

广域雷达微波检测器

DTAM D29通讯协议V2.1

南京慧尔视智能科技有限公司

2019年6月

**目 录**

[1. 系统简介 1](#_Toc21954234)

[2. 系统架构 1](#_Toc21954235)

[3. 术语说明 2](#_Toc21954236)

[4. 协议概述 7](#_Toc21954237)

[5. 卡口触发协议 8](#_Toc21954238)

[6. 信号机专用协议 9](#_Toc21954239)

[6.1 规范 9](#_Toc21954240)

[6.2 接口形式 9](#_Toc21954241)

[6.3 数据帧 10](#_Toc21954242)

[6.3.1 帧结构 10](#_Toc21954243)

[6.3.2 校验码、帧开始和帧结束 10](#_Toc21954244)

[6.3.3 数据表 10](#_Toc21954245)

[6.4 通讯规程 12](#_Toc21954246)

[6.4.1 一般要求 12](#_Toc21954247)

[6.4.2 字节顺序 12](#_Toc21954248)

[6.4.3 链路地址 13](#_Toc21954249)

[6.5 通信内容 13](#_Toc21954250)

[6.5.1 校时 13](#_Toc21954251)

[6.5.2 周期设置 14](#_Toc21954252)

[6.5.3 统计数据 14](#_Toc21954253)

[6.5.4 脉冲数据 17](#_Toc21954254)

[6.5.5 故障消息 17](#_Toc21954255)

[6.5.6 实时目标数据 19](#_Toc21954256)

[6.5.7 排队数据 20](#_Toc21954257)

[6.5.8 区域状态 20](#_Toc21954258)

[6.5.9 评价数据 22](#_Toc21954259)

[6.5.10 溢出报警 22](#_Toc21954260)

[6.5.11 动态排队数据 23](#_Toc21954261)

[6.5.12 相位数据 25](#_Toc21954262)

[6.5.13 心跳消息 26](#_Toc21954263)

[7. 交通平台通信协议 27](#_Toc21954264)

[7.1 雷达配置通信协议 27](#_Toc21954265)

[7.1.1 获取安装信息 27](#_Toc21954266)

[7.1.2 获取车道配置 28](#_Toc21954267)

[7.1.3 获取测量线配置 31](#_Toc21954268)

[7.1.4 获取线圈配置 32](#_Toc21954269)

[7.1.5 获取设备时间 34](#_Toc21954270)

[7.1.6 设置设备时间 34](#_Toc21954271)

[7.1.7 获取NTP校时服务配置 35](#_Toc21954272)

[7.1.8 设置NTP校时服务 36](#_Toc21954273)

[7.1.9 获取交通平台统计周期 37](#_Toc21954274)

[7.1.10 设置交通平台统计周期 38](#_Toc21954275)

[7.1.11 获取数据使能状态 39](#_Toc21954276)

[7.1.12 设置数据使能状态 40](#_Toc21954277)

[7.1.13 获取交通平台链路 41](#_Toc21954278)

[7.1.14 设置交通平台链路 41](#_Toc21954279)

[7.1.15 错误回复 42](#_Toc21954280)

[7.2 交叉口应用通信协议 43](#_Toc21954281)

[7.2.1 即时数据 43](#_Toc21954282)

[7.2.2 过车数据 45](#_Toc21954283)

[7.2.3 静态排队数据 47](#_Toc21954284)

[7.2.4 动态排队数据 48](#_Toc21954285)

[7.2.5 区域状态 50](#_Toc21954286)

[7.2.6 统计数据 52](#_Toc21954287)

[7.2.7 评价数据 59](#_Toc21954288)

[7.3 路段应用通信协议 61](#_Toc21954289)

[7.3.1 即时数据 61](#_Toc21954290)

[7.3.2 过车数据 63](#_Toc21954291)

[7.3.3 区域状态 64](#_Toc21954292)

[7.3.4 统计数据 66](#_Toc21954293)

[7.4 故障信息 74](#_Toc21954294)

[8. 事件平台通信协议 75](#_Toc21954295)

[8.1 交叉口交通事件数据 75](#_Toc21954296)

[8.1.1 排队超限事件 75](#_Toc21954297)

[8.1.2 溢出事件 76](#_Toc21954298)

[8.1.3 行人事件 77](#_Toc21954299)

[8.2 路段交通事件数据 78](#_Toc21954300)

[8.2.1 停车事件 79](#_Toc21954301)

[8.2.2 超速事件 80](#_Toc21954302)

[8.2.3 逆行事件 82](#_Toc21954303)

[8.2.4 拥堵事件 83](#_Toc21954304)

[8.2.5 变道事件 84](#_Toc21954305)

[8.2.6 占用应急车道事件 85](#_Toc21954306)

[8.2.7 闯入事件 85](#_Toc21954307)

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 修改日期 | 修改人 | 修改内容 |
| 信号机专用协议V2.0.1 | 2017-10-25 | 陈俊德 | 根据GA/T 920—2010制定信号机专用协议 |
| 信号机专用协议  V2.1.1 | 2018-04-20 | 陈俊德 | 增加网络通讯的周期设置、校时设置  统计数据删除排队长度  评价指数增加流量、最大排队长度。  增加溢出报警 |
| 交通平台通信协议  V2.1.1 | 2018-07-11 | 陈俊德 | “路况信息”更名为“区域状态”  增加测量线和线圈的方向说明 |
| 交通平台通信协议  V2.1.2 | 2018-09-11 | 王栋 | 新增动态排队数据；  新增转向速度 |
| 事件平台通信协议  V2.0.1 | 2018-09-12 | 王栋 | 溢出事件删除末车位置距离雷达位置的字段 |
| 事件平台通信协议  V2.1.4 | 2018-11-30 | 史灵清 | 事件平台通信协议：  部分事件加入目标XY坐标；  拥堵事件加入拥堵消散 |
| V2.1.5 | 2018-12-12 | 史灵清 | 按设备型号分割协议；  统计数据加入交通当量（后删除） |
| V2.1.6 | 2018-12-15 | 曹磊 | 信号机专用协议增加动态排队、增加随机统计/周期统计切换 |
| V2.1.7 | 2019-02-21 | 薛家诚 | 新增相位协议 |
| V2.1.8 | 2019-03-15 | 史灵清 | 修改即时数据坐标系；  更正雷达配置通信协议样例中的端口号 |
| V2.1.9 | 2019-4-17 | 薛家诚 | 交通平台雷达配置通信协议中，  新增NTP校时服务设置和配置获取 |
| V2.1.10 | 2019-5-18 | 史灵清  王岩 | 重新分割协议；  更正数据使能的数据类型；  核对交通平台各样例；  断点续传/错误回复按实际情况修改 |
| V2.1.11 | 2019-6-23 | 席国庆  史灵清 | 修改排队超限协议；  删除断点续传协议 |
|  | 2019-11-8 | 章庆 | 平台对接增加评价数据一次通过率、行程时间  备注：评价数据红色为修改部分 |
| V2.1.12 | 2020-4-8 | 王岩 | 应用通信协议中含速度、位置的字段类型改为浮点数Float；  交叉口交通事件中的排队超限和溢出改为10秒上报一次 |

1. 系统简介

广域雷达微波检测器DTAM D29采用主动扫描式阵列雷达技术，它是微波检测器发展史上的一次跨时代的技术飞跃。与其他检测技术相比，DTAM D29可检测双向不低于8车道，最大检测长度可达200米。不仅能提供流量、速度、占有率等交通统计数据，同时能提供实时车辆脉冲信息，实时区域状态，包括排队长度、区间车辆数等，停车次数、延误时间、燃油消耗等评价指数。在国内各种交通条件、气候环境下已经得到了广泛成功应用，为交通信号控制系统提供全面、准确、实时的交通信息。



图1-1广域雷达微波检测器DTAM D29

1. 系统架构

广域雷达微波检测器数据推送方向主要有信号机和智能交通平台，不同系统对于数据的需求和通讯方式有较大区别，主要系统架构如下图所示。

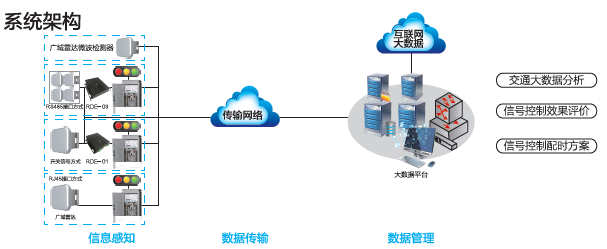
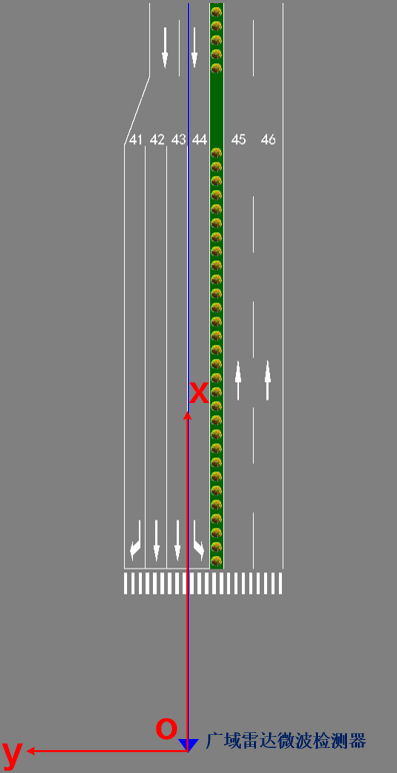


图2-1 系统对接架构图

1. 术语说明

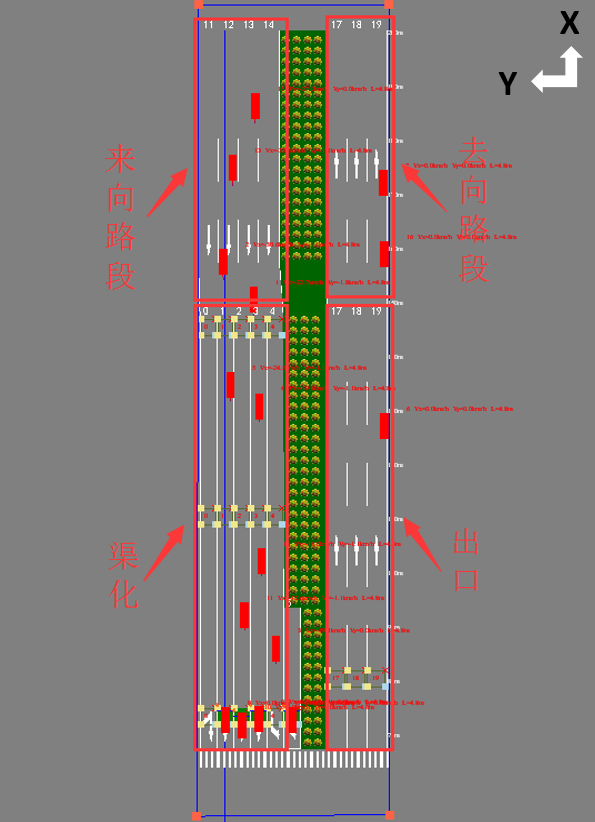
* **雷达坐标系介绍**

即以雷达为原点、雷达法线方向为x轴、垂直雷达法线的左方向为y轴的直角坐标系，如下图所示（在调试软件仿真界面，蓝色三角形的顶点为雷达位置，连接该三角形的蓝色轴线为雷达法线）。



* **车道类型**

车道分为渠化、来向路段、出口、去向路段四种类型，如下图所示。



* **空间参数检测区**

广域雷达能监测200米范围内多达8车道的交通状态，检测区是指画设从停止线开始，到设定的检测区距离的一段范围，用于统计和空间有关的交通信息，一般画设到渠化末端，统计比如区间内车辆数，空间占有率等信息。

* **即时速度**

目标车辆在某一时刻点的速度。

* **即时位置信息**

车辆在以雷达建立的检测坐标系中的（X,Y）坐标。单位为m。

* **车型分类**

按照车辆的行人、非机动车、小车、中车、大车进行分类型输出。

* **车头时距**

在同一车道上同向行驶的车辆队列中，两个连续车辆车头端部进入某一虚拟线圈的时间间隔，单位s。

* **车身间距**

在同一条车道上同向行驶的一列车队中，两个连续车辆前车尾部离开线圈与后车头部进入虚拟线圈的时间差，也称为间隙，单位s。

* **85%位速度**

在该车道行驶的所有车辆中，有85%的车辆行驶速度在此速度之下。

* **静态排队**

车辆在接近停止线时行驶状态发生改变，车道内目标静止且目标之间间隔小于20米的，计入静态排队的排队车辆队列。

* **动态排队**

车辆在接近停止线时行驶状态发生改变，车道内目标速度低于设定速度阈值且目标之间间隔小于设定的间隔阈值时，计入动态排队的排队车辆队列。

* **排队长度**

车道内排队车辆队列的第一辆车车头到最后一辆车车尾的距离（以渠化段车道划分为依据）。

* **排队首车位置**

车道内排队车辆队列的第一辆车车头距停止线的距离。

* **排队末车位置**

车道内排队车辆队列的最后一辆车车尾距停止线的距离。

* **排队数量**

车道内排队车辆队列中的车辆数目。

* **区域车辆数**

在设定的空间参数检测区内，车道内的所有动态静态车辆的数量。

* **分布情况**

在设定的空间参数检测区内，车道内各车辆之间距离（即前车车尾到后车车头的距离）的方差。

* **区域状态的头车位置及速度**

在设定的空间参数检测区内，车道内首车距停止线的距离和当前速度。

* **区域状态的末车位置及速度**

在设定的空间参数检测区内，车道内末车距停止线的距离和当前速度。

* **统计数据的时间占有率**

统计周期内，车流压占线圈的时间与统计周期的比率。

* **区域状态的空间占有率**

在设定的空间参数检测区内，车道内所有车辆所占有的面积与检测区域总面积之比（按长度比计算）。

* **统计数据的平均速度**

统计周期内，通过线圈车辆的速度平均值。

* **区域状态的平均速度**

在设定的空间参数检测区内，车道所有车辆当前速度平均值。

* **平均停车次数**

目标车辆进入雷达检测范围后统计停车次数，统计周期内，计算通过停止线车辆的平均停车次数。

* **平均延误时间**

目标车辆进入雷达检测范围后统计速度低于5km/h的延误时间，统计周期内，计算通过停止线车辆的平均延误时间。

* **燃油消耗**

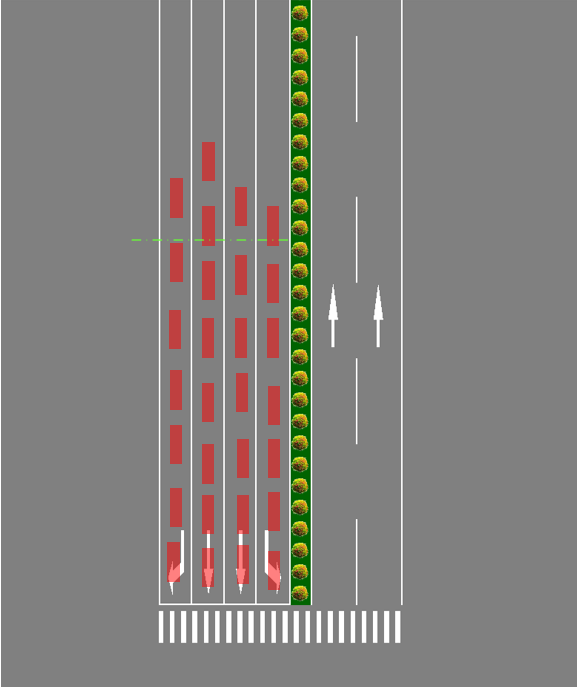
目标车辆进入雷达检测范围后，按不同车型不同形式状态统计燃油消耗，统计周期内，计算通过停止线车辆的总燃油消耗量。

* **尾气排放**

目标车辆进入雷达检测范围后，按不同车型不同形式状态统计尾气排放，统计周期内，计算通过停止线车辆的总尾气排放量。

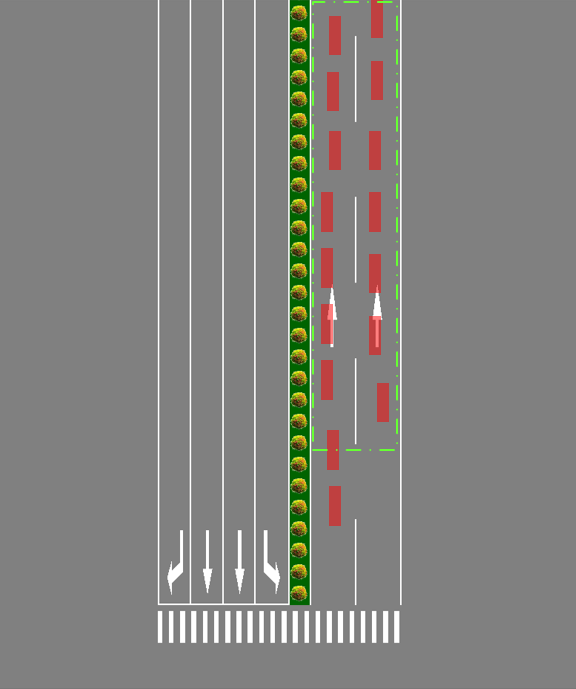
* **排队超限事件**

车辆因受到信号控制或交通拥堵的影响，造成车辆排队直到超过设定位置的交通事件，示意图如下。



* **溢出事件**

因信号控制导致下游车辆排队过长直到堵塞本交叉口的交通事件，示意图如下。



* **行人事件**

在画设的行人检测区域检测到行人目标。

* **异常停车事件**

车辆非拥堵原因在行车道上停车的交通事件。

* **违法停车事件**

车辆在禁止停车的区域停车的交通事件。

* **逆行事件**

车辆在道路上的行驶方向与规定方向相反的交通事件。

* **拥堵事件**

由于车辆多、秩序乱或道路狭窄等造成车辆拥挤、道路堵塞的交通事件。

* **变道事件**

车辆在道路上行驶过程中，在禁止变道的路段，如白实线区域，改变其行驶车道的交通事件。

* **占用应急车道事件**

车辆在应急车道行驶或者停车的交通事件。

* **闯入事件**

车辆驶入禁止驶入的区域的交通事件。

1. 协议概述

广域雷达微波检测器提供卡口触发信息和交通流采集信息，其中，交通流采集信息通讯协议包括即时数据协议、信号机专用协议、交通平台专用协议和事件平台专用协议。

* **卡口触发协议**

广域雷达微波检测器通过RS485通讯方式给相机提供卡口触发信息，包括车道号、速度值。

* **信号机通讯协议**

广域雷达微波检测器通过I/O、RS485或RJ45方式给信号机提供交通流采集信息，包括统计数据、脉冲数据、故障信息，并与信号机实现相位数据的交互。

* **交通平台通讯协议**

广域雷达微波检测器通过RJ45网络传输方式给交通平台提供交通流采集信息，包括即时数据、过车数据、统计数据、故障信息。

* **事件平台通讯协议**

广域雷达微波检测器通过RJ45网络传输方式给事件平台提供交通异常事件信息，包括停车、逆行、超速、变道、排队超限、溢出、拥堵等事件。

* **API**

API使用HTTP协议并遵循REST原则设计的Web服务接口，可以使用几乎任何客户端和任何编程语言与REST API进行交互。通过发送简单的HTTP POST请求就可以轻松接入使用。所有接口请求，响应都使用json格式。

建立HttpServer,并按协议规定url接收雷达数据，公布IP及端口号，雷达设置此IP和端口号，会自动连接该HttpServer，并保持长连接，按协议规定的url推送数据。

表1 HTTP标准包头表

|  |  |  |
| --- | --- | --- |
| **属性** | **类型** | **描述** |
| Accept | String | 客户端响应接收数据格式： application/json |
| Content-Type | String | 请求内容类型：application/json;charset=utf-8 |
| **Connection** | **String** | **keep-alive，长连接** |

1. 卡口触发协议

**接口方式：**RS485

**串口设置：**波特率：9600bit/s、停止位1bit、数据位8bit、校验位无

表2 卡口触发信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 协议头 | 1 | 0xFF | 固定格式为0xFF |
| 2 | 车道号 | 1 | 1～15 | 相机根据此值来对应车道关联 |
| 3 | 速度 | 2 | 0～6553 | 单位：km/h |
| 4 | 校验和 | 1 | —— | 等于协议头异或车道异或速度 |

**样例：**

FF 01 00 4D B3（1车道，速度为77km/h）

FF 02 00 B1 4C（2车道，速度为177km/h）

1. **信号机专用协议**
   1. **规范**

广域雷达微波检测器推送数据给信号机时，协议的制定主要依据的标准规范为《GA/T 920—2010道路交通信号控制机与车辆检测器间的通信协议》。

* 1. **接口形式**

广域雷达微波检测器对接信号机的接口形式主要分I/O、RS485、和RJ45。

* **I/O**

信号机的接口形式是I/O时，只能推送脉冲信息给信号机，其他信息需要其他接口推送，比如排队长度等。脉冲信号的推送，主要采用RDE-01或RDE-04雷达数据处理器实现。车辆进入检测虚拟线圈时给出信号，离开检测虚拟线圈时清除信号，常开常闭可根据实际情况选择。

* **RS485**

信号机的接口形式是RS485时，可推送所有信息，包括脉冲信息、统计数据等，当信号机端有多个RS485时，可直接按方向对接，当信号机仅具备一个RS485接口时，可选择RDE-03将各方向的数据进行汇总，推送给信号机，数据内具有方向标识。波特率根据需求设定。

* **RJ45**

信号机的接口形式为RJ45时，可推送所有信息，包括脉冲信息、统计数据等，信号机为Server端，检测器为Client端，IP和端口可指定。

* 1. **数据帧**
     1. **帧结构**

数据帧由帧开始、数据表、校验码和帧结束构成，其封装格式见图1

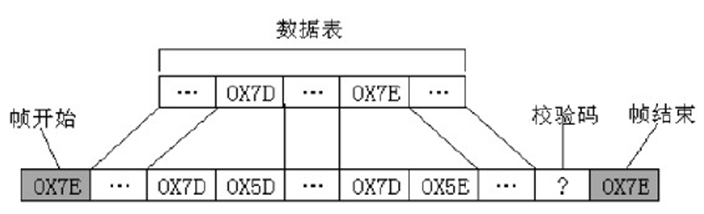


图1数据帧封装格式

* + 1. **校验码、帧开始和帧结束**

校验码和帧开始、帧结束应符合下列要求：

1. 帧开始和帧结束应为一字节十六进制数 **0x7E**；
2. 帧结束之前，应有校验码，其值为数据表所有字节**按位异或**；
3. 校验码值为 **0x7E** 时，应以 **0x7D**和 **0x5E** 两个字节取代；
4. 校验码值为 **0x7D**时，应以 **0x7D** 和 **0x5D**两个字节取代。
   * 1. **数据表**

数据表由链路地址、协议版本号、操作类型、对象标识及消息内容五部分构成，见图2



图2数据表结构

1. 链路地址：链路地址最多由 2 个字节组成。如果链路地址的第一个地址字节的第一个比特为 1，链路地址为单字节，见图3a）；如果链路地址的第一个地址字节的第一个比特为 0，那么随后的字节应是链路地址的一个扩展,见图3b)。链路地址应以 1 开头的一个字节来结束。链路地址的第一个字节的第二位为保留位（G），单字节地址的有效地址域是高六位，取值范围是 0～63，双字节地址的有效地址域是第一字节的高六位连接上第二字节的高七位，取值范围是 64～8191。**约定采用单字节链路地址**。**地址0为广播地址，无论检测器的实际链路地址是多少，在接收到广播地址命令时，都要响应。**



图3链路地址

1. 协议版本号：标识通信协议的版本，用一字节十六进制数 **0x10** 表示；
2. 操作类型：标识数据表的操作类型，用一个字节表示，见表3

表3操作类型

|  |  |  |
| --- | --- | --- |
| **操作码** | **含义** | **说明** |
| 0x80 | 查询请求 | 信号机发送查询消息 |
| 0x81 | 设置请求 | 信号机发送设置消息 |
| 0x82 | 主动上传 | 检测器主动上传数据 |
| 0x83 | 查询应答 | 检测器对查询请求的应答 |
| 0x84 | 设置应答 | 检测器对设置请求的应答 |
| 0x85 | 主动上传应答 | 信号机对主动上传的应答 |
| 0x86 | 出错应答 | 接收到的数据包存在错误 |
| 其他 | 保留 |  |

1. 对象标识：标识数据表的操作对象，用一个字节表示，见表4

表4对象标识

|  |  |  |
| --- | --- | --- |
| **取值** | **含义** | **说明** |
| 0x02 | 时间设置 | 信号机对雷达进行校时 |
| 0x04 | 周期设置 | 统计周期的设置 |
| 0x05 | 统计数据 | 线圈的流量速度占有率等统计数据 |
| 0x08 | 脉冲数据 | 实时脉冲数据 |
| 0x09 | 故障消息 | 检测器故障 |
| 0x50 | 实时目标数据 | 检测区域内所有目标的状态信息 |
| 0x51 | 排队数据 | 通道的实时排队情况 |
| 0x52 | 区域状态 | 通道的实时车辆分布情况 |
| 0x53 | 评价数据 | 通道的评价数据 |
| 0x54 | 溢出报警 | 去向的溢出报警 |
| 0x56 | 动态排队数据 | 通道的实时动态排队情况 |
| 0x57 | 相位数据 | 当前交叉口的信号机相位 |
| 0x58 | 心跳消息 | 一定时间间隔向信号机发送心跳包告知雷达在线状态 |

