主动触发系统复位机制保证能够再次正常使用。

Why is the watchdog necessary? The reason is that the software running in the hardware world will be affected by a variety of external factors, such as: many parts and components used

in the car, in view of the harsh environment of the car, all types of ECU software may be subjected to such as external electromagnetic interference, high temperature and other environmental factors, which leads to the program “fly” or “dead” phenomenon, at this time, if the existence of the watchdog can actively trigger the system reset mechanism to ensure that it can be used normally again, rather than just watch being watched. “crash” phenomenon, at this time if the existence of the watchdog, you can take the initiative to trigger the system reset mechanism to ensure that it can be used normally again.

看门狗在使用时软件必须在规定的时间间隔内向其发送特定信号，这个行为被形象地称为"喂狗"，以免 watchdog 记时超时引发系统重启。硬件看门狗一般是一块专门的芯片，它通过监控电路的电压和电流等指标来判断系统是否正常运行。在下图是 NXP 的一款 SBC 芯片，该芯片就提供了看门狗功能，在使用时需要 MCU 通过 SPI 进行看门狗初始化及喂狗操作:

Watchdog in use when the software must be in the specified time interval to send a specific signal, this behavior is figuratively known as “feed the dog”, so as to avoid watchdog timeout triggered by the system reboot. Hardware watchdog is generally a specialized chip, which monitors the circuit voltage and current and other indicators to determine whether the system is operating normally. In the following figure is a NXP SBC chip, the chip provides the watchdog function, when using the MCU through the SPI watchdog initialization and feed the dog operation:

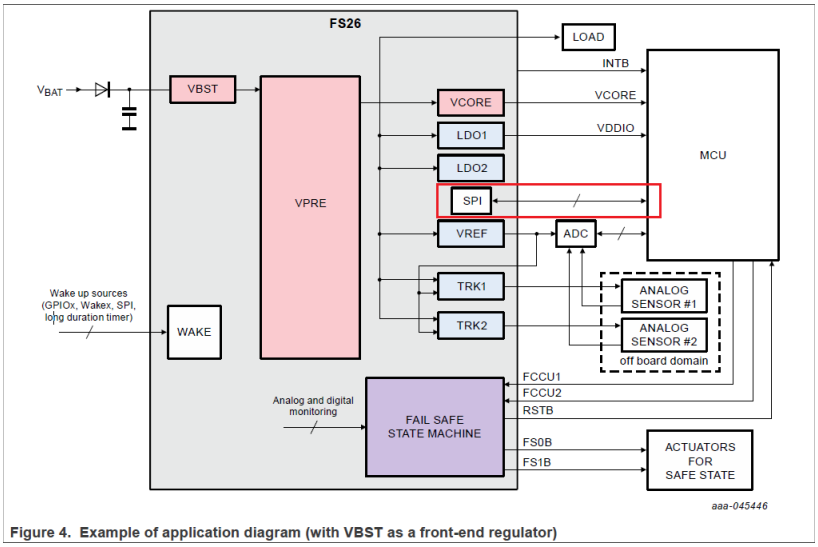


Figure 4. Example of application diagram (with VBST as a front-end regulator)

图 2：外狗模块框图

FIGURE 2: EXTERNAL WDG MODULE

在汽车电子系统中，功能安全（Functional Safety）的核心目标是通过系统化的设计和安全机制，防止因电子电气系统故障导致的人身伤害或财产损失。WdgM（Watchdog Manager） 作为 AUTOSAR 标准中的关键安全模块，其设计严格遵循 ISO 26262 标准，并通过多层次的监控、容错和恢复机制来满足不同 ASIL（Automotive Safety Integrity Level）等级的要求。

In automotive electronic systems, the core objective of Functional Safety is to prevent personal injury or property damage due to electrical and electronic system failures through systematic design and safety mechanisms. WdgM (Watchdog Manager), as a key safety module in the AUTOSAR standard, is designed in strict compliance with the ISO 26262 standard and meets the requirements of different ASIL (Automotive Safety Integrity Level) through multi-layered monitoring, fault tolerance and recovery mechanisms. The WdgM (Watchdog Manager), as a key safety module in the AUTOSAR standard, is designed in strict compliance with ISO 26262 and meets the requirements of different ASIL (Automotive Safety Integrity Level) through multi-level monitoring, fault tolerance and recovery mechanisms.

