

# 团 体 标 准

T/ITS 0224.2—XXXX

## 车路协同 边缘计算设施 第 2 部分：操作系统

Vehicle-infrastructure cooperative system MEC facility  
—Part 2: Operating system

(征求意见稿)

(本草案完成时间：2026 年 1 月)

— XX — XX 发布

XXXX — XX — XX 实施

中国智能交通产业联盟 发布

中国智能交通产业联盟

# 目 次

前 言 .....	II
1 范围 .....	3
2 规范性引用文件 .....	3
3 术语和定义 .....	3
4 缩略语 .....	4
5 边缘计算操作系统总体要求 .....	4
5.1 系统架构 .....	4
5.2 互操作性要求 .....	5
5.3 系统性能要求 .....	5
6 边缘计算操作系统分层定义要求 .....	5
6.1 内核层要求 .....	6
6.2 硬件设备抽象层要求 .....	6
6.3 计算与通信中间件层要求 .....	7
6.4 服务框架层要求 .....	7
6.5 应用框架层要求 .....	8
6.6 工具链要求 .....	9
6.7 安全与合规要求 .....	9
7 API 定义与服务要求 .....	10
7.1 硬件设备抽象层 API .....	10
7.2 计算与通信中间件层 API .....	12
7.3 服务框架层 API .....	13
7.4 应用框架层 API .....	14
附 录 A （规范性） API 详细结构定义 .....	15
A.1 硬件设备抽象层详细结构定义 .....	15
A.2 服务框架层详细结构定义 .....	18
参 考 文 献 .....	22

## 前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

T/ITS 0224《车路协同 边缘计算设施》拟由三个部分构成。

——第1部分：总体技术要求；

——第2部分：操作系统；

——第3部分：测试方法。

本文件为T/ITS 0224的第2部分。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国智能交通产业联盟（C-ITS）提出并归口。

本文件起草单位：XXX。

本文件主要起草人：XXXX。

# 车路协同 边缘计算设施

## 第 2 部分：操作系统

### 1 范围

本文件规定了车路协同边缘计算操作系统的总体要求、分层定义要求、API定义与服务要求等。本文件适用于城市车路协同场景下边缘计算设施操作系统的开发与设计。

### 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- GB 5768.2—2009 道路交通标志和标线 第2部分：道路交通标志
- GB/T 27967—2011 公路交通气象预报格式
- GB/T 29100—2012 道路交通信息服务 交通事件分类与编码
- GB/T 31024.1—2014 合作式智能运输系统 专用短程通信 第1部分：总体技术要求
- T/ITS 0058—2017 合作式智能运输系统 车用通信系统 应用层及应用数据交互标准
- T/ITS 0224.1—2025 车路协同 边缘计算设施 第1部分：总体要求

### 3 术语和定义

T/ITS 0224.1—2025 界定的以及下列术语和定义适用于本文件。

#### 3.1

**边缘计算操作系统** multiple access edge computing operating system (MEC OS)

部署在边缘计算设施上的基础软件平台，包含硬件资源管理、网络通信、计算调度、服务框架及安全保障等功能。

#### 3.2

**内核层** kernel layer

系统的核心运行支撑环境，包含进程和线程管理、内存管理、文件系统、网络栈、设备驱动、实时补丁、高可用支持等关键要素。

#### 3.3

**硬件设备抽象层** hardware device abstraction layer

对摄像机、激光雷达、毫米波雷达、信号机及其他道路传感器进行统一抽象与驱动管理，提供标准化的设备接入接口，实现硬件异构屏蔽与资源统一调用。

#### 3.4

**计算与通信中间件层** computing and communication middleware layer

负责边缘节点的计算任务调度、通信交互、服务注册等功能，确保高效数据流转与低时延通信，提供统一的通信接口。

#### 3.5

**服务框架层** service framework layer

提供系统核心服务和功能组件的标准化接口，实现应用与底层硬件及基础软件的解耦。支持统一调用、资源管理及服务复用。

#### 3.6

**应用框架层** application framework layer

为业务功能组件提供统一的应用开发接口、生命周期管理机制和运行时支撑，实现应用逻辑的模块化与可扩展性。协调应用与服务框架交互，屏蔽底层复杂性。

### 3.7

#### 组件 component

操作系统内运行具体逻辑的实例，符合操作系统定义的接口，可由框架按照标准接口进行加载和运行。

## 4 缩略语

以下缩略语适用于本文件：

BSM：基本安全消息（Basic Safety Message）

HTTP：超文本传送协议（Hyper Text Transport Protocol）

HTTPS：超文本传送安全协议（Hyper Text Transport Protocol over SecureSocket Layer）

MEC：多接入边缘计算（Multiple Access Edge Computing）

NVMe：非易失性内存主机控制器接口规范（Non-Volatile Memory Host Controller Interface Specification）