1. 消息内容：对对象值进行具体描述，部分数据表不包含消息内容字段，具体参看下面章节；
2. 数据表中出现数据 0x7E 时，应以 0x7D 和 0x5E 两个字节取代；
3. 数据表中出现数据 0x7D 时，应以 0x7D 和 0x5D 两个字节取代；

**注意：校验码是通过原始数据计算而成，即发送时先校验后替换，接收时先替换后校验。**

* 1. 通讯规程
     1. 一般要求

采用主从方式进行通信，信号机为主站，检车器为从站。

**RJ45网络通讯支持以下协议所有命令，RS485通讯方式，虽然支持，但不建议信号机给雷达发送命令，包括查询和应答，因雷达通讯数据量较大，避免冲突，尽量采用雷达主动上传的方式。**

* + 1. **字节顺序**

长度大于一个字节的数据对象，应先发送低字节，后发送高字节。

* + 1. **链路地址**

链路地址约定单字节，高6位表达雷达编号（检测方向编号），见图3。

各向编号对应来链路地址**：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **北** | **东** | **南** | **西** | **东北** | **东南** | **西南** | **西北** |
| 0x07 | 0x0F | 0x17 | 0x1F | 0x0B | 0x13 | 0x1B | 0x23 |

* 1. **通信内容**
     1. **校时**
        1. **时间设置**

具备外部统一NTP校时服务器时，设备设置与统一的NTP校时服务器校时，不具备外部统一NTP校时服务器时，信号机与检测器建立通讯连接后应设置检测器时间，并按一定时间频率对检测器进行校时。**不建议两种校时方式并存**。信号机设置检测器时间时发送时间设置表格，**该命令只支持RJ45网络通讯**，见表5。

表5 时间设置表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x81 | 0x02 | 表6 |

表6 本地时间表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 本地时间 | 4 | 0～0XFFFFFFFF | 自1970年1月1日零点以来的秒数。 |

* + - 1. **时间设置应答**

检测器收到时间设置命令后调整本机时间并发送应答表格，见表7。

表7 时间设置应答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x84 | 0x02 | 无 |

* + 1. **周期设置**
       1. **周期设置**

信号机设置检测器的统计周期时发送周期参数设置表，**该命令只支持RJ45网络通讯**，见表8。

表8 周期参数设置表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x81 | 0x04 | 表9 |

表9 周期参数表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 统计周期 | 2 | 1～3600 | 统计数据的计算周期，单位:s，不建议大于300s，避免流量超过255造成溢出。 |
| 2 | 保留数据 | 7 | — | 保留字段 |

* + - 1. **周期设置应答**

检测器收到周期设置命令后**回复当前已统计的数据**，调整采集周期并发送应答表格，见表10。

表10 周期设置应答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x84 | 0x04 | 无 |

* + 1. **统计数据**
       1. **随机统计数据开始设置**

信号机可根据需求使用不定周期的统计数据模式，发送开始随机统计数据设置命令，**该命令只支持RJ45网络通讯**，见表11。

表11 随机统计数据设置表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x81 | 0x05 | 无 |

* + - 1. **随机统计数据设置应答**

检测器收到随机统计数据设置命令后发送应答表格，见表12。

表12 随机统计数据设置应答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x84 | 0x05 | 无 |

* + - 1. **统计数据查询**

信号机处于随机统计数据模式下可根据需求发送统计数据查询命令，建议为5min之内，**该命令只支持RJ45网络通讯**，见表13。

表13 统计数据查询表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x80 | 0x05 | 无 |

* + - 1. **统计数据查询应答**

检测器收到统计数据查询命令后发送应答表格，见表14。

表14 统计数据查询应答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x83 | 0x05 | 表15 |

表15 统计数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 统计数据生成的本地时间 | 4 | 0～0XFFFFFFFF | 表16 |
| 2 | 配置参数 | 9 | — | 表17 |
| 3 | 检测通道数 | 1 | N（1≤N≤48） | 统计数据对应的检测通道数 |
| 4 | 单路检测通道统计数据 | 13 | — | 表18 |
| … | … | … | … | … |
| 3+N | 单路检测通道统计数据 | 13 | — | 表18 |

表16 本地时间数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 本地时间 | 4 | 0～0XFFFFFFFF | 自1970年1月1日零点以来的秒数。 |

表17 统计数据配置参数表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 统计周期 | 2 | 1～3600 | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | A类车车长 | 1 | 130 | A类>13m  6m< B类 <=13m  C类<=6m |
| 3 | B类车车长 | 1 | 60 |
| 4 | C类车车长 | 1 | 0 |
| 5 | 保留数据 | 4 | 0 |  |

表18 统计数据单数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道编号 | 1 | 1～255 | 检测通道的编号（线圈编号）。 |
| 2 | A类车流量 | 1 | 1～255 |  |
| 3 | B类车流量 | 1 | 1～255 |  |
| 4 | C类车流量 | 1 | 1～255 |  |
| 5 | 占有率 | 1 | 0～200 | 车辆平均占有率。单位:0.5% |
| 6 | 平均车速 | 1 | 1～255 | 车辆平均行驶速度，255表示溢出。单位:km/h |
| 7 | 平均车长 | 1 | 1～255 | 车辆平均车长，255表示溢出。单位:0.1m |
| 8 | 平均车头时距 | 1 | 1～255 | 车辆平均车头时距，255表示溢出。单位:s |
| 9 | 保留 | 5 | —— | 保留字节。 |

* + - 1. **统计数据主动上传**

联机状态下，一个统计周期结束后检测器主动上传最新统计数据， 见表19。

表19 统计数据主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x05 | 表15 |

* + - 1. **统计数据主动上传应答**

信号机收到检测器主动上传的统计数据后发送应答表格，见表20。

表20 统计数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x05 | 无 |

* + 1. **脉冲数据**
       1. **脉冲数据主动上传**

广域雷达根据信号机检测要求，可在不同车道，不同距离，设置不同长度的虚拟线圈，可编辑通道号，检测器在检测到有车辆进入或离开虚拟线圈时应主动向信号机发送脉冲数据，见表21。

表21 脉冲数据主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x08 | 表22 |

表22 脉冲数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行，状态发生改变的通道数量。 |
| 2 | 第一行 | 2 | — | 表23 |
| … | … | … | … | … |
| N+1 | 第N行 | 2 | — | 表23 |

表23 脉冲数据单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（线圈编号） |
| 2 | 通道状态 | 1 | 0～1 | 0:车辆离开检测区； 1:车辆进入检测区 |

* + - 1. **脉冲数据主动上传应答**

信号机收到检测器主动上传的脉冲数据后应发送应答表格，见表24。

表24 脉冲数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x08 | 无 |

* + 1. **故障消息**
       1. **故障信息查询**

信号机在与检测器联机成功后发送故障信息查询命令，**该命令只支持RJ45网络通讯**，见表25。

表25 故障信息查询表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x80 | 0x09 | 无 |

* + - 1. **故障信息查询应答**

检测器收到配置参数查询命令后发送应答表格，见表26。

表26 故障信息查询应答表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x83 | 0x09 | 表27 |

表27 故障信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1~N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 3 | — | 表28 |
| … | … | … | … | … |
| N+1 | 第N行 | 3 | — | 表28 |

表28 故障信息单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（线圈编号） |
| 2 | 通道状态 | 1 | 0～1 | 0:通道正常，1:通道故障 |
| 3 | 故障类型 | 1 | 0～255 | 0:未知故障，1:前端故障，2:欠压，3:高温 |

* + - 1. **故障信息主动上传**

检测器在检测到发生故障或故障恢复时应主动向信号机发送故障消息，故障状态持续时，每隔一秒上报一次故障，直至清除，见表29。

表29 故障信息主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x09 | 表27 |

* + - 1. **故障信息主动上传应答**

信号机收到检测器主动上传的故障信息后应发送应答表格，见表30。

表30 故障信息主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x09 | 无 |

* + 1. 实时目标数据

检测器主动向信号机上传实时目标数据，1秒推送一次，分别给出所有目标的所在车道号，所在车道的车道类型，距停止线的位置，速度，车长。

信号机需要根据雷达车道参数自行对目标做对应。

* + - 1. 实时目标数据主动上传

表 1 实时目标数据上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x50 | 表32 |

表 2 实时目标数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 目标数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 6 | — | 表33 |
| … | … | … | … | … |
| N+1 | 第N行 | 6 | — | 表33 |

表 3 实时目标数据单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 通道号 | 1 | 0～255 | 目标车辆所在通道号（车道编号） |
| 2 | 车道类型 | 1 | 0～3 | 目标车辆所在车道类型（0：渠化，1：来向路段，2：出口道，3：去向路段） |
| 3 | 目标位置 | 1 | 0～255 | 目标离停止线的位置，255表示溢出，单位：m |
| 4 | 目标速度 | 2 | 0～600 | 目标当前车速。速度为负：目标朝向雷达行驶，速度为正：目标背离雷达行驶），偏移量300（如：-60km/h表示为240，80km/h表示为380），超出范围表示无效，单位：km/h， |
| 5 | 目标车长 | 1 | 0～255 | 目标车身长度，255表示溢出，单位：0.1m |

* + - 1. 实时目标数据主动上传应答

信号机收到检测器主动上传的实时目标数据信息后应发送应答表格，见表34。

表 4 实时目标数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x50 | 无 |

* + 1. **排队数据**

数据采用主动上传，1秒推送一次，按车道给出排队长度,队首，队尾及排队数量。

* + - 1. **排队数据主动上传**

表31 排队数据上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x51 | 表32 |

表32 排队数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 5 | — | 表33 |
| … | … | … | … | … |
| N+1 | 第N行 | 5 | — | 表33 |

表33 排队数据单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（车道编号） |
| 2 | 通道排队长度 | 1 | 0～255 | 单位：m，队首到队尾的距离 |
| 3 | 队首 | 1 | 0～255 | 单位：m，距离停止线 |
| 4 | 队尾 | 1 | 0～255 | 单位：m，距离停止线 |
| 5 | 通道排队数量 | 1 | 0～255 | 单位：辆 |

* + - 1. **排队数据主动上传应答**

表34 排队数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x51 | 无 |

* + 1. **区域状态**

数据采用主动上传，1秒推送一次，按车道给出区间内车辆数、空间占有率、空间密度、平均速度、末车位置。

* + - 1. **区域状态主动上传**

表35 区域状态上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x52 | 表36 |

表36 区域状态表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 9 | — | 表37 |
| … | … | … | … | … |
| N+1 | 第N行 | 9 | — | 表37 |

表37 区域状态单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（车道编号） |
| 2 | 通道车辆数 | 1 | 0～255 | 单位：辆 |
| 3 | 空间占有率 | 1 | 0～100 | 单位：% |
| 4 | 平均速度 | 1 | 0～255 | 区间内车辆的实时平均速度，单位：km/h |
| 5 | 分布情况 | 1 | 0～255 | 单位：m，车辆间距方差 |
| 6 | 头车位置 | 1 | 0～255 | 单位：m，通道内第一辆车距停止线距离 |
| 7 | 头车速度 | 1 | 0～255 | 单位：km/h |
| 8 | 末车位置 | 1 | 0～255 | 单位：m，通道内最后一辆车距停止线距离 |
| 9 | 末车速度 | 1 | 0～255 | 单位：km/h |

* + - 1. **区域状态主动上传应答**

表38 区域状态主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x52 | 无 |

* + 1. **评价数据**
       1. **评价数据主动上传**

表39 评价数据主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x53 | 表40 |

表40 评价数据数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 19 | — | 表41 |
| … | … | … | … | … |
| N+1 | 第N行 | 19 | — | 表41 |

表41评价数据数据单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（车道编号） |
| 2 | 流量 | 1 | 0～255 | 统计周期内车流量，单位:辆 |
| 3 | 最大排队长度 | 1 | 0～255 | 统计周期内最大排队长度，单位:m |
| 4 | 平均停车次数 | 4 | Float | 单位：次 |
| 5 | 平均延误时间 | 4 | Float | 单位：s |
| 6 | 燃油消耗 | 4 | Float | 单位：ml |
| 7 | 尾气排放 | 4 | Float | 单位：mg |

* + - 1. **评价数据主动上传应答**

表42评价数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x53 | 无 |

* + 1. 溢出报警
       1. **溢出报警信息主动上传**

检测器在检测到道路溢出时应主动向信号机发送溢出报警信息，如果状态解除则检测器发送溢出报警解除信息，见表43。

表43 溢出报警信息主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x54 | 表44 |

表44 溢出报警信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 报警开始 | 1 | 0 | 溢出报警开始 |

表45 溢出报警信息解除主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x54 | 表46 |

表46 溢出报警解除信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 报警解除 | 1 | FF | 溢出报警解除 |

