**东大对接协议**

**轨迹数据推送协议（kafka）**

* Topic: trafficTrajectory
* 发送方式：每个设备的每一帧分别作为一条 JSON 消息推出
* 发送周期
* 隼眼雷达: 50ms (20 fps)
* 视频(25fps)

**消息结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必须 | 字段说明 | 备注 |
| deviceID | string | 是 | 设备名称 | 雷达或者摄像机名称 |
| deviceType | string | 是 | 设备类型 | 1-雷达、2-摄像机 |
| targets | array | 是 | 轨迹数组 | 数组，元素结构见下目标结构 |

**目标结构**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必须 | 字段说明 | 备注 |
| timestamp | int64 | 是 | 时间戳 | 数据发送时间，Unix 毫秒时间戳。 |
| id | int64 | 否 | 目标id | 目标轨迹编号，不重复，递增循环 |
| lane | uint32 | 是 | 目标车道号 | 目标所在车道位置编号 |
| y | double | 是 | 纵向距离 | 设定原点，构建局部坐标系，目标的纵向距离，单位 m |
| x | double | 是 | 横向距离 | 设定原点，构建局部坐标系，目标的横向距离，单位 m |
| cls | uint32 | 是 | 目标类型 | 类型：0-未定义目标，1-car，2-truck，3-bus，4-摩托，5-自行车，6-行人 |
| speed | double | 是 | 速度 | 单位: km/h |
| vx | double | 是 | X 轴速度 | 单位: km/h |
| vy | double | 是 | Y 轴速度 | 单位: km/h |
| latitude | double | 是 | 纬度 | WGS84坐标系 |
| longitude | double | 是 | 经度 | WGS84坐标系 |

示例

|  |
| --- |
| JSON {  "deviceID": "K68+366",  "deviceType": 1,  "targets": [  {  "timestamp": 1705910562380,  "id": 696,  "lane": 2,  "x": -7.084577829007044,  "y": 7.085103535729672,  "latitude": 33.3333,  "longitude": 111.1111,  "cls": 0,  "speed": 63.20526478463193,  "vx": 1.0,  "vy": 60.0  },  {  "timestamp": 1705910562380,  "id": 52,  "lane": 2,  "x": -93.31471856189998,  "y": -1.4275749755241487,  "latitude": 33.3333,  "longitude": 111.1111,  "cls": 0,  "speed": 59.291003819294794,  "vx": 1.0,  "vy": 60.0  }  ] } |

**事件接收协议（HTTP）**

|  |
| --- |
| HTTP POST /raoyue/seu-events Content-Type: application/json |

|  |
| --- |
| JSON {  "type": 1,  "level": 1,  "start\_time": "2023-12-06 11:11:11",  "end\_time": "2023-12-06 11:11:11",  "lane": 1,  "raw\_class": 1,  "point\_wgs84": {  "lat": 33.33,  "lon": 111.11  },  "device\_type": 1,  "device\_id": "K70+800" } |

* 每个请求上报一起事件，具体字段定义如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否必须 | 字段说明 | 备注 |
| type | int64 | 是 | 事件类型 | 0-抛洒物，1-机动车异常停车，2-行人闯入，3-机动车逆行 |
| level | int64 | 否 | 事件等级 | 预留字段 |
| start\_time | string | 是 | 事件开始时间 |  |
| end\_time | string | 是 | 事件结束时间 |  |
| lane | uint32 | 是 | 车道号 |  |
| raw\_class | int64 | 是 | 目标类型 | 类型：0-未定义目标，1-car，2-truck，3-bus，4-摩托，5-自行车，6-行人 |
| point\_wgs84 | Object | 是 | 经纬度坐标 | 参考例子 |
| device\_type | string | 是 | 设备类型 |  |
| device\_id | string | 是 | 设备 id |  |

**事件类型说明**

根据项目前述监测的事件，具体包括：

["spill", "stop", "lowSpeed", "highSpeed", "emergencyBrake", "incident", "crowd", "illegalOccupation"]

抛洒物监测，停车监测，低速行驶，超速行驶，急刹车，交通事故，车道拥堵，非法占用应急车道。

**上报事件说明**

抛洒物检测基于道路元胞检测，无法给出具体的latlon，可以根据lane确认出现抛洒物的车道。借用latlon，lat用于存储元胞的order，lon用于存储元胞的start(该设备监测范围内, 从该lane正向行驶的方向，在监测范围内从道路边界值，逐个划分cell的起始点位置，order是元胞划分出的序号)。故从而确定发生抛洒物的位置。

同样的，车道拥堵也无法给出具体latlon，lat用于存储laneID，lon用于存储交通量q。

**部署时可能需要调整的地方**

（1）参数config.yml

根据应用场景(快速路/高速)，调整config数值

（2）标定clb.yml

注意部署测试可能跟开发的测试数据场景不同，需要在config中开启recalib,对clb.yml进行重新标定。当前标定仅考虑单设备。

（3）后期如果协议有变化

如果来的数据有修改，在driverOnline的receive修改。

如果发的event协议有修改，在driverOnline的\_eventsInner2Outer修改。