自动驾驶软件说明书

——柴志军，2022.05.05

# 1 系统框架

## **1.1总体架构**

享智ROS环境总体架构遵循 “传感驱动--感知--规划--控制--底盘运动” 的逻辑顺序。不同的功能模块依照功能划分若干节点模块，节点间通过topic通信互相连接。

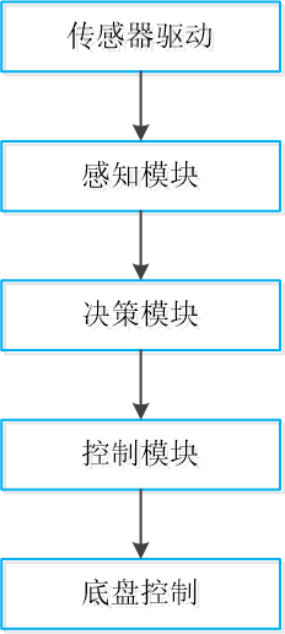


图1.1 享智ROS环境软件功能模块示意图

## **1.2模块架构**

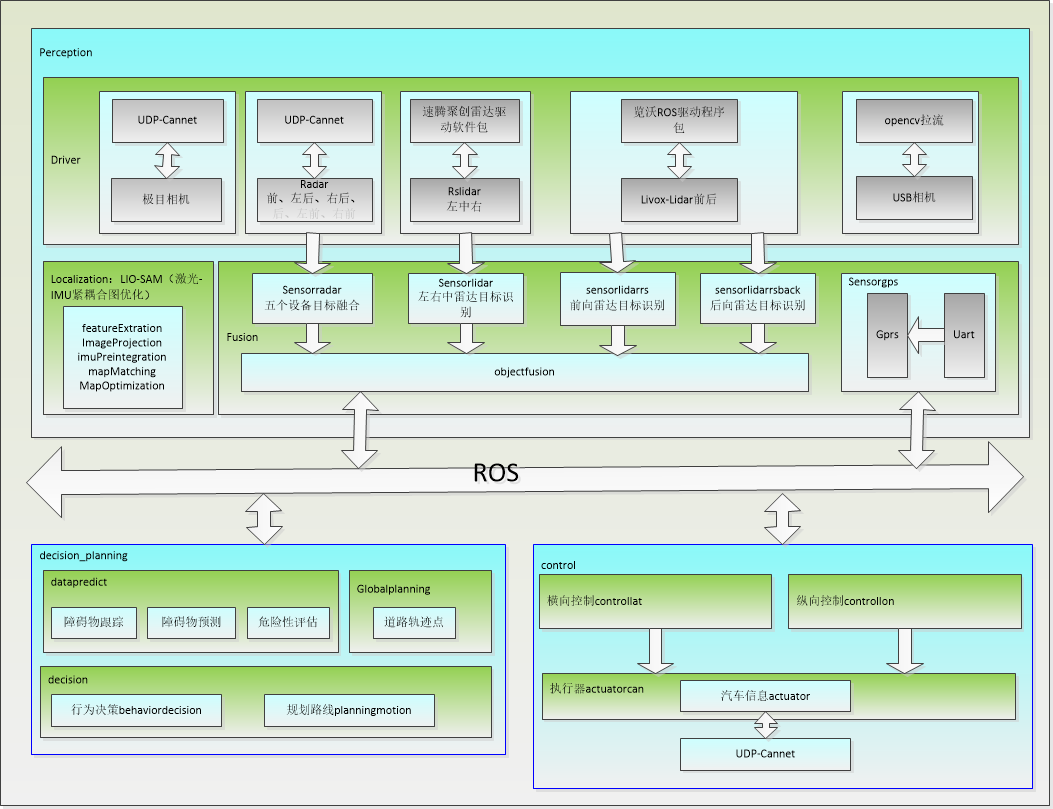


图1.2 模块架构图

整个系统框架分为系统感知perception、决策规划decision\_planning、控制control，三部分。消息体都是通过Ros topic通信互相传输。

（1）感知perception由传感驱动、融合、定位，三部分组成：

传感驱动，负责读取各个传感器设备的原始数据，包括极目相机、USB相机、五个毫米波雷达、三个速腾聚创激光雷达、两个大疆激光雷达等；

融合，先由各个传感模块自行目标融合，再进行全部目标融合。同时进行组合导航sensorgps，输出gps信息；

定位，使用lio\_sam进行高精度定位。

（2）决策规划，包含globalplanning、datapredict、decision三部分：

Globalplanning，节点负责加载离线地图，根据车辆当前位置栅格化地图并输出；

Datapredict，实现障碍物跟踪、障碍物预测、危险性评估等功能；

Decision，依据定位、车辆、地图、控制、障碍物等信息，给出适当驾驶行为和路径规划。

（2）控制，包括横向控制controllat、纵向控制controllon、执行器actuatorcan、桥接bridgecan：

横向控制controllat，主要输出车辆转向信息；

纵向控制controllon，主要输出车加减速信息；

执行器actuatorcan，通过Cannet获取"actuator"汽车信息；

桥接bridgecan，从EMUC202中读取can数据，并通过udp将can接收到的radar数据发送给radarleftback\_driver和radarrightback\_driver。

