

一. 类的介绍

- Vehicle 类

- 成员函数:

```
方法名: obtainCtrlAuthority();  
参数: 无  
功能: 使 vehicle 对象获得飞机的控制权限  
返回值: 无
```

- UWBPosition 类

- 成员变量:

```
int s; // socket id  
float x; // 当前的 x 坐标  
float y; // 当前的 y 坐标
```

- 成员函数:

```
方法名: UWBPosition();  
参数: 无  
功能: 该类的构造函数, 和 UWB 服务器建立 Socket 通信  
返回值: 无
```

```
方法名: UWBPosition();  
参数: 无  
功能: 该类的构造函数, 和 UWB 服务器建立 Socket 通信  
返回值: 无
```

```
方法名: Point getCurPosition();  
参数: 无  
功能: 获得当前基于 UWB 坐标系的位置信息  
返回值: Point 结构体
```

- TelemetryBroadcast 类

- 成员变量:

```
UWBPosition uwb;  
Vehicle* vehicle; // Vehicle 对象  
Telemetry::Status status; // 飞机当前飞行模式  
Telemetry::GlobalPosition globalPosition; // 飞机当前绝对位置坐标  
Telemetry::RC rc; // 当前接收到的遥控指令  
Telemetry::Vector3f velocity; // 飞机当前速度信息  
Telemetry::Quaternion quaternion; // 飞机当前四元素信息
```

■ 成员函数:

方法名: `TelemetryBroadcast(Vehicle* vehicle, UWBPosition uwb)`

参数: `Vehicle` 对象, `UWBPosition` 对象

功能: 该类的构造函数

返回值: 无

方法名: `void teleBroadcastPrint();`

参数: 无

功能: 持续获得当前飞机的传感器信息并输出至屏幕

返回值: 无

方法名: `void getDataFromBroadcast();`

参数: 无

功能: 更新当前飞机的传感器信息

返回值: 无

方法名: `void teleBroadcastSave2File()`

参数: 无

功能: 持续获得当前飞机的传感器信息并保存成文件

● `VehicleControl` 类

■ 成员变量:

`Vehicle* vehicle;` // `Vehicle` 类对象

`UWBPosition uwbPos;` // `UWBPosition` 类对象

■ 成员函数:

方法名: `VehicleControl(Vehicle* vehicle, UWBPosition p);`

参数: `Vehicle` 对象, `UWBPosition` 对象

功能: 该类的构造函数

返回值: 无

方法名: `void attitudeControl(float roll, float pitch,
float z, float yaw)`

参数: `float roll` (翻滚角)

`float pitch` (俯仰角)

`float z` (相对高度)

`float yaw` (偏航角)

功能: 基于姿态控制飞机飞行

返回值: 无

方法名: `void velocityControl(float32_t Vx, float32_t Vy,
float32_t Vz, float32_t yawRate);`

参数: `float32_t Vx` (x 方向速度)

`float32_t Vy` (y 方向速度)

`float32_t Vz` (z 方向速度)

`float32_t yawRate` (偏航角角速率)

功能： 基于速度控制飞机飞行

返回值： 无

方法名： `bool moveByUWBPosOffset(float xOffsetDesired, float yOffsetDesired, float posThresholdInM);`

参数： `float xOffsetDesired` (x 方向飞行位移)

`float yOffsetDesired` (y 方向飞行位移)

`float posThresholdInM` (允许误差控制量)

功能： 基于 UWB 位置信息实现飞行距离控制

- M100 类

- 成员变量：

```
Vehicle* vehicle;    // Vehicle 类对象
UWBPosition* uwb;    // UWBPosition 类对象
TelemetryBroadcast* broadcast; // TelemetryBroadcast 类对象
VehicleControl* control // VehicleControl 类对象
```

- 成员函数：

方法名： `M100 (Vehicle* vehicle);`

参数： `Vehicle` 对象

功能： 该类的构造函数

返回值： 无

二. 例程序

```
#include "TelemetryBroadcast.hpp"
#include <dji_control.hpp>
#include "VehicleControl.hpp"
#include "UWBPosition.hpp"
#include "M100.hpp"
using namespace DJI::OSDK;
using namespace DJI::OSDK::Telemetry;

int main(int argc, char** argv)
{
    /*
    功能描述：
        基于 UWB+guidance 实现按照预设矩形轨迹飞行，飞行过程中不断获取传感器及
        UWB 信息并输出到文件
    */
```

```
// 初始化环境
LinuxSetup linuxEnvironment(argc, argv);
// 实例化 Vehicle 对象
Vehicle* vehicle = linuxEnvironment.getVehicle();
// 使得 vehicle 对象获得飞行器控制权限
vehicle->obtainCtrlAuthority();
// 实例化 M100 对象
M100 m100(vehicle);
// 实时获取 UWB 和传感器信息并输出至文件
m100.broadcast->teleBroadcastSave2File();
// 起飞
m100.control->takeoff();
// 睡眠 2s 等待起飞完成
usleep(2000000);
// 向前飞行 2 米
m100.control->moveByUWBPosOffset(2,0,0,1);
// 向右飞行 2 米
m100.control->moveByUWBPosOffset(0,2,0,1);
// 向后飞行 2 米
m100.control->moveByUWBPosOffset(-2,0,0,1);
// 向左飞行 2 米
m100.control->moveByUWBPosOffset(0,-2,0,1);
// 降落
m100.control->land();
}
```