* + - 1. **溢出报警信息主动上传应答**

信号机收到检测器主动上传的溢出报警信息后应发送应答表格，见表47。

表47 溢出报警信息主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x54 | 无 |

* + 1. 动态排队数据

数据采用主动上传，1秒推送一次，按车道给出排队长度,队首，队尾及排队数量。

* + - 1. 动态排队参数设置

信号机设置检测器的动态排队参数时发送动态排队参数设置表，**该命令只支持RJ45网络通讯**，见表48。

表48 动态排队参数设置表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x81 | 0x56 | 表49 |

表49 动态排队参数表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 间隔 | 1 | 0～255 | 当车道内车辆间隔低于此阈值时，计入动态排队长度 |
| 2 | 速度 | 1 | 0～255 | 当车道内车辆速度低于此阈值时，计入动态排队长度 |

* + - 1. 动态排队参数设置应答

检测器收到动态排队参数设置命令后发送应答表格，见表50。

表50 动态排队参数设置应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x84 | 0x56 | 无 |

* + - 1. 动态排队数据主动上传

表51 排队数据上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x56 | 表52 |

表52 排队数据表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道数 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 2 | 第一行 | 5 | — | 表53 |
| … | … | … | … | … |
| N+1 | 第N行 | 5 | — | 表53 |

表53 排队数据单表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测通道号 | 1 | 0～255 | 检测通道号（车道编号） |
| 2 | 通道排队长度 | 1 | 0～255 | 单位：m，队首到队尾的距离 |
| 3 | 队首 | 1 | 0～255 | 单位：m，距离停止线 |
| 4 | 队尾 | 1 | 0～255 | 单位：m，距离停止线 |
| 5 | 通道排队数量 | 1 | 0～255 | 单位：辆 |

* + - 1. 动态排队数据主动上传应答

表54 排队数据主动上传应答

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x85 | 0x56 | 无 |

* + 1. 相位数据
       1. 信号机主动推送

信号机进行相位切换时，主动将路口所有方向的相位信息发送给雷达，见表55。

信号机无法提供的信号，可以不发送。但是，**左转、直行、右转信号不可空缺，如果空缺或取值无效，则检测器默认绿灯（放行）。**

表55 信号机相位推送表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x81 | 0x57 | 表56 |

表56 相位信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 相位方案 | 1 | char | 默认填0；  如果信号机有相位方案或类似概念，将当前相位方案的名称或编号填入这个字段。 |
| 2 | 进口数量 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 3 | 第一行 | N | — | 表57 |
| … | … | … | … | … |
| N+2 | 第N行 | N | — | 表57 |

表57 单个进口信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 检测方向 | 1 | char | 检测器方向，按“链路地址”定义取值 |
| 2 | 流向数量 | 1 | 1～N | 实际表行数，这个字段表明后面数据总共有多少行 |
| 3 | 第一行 | 2 | — | 表58 |
| … | … | … | … | … |
| N+2 | 第N行 | 2 | — | 表58 |

表58 单个流向信息表结构

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **名称** | **字节数** | **取值** | **描述** |
| 1 | 流向信号 | 1 | 1～8 | 表59 |
| 2 | 灯色状态 | 1 | 0～5 | 0：无效，1：绿灯，2：绿闪，3：黄灯，4：黄闪，5：红灯 |

表59 流向信号定义

|  |  |  |
| --- | --- | --- |
| **取值** | **含义** | **说明** |
| 0x01 | 公交 | 公交通行信号 |
| 0x02 | 右转 | 右转信号或包含右转的信号 |
| 0x03 | 直行 | 直行信号或包含直行的信号 |
| 0x04 | 左转 | 左转信号或包含左转的信号 |
| 0x05 | 掉头 | 掉头信号或包含掉头的信号 |
| 0x06 | 行人1 | 行人一次过街信号或普通行人信号，见图4 |
| 0x07 | 行人2 | 行人二次过街信号，见图4 |
| 0x08 | 非机动车 | 非机动车信号 |

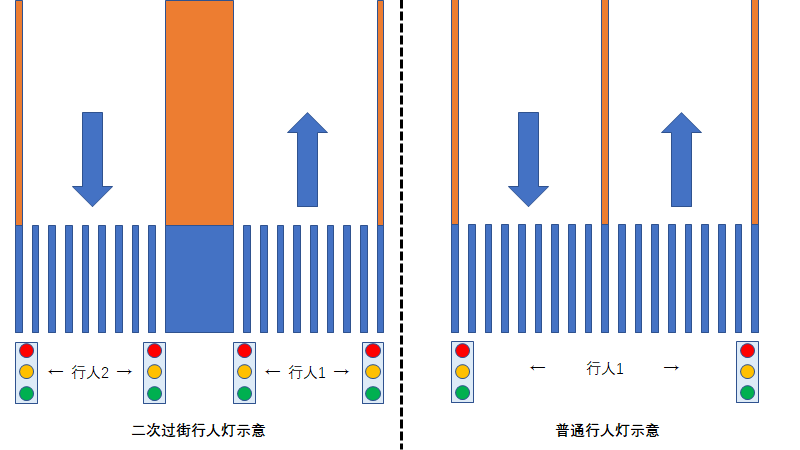


图6-4 行人信号说明示意图

* + 1. **心跳消息**
       1. **心跳消息主动上传**

广域雷达每5秒主动上传一次心跳消息包，用于向信号机告知设备在线状态。

表60 心跳消息主动上传表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **名称** | **链路地址** | **协议版本** | **操作类型** | **对象标识** | **消息内容** |
| 内容 | 链路地址 | 0x21 | 0x82 | 0x58 | 无 |

1. 交通平台通信协议

广域雷达微波检测器通过RJ45网络传输方式给交通平台提供交通流采集信息，协议主体包括配置信息、交通流采集信息（分交叉口信息采集应用和路段信息采集应用）。

* 1. 雷达配置通信协议

**接口说明：**广域雷达默认打开4004端口接收http请求，请求后，雷达回复后会自动断开连接，用于平台获取雷达基本配置信息和设置有关参数。

* + 1. 获取安装信息

**请求地址：**/radarConfigure/getInstallationCfg

**请求样例:**

GET /radarConfigure/getInstallationCfg HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表61 安装信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| IPAddress | String | Y | IP地址 |
| Address | String | Y | 安装地址 |
| Direction | String | Y | 检测方向  North，East，South，West， NorthEast，NorthWest，SouthEast，SouthWest |
| Height | Float | Y | 安装高度 |
| Version | String | Y | 固件版本号 |

**响应样例:**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 134

{

"DeviceNo":"204",

"IPAddress":"192.168.1.204",

"Address":"清水亭西路",

"Direction":"SouthEast",

"Height":6.500000,

"Version":"4.0.0"

}

* + 1. 获取车道配置

**请求地址：**/radarConfigure/getLaneCfg

**请求样例:**

GET /radarConfigure/getLaneCfg HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表62 车道配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| LaneNum | Int | Y | 车道数量 |
| LaneCfg\_List | Array | Y | 车道配置列表 |
| LaneNo | Int | Y | 车道编号 |
| LimitSpeed | Int | Y | 限速值 |
| XPos | Float | Y | 车道起点中心X坐标（雷达坐标系） |
| YPos | Float | Y | 车道起点中心Y坐标（雷达坐标系） |
| Width | Float | Y | 车道宽度 |
| Length | Float | Y | 车道长度 |
| Direction | String | Y | 行驶方向 |
| Type | Int | Y | 车道类型  0：渠化，1：来向路段，2：出口，3去向路段 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 1174

{

"DeviceNo":"204",

"LaneNum":3,

"LaneCfg\_List":

[{

"LaneNo":11,

"LimitSpeed":60,

"XPos":69,

"YPos":-1.100000,

"Width":3,

"Length":70,

"Direction":"右转＋直行",

"Type":0

},{

"LaneNo":12,

"LimitSpeed":60,

"XPos":69,

"YPos":-4.100000,

"Width":3,

"Length":70,

"Direction":"直行",

"Type":0

},{

"LaneNo":13,

"LimitSpeed":60,

"XPos":69,

"YPos":-7.100000,

"Width":3,

"Length":70,

"Direction":"直行",

"Type":0

},{

"LaneNo":1,

"LimitSpeed":60,

"XPos":147,

"YPos":-0.800000,

"Width":3,

"Length":53,

"Direction":"直行",

"Type":1

},{

"LaneNo":2,

"LimitSpeed":60,

"XPos":147,

"YPos":-3.800000,

"Width":3,

"Length":53,

"Direction":"直行",

"Type":1

},{

"LaneNo":3,

"LimitSpeed":60,

"XPos":147,

"YPos":-6.800000,

"Width":3,

"Length":53,

"Direction":"直行",

"Type":1

},{

"LaneNo":7,

"LimitSpeed":100,

"XPos":69,

"YPos":-11.200000,

"Width":3.600000,

"Length":70,

"Direction":"去向",

"Type":2

},{

"LaneNo":8,

"LimitSpeed":100,

"XPos":69,

"YPos":-14.800000,

"Width":3.600000,

"Length":70,

"Direction":"去向",

"Type":2

},{

"LaneNo":4,

"LimitSpeed":60,

"XPos":147,

"YPos":-11.200000,

"Width":3,

"Length":53,

"Direction":"去向",

"Type":3

},{

"LaneNo":5,

"LimitSpeed":60,

"XPos":147,

"YPos":-14.800000,

"Width":3,

"Length":53,

"Direction":"去向",

"Type":3

}]

}

* + 1. 获取测量线配置

**请求地址：/radarConfigure/getMeaslineCfg**

**请求样例：**

GET /radarConfigure/getMeaslineCfg HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表63 测量线配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| MeaslineNum | Int | Y | 测量线数量 |
| MeasCfg\_List | Array | Y | 测量线配置列表 |
| MeaslineNo | Int | Y | 测量线标号 |
| XPos | Float | Y | 测量线X坐标（雷达坐标系） |
| Direction | Int | Y | 0:来向 /1:去向 |

**响应样例：**

HTTP/1.1 200 OK

Connection: close

Server: Mongoose/6.11

Content-Type: application/json;charset=utf-8

Content-Length: 136

{

"DeviceNo":"204",

"MeaslineNum":2,

"MeasCfg\_List":

[{

"MeaslineNo":1,

"XPos":80,

"Direction":0

},{

"MeaslineNo":2,

"XPos":130,

"Direction":0

}]

}

* + 1. 获取线圈配置

**请求地址：/radarConfigure/getCoilCfg**

**请求样例：**

GET /radarConfigure/getCoilCfg HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表64 线圈配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| CoilNum | Int | Y | 线圈数量 |
| CoilCfg\_List | Array | Y | 线圈配置列表 |
| CoilNo | Int | Y | 线圈编号 |
| XPos | Float | Y | 线圈中心点X坐标（雷达坐标系） |
| YPos | Float | Y | 线圈中心点Y坐标（雷达坐标系） |
| Width | Float | Y | 线圈宽度 |
| Length | Float | Y | 线圈长度 |
| Measline | Int | Y | 所属测量线编号 |
| LaneNo | Int | Y | 所属车道编号 |
| Direction | Int | Y | 0:来向 /1:去向 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 308

{

"DeviceNo":"204",

"CoilNum":3,

"CoilCfg\_List":

[{

"CoilNo":11,

"XPos":80,

"YPos":-1.100000,

"Width":3,

"Length":3,

"Measline":1,

"LaneNo":11,

"Direction":0

},{

"CoilNo":12,

"XPos":80,

"YPos":-4.100000,

"Width":3,

"Length":3,

"Measline":1,

"LaneNo":12",

“Direction":0

},{

"CoilNo":13,

"XPos":80,

"YPos":-7.100000,

"Width":3,

"Length":3,

"Measline":1,

"LaneNo":13,

"Direction":0

}]

}

* + 1. 获取设备时间

**请求地址：/radarConfigure/getDeviceTime**

**请求样例：**

GET /radarConfigure/getDeviceTime HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表65 获取设备时间

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| DeviceTime | String | Y | 设备时间 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 50

{

"DeviceNo":"204",

"DeviceTime":”2018-07-30 12:00:00”

}

* + 1. 设置设备时间

**请求地址：/radarConfigure/setDeviceTime**

**请求数据：**

表66 设置设备时间

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceTime | string | Y | 设置的时间 |

**请求样例：**

POST /radarConfigure/setDeviceTime HTTP/1.1

Host: 127.0.0.1:4004

Accept: application/json

Content-Type: application/json

Content-Length: 34

{

"DeviceTime":"2018-04-28 20:00:00"

}

**响应消息：**

表67 设置设备时间的响应

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Status | string | Y | 设置状态：success设置成功，fail设置失败 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json; charset=utf-8

Content-Length: 20

{"status":"success"}

* + 1. 获取NTP校时服务配置

**请求地址：/radarConfigure/getNTP**

**请求样例：**

GET /radarConfigure/getNTP HTTP/1.1

Host: 192.168.1.212:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表68 获取NTP校时配置