OTA：空中下载技术（Over the Air）

POSIX：可移植操作系统接口（Portable Operating System Interface）

RSI：路侧交通信息（Road Side Information）

RSM：路侧单元消息（Road Side Message）

RSU：路侧单元（Road Side Unit）

SSM：感知共享消息（Sensor Sharing Message）

SPAT：信号灯消息（Signal Phase and Timing Message）

UDP：用户数据包协议（User Datagram Protocol）

UFS：通用闪存存储（Universal Flash Storage）

V2X：车载单元与其他设备通信（Vehicle to Everything）

## 5 边缘计算操作系统总体要求

### 5.1 系统架构

车路协同总体架构见T/ITS 0224.1的5.1。

边缘计算操作系统是车路云一体化系统的基础软件平台，功能涵盖硬件资源管理、网络通信、计算调度、服务框架及安全保障等方面，边缘计算操作系统总体架构如图 1 所示：

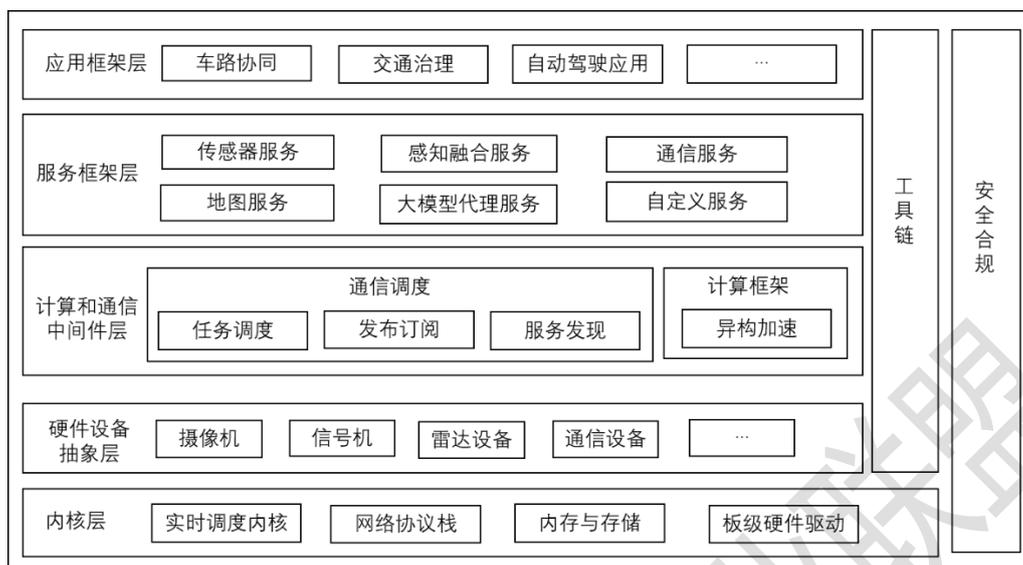


图 1 边缘计算操作系统架构图

边缘计算操作系统各层之间关系如下：

- 内核层：向上提供资源调度、网络传输、内存管理等核心接口；
- 硬件设备抽象层：屏蔽硬件差异，向上提供统一数据接口；
- 计算与通信中间件层：提供计算任务接口，实现任务调度、总线通信及服务协调；
- 服务框架层：提供边缘计算基础服务，通过标准 API 向应用框架提供服务数据；
- 应用框架层：提供服务数据获取接口，基于服务数据实现车路云场景应用；
- 工具链与安全合规：横向贯穿各层，提供全局监控、OTA 升级、安全策略与合规管控等。

## 5.2 互操作性要求

边缘计算操作系统与外部设备和平台之间的通信协议应符合 T/ITS0224.1—2025 的规定，系统内部的互操作性要求如下：

- 硬件设备抽象层及框架层应支持组件的注册、加载与调用，并提供数据的输入或输出接口；
- 系统框架层接口应符合第 7 章的定义，各层组件应实现对应接口并被框架加载；
- 操作系统内部模块应基于通信中间件进行消息交互，数据接口应符合附录 A 的要求；
- 应支持跨平台构建，提供统一的编译工具链、接口文档与调试工具；
- 系统应实现组件解耦与可替换，支持动态加载、插件式架构与服务扩展；
- 应支持第三方应用、模型和算法的快速部署，支持多厂商设备互联互通；
- 应支持容器化部署，支持虚拟化环境。

## 5.3 系统性能要求

边缘计算操作系统性能要求如下：

- 实时性：应支持毫秒级任务调度响应，同节点数据传输时延宜  $\leq 5\text{ms}$ ，跨节点数据传输时延宜  $\leq 10\text{ms}$ ；
- 并发与吞吐量：系统应支持至少 1000 条/秒的消息传输与处理能力，高负载情况下网络数据丢包率  $\leq 0.01\%$ ；
- 可靠性：系统应具备自恢复能力，在单点故障情况下可通过冗余节点实现自动切换，系统可用性  $\geq 99.9\%$ ；
- 可维护性：系统应支持统一日志、监控与远程诊断功能。

## 6 边缘计算操作系统分层定义要求

## 6.1 内核层要求

### 6.1.1 操作系统内核

边缘计算操作系统内核层要求如下：

- a) 内核架构应支持多核并行、实时调度和虚拟化扩展；
- b) 应具备毫秒级中断响应和确定性调度能力；
- c) 应支持CPU、内存、I/O等资源的虚拟化与隔离机制；
- d) 系统应具备异常监控、异常恢复与日志追踪机制；
- e) 应提供内核零拷贝I/O、epoll或io\_uring等机制，降低通信延迟；
- f) 应提供符合POSIX标准的系统调用接口，并支持内核模块扩展。

### 6.1.2 网络协议栈

边缘计算操作系统网络协议栈要求如下：

- a) 应支持IPv4/IPv6、TCP/UDP、HTTP/HTTPS等网络协议；
- b) 应提供SSL/TLS加密、基于防火墙的访问控制和基本入侵防护机制。

### 6.1.3 时间同步

边缘计算操作系统应具备高精度时钟同步能力：

- a) 授时方式：应支持通过全球导航卫星系统（GNSS）或网络进行授时；
- b) 同步协议：当采用网络授时时，应支持网络时间协议（NTP）或精确时间协议（PTP）；
- c) 同步精度：时钟同步误差应优于 0.5 ms（相对于协调世界时 UTC 或指定的主时钟源）。

