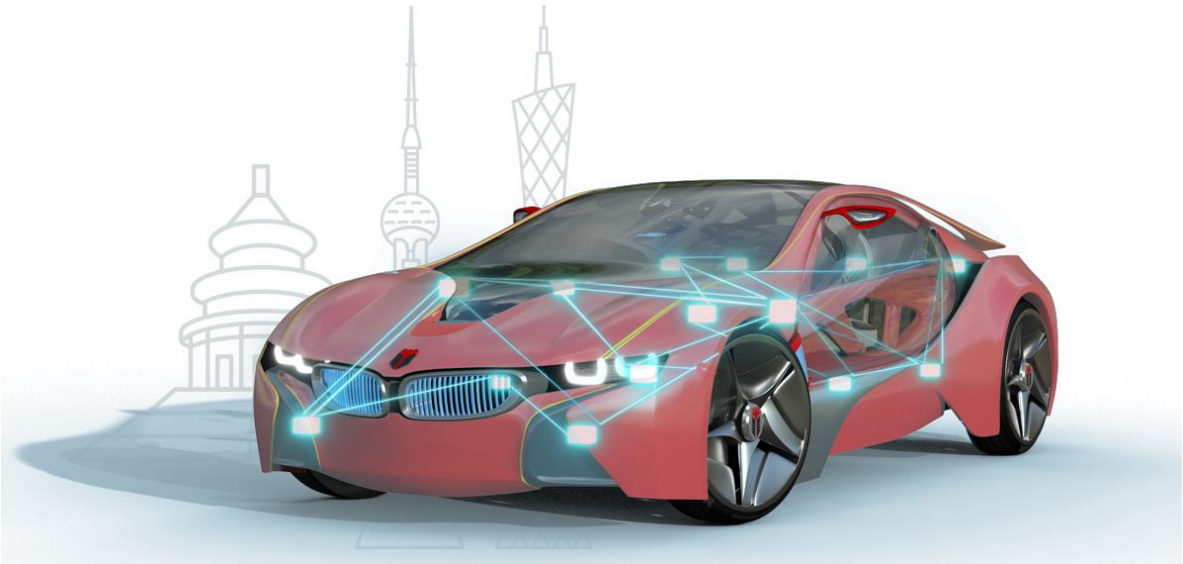
知从木牛基础软件 OS 时间保护机制功能介绍

AUTOSAR OS TIMING PROTECTION MECHANISM IN ZC.MUNIU

知从木牛基础软件平台

ZC.MuNiu Basic Software Platform



知从木牛基础软件 OS 时间保护机制功能介绍

AUTOSAR OS TIMING PROTECTION MECHANISM IN ZC.MUNIU

知从木牛基础软件平台

ZC.MuNiu Basic Software Platform

1 概述 OVERVIEW

在汽车工业向智能化、网联化快速迈进的今天，汽车电子系统的复杂度呈指数级增长。从动力控制到自动驾驶，从车载娱乐到车身管理，上百个电子控制单元（ECU）通过复杂的网络协同工作，而这一切的稳定运行离不开一套可靠的软件架构。AUTOSAR（汽车开放系统架构）作为全球汽车行业广泛认可的标准化软件框架，始终致力于解决汽车电子系统的兼容性、安全性与可扩展性问题。其中，面向功能安全的 AUTOSAR SC2（Scalability Classes 2）更是针对高安全等级需求场景（如 ISO 26262 ASIL B - D）提供了关键技术支撑，而“时间保护”特性便是 AUTOSAR SC2 保障系统时序安全性的核心手段之一。

Today, as the automotive industry is rapidly advancing toward intelligence and connectivity, the complexity of automotive electronic systems is growing exponentially. Ranging from power control to autonomous driving, and from in-vehicle entertainment to body management, hundreds of Electronic Control Units (ECUs) work in synergy through complex networks. The stable operation of all these relies on a reliable software architecture.

