

API开发参考

API接口参考

概览

- 文件

文件名	描述
npuice_api.cpp	——
npuice_data.cpp	——
npuice_exception.cpp	——
npuice_geometry.cpp	——
npuice_map.cpp	——
npuice_param.cpp	——

- 类列表

类名	描述
Acc2D	二维加速度
Acc3D	三维加速度
BatteryParam	Sensor > Battery
BatteryStatus	电池状态
BumperArray	碰撞开关数组
BumperData	碰撞开关数据
BumperParam	碰撞开关参数

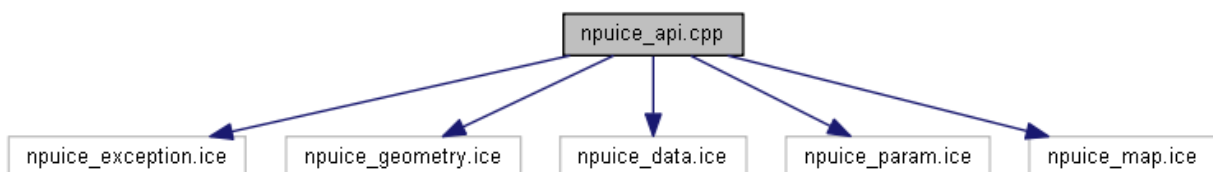
CellMat	栅格矩阵
ChassisParam	底盘参数
FootprintParam	足迹参数
GpsData	GPS数据
GpsParam	Sensor>Gps
ImgInfrdScan	图像红外参数
ImgLidarScan	图像雷达参数
ImgMap	图像地图
ImgPath	图像路径
ImgPoint	像素坐标
ImgPose	图像位姿
ImgSonarScan	图像声呐参数
ImgStation	图像站点
ImuData	Imu数据
ImuParam	Sensor/imu
InfrdParam	Sensor>infrd
InfrdScan	红外参数
LidarParam	雷达参数
LidarScan	雷达参数
Map2D	二维地图信息
MapInfo	地图信息
MotorEnc	电机编码器
MotorParam	电机参数
MotorSpd	电机转速
NaviParam	导航参数
NpuException	NPU异常

NpuParam	NPU参数
Path	路径
PathInfo	路径信息
PidParam	PID参数
PixelMat	像素矩阵
Point3D	三维坐标
Pose3D	三维位姿
Quaternion	四元数
SensorParam	Sensor>all
SlamParam	SLAM参数
SonarParam	Sensor>sonar
SonarScan	声呐参数
Station	站点
StationInfo	站点信息
SystemDiagInfo	系统诊断信息
Vector3D	三维向量
Vel2D	二维速度
Vel3D	三维速度

文件说明

· npuice_api.cpp

- npuice_api.cpp 文件参考的引用关系图



枚举

```
enum ServerState { UNKNOWN, CONNECTED, TIMEOUT, DISCONNECTED, MEM_EXCEED }
```

服务器状态

枚举值	描述
UNKNOWN	未知状态
CONNECTED	连接成功
TIMEOUT	连接超时
DISCONNECTED	无法连接
MEM_EXCEED	不能匹配版本

```
enum ManualCmd { MOVE_FWD, MOVE_BCK, TURN_LFT, TURN_RGT, MOVE_FWD_LFT, MOVE_FWD_RGT, STOP_MOVE }
```

遥控命令

枚举值	描述
MOVE_FWD	前进
MOVE_BCK	后退
TURN_LFT	左转
TURN_RGT	右转
MOVE_FWD_LFT	左前
MOVE_FWD_RGT	右前
STOP_MOVE	停止移动

```
enum NaviMode { P2P_NAVI, PF_NAVI }
```

导航模式

枚举值	描述
-----	----

P2P_NAVI	点对点导航
PF_NAVI	循迹导航

```
enum NaviState { PENDING, ACTIVE, PREEMPTED, SUCCEEDED, ABORTED, REJECTED, PREEMPTING, RECALLING, RECALLED, LOST, IDLE }
```