### 6.1.4 内存与存储管理

边缘计算操作系统内存与存储管理要求如下：

- a) 内存应支持任务级内存空间隔离与动态分配，防止越界访问；
- b) 系统应提供多级缓存一致性管理与NUMA优化；
- c) 应支持NVMe、UFS、SSD等多种介质，并具备文件系统加密、快照与日志恢复机制。

## 6.2 硬件设备抽象层要求

### 6.2.1 硬件设备抽象层

硬件设备抽象层要求如下：

- a) API应符合7.1的规定，数据接口应符合附录A.1的规定；
- b) 符合硬件设备抽象层规范的组件应被服务框架加载作为设备服务并提供数据；
- c) 应支持热插拔与动态更新。

### 6.2.2 RSU 接入

操作系统与RSU的通信协议应符合T/ITS 0224.1—2025 7.2.1的规定。

### 6.2.3 摄像机接入

操作系统与摄像机的通信协议应符合T/ITS 0224.1—2025 7.3.1的规定。

### 6.2.4 激光雷达接入

操作系统与激光雷达的通信协议应符合T/ITS 0224.1—2025 7.3.2的规定。

### 6.2.5 毫米波雷达接入

操作系统与毫米波雷达的通信协议应符合T/ITS 0224.1—2025 7.3.2的规定。

### 6.2.6 信号机接入

应支持符合 GA/T 1743-2020 等国家标准、行业标准规定的接入方式。

### 6.2.7 其他传感器接入

应遵循相关国家标准、行业标准或通用协议。

## 6.3 计算与通信中间件层要求

### 6.3.1 计算中间件

计算中间件是操作系统在硬件与上层服务框架之间的算力抽象与加速层，主要功能是屏蔽硬件差异，提供统一的计算接口和异构加速能力，以支持上层算法、感知、融合等服务模块的高性能运行，计算中间件满足以下要求：

- a) 应支持对 AI 推理、图像预处理、视频编解码、深度学习算子加速等功能模块的封装；
- b) 宜具备算力资源监控与使用统计功能，为系统调度与运维提供基础数据；
- c) 宜与任务调度机制协同工作，实现算力感知的动态任务分配与迁移。

### 6.3.2 任务调度机制

通信中间件应支持统一的任务调度框架，并具备以下功能：

- a) 支持基于优先级的任务调度，可结合资源感知策略进行优化；
- b) 提供任务抽象接口，至少包括任务标识、优先级、依赖关系；
- c) 支持周期性和事件驱动两种任务触发模式。

### 6.3.3 发布/订阅通信机制

为支持高并发、低延迟的数据流传输，中间件应实现标准化的发布/订阅（Pub/Sub）通信机制。具体要求如下：

- a) 应支持基于主题（Channel/Topic）的发布/订阅通信模型，实现数据的解耦与异步传输；
- b) 应提供主题注册、订阅等基本接口，支持通过回调函数处理订阅消息；
- c) 应支持可配置的消息缓冲机制，至少包括消息队列深度配置，防止快速消息丢失；
- d) 应支持多种传输机制适配，包括进程内通信、共享内存、网络协议等，统一接口屏蔽底层差异；
- e) 应采用统一的消息格式（如Protocol Buffers、JSON等），支持序列化与反序列化；
- f) 应提供QoS配置选项，至少包括可靠性和消息保留策略；
- g) 宜支持消息过滤、压缩与访问控制机制。

### 6.3.4 服务注册与发现

中间件应提供统一的服务注册与发现机制，支持跨节点、跨进程的动态服务协同，具体要求如下：

- a) 应提供服务注册与发现接口，支持自动失效检测与下线机制；
- b) 应采用分布式发现机制，不依赖中央管理节点；
- c) 宜实现身份认证与访问控制机制。

## 6.4 服务框架层要求

### 6.4.1 感知融合服务

服务框架层的融合感知服务功能要求如下：

- a) 服务框架应支持端到端和模块化感知融合；
- b) 服务框架应支持符合7.3API定义的感知融合组件的注册、加载和运行；
- c) 应基于底层传感器抽象层接口提供图像输入能力，数据符合附录A.1.2的要求；
- d) 应基于通信中间件提供对外的感知服务数据发送接口，符合数据附录A.2的要求；
- e) 应支持多源传感数据（摄像机、雷达等）的接入与时间同步；
- f) 应保证感知数据的时间戳精度与空间对齐一致性；

- g) 应支持多目标连续跟踪，在感知范围内应保持同一目标 ID 不变；
- h) 宜支持动态感知模型热更新与精度评估。

#### 6.4.2 传感器服务

服务框架层的传感器服务要求如下：

- a) 传感器服务框架应支持符合7.1 API定义的硬件设备组件的注册、加载和运行；
- b) 应提供符合附录A.1的设备数据发送接口；
- c) 应支持设备控制接口或设备数据写入接口。

#### 6.4.3 地图服务

服务框架层的地图服务要求如下：

- a) 应支持高精地图的加载、更新与分发；
- b) 地图数据格式应符合相关国家标准或行业规范；
- c) 应提供地图数据访问标准 API 服务；
- d) 绝对精度不低于50 cm；每100 m相对误差不超过20 cm；
- e) 地图数据访问99分位时延应小于0.1 ms。

#### 6.4.4 通信服务

服务框架层的通信服务要求如下：

- a) 应提供符合标准V2X消息结构（如 BSM、MAP、SPAT、RSI 等）编解码实现；
- b) 应支持消息加密、签名与完整性验证；
- c) 路云数据交互应符合CSAE 295.3-2023的规定。