AUTOSAR (Automotive Open System Architecture), as a widely recognized standardized software framework in the global automotive industry, is always committed to addressing issues related to compatibility, safety, and scalability of automotive electronic systems. Among its components, AUTOSAR SC2 (Scalability Classes 2), which focuses on functional safety, provides crucial technical support specifically for scenarios with high safety level requirements (such as ISO 26262 ASIL B - D). The "Time Protection" feature is one of the core means by which AUTOSAR SC2 ensures the timing safety of the system.

2 时间保护机制介绍 INTRODUCTION TO TIMING PROTECTION MECHANISM

2.1 什么是“时间保护”? What is the Timing Protection

在 AUTOSAR SC2 的定义中，“时间保护”（Time Protection）是一套主动监控与干预系统时序行为的机制，其核心目标是：通过预先定义关键任务、通信或资源访问的“时间约束”（如最大执行时间、最小周期、通信延迟上限等），实时检测是否存在违反约束的情况，并在检测到时序故障时，触发预设的安全响应（如任务重启、故障报警、系统降级等），从而防止时序异常扩散，保障系统安全。

In the definition of AUTOSAR SC2, "Time Protection" is a mechanism for actively monitoring and intervening in the timing behavior of a system. Its core objectives are as follows: by predefining "time constraints" for critical tasks, communications, or resource accesses (such as maximum execution time, minimum cycle, upper limit of communication delay, etc.), it can real-time detect whether there are any violations of these constraints. When a timing fault is detected, it triggers preset safety responses (such as task restart, fault alarm, system degradation, etc.), thereby preventing the spread of timing anomalies and ensuring system safety.

简单来说，“时间保护”就像给汽车电子系统的“时间行为”装上了“监控摄像头”和“紧急刹车”：“摄像头”实时盯着每个关键功能的“时间进度”，一旦发现“超时”或“节奏紊乱”，“紧急刹车”就会启动，避免故障扩大。

Put simply, "Time Protection" is like installing "surveillance cameras" and an "emergency brake" for the "temporal behavior" of automotive electronic systems: the "cameras" keep a real-time eye on the "time progress" of each critical function, and once "timeouts" or "rhythm disorders" are detected, the "emergency brake" is activated to prevent the fault from escalating.

2.2 “时间保护”的核心工作流程 Core Workflow of "Time Protection"

AUTOSAR SC2 的“时间保护”并非单一功能，而是由“时间约束定义”“时序监控”“故障响应”三个核心环节组成的闭环机制，具体流程如下：

The "Time Protection" in AUTOSAR SC2 is not a single function, but a closed-loop mechanism consisting of three core links: "Time Constraint Definition", "Timing Monitoring" and "Fault Response". The specific process is as follows:

（1）第一步：预先定义“时间约束”—— 明确“安全红线” Step 1: Predefine "Time Constraints" – Define Clear "Safety Red Lines"

在系统设计阶段，工程师需要根据功能安全需求（如 ISO 26262 的 ASIL 等级），为每个关键的“时序实体”（包括任务、中断服务程序、通信消息、资源访问等）定义明确的时间约束。这些约束是“时间保护”的判断依据，常见的约束类型包括：

During the system design phase, engineers need to define clear time constraints for each critical "timing entity" (including tasks, interrupt service routines, communication messages, resource accesses, etc.) based on functional safety requirements (such as ASIL levels in ISO 26262). These constraints serve as the judgment basis for "Time Protection" and commonly include the following types:

1. 任务执行时间约束：比如“自动驾驶数据处理任务的最大执行时间不得超过 50ms”； Task execution time constraints: For example, "The maximum execution time of an autonomous driving data processing task must not exceed 50ms."
2. 任务周期约束：比如“制动压力控制任务的最小执行周期为 10ms，最大周期偏差不得超过 1ms”； Task cycle constraints: For example, "The minimum execution cycle of the brake pressure control task is 10ms, and the maximum cycle deviation must not exceed 1ms."
3. 任务抢占约束：比如“高优先级的安全监控任务不得被低优先级任务抢占超过 2ms”； Task Preemption Constraints: For example, "A high-priority safety monitoring task must not be preempted by low-priority tasks for more than 2ms."