WdgM 里面主要部分是 Supervised Entity(SE) 和 CheckPoint(CP) 。

The main parts of WdgM are the Supervised Entity (SE) and CheckPoint (CP).

- SE ： 每一个 SE 可以有 Alive 、 deadline 、 Logical 三种形式的监督方式【 Logical Supervision 分为两种，同一个 SE 内部的程序流监督、不同 SE 内部的程序流监督】。一个 SE 可以是一个算法、一个函数、一个任务。

  SE: Each SE can have three forms of supervision: Alive, deadline, and Logical [Logical Supervision is divided into two types, program flow supervision within the same SE and program flow supervision within different SEs]. A SE can be an algorithm, a function, a task.

- Check Point ： 用于作为区分不同的监控方式，可以属于下面的一种或多种监控方式的类型

  Check Point : Used as a way to differentiate between different monitoring methods and can belong to one or more of the following types of monitoring methods.

6.1 周期监控 Alive Supervision

程序流监控：外部看门狗实现 WdgM 模块的 Alive Supervision 监控机制，用于监控周期性的 Task；WdgM 模块通过计算程序运行时检查点出现的个数，然后与期望值(配置信息)进行比较，如果超出容差范围，即视为违反程序流程, WdgM 模块将会记录为检查点故障。

Program Flow Monitoring: The external watchdog implements the WdgM module's Alive Supervision monitoring mechanism to monitor periodic Tasks; the WdgM module calculates the number of checkpoints that occur while the program is running and compares it with the expected value (configuration information), if it is out of tolerance, it is considered as a violation of the program flow, and will be recorded as a Checkpoint Fault by the WdgM module. The WdgM module records this as a checkpoint fault.

以 AEB 模块 Task 的 Alive Supervision 举例：

Example of Alive Supervision for the AEB module Task:

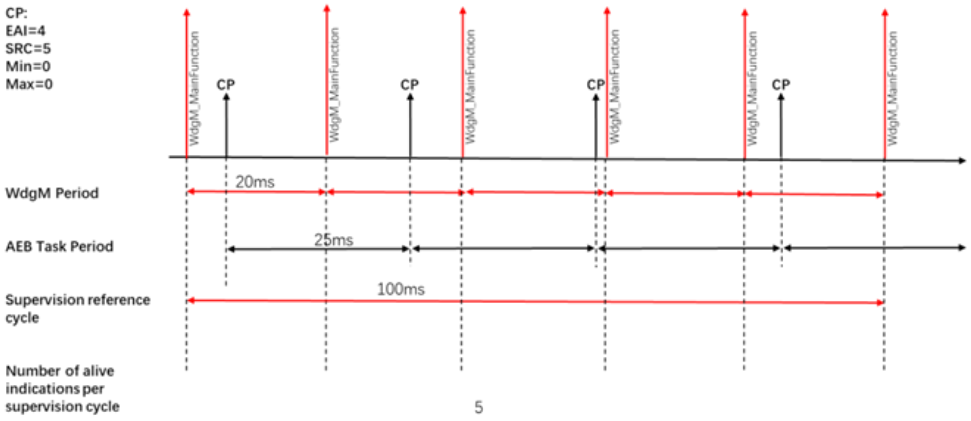


图 3：ALIVE 监控示例

FIGURE 3: ALIVE SUPERVISION

WdgM 的调用周期为 20ms，AEB 模块 Task 的周期为 25ms，如图 6.21 所示, 配置监控周期为 100ms，即：在 5 个 WdgM_MainFunction 区间，CheckPoint 到达数量为 3-5 个为正常。 如果发生的次数不在 3-5 之间 ,WdgM 模块将认为检测到 AEB 模块 Task 的周期异常，同时通过图 6.20 所示的喂狗数据流传输不喂狗请求给 PMIC 模块，PMIC 模块停止喂狗, 等待看门狗喂狗超时（超时时间 64ms*3）后，触发复位。

The call cycle of WdgM is 20ms and the cycle of AEB module Task is 25ms, as shown in Figure 6.21, the monitoring cycle is configured to be 100ms, i.e., the number of CheckPoint arrivals in the interval of 5 WdgM_MainFunction is 3-5 as normal. If the number of occurrences is not between 3-5, the WdgM module will consider that the cycle of AEB module Task is

detected abnormally, and at the same time, it will transmit the request of not feeding the dog to the PMIC module through the dog feeder data stream shown in Fig. 6.20, and the PMIC module will stop feeding the dog, wait for the timeout of watchdog dog feeder (timeout time of 64ms*3), and then trigger the reset.

