广域雷达微波车辆检测器
交通平台通讯协议 V2.1

南京慧尔视智能科技有限公司

2018 年 9 月

目　　录

修订记录

版本	修改日期	修改人	修改内容
2.0.0	2017-10-25	陈俊德	制定新版 2.0 协议
2.1.0	2018-04-20	陈俊德	审核修订，分割协议，确立版本号。
2.1.1	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
2.1.2	2018-09-11	王栋	新增大数据平台链路设置、获取； 新增动态排队数据； 新增转向速度

1. 系统简介

广域雷达微波车辆检测器 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 DTAM D29 广域雷达微波车辆检测器

2. 系统架构

广域雷达微波车辆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却别，主要系统架构如下图所示。

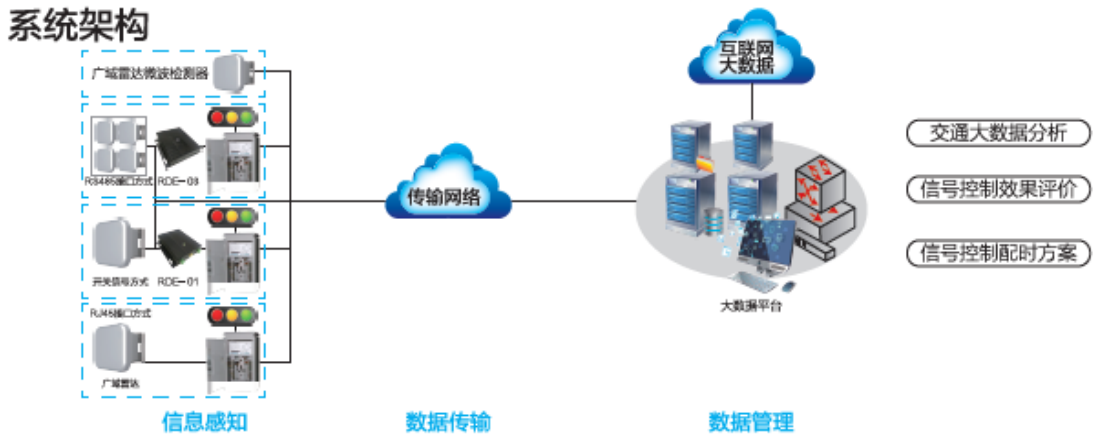


图 2-1 系统对接架构图

3. 术语说明

✧ **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。

- **排队长度**

  车辆在接近停止线时受信号控制由行驶状态改变为静止状态，车道内所有静止车辆队列的第一辆车车头到最后车辆车尾的距离（以渠化段车道划分为依据）。

- **排队首车位置**

  车道排队队列首车距停止线的距离。

- **排队末车位置**

  车道排队队列末车距停止线的距离。

- **排队数量**

  车道排队队列末中的车辆数目。

- **区域车辆数**

  检测区域内，车道内的所有动态静态车辆的数量。

- **分布情况**

  检测区域内，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**

  检测区域内，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**

  检测区域内，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**

  检测区域内，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**

  检测区域内，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。

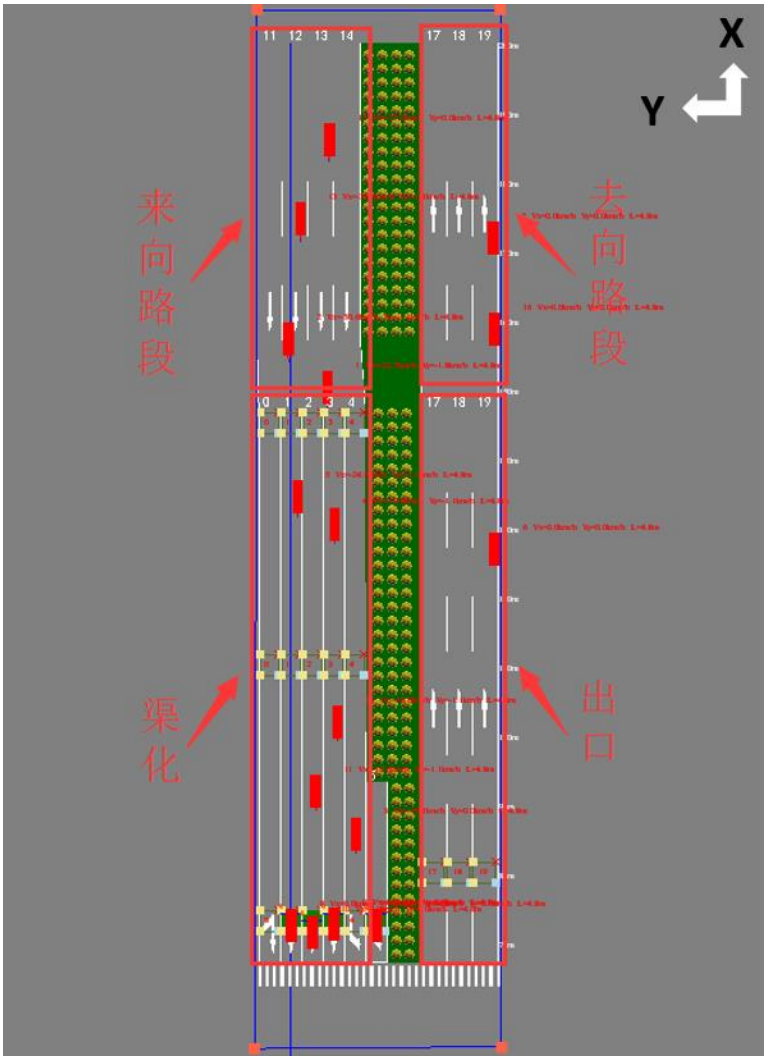
4. 协议概述

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。

5. 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。

5.1 雷达坐标系介绍



5.2 获取安装信息

请求地址：/radarConfigure/getInstallationCfg

请求样例:

```
GET /radarConfigure/getInstallationCfg HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
```

```
Content-Type: application/json
```

响应消息：

表 1 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
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"
}
```

5.3 获取车道配置

请求地址：/radarConfigure/getLaneCfg

请求样例:

```
GET /radarConfigure/getLaneCfg HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 2 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
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
        }]
}
```


5.4 获取测量线配置

请求地址：/radarConfigure/getMeaslineCfg

请求样例：

```
GET /radarConfigure/getMeaslineCfg HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 3 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
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