导航状态

枚举值	描述
PENDING	等待
ACTIVE	活跃的
PREEMPTED	抢占
SUCCEEDED	成功
ABORTED	中止
REJECTED	拒绝
PREEMPTING	抢占
RECALLING	回顾
RECALLED	回调
LOST	迷路
IDLE	闲置

```
enum CcpMode { ZIGZAG_CCP, SPIRAL_CCP }
```

区域覆盖模式

枚举值	描述
ZIGZAG_CCP	Z字形
SPIRAL_CCP	螺旋形

```
enum SlamMode { PF_SLAM, ICP_SLAM }
```

Slam模式

枚举值	描述
PF_SLAM	粒子滤波算法
ICP_SLAM	图优化算法

函数说明

```
void Connect (string version)
```

连接npu

参数

参数名称	描述
version	npu版本号

```
ServerState GetServerState ()
```

获取NPU状态

```
SystemDiagInfo GetSystemDiagInfo ()
```

获取系统诊断信息

```
void Shutdown ()
```

关闭NPU电源

```
void Reboot ()
```

重启NPU

```
StringArray GetParamIdArray ()
```

获取参数ID列表

```
void SetParamId (string id)
```

设置参数ID

参数

参数名称	描述
id	底盘ID

```
NpuParam GetNpuParam ()
```

获取NPU参数

```
void SetNpuParam (NpuParam param)
```

设置NPU参数

参数

参数名称	描述
NPUParam	电机参数

```
MotorParam GetMotorParam ()
```

获取电机参数

```
void SetMotorParam (MotorParam param)
```

设置电机参数

参数

参数名称	描述
MotorParam	NPU参数

```
PidParam GetPidParam ()
```

获取PID参数

```
void SetPidParam (PidParam param)
```

设置PID参数

参数

参数名称	描述
PidParam	PID参数

```
SensorParam GetSensorParam ()
```

获取传感器参数

```
void SetSensorParam (SensorParam param)
```

设置传感器参数

参数

参数名称	描述
SensorParam	传感器参数

```
ChassisParam GetChassisParam ()
```

获取底盘参数

```
void SetChassisParam (ChassisParam param)
```

设置底盘参数

参数

参数名称	描述
ChassisParam	底盘参数

```
NaviParam GetNaviParam ()
```

获取导航参数


```
void SetNaviParam (NaviParam param)
```

设置导航参数

参数

参数名称	描述
NaviParam	导航参数

```
SlamParam GetSlamParam ()
```

获取SLAM参数

```
void SetSlamParam (SlamParam param)
```

设置SLAM参数

参数

参数名称	描述
SlamParam	SLAM参数

```
void FeedMotorEnc (MotorEnc enc)
```

反馈电机编码器数据

参数

参数名称	描述
MotorEnc	电机编码器数据

```
void FeedActMotorSpd (MotorSpd spd)
```

反馈电机实际转速

参数

参数名称	描述
MotorSpd	电机实际转速

```
MotorEnc GetMotorEnc ()
```

获取电机编码器数据

```
void ClearMotorEnc ()
```

清除电机编码器数据

```
MotorSpd GetActMotorSpd ()
```

获取电机实际转速

```
MotorSpd GetCmdMotorSpd ()
```

设置电机转速

```
LidarScan GetLidarScan ()
```

获取雷达参数

```
ImgLidarScan GetImgLidarScan ()
```

获取雷达图像参数

```
ImuData GetImuData ()
```

获取IMU数据

```
SonarScan GetSonarScan ()
```

获取超声波传感器参数

```
ImgSonarScan GetImgSonarScan ()
```

获取超声波传感器图像参数

```
InfrdScan GetInfrdScan ()
```

获取红外传感器参数

```
ImgInfrdScan GetImgInfrdScan ()
```

获取红外传感器图像参数

```
BumperArray GetBumperArray ()
```

获取碰撞数据列表

```
BatteryStatus GetBatteryStatus ()
```

获取电池状态信息

```
void EnableManualMode (bool enb)
```

是否启动遥控模式

```
void ManualControl (ManualCmd cmd)
```

遥控控制

参数

参数名称	描述
ManualCmd	遥控命令

```
void SetManualVel (Vel3D vel, Acc3D acc)
```

设置遥控速度

参数

参数名称	描述
Vel3D	遥控速度
Acc3D	遥控加速度

```
Pose3D GetCurrentPose ()
```