#### 6.4.5 大模型代理服务

服务框架层大模型代理服务要求如下：

- a) 应提供模型管理接口，支持模型注册、加载、卸载、版本管理；
- b) 宜提供统一的推理调用接口，支持同步、异步、流式等调用模式，接收标准化的输入请求，返回结构化的推理结果；
- c) 可提供模型推理性能监测与统计接口，支持关键指标的实时查询。

#### 6.4.6 自定义服务开发

服务框架层自定义服务开发要求如下：

- a) 系统应提供 SDK、API 及开发模板，支持第三方服务扩展；
- b) 自定义服务API调用应符合硬件设备抽象层数据接口与中间件层接口的要求；
- c) 宜提供开发测试工具链，确保服务兼容性与稳定性。

### 6.5 应用框架层要求

#### 6.5.1 应用框架接口

应用框架接口要求如下：

- a) 应支持符合7.4节API定义的应用组件的注册、加载和运行；
- b) 应提供服务数据获取接口，支持融合感知数据与地图实现车路协同场景应用；
- c) 应基于通信服务提供统一的数据发送接口；
- d) 宜提供应用模板创建、开发、测试发布的完整工具链。

#### 6.5.2 应用部署与生命周期

应用部署与生命周期要求如下：

- a) 应提供应用安装、加载、运行、停止、卸载的全生命周期管理；
- b) 应支持应用健康监测与自愈机制；

- c) 宜支持跨节点任务迁移与灾备恢复。

### 6.5.3 应用安全与隔离

应用安全与隔离要求如下：

- a) 应采用最小权限访问策略；
- b) 应采用资源限额及沙箱机制；
- c) 应用间通信宜经中间件受控通道。

## 6.6 工具链要求

### 6.6.1 模型闭环工具接口

模型闭环工具接口要求如下：

- a) 宜支持智能数据标注、模型训练、验证、部署与数据上传的闭环管理；
- b) 宜支持模型自动测试与回归分析工具；
- c) 宜支持模型配置、模型推理及模型管理接口。

### 6.6.2 OTA 工具接口

OTA工具接口要求如下：

- a) 应支持设备接入、服务、应用及模型的分层 OTA 更新；
- b) 应确保软件更新符合ISO 26262功能安全的要求；
- c) 应提供断点续传、差分更新与版本回滚功能。

### 6.6.3 质量监控与日志工具接口

质量监控与日志工具接口要求如下：

- a) 应提供性能、资源、异常事件的实时采集接口；
- b) 应支持日志分级存储与远程上传；
- c) 宜提供可视化运维监控仪表盘。

### 6.6.4 运维管理工具接口

运维管理工具接口要求如下：

- a) 应支持远程节点管理、任务调度与配置下发；
- b) 应提供报警通知与自动恢复机制；
- c) 应支持与云端管理平台的数据互通。

### 6.6.5 模型算法仓和算力调度

模型算法仓和算力调度要求如下：

- a) 应提供集中式模型与算法仓库，具备权限与版本控制；
- b) 宜支持跨节点异构算力调度。

### 6.6.6 可视化工具

系统应支持感知效果展示、设备监控、配置等可视化功能。

## 6.7 安全与合规要求

边缘计算操作系统安全合规要求如下：

- a) 总体：边缘计算设施应符合T/ITS 0224.1—2025的安全管理相关规定；
- a) 数据存储：静态密码、私钥文件、模型文件等敏感数据宜进行加密存储；
- b) 数据传输：应采用 TLS/SSL 加密与双向认证机制，支持数据脱敏与访问溯源；
- b) 应用安全：应用包应签名验证并具备运行时完整性保护，应用访问系统资源应通过授权机制；
- d) 安全OTA：应确保OTA网络安全，支持设备认证、防降级、升级包加密与校验等安全策略。

## 7 API定义与服务要求

### 7.1 硬件设备抽象层API

#### 7.1.1 信号机抽象层 API

##### 7.1.1.1 功能要求

信号机抽象层 API 用于交通信号灯设备的数据接入与控制交互，其功能要求如下：

- a) 接口应支持信号灯设备的状态采集、数据回传及上层控制指令下发；
- b) 模块应由操作系统框架加载，并通过统一回调机制与系统通信；
- c) 模块应实现初始化、运行、写入设备及状态获取等核心接口。

##### 7.1.1.2 接口定义要求

- a) 信号灯设备 API 应至少包括以下接口：
  - 1) Init: 初始化接口；
  - 2) Start: 主逻辑入口；
  - 3) WriteToDevice: 设备写入接口；
  - 4) GetState: 状态获取接口；
- b) 初始化接口应支持传入配置文件绝对路径，以完成设备通信配置加载；
- c) WriteToDevice接口应支持上层应用发送控制指令，用于实现信号灯控制功能；
- d) GetState接口应返回设备运行状态，供框架或上层监控模块调用；
- e) 模块输出的数据结构应符合 附录A 1.1 的定义要求。

#### 7.1.2 摄像机抽象层 API

##### 7.1.2.1 功能要求

摄像机抽象层 API 用于摄像机设备的数据接入交互，其功能要求如下：

- a) 接口应支持摄像机设备的视频码流数据或者图片数据推送；
- b) 模块应由操作系统框架加载，并通过统一回调机制与系统通信；
- c) 模块应实现初始化等核心接口。