# 2 系统感知

## **2.1传感器驱动**

### **2.1.1 USB相机**

perception/driver/jmcamera/src/sensorcamera中，使用opencv拉取视频流，依据ROI参数裁剪得到所需图像，输出红绿灯信息(sensorcameralight)。

表 2.1 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| usbcameralight | usbcamera0 | “sensorgps” | "cameralight"、"usbcamera\_status" |

### **2.1.2 极目相机**

perception/driver/jmcameralight/src/EyecameraNode中，udp（port:4003）读取can（port:4003）报文信息，输出车道线中心线(cameracenter)、障碍物数据（eyeobjects）。

表 2.2 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| eyecamera | eyecamera | none | "eyecamera"、"cameracenter"、"centertrajectory"、"eyeobjects"、"jmcamera\_status" |

### **2.1.3 毫米波radar**

perception/driver/jmcameralight/src/EyecameraNode中，udp（port:4003）读取can（port:4003）报文信息，协议解析障碍物信息和车道线信息。

在本版本中，只用到了前、左后、右后三个雷达，使用的是deal\_radar.launch文件。

表 2.3 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| radar\_driver | radarfront\_driver | "sensorgps" | "/radarfront\_driver/radardriver" |
| radarleftback\_driver | "sensorgps" | "/radarleftback\_driver/radardriver" |
| radarrightback\_driver | "sensorgps" | "/radarrightback\_driver/radardriver" |
| radarleftfront\_driver | "sensorgps" | "/radarleftfront\_driver/radardriver" |
| radarrightfront\_driver | "sensorgps" | "/radarrightfront\_driver/radardriver" |
| radarback\_driver | "sensorgps" | "/radarback\_driver/radardriver" |
| radar\_deal | radarfront\_deal | "actuator"  "/radarfront\_driver/radardriver" | "/radarfront\_deal/radar" |
| radarleftback\_deal | "actuator"  "/radarleftback\_driver/radardriver" | "/radarleftback\_deal/radar" |
| radarrightback\_deal | "actuator"  "/radarrightback\_driver/radardriver" | "/radarrightback\_deal/radar" |
| radarleftfront\_deal | "actuator"  "/radarleftfront\_driver/radardriver" | "/radarleftfront\_deal/radar" |
| radarrightfront\_deal | "actuator"  "/radarrightfront\_driver/radardriver" | "/radarrightfront\_deal/radar" |
| radarback\_deal | "actuator"  "/radarback\_driver/radardriver" | "/radarback\_driver/radar" |

### **2.1.4 左中右lidar**

左中右激光雷达rslidar\_sdk，rs\_driver速腾聚创雷达驱动软件包，是以api接口方式提供和调用，节点调用在rslidar\_sdk/node/rslidar\_sdk\_node.cpp中。

### **2.1.5 前后lidar**

前后激光雷达livox\_ros\_driver，览沃ROS驱动程序包，需要安装和运行Livox-SDK，Ros实现在livor\_ros\_driver/livor\_ros\_driver.cpp中。

## **2.2 融合**

### **2.2.1 毫米波融合sensorradar**

perception/fusion/sensorradar/src/sensorradar.cpp，获取五个毫米波的目标后，毫米波目标融合（重叠目标剔除，不同目标融合）。

表 2.4 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| sensorradar | sensorradar | /radarleftfront\_deal/radar  /radarrightfront\_deal/radar  /radarleftback\_deal/radar  /radarrightback\_deal/radar  /radarfront\_deal/radar  radarback\_deal/radar | sensorradar  radar\_cloud  front\_radar\_status  rightfront\_radar\_status  leftfront\_radar\_status  back\_radar\_status  rightback\_radar\_status  leftback\_radar\_status |

### **2.2.2 左中右激光sensorlidar**

perception/fusion/sensorlidar/src/sensorlidar.cpp，第一步，雷达点云数据进行栅格化，减少数据量；第二步，利用高度信息对地面数据进行滤除；第三步，对剩下的障碍物点云信息利用最近邻算法进行聚类；第四步，聚类完成以后，将聚类后的数据发送给融合节点，进行下一步处理网格化前后向点云。