获取当前位姿

```
ImgPose GetImgPose ()
```

获取当前位姿图像

```
Vel3D GetCurrentVel ()
```

获取当前速度

```
Acc3D GetCurrentAcc ()
```

获取当前加速度

```
MapInfoList GetMapInfos ()
```

获取当前地图信息

```
void SetMapInfos (MapInfoList list)
```

设置地图信息

参数

参数名称	描述
MapInfoList	地图信息列表

```
Map2D GetCurrentMap ()
```

获取当前地图

```
ImgMap GetCurrentImgMap ()
```

获取当前地图图像

```
StationList GetStations (string map_id)
```

获取站点信息

参数

参数名称	描述
map_id	地图名称

```
void SetStations (string map_id, StationList list)
```

设置站点信息

参数

参数名称	描述
StationList	站点信息列表

```
ImgStationList GetImgStations (string map_id)
```

获取站点图像信息

```
void SetImgStations (string map_id, ImgStationList list)
```

设置站点图像信息

参数

参数名称	描述
StationList	站点信息列表

```
PathList GetPaths (string map_id)
```

获取路径信息

```
void SetPaths (string map_id, PathList list)
```

设置路径信息

参数

参数名称	描述
PathList	路径信息列表

```
ImgPathList GetImgPaths (string map_id)
```

获取路径图像

```
void SetImgPaths (string map_id, ImgPathList list)
```

设置路径图像

参数

参数名称	描述
ImgPathList	路径图像信息列表

```
void LoadMap (string id)
```

加载地图

```
void SetInitPose (Pose3D pose)
```

设置初始位姿

参数

参数名称	描述
Pose3D	位置及姿态

```
void SetInitImgPose (ImgPose pose)
```

设置图像上的初始位姿

参数

参数名称	描述
------	----

```
void StartNavi (NaviMode mode)
```

开始导航

参数

参数名称	描述
NaviMode	导航模式

```
void StopNavi ()
```

停止导航

```
void PauseTask ()
```

暂停任务

```
void ContinueTask ()
```

继续任务

```
void CancelTask ()
```

取消任务

```
float GetTaskProgress ()
```

反馈任务进度

```
NaviState GetNaviState ()
```

获取导航状态

```
Path GetCurrentPath ()
```

获取当前路径

```
ImgPath GetCurrentImgPath ()
```

获取当前路径图像

```
void GotoGoal (Pose3D goal)
```

目标点导航

```
void GotoImgGoal (ImgPose goal)
```

定位图像图标

```
void GotoStation (string map_id, string station_id)
```

定位站点

```
void FollowTempPath (Pose3DList poses)
```

跟踪临时路径

```
void FollowTempImgPath (ImgPoseList poses)
```

跟踪临时路径图像

```
void FollowPath (string map_id, string path_id)
```

跟踪已保存路径

```
Pose3DList PlanCoveragePath (Point3DList vertices)
```

区域覆盖路径

```
ImgPoseList PlanCoverageImgPath (ImgPointList vertices)
```

区域覆盖路径图像


```
void StartSlam (SlamMode mode)
```

开始SLAM建图

```
void StopSlam (string map_id)
```

停止SLAM建图

```
ZipFile ExportParamFile (string id)
```

导出参数文件

```
void ImportParamFile (ZipFile file)
```

导入参数文件

参数

参数名称	描述
ZipFile	Zip文件

```
ZipFile ExportMapFile (string id)
```

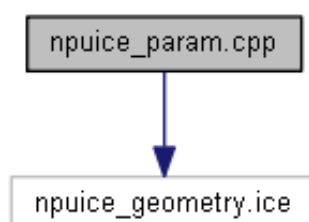
导出地图文件

```
void ImportMapFile (ZipFile file)
```

导入地图文件

· npuice_param.cpp

- npuice_param.cpp文件参考的引用关系图



枚举

```
enum BrakeType { SOFT_BRK, HARD_BRK }
```

制动类型

枚举值	描述
SOFTBRK	软制动
HARD_BRK	硬制动

```
enum ValidOutputLevel { LOW_VALID, HIGH_VALID }
```

