Baseline\_planner

grid map 第1394行 .cast<类型>() 函数？Eigen库中的函数，向量元素类型强制转换。

map\_quat\_size 检测环境范围的大小；map\_shape向量保存地图形状；

x\_lim y\_lim z\_lim 每个方向栅格数量；

向量嵌套表示map[][][];

Class Astar 中的get\_path() 和 get\_path\_long() 有什么区别？

结构体agent\_local 中的planning\_in\_bounding\_box 和 in\_bounding\_box 有什么区别？一个表示规划的路径是否在边界框内，一个表示智能体是否在边界框内。

path\_final\_global 三维向量记录全局规划的终点；

* path\_index 三维向量列表list ，保存路径点集，全局使用，generate\_the\_global\_path();；

path\_global\_show\_message 全局路径massage消息类型；nav\_msgs::Path

fly\_in\_index 三维向量，表示什么？ 目标点。

update\_fly\_in\_index(bool replan)

finish\_exp\_flag[i] 记录什么？智能体是否完成边界框内的局部搜索

Dijkstra\_search\_fly\_in\_xy（）；路径规划

struct agent\_local

{

    bool in\_bounding\_box = false;

    bool planning\_in\_bounding\_box = false;

    Eigen::Vector3i position\_index;

    Eigen::Vector3i planning\_index;

    double time = 0;

    double state = 0;

    double priority = 0;

};

class mainbrain

Nbr\_point : 存储点云中的点point；

map\_set[] 和 global\_map[] 分别表示什么地图？

map\_set[]：几个地图区域，每个边界框；global\_map[]：全局地图；

now\_id 表示什么？ //第几个边界框

flag=false 获取路径成功；

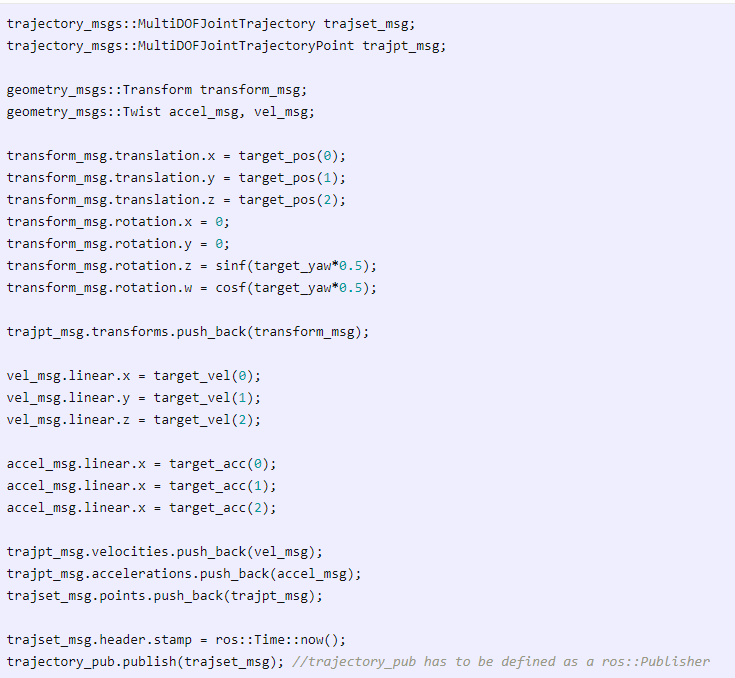
is\_transfer 是否到达目标。

State=0 到达目标点；1：搜索完边界框没有完成全局搜索；2：

not\_delete 记录什么？

gimbal表示什么？ 相机云台方向相关信息

无人机控制命令消息：官网中的说明；



trajectory\_msgs::MultiDOFJointTrajectory

这个的消息格式是

std\_msgs/Header header

string[] joint\_names

trajectory\_msgs/MultiDOFJointTrajectoryPoint[] points

trajectory\_msgs::MultiDOFJointTrajectoryPoint 消息包含以下字段：

transforms：这是一个 geometry\_msgs::Transform[] 类型的数组，表示多自由度关节在该时间点上的变换信息。通常，每个变换表示一个关节的位姿或姿态信息。

velocities：这是一个 geometry\_msgs::Twist[] 类型的数组，表示多自由度关节在该时间点上的速度信息。每个速度消息对应于一个关节的运动速度。

accelerations：这是一个 geometry\_msgs::Twist[] 类型的数组，表示多自由度关节在该时间点上的加速度信息。每个加速度消息对应于一个关节的运动加速度。

time\_from\_start：这是一个 ros::Duration 类型的字段，表示从轨迹的起始点到当前时间点的时间间隔。它用于确定轨迹上的时间点的相对时间。

"/nbr\_odom\_cloud" 和 "/cloud\_inW" 话题分别是什么的点云信息？

Lunch 文件

run.launch

启动run\_mbs.launch， 启动节点caric\_baseline/gcs\_task ，

以五个无人机的名称作为命名空间，分别启动节点 caric\_baseline / baseline\_planner .根据命名空间重映射通信话题。

使用ns属性的命名空间，不需要显式地在命名空间前面加上“/”，

run\_mbs.launch

初始化gazebo地图；启动5个spawn\_uav.launch 文件初始化5个无人机；

启动的节点 ： (pkg/ name)

