



AUTOSAR OS 介绍及特性应用

Leon Guan, Elektrobit

15 Aug 2024

E-mail: zhao.guan@elektrobit.com



BOSCH Continental



STELLANTIS

TOYOTA VOLKSWAGEN GROUP

Agenda

- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > AUTOSAR OS 主要特性
- > AUTOSAR OS 特性应用
- > Q&A

Agenda

- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > AUTOSAR OS 主要特性
- > AUTOSAR OS 特性应用
- > Q&A

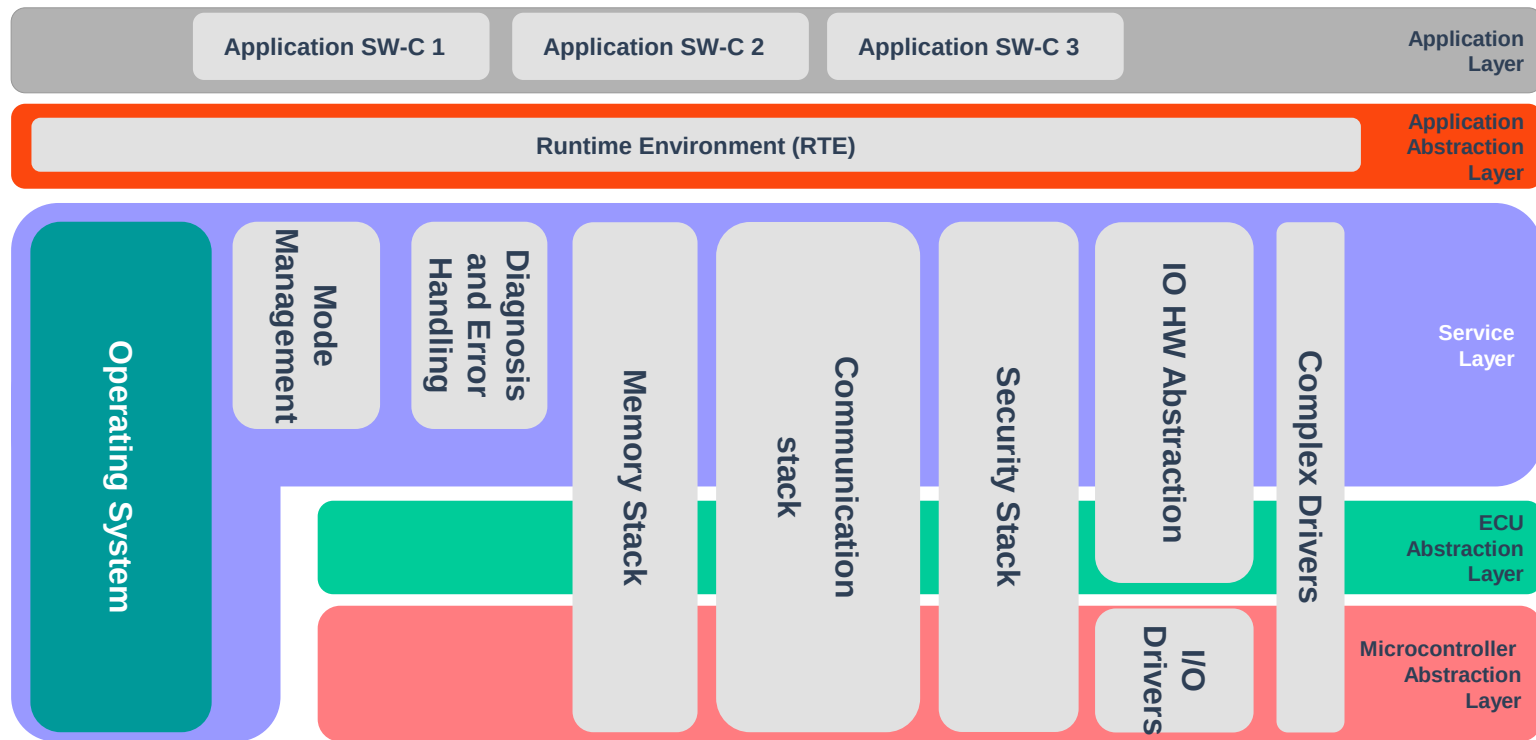
AUTOSAR OS 简介

OS 是什么?

