MEC需求分析与系统设计

编制： 王 科 威

审核：

版本： V0.0.1

二〇二一年十二月二十五日

目录

[修改历史 I](#_Toc28961)

[一 引言 1](#_Toc24726)

[1.1 编写目的 1](#_Toc18680)

[1.2 背景 1](#_Toc14149)

[1.3 定义 1](#_Toc21239)

[1.4 参考资料 1](#_Toc30023)

[二 基本约定 1](#_Toc29331)

[2.1 程序名称及ID 1](#_Toc16551)

[2.2 端口号 2](#_Toc27759)

[2.3 运行命令 2](#_Toc12091)

[2.4 目录配置 2](#_Toc15325)

[三 需求分析 3](#_Toc24724)

[3.1 功能需求 3](#_Toc30091)

[3.2 非功能需求 6](#_Toc17539)

[四 系统设计 6](#_Toc10359)

[4.1 系统架构图 6](#_Toc23219)

[4.2 系统部署图 7](#_Toc30623)

[4.3 网络结构图 7](#_Toc23596)

[4.4 URL格式定义 7](#_Toc14852)

[五 模块设计 8](#_Toc1457)

[5.1 XMQ模块 8](#_Toc7164)

[5.2 XMQ客户端模块 10](#_Toc12891)

[六 产品授权 11](#_Toc21481)

[七 附表 11](#_Toc12269)

# 修改历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 说明 | 作者 | 审核 | 日期 |
| V0.0.1 | 初稿 | 王科威 |  | 2021-12-25 |

# 引言

## 编写目的

## 背景

在现有项目基础上进行第二次迭代。

## 定义

|  |  |
| --- | --- |
| 缩写 | 全称 |
|  |  |

## 参考资料

# 基本约定

## 程序名称及ID

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **程序名称** | **ID** | **简述** |
| 核心部分 | | | |
| 1 | xmq\_host\_server | xmq\_host\_server\_ + uuid | XMQ的ID标识必须结合UUID，用于在交通协作节点拓扑结构中唯一标识 |
| 2 | lidar\_point\_cloud\_cache\_server | lidar\_point\_cloud\_cache\_server | 雷达点云数据缓存服务 |
| 3 | traffic\_break\_law\_publish\_server | traffic\_break\_law\_publish\_server | 交通违法事件发布服务 |
| 4 | vehicle\_drive\_warning\_publish\_server | vehicle\_drive\_warning\_publish\_server | 车辆行驶预警事件发布服务 |
| 5 | vehicle\_drive\_tip\_publish\_server | vehicle\_drive\_tip\_publish\_server | 车辆行驶服务信息发布服务 |
| 核心扩展部分 | | | |
| 1 |  |  |  |
| 外部设备接入 | | | |
| 1 | dvs\_host\_server | dvs\_host\_server | DVS设备接入服务 |
| 2 | lidar\_host\_server | lidar\_host\_server | 雷达设备接入服务 |
| 3 | sensor\_host\_server | sensor\_host\_server | 传感器设备接入服务 |
| 4 | traffic\_cooperate\_server | traffic\_cooperate\_server | 交通协作设备接入服务 |
| 上位机 | | | |
| 1 | 单上位机节点 | UUID | 不同上位机应用程序生成不同的UUID标识 |

## 端口号

|  |  |  |
| --- | --- | --- |
| 数据交互 | | |
| ***序号*** | ***程序名*** | ***协议与端口号*** |
| 1 | xmq\_host\_server | TCP/60531 |
| 2 | dvs\_host\_server | TCP/60820 |
| 数据发布 | | |
| ***序号*** | 发布ID | ***协议与端口号*** |
| 3 | traffic\_break\_law\_publish | TCP/53101 |
| 4 | vehicle\_drive\_warning\_publish | TCP/53102 |
| 5 | vehicle\_drive\_tip\_publish | TCP/53103 |
| 6 | lidar\_point\_cloud\_data\_publish | TCP/53150 |

## 运行命令

## 目录配置

### 程序文件

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

### 配置文件

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

### 日志

* Windows

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

* Linux

|  |  |
| --- | --- |
| 目录 | 描述 |
|  |  |

# 需求分析

## 功能需求

### 构建交通协作节点拓扑图

#### 功能描述

当前节点服务启动时读取本地MAP配置文件，获取与当前节点关联的下一级节点信息；下级节点通过注册/心跳的方式和上级节点保持一致，上级节点通过3×30s的时间检测下级节点的在/离线状态，对于离线状态的下级节点产生离线事件推送以告知需要关注该事件的用户。下级节点每间隔10s向上级节点发送下级节点单车道车流量流入信息，下级节点每间隔30s向上级节点发送下级节点所有车道车流量流出统计信息，所有流出和流入信息都已事件推送以告知需要关注该事件的用户。

#### 功能分解

1. 各级节点的服务程序启动时必须解析本地的MAP配置文件，获取需要各自需要关注的下级节点；
2. 下级节点服务启动成功后，通过单独的链路与上级节点进行连接；
3. 连接建立成功后，下级节点每间隔30s向上级节点发送注册和心跳消息，上级节点负责监视下级节点的在/离线状态；
4. 超过3×30s时间未检测到注册/心跳消息的下级节点，上级节点将产生离线事件推送；
5. 上级节点接收到离线状态的下级节点注册/心跳消息后，产生在线事件推送；
6. 下级节点每间隔10s向上级节点发送该节点单车道车流流入信息；
7. 下级节点每间隔30s向上级节点发送该节点所有车道车流流出信息；

### 实时图形化配置数据接入设备

#### 功能描述

实时图形化以独立应用程序执行，通过MEC平台系统提供的信令客户端和流媒体客户端接入到平台系统中。应用启动时主动从MEC平台系统中加载已配置成功的设备信息。使用者通过操作图形界面完成对系统接入设备的相关配置操作，包括设备的新增、删除和修改等。接入设备的类型包括视频设备、雷达设备和传感器设备等。新增成功的设备信息将被持久化存储到MEC平台系统中。

#### 功能分解

1. MEC平台系统信令客户端接入；
2. MEC平台系统流媒体客户端接入；
3. 接入MEC平台系统成功后，主动从MEC平台系统中读取设备持久化数据，并将读取的设备信息配置到相应的设备接入模块；
4. 使用UI界面将主动获取的接入设备信息以树型节点形式进行显示；
5. 使用UI界面新增接入设备相关信息，如串口号、地址码、校验码、摄像机个数、IP、端口、登录用户和密码等；
6. 使用UI界面操作树型节点编辑接入设备相关信息，并将编辑后的信息发送到设备接入模块和数据持久化模块；
7. 使用UI界面操作树型节点删除接入设备，并将编辑后的信息发送到设备接入模块和数据持久化模块。

### 实时图形化标定

#### 功能描述

实时图形化以独立应用程序执行，通过MEC平台系统提供的信令客户端接入到平台系统中。应用启动时主动从MEC平台系统中加载已配置成功的标定数据配置。使用者通过操作图形界面完成对标定数据的相关配置操作，包括标定数据的新增、删除和修改，标定数据的类型为事件检测、道路信息配置、GPS信息配置、车速线、事件播报区域和设备位置信息等，标定数据在检测结果发布时作为辅助信息一起推送。

#### 功能分解

1. MEC平台系统信令客户端接入；
2. MEC平台系统流媒体客户端接入；
3. 接入MEC平台系统成功后，主动从MEC平台系统中读取标定配置持久化数据，并将读取的标定数据配置到相应的协同设备接入模块；
4. 使用UI界面将主动获取的标定数据结合实时图像信息同步的形式进行显示；
5. 使用UI界面新增标定数据，事件检测、道路信息配置、GPS信息配置、车速线、事件播报区域和设备位置信息等；
6. 使用UI界面操作标定画线编辑标定数据，并将编辑后的数据发送到协同设备接入模块；
7. 使用UI界面操作标定画线删除标定数据，并将编辑后的信息发送到数据协同设备接入模块。

### 实时图形化检测算法配置

#### 功能描述

实时图形化以独立应用程序执行，通过MEC平台系统提供的信令客户端和流媒体客户端接入到平台系统中。应用启动时主动从MEC平台系统中加载已配置成功的检测算法配置。使用者通过操作图形界面完成对检测算法的相关配置操作，包括检测算法的新增、删除和修改，以及检测算法与实时流数据的关联和检测结果发布类型等。检测算法的类型为行人区域闯入检测，检测结果发布类型为交通违法事件推送类型。

#### 功能分解

1. MEC平台系统信令客户端接入；
2. MEC平台系统流媒体客户端接入；
3. 接入MEC平台系统成功后，主动从MEC平台系统中读取检测算法配置持久化数据，并将读取的检测算法信息配置到相应的数据分析模块；
4. 使用UI界面将主动获取的接入设备信息以列表形式进行显示；
5. 使用UI界面新增检测算法相关信息，如检测类型、检测阈值、摄像机关联、检测结果推送关联等；
6. 使用UI界面操作列表编辑检测算法相关信息，并将编辑后的信息发送到数据分析模块和数据持久化模块；
7. 使用UI界面操作列表删除检测算法，并将编辑后的信息发送到数据分析模块和数据持久化模块。

### 实时图形化预警信息显示

#### 功能描述

实时图形化以独立应用程序执行，通过MEC平台系统提供的信令客户端接入到平台系统中。应用启动时主动向MEC平台系统订阅交通违法事件推送。使用者通过操作图形界面显示推送的事件相关信息，其中包括事件图片、检测目标标记和文字描述等。

#### 功能分解

1. MEC平台系统信令客户端接入；
2. 接入MEC平台系统成功后，主动向MEC平台系统订阅交通违法事件推送；
3. 使用UI界面以摄像机为单位实时显示交通违法事件推送信息，可同时支持多窗口显示多个摄像机的推送信息。

### 弱势交通参与者危险区域闯入检测

#### 功能描述

弱势交通参与者危险区域闯入检测以独立模块运行。使用者操作可视化界面或者直接输入命令通过局域网连接平台服务。平台连接成功后，使用者通过界面或者直接输入命令向平台服务配置检测功能所需要关联的视频设备、摄像机和雷达设备等数据源信息以及输入源位置标定信息；配置数据源信息完成后，使用者操作可视化界面或者直接输入命令在相应摄像机监视区域内设置危险检测区域；区域设置成功后，使用者操作可视化界面或直接输入命令向平台服务发送启动检测运算请求并接收启动检测运算应答。

启动检测运算成功后，从平台服务内部获取摄像机实时视频流数据，将获取的视频流数据进行解码和数据转换，再将转换后的数据传输给检测算法进行运算。当算法检测到弱智交通参与者进入检测区域时，对检测目标进行跟踪并触发危险区域闯入事件。

当事件被触发时，MEC服务将检测结果推送给订阅节点，推送内容包括检测图片和行人位置，订阅者可依据该数据进行可视化展示；于此同时，MEC服务将检测结果以及相关联的实时视频数据进行数据持久化处理，且对触发时间点的前后30秒实时视频数据进行存储，使用者可操作可视化界面或直接输入命令获取到已持久化处理的所有或部分数据内容。

#### 功能分解

1. 使用者端需要通过3条网络链路与MEC服务进行通信；其中，第一条链路负责信令类数据交互，第二条链路负责单向订阅类数据获取，第三条链路负责视频流数据和控制信令的双向交互；
2. MEC服务提供IP摄像机接入服务，通过该服务可实现IP摄像机进行增加、删除、修改、登录、注销、设备信息获取、设备信息配置、视频流数据获取等功能；
3. MEC服务提供雷达接入服务，通过该服务可对雷达增加、删除、修改、连接、断开、目标位置获取、目标类型获取、目标尺寸获取等功能；
4. MEC服务提供关系型数据库管理服务，通过该服务可对摄像机、雷达、基本配置类信息等数据进行持久化存储和数据读写操作；
5. MEC服务提供实时视频流数据存储管理服务，通过该服务支持时间长度可配置的预录和延迟存储功能；
6. MEC服务提供视频流数据PS到ES格式转换，以及转换后ES组包分析；
7. MEC服务提供对不同设备接入的统一数据结构化封装，以使得在内部交互的数据保持统一的数据结构；
8. MEC服务提供基于计算机视觉技术的行人目标检测算法，并在检测结果中提供目标的检测信息；
9. MEC服务提供基于雷达信号处理的行人目标检测算法，并在检测结果中提供目标的检测信息；
10. 行人目标检测结果由MEC服务主动推送给订阅者，并触发检测事件以供视频数据进行存储。

### 弱势交通参与者实时碰撞预警发布

#### 功能描述

使用者操作可视化界面或者直接输入命令完成[3.1.3弱势交通参与者危险区域闯入检测](#_弱势交通参与者危险区域闯入检测)所描述的配置后，再通过可视化界面或者直接输入命令向平台服务发送订阅车辆行驶预警事件请求，事件订阅成功后，使用者可实时接收平台在事件发生时的场景图片和事件标记信息等。可视化界面通过界面显示实时事件场景图片并描绘事件标记。对于不再需要关注的事件，使用者可以操作可视化界面或者直接输入命令向平台服务发送取消订阅车辆行驶预警事件请求。

#### 功能分解

1. 使用者操作可视化界面或者直接输入命令行获取平台支持的事件订阅信息；
2. 使用者选择车辆行驶预警事件类型，向平台发送事件订阅请求，并接收事件订阅应答；
3. 使用者被动接收平台推送的实时事件通知，通知内容包括实时事件场景图片和事件标记信息；
4. 可视化界面通过界面显示实时事件场景图片并描绘事件标记；
5. 使用者操作可视化界面或者直接输入命令向平台发送取消事件订阅请求，并接收事件取消订阅应答。

### 车辆违停检测

#### 功能描述

使用者操作可视化界面或者直接输入命令通过网络连接上MEC服务；连接成功后可向MEC服务添加需要使用的IP摄像机或雷达设备，也可以对已有的设备进行编辑；可视化界面向使用者提供实时视频流播放，使用者操作可视化界面或者直接输入命令在所见范围内设置检测区域；检测区域设置成功后，使用者继续操作可视化界面或直接输入命令将IP摄像机或雷达设备和车辆检测、车牌识别算法进行关联或取消关联设置，关联设置成功后，使用者操作可视化界面或者输入命令向MEC服务订阅事件检测结果通知，当车辆在检测区域内停止不动时触发检检测事件。当事件被触发时，MEC服务将检测结果推送给订阅节点，推送内容包括检测图片、车牌信息和车辆位置，订阅者可依据该数据进行可视化展示；于此同时，MEC服务将检测结果以及相关联的实时视频数据进行数据持久化处理，且对触发时间点的前后30秒实时视频数据进行存储，使用者可操作可视化界面或直接输入命令获取到已持久化处理的所有或部分数据内容。

#### 功能分解

1. 使用者端需要通过3条网络链路与MEC服务进行通信；其中，第一条链路负责信令类数据交互，第二条链路负责单向订阅类数据获取，第三条链路负责视频流数据和控制信令的双向交互；
2. MEC服务提供IP摄像机接入服务，通过该服务可实现IP摄像机进行增加、删除、修改、登录、注销、设备信息获取、设备信息配置、视频流数据获取等功能；
3. MEC服务提供雷达接入服务，通过该服务可对雷达增加、删除、修改、连接、断开、目标位置获取、目标类型获取、目标尺寸获取等功能；
4. MEC服务提供关系型数据库管理服务，通过该服务可对摄像机、雷达、基本配置类信息等数据进行持久化存储和数据读写操作；
5. MEC服务提供实时视频流数据存储管理服务，通过该服务支持时间长度可配置的预录和延迟存储功能；
6. MEC服务提供视频流数据PS到ES格式转换，以及转换后ES组包分析；
7. MEC服务提供对不同设备接入的统一数据结构化封装，以使得在内部交互的数据保持统一的数据结构；
8. MEC服务提供基于计算机视觉技术的车辆目标检测和车牌提取与识别算法，并在检测结果中提供目标的检测信息；
9. MEC服务提供基于雷达信号处理的车辆目标检测算法，并在检测结果中提供目标的检测信息；
10. 行人目标检测结果由MEC服务主动推送给订阅者，并触发检测事件以供视频数据进行存储。

### 激光雷达设备数据接入

#### 功能描述

使用者操作可视化界面或者直接输入命令通过网络连接上MEC服务；连接成功后可向MEC服务添加需要使用的激光雷达设备，也可以对已有的设备进行编辑；

MEC接收激光雷达结构化数据，将结构化数据转换为json格式的VSD(VRC SensorData )数据，推送给订阅节点。

#### 功能分解

1. 使用者端需要通过2条网络链路与MEC服务进行通信；其中，第一条链路负责信令类数据交互，第二条链路负责单向订阅类数据推送;
2. MEC服务提供激光雷达接入服务，通过该服务可实现激光雷达进行增加、删除、修改、结构化数据获取、转换等功能；
3. 激光雷达配置参数包括：激光雷达IP地址、通信端口、该激光雷达对应的RSU ID列表等。
4. MEC服务提供关系型数据库管理服务，通过该服务可对激光雷达配置类信息等数据进行持久化存储和数据读写操作；
5. MEC服务提供将VSD推送给订阅节点的功能。

### 实时视频设备数据接入

#### 功能描述

实时视频设备数据接入以独立模块运行，负责管理实时视频数据设备的登录、注销、摄像机信息获取、实时流打开和关闭等操作，并按照使用者配置信息可将不同厂商的实时视频设备数据通过厂商SDK汇集到模块中。无论使用如何操作设备或实时流数据，模块都必须保证模块到设备之间有且仅有一次登录、注销、实时流打开和关闭等操作。汇集到模块的实时视频数据由模块负责将PS格式的实时视频流数据转换为PES流数据。于此同时，转换后的PES流数据被主动发送到接收端，为保证PES流数据的高效传输，同一摄像机的是视频流数据只向接收端发送一次。

#### 功能分解

1. 模块使用XMQ服务提供的通信客户端与XMQ服务进行通信；
2. 模块使用XMS服务提供的通信客户端与XMS服务进行通信；
3. 模块使用厂商SDK与物理设备进行通信；
4. 模块接收并处理设备增、删、改、查等操作请求，并管理设备资源；
5. 模块使用厂商SDK对物理设备进行登录和注销操作，每个设备仅登录或注销一次；
6. 模块对设备的登录和注销操作结果，由模块向请求端发送应答；模块登录设备成功后，主动查询设备的摄像机信息，登录成功的操作结果中必须包含设备的摄像机信息；
7. 模块登录设备成功后，主动打开设备的摄像机实时视频流，并负责接收实时视频帧数据；
8. 每个摄像机实时流数据通道和XMS服务构建独立的数据交互通道，每个通道上的实时视频帧只推送一次；
9. 接收到的实时视频帧数据通过PS流解析为PES流数据；
10. PES流数据通过数据格式化封装接口进行统一的帧数据封装并发送到XMS；
11. 对设备的修改和删除操作，模块负责相应的摄像机和实时流通道的重构操作。

### 结构化感知数据融合

#### 功能描述

使用者操作可视化界面或者直接输入命令通过网络连接上MEC服务；连接成功后可向MEC服务配置该服务，配置选项包括是否旁路该融合功能以及其他需要的选项；

该服务提供多个雷达、多个摄像头、多个雷达和摄像头之间的感知数据融合，融合包括: 合并目标感知数据，感知数据去除重合目标数据。

该服务将融合后的json格式的VSD(VRC SensorData )数据，推送给订阅节点。

#### 功能分解

1. 使用者端需要通过3条网络链路与MEC服务进行通信；其中，第一条链路负责信令类数据交互，第二条链路负责单向订阅类数据获取;第三条链路负责单向发布数据。
2. MEC服务根据配置选项决定，是启动数据融合功能，还是旁路融合数据。
3. MEC服务提供多雷达融合服务，订阅多个雷达数据(VSD)，对多个雷达数据进行融合，合并多个雷达的目标数据、去除重合的目标数据、将融合的数据推送给订阅节点。
4. MEC服务提供雷达和视频融合服务，订阅雷达和视频数据(VSD)，对雷达和视频数据进行融合，合并雷达和视频的目标数据、去除重合的目标数据、将融合的数据推送给订阅节点。
5. MEC服务提供多路视频目标数据融合服务，订阅多路视频感知目标数据(VSD)，多路视频数据进行融合，合并多路视频的目标数据、去除重合的目标数据、将融合的数据推送给订阅节点。