##### 7.1.2.2 接口定义要求

摄像机抽象层接口定义要求如下：

- a) 摄像机设备应提供两种接口标准：
  - 1) 图片数据推送；
  - 2) 视频码流数据推送；
- b) 图片数据推送应实现如下API接口：
  - 1) Init: 初始化接口；
  - 2) CameraImageCallBack: 图片回调接口；
- c) 视频数据推送应实现的API接口：
  - 1) Init: 初始化接口；
  - 2) Start: 设备启动推送视频流；
  - 3) Stop: 设备停止推送视频流；
  - 4) GetParam: 视频流相关参数获取接口；
  - 5) VideoStreamCallBack: 视频流回调接口；
- d) 初始化接口应支持传入配置文件绝对路径，以完成设备通信配置加载；
- e) Start接口应实现开启获取实时视频流数据的功能；
- f) GetParam接口应返回当前推送的视频流相关属性信息，供框架或上层监控模块调用；
- g) 输出的图片数据结构应符合A1.2的定义要求，视频码流数据应包含视频字节流和必要信息。

### 7.1.3 毫米波雷达抽象层API

#### 7.1.3.1 功能要求

毫米波雷达设备抽象层 API 用于毫米波雷达设备的数据接入与控制交互，其功能要求如下：

- a) 接口应支持毫米波雷达设备的状态采集、数据推送及上层数据下发；
- b) 模块应由操作系统框架加载，并通过统一回调机制与系统通信；
- c) 模块应实现初始化、运行、写入设备及状态获取等核心接口。

#### 7.1.3.2 接口定义要求

接口定义要求如下：

- a) 毫米波雷达设备 API 应至少包括以下接口：
  - 1) Init：初始化接口；
  - 2) Start：主逻辑入口；
  - 3) WriteToDevice：设备写入接口；
  - 4) GetState：状态获取接口；
- b) 初始化接口应支持传入配置文件绝对路径，以完成设备通信配置加载；
- c) WriteToDevice接口应支持上层应用发送数据（如：控制指令等），下发数据给雷达设备；
- d) GetState接口应返回设备运行状态，供框架或上层监控模块调用；
- e) 模块输出的数据结构应符合附录A 1.3 的定义要求。

### 7.1.4 激光雷达抽象层API

#### 7.1.4.1 功能要求

激光雷达抽象层 API 用于激光雷达设备的数据接入与控制交互，其功能要求如下：

- a) 接口应支持激光雷达设备的状态采集、数据推送及上层数据下发；
- b) 模块应由操作系统框架加载，并通过统一回调机制与系统通信；
- c) 模块应实现初始化、运行、写入设备及状态获取等核心接口。

#### 7.1.4.2 接口定义要求

- a) 激光雷达设备 API 应至少包括以下接口：
  - 1) Init：初始化接口；
  - 2) Start：主逻辑入口；
  - 3) WriteToDevice：设备写入接口；
  - 4) GetState：状态获取接口；
- b) 初始化接口应支持传入配置文件绝对路径，以完成设备通信配置加载；
- c) WriteToDevice接口应支持上层应用发送数据（如：控制指令等），下发数据给雷达设备；
- d) GetState接口应返回设备运行状态，供框架或上层监控模块调用；
- e) 模块输出的数据结构应符合附录A 1.4的定义要求。

### 7.1.5 RSU 设备抽象层 API

#### 7.1.5.1 功能要求

RSU设备抽象层 API 用于MEC和RSU设备进行V2X数据的双向通信，其功能要求如下：

- a) 接口应支持RSU设备的状态采集、数据推送以及数据接收；
- b) 模块应由操作系统框架加载，并通过统一回调机制与系统通信；
- c) 模块应实现初始化、运行、发送数据给设备及设备运行状态获取等核心接口。

#### 7.1.5.2 接口定义要求

- a) RSU设备 API 应至少包括以下接口：

- 1) Init: 初始化接口;
- 2) Start: 主逻辑入口;
- 3) WriteToDevice: 设备写入接口;
- 4) GetState: 状态获取接口;
- b) 初始化接口应支持传入配置文件绝对路径, 以完成设备通信配置加载;
- c) WriteToDevice接口应支持上层应用发送V2X消息数据给RSU设备;
- d) GetState接口应返回设备运行状态, 供框架或上层监控模块调用;
- e) 模块输出的数据结构应符合附录A1.5的定义要求。

## 7.2 计算与通信中间件层API

### 7.2.1 计算中间件 API

#### 7.2.1.1 功能要求

计算中间件 API应对视频编解码、图像预处理、模型推理等功能模块进行统一封装。

#### 7.2.1.2 接口定义要求

计算中间件 API 应至少包括以下函数接口:

- a) 视频编解码接口
  - 1) Init: 初始化解码器;
  - 2) PutStreamData: 输入视频流;
  - 3) RegisterImageCallback: 注册解码图像回调;
- b) 图像预处理
  - 1) Resize: 图像缩放;
  - 2) ConvertColor: 色彩空间转换;
- c) 模型推理
  - 1) Init: 初始化模型推理引擎;
  - 2) Inference: 模型推理;
  - 3) Unload: 模型卸载。

### 7.2.2 通信中间件 API

#### 7.2.2.1 功能要求

通信中间件抽象层接口支持多种底层通信中间件实现的切换及配置, 其功能要求如下

- a) 统一通信接口应支持两种开发模式:
  - 1) 组件模式: 通过实现抽象适配层接口实现业务组件;
  - 2) 节点模式: 通过实现 Reader/Writer 实现消息订阅与发布;
- b) 通信中间件在运行时对上层保持接口一致;
- c) API 应支持多线程安全的消息传输机制, 并提供回调注册与自动释放机制;
- d) API 应支持结构化配置加载、通道消息发布、订阅、回调处理等基础功能。

#### 7.2.2.2 接口定义要求

- a) 组件模式通信中间件 API 应至少包括以下函数接口:
  - d) Init: 模块初始化接口;
  - e) Proc: 消息处理接口;
  - f) LoadConfig: 配置文件加载接口;
  - g) Send: 发送消息接口;
- b) 节点模式通信中间件 API 应至少包括以下函数接口:
  - 1) CreateReader: 创建订阅者;
  - 2) CreateWriter: 创建发布者;