有效输出电平

枚举值	描述
LOW_VALID	低电平有效
HIGH_VALID	高电平有效

```
enum LidarType { RPLIDAR, RSLIDAR, LSLIDAR, FSLIDAR, SICK_TIM551, SICK_TIM561, SICK_TIM571, SICK_LMS111, SICK_LMS151, SICK_LMS511, UST_10LX, UTM_30LX, PF_R2000_20HZ, PF_R2000_50HZ }
```

雷达类型

枚举值	描述
RPLIDAR	思岚RPLIDAR
RSLIDAR	速腾RSLIDAR
LSLIDAR	镭神LSLIDAR
FSLIDAR	砵石FSLIDAR
SICK_TIM551	西克SICK_TIM551
SICK_TIM561	西克SICK_TIM561
SICK_TIM571	西克SICK_TIM571
SICK_LMS111	西克SICK_LMS111

SICK_LMS151	西克SICK_LMS151
SICK_LMS511	西克SICK_LMS511
UST_10LX	北阳UST_10LX
UTM_30LX	北阳UTM_30LX
PF_R2000_20HZ	倍加福PF_R2000_20HZ
PF_R2000_50HZ	倍加福PF_R2000_50HZ

```
enum InterfaceType { ETHERNET, SERIAL }
```

接口类型

枚举值	描述
ETHERNET	以太网
SERIAL	串口

```
enum BumperLocation { FRONT BUMPER, FRONT_LEFT BUMPER, FRONT_RIGHT BUMPER, BACK BUMPER, BACK_LEFT BUMPER, BACK_RIGHT BUMPER, LEFT BUMPER, RIGHT BUMPER }
```

碰撞开关位置

枚举值	描述
FRONT BUMPER	前碰撞开关
FRONT_LEFT BUMPER	左前碰撞开关
FRONT_RIGHT BUMPER	右前碰撞开关
BACK BUMPER	后碰撞开关
BACK_LEFT BUMPER	左后碰撞开关
BACK_RIGHT BUMPER	右后碰撞开关
LEFT BUMPER	左碰撞开关
RIGHT BUMPER	右碰撞开关

```
enum ModelType { CARLIKE, DIFFDRV, UNIWHEEL, OMNIWHEEL }
```

底盘类型

枚举值	描述
CARLIKE	车型底盘
DIFFDRV	差速底盘
UNIWHEEL	四全向轮
OMNIWHEEL	三全向轮

```
enum SteerEncLocation { CENTOR, LEFT, RIGHT }
```

车型底盘编码器位置

枚举值	描述
CENTOR	中心
LEFT	左边
RIGHT	右边

```
enum ShapeType { ROUND, RECTANGLE, POLYGON }
```

底盘形状类型

枚举值	描述
ROUND	圆
RECTANGLE	矩形
POLYGON	多边形

参考样例程序（安卓）

NPU初始化

· 实例化

```
NpuIceI mynpu = new NpuIceI();  
  
WizRoboNpuUdp udpSendIp = new WizRoboNpuUdp();
```

在工程中导入ICE.jar和NPU.jar包，进行实例化。

· 获取NPU IP地址

```
public String npuIp;  
  
npuIp=udpSendIp.GetIp();
```

循环调用，直至获取到NPU的IP地址。

· 连接NPU

```
public bool npuIsReady=false;  
  
npuIsReady=mynpu.NpuInit(npuIp);
```

返回true则表示连接成功，该状态位可供其他功能判断使用。

SLAM建图

· 建图模式

```
mynpu.StartSlam(SlamMode.ICP_SLAM);  
  
mynpu.StartSlam(SlamMode.PF_SLAM);
```