6.2 截止时间监控 Deadline Supervision

Deadline Supervision 需要关注某段程序运行的时间，过长或过短都说明程序执行异常。抽象为监控两个 Check Point 之间运行的时间 。 具体算法如下： 在 WdgM 中配置 Deadline Supervision 的起始 Check Point、结束 Check Point、最小时间门限和最大时间门限 。 在运行到起始 Check Point 时启动 Deadline Supervision， 获取系统时间， 在运行到结束 Check Point 时， 获取系统时间， 计算运行时间是否配置合理范围内。

注意： 两个 CP 属于相同的 SE，两个 CP 不能相同 。

Deadline Supervision is concerned with how long a program is running, whether it is too long or too short indicates that the program is executing abnormally. The abstraction is to monitor the running time between two Check Points. The algorithm is as follows: Configure the Start Check Point, End Check Point, Minimum Time Threshold and Maximum Time Threshold of Deadline Supervision in WdgM. Start Deadline Supervision at the Start Check Point, get the system time, and at the End Check Point, get the system time, and calculate whether the runtime is within the reasonable range.

Note: Two CPs belong to the same SE and two CPs cannot be the same.

6.3 逻辑监控 Logical Supervision

Logical Supervision 主要用于监控应用程序的运行顺序是否正确，包括各个 SE 本地的运行路径的检查，和 SE 之间的全局路径检查。

Logical Supervision is mainly used to monitor whether the running order of the application is correct or not, including the checking of the local running paths of each SE, and the checking of the global paths between SEs.

比如程序按照： 1 -> 2 -> 3 -> 4 的顺序运行。程序在运行至 1 时，检查是否为第一个应该运行到的点，运行至 2 时则检查前一个到达的 check point 点是否为 1 。依次检查相邻两个点之间的实际运行顺序与配置之间的运行顺序是否匹配。当程序以 1 -> 2 -> 4 的顺序运行时，在运行至 4 处会检查到程序出错，更改 Logical Supervison 的 SE 状态。

Let's say the program runs in the order: 1 -> 2 -> 3 -> 4. When the program reaches 1, it checks to see if it is the first point it should run to, and when it reaches 2, it checks to see if the previous check point reached is 1. In turn, the program checks to see if the actual running order between the two neighboring points matches the running order between the

configurations. When the program is run in the order of 1 -> 2 -> 4, a program error is checked at run 4, and the SE status of the Logical Supervison is changed.