- OS 的缩写
 - OS 是单词 Operating System 的缩写，操作系统。
- AUTOSAR OS 的功能描述
 - 嵌入式汽车 ECU 中的实时操作系统，为软件的动态行为奠定了基础。
 - 管理任务和事件的调度、对于不同任务之间的数据流，提供监视和错误处理的特性。
 - AUTOSAR OS 仅涵盖调度和同步的核心功能。

AUTOSAR OS 简介

Overview of Architecture



AUTOSAR OS 简介

AUTOSAR OS 可伸缩类型

Scalability Class	Features
SC1	• OSEK OS features • ScheduleTables
SC2	• SC1 • Timing protection • ScheduleTable synchronization
SC3	• SC1 • Memory protection • Service protection • Trusted functions • OS-Applications
SC4	• SC2 and SC3

AUTOSAR OS 简介

AUTOSAR OS 可伸缩类型

- 总结概括
 - 在实际的软件开发中，了解SC分类是有助于知晓软件包中的OS是否涵盖实际的软件所需要的功能需求。
 - 通常情况，在确定了SC的类型之后，需要检查所支持的feature，比如：(Timing protection and memory protection)。
 - 但请注意，支持的功能可能会受到硬件功能的限制。例如，如果硬件没有 MPU/MMU，则操作系统不支持内存保护。
 - 无论如何，即使拥有 SC4 OS 实现也并不意味着支持 AUTOSAR 定义的所有功能。您必须检查 OS 实现的偏差和限制，以获取所有微小的细节。

Agenda

- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > AUTOSAR OS 主要特性
- > AUTOSAR OS 特性应用
- > Q&A

AUTOSAR OS 详细概述

AUTOSAR OS由哪些部分组成？

- 组成元素
 - OS Object: OS Object 是一个由操作系统管理的数据结构。 OS Objects 包含 Task, ISR, Counter, Alarm 和 Schedule Table。
 - OS Application: OS Application 包含 OS Object, 可以比喻为 OS Object 的容器。
 - Resource: 资源是提供互斥的 OS Object。
- 调度
 - 操作系统运行时, 调度任务和事件。

AUTOSAR OS 详细概述

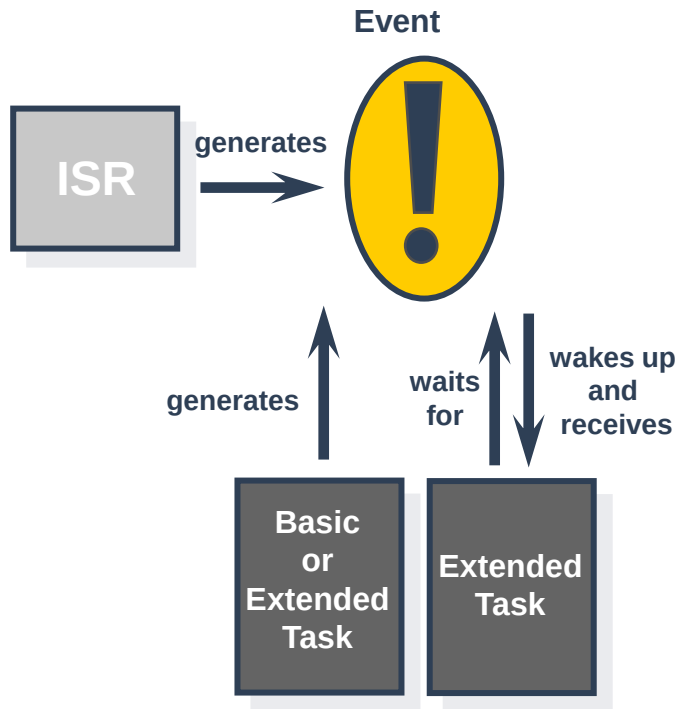
Task

- Task 是一个 C 函数，能够被操作系统调度。
- 通常，在实际的操作系统中，应用程序被执行至少有一个 Task 参与。
- Task 必须被激活去参与操作系统的调度。Task 的激活方式可以通过调用 API 或其他 OS 机制。
- Task 在实际调度中，存在优先级，可以通过配置的方式来定义。

AUTOSAR OS 详细概述

Event

- Event 是一种通知机制，用于通知 extended Task。
- Extended Task 可以等待（一个或多个）Event。等待期间，处理器可以执行其他任务。
- 可以通过任务（Basic Task 或 Extended Task）、（类别 2）ISR 或操作系统设置事件。
- Event 只是通知作用。它们不包含任何数据，也不包含任何其他信息（例如事件来源）。

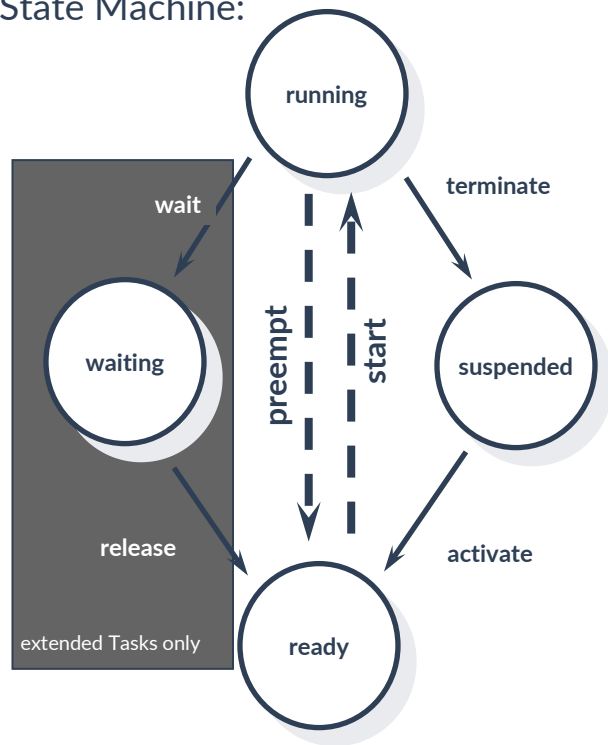


AUTOSAR OS 详细概述

Basic & Extended Task

- AUTOSAR OS 中 Task 分为 Basic Task 和 Extended Task
 - Basic task 和 Extended task 根据其状态机运行切换。
 - Extended task 支持事件触发 waiting 状态。
- AUTOSAR OS 根据以下属性来安排任务
 - 在所有准备运行的任务中，将采用优先级最高的任务。
 - 如果有多个具有相同最高优先级的任务准备运行，则运行最旧的任务。

Task State Machine:

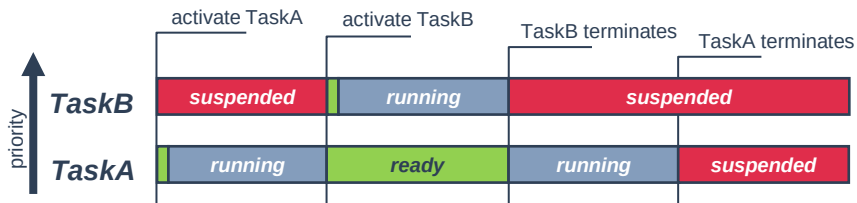


AUTOSAR OS 详细概述

Preemptability: 一个 Task 可以被配置为 Preemptive 或者 Non-preemptive

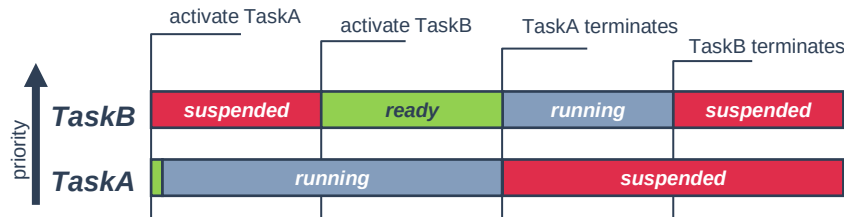
- Preemptive

当更高优先级的任务被激活时，操作系统会抢占此时正在执行的任务。



- Non-preemptive

操作系统不得抢占此类任务以执行其他任务。



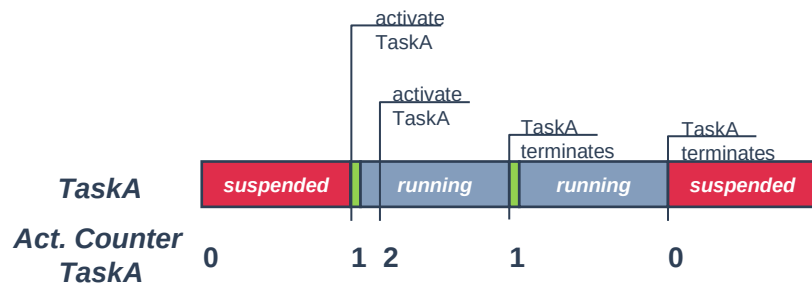
① 非抢占式任务可以使用 API Schedule() 来允许具有更高优先级的任务运行。

AUTOSAR OS 详细概述

Multiple activations

Activation 特点

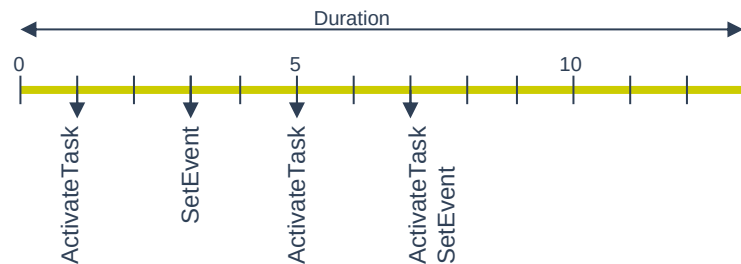
- 对于每个任务，操作系统都有一个当前任务激活的计数器。可以为每个任务单独配置允许的最大值。
- 只要计数器小于配置的最大值，才允许额外的任务激活，并增加任务激活计数器。
- 一旦达到最大激活次数，新的激活将被拒绝。
- 当任务终止时，计数器会减少，任务会直接进入就 ready 状态（而不是 suspended）。
- 每个任务（始终）只有一个 Instance。



AUTOSAR OS 详细概述

Time-triggered Task activation

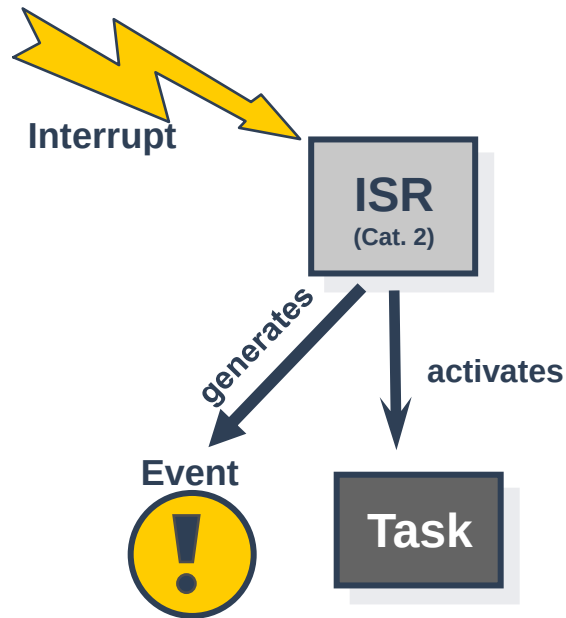
- Alarm
 - 当附加的计数器达到给定值时触发动作（例如：任务激活）的机制。
- Counter
 - 一种 AUTOSAR OS 机制，用于计数事件。例如：timer ticks.
- ScheduleTable
 - 附加到计数器的预定义操作序列（Expiry Points）。
 - 如果计数器达到到 Expiry Point 的值，则会触发相应的操作。
 - 可以执行多个到 Expiry Point 操作，例如任务或设置事件。



AUTOSAR OS 详细概述

Interrupt Service Routines (ISRs)

- ISR 是用于处理硬件事件（中断）的 C 函数。
- ISR 不像任务那样进行调度。相反，它们直接由硬件中断请求触发。
- ISR 抢占任务（除非任务明确禁用中断）。
- 通常支持 ISR 嵌套（ISR 具有由硬件定义的自己的优先级方案）。
- AUTOSAR 定义了两个 ISR 类别：
 - 类别 1：仅允许使用少数 OS API。ISR 使用当前堆栈。实现取决于编译器。
 - 类别 2：OS 提供 ISR 框架，允许访问其他 OS API，尤其是包括激活任务或设置事件。ISR 有自己的堆栈。



AUTOSAR OS 详细概述

要点总结

- 任务是 C 函数，可由 AUTOSAR OS 同时调度。
- 有基本任务和扩展任务。扩展任务支持等待事件。
- 任务可以是抢占式或非抢占式。
- 任务调度基于优先级。对于具有相同优先级的任务，任务激活时间决定下一个应调度哪个任务。
- ISR 是硬件事件（中断）的处理程序。
- ISR 中断任务的执行（除非在正在运行的任务中明确禁用中断）。

Agenda

- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > **AUTOSAR OS 主要特性**
- > AUTOSAR OS 特性应用
- > Q&A

AUTOSAR OS 主要特性

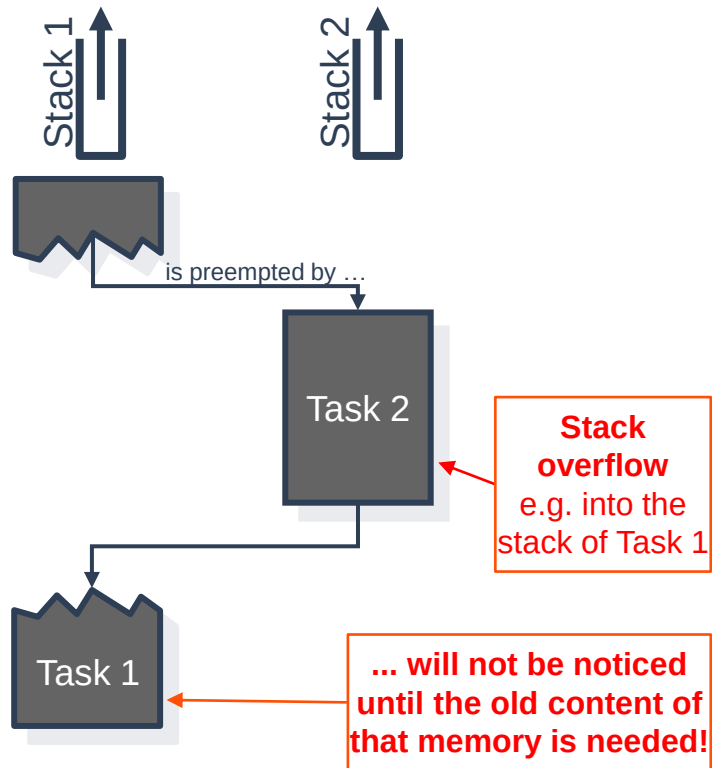
Main characteristics

- AUTOSAR OS 是一个静态的实时操作系统
 - 没有动态处理系统资源。
 - AUTOSAR OS 是可以配置的。
 - AUTOSAR OS 必须为每个应用程序单独配置和生成操作系统。
- Protection Facilities
 - AUTOSAR 概念要求多源 OS Application 共存于同一处理器上。为了防止这些 OS 应用程序之间发生意外交互，有必要提供保护它们彼此不受干扰的机制。
- Monitoring Facilities
 - 监控功能会在执行的适当阶段检测错误，而不是在错误发生时立即检测。因此，任何监控功能都是在运行时检测故障，而不是预防故障。

AUTOSAR OS 主要特性

Stack

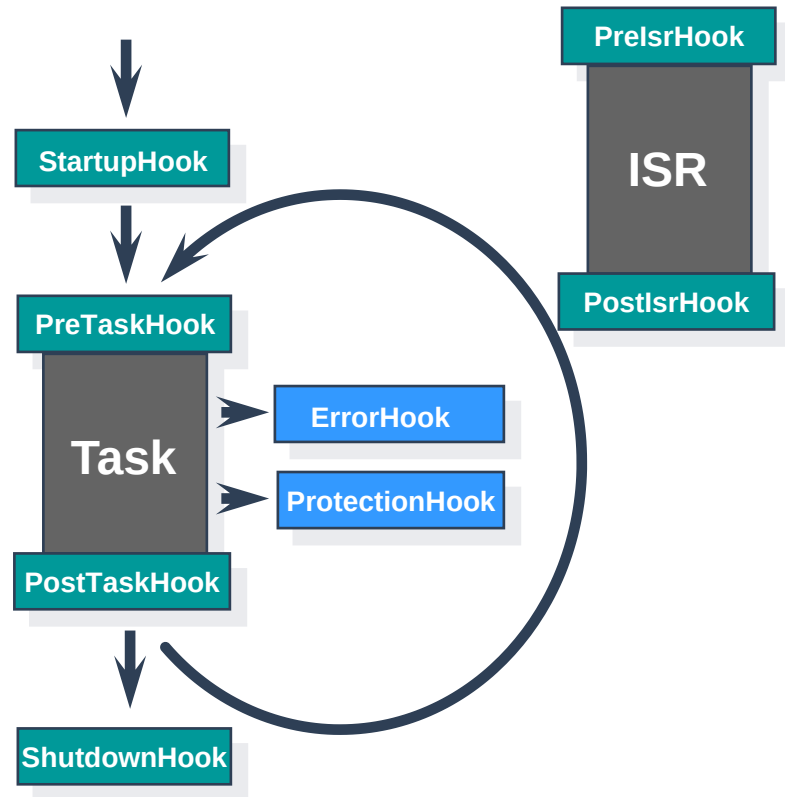
- 堆栈：用于以下用途的内存
 - 局部变量（编译器）。
 - 存储函数调用时应保留的数据（编译器）。
 - 操作系统运行时的抢占。
- 根据操作系统配置，您可以拥有许多堆栈，例如操作系统内核、任务和 ISR。
- AUTOSAR OS 为堆栈保留内存，堆栈大小是可配置的。
 - 如果堆栈太小，系统在大多数情况下会因为堆栈溢出而失败，即当数据存储在“堆栈上”时，与堆栈相邻的其他内存会被覆盖。



AUTOSAR OS 主要特性

Hook

- Hook 是回调函数，在特定情况下由 AUTOSAR OS 调用。
- 如果在配置中启用了 Hooks，则用户必须使用预定义的名称和原型来定义它们。
- Inside Hooks 中，并非所有的 OS API 函数都可用。
- 对于 Hook 函数实际的应用，还需要参阅操作系统文档，以了解详细信息与 AUTOSAR 的可能偏差。



AUTOSAR OS 主要特性

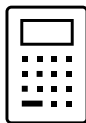
Debugging

- Error Checking
 - 启用扩展错误检查并检查 API 返回值。
 - 启用 ErrorHook 并检查它是否从未在您的 ECU 上被调用。
 - 启用 ProtectionHook 并妥善应用。
- Additional Help
 - 检查有效的文档，以获取扩展调试的支持，例如堆栈检查支持。
 - AUTOSAR OS 支持生成 ORTI（OSEK 运行时接口）文件。

Agenda

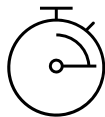
- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > AUTOSAR OS 主要特性
- > **AUTOSAR OS 特性应用**
- > Q&A

AUTOSAR OS 特性应用



CPU Load Calculation

OsMon Module

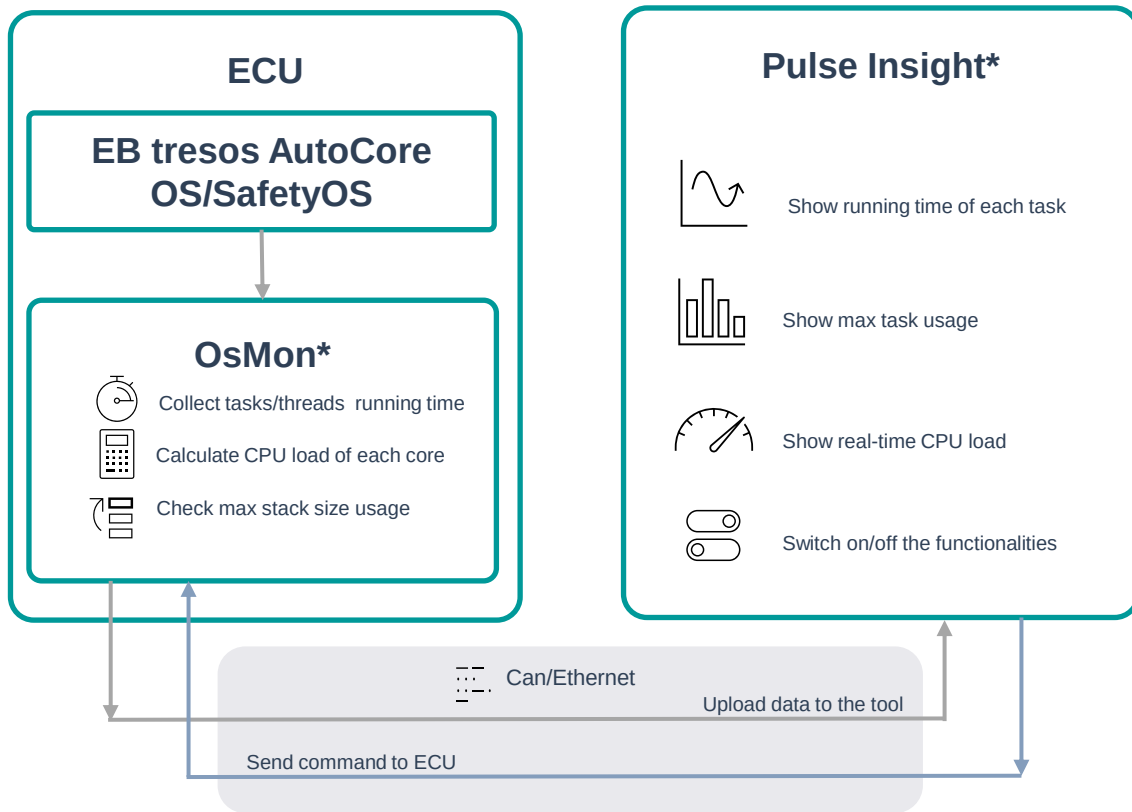


Task Running Time



Max Stack Usage

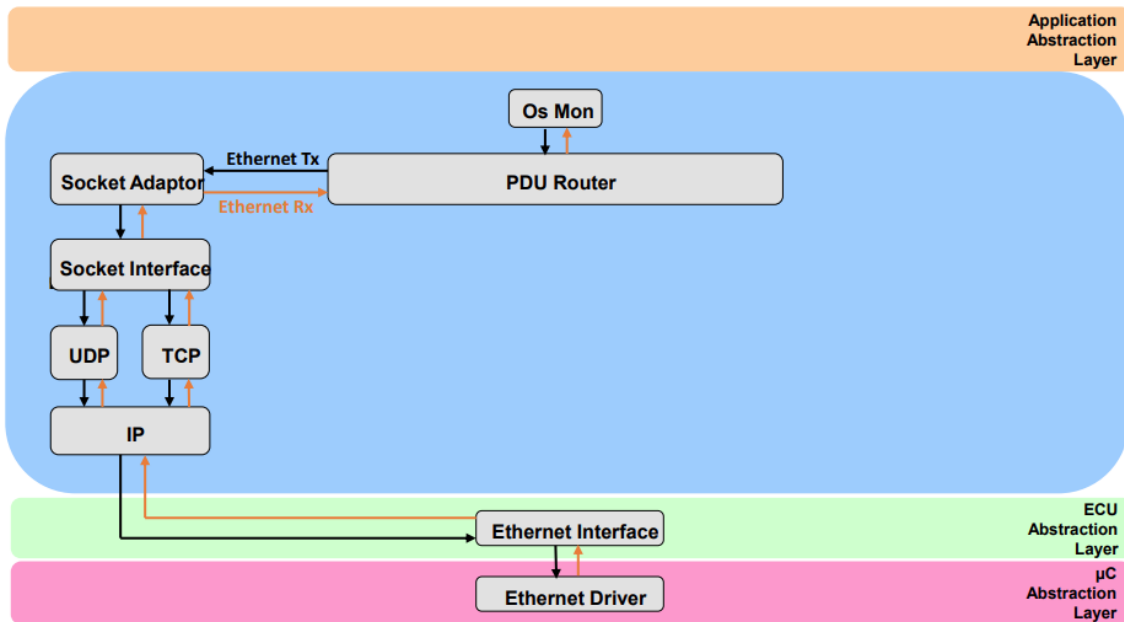
AUTOSAR OS 特性应用



AUTOSAR OS 特性应用

Os Monitoring 模块的架构图

Os Monitoring - Overview



Agenda

- > AUTOSAR OS 简介
- > AUTOSAR OS 详细概述
- > AUTOSAR OS 主要特性
- > AUTOSAR OS 特性应用
- > Q&A

Q&A