    }]
}
```

5.5 获取线圈配置

请求地址：/radarConfigure/getCoilCfg

请求样例：

```
GET /radarConfigure/getCoilCfg HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 4 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
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
    }]
}
```


5.6 获取设备时间

请求地址：/radarConfigure/getDeviceTime

请求样例：

```
GET /radarConfigure/getDeviceTime HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 5 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
Connection: close
Content-Type: application/json;charset=utf-8
Content-Length: 50

{
    "DeviceNo":"204",
    "DeviceTime":"2018-07-30 12:00:00"
}
```

5.7 设置系统时间

请求地址：/radarConfigure/setDeviceTime

请求数据：

表 6 设置系统时间

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: 127.0.0.1:15000
Accept: application/json
Content-Type: application/json
Content-Length: 34

{
    "DeviceTime":"2018-04-28 20:00:00"
}
```

响应消息：

```
设置成功：
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html
Content-Length: 0
```

5.8 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: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 7 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
Connection: close
Content-Type: application/json;charset=utf-8
Content-Length: 29

{
    "DeviceNo":"204",
    "Cycle":10
}
```

5.9 设置交通平台统计周期

请求地址：/radarConfigure/setCycle

请求数据：

表 8 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: 127.0.0.1:15000
Accept: application/json
Content-Type: application/json
Content-Length: 21

{
    "Cycle":60
}
```

响应消息：

设置成功：

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html
```

```
Content-Length: 0
```

5.10 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: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 9 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	即时数据推送频率 单位 100ms，0 表示不推送
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
Connection: close
Content-Type: application/json;charset=utf-8
Content-Length: 147

{    "DeviceNo":"204",
     "ObjDataEnable":10,
     "QueueEnable":"True",
     "RoadEnable":"False",
     "CycleEnable":"True",
     "EvaluationEnable":"True",
     "PassEnable":"False"
}
```