- c) 框架提供统一的生命周期管理接口，负责节点注册与销毁；
- d) 所有 Reader 与 Writer 对象应由节点 Node 接口创建，框架负责资源自动释放；
- e) 消息发送接口应为线程安全，并在发送失败时返回错误码。

### 7.3 服务框架层API

#### 7.3.1 感知融合服务 API

##### 7.3.1.1 功能要求

感知融合服务API功能要求如下：

- a) 系统应提供端到端感知融合API，用于端到端感知融合模型数据的采集、处理及结构化输出；
- b) 系统应提供单摄像机感知API和多传感器融合API，用于模块化感知融合数据的采集、处理及结构化输出；
- c) 感知融合模块应支持由操作系统框架加载与卸载，并提供统一的初始化与运行接口；
- d) 模块应支持接收摄像机图像数据输入，并输出包含障碍物信息统一结构化数据。

##### 7.3.1.2 接口定义要求

- a) 端到端感知融合API应至少包括以下函数接口：
  - 1) Init：初始化接口；
  - 2) PerceptionFusion：感知融合串行处理接口；
  - 3) PerceptionFusionAsync：感知融合并行处理接口；
  - 4) WaitAsyncResult：感知融合并行结果获取接口；
  - 5) Name：模块名称获取接口；
- b) 模块化感知融合 API 应至少包括以下函数接口：
  - 1) 感知模块API
    - 1. Init：初始化接口；
    - 2. Perception：摄像机感知处理接口；
    - 3. PerceptionAsync：摄像机感知并行处理接口；
    - 4. WaitAsyncResult：摄像机感知并行结果获取接口；
    - 5. Name：模块名称获取接口；
  - 2) 多传感器融合模块API
    - 1. Init：初始化接口；
    - 2. Process：多传感器融合逻辑处理接口；
    - 3. Name：模块名称获取接口；
- c) 感知融合接口应以符合硬件设备抽象层A.1.2摄像机数据接口作为输入，并以 A2.1.1目标检测结构定义作为输出。

##### 7.3.2 地图服务 API

系统应提供地图服务API，用于应用或其他服务调用，地图服务 API 应至少包括以下函数接口：

- a) GetJunctions：获取坐标点附近固定范围的路口信息；
- b) GetLanes：获取坐标点附近固定范围的所有车道；
- c) GetLaneID：获取坐标点所在的车道ID；
- d) GetNearestLane：获取离坐标点最近的车道信息；
- e) GetCrosswalks：获取离坐标点最近的人行道信息。

##### 7.3.3 大模型代理服务 API

系统应提供Agent API，用于系统服务或应用调用大模型能力。Agent API 应至少包括以下函数接口：

- a) ModelList：查询当前平台可用模型列表；

- b) Invoke: 通用调用接口, 支持多模态输入;
- c) Execute: 触发特定Agent任务执行, 支持结果异步回调;
- d) ToolRegister: 注册外部工具或函数, 供Agent调用;
- e) ToolInvoke: 同步或异步执行已注册的外部工具。

## 7.4 应用框架层API

### 7.4.1 功能要求

应用API用于边缘计算平台上应用模块的统一接入、运行与数据交互, 其功能要求如下:

- a) 应用模块应由操作系统框架加载和管理, 支持生命周期控制;
- b) 应用应通过系统回调机制与底层框架进行通信, 实现消息上报和控制指令响应;
- c) 应用输出数据应符合统一的结构化数据格式, 并支持向路侧通信设备或云端平台发送;
- d) 应用的配置参数应通过标准结构传递, 确保跨平台一致性。

### 7.4.2 接口定义要求

应用API接口要求如下:

- a) 应至少包含以下核心接口:
  - 1) Init: 初始化接口;
  - 2) Start: 主逻辑入口;
- b) 初始化接口应支持从框架读取的配置文件路径及安装路径信息;
- c) 组件应基于Start接口实现主逻辑。

附 录 A  
(规范性)  
API详细结构定义

### A.1 硬件设备抽象层详细结构定义

#### A.1.1 消息头

硬件设备消息头内容见表A.1。

表A.1 消息头

字段名	类型	说明
timestamp	uint64	消息产生时间戳(UNIX毫秒级时间戳)：如：曝光时间戳
send_timestamp	uint64	消息发送时间戳(UNIX毫秒级时间戳)
sequence_num	string	设备序列号
id	string	设备编号/设备名称
vendor	string	设备厂商标识
model	string	设备型号
firmware	string	软件或固件版本号
status	枚举	设备运行状态 0：正常 1：异常
index	uint32	消息序列号

#### A.1.2 信号机数据接口

信号机数据接口定义见表A.2~A.4。

表A.2 信号机接入模块数据定义

字段名	类型	说明
header	Header(表A.1)	消息头
light_info_list	TrafficLightBaseData(表A.3)数组	信号机控制的所有灯组的步色信息列表
period	int32	红绿灯方案总周期(单位：秒)
confidence	double	数据置信度(0 - 1之间)
data_source	枚举	0：未知 1：信号采集器 2：信号控制机 3：视觉检测
control_mode	int32	控制模式，符合《GA/T 1743-2020》
raw_data_list	bytes	信号机原始报文帧数据

表A.3 信号机灯组数据

字段名	类型	说明
light_id	int32	灯组编号
light_type	int32	灯组类型，符合《GA/T 1743-2020》
light_status	枚举	灯组状态： 0：未知 1：黑灯/灭灯 2：红灯 3：红闪灯 4：绿灯 5：绿闪灯 6：黄灯