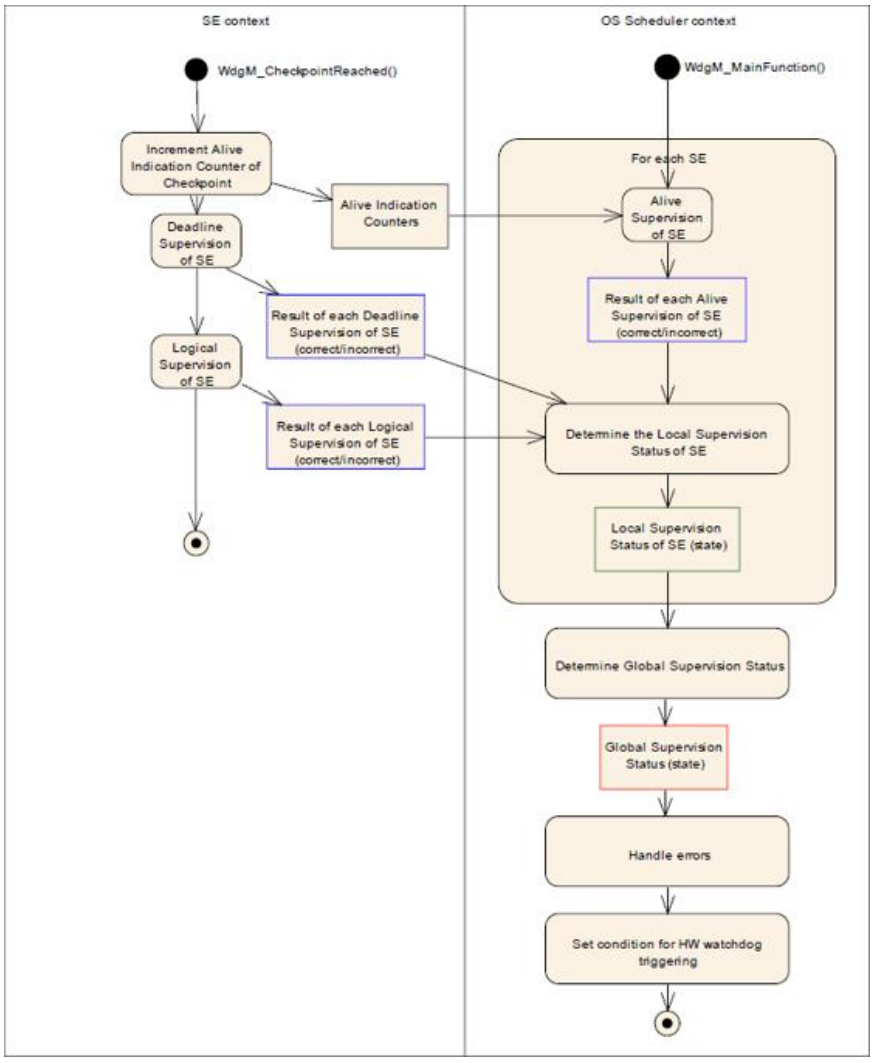


图 4：WDGM 监控概述

FIGURE 4: OVERVIEW OF WATCHDOG MANAGER SUPERVISION

7 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。

ZC MuNiu configuration tool is based on the latest ARTOP architecture, supports the latest AUTOSAR R21-11 standard provided by the basic platform, according to the ECU configuration steps defined in the AUTOSAR development methodology, realizes the ECU configuration from the configuration, validation to the code generation of the whole process of

the function. 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 methodology.

下面将简单介绍一下使用 ZC MuNiu 进行 WdgM 模块配置的过程:

The following will briefly describe the process of WdgM module configuration using ZC MuNiu:

7.1 WdgM 配置简介 Introduction to WdgM Configuration

7.1.1 添加 SE 监控实例 ADDING SE MONITORING INSTANCES

通过在 WdgMMode 界面可以添加 Alive 监控相关的监控实例。

You can add monitoring instances related to Alive monitoring through the WdgMMode interface.

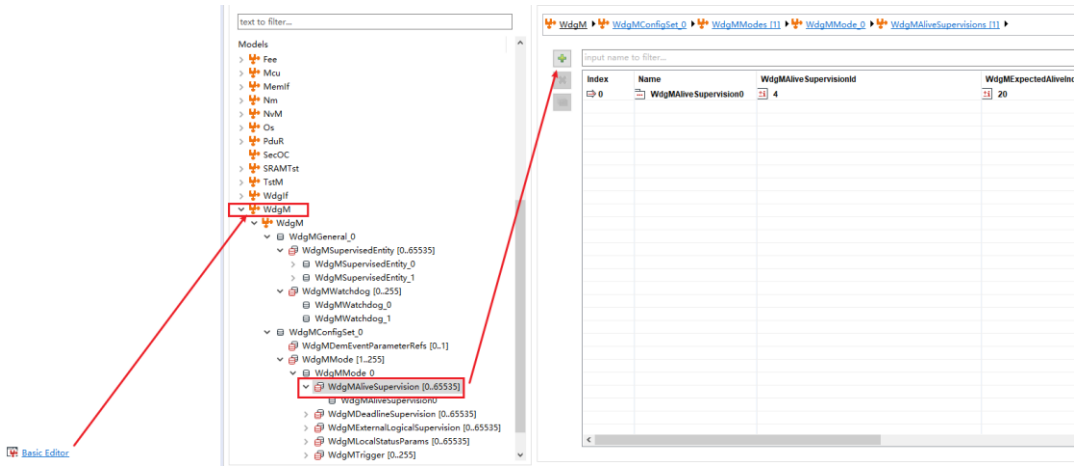


图 5: 木牛 SUPERVISION ENTITIES 配置界面示例 1

FIGURE 5: MUNIU SUPERVISION ENTITIES CONFIGURATION INTERFACE EXAMPLE1

7.1.2 配置 SE 打卡点 CONFIGURING SE PUNCH POINTS

在对应的监控实例配置页面下可以添加多个 SE 打卡点。

Multiple SE punch points can be added under the corresponding monitoring instance configuration page.