5.11 设置数据使能状态

请求地址：/radarConfigure/setEnable

请求消息：

表 10 数据使能状态信息

属性名称	字段类型	是否必填	字段说明
ObjDataEnable	Int	Y	即时数据推送频率 单位 100ms，0 表示不推送
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）

样例

```
POST /radarConfigure/setEnable HTTP/1.1
Host: 127.0.0.1:15000
Accept: application/json
Content-Type: application/json
Content-Length: 152

{
    "ObjDataEnable":10,
    "QueueEnable":"True",
    "RoadEnable":"False",
    "CycleEnable":"True",
    "EvaluationEnable":"True",
    "PassEnable":"False"

}
```

响应消息：

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html
Content-Length: 0
```

5.12 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: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 11 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

样例

```
HTTP/1.1 200 OK
Connection: close
Content-Type: application/json;charset=utf-8
Content-Length: 57

{
    "DeviceNo":"202",
    "IPAddress":"192.168.1.83",
    "Port":4005
}
```

5.13 设置交通平台链路

请求地址：/radarConfigure/setTrafficIPPort

请求消息：

表 12 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

样例

```
POST /radarConfigure/setTrafficIPPort HTTP/1.1
Host: 192.1.1.201:15000
Accept: application/json
Content-Type: application/json
Content-Length: 52

{
    "IPAddress":"192.168.1.88",
    "Port":4005

}
```

响应消息：

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html
Content-Length: 0
```

5.14 获取大数据平台链路

请求地址：/radarConfigure/getBigDataIPPort

请求样例：

```
GET /radarConfigure/getBigDataIPPort HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
```

响应消息：

表 13 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	大数据平台 IP
Port	Int	Y	大数据平台端口

样例

```
HTTP/1.1 200 OK
Connection: close
```

```
Content-Type: application/json;charset=utf-8
Content-Length: 57

{
    "DeviceNo":"202",
    "IPAddress":"192.168.1.83",
    "Port":4005
}
```

5.15 设置大数据平台链路

请求地址：/radarConfigure/setBigDataIPPort

请求消息：

表 14 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	大数据平台 IP
Port	Int	Y	大数据平台端口

样例

```
POST /radarConfigure/setBigDataIPPort HTTP/1.1
Host: 192.1.1.201:15000
Accept: application/json
Content-Type: application/json
Content-Length: 52

{
    "IPAddress":"192.168.1.88",
    "Port":4005

}
```

响应消息：

```
HTTP/1.1 200 OK
Connection: close
Content-Type: text/html
Content-Length: 0
```

5.16 请求历史数据（断点续传）

请求地址：/radarHistoricalData/getData

请求消息：

表 15 断点续传

属性名称	字段类型	是否必填	字段说明
BeginTime	String	Y	查询开始时间
EndTime	String	Y	查询结束时间

请求样例：

```
POST /radarHistoricalData/getData HTTP/1.1
Host: 192.168.1.201:15000
Accept: application/json
Content-Type: application/json
Content-Length: 77

{
    "BeginTime":"2018-04-28 10:00:00",
    "EndTime":"2018-04-28 20:00:00"
}
```

注：

因大范围的查询和传输非常消耗雷达的资源，避免高峰时段降低雷达性能，约定的请求时间在凌晨 1:00~4:00 之间，其他时间不响应。平台可在 1:00 后检查数据库是否缺失数据，并发出断点续传请求，3:00 再检查一次，补发断点续传请求。请求后，按时间范围，在存储器内进行检索，按交通平台设置的 IP 和端口，及章节 6.5 和 6.6 协议推送历史数据。不同时段的多次请求，检测器将按顺序依次处理。约定每个请求的时间范围不超过 1 小时，如果超出，将以起始时间开始 1 小时范围内查询和传输数据。大范围的缺失数据，应以多条请求的方式实现。

检测器的统计和评价两个数据表，默认按 5 万条循环存储，可根据周期时长、线圈数量、当前时间，计算缺失的数据是否存在，如已超出存储范围，数据已经被覆盖，请求断点续传将没有意义。如需更大离线存储容量，采购时应增加存储卡需求。已出厂的设备，无法实现存储容量升级。