（2）第二步：实时“时序监控”—— 检测“越线行为”Second Step: Real-time "Timing Supervision" - Detecting "Overstepping"

系统运行时，AUTOSAR SC2 会通过专门的“时间监控器”（Time Monitor）组件，对每个时序实体的时间行为进行实时跟踪。时间监控器的实现通常依赖于硬件定时器（如 MCU 的系统定时器）和软件层面的时间戳记录，具体监控逻辑包括：

In the AUTOSAR SC2 (Scalability Class 2) environment, during runtime, a dedicated "Time Monitor" component is used to perform real-time tracking of the timing behavior of each timing entity. The Time Monitor's implementation typically relies on hardware timers (like the MCU's system timer) and software-level timestamping.The specific monitoring logic includes:

4. 任务执行时间监控：任务启动时，时间监控器记录“开始时间戳”；任务结束时，计算“实际执行时间”（结束时间戳 - 开始时间戳），并与预设的“最大执行时间”对比 —— 若实际时间超过最大值，判定为“执行超时故障”； When a task starts, the Time Monitor records a start timestamp. When the task finishes, it calculates the actual execution time (end_timestamp−start_timestamp) and compares it against a pre-configured maximum execution time. If the actual time exceeds the maximum, it's flagged as an "execution timeout failure".
5. 任务周期监控：记录每次任务启动的时间戳，计算相邻两次启动的“实际周期”，与预设的“最小 / 最大周期”对比 —— 若周期过短（可能导致 CPU 资源过载）或过长（可能导致功

能响应延迟），判定为“周期异常故障”； Task Period Monitoring:The system records a timestamp each time a task starts. It then calculates the actual period between two consecutive starts and compares it to a pre-set minimum/maximum period.If the period is too short, it's flagged as a "period abnormality fault" because it could overload CPU resources.If the period is too long, it's also flagged as a "period abnormality fault" because it could cause functional response delays.

值得注意的是，AUTOSAR SC2 的时间监控器具备“独立性”—— 它不会与被监控的任务共享 CPU 资源，避免因监控逻辑本身的延迟导致误判。例如，时间监控器可能运行在独立的硬件核心上，或采用“中断级监控”的方式，确保监控的实时性与准确性。

What's noteworthy is that the AUTOSAR SC2 Time Monitor is designed with a key principle of "Independence". It doesn't share CPU resources with the tasks it monitors, which prevents its own delays from causing false positive fault detections.For example, the Time Monitor might run on a separate hardware core or use an "interrupt-level monitoring" approach to ensure its real-time performance and accuracy.

（3）第三步：触发“故障响应”—— 阻断“故障扩散”Step 3: Triggering "Fault Response" - Blocking "Fault Propagation"

当时间监控器检测到时序故障时，不会仅停留在“报警”层面，而是会根据预设的“安全机制”触发主动干预。AUTOSAR SC2 支持多种故障响应策略，具体选择需根据故障的严重程度和功能安全需求确定：When the Time Monitor detects a timing fault, it doesn't just raise an "alarm." Instead, it triggers proactive intervention based on a pre-configured "safety mechanism." AUTOSAR SC2 supports multiple fault response strategies, with the specific choice depending on the severity of the fault and the functional safety requirements.

6. 轻度故障：仅记录故障日志（如通过 DTC 诊断 trouble code 上报），同时通知系统管理模块，由管理模块判断是否需要调整任务优先级（如提升关键任务优先级，避免再次被抢占）；Minor Faults: These are logged (e.g., reported as a DTC, or Diagnostic Trouble Code) and the system management module is notified. This module then decides whether to adjust task priorities (e.g., raise the priority of a critical task to prevent it from being preempted again).
7. 中度故障：重启发生时序异常的任务或 ECU（如数据处理任务超时，重启该任务并重新加载初始数据），确保功能恢复正常；Moderate Faults: The system will restart the task or ECU that experienced the timing anomaly. For instance, if a data processing task times out, it will be restarted and its initial data reloaded to ensure the function returns to a normal state.