地图构建模式包含ICP_SLAM模式和PF_SLAM模式，推荐使用ICP_SLAM模式。

· 获取地图

```
ImgMap imgMap = mynpu.GetCurrentImgMap();
```

建图模式下，需循环调用，建议刷新闻隔时间为2s。

· 获取机器人实时位置

```
ImgPose actImgPose = mynpu.GetCurrentImgPose();
```

建图模式下，需循环调用，建议刷新闻隔时间为2s。

· 获取雷达数据

```
ImgLidarScan lidarScanData = mynpu.GetImgLidarScan();
```

对数据量较大的雷达，建议需要查看时调用即可，不需要循环调用以免造成网络阻塞。

· 保存地图

```
String mapName="xxx"; //xxx为地图名称  
  
mynpu.StopSlam(mapName);
```

若mapName为空，则NPU不保存该地图；
mapName里不能包含@ ¥ % & * #等特殊字符；
地图保存后，可在地图列表中进行查看。

地图信息

· 获取地图列表

```
MapInfo[] mapInfoList=mynpu.GetMapInfos();
```

列表中包含地图名称，数据，创建时间等相关信息。

· 删除地图

```
MapInfo[] mapInfoList=mynpu.GetMapInfos();  
  
mynpu.SetMapInfos(newMapInfoList);
```

对地图列表进行编辑，并调用函数获取新的地图列表。

智能导航

· 选择地图并导入

```
mynpu.SelectMap(mapName);  
  
ImgMap imgMap = mynpu.GetCurrentImgMap();
```

从地图列表中选择需要导入的地图名称并完成地图导入。

· 启动导航

```
mynpu.StartNavi(NaviMode.P2P_NAVI); //启动点对点自由导航  
  
mynpu.StartNavi(NaviMode.PF_NAVI); //启动循迹模式导航
```

· 获取机器人实时位置

```
ImgPose actImgPose = mynpu.GetCurrentImgPose();
```

循环调用该语句可实时获取机器人当前位置，建议刷新闻隔时间为2s。

· 获取雷达数据

```
ImgLidarScan lidarScanData = mynpu.GetImgLidarScan();
```

· 设置初始位置

```
ImgPose initPose=new ImgPose(init_img_pose_u,init_img_pose_v,init_i  
mg_pose_theta);  
  
mynpu.SetInitImgPose(initPose);
```

· 设置目标点 / 设置路径

```
ImgPose goalPose=new ImgPose(set_goal_pose_u,set_goal_pose_v,set_goal_pose_theta);

mynpu.GotoImgGoal(goalPose); //点对点自由导航

mynpu.FollowTempImgPath(ImgPose[] imgPose); //轨迹跟踪导航

mynpu.PlanCoverageImgPath(ImgPoint[] imgPoint); //区域覆盖导航
```

智能导航包含点对点自由导航模式及循迹导航（轨迹跟踪/区域覆盖）模式，通过调用不同的接口可实现不同的导航模式。

· 获取移动路径

```
ImgPath imgPath = mynpu.GetCurrentImgPath();
```

循环调用该语句获取并不断刷新移动路径。

· 暂停与继续

```
mynpu.PauseTask(); //暂停执行任务

mynpu.ContinueTask(); //继续执行任务
```

· 停止导航

```
mynpu.StopNavi();
```

路径信息

· 获取路径列表

```
ImgPath[] imgPathList=mynpu.GetImgPaths(mapName);
```

路径信息保存在地图中，通过mapName选择地图后方可获取路径列表；路径列表中包含路径名称，数据及创建时间等信息。

· 删除路径 / 添加路径

```
ImgPath[] imgPathList=mynpu.GetImgPaths();  
  
mynpu.SetImgPaths(mapName, newImgPathList);
```

对路径列表进行编辑（删除或添加），并调用函数获取新的路径列表。

· 路径跟随

```
mynpu.FollowPath(mapName, pathName);
```

在循迹导航模式下，选择路径名称后可直接进行路径跟随导航。

站点信息

· 获取站点列表

```
ImgStation[] imgStationList=mynpu.GetImgStations(mapName);
```

站点信息保存在地图中，通过mapName选择地图后方可获取站点列表；站点列表中包含站点名称，数据及创建时间等信息。

· 删除站点 / 添加站点

```
ImgStation[] imgStationList=mynpu.GetImgStations();  
  
mynpu.SetImgStations(mapName, newImgStationList);
```

对站点列表进行编辑（删除或添加），并调用函数获取新的站点列表。

· 站点导航

```
mynpu.GotoStation(mapName, stationName);
```

在自由导航模式下，选择站点名称后可直接运动至该站点。

系统设置

· 重启

```
mynpu.Reboot();
```

· 关机

```
mynpu.Shutdown();
```

遥控

```
mynpu.SetManualCmd(ManualCmd manualcmd);
```