## 非功能需求

# 系统设计

## 系统架构图



## 系统部署图

## 网络结构图



## URL格式定义

### 信令格式定义

*信令名称://目的地址 [ /路由地址/... [ ? 参数名=参数值 [ & ...] ] ]*

### 信令名称定义

|  |  |
| --- | --- |
| ***命令名称*** | ***说明*** |
| register | 模块注册 |
| query | 模块查询 |
| config | 模块参数配置 |
| info | 数据交互 |

### 流格式定义

*命令名称://设备ID [ ? 参数名=参数值 [ & ...] ]*

### 流命令定义

|  |  |
| --- | --- |
| ***命令名称*** | ***说明*** |
| realplay | 实时流 |
| playback | 回放流 |
| download | 下载流 |
| control | 流控，仅对playback和download有效 |

## 流数据格式

|  |  |  |  |
| --- | --- | --- | --- |
| ***字段名*** | ***类型*** | ***大小*** | ***默认值*** |
| Flag | int | 4Bytes | 0xFF050301 |
| Data type | int | 4Bytes |  |
| Stream type | int | 4Bytes |  |
| Frame type | int | 4Bytes |  |
| Frame size | int | 4Bytes |  |
| Frame sequence | long long | 8Bytes |  |
| Frame timestamp | long long | 8Bytes |  |
| Frame data | | | |

# 模块设计

## XMQ模块(xmq\_host\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 所有模块间信令数据交互枢纽，所有的信令消息都必须经过该模块转发；
3. 使用字符串类型的ID标识来区分不同模块，功能模块ID标识定义参见2.1[程序名称及ID](#_程序名称及ID)；
4. 接收其他模块的注册请求消息，超过3\*30（s）无注册时间戳更新的模块执行自动清理；
5. 接收其他模块的查询请求，以获取所有在线模块信息；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| XMQ-001 | 启动本地监听 | 运行模块服务进程，打开TCP监听端口号60531，监听地址0.0.0.0。 |  |
| XMQ-002 | 接收其他模块注册请求 | 异步接收模块注册请求，内部对注册的模块ID做持久化处理。 | URL示例：  *register://dvs\_host\_server?timestamp=\*[&sequence=\*]* |
| XMQ-003 | 发送模块注册应答 | 异步发送注册应答。 | URL示例：  *register://xmq\_host\_server?timestamp=\*[&sequence=\*]* |
| XMQ-004 | 模块状态管理 | 1. 接收注册请求后先判断模块ID是否存在，不存在则新增模块，存在则更新模块注册时间戳； 2. 应用进程启动后开始每间隔30s对注册模块的时间戳进行检查，如果当前时间与模块注册时间间隔大于90s，则认为模块离线，删除模块记录。 |  |
| XMQ-005 | 接收模块查询请求 | 异步接收模块查询请求。 | URL示例：  *query://xmq\_host\_server[?&sequence=\*]* |
| XMQ-006 | 发送模块查询应答 | 异步发送模块查询应答。应答消息中去除发送模块的信息。 | URL示例：  *query://dvs\_host\_server[?name=\*[&name=\*[&sequence= \*]]]* |
| XMQ-007 | 信令消息转发 | 信令名称不是*register*的请求和应答一律转发。转发过程不改变路径和参数列表。 | URL示例：  *config://dvs\_host\_server?timestamp=\*[&sequence=\*]* |

## XMQ客户端模块(libxmqhostclient)

### 模块说明

1. 包含于应用程序进程内；
2. 所有与平台系统交互的信令消息都只能使用该模块；
3. 使用应用程序进程指定的字符串类型的ID标识来标识自身；
4. 模块创建成功后，每间隔30s主动发送一次注册请求，超过3\*30（s）无注册应答时间戳更新则通知应用程序；
5. 接收平台系统的注册应答后主动发送一次查询请求；
6. 接收平台系统的查询应答并通知应用程序；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| XMQC-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| XMQC-002 | 发送注册请求 | 模块创建成功后，每间隔30s主动发送一次注册请求。 | URL示例：  *register://dvs\_host\_server?timestamp=\*[&sequence=\*]* |
| XMQC-003 | 接收注册应答 | 超过3\*30（s）无注册应答时间戳更新则通知应用程序。 | URL示例：  *register://xmq\_host\_server?timestamp=\*[&sequence=\*]* |
| XMQC-004 | 发送查询请求 | 接收平台系统的注册应答后主动发送一次查询请求。 | URL示例：  *query://xmq\_host\_server[?&sequence=\*]* |
| XMQC-005 | 接收查询应答 | 接收平台系统的查询应答并通知应用程序。 | URL示例：  *query://dvs\_host\_server[?name=\*[&name=\*[&sequence= \*]]]* |
| XMQC-006 | 其他信令消息发送与接收 | 信令封装和解析格式遵照[4.4.1格式定义](#_格式定义)。 |  |