8. 重度故障：触发系统“安全降级”或“紧急停机”—— 例如，自动驾驶系统的核心任务连续多次超时，此时系统会立即切换到“人工驾驶接管”模式，并通过仪表盘报警提示驾驶员，若驾驶员未响应，则逐步降低车速直至停车，最大程度降低风险。Severe Faults: The system triggers a "safety degradation" or "emergency shutdown." For example, if a core task in an autonomous driving system times out multiple times in a row, the system will immediately switch to a "manual driving takeover" mode and alert the driver via the instrument cluster. If the driver doesn't respond, the system will gradually reduce the vehicle's speed until it comes to a stop, minimizing risk as much as possible.

3 知从木牛 OS：支撑“时间保护”特性的 AUTOSAR 实现 ZCMUNIU OS: AN AUTOSAR IMPLEMENTATION SUPPORTING "TIMING PROTECTION" FEATURES.

“时间保护”特性的实现离不开 知从木牛 OS 架构中多个模块的协同工作，其中最核心的模块包括：

Implementing the "Timing Protection" feature relies on the collaborative work of multiple modules within the Zhi Cong Mu Niu OS architecture. The most crucial modules for this are:

3.1 操作系统模块（OS）：时序控制的“基础平台”Operating System Module (OS): The Foundational Platform for Timing Control

知从木牛 OS 是“时间保护”的底层支撑，它负责：The ZCMuNiu OS is the underlying support for "Timing Protection," and it's responsible for:

1. 管理任务的创建、启动、暂停与终止，确保任务按照预设的优先级和周期运行；It manages the creation, activation, suspension, and termination of tasks, ensuring they run according to their predefined priorities and periods.
2. 提供“时间服务”接口，如获取系统时间戳、设置定时器等，为时间监控器提供时间数据；It provides "time service" interfaces, like getting system timestamps or setting timers, to supply time data for the Time Monitor.
3. 支持“任务超时检测”—— 当任务执行时间超过 OS 中配置的“超时阈值”时，OS 会直接触发中断，通知时间监控器。It supports "task timeout detection." When a task's execution time exceeds the configured "timeout threshold" in the OS, the OS directly triggers an interrupt to notify the Time Monitor.

此外，木牛 OS 还强化了“资源保护”功能，避免低优先级任务长时间占用关键资源（如共享内存、硬件外设），导致高优先级任务超时 —— 例如，OS 会限制某个任务对 CAN 控制器的最大占用时间，确保其他任务能及时访问通信资源。

Additionally, the Mu Niu OS enhances "resource protection" to prevent low-priority tasks from holding critical resources (like shared memory or hardware peripherals) for too long, which could cause high-priority tasks to time out. For example, the OS can limit a task's maximum usage time for a CAN controller, ensuring other tasks have timely access to communication resources.