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Enable | Bool | Y | NTP使能 |
| IPAddress | String | Y | NTP服务器IP地址 |
| Port | Int | Y | NTP服务端口 |
| Interval | Int | Y | 校准时间间隔 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json; charset=utf-8

Content-Length: 103

{

"DeviceNo": "212",

"Enable": true,

"IPAddress": "192.168.1.101",

"Port": 123,

"Interval": 3600

}

* + 1. 设置NTP校时服务

**请求地址：/radarConfigure/setNTP**

**请求数据：**

表69 设置设备时间

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Enable | Bool | Y | NTP使能 |
| IPAddress | String | Y | NTP服务器IP地址 |
| Port | Int | Y | NTP服务端口 |
| Interval | Int | Y | 校准时间间隔 |

**请求样例：**

POST /radarConfigure/setNTP HTTP/1.1

Host: 192.168.1.212:4004

Accept: application/json

Content-Type: application/json

Content-Length: 83

{

"Enable": true,

"IPAddress": "192.168.1.137",

"Port": 123,

"Interval": 3600

}

**响应消息：**

表70 设置NTP校时的响应

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Status | string | Y | 设置状态：success设置成功，fail设置失败 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11  
Connection: close  
Content-Type: application/json; charset=utf-8  
Content-Length: 20  
  
{"status":"success"}

* + 1. 获取交通平台统计周期

**请求地址：/radarConfigure/getCycle**

**请求样例：**

GET /radarConfigure/getCycle HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表71 统计周期配置信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Cycle | Int | Y | 统计周期时长，单位s，范围1-3600 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 29

{

"DeviceNo":"204",

"Cycle":10

}

* + 1. 设置交通平台统计周期

**请求地址：/radarConfigure/setCycle**

**请求数据：**

表72 设置统计周期

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Cycle | Int | Y | 周期时间，单位s，范围：1—3600 |

**请求样例：**

POST /radarConfigure/setCycle HTTP/1.1

Host: 127.0.0.1:4004

Accept: application/json

Content-Type: application/json

Content-Length: 21

{

"Cycle":60

}

**响应消息：**

表73 设置统计周期的响应

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Status | string | Y | 设置状态：success设置成功，fail设置失败 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11  
Connection: close  
Content-Type: application/json; charset=utf-8  
Content-Length: 20  
  
{"status":"success"}

* + 1. 获取数据使能状态

**请求地址：/radarConfigure/getEnable**

**请求样例：**

GET /radarConfigure/getEnable HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表74 数据使能状态信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| ObjDataEnable | Int | Y | 0表示不推送, 1-255表示推送 |
| QueueEnable | Bool | Y | 排队数据使能状态（true,false） |
| RoadEnable | Bool | Y | 区域状态使能状态（true,false） |
| CycleEnable | Bool | Y | 统计数据使能状态（true,false） |
| EvaluationEnable | Bool | Y | 评价数据使能状态（true,false） |
| PassEnable | Bool | Y | 过车数据使能状态（true,false） |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 163

{

"DeviceNo":"204",

"ObjDataEnable":10,

"QueueEnable":true,

"RoadEnable":false,

"CycleEnable":true,

"EvaluationEnable":true,

"PassEnable":false

}

* + 1. 设置数据使能状态

**请求地址：/radarConfigure/setEnable**

**请求消息：**

表75 数据使能状态信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| ObjDataEnable | Int | Y | 0表示不推送, 1-255表示推送 |
| QueueEnable | Bool | Y | 排队数据使能状态（true,false） |
| RoadEnable | Bool | Y | 区域状态使能状态（true,false） |
| CycleEnable | Bool | Y | 统计数据使能状态（true,false） |
| EvaluationEnable | Bool | Y | 评价数据使能状态（true,false） |
| PassEnable | Bool | Y | 过车数据使能状态（true,false） |

**请求样例：**

POST /radarConfigure/setEnable HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

Content-Length: 142

{

"ObjDataEnable":10,

"QueueEnable":true,

"RoadEnable":false,

"CycleEnable":true,

"EvaluationEnable":true,

"PassEnable":false

}

**响应消息：**

表76 设置数据使能状态的响应

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Status | string | Y | 设置状态：success设置成功，fail设置失败 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11  
Connection: close  
Content-Type: application/json; charset=utf-8  
Content-Length: 20  
  
{"status":"success"}

* + 1. 获取交通平台链路

**请求地址：/radarConfigure/getTrafficIPPort**

**请求样例：**

GET /radarConfigure/getTrafficIPPort HTTP/1.1

Host: 192.168.1.201:4004

Accept: application/json

Content-Type: application/json

**响应消息：**

表77 交通平台链路信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| IPAddress | String | Y | 交通平台IP |
| Port | Int | Y | 交通平台端口 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11

Connection: close

Content-Type: application/json;charset=utf-8

Content-Length: 57

{

"DeviceNo":"202",

"IPAddress":"192.168.1.83",

"Port":4005

}

* + 1. 设置交通平台链路

**请求地址：/radarConfigure/setTrafficIPPort**

**请求消息：**

表78 交通平台链路信息

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| IPAddress | String | Y | 交通平台IP |
| Port | Int | Y | 交通平台端口 |

**请求样例：**

POST /radarConfigure/setTrafficIPPort HTTP/1.1

Host: 192.1.1.201:4004

Accept: application/json

Content-Type: application/json

Content-Length: 52

{

"IPAddress":"192.168.1.88",

"Port":4005

}

**响应消息：**

表79 设置交通平台链路的响应

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| Status | string | Y | 设置状态：success设置成功，fail设置失败 |

**响应样例：**

HTTP/1.1 200 OK

Server: Mongoose/6.11  
Connection: close  
Content-Type: application/json; charset=utf-8  
Content-Length: 20  
  
{"status":"success"}

* + 1. 错误回复

请求标识错误回复：

HTTP/1.1 404 Not Found

Server: Mongoose/6.11

Connection: close

Content-Type: text/plain; charset=utf-8

Content-Length: 12

URI illegal.

请求IP地址或端口错误回复：

HTTP/1.1 403 Forbidden

Server: Mongoose/6.11

Connection: close

Content-Type: text/plain; charset=utf-8

Content-Length: 52

Host illegal. Make sure HOST ip and port is correct.

请求内容错误回复：

HTTP/1.1 403 Forbidden

Server: Mongoose/6.11

Connection: close

Content-Type: text/plain; charset=utf-8

Content-Length: 43

Body illegal. Make sure content is correct.

请求历史数据不在约定时间内回复：

HTTP/1.1 403 Forbidden

Server: Mongoose/6.11

Connection: close

Content-Type: text/plain; charset=utf-8

Content-Length: 13

Time illegal.

* 1. 交叉口应用通信协议

雷达推送交叉口交通流采集信息，包括即时数据、过车数据、排队数据、区域状态、统计数据和评价数据。

* + 1. 即时数据

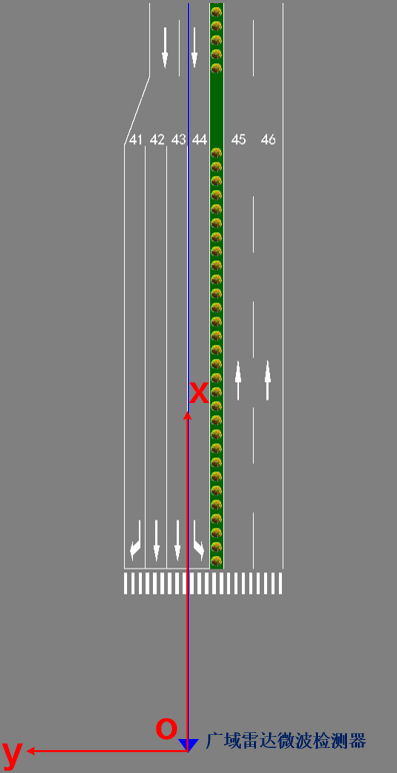
**接口说明**：推送目标即时数据

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/objData

**推送频率**：频率可设置，参考“数据使能状态”。

以雷达为原点、雷达法线方向为x轴、垂直雷达法线的左方向为y轴建立雷达坐标系，推送该坐标系下的实时目标编号、长度、位置（XPos，YPos）、速度（XSpeed，YSpeed），见下图。



**请求JSON数据说明**：

表82 即时数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| Obj\_List | Array | Y | 目标信息列表 |
| ID | Int | Y | 目标车辆编号 |
| Length | Float | Y | 目标车辆长度(m) |
| XPos | Float | Y | 目标车辆X坐标(m) |
| YPos | Float | Y | 目标车辆Y坐标(m) |
| XSpeed | Float | Y | 目标车辆X方向速度(km/h) |
| YSpeed | Float | Y | 目标车辆Y方向速度(km/h) |

**样例：**

POST /radarDataCollect/objData HTTP/1.1

HOST: [192.168.1.85:789](qq://txfile/)

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 482

{

"DeviceNo": "51020414",

"Timestamp": "2017-11-17 13:57:20.265",

"Obj\_List": [{

"ID": 1,

"Length": 4.800000,

"XPos": 20.000011,

"YPos": 3.700004,

"XSpeed": -14.750000,

"YSpeed": -0.250000

}, {

"ID": 2,

"Length": 4.800000,

"XPos": -5.200001,

"YPos": 15.299994,

"XSpeed": -8.500000,

"YSpeed": 0.250000

}]

}

* + 1. 过车数据

**接口说明**：车辆压占线圈并离开线圈后，实时推送过车数据。

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/passData

**推送频率**：实时

**请求JSON数据说明**：

表83 过车数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| MeasNo | Int | Y | 测量线编号 |
| LaneNo | Int | Y | 车道编号 |
| CoilNo | Int | Y | 线圈编号 |
| Speed | Float | Y | 目标离开线圈时的速度(km/h) |
| Vehicle\_Len | Float | Y | 车长(m) |
| Vehicle\_Type | Int | Y | 1行人，2非机动车，3小型车，4中型车，5大型车 |
| DriveIntoTime | String | Y | 目标进入线圈的时间 |
| PresenceTime | Int | Y | 压占时间/存在时间（ms） |

**样例：**

POST /radarDataCollect/passData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 180

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:25:09.746",

"MeasNo": 1,

"CoilNo": 15,

"LaneNo": 43,

"Speed": 49.500000,

"Vehicle\_Len": 5.900000,

"Vehicle\_Type": 3,

"DriveIntoTime”: "2017-11-10 12:25:09.171",

"PresenceTime": 575

}

* + 1. 静态排队数据

**接口说明**：雷达服务每秒推送各设备实时静态排队数据，包含每个车道内排队车辆数、排队队列第一辆车距停止线距离、排队队列最后一辆车距离停止线位置。

静态排队数据和区域状态主要区别是：区域状态不区分车道车流是静止还是运动状态，而排队数据车道内车流是静态的。

静态排队数据目标是静止的并且目标之间的间隔小于20米。

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/queueData

**推送频率**：每秒推送

**请求JSON数据说明**：

表84 静态排队数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| LaneNum | Int | Y | 车道总数 |
| Queue\_List | Array | Y | 各车道的排队信息列表 |
| LaneNo | Int | Y | 车道编号 |
| Queue\_Len | Float | Y | 排队长度（m） |
| Queue\_Head | FLoat | Y | 排队第一辆车距离停止线距离（m） |
| Queue\_Tail | Float | Y | 排队最后一辆车距离停止线距离（m） |
| Queue\_Num | Int | Y | 排队车辆数（辆） |

**样例：**

POST /radarDataCollect/queueData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 493

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:21:41.412",

"LaneNum": 4,

"Queue\_List": [{

"LaneNo": 1,

"Queue\_Len": 12.363007,

"Queue\_Head": 2.480278,

"Queue\_Tail": 14.843285,

"Queue\_Num": 2

}, {

"LaneNo": 2,

"Queue\_Len": 12,

"Queue\_Head": 2.475670,

"Queue\_Tail": 14.475670,

"Queue\_Num": 1

}, {

"LaneNo": 3,

"Queue\_Len": 12,

"Queue\_Head": 2.493210,

"Queue\_Tail": 14.493210,

"Queue\_Num": 1

}, {

"LaneNo": 4,

"Queue\_Len": 4.800003,

"Queue\_Head": 2.440689,

"Queue\_Tail": 7.240692,

"Queue\_Num": 1

}]

}

* + 1. 动态排队数据

**接口说明**：雷达服务每秒推送各设备实时动态排队数据，包含每个车道内排队车辆数、排队队列第一辆车距停止线距离、排队队列最后一辆车距离停止线位置。

动态排队数据和静态排队数据主要区别是：静态排队数据目标是静止不动的，动态排队数据目标的速度以及间隔可设。

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/queueDataDynamic

**推送频率**：每秒推送

**请求JSON数据说明**：

表85 动态排队数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| LaneNum | Int | Y | 车道总数 |
| Queue\_List | Array | Y | 各车道的排队信息列表 |
| LaneNo | Int | Y | 车道编号 |
| Queue\_Len | Float | Y | 排队长度（m） |
| Queue\_Head | Float | Y | 排队第一辆车距离停止线距离（m） |
| Queue\_Tail | Float | Y | 排队最后一辆车距离停止线距离（m） |
| Queue\_Num | Int | Y | 排队车辆数（辆） |