## DVS设备接入模块(dvs\_host\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载设备厂商第三方SDK库与物理设备进行数据交互；
3. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互;
4. 模块接收应用程序发送的设备新增、删除、查询请求等消息并应答操作结果;
5. 模块负责有且仅有一次相同设备的登录和注销操作；
6. 每个登录成功的设备由模块负责主动获取摄像机信息，并为每个摄像机建立一个实时流数据分发通道；
7. 模块负责有且仅有一次相同摄像机实时流的打开和关闭操作；
8. 模块接收应用程序发送的实时流播放和停止播放请求消息并应答操作结果；
9. 模块接收应用程序发送的物理设备查询请求消息并应答操作结果；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| DVS-001 | 连接平台系统 | 应用程序指定ID标识、平台系统IP地址和端口号连接平台系统。 |  |
| DVS-002 | 启动本地监听 | 运行模块服务进程，打开TCP监听端口号60820，监听地址0.0.0.0。 |  |
| DVS-003 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| DVS-004 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| DVS-005 | 接收DVS设备信息列表查询请求 | 模块接收DVS设备信息列表查询请求 | URL示例：  *config://dvs\_host\_server?command=query* |
| DVS-006 | 发送DVS设备信息列表查询应答 | 模块发送DVS设备信息列表查询应答。 | URL示例：  *config://”requester”?command=query[&dvs=1\_192.168.0.100\_4\_test[&dvs=\*]]*  dvs字段参数定义示例：*dvs=1\_192.168.0.100\_4\_test*  字段值以“\_”作为分隔符，第一部分表示设备ID，第二部分表示设备IP，第三部分表示摄像机个数，第四部分表示设备名称 |
| DVS-007 | 接收设备新增请求 | 接收应用程序的设备新增请求。 | URL示例：  *config://dvs\_host\_server?command=add&ip=192.168.0.100&port=8000&user=admin&passwd=Vrc123456&name=test* |
| DVS-008 | 发送设备新增应答 | 发送设备新增应答。 | URL示例：  *config://”requester”?command=add&error=0[&dvs=1\_192.168.0.100\_4\_test]* |
| DVS-009 | 接收设备删除请求 | 接收应用程序的设备删除请求。 | URL示例：  *config://dvs\_host\_server?command=remove&id=1* |
| DVS-010 | 发送设备删除应答 | 发送设备删除应答。 | URL示例：  *config://”requester”?command=remove&error=0&id=1* |
| DVS-011 | 设备数据持久化 | 模块对配置的物理设备信息进行内存数据持久化。 |  |
| DVS-012 | 接收实时视频流打开请求 | 接收应用程序的实时流打开请求。 | URL示例：  *realplay://1?command=1&channel=1&stream=0*  路径部分表示设备ID；  参数说明：  command为1表示打开，0表示关闭；  channel表示通道索引；  stream表示码流类型，取值范围[0, 10] |
| DVS-013 | 发送实时视频流打开应答 | 发送实时视频流打开应答。 | URL示例：  *realplay://1?command=1&channel=1&stream=0&error=0* |
| DVS-014 | 接收实时视频流关闭请求 | 接收应用程序的实时流关闭请求。 | URL示例：  *realplay://1?command=0&channel=1&stream=0* |
| DVS-015 | 发送实时视频流关闭应答 | 发送实时视频流关闭应答。 | URL示例：  *realplay://1?command=0&channel=1&stre6m=0&error=0* |
| DVS-016 | 实时视频流链路异常处理 | 模块主动监测本地或远程链接异常事件捕获，对于产生的异常对实时流会话进行销毁。 |  |
| DVS-017 | 实时流复用 | 模块对相同设备、通道和码流类型的实时流数据进行复用传输，保证1对N的扇出。 |  |
| DVS-018 | PS流格式解析 | 模块对接收到的摄像机数据进行PS流格式解析，输出为PES流格式。 |  |

## XMS客户端模块(libxmshostclient)

### 模块说明

1. 包含于应用程序进程内；
2. 所有与平台系统视频流相关的数据交互都只能使用该模块；
3. 模块使用应用程序进程指定的参数与平台系统建立视频流传输会话；
4. 模块建立应用程序会话创建参数与会话ID间的映射关系；
5. 模块负责视频流传输会话的内存数据持久化管理；
6. 模块发送视频流传输会话的打开请求；
7. 模块接收视频流传输会话的打开应答并通知应用程序；
8. 模块发送视频流传输会话的关闭请求；
9. 模块接收视频流传输会话的关闭应答并通知应用程序；
10. 模块捕获视频流传输会话异常并通知应用程序。