响应消息：

```
HTTP/1.1 200 OK
```

Connection: close
Content-Type: text/html
Content-Length: 0

5.17 错误回复

请求地址错误回复：
HTTP/1.1 404 Not Found
Connection: close
Content-Type: text/html
Content-Length: 0

请求内容错误回复：
HTTP/1.1 403.10 Configuration Error
Connection: close
Content-Type: text/html
Content-Length: 0

6. 雷达数据通讯协议

平台应当建立 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 16 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

6.1 即时数据

接口说明：推送目标信息只平台，用于在线仿真，具体数据格式有两种方式，见下文。

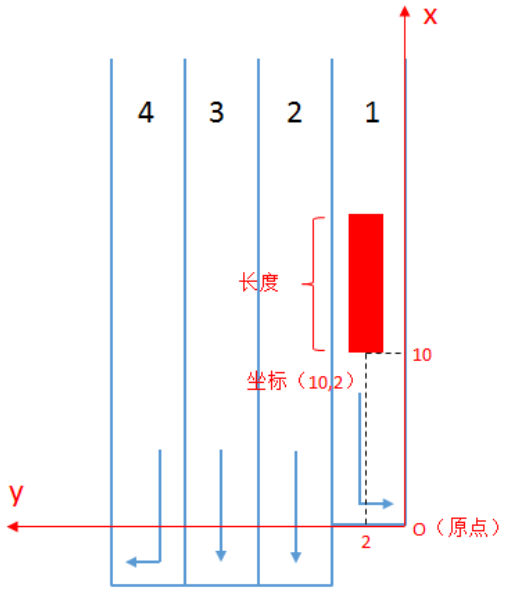
调用方式：微波服务推送数据

推送地址： /radarDataCollect/objData

推送频率：频率可设置，见章节 5.9“设置数据使能状态”。

方式一（默认）：

当背景是根据道路情况真实绘制时，以渠化车道最右边车道顶点为原点的 XY 坐标系，推送实时目标编号、长度、位置（XPos，YPos）、速度（XSpeed，YSpeed），见下图。



请求 JSON 数据说明：

表 17 即时数据主动上传表（方式一）

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
Accept: application/json
```

Content-Type: application/json;charset=utf-8
Connection: keep-alive
Content-Length: 482

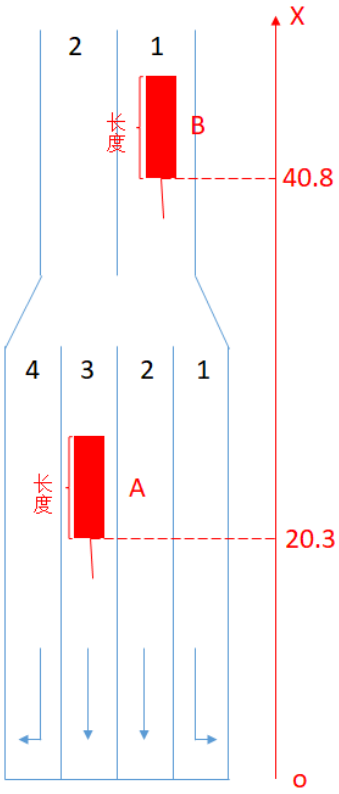
{
"DeviceNo": "51020414",
"Timestamp": "2017-11-17 13:57:20.265",
"Obj_List": [{
"ID": 1,
"Length": 4.800000,
"XPos": 20.000011,
"YPos": 3.700004,
"XSpeed": -14.750000,
"YSpeed": -0.250000
}, {
"ID": 2,
"Length": 4.800000,
"XPos": -5.200001,
"YPos": 15.299994,
"XSpeed": -8.500000,
"YSpeed": 0.250000
}]
}

方式二（定制）：

当背景仅绘制了车道，并未根据实际宽度和长度绘制时，推送实时目标编号、长度、位置（车道类型、车道编号、车道内距离停止线位置）、速度（XSpeed，YSpeed），见下图。

目标 A 信息：ID:1，Length：4.1，LaneNo：3，LaneType：0，ObjPos：20.3，XSpeed：35.2，YSpeed：1.3

目标 B 信息：ID:2，Length：4.3，LaneNo：1，LaneType：1，ObjPos：40.8，XSpeed：46.7，YSpeed：1.8



请求 JSON 数据说明：

表 18 即时数据主动上传表(方式二)

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)
LaneNo	Int	Y	目标车辆所在车道编号
LaneType	Int	Y	目标车辆所在车道类型（0：进口道，1：来向路段，2：出口道，3：去向路段）
ObjPos	Float	Y	目标车辆距离停止线的距离（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)

6.2 过车信息

接口说明：车辆压占线圈并离开线圈后，实时推送过车信息。

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 19 过车信息主动上传表

属性名称	字段类型	是否必填	字段说明
DeviceNo	String	Y	设备编号
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	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
}
```