**样例：**

POST /radarDataCollect/queueDataDynamic HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 493

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:21:41.412",

"LaneNum": 4,

"Queue\_List": [{

"LaneNo": 1,

"Queue\_Len": 12.363007,

"Queue\_Head": 2.480278,

"Queue\_Tail": 14.843285,

"Queue\_Num": 2

}, {

"LaneNo": 2,

"Queue\_Len": 12,

"Queue\_Head": 2.475670,

"Queue\_Tail": 14.475670,

"Queue\_Num": 1

}, {

"LaneNo": 3,

"Queue\_Len": 12,

"Queue\_Head": 2.493210,

"Queue\_Tail": 14.493210,

"Queue\_Num": 1

}, {

"LaneNo": 4,

"Queue\_Len": 4.800003,

"Queue\_Head": 2.440689,

"Queue\_Tail": 7.240692,

"Queue\_Num": 1

}]

}

* + 1. 区域状态

**接口说明**：雷达服务每秒推送各设备实时区域状态，包含每个车道内车辆数、空间占有率、空间平均速度、分布系数、首/末车相对停止线的位置及速度。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/roadData

**推送频率**：每秒推送

**请求JSON数据说明**：

表86 区域状态主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNum | Int | Y | 车道总数 |
| RoadStatus\_List | Array | Y | 各车道区域状态列表 |
| LaneNo | Int | Y | 车道编号 |
| Vehicle\_Num | Int | Y | 车道内车辆数（辆） |
| Occupancy | Float | Y | 车道内空间占有率（%） |
| AVSpeed | Float | Y | 平均速度（km/h） |
| Pareto | Float | Y | 车道内车辆分布情况（车间距方差）（m） |
| Head\_Pos | Float | Y | 车道内第一辆车位置（m） |
| Head\_Speed | Float | Y | 头车速度（km/h） |
| Last\_Pos | Float | Y | 车道内最后一辆车位置（m） |
| Last\_Speed | Float | Y | 末车速度（km/h） |

**样例：**

POST /radarDataCollect/roadData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 801

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:21:39.404",

"LaneNum": 4,

"RoadStatus\_List": [{

"LaneNo": 1,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}, {

"LaneNo": 2,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}, {

"LaneNo": 3,

"Vehicle\_Num": 5,

"Occupancy": 73.846153,

"AVSpeed": 6.480000,

"Pareto": 0.405516,

"Head\_Pos": 1.960159,

"Head\_Speed": 0,

"Last\_Pos": 29.445366,

"Last\_Speed": 32.400002

}, {

"LaneNo": 4,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}]

}

* + 1. 统计数据

**接口说明**：雷达服务按周期推送交通统计数据，统计数据包括周期内车流量总数、平均速度、时间占有率、车头时距等。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/cycleData

**推送频率**：按周期推送（一般设置3分钟或5分钟）

**请求JSON数据说明**：

表87 统计数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| Cycle | Int | Y | 统计周期（s） |
| CoilNum | Int | Y | 线圈数量 |
| Coil\_List | Array | Y | 各虚拟线圈统计信息列表 |
| MeasNo | Int | Y | 检测线编号 |
| LaneNo | Int | Y | 车道编号 |
| CoilNo | Int | Y | 线圈编号 |
| Volume | Int | Y | 不区分车型机动车总流量 |
| Volume1 | Int | Y | 行人流量 |
| Volume2 | Int | Y | 非机动车流量 |
| Volume3 | Int | Y | 小车流量 |
| Volume4 | Int | Y | 中车流量 |
| Volume5 | Int | Y | 大车流量 |
| AVSpeed | Float | Y | 平均速度（km/h） |
| Occupancy | Float | Y | 时间占有率（%） |
| Headway | Float | Y | 平均车头时距（s） |
| Gap | Float | Y | 平均车身间距（s） |
| Speed\_85 | Float | Y | 85位速度（km/h） |

**样例：**

POST /radarDataCollect/cycleData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 5273

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:50:00",

"Cycle": 300,

"CoilNum": 18,

"Coil\_List": [{

"MeasNo": 1,

"LaneNo": 41,

"CoilNo": 1,

"Volume": 17,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 17,

"Volume5": 0,

"AVSpeed": 21.140471,

"Occupancy": 35.867729,

"Headway": 13.837500,

"Gap": 12.986052,

"Speed\_85": 25

}, {

"MeasNo": 1,

"LaneNo": 42,

"CoilNo": 2,

"Volume": 15,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 15,

"Volume5": 0,

"AVSpeed": 29.940001,

"Occupancy": 29.809317,

"Headway": 15.330000,

"Gap": 14.728798,

"Speed\_85": 31

}, {

"MeasNo": 1,

"LaneNo": 43,

"CoilNo": 3,

"Volume": 16,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 16,

"Volume5": 0,

"AVSpeed": 28.012503,

"Occupancy": 31.949806,

"Headway": 14.324000,

"Gap": 13.681430,

"Speed\_85": 28

}, {

"MeasNo": 1,

"LaneNo": 44,

"CoilNo": 4,

"Volume": 2,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 2,

"Volume5": 0,

"AVSpeed": 13.500000,

"Occupancy": 55.019306,

"Headway": 100.620003,

"Gap": 99.286669,

"Speed\_85": 13

}, {

"MeasNo": 2,

"LaneNo": 41,

"CoilNo": 5,

"Volume": 15,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 15,

"Volume5": 0,

"AVSpeed": 30.396002,

"Occupancy": 9.408926,

"Headway": 13.752857,

"Gap": 13.160674,

"Speed\_85": 43

}, {

"MeasNo": 2,

"LaneNo": 42,

"CoilNo": 6,

"Volume": 13,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 13,

"Volume5": 0，

"AVSpeed": 31.638460,

"Occupancy": 4.293295,

"Headway": 17.670000,

"Gap": 17.101072,

"Speed\_85": 45

}, {

"MeasNo": 2,

"LaneNo": 43,

"CoilNo": 7,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"AVSpeed": 26.246572,

"Occupancy": 12.228654,

"Headway": 16.481539,

"Gap": 15.795735,

"Speed\_85": 27

}, {

"MeasNo": 2,

"LaneNo": 44,

"CoilNo": 8,

"Volume": 3,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 3,

"Volume5": 0,

"AVSpeed": 29.699999,

"Occupancy": 0.868097,

"Headway": 63,

"Gap": 62.393939,

"Speed\_85": 26

}, {

"MeasNo": 3,

"LaneNo": 41,

"CoilNo": 9,

"Volume": 22,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 22,

"Volume5": 0,

"AVSpeed": 37.595459,

"Occupancy": 5.449723,

"Headway": 8.882857,

"Gap": 8.404076,

"Speed\_85": 45

}, {

"MeasNo": 3,

"LaneNo": 42,

"CoilNo": 10,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"AVSpeed": 44.164284,

"Occupancy": 2.941176,

"Headway": 12.600000,

"Gap": 12.192431,

"Speed\_85": 48

}, {

"MeasNo": 3,

"LaneNo": 43,

"CoilNo": 11,

"Volume": 13,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 13,

"Volume5": 0,

"AVSpeed": 42.438461,

"Occupancy": 2.892961,

"Headway": 17.475000,

"Gap": 17.050857,

"Speed\_85": 43

}, {

"MeasNo": 3,

"LaneNo": 44,

"CoilNo": 12,

"Volume": 3,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 3,

"Volume5": 0,

"AVSpeed": 46.799999,

"Occupancy": 0.626657,

"Headway": 62.669998,

"Gap": 62.285383,

"Speed\_85": 39

}, {

"MeasNo": 4,

"LaneNo": 41,

"CoilNo": 13,

"Volume": 5,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 5,

"Volume5": 0,

"AVSpeed": 28.439999,

"Occupancy": 1.422029,

"Headway": 10.395000,

"Gap": 9.762089,

"Speed\_85": 34

}, {

"MeasNo": 4,

"LaneNo": 42,

"CoilNo": 14,

"Volume": 23,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 23,

"Volume5": 0,

"AVSpeed": 47.200695,

"Occupancy": 3.975904,

"Headway": 8.419091,

"Gap": 8.037741,

"Speed\_85": 58

}, {

"MeasNo": 4,

"LaneNo": 43,

"CoilNo": 15,

"Volume": 24,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 24,

"Volume5": 0,

"AVSpeed": 53.362499,

"Occupancy": 4.119489,

"Headway": 9.159130,

"Gap": 8.821815,

"Speed\_85": 60

}, {

"MeasNo": 5,

"LaneNo": 45,

"CoilNo": 16,

"Volume": 0,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 0,

"Volume5": 0,

"AVSpeed": 0,

"Occupancy": 0,

"Headway": 0,

"Gap": 0,

"Speed\_85": 0

}, {

"MeasNo": 5,

"LaneNo": 46,

"CoilNo": 17,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"AVSpeed": 52.971428,

"Occupancy": 2.552987,

"Headway": 17.773846,

"Gap": 17.434040,

"Speed\_85": 62

}, {

"MeasNo": 5,

"LaneNo": 47,

"CoilNo": 18,

"Volume": 30,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 30,

"Volume5": 0,

"AVSpeed": 50.310001,

"Occupancy": 5.491330,

"Headway": 7.930345,

"Gap": 7.572563,

"Speed\_85": 59

}]

}

* + 1. 评价数据

**接口说明**：雷达服务按统计周期推送各设备评价数据，包含每个车道平均停车次数、平均延误时间、燃油消耗和尾气排放。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/evaluation

**推送频率**：按统计周期（一般设置3分钟或5分钟）

**请求JSON数据说明**：

表88 评价数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| CycleTime | Int | Y | 评价数据周期 |
| SteeringInfo | Array | Y | 路口转向比信息 |
| TotalVolume | Int | Y | 路口总计流量(辆) |
| TotalSpeed | Float | Y | 转向目标总平均速度（KM/H） |
| LeftVolume | Int | Y | 左转流量（辆） |
| LeftSpeed | Float | Y | 左转目标平均速度（KM/H） |
| StraightVolume | Int | Y | 直行流量（辆） |
| StraightSpeed | Float | Y | 直行目标平均速度（KM/H） |
| RightVolume | Int | Y | 右转流量（辆） |
| RightSpeed | Float | Y | 右转目标平均速度（KM/H） |
| LaneNum | String | Y | 车道总数 |
| Evaluation\_List | Array | Y | 各车道评价数据信息列表 |
| LaneNo | Int | Y | 车道编号 |
| Volume | Int | Y | 车道内过停止线流量（辆） |
| MaxQueueLen | Float | Y | 车道内最大排队长度（m） |
| SampleNum | Int | Y | 评价数据计算样本量 |
| Stops | Float | Y | 车道内平均停车次数（次） |
| Delay | Float | Y | 车道内平均延误时间（s） |
| NoStopRate | Float | Y | 车道内一次通过率 |
| TravelDist | Float | Y | 车道内检测行程距离（m） |
| TravelTime | Float | Y | 车道内平均行程时间（s） |
| Fuel | Float | Y | 车道内燃油消耗(ml) |
| Emissions | Float | Y | 车道内尾气排放(mg) |

**样例：**

POST /radarDataCollect/evaluation HTTP/1.1

HOST: 192.168.1.102:4005

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 917

{

"DeviceNo":"203" ,

"Timestamp":"2018-09-17 15:07:21" ,

"CycleTime":300 ,

"SteeringInfo":

{

"TotalVolume":11 ,

"TotalSpeed":35 ,

"LeftVolume":2 ,

"LeftSpeed":12 ,

"StraightVolume":9 ,

"StraightSpeed":40 ,

"RightVolume":0 ,

"RightSpeed":0

} ,

"LaneNum":4 ,

"Evaluation\_List":

[{

"LaneNo":7 ,

"Volume":3 ,

"MaxQueueLen":0 ,

"SampleNum":3 ,

"Stops":0 ,

"Delay":0 ,

"NoStopRate":100 ,

"TravelDist":200 ,

"TravelTime":5 ,

"Fuel":21 ,

"Emissions":270.899994

} ,{

"LaneNo":6 ,

"Volume":3 ,

"MaxQueueLen":0 ,

"SampleNum":3 ,

"Stops":0 ,

"Delay":1 ,

"NoStopRate":66.666667 ,

"TravelDist":200 ,

"TravelTime":10 ,

"Fuel":21.240000 ,

"Emissions":273.996002

} ,{

"LaneNo":5 ,

"Volume":5 ,

"MaxQueueLen":6 ,

"SampleNum":5 ,

"Stops":0.200000 ,

"Delay":14 ,

"NoStopRate":0 ,

"TravelDist":200 ,

"TravelTime":15 ,

"Fuel":24.360001 ,

"Emissions":314.244019

} ,{

"LaneNo":4 ,

"Volume":0 ,

"MaxQueueLen":6 ,

"SampleNum":0 ,

"Stops":0 ,

"Delay":0 ,

"NoStopRate":0 ,

"TravelDist":200 ,

"TravelTime":0 ,

"Fuel":0 ,

"Emissions":0

}]