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| XMSC-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| XMSC-002 | 创建实时视频流传输会话 | 模块销毁应用程序会话创建参数与会话ID间的映射关系 |  |
| XMSC-003 | 销毁实时视频流传输会话 | 模块创建应用程序会话创建参数与会话ID间的映射关系 | 销毁前必须先关闭打开的实时流。 |
| XMSC-004 | 发送实时视频流打开请求 | 发送实时视频流打开请求。 | URL示例：  *realplay://1?command=1&channel=1&stream=0*  路径部分表示设备ID；  参数说明：  command为1表示打开，0表示关闭；  channel表示通道索引；  stream表示码流类型，取值范围[0, 10] |
| XMSC-005 | 接收实时视频流打开应答 | 接收实时视频流打开应答。 | URL示例：  *realplay://1?command=1&channel=1&stream=0&error=0* |
| XMSC-006 | 发送实时视频流关闭请求 | 发送实时视频流关闭请求。 | URL示例：  *realplay://1?command=0&channel=1&stream=0&* |
| XMSC-007 | 接收实时视频流关闭应答 | 接收实时视频流关闭应答。 | URL示例：  *realplay://1?command=0&channel=1&stream=0&error=0* |
| XMSC-008 | 视频数据封装和解析 | 数据封装和解析格式遵照[4.5流数据](#_流数据格式)格式。 |  |

## Lidar设备接入模块(lidar\_host\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互；
3. 模块内加载设备厂商第三方SDK库与物理设备进行数据交互；
4. 模块接收应用程序发送的设备新增、删除、查询请求等消息并应答操作结果;
5. 模块负责有且仅有一次相同设备的登录和注销操作；
6. 模块每间隔100ms获取一次物理设备发送的点云数据；
7. 模块发送单词点云数据到系统平台的雷达点云数据缓存模块。

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| Lidar-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| Lidar-002 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| Lidar-003 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| Lidar-004 | 接收雷达设备信息列表查询请求 | 模块接收雷达设备信息列表查询请求 | URL示例：  *config://lidar\_host\_server?command=query* |
| Lidar-005 | 发送雷达设备信息列表查询应答 | 模块发送雷达设备信息列表查询应答。 | URL示例：  *config://”requester”?command=query[&lidar=1\_192.168.0.100\_test[&lidar=\*]]*  lidar字段参数定义示例：*lidar=1\_192.168.0.100\_test*  字段值以“\_”作为分隔符，第一部分表示设备ID，第二部分表示设备IP，第三部分表示设备名称 |
| Lidar-006 | 接收设备新增请求 | 接收应用程序的设备新增请求。 | URL示例：  *config://lidar\_host\_server?command=add&ip=192.168.0.100&port=8000&name=test* |
| Lidar-007 | 发送设备新增应答 | 发送设备新增应答。 | URL示例：  *config://”requester”?command=add&error=0[&lidar=1\_192.168.0.100\_test]* |
| Lidar-008 | 接收设备删除请求 | 接收应用程序的设备删除请求。 | URL示例：  *config://lidar\_host\_server?command=remove&id=1* |
| Lidar-009 | 发送设备删除应答 | 发送设备删除应答。 | URL示例：  *config://”requester”?command=remove&error=0&id=1* |
| Lidar-010 | 点云数据读取 | 模块使用设备厂商SDK接口每100ms主动读取一次点云数据 |  |
| Lidar-011 | 点云数据推送 | 模块每读取一次点云数据就向平台系统的雷达点云数据缓存模块推送一次数据 | URL示例：  *info://lidar\_point\_cloud\_data\_cache\_server?id=1&data=\*\*\** |

## Sensor设备接入模块(sensor\_host\_server)

## 交通协作设备接入(traffic\_cooperate\_server)

## 雷达点云数据缓存模块(lidar\_point\_cloud\_data\_cache\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互；
3. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库创建数据发布服务；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| LidarPtCloud-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| LidarPtCloud-002 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| LidarPtCloud-003 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 | 不处理 |
| LidarPtCloud-004 | 接收雷达点云数据信息 | 模块接收雷达接入设备发送的雷达点云数据推送数据 | URL示例：  *info://lidar\_point\_cloud\_data\_cache\_server?id=1&data=\*\*\** |
| LidarPtCloud-005 | 创建雷达点云数据发布服务 | 应用程序主动调用5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))接口创建雷达点云数据发布服务。 |  |
| LidarPtCloud-006 | 发布雷达点云数据信息 | 将接收到的雷达点云数据向订阅者发布。 | URL示例：  *info://lidar\_point\_cloud\_data\_cache\_server?id=1&data=\*\*\** |
| LidarPtCloud-007 | 雷达点云数据缓存管理 | 单个雷达点云数据单帧数据平均2M字节大小，每1秒平均20M字节大小。缓存大小为10个数据。 |  |

## 交通违法事件发布模块(traffic\_break\_law\_publish\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互；
3. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库创建数据发布服务；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| TrafficBreakLawEventPub-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| TrafficBreakLawEventPub-002 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| TrafficBreakLawPubEventPub-003 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 | 不处理 |
| TrafficBreakLawEventPub-004 | 接收事件数据信息 | 模块接收数据分析模块推送的检测数据 | URL示例：  *info://traffic\_break\_law\_publish\_server?data=\*\*\** |
| TrafficBreakLawEventPub-005 | 创建交通违法事件数据发布服务 | 应用程序主动调用5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))接口创建交通违法事件数据发布服务。 |  |
| TrafficBreakLawEventPub-006 | 发布交通违法事件数据信息 | 将接收到的交通违法数据向订阅者发布。 | URL示例：  *info://traffic\_break\_law\_publish\_server?data=\*\*\** |
| TrafficBreakLawEventPub-007 | 交通违法事件数据缓存管理 | 缓存大小为3个数据。 |  |