caric\_mission/ mesh\_visualizer

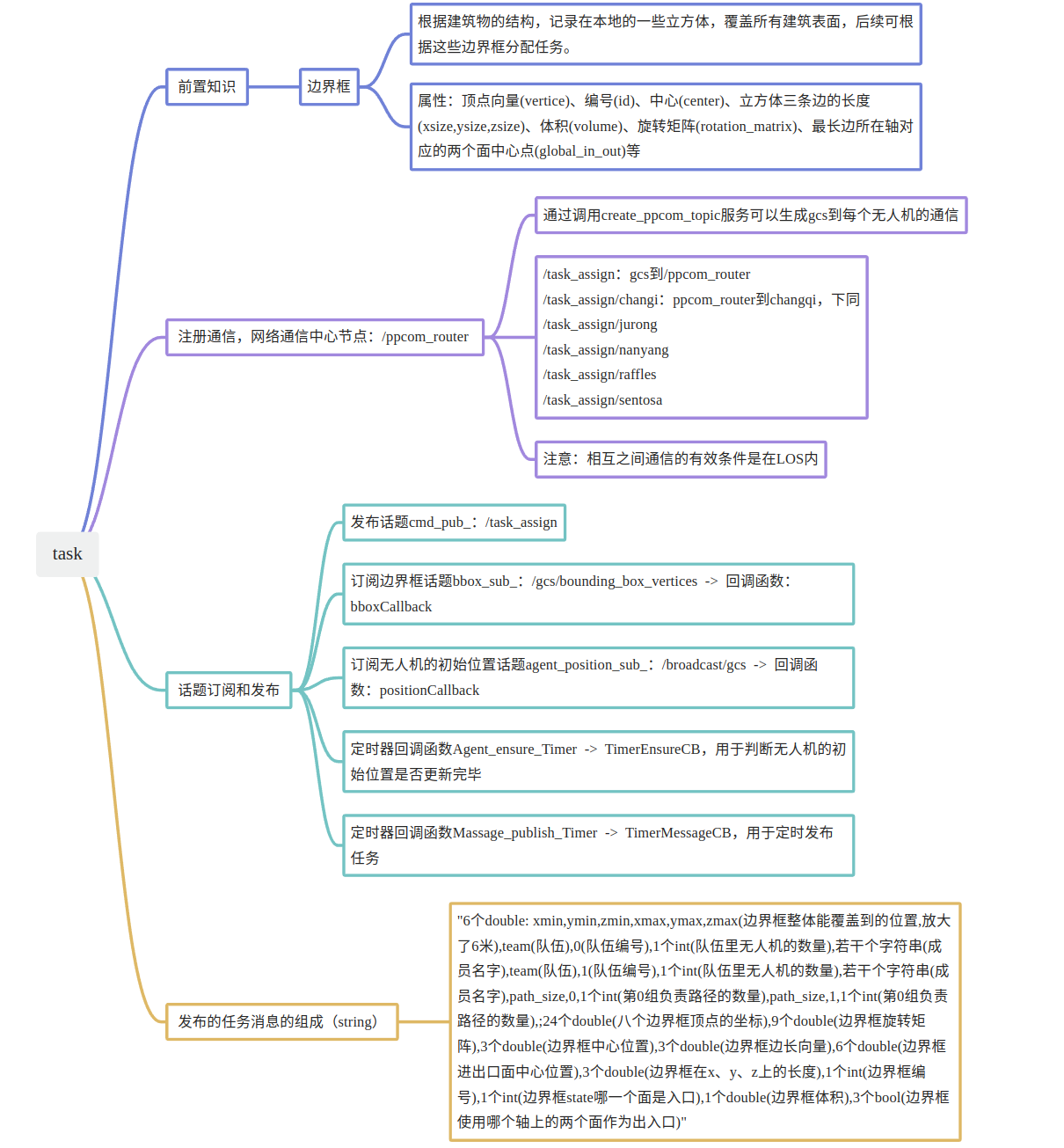
caric\_mission/ mission\_manager

caric\_mission/ ppcom\_router // 通信节点

tf2\_ros/ tf\_world\_to\_score\_report // 初始化地面站 spawn\_gcs.launch

Rviz/ rviz\_caric // 打开rviz可视化界面

path\_index 整数向量；



**Baseline\_planner.h 中四个class**

|  |  |
| --- | --- |
| class info\_agent | 用于处理保存无人机信息，包含无人机的分组、记录leader及其位置；  可用于更新无人机位置和路径信息。 |
| class grid\_map | 主要处理地图和智能体初始化及更新。根据边界框和栅格大小确定地图map，用三维向量嵌套表示。获取无人机分组，初始化每个组的follower；包含数据到字符串的转换，  定义一个**AStar类astar\_planner**，用于获取路径。  get\_path() & get\_path\_long() 区别：规划的路径长度节点数量不一样。 |
| class mainbrain | 解决信息处理问题，更新信息获取路径点更新地图，replan路径规划采用A\*局部规划；  定义一个**grid\_map类global\_map，用于存储全局地图，规划局部路径**，  定义一个**grid\_map型向量map\_set，用于存储局部地图（边界框划分）**；  定义一个**info\_agent类info\_mannager，获取无人机分组信息**（领导者名称及全局位置、状态  通信字符串格式输入：str  (第一个字符串topic：position/state/state\_set/map/mapglobal/visit/flyin)  每一条信息的标准形式：  position;智能体的位置信息 local\_dict[name];  state;智能体名称;state;  state\_set;组别信息;智能体名称;state;  map;组别信息;地图所占栅格数量;所有地图栅格坐标;;;;  mapglobal;组别信息;地图所占栅格数量;所有栅格坐标;;;;  visit;组别信息;目标点; |
| class Agent | 信息处理、控制，启动通信。初始化一个mainbrain类mm，处理接受到的各个无人机的msg消息，并发布本机相应的msg消息，获取控制命令发布，获取可视化地图和路径并发布。  四个定时器回调，分别处理：   1. 发布当前无人机的全局位置 2. 定时规划路径 3. 发布控制指令 4. 发布可视化信息（地图、路径）   订阅两个点云消息和一个地面实况里程计消息同步处理，更新地图map。 |