表 2.5 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| sensorlidar | sensorlidar | sensorgps  globalplanning  actuator  /vlp\_left/velodyne\_points  /vlp\_back/rslidar\_points  /vlp\_front/velodyne\_points  /vlp\_right/velodyne\_points  /vlp\_back/velodyne\_points | /points\_filter  /sensorlidar  /lidar\_left  /lidar\_mid  /lidar\_front  /lidar\_right  /lidar\_back  /cali\_lidar\_left  /cali\_lidar\_right  /cali\_lidar\_mid  /bounding\_boxes  left\_lidar\_status  right\_lidar\_status  mid\_lidar\_status  front\_lidar\_status  back\_lidar\_status |

### **2.2.3 前向激光sensorlidarrs**

perception/fusion/sensorlidarrs/src/sensorlidar.cpp，第一步，雷达点云数据进行栅格化，减少数据量；第二步，利用高度信息对地面数据进行滤除；第三步，对剩下的障碍物点云信息利用最近邻算法进行聚类；第四步，聚类完成以后，将聚类后的数据发送给融合节点，进行下一步处理网格化前后向点云。

表 2.6 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| sensorlidarrs | sensorlidarrs | globalplanning  /livox/lidar\_OTFDJ4V0014GM81 | rslidar\_obs  rslidar\_raw  sensorlidarrs  front\_lidar\_status |

### **2.2.4 后向激光sensorlidarrsback**

perception/fusion/sensorlidarrsback/src/sensorlidar.cpp，第一步，雷达点云数据进行栅格化，减少数据量；第二步，利用高度信息对地面数据进行滤除；第三步，对剩下的障碍物点云信息利用最近邻算法进行聚类；第四步，聚类完成以后，将聚类后的数据发送给融合节点，进行下一步处理网格化前后向点云。

表 2.7 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| sensorlidarrsback | sensorlidarrsback | /livox/lidar\_OTFDJ4T001TM101 | rslidar\_obsback  rslidar\_rawback  sensorlidarrsback  back\_lidar\_status |

### **2.2.5 组合导航sensorgps**

perception/fusion/sensorrgps/src/sensorgps.cpp，开启uart读取gps定位信息，并解析发布。回调app信息进行地图采集，回调app数据进行泊车点、红绿灯点采集，接收app数据请求进行地图数据上传。

表 2.8 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| sensorgps | sensorgps | actuator  collectmap  requestmap  collectpoint | sensorgps  /imu\_gps  lonlatmappoints  gps\_status |

### **2.2.6 目标融合objectfusion**

perception/fusion/src/objectfusion.cpp，获取激光目标后的topic处理，所有的激光目标集合，所有障碍物归类（相机障碍物、激光障碍物、毫米波障碍物），数据目标融合。

激光雷达可以提供目标的精准位置和粗略的速度信息，毫米波雷达可以提供目标的粗略位置和精准的相对速度信息，单目相机可以提供目标精准的类型、宽度信息。三者序列经过融合后，输出完整的位置、相对速度、宽度、类型信息。

表 2.9 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| objectfusion | objectfusion | sensorradar  sensorlidar  sensorlidarrs  sensorlidarrsback  globalplanning  eyecamera  sensorgps | objectfusion  fusion\_cloud  /fusion\_bounding\_boxes  /fusion\_cluster\_ids |

## **2.3 定位localization**

表 2.10 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| lio\_sam | lio\_sam\_featureExtration | lio\_sam/deskew/cloud\_info | lio\_sam/deskew/cloud\_info  lio\_sam/feature/cloud\_info  lio\_sam/feature/cloud\_corner  lio\_sam/feature/cloud\_surface |
|  | lio\_sam\_imageProjection | lio\_sam/imuTopic  lio\_sam/odomTopic  lio\_sam/pointCloudTopic | lio\_sam/deskew/cloud\_info  lio\_sam/deskew/cloud\_deskewed |
|  | lio\_sam\_imuPreintegration | lio\_sam/mapping/odometry  lio\_sam/odomTopic\_incremental  lio\_sam/gpsTopic  actuator  和lio\_sam/mapping/odometry\_incremental | lio\_sam/odomTopic  lio\_sam/imu/path  integrated\_pose  lio\_sam/lidargpsTopic  和  lio\_sam/imuTopic  lio\_sam/odomTopic\_incrememtal |
|  | lio\_sam\_mapMatching | lio\_sam/deskew/cloud\_info  lio\_sam/gpsTopic  actuator | /key\_gpspose\_origin  /key\_pose\_origin  /lio\_sam/mapping/trajectory  /lio\_sam/mapping/map\_global  /lio\_sam/mapping/odometry  /lio\_sam/mapping/odometry\_incrememtal  lio\_sam/mapping/path  /odometry/gps  reliability |
|  | lio\_sam\_mapOptimization | lio\_sam/feature/cloud\_info  lio\_sam/gpsTopic  lio\_loop/loop\_closure\_detection | /key\_gpspose\_origin  /key\_pos\_origin  lio\_sam/mapping/trajectory  lio\_sam/mapping/map\_global  lio\_sam/mapping/odometry  lio\_sam/mapping/odometry\_incremental  lio\_sam/mapping/path  /odometry/gps  lio\_sam/mapping/ipc\_loop\_closure\_history\_cloud  lio\_sam/mapping/icp\_loop\_closure\_corrected\_cloud  lio\_sam/mapping/icp\_loop\_closure\_constraints  lio\_sam/mapping/map\_local  lio\_sam/mapping/cloud\_registered  lio\_sam/mapping/cloud\_registered\_raw |