}

* 1. 路段应用通信协议

雷达推送路段交通流采集信息，包括即时数据、过车数据、区域状态和统计数据。

* + 1. 即时数据

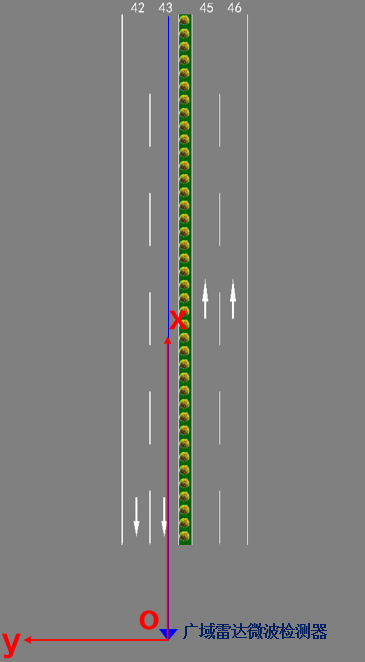
**接口说明**：推送目标即时数据

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/objData

**推送频率**：频率可设置，参考“数据使能状态”。

以雷达为原点、雷达法线方向为x轴、垂直雷达法线的左方向为y轴建立雷达坐标系，推送该坐标系下的实时目标编号、长度、位置（XPos，YPos）、速度（XSpeed，YSpeed），见下图。



**请求JSON数据说明**：

表89 即时数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| Obj\_List | Array | Y | 目标信息列表 |
| ID | Int | Y | 目标车辆编号 |
| Length | Float | Y | 目标车辆长度(m) |
| XPos | Float | Y | 目标车辆X坐标(m) |
| YPos | Float | Y | 目标车辆Y坐标(m) |
| XSpeed | Float | Y | 目标车辆X方向速度(km/h) |
| YSpeed | Float | Y | 目标车辆Y方向速度(km/h) |

**样例：**

POST /radarDataCollect/objData HTTP/1.1

HOST: [192.168.1.85:789](qq://txfile/)

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 482

{

"DeviceNo": "51020414",

"Timestamp": "2017-11-17 13:57:20.265",

"Obj\_List": [{

"ID": 1,

"Length": 4.800000,

"XPos": 20.000011,

"YPos": 3.700004,

"XSpeed": -14.750000,

"YSpeed": -0.250000

}, {

"ID": 2,

"Length": 4.800000,

"XPos": -5.200001,

"YPos": 15.299994,

"XSpeed": -8.500000,

"YSpeed": 0.250000

}]

}

* + 1. 过车数据

**接口说明**：车辆压占线圈并离开线圈后，实时推送过车数据。

**调用方式**：微波服务推送数据

**推送地址**： /radarDataCollect/passData

**推送频率**：实时

**请求JSON数据说明**：

表90 过车数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| MeasNo | Int | Y | 测量线编号 |
| LaneNo | Int | Y | 车道编号 |
| CoilNo | Int | Y | 线圈编号 |
| Speed | Float | Y | 目标离开线圈时的速度(km/h) |
| Vehicle\_Len | Float | Y | 车长(m) |
| Vehicle\_Type | Int | Y | 1行人，2非机动车，3小型车，4中型车，5大型车 |
| DriveIntoTime | String | Y | 目标进入线圈的时间 |
| PresenceTime | Int | Y | 压占时间/存在时间（ms） |

**样例：**

POST /radarDataCollect/passData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 180

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:25:09.746",

"MeasNo": 1,

"CoilNo": 15,

"LaneNo": 43,

"Speed": 49.500000,

"Vehicle\_Len": 5.900000,

"Vehicle\_Type": 3,

"DriveIntoTime”: "2017-11-10 12:25:09.171",

"PresenceTime": 575

}

* + 1. 区域状态

**接口说明**：雷达服务每秒推送各设备实时区域状态，包含每个车道内车辆数、空间占有率、空间平均速度、分布系数、首/末车相对车道起点位置及速度。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/roadData

**推送频率**：每秒推送

**请求JSON数据说明**：

表91 区域状态主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNum | Int | Y | 车道总数 |
| RoadStatus\_List | Array | Y | 各车道区域状态列表 |
| LaneNo | Int | Y | 车道编号 |
| Vehicle\_Num | Int | Y | 车道内车辆数（辆） |
| Occupancy | Float | Y | 车道内空间占有率（%） |
| AVSpeed | Float | Y | 平均速度（km/h） |
| Pareto | Float | Y | 车道内车辆分布情况（车间距方差）（m） |
| Head\_Pos | Float | Y | 车道内第一辆车位置（m） |
| Head\_Speed | Float | Y | 头车速度（km/h） |
| Last\_Pos | Float | Y | 车道内最后一辆车位置（m） |
| Last\_Speed | Float | Y | 末车速度（km/h） |

**样例：**

POST /radarDataCollect/roadData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 801

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:21:39.404",

"LaneNum": 4,

"RoadStatus\_List": [{

"LaneNo": 1,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}, {

"LaneNo": 2,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}, {

"LaneNo": 3,

"Vehicle\_Num": 5,

"Occupancy": 73.846153,

"AVSpeed": 6.480000,

"Pareto": 0.405516,

"Head\_Pos": 1.960159,

"Head\_Speed": 0,

"Last\_Pos": 29.445366,

"Last\_Speed": 32.400002

}, {

"LaneNo": 4,

"Vehicle\_Num": 0,

"Occupancy": 0,

"AVSpeed": 0,

"Pareto": 0,

"Head\_Pos": 0,

"Head\_Speed": 0,

"Last\_Pos": 0,

"Last\_Speed": 0

}]

}

* + 1. 统计数据

**接口说明**：雷达服务按周期推送交通统计数据，统计数据包括周期内车流量总数、平均速度、时间占有率、车头时距等。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/cycleData

**推送频率**：按周期推送（一般设置3分钟或5分钟）

**请求JSON数据说明**：

表92 统计数据主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| Cycle | Int | Y | 统计周期（s） |
| CoilNum | Int | Y | 线圈数量 |
| Coil\_List | Array | Y | 各虚拟线圈统计信息列表 |
| MeasNo | Int | Y | 检测线编号 |
| LaneNo | Int | Y | 车道编号 |
| CoilNo | Int | Y | 线圈编号 |
| Volume | Int | Y | 不区分车型机动车总流量 |
| Volume1 | Int | Y | 行人流量 |
| Volume2 | Int | Y | 非机动车流量 |
| Volume3 | Int | Y | 小车流量 |
| Volume4 | Int | Y | 中车流量 |
| Volume5 | Int | Y | 大车流量 |
| PCU | Float | Y | 交通当量（pcu） |
| AVSpeed | Float | Y | 平均速度（km/h） |
| Occupancy | Float | Y | 时间占有率（%） |
| Headway | Float | Y | 平均车头时距（s） |
| Gap | Float | Y | 平均车身间距（s） |
| Speed\_85 | Float | Y | 85位速度（km/h） |

**样例：**

POST /radarDataCollect/cycleData HTTP/1.1

HOST: 192.168.1.85:987

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 5550

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 12:50:00",

"Cycle": 300,

"CoilNum": 18,

"Coil\_List": [{

"MeasNo": 1,

"LaneNo": 41,

"CoilNo": 1,

"Volume": 17,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 17,

"Volume5": 0,

"PCU": 17,

"AVSpeed": 21.140471,

"Occupancy": 35.867729,

"Headway": 13.837500,

"Gap": 12.986052,

"Speed\_85": 25

}, {

"MeasNo": 1,

"LaneNo": 42,

"CoilNo": 2,

"Volume": 15,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 15,

"Volume5": 0,

"PCU": 15,

"AVSpeed": 29.940001,

"Occupancy": 29.809317,

"Headway": 15.330000,

"Gap": 14.728798,

"Speed\_85": 31

}, {

"MeasNo": 1,

"LaneNo": 43,

"CoilNo": 3,

"Volume": 16,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 16,

"Volume5": 0,

"PCU": 16,

"AVSpeed": 28.012503,

"Occupancy": 31.949806,

"Headway": 14.324000,

"Gap": 13.681430,

"Speed\_85": 28

}, {

"MeasNo": 1,

"LaneNo": 44,

"CoilNo": 4,

"Volume": 2,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 2,

"Volume5": 0,

"PCU": 2,

"AVSpeed": 13.500000,

"Occupancy": 55.019306,

"Headway": 100.620003,

"Gap": 99.286669,

"Speed\_85": 13

}, {

"MeasNo": 2,

"LaneNo": 41,

"CoilNo": 5,

"Volume": 15,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 15,

"Volume5": 0,

"PCU": 15,

"AVSpeed": 30.396002,

"Occupancy": 9.408926,

"Headway": 13.752857,

"Gap": 13.160674,

"Speed\_85": 43

}, {

"MeasNo": 2,

"LaneNo": 42,

"CoilNo": 6,

"Volume": 13,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 13,

"Volume5": 0，

"PCU": 13,

"AVSpeed": 31.638460,

"Occupancy": 4.293295,

"Headway": 17.670000,

"Gap": 17.101072,

"Speed\_85": 45

}, {

"MeasNo": 2,

"LaneNo": 43,

"CoilNo": 7,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"PCU": 14,

"AVSpeed": 26.246572,

"Occupancy": 12.228654,

"Headway": 16.481539,

"Gap": 15.795735,

"Speed\_85": 27

}, {

"MeasNo": 2,

"LaneNo": 44,

"CoilNo": 8,

"Volume": 3,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 3,

"Volume5": 0,

"PCU": 3,

"AVSpeed": 29.699999,

"Occupancy": 0.868097,

"Headway": 63,

"Gap": 62.393939,

"Speed\_85": 26

}, {

"MeasNo": 3,

"LaneNo": 41,

"CoilNo": 9,

"Volume": 22,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 22,

"Volume5": 0,

"PCU": 22,

"AVSpeed": 37.595459,

"Occupancy": 5.449723,

"Headway": 8.882857,

"Gap": 8.404076,

"Speed\_85": 45

}, {

"MeasNo": 3,

"LaneNo": 42,

"CoilNo": 10,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"PCU": 14,

"AVSpeed": 44.164284,

"Occupancy": 2.941176,

"Headway": 12.600000,

"Gap": 12.192431,

"Speed\_85": 48

}, {

"MeasNo": 3,

"LaneNo": 43,

"CoilNo": 11,

"Volume": 13,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 13,

"Volume5": 0,

"PCU": 13,

"AVSpeed": 42.438461,

"Occupancy": 2.892961,

"Headway": 17.475000,

"Gap": 17.050857,

"Speed\_85": 43

}, {

"MeasNo": 3,

"LaneNo": 44,

"CoilNo": 12,

"Volume": 3,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 3,

"Volume5": 0,

"PCU": 3,

"AVSpeed": 46.799999,

"Occupancy": 0.626657,

"Headway": 62.669998,

"Gap": 62.285383,

"Speed\_85": 39

}, {

"MeasNo": 4,

"LaneNo": 41,

"CoilNo": 13,

"Volume": 5,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 5,

"Volume5": 0,

"PCU": 5,

"AVSpeed": 28.439999,

"Occupancy": 1.422029,

"Headway": 10.395000,

"Gap": 9.762089,

"Speed\_85": 34

}, {

"MeasNo": 4,

"LaneNo": 42,

"CoilNo": 14,

"Volume": 23,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 23,

"Volume5": 0,

"PCU": 23,

"AVSpeed": 47.200695,

"Occupancy": 3.975904,

"Headway": 8.419091,

"Gap": 8.037741,

"Speed\_85": 58

}, {

"MeasNo": 4,

"LaneNo": 43,

"CoilNo": 15,

"Volume": 24,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 24,

"Volume5": 0,

"PCU": 24,

"AVSpeed": 53.362499,

"Occupancy": 4.119489,

"Headway": 9.159130,

"Gap": 8.821815,

"Speed\_85": 60

}, {

"MeasNo": 5,

"LaneNo": 45,

"CoilNo": 16,

"Volume": 0,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 0,

"Volume5": 0,

"PCU": 0,

"AVSpeed": 0,

"Occupancy": 0,

"Headway": 0,

"Gap": 0,

"Speed\_85": 0

}, {

"MeasNo": 5,

"LaneNo": 46,

"CoilNo": 17,

"Volume": 14,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 14,

"Volume5": 0,

"PCU": 14,

"AVSpeed": 52.971428,

"Occupancy": 2.552987,

"Headway": 17.773846,

"Gap": 17.434040,

"Speed\_85": 62

}, {

"MeasNo": 5,

"LaneNo": 47,

"CoilNo": 18,

"Volume": 30,

"Volume1": 0,

"Volume2": 0,

"Volume3": 0,

"Volume4": 30,

"Volume5": 0,

"PCU": 30,

"AVSpeed": 50.310001,

"Occupancy": 5.491330,

"Headway": 7.930345,

"Gap": 7.572563,

"Speed\_85": 59

}]