		7: 黄闪灯
count_down	int32	倒计时
right_way_time	int32	获得路权时长, 可通行时长
entry_direction	bytes	进口方向角度, 取值范围[0, 359], 正北方向为起点的顺时针旋转角度, 单位为度
step_info	OneStep Info(表A.4)数组	灯组的整个周期的步色信息

表A.4 灯组色步数据

字段名	类型	说明
light_status	int32	灯组状态, 同表A.3
duration	int32	当前色步时长
time_confidence	int32	当前色步时长的置信度 (0 - 1之间)

## A.1.3 摄像机数据接口

摄像机数据接口定义见A.5。

表A.5 摄像机接入模块数据定义

字段名	类型	说明
header	Header(表A.1)	消息头
raw_data_gpu	uint8	图片GPU地址指针
raw_data_cpu	uint8	图片CPU地址指针
lens_type	枚举	摄像机镜头类型 0: 枪机 1: 鱼眼
mode	枚举	图像模式 0: 未知格式 1: 单通道灰度图 2: 三通道RGB图 3: 三通道BGR图
height	uint32	图像像素高度
width	uint32	图像像素宽度

## A.1.4 毫米波雷达数据接口

毫米波雷达数据接口定义见表A.6~A.8。

表A.6 毫米波雷达接入模块数据定义

字段名	类型	说明
header	Header(表A.1)	消息头
obstacle	Obstacle(表A.7)数组	检测到的障碍物列表

表A.7 毫米波雷达障碍物数据

字段名	类型	说明
position	Ponit(表A.8)	中心点位置坐标 (UTM坐标)
yaw_position	Ponit(表A.8)	雷达为原点坐标系的位置
longitude	double	经度
latitude	double	纬度
altitude	double	海拔
theta	double	障碍物航向角 (弧度值) 东为0, 北为 $\tau/2$ , 南为 $-\tau/2$ , 西为 $\tau$

velocity	Ponit(表A.8)	移动速度（三方向分量），单位m/s
acceleration	Ponit(表A.8)	移动加速度（三方向分量）
length	double	障碍物的长
width	double	障碍物的宽
height	double	障碍物的高
tracking_time	double	障碍物持续跟踪时长
timestamp	double	障碍物实时时间戳
position_covariance	double数组	障碍物位置协方差
velocity_covariance	double数组	障碍物速度协方差
acceleration_covariance	double数组	障碍物加速度协方差
measure_state	枚举	0: 预测的障碍物 1: 测量的障碍物
type	枚举	0: 未知 1: 未知移动物体 2: 未知静态物体 3: 行人 4: 小汽车 5: 厢式货车 6: 运货卡车 7: 公交车 8: 自行车 9: 摩托车 10: 三轮车
lane_id	uint32	车道编号：最左侧为1

表A.8 三维坐标信息

字段名	类型	说明
x	double	X方向信息
y	double	Y方向信息
z	double	Z方向信息

### A.1.5 激光雷达数据接口

激光雷达数据接口定义见表A.9~A.10。

表A.9 激光雷达接入模块数据定义

字段名	类型	说明
header	Header(表A.1)	消息头
is_dense	boolean	True: 表示点云数据中没有无效点，所有点的坐标（如 x、y、z）和其他属性（如强度、反射率）均为有效值。 False: 表示点云数据中可能存在无效点（如因传感器误差、遮挡或距离过远导致的 NaN 值），需在处理时进行检查或过滤。
start_timestamp	uint64	扫描起始时间戳
finish_timestamp	uint64	扫描结束时间戳
width	uint32	点云总宽
height	uint32	点云总高
laser_count	uint32	扫描激光束数量
scanner_direction	枚举	扫描方向： 0: 从上而下扫描 1: 从下而上扫描
point	PointXYZIT(表A.10)数组	所有点列表

表A.10 激光雷达扫描点数据定义

字段名	类型	说明
x	float	点x坐标
y	float	点y坐标
z	float	点z坐标
laser_id	uint32	所属扫描线id
reflectance	uint32	反射率
intensity	uint32	反射强度
confidence	uint32	置信度
stamp	uint64	时间戳
row_number	uint32	所属行号
column_number	uint32	所属列号

## A.1.6 RSU数据接口

RSU数据接口定义见表A.11。

表A.11 RSU接入模块数据定义

字段名	类型	说明
header	Header (表A.1)	消息头
type	枚举	0: 设备状态 1: BSM消息 2: RSM消息 3: SPAT消息 4: MAP消息 5: RSI消息 6: 私有消息 20: SSM消息 21: VIR消息 22: RSC消息 23: RTCM消息 24: PAM消息 25: PSM消息 26: PMM消息 27: VPM消息 40: CIM消息 41: RSCV消息 42: RAM消息 43: SAM消息 44: ISM消息 0xFF: 未知消息
version	枚举	0: CASE 53 - 2017 1: CASE 53 - 2020 2: YDT 3709-2020 3: CASE 315-1-2023
data	bytes	消息数据

## A.2 服务框架层详细结构定义

感知融合服务数据接口定义见A.12~A.16。

表A. 12 感知消息头定义

字段名	类型	说明
timestamp_sec	double	消息发布时间（单位：秒）
module_name	string	模块名称
sequence_num	uint32	消息序列号（每个模块独立计数，从0开始）
lidar_timestamp	uint64	激光雷达时间戳（纳秒）
camera_timestamp	uint64	摄像机时间戳（纳秒）
radar_timestamp	uint64	毫米波雷达时间戳（纳秒）
frame_id	string	消息帧ID
stamp	uint64	消息发布时的系统时间戳（纳秒）