# 3 决策规划

## **3.1全局轨迹globalplanning**

decision\_planning/globalplanning/src/globalplanning，利用预先采集的GPS路径和实时的GPS定位结果，生成供车辆行驶的全局轨迹，该全局轨迹又称为一次轨迹。加载离线地图，根据车辆当前位置栅格化地图并输出。

表 2.11 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| globalplanning | globalplanning | sensorgps  app  changepath  actuator  behaviordecision  controllat  /integrated\_posereliability | globalplanning  globalplanningback  newmap  setchangedis  gpstrajectory  lidartrajectory |

## **3.2数据预测datapredict**

decision\_planning/datapredict/src/process，整个预测模块的主处理流程：有效数据获取、障碍物跟踪、ROI内障碍物标记、障碍物预测、危险性评估、栅格化、数据发布。响应算法函数都在src下的predict、subscribe（gps和obs）、track。

表 2.12 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| datapredick | datapredick | Sensorgps  Objectfusion  Actuator  Globalplanning  globalplanningback | obsinfo |

## **3.3行驶决策decision**

decision\_planning/decisionlon/src/decision\_lon，输出行为决策drivebehavior和规划路径点。

表 2.13 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| decisionlon | decisionlon | sensorgps  actuator  app  controllat  setchangedis  globalplanning  objectfusion  v2xapp  nextproperty  globalplanningback  monitor  appcontrol  sensorultra  predictresult  cameracenter | behaviordecision  reversemap  planningmotion  changepath |

# 4 控制control

控制模块接收来自规划模块的结果，并根据车辆当前状态决定车辆加速，刹车，转向等操作。

控制模块共有三个节点，actuator，controllat，controllon。

## **4.1横向控制controllat**

Control/controllat/src/controllat，横向控制节点主要输出车辆转向信息。输出信息：epsangle目标角度、apsstate、apavstatus、epsmethod转角使能、limitspeed限速、epstorque转角速度、isvalid有效位、deviation横向偏移距离、lights目标灯光、timestamp时间戳、curve。

表 2.14 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| controllat | controllat | behaviordecision  monitor  sensorgps  planningmotion  actuator  appLight  controllon | controllat |

## **4.2纵向控制controllon**

Control/controllon/src/controllon，纵向控制节点主要输出车加减速信息。发布输出信息：objtype、gasPedal、light、brakePedal、targetSpeed、objrel前方最危险目标的相对速度、objdis前方最危险目标距离、mode纵向控制模式、epb手刹控制状态、gear档位控制状态、apaDis、timestamp时间戳。

表 2.15 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| controllon | controllon | controllat  behaviordecision  sensorgps  actuator  obu  planningmotion  globalplanning  /camera/sensorcameralight  appMsg | controllon  controlshow |

## **4.3执行器actuatorcan**

control/actuatorcan/src/actuator（Emuc202）。通过udp（can端口8888，pc7777）读取can报文信息，协议解析自车信息：speed、EpbState、TurnLight、SysState、GearState、EPS\_Angle、EPS\_Angle、EPS\_Angle。通过udp转can发送横向和纵向控制信息到执行器。

表 2‑16 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| actuator | actuator | controllat  controllon  sensorgps  appMsg | actuator |

## **4.4桥接bridgecan**

Control/bridgecan/src/bridge（Emuc202）。通过udp读取can报文信息，canetport0：6666，pcport0：5555；canetport0：4444，pcport0：3333；canetport0：2222，pcport0：1111。

暂时没调用。

表 2.17 节点释义

|  |  |  |  |
| --- | --- | --- | --- |
| Pkg(包) | Node（节点） | 订阅Topic（消息） | 发布Topic（消息） |
| bridge | bridge |  |  |