}

* 1. 故障信息

**接口说明**：雷达前端出现故障时推送各设备故障信息，包含欠压故障、高温故障、干扰故障、临时性故障、持续性故障。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/fault

**推送频率**：故障时推送

**请求JSON数据说明**：

表93 故障信息主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| FaultType | Int | Y | Persistent\_Error:持续性故障  Interference：干扰性故障  Temperature\_Error：高温故障  Temporary\_Error：临时性故障  Voltage\_Error:电压故障  Antenna\_Error:天线无采集数据 |

**样例：**

POST /radarDataCollect/fault HTTP/1.1

HOST: 192.168.1.85:789

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 73

{

"DeviceNo": "1",

"Timestamp": "2017-11-10 13:04:05.354",

"FaultType":” Voltage\_Error”

}

1. 事件平台通信协议

广域雷达微波检测器通过RJ45网络传输方式给事件平台提供交通异常事件信息（分交叉口交通事件检测和路段交通事件检测）。

* 1. 交叉口交通事件数据

**接口说明**：雷达实时推送交叉口发生的交通事件相关信息，如时间，车道，类型等。常见交叉口交通事件类型有排队超限、溢出等。溢出和排队超限如果长期存在，10秒上报一次，如果超过10秒未上报，表示状态清除。

**推送地址**： /radarDataCollect/eventData/

**推送频率**：实时

* + 1. 排队超限事件

**接口说明**：广域雷达微波检测器检测到排队末车位置超过设定的排队超限位置时，实时推送排队超限事件信息，包括排队长度、排队车辆数等。

**调用方式**：雷达服务推送数据

**推送地址**： /radarDataCollect/eventData/QueueOverrun

**数据说明**：

表94排队超限事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo | Int | Y | 车道编号 |
| Length | Float | Y | 排队长度 |
| HeadPos | Float | Y | 头车位置 |
| TailPos | Float | Y | 尾车位置 |
| Num | Int | Y | 排队车辆数 |

**样例：**

POST /radarDataCollect/eventData/QueueOverrun HTTP/1.1

HOST: 192.168.1.128:8088

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 359

{

"DeviceNo": "212",

"Timestamp": "2019-08-21 20:29:01.749",

"Queue\_List": [{

"LaneNo": 1,

"Length": 6,

"HeadPos": 0,

"TailPos": 6,

"Num": 1

}, {

"LaneNo": 3,

"Length": 6,

"HeadPos": 1,

"TailPos": 7,

"Num": 1

}, {

"LaneNo": 4,

"Length": 29,

"HeadPos": 1,

"TailPos": 30,

"Num": 4

}]

}

* + 1. 溢出事件

**接口说明**：广域雷达微波检测器检测到溢出事件检测区域内的空间占有率高于设定阈值且平均速度低于设定阈值时，实时推送溢出事件信息。

**调用方式**：雷达服务推送数据

**推送地址：**/radarDataCollect/eventData/OverFlow

**数据说明：**

表95 溢出事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长 20 位 |
| Timestamp | String | Y | 时间戳 |

**样例：**

POST /radarDataCollect/eventData/OverFlow HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 61

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:45:04.357"

}

* + 1. 行人事件

**接口说明：**广域雷达微波检测器在行人检测区域内检测到目标且目标XY方向速度分别满足一定要求时，实时推送行人事件信息，包括目标坐标与速度等。

**调用方式**：雷达服务推送数据

**推送地址：**/radarDataCollect/eventData/Pedestrian\_Detection

**数据说明：**

表96 行人检测事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号 |
| Timestamp | String | Y | 时间戳 |
| Obj\_List | Array | Y | 目标信息列表 |
| ID | Int | Y | 行人编号 |
| Length | Float | Y | 目标大小(m) |
| XPos | Float | Y | 目标行人X坐标(m) |
| YPos | Float | Y | 目标行人Y坐标(m) |
| XSpeed | Float | Y | 目标行人X方向速度(km/h) |
| YSpeed | Float | Y | 目标行人Y方向速度(km/h) |

**样例：**

POST /radarDataCollect/eventData/Pedestrian\_Detection HTTP/1.1

HOST: 192.168.1.102:8080

Accept: application/json

Content-Type: application/json;charset=utf-8

Content-Length: 257

{"DeviceNo":"216",

"Timestamp":"2018-03-06 08:20:06.660",

"Obj\_List":[{

"ID":7,

"Length":1.500,

"XPos":-9.288,

"YPos":-8.853,

"XSpeed":-1.750,

"YSpeed":3.250

},{

"ID":9,

"Length":1.500,

"XPos":-7.214,

"YPos":8.873,

"XSpeed":0.250,

"YSpeed":-2.201}]

}

* 1. 路段交通事件数据

**接口说明**：雷达实时推送路段内发生的交通事件相关信息，如时间，车道，类型等。常见的路段交通事件类型有停车、超速、拥堵、占用应急车道等。其中，拥堵事件当拥堵级别改变时会再次推送，当拥堵级别推送“0”时，表示拥堵状态解除。

**推送地址**： /radarDataCollect/eventData/

**推送频率**：实时

* + 1. 停车事件

**接口说明：**停车事件包括超违法停车事件和异常停车事件，推送的事件信息中以停车类型作区分。

违法停车事件：广域雷达微波检测器在违停事件检测区域内检测到目标速度为零且静止时间不小于某一设定值时，实时推送违停事件信息，包括车道号、目标坐标等。

异常停车事件：广域雷达微波检测器在违停事件检测区域外、异停事件检测区域内检测到车道内目标速度为零、与前车间距大于一定距离且静止时间不小于某一设定值时，实时推送异停事件信息，包括车道号、目标坐标等。

**调用方式**：雷达服务推送数据

**推送地址：**/radarDataCollect/eventData/Parking

**数据说明：**

表97 停车事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo | Int | Y | 车道编号 |
| ParkingType | String | Y | Illegal\_Parking(违法停车)  Abnormal\_Parking (异常停车) |
| XPos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

**违停**

POST /radarDataCollect/eventData/Parking HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 146

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:49:16.956",

"LaneNo": 2,

"ParkingType": "Illegal\_Parking",

"XPos": 78,

"YPos": 3

}

**异停**

POST /radarDataCollect/eventData/Parking HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 147

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:49:16.956",

"LaneNo": 2,

"ParkingType": " Abnormal\_Parking ",

"XPos": 78,

"YPos": 3

}

* + 1. 超速事件

**接口说明：**超速事件包括超低速事件和超高速事件，推送的事件信息中以超速类型作区分。

超低速事件：广域雷达微波检测器在超低速事件检测区域内检测到目标速度大于一定值但低于最低限速值，且与前车间距大于一定距离时，实时推送超低速事件信息，包括车道号、目标速度与坐标等。

超高速事件：广域雷达微波检测器在超高速事件检测区域内检测到目标速度大小超过车道最高限速时，实时推送超高速事件信息，包括车道号、目标速度与坐标等。

**调用方式**：雷达服务推送数据

**推送地址**：/radarDataCollect/eventData/Speeding

**数据说明**：

表98超速事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo | Int | Y | 车道编号 |
| SpeedType | String | Y | High\_speed(超高速行驶)  Low\_speed(超低速行驶) |
| Speed | Float | Y | 速度(km/h) |
| XPos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

**超低速：**

POST /radarDataCollect/eventData/Speeding HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 150

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:37:05.624",

"LaneNo": 3,

"SpeedType": "Low\_speed",

"Speed": 4,

"XPos": 78,

"YPos": 3

}

**超高速：**

POST /radarDataCollect/eventData/Speeding HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 152

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:42:09.259",

"LaneNo": 4,

"SpeedType": "High\_speed",

"Speed": 50,

"XPos": 78,

"YPos": 3

}

* + 1. 逆行事件

**接口说明：**广域雷达微波检测器在逆行事件检测区域内检测到目标速度方向与车道方向相反、速度大小达到一定值且行驶时间不小于某一设定值时，实时推送逆行事件信息，包括车道号、目标速度与坐标等。

**调用方式**：雷达服务推送数据

**推送地址**：/radarDataCollect/eventData/Retrograde

**数据说明**：

表99逆行事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo | Int | Y | 车道编号 |
| Speed | Float | Y | 逆行速度(km/h) |
| Xpos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

POST /radarDataCollect/eventData/Retrograde HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 123

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:37:42.259",

"LaneNo": 4,

"Speed": 35,

"XPos": 78,

"YPos": 3

}

* + 1. 拥堵事件

**接口说明：**广域雷达微波检测器检测到拥堵事件检测区域内的空间占有率高于设定阈值且平均速度低于设定阈值时，实时推送拥堵事件信息，包括拥堵方向、拥堵级别、路段车辆数、平均速度等。

拥堵事件可设置三个拥堵级别，不同拥堵级别可分别设置相应的空间占有率、平均速度阈值。当拥堵级别改变时会再次推送拥堵事件信息，当拥堵级别推送“0”时，表示拥堵状态解除。

**调用方式**：雷达服务推送数据

**推送地址**：/radarDataCollect/eventData/Congestion

**数据说明**：

表100拥堵事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长 20 位 |
| Timestamp | String | Y | 时间戳 |
| Direction | String | Y | 方向：oncoming,outgoing |
| CongestionGrade | Int | Y | 0/1/2/3(数字越大表示拥堵越严重,0表示拥堵消散) |
| Vehicle\_Num | Int | Y | 路段内车辆数（辆） |
| AVSpeed | Float | Y | 平均速度（km/h） |

**样例：**

POST /radarDataCollect/eventData/ Congestion HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 155

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:32:19.063",

"Direction": "oncoming",

"CongestionGrade": 2,

"Vehicle\_Num": 8,

"AVSpeed": 16

}

* + 1. 变道事件

**接口说明：**广域雷达微波检测器在变道事件检测区内检测到车辆越过临近车道中线一定距离时，实时推送变道事件信息，包括变道前后的车道号、目标的坐标与速度等。

**调用方式**：雷达服务推送数据

**推送地址：** /radarDataCollect/eventData/Lane\_Change

**数据说明：**

表101变道事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长 20 位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo\_Original | Int | Y | 变道前车道编号 |
| LaneNo\_Current | Int | Y | 变道后车道编号 |
| Speed | Float | Y | 速度(km/h) |
| XPos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

POST /radarDataCollect/eventData/Lane\_Change HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 155

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:37:19.063",

"LaneNo\_Original": 2,

"LaneNo\_Current": 1,

"Speed": 35,

"XPos": 78,

"YPos": 3

}

* + 1. 占用应急车道事件

**接口说明：**广域雷达微波检测器在占用应急车道检测区内的应急车道中检测到目标时，实时推送占用应急车道事件信息，包括车道号、目标的坐标与速度等。

**调用方式**：雷达服务推送数据

**推送地址**：/radarDataCollect/eventData/Occupied\_Emergency\_Lane

**数据说明**：

表102占用应急车道事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长20位 |
| Timestamp | String | Y | 时间戳 |
| LaneNo | Int | Y | 车道编号 |
| Speed | Float | Y | 速度(km/h) |
| XPos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

POST /radarDataCollect/eventData/Occupied\_Emergency\_Lane HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 123

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:53:27.688",

"LaneNo": 5,

"Speed": 18,

"XPos": 78,

"YPos": 3

}

* + 1. 闯入事件

**接口说明：**广域雷达微波检测器在闯入事件检测区域检测到目标时，实时推送闯入事件报警信息，包括目标的坐标与速度等。

**调用方式**：雷达服务推送数据

**推送地址：**/radarDataCollect/eventData/Restricted\_Area

**请求 JSON 数据说明：**

表103闯入事件主动上传表

|  |  |  |  |
| --- | --- | --- | --- |
| **属性名称** | **字段类型** | **是否必填** | **字段说明** |
| DeviceNo | String | Y | 设备编号，最长 20 位 |
| Timestamp | String | Y | 时间戳 |
| Speed | Float | Y | 速度(km/h) |
| XPos | Float | Y | 目标X方向坐标（m） |
| YPos | Float | Y | 目标Y方向坐标（m） |

**样例：**

POST /radarDataCollect/eventData/Restricted\_Area HTTP/1.1

HOST: 192.168.1.102:4011

Accept: application/json

Content-Type: application/json;charset=utf-8

Connection: keep-alive

Content-Length: 108

{

"DeviceNo": "201",

"Timestamp": "2018-01-30 13:42:04.357",

"Speed": 17,

"XPos": 78,

"YPos": 3

}