## 车辆行驶预警事件发布模块(vehicle\_drive\_warning\_publish\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互；
3. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库创建数据发布服务；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| VehicleDriveWarnEventPub-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| VehicleDriveWarnEventPub-002 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| VehicleDriveWarnEventPub-003 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 | 不处理 |
| VehicleDriveWarnEventPub-004 | 接收事件数据信息 | 模块接收数据分析模块推送的检测数据 | URL示例：  *info://vehicle\_drive\_warning\_publish\_server?data=\*\*\** |
| VehicleDriveWarnEventPub-005 | 创建车辆行驶预警事件数据发布服务 | 应用程序主动调用5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))接口创建车辆行驶预警事件数据发布服务。 |  |
| VehicleDriveWarnEventPub-006 | 发布车辆行驶预警事件数据信息 | 将接收到的车辆行驶预警数据向订阅者发布。 | URL示例：  *info://vehicle\_drive\_warning\_publish\_server?data=\*\*\** |
| VehicleDriveWarnEventPub-007 | 车辆行驶预警事件数据缓存管理 | 缓存大小为3个数据。 |  |

## 车辆行驶服务信息发布模块(vehicle\_drive\_tip\_publish\_server)

### 模块说明

1. 平台系统中单进程运行；
2. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库与平台系统进行数据交互；
3. 模块内加载5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))库创建数据发布服务；

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| VehicleDriveTipEventPub-001 | 连接平台系统 | 使用应用程序指定的平台系统IP地址和端口号连接平台系统。 |  |
| VehicleDriveTipEventPub-002 | 获取平台系统连接状态通知 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 |  |
| VehicleDriveTipEventPub-003 | 获取平台系统模块信息 | 应用程序被动接收5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))通知。 | 不处理 |
| VehicleDriveTipEventPub-004 | 接收事件数据信息 | 模块接收数据分析模块推送的检测数据 | URL示例：  *info://vehicle\_drive\_tip\_publish\_server?data=\*\*\** |
| VehicleDriveTipEventPub-005 | 创建车辆行驶服务信息事件数据发布服务 | 应用程序主动调用5.2 [XMQ客户端模块(libxmqhostclient)](#_XMQ客户端模块(libxmqhostclient))接口创建车辆行驶服务信息事件数据发布服务。 |  |
| VehicleDriveTipEventPub-006 | 发布车辆行驶服务信息事件数据信息 | 将接收到的车辆行驶服务信息数据向订阅者发布。 | URL示例：  *info://vehicle\_drive\_tip\_publish\_server?data=\*\*\** |
| VehicleDriveTipEventPub-007 | 车辆行驶服务信息事件数据缓存管理 | 缓存大小为3个数据。 |  |

## 视频帧解码与转换流程图模块(libavcodec)

### 模块说明

1. 包含于应用程序进程内；
2. 模块建立应用程序流程图创建参数与流程图ID间的映射关系；
3. 模块负责建立推和拉模式的数据传输过程；
4. 模块负责流程图实例的内存数据持久化管理；
5. 模块视频帧数据拆包解析，视频帧数据格式遵照4.5[流数据格式](#_流数据格式)；
6. 模块视频帧PS格式转换PES格式；
7. 模块视频帧PES数据软解码；
8. 模块视频帧YUV420P转换BGR24；
9. 模块视频帧YUV420P编码JPEG。

### 功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| ***编号*** | ***名称*** | ***描述*** | ***备注*** |
| AVC-001 | 创建流程图 | 模块创建应用程序创建参数与流程图ID的映射关系 |  |
| AVC-002 | 销毁流程图 | 模块销毁应用程序创建参数与流程图ID间的映射关系 | 销毁前必须先释放流程图各模块的系统资源。 |
| AVC-003 | 创建数据源模块 | 创建网络数据缓存模式的数据源模块，模块负责将流数据缓存并解析为单帧数据包，如果单帧数据包为音频则过滤，只保留视频帧数据。 |  |
| AVC-004 | 创建视频帧解析模块 | 创建输入帧数据封装格式为PS的帧解析模块，对于帧数据已经是PS格式的则直接跳过。 |  |
| AVC-005 | 创建视频帧解码模块 | 创建视频帧数据解码模块，模块使用FFMPEG库解码，解码应同时支持软、硬解码方式。 |  |
| AVC-006 | 创建视频帧格式转换模块 | 创建视频帧数据格式转换模块，模块使用FFMPEG库支持的格式进行转换。 |  |
| AVC-007 | 创建视频帧播放模块 | 创建视频帧数据播放模块，模块使用Windows D3D库支持的顶点渲染方式进行播放。 |  |
| AVC-008 | 创建图片编码模块 | 创建视频帧数据编码JPEG/PNG图片模块，模块使用FFMPEG库支持的格式进行编码。 |  |

## 数据结构化封装/解析模块(libdatastructure)

## 数据去重分析模块(libdatadeduplication)

## 行人目标危险区域闯入检测模块(libbreakindetect)

## 车辆目标检测模块(libvehicledectect)

## 机动车车牌提取模块(liblicenseplateextract)

## 机动车车牌识别模块(liblicenseplaterecognize)

# 产品授权

# 附表