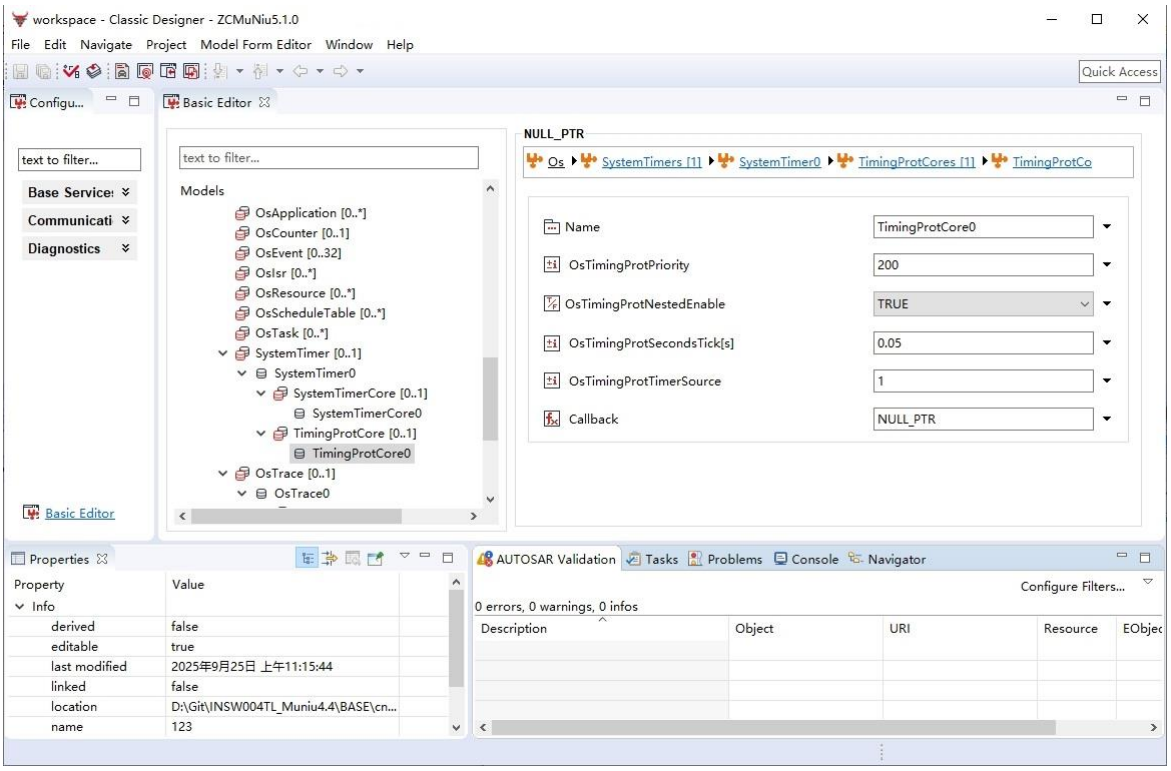


图 1. 木牛 OS 时间保护 SystemTimer 参数配置页面

Figure 1 .MuNiu OS: SystemTimer Parameter Settings for Timing Protection

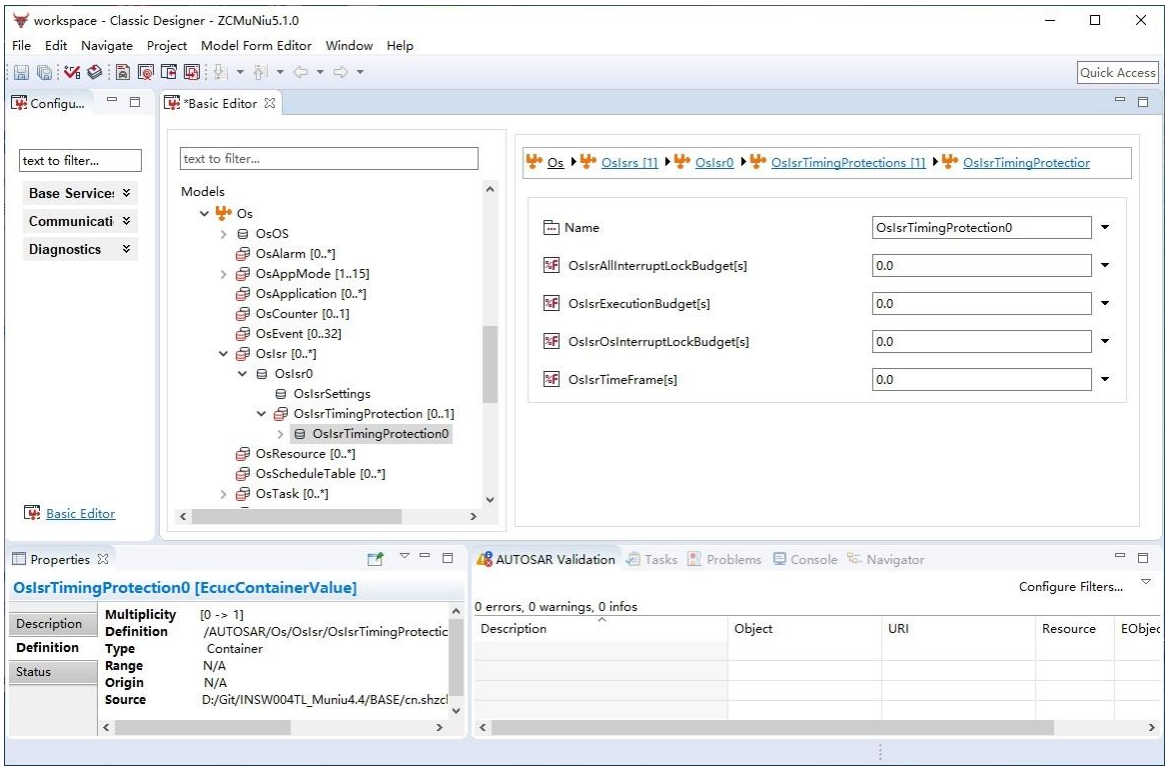


图 2. 木牛 OS 时间保护 OsIsr 参数配置页面

Figure 2 .MuNiu OS: OsIsr Parameter Settings for Timing Protection

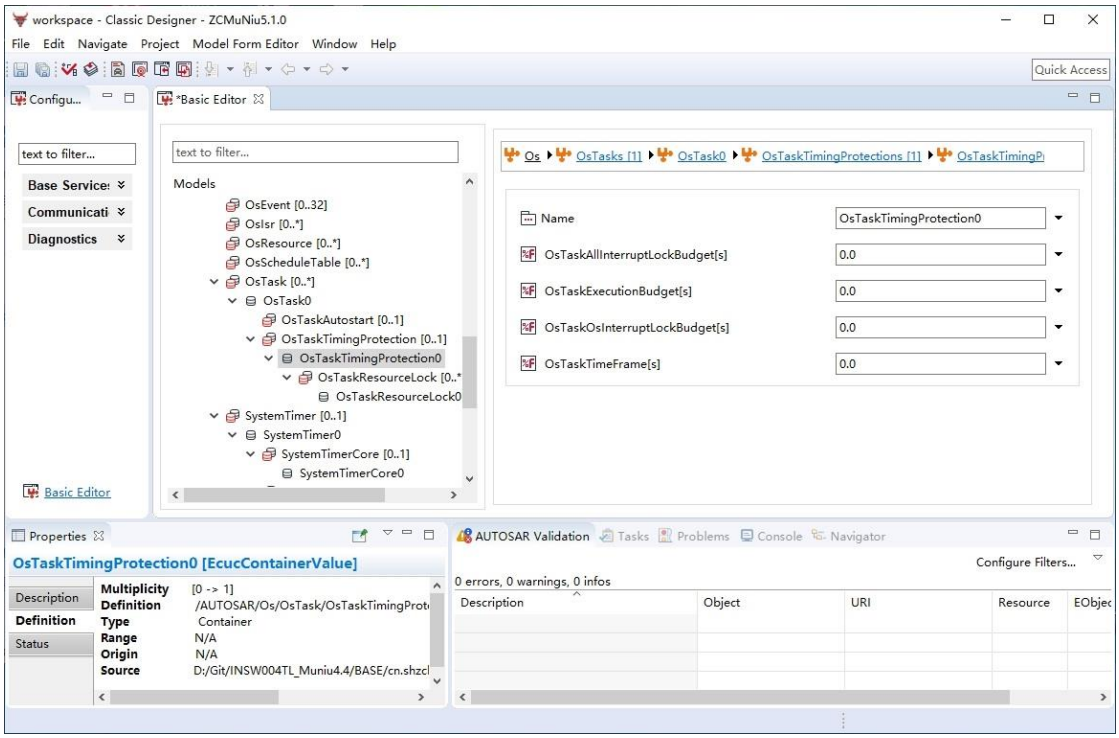


图 3. 木牛 OS 时间保护 OsTask 参数配置页面

Figure 3 .MuNiu OS: OsTask Parameter Settings for Timing Protection

3.2 WdgM 在功能安全中的应用 Application of the WdgM in Functional Safety

除了时间保护机制，在 AUTOSAR 中还有一种 WatchDog(WdgM)机制，在功能安全的保障中发挥着重要作用，相比于时间保护机制 WdgM 侧重于：

In addition to the timing protection mechanism, AUTOSAR also includes a Watchdog Manager (WdgM) mechanism, which plays a critical role in ensuring functional safety. Compared to the Timing Protection mechanism, the WdgM focuses on:

