

网络RTK模块流程

1.接收拿到rtk账号

账号一般包含 mHost, Port =, UserName , Password , MountPoint

2.判断飞机rtk模块是否就绪

目前可以通过两种方式判断：

```
//接收飞机上报的状态preflightCheckResultGcs
if (it.preflightCheckResultGcs.isEmpty() ||
!it.preflightCheckResultGcs.contains(FlightCheckException.RTK_EXCEPTION
)) {

//或者onMainServiceValid拿到回调
override fun onMainServiceValid(valid: Boolean, drone: IBaseDevice)
```

3.初始化设置网络状态，启用RTK定位

```
/**
 * 用户根据需要设置网络RTK类型，国内建议采用千寻SDK账号体系，否则网络RTK采用Ntrip方式实现,默认为Ntrip
 * @param isQianxun 是否为千寻原生网络RTK
 * @param mHost 网络RTK访问地址
 * @param mPort 网络RTK访问端口
 * @param callback 更新配置回调
 */
abstract fun updateNetRtkType(isQianxun: Boolean, mHost: String?, mPort: Int?, callback: ChangeRTKConfigCallback)

/**
 * 是否启用RTK定位
 * @param isEnabled true 启用, false关闭
 * @param callback 更新配置回调
 */
abstract fun enableRTKLocation(
    isEnabled: Boolean,callback: ChangeRTKConfigCallback
)
```

4.设置RTK信号类型

```
/**
 * 设置RTK信号类型
 * @param rtkSignalEnum
 *
 * 机巢RTK 1
 * SELF_NETWORK(1),
 *
 * 网络RTK, 2
```

```

    * NETWORK(2),
    *
    * 4G/5G移动网络 3
    * MOBILE_NETWORK_SERVICES(3);
    */
    abstract fun switchRTKSignalEnum(
        rtkSignalEnum: RTKSignalEnum, callback: ChangeRTKConfigCallback
    )

```

5.需要登录授权的账号

```

/**
 * 机巢网络RTK授权账号
 */
    abstract fun autherNetRtk(userName:String, password :String, mountPoint
:String?, callback: RTKAuthoCallback)

/**
 * 飞机4G/5G移动网络授权登录
 */
    abstract fun autherMobileServiceRtk(mHost: String, mPort:
Int,userName:String, password :String, mountPoint :String, callback:
RTKAuthoCallback,retryCount:Int?=5)

```

6.接收/发送差分包数据

```

ntripSocket.setNtripDataListener(new NtripSocketImpl.NtripDataListener() {
    @Override
    public void onData(byte[] data) {
        long mTime = System.currentTimeMillis();
        if (mTime - lastTime < 500) {
            return;
        }
        lastTime = mTime;
        if(callback!=null) {
            callback.encapsulatedDiffData(data);
        }
    }

    @Override
    public void connect() {
        requestConnectMountCmd(connectMountCmd);
    }

    @Override
    public void fail() {
        connectMountPointFail(CONNECