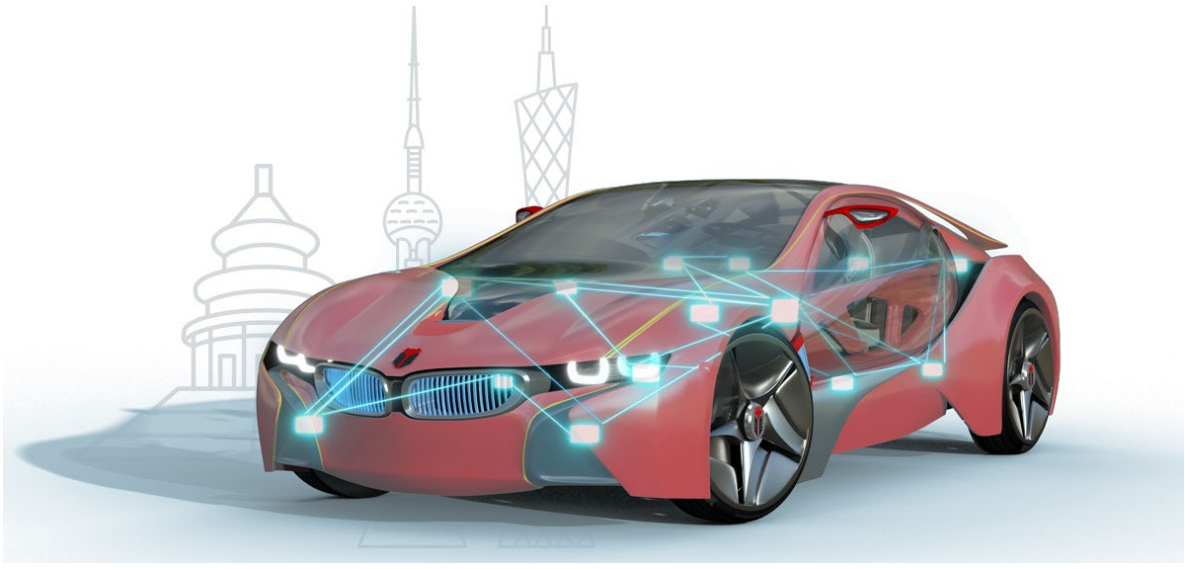
知从木牛 SOC 产品手册

ZC.MUNIU SOC PRODUCT MANUAL

知从木牛基础软件平台

ZC.MuNiu Basic Software Platform



知从木牛 SOC 产品手册

ZC.MUNIU SOC PRODUCT MANUAL

知从木牛基础软件平台

ZC.MuNiu Basic Software Platform

1 功能概述 FUNCTIONAL OVERVIEW

知从木牛 SOC 产品基于 SOA 软件架构设计，兼容 Adaptive AutoSAR 的车辆解决方案。知从木牛 SOC 将经过量产验证的软件、开源软件以及第三方软件汇集一身，同时也融入对于 HPC 环境至关重要但并不区分不同汽车的工具和服务。知从木牛 SOC 集成了基于 Linux 和 Adaptive AUTOSAR 的高性能功能安全软件栈、一个基于 Classic AUTOSA 的实时安全软件栈、一个 hypervisor ， 以及用于 HPC 更新和平台健康管理功能的软件。另外还包括用于自动化构建和集成的工具和服务。

ZC.MuNiu SOC (System on Chip) product is designed based on the SOA (Service-Oriented Architecture) software architecture and is compatible with Adaptive AutoSAR vehicle solutions. It integrates software that has been mass-production verified, open-source software, and third-party software. It also incorporates tools and services that are crucial for the HPC (High-Performance Computing) environment but are not specific to different automobiles.ZC.MuNiu SOC product includes a high-performance functional safety software stack based on Linux and Adaptive AUTOSAR, a real-time safety software stack based on Classic AUTOSAR, a hypervisor, and software for HPC updates and platform health management functions. Additionally, it includes tools and services for automated build and integration.

2 应用领域 APPLICATION FIELD

知从木牛 SOC 产品，支持 SOA 框架的服务开发。目前主要应用于如下场景：

ZC.MuNiu SOC product facilitates the development of services based on the SOA framework. Its primary applications currently include:

- 高性能 ECU，如 ADAS，多媒体和联网应用
  High-performance ECUs for applications like ADAS, multimedia, and connectivity.
- 高度自动化操作相关的安全相关的系统
  Systems related to highly automated operations and safety-related functions.
- Adaptive AUTOSAR 相关的 C++开发
  Development in C++ related to Adaptive AUTOSAR.
- 车辆相关的独立软件开发
  Independent vehicle-related software development.
- SOA 相关的服务开发
  Service development related to SOA.
- SOC 快速安全启动方案
  Rapid SecureBoot solutions for SOC

3 开发背景 DEVELOPMENT BACKGROUND

随着时代的进步，智能化、网连化变成了当前汽车发展的大趋势。Classic AutoSAR 是基于强实时性的嵌入式 OS 上开发出来的软件架构，能满足传统汽车定制化的功能需求，且能很好胜任；但是一旦要汽车接入网络，网络很可能有延迟、干扰，很可能无法满足强实时性。这种情况下 Classic AutoSAR 就无能为力了。

With the advancement of the times, intelligence and connectivity have become the major trends in the development of automobiles today. Classic AUTOSAR is a software architecture developed based on a highly real-time embedded operating system, which can meet the customized functional requirements of traditional automobiles and perform well. However, once automobiles are connected to the network, there may be delays and interference, which may not meet the strong real-time requirements. In such cases, Classic AUTOSAR is powerless.

Adaptive Autosar 与 Classic Autosar 相比，虽实时性要求有所降低，但在保证一定功能安全等级的基础上，大大提高了对高性能处理能力的支持，以支持智能互联应用功能的开发，因此 C + + 将成为 Adaptive Autosar 平台的主要开发语言。

Compared with Classic AUTOSAR, Adaptive AUTOSAR has somewhat reduced real-time requirements, but it has greatly improved the support for high-performance processing capabilities on the basis of ensuring a certain level of functional safety, in order to support the development of intelligent and connected application functions. Therefore, C++ will become the main development language for the Adaptive AUTOSAR platform.

微处理器上执行的软件组件通常不是基于 AUTOSAR Classic 标准，取而代之的是 AUTOSAR Adaptive，以满足模块化、动态性和持续更新能力的需求。AUTOSAR Adaptive 正在成为汽车高性能平台上约定俗成的软件标准。AUTOSAR Adaptive 使用符合 POSIX 标准的操作系统，如 Linux、 PikeOS 或 QNX，并为其在汽车领域的应用提供了功能扩展。AUTOSAR Adaptive 还为诊断、信息安全和功能安全提供功能支持。

Software components running on microprocessors are usually not based on the AUTOSAR Classic standard, but are replaced by AUTOSAR Adaptive, to meet the requirements of modularity, dynamism, and continuous update capabilities. AUTOSAR Adaptive is becoming a generally recognized software standard on high-performance automotive platforms. AUTOSAR Adaptive uses operating systems that comply with POSIX standards, such as Linux, PikeOS, or QNX, and provides functional extensions for its application in the automotive field. AUTOSAR

Adaptive also provides functional support for diagnostics, information security, and functional safety.

在 AUTOSAR Adaptive 项目中，软硬件隔离使得 OEM 和供应商之间的任务分配有了新的变化。从前一个功能块通常被作为汽车中的一个物理设备来订购，现在完全可以做到只采购软件。为了实现这种方式，每个 AUTOSAR Adaptive 应用都是一个独立的二进制文件，应用开发将独立于 ECU 开发。

In AUTOSAR Adaptive projects, the isolation of software and hardware has brought new changes to the task allocation between OEMs and suppliers. In the past, a functional block was usually ordered as a physical device in a car, but now it is completely possible to purchase only software. To achieve this approach, each AUTOSAR Adaptive application is an independent binary file, and application development will be independent of ECU development.

随着无人驾驶技术的如火如荼，车联网及万物互连、云技术的日益发展，Adaptive Autosar 的出现不仅可满足现有需求，还可满足未来汽车技术的革新变化，由于其支持各种自适应的部署、复杂的微控制器以及各种非 Auosar 系统的互动，未来汽车将拥有不同类型的架构并互相进行补充。

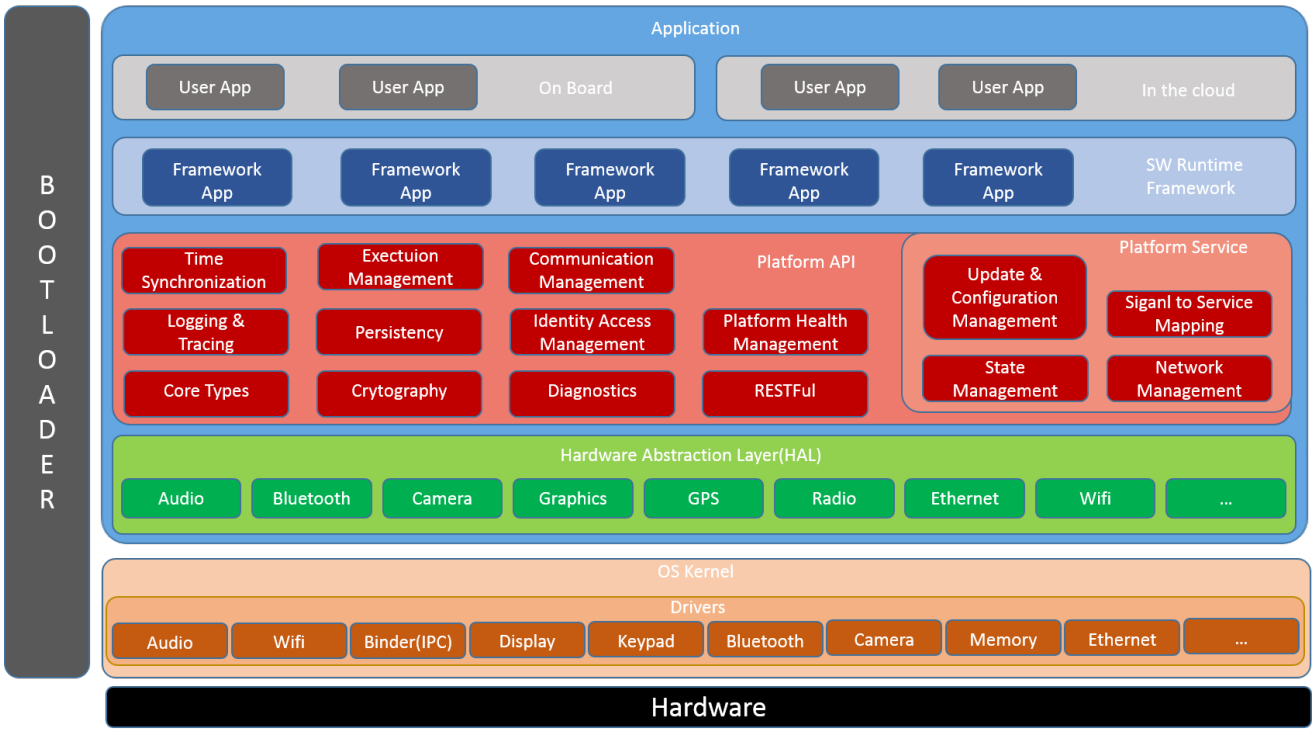
With the vigorous development of unmanned driving technology, the increasing development of vehicle networking and the Internet of everything, and cloud technology, the emergence of Adaptive AUTOSAR can not only meet the current requirements but also the innovative changes in future automotive technology. Because it supports various adaptive deployments, complex microcontrollers, and interactions with various non- AUTOSAR systems, future automobiles will have different types of architectures that complement each other.

知从木牛 SOC 产品是符合 Adaptive AUTOSAR 的 ECU 开发解决方案。该方案采用 SOA 架构设计，包括从系统功能设计，ECU 功能映射和组件配置，运行时环境自动代码生成等一系列的工具套件，提供了一套经过实践验证的，开发可重用 ECU 应用软件的基础。

ZC.MuNiu SOC product is an ECU development solution that complies with Adaptive AUTOSAR. This solution adopts an SOA architecture design, including a suite of tools from system functional design, ECU function mapping and component configuration, to automatic code generation of the runtime environment, providing a set of practical and verified foundations for developing reusable ECU application software.

4 系统框架 SYSTEM FRAMEWORK

系统框架图如下：The system framework diagram is as follows:



知从木牛 SOC 将整个系统分成如下几个部分：

ZC.MuNiu SOC divides the entire system into the following parts:：

- Hardware：硬件层，涉及到具体的芯片开发板；
  Hardware：The hardware layer, which involves specific chip development boards.
- BOOTLOADER：引导层，负责引导 OS 系统；
  BOOTLOADER：The bootloader layer, responsible for booting the OS system.
- OS Kernel： 操作系统内核层，驱动各种外部设备，如 Audio，Bluetooth，Wifi 等，OS Kernel 除了驱动外设外，还负责其他多进程调用，内存管理，磁盘管理等其他 OS 常规功能；
  OS Kernel： The operating system kernel layer, which drives various external devices such as Audio, Bluetooth, Wifi, etc. In addition to driving peripherals, the OS Kernel is also responsible for other OS standard functions such as multi-process calls, memory management, disk management, etc.
- Application：应用层负责具体的功能实现，至下而上分别实现了 HAL，Platform API， SW Runtime FrameWork 和 User App：
  Application：The application layer is responsible for the specific functional

implementation, from bottom to top, it implements HAL, Platform API, SW Runtime Framework, and User App:

- HAL：虚拟层，负责将底层设备抽象，变成标准 API 供上层应用调用；
  HAL：A virtual layer that abstracts the underlying devices into standard APIs for upper-layer applications to call.
- Platform API：Platform API 实现了 Adaptive 平台基础组件，包括十一个 API 和四个常用服务：
  Platform API：The Platform API implements the Adaptive platform's basic components, including eleven APIs and four common services:
  - Time Synchronzation：时间同步，负责系统时钟与外部时钟的同步；
    Time Synchronzation：Responsible for synchronizing the system clock with an external clock.
  - Execution Mangement：执行管理，负责程序的执行以及解决程序的相互依赖；
    Execution Mangement：Manages program execution and resolves program interdependencies.
  - Communication Management：负责通信管理，包括外部通信和内部通信管理；
    Communication Management：Manages communication, including both external and internal communication management.
  - Logging &Tracing：日志和跟踪，分析系统风险；
    Logging &Tracing：Analyzes system risks through logging and tracing.
  - Persitency：访问非易失性存储器的接口；
    Persitency：Interface for accessing non-volatile storage.
  - Identity Acess Management：身份和访问管理，进行风险管理；
    Identity Acess Management：Manages identities and access, conducting risk management.
  - Platform Health Management：平台健康管理，系统负载，状态等管理；
    Platform Health Management：Manages platform health, including system load and status.
  - Core Types：系统通用类和功能，如错误处理和复杂数据类型；
    Core Types：System common types and functionalities, such as error handling and complex data types.
  - Cryptography：通用加密操作和安全密钥管理的接口；
    Cryptography：Interface for general cryptographic operations and secure key management.
  - Diagnostics：诊断相关服务管理；
    Diagnostics：Manages services related to diagnostics.

- RESTFul：负责构建 RESTful 以及特定服务；
  RESTFul：Responsible for building RESTful services and specific services.

四个常用服务：Four common services：

- Update &Configuration Management：处理软件更新请求的服务；
  Update &Configuration Management：A service for handling software update requests.
- State Management：状态管理，负责平台运行状态的所有方面，包括处理传入事件，确定这些事件/请求的优先级以设置相应的内部状态。
  State Management：State management, responsible for all aspects of the platform's operational status, including handling incoming events and determining the priority of these events/requests to set the corresponding internal state.
- Signal To Service Mapping：信号对服务的映射；
  Signal To Service Mapping：Signal to service mapping.
- Network Management：网络管理服务，协调内部协调状态机中基础网络的正常运行和总线睡眠模式之间的转换；
  Network Management：Network management service, coordinating the normal operation of the basic network in the internal coordination state machine and the transition between bus sleep mode.

- SW Runtime Framework：软件运行框架，一般指系统级程序运行环境；
  SW Runtime Framework：Software runtime framework, generally refers to the system-level program running environment.
- User App：用户自定义应用程序，可以在本地运行，也可以远程运行，通过 API 接口与系统交互；
  User App：User-defined applications, which can run locally or remotely, and interact with the system through API interfaces.

5 技术优势 TECHNICAL ADVANTAGE

- 支持 SOME/IP 协议，包括 ECE 保护；
  Support for SOME/IP Protocol: Including ECE protection.
- 支持 DoIP 协议，可以使用 UDSOnIP 进行 OTA 升级；
  Support for DoIP Protocol: Capable of performing OTA updates using UDS on IP.
- 使用 TLS 和 SecOC 增加安全性；
  Increased Security with TLS and SecOC
- 内部通信使用 IPC 机制，以获得更好的性能和简化的配置；
  Employing Inter-Process Communication (IPC) for better performance and simplified configuration.
- 实时检查 TCP 连接，确保服务稳定性；
  Ensuring service stability through continuous monitoring of TCP connections.
- UDP 帧缓冲机制减少网络过载；
  Reducing network overload with a buffering system for UDP frames.
- IP 报文中进行优先级分配，保障高优先级应用的实时性；
  Guaranteeing real-time performance for high-priority applications by assigning priorities within IP packets.
- 可选静态服务发现机制，加速启动；
  Accelerating startup with an optional static service discovery process.
- 支持 XCP 测量和标定；
  Support for XCP Measurement and Calibration
- 支持 AutoIP 机制；
  Support for AutoIP Mechanism