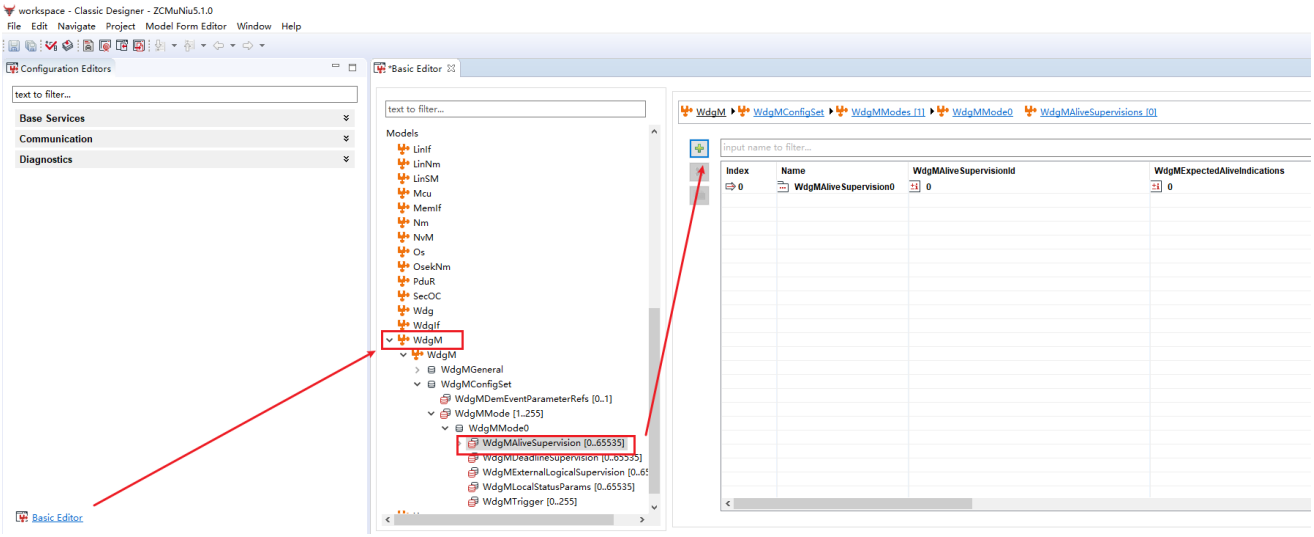


图 6: 木牛 CHECKPOINTS 配置界面示例

FIGURE 6: MUNIU CHECKPOINTS CONFIGURATION INTERFACE EXAMPLE

另外，在 WdgMInternalTransition 页面中可以设置打卡点的执行顺序，以此来实现逻辑监控功能。

In addition, you can set the execution order of punch points in the WdgMInternalTransition page as a way to implement the logic monitoring function.