6.3 静态排队数据

接口说明：雷达服务每秒推送各设备实时排队数据，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 20 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	Int	Y	排队长度（m）
Queue_Head	Int	Y	排队第一辆车距离停止线距离（m）
Queue_Tail	Int	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":  0,
            "Queue_Head": 0,
```

```
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":2,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":3,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":4,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }]
}
```


6.4 动态排队数据

接口说明：雷达服务每秒推送各设备实时排队数据，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距可设。

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 21 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	Int	Y	排队长度（m）
Queue_Head	Int	Y	排队第一辆车距离停止线距离（m）
Queue_Tail	Int	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/queueDataStaticDynamic 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":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":2,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":3,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
            "Queue_Num":  0
        }, {
            "LaneNo":4,
            "Queue_Len":  0,
            "Queue_Head": 0,
            "Queue_Tail": 0,
```

```
            "Queue_Num":  0
        }]
}
```

6.5 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、车道内排列情况、最后一辆车距离停止线距离、车道内所有车平均速度。

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 22 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	Int	Y	车道内第一辆车位置（m）
Head_Speed	Int	Y	头车速度（km/h）
Last_Pos	Int	Y	车道内最后一辆车位置（m）
Last_Speed	Int	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":    1,
            "Occupancy":  0.035294,
            "AVSpeed":    30.600000,
            "Pareto":0,
            "Head_Pos":   20,
            "Head_Speed": 30,
            "Last_Pos":   20,
            "Last_Speed": 30
        }, {
            "LaneNo":4,
            "Vehicle_Num":    0,
            "Occupancy":  0,
            "AVSpeed":    0,
            "Pareto":0,
            "Head_Pos":   0,
            "Head_Speed": 0,
            "Last_Pos":   0,
            "Last_Speed": 0
        }]
}
```

6.6 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 23 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	Int	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: 5155

{
    "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
        }]
}
```


6.7 评价指数

接口说明：雷达服务按统计周期推送各设备评价指数，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 24 评价指数主动上传表

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
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	各车道评价指数信息列表
LaneNo	Int	Y	车道编号
Volume	Int	Y	车道内过停止线流量（辆）
MaxQueueLen	Int	Y	车道内最大排队长度（m）

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）
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: 617

{
    "DeviceNo":"203" ,
    "Timestamp":"2018-09-17 15:07:21" ,
    "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 ,
        "Stops":0 ,
        "Delay":0 ,
        "Fuel":21 ,
        "Emissions":270.899994
    } ,{
        "LaneNo":6 ,
        "Volume":3 ,
        "MaxQueueLen":0 ,
        "Stops":0 ,
        "Delay":1 ,
        "Fuel":21.240000 ,
```

```
        "Emissions":273.996002
    } ,{
        "LaneNo":5 ,
        "Volume":5 ,
        "MaxQueueLen":6 ,
        "Stops":0.200000 ,
        "Delay":14 ,
        "Fuel":24.360001 ,
        "Emissions":314.244019
    } ,{
        "LaneNo":4 ,
        "Volume":0 ,
        "MaxQueueLen":6 ,
        "Stops":0 ,
        "Delay":0 ,
        "Fuel":0 ,
        "Emissions":0
    }]
}
```


6.8 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 25 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"
}
```