**示例程序load\_pcd\_map使用说明**

为了使参赛成员能更快的使用JMapDataEngine SDK进行开发，特此提供load\_pcd\_map示例程序，该程序封装为ros kinetic的一个package,所以要运行该程序需要在ubuntu16.04 ros kinetic 环境下运行，并且把它放到一个工作空间下。

**一．程序组成：**

load\_pcd\_map 包括两个node:map\_engine\_node,load\_pcd\_map\_node；其中，map\_engine\_node负责调用JMapDataEngine SDK下载地图，然后将地图的路径通过topic:“map\_list”广播出来；load\_pcd\_map\_node通过订阅“map\_list”，加载点云地图，然后将加载的点云地图以“pointcloud\_map” topic广播出来（广播点云地图只是为了可视化显示，实际上使用时可以不用广播）。

正常情况下，参赛者只需要同时运行以上两个node，便可以得到点云地图，如果对实现细节感兴趣，可以具体研究程序。

1. **将程序运行起来：**
2. source 程序所在工作空间。
3. roslaunch load\_pcd\_map load\_pcd\_map.launch
4. **细节数据说明：**
5. 点云地图的坐标原点可以从load\_pcd\_map\_node.cpp MapListCb中的origin\_x,origin\_y得到。
6. 从load\_pcd\_map\_node.cpp MapListCb()函数中得到的cloud\_map指针，即为点云地图所指向的指针，类型为pcl::PointCloud<PointXYZIL>::Ptr。其中PointXYZIL为自定义类型的点格式。

其中：（1）XYZ表示点的空间位置，将该位置分别加上坐标原点origin\_x，origin\_y即可将该点恢复到utm坐标系下。

1. I通道存储intensity信息；
2. L通道存储label信息，为了方便参赛人员使用，将点云地图中的点分为两类，如果L为0则表示为地面点，L为1则表示非地面点。