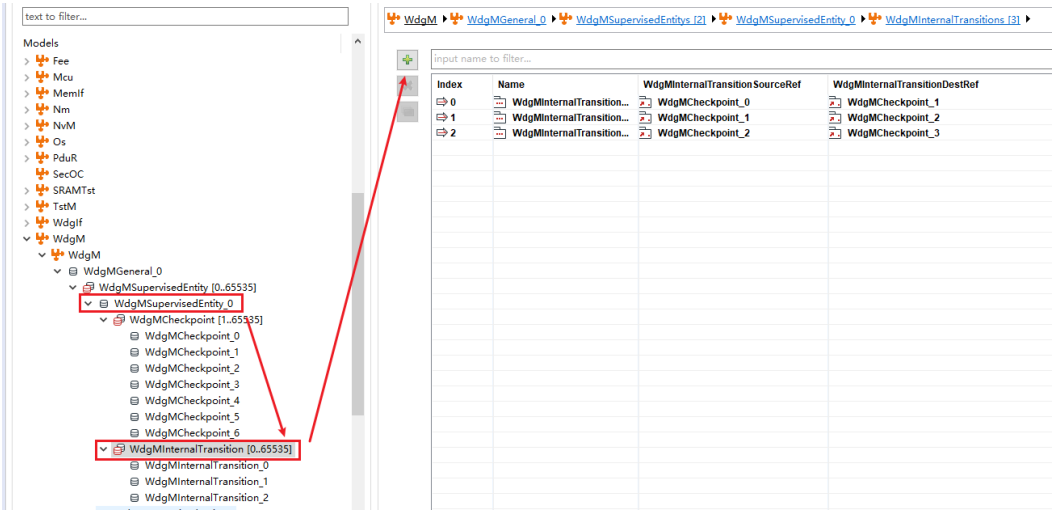


图 7: 木牛 INTERNAL TRANSACTION 配置界面示例

FIGURE 7: MUNIU INTERNAL TRANSACTION CONFIGURATION INTERFACE EXAMPLE

7.1.3 添加监控实例的状态参数配置 ADDING STATE PARAMETER CONFIGURATION FOR MONITORING INSTANCES

在 WdgMAliveSupervision0 页面下，可以配置 Alive 监控需要的相关参数，例如监控周期、可容忍的打卡次数限制、监控目标等等。

Under the WdgMAliveSupervision0 page, you can configure the relevant parameters needed for Alive monitoring, such as the monitoring period, the limit of the number of punches that can be tolerated, the monitoring target, and so on.

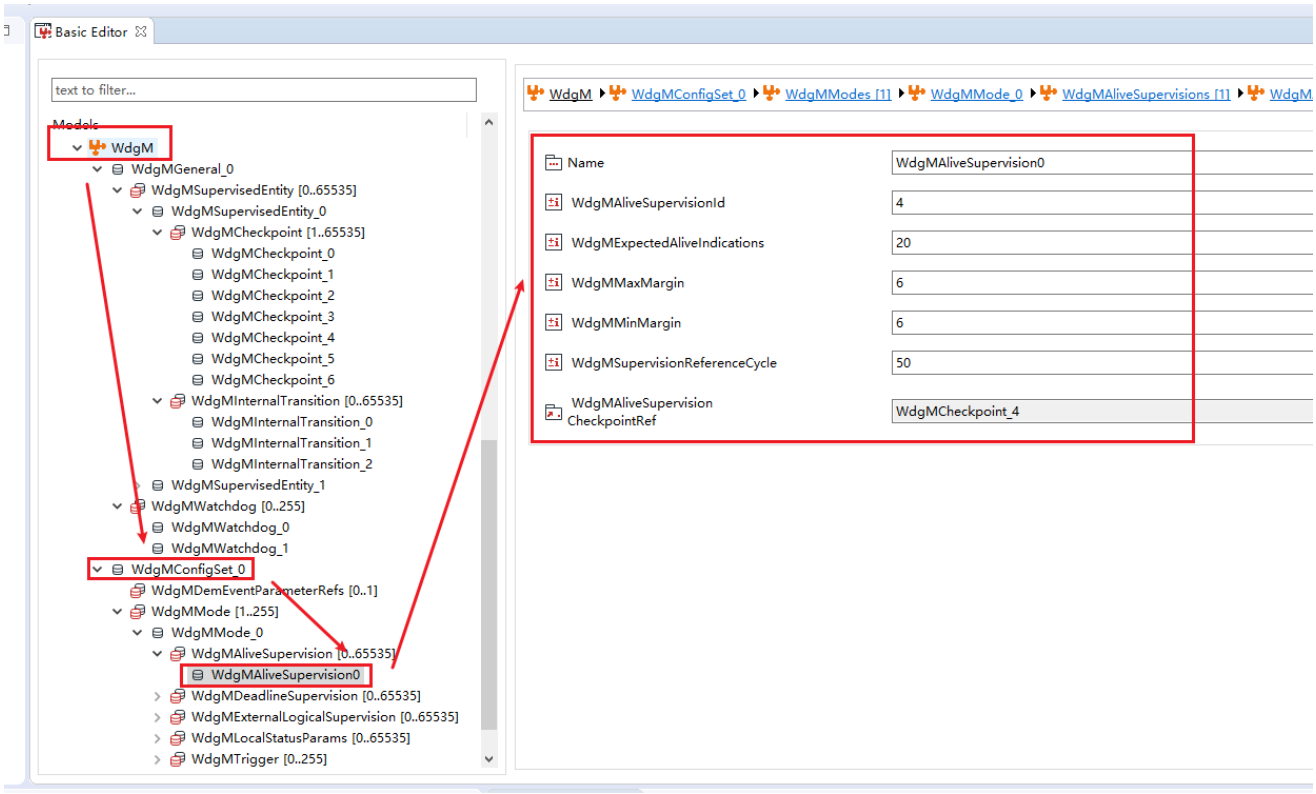


图 8: 木牛 ALIVE SUPERVISON 配置界面示例

FIGURE 8: MUNIU ALIVE SUPERVISON CONFIGURATION INTERFACE EXAMPLE

7.1.4 生成配置代码 GENERATING CONFIGURATION CODE

在配置完所有功能后即可生成 WdgM 相关的配置代码。

WdgM related configuration code can be generated after all functions are configured.