表A. 13感知障碍物集合定义

字段名	类型	说明
header	PerceptionHeader（表A. 12）	消息头
perception_obstacle	PerceptionObstacle（表A. 14）数组	障碍物列表
error_code	枚举	错误码 0: ERROR_NONE 1: ERROR_TF 2: ERROR_PROCESS 3: UNKNOWN

表A. 14感知障碍物定义

字段名	类型	说明
id	int32	障碍物ID
position	Point（表A. 8）	世界坐标系下位置
theta	Double	朝向角
theta_variance	Double	朝向方差
velocity	Point（表A. 8）	速度向量
length	Double	长度（米）
width	Double	宽度（米）
height	Double	高度（米）
polygon_point	Point（表A. 8）数组	障碍物多边形顶点列表
tracking_time	double	持续跟踪时间（秒）
type	枚举	目标大类 0: UNKNOWN 1: UNKNOWN_MOVABLE 2: UNKNOWN_UNMOVABLE 3: PEDESTRIAN 4: BICYCLE 5: VEHICLE
timestamp	double	GPS时间戳（秒）
point_cloud	double数组	点云数据 [x0, y0, z0, ...]
anchor_point	Point（表A. 8）	锚点（稳定参考点）
position_covariance	double数组	3×3位置协方差矩阵（行优先）
bbox2d	BBox2D（表A. 15）	二维检测框
cube_pts8	Point（表A. 8）数组	八点三维包围框
sub_type	SubType	目标子类型 0: UNKNOWN 1: UNKNOWN_MOVABLE 2: UNKNOWN_UNMOVABLE

		3: CAR 4: VAN 5: TRUCK 6: BUS 7: CYCLIST 8: MOTORCYCLIST 9: TRICYCLIST 10: PEDESTRIAN 11: TRAFFICONE 12: SAFETY_TRIANGLE 13: BARRIER_DELINEATOR 14: BARRIER_WATER
measurements	SensorMeasurement (表A.16) 数组	传感器测量信息集合
acceleration	Point (表A.8)	加速度
occ_state	枚举	遮挡状态 0: OCC_UNKNOWN 1: OCC_NONE_OCCLUDED 2: OCC_PARTIAL_OCCLUDED 3: OCC_COMPLETE_OCCLUDED
trunc_state	枚举	截断状态 0: TRUNC_UNKNOWN 1: TRUNC_TRUE 2: TRUNC_FALSE
velocity_converged	bool	速度是否收敛
sub_type_id	int32	子类型ID
sub_type_id_confidence	double	子类型识别置信度

表A.15 二维坐标信息

字段名	类型	说明
xmin	double	X方向最小像素点位置
ymin	double	Y方向最小像素点位置
xmax	double	X方向最大像素点位置
ymax	double	Y方向最大像素点位置

表A.16 传感器测量信息

字段名	类型	说明
sensor_id	string	传感器ID
id	int32	测量对象ID
position	Point (表A.8)	位置坐标
theta	double	朝向角度 (弧度)
length	double	长度 (米)
width	double	宽度 (米)
height	double	高度 (米)
velocity	Point (表A.8)	速度向量 (m/s)
type	枚举	障碍物大类 0: UNKNOWN 1: UNKNOWN_MOVABLE 2: UNKNOWN_UNMOVABLE 3: PEDESTRIAN 4: BICYCLE

		5: VEHICLE
sub_type	枚举	目标子类型 0: UNKNOWN 1: UNKNOWN_MOVABLE 2: UNKNOWN_UNMOVABLE 3: CAR 4: VAN 5: TRUCK 6: BUS 7: CYCLIST 8: MOTORCYCLIST 9: TRICYCLIST 10: PEDESTRIAN 11: TRAFFICCONE 12: SAFETY_TRIANGLE 13: BARRIER_DELINEATOR 14: BARRIER_WATER
timestamp	double	时间戳 (秒)
box	BBox2D(表A. 15)	仅摄像机检测有效的2D框
velocity_converged	bool	速度是否收敛
occ_state_2d	枚举	遮挡状态 0: OCC_UNKNOWN 1: OCC_NONE_OCCLUDED 2: OCC_PARTIAL_OCCLUDED 3: OCC_COMPLETE_OCCLUDED
confidence	float	置信度, 范围[0, 1]

### 参 考 文 献

- [1] GB/T 2312 信息交换用汉字编码字符集 基本集
- [2] GB 5768.5—2017 道路交通标志和标线 第5部分：限制速度
- [3] GB/T 31024.1—2014 合作式智能运输系统 专用短程通信 第1部分：总体技术要求
- [4] CJJ 37—2012 城市道路工程设计规范（2016年版）
- [5] GA/T 1090—2013 天气状况分类与代码
- [6] YD/T 3400—2018 基于LTE的车联网无线通信技术 总体技术要求
- [7] YD/T 3709—2020 基于LTE的车联网无线通信技术 消息层技术要求
- [8] YD/T 3754—2020 基于LTE网络的边缘计算总体技术要求
- [9] YD/T 3839—2021 基于LTE技术的宽带集群通信（B-TrunC）系统（第二阶段）总体技术要求
- [10] 收费公路联网收费技术要求（交通运输部（2007）35号）
- [11] 电子收费 单片式车载单元（OBU）技术要求（交通运输部（2019）35号）

中国智能交通产业联盟

中国智能交通产业联盟

标准

车路协同 边缘计算设施 第 2 部分：操作系统

T/ITS XXXX.2-20XX

北京市海淀区西土城路 8 号（100088）

中国智能交通产业联盟印刷

网址：<http://www.c-its.org.cn>

2026 年 X 月第一版 2026 年 X 月第一次印刷