1. 核心功能：监视软件实体的执行行为。它关注的是整个任务或一组任务的“生命体征”，即它们是否按照预定的时间表和逻辑顺序正常执行。Core Function: Monitoring the execution behavior of software entities. It's concerned with the "vital signs" of an entire task or a group of tasks—that is, whether they're executing correctly according to a predefined schedule and logical sequence.
2. 主要目标：检测和响应系统级故障。例如，检测任务死锁、无限循环、调度器故障等。当出现严重故障时，WdgM 通常会触发一个硬件复位，让系统从不安全状态恢复到初始的安全状态。Primary Goal: Detecting and responding to system-level faults. For example, it detects task deadlocks, infinite loops, and scheduler failures. When a severe fault occurs, the WdgM typically triggers a hardware reset to help the system recover from an unsafe state to a safe, initial state.

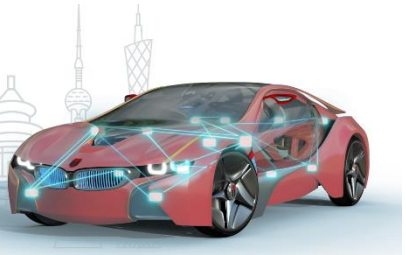
3. 作用层次：主要在系统层面。它监视多个可执行实体（Runnables）或任务（Tasks）的整体行为。它并不深入到单个任务的执行细节，而是关心它们是否“活着”以及是否按时完成。Level of Operation: It operates primarily at the system level. It monitors the overall behavior of multiple runnables or tasks. It doesn't delve into the execution details of a single task; instead, it cares about whether they are "alive" and completing on time.
4. 检测粒度：粗粒度。它通常检测的是相对较长时间的超时或逻辑错误，例如一个任务在几毫秒或几十毫秒内没有被执行。Detection Granularity: Coarse-grained. It typically detects timeouts or logical errors over a relatively long period, such as a task not being executed for tens or hundreds of milliseconds.
5. 主要作用：故障恢复。它的主要目的是在检测到严重故障后，通过复位等手段，使系统从一个不安全的状态强制恢复到安全状态。Primary Role: Fault recovery. Its main purpose, upon detecting a severe fault, is to force the system to recover from an unsafe state to a safe one, often through a reset.

4 总结 SUMMARY

在汽车智能化的浪潮中，“安全”始终是不可逾越的底线，而时序安全作为汽车电子系统安全的重要组成部分，直接决定了功能的可靠性与用户的生命安全。符合 AUTOSAR SC2 标准的知从木牛 OS 的“时间保护”特性通过标准化的“约束定义 - 实时监控 - 故障响应”机制，为分布式 ECU 场景筑起了成熟的安全屏障。

In the wave of automotive intelligence, "safety" remains the absolute bottom line. Timing safety, as a critical component of automotive electronic systems, directly determines the reliability of functions and the safety of users' lives.The "Timing Protection" feature of the Zhi Cong Mu Niu OS, which complies with the AUTOSAR SC2 standard, builds a mature safety barrier for distributed ECU scenarios through a standardized mechanism of Constraint Definition, Real-time Monitoring, and Fault Response.

对于普通消费者而言，这些隐藏在操作系统深处的“时间守护者”或许无形，但正是它们的精准运作，让自动驾驶、智能制动等功能在毫秒之间做出安全决策；对于汽车行业而言，标准化与本土化技术的并行发展，既保障了技术的安全性与兼容性，又推动了智能驾驶技术的快速落地。随着车载计算能力的持续提升，时间保护技术将不断迭代，为更复杂、更智能的汽车电子系统保驾护航。

For the average consumer, these "time guardians" hidden deep within the operating system may be invisible, but their precise operation is what allows functions like autonomous driving and smart braking to make safe decisions in milliseconds. For the automotive industry, the parallel development of standardized and localized technologies not only ensures the safety and compatibility of the technology but also accelerates the rapid adoption of intelligent driving. As in-vehicle computing capabilities continue to improve, timing protection technology will constantly evolve, ensuring the safety of more complex and intelligent automotive electronic systems.

公众号
业务联系
成为全球领先的汽车基础软件公司
To Be the Global Leading Automotive Basic Software Company