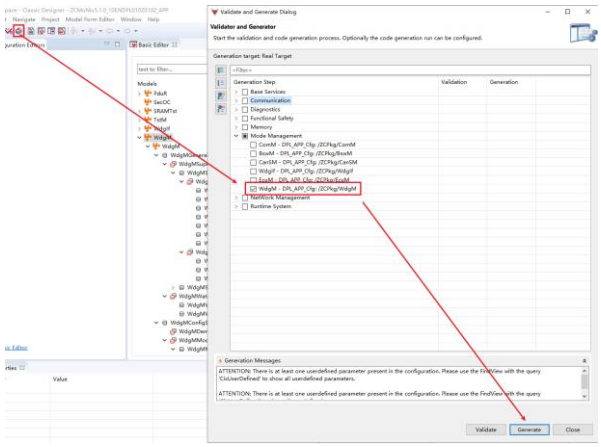


图 9: 木牛代码生成配置界面示例

FIGURE 9: MUNIU CODE GENERATION CONFIGURATION INTERFACE EXAMPLE

7.2 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:

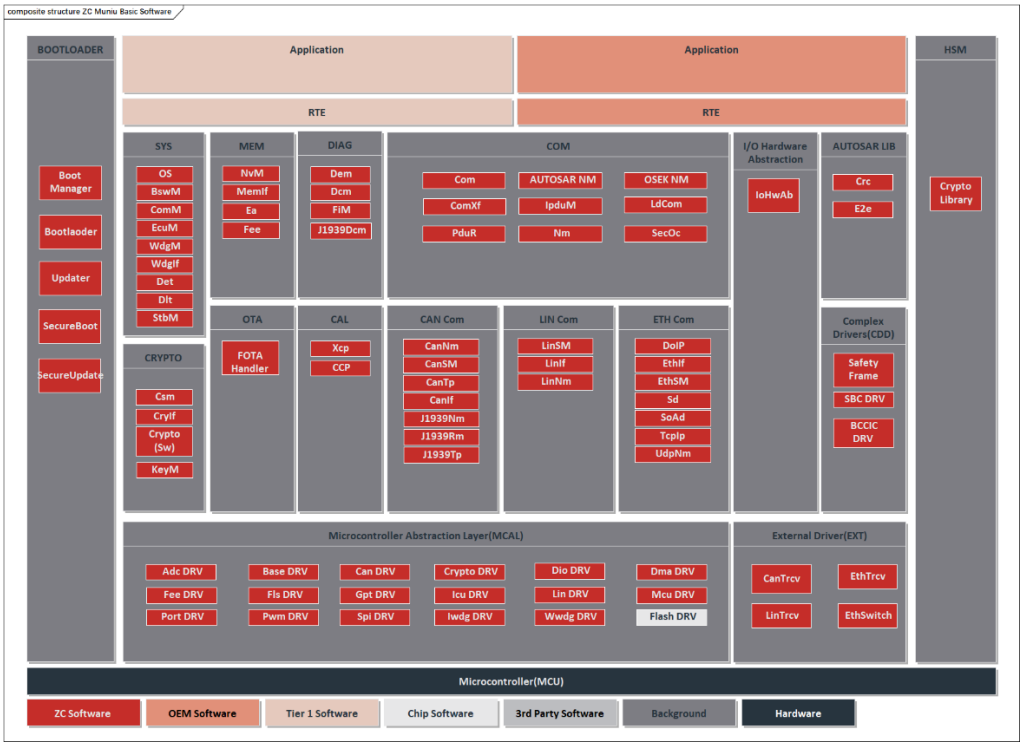


图 10:知从木牛标准平台框图

FIGURE 10: MUNIU STANDARD PLATFORM BLOCK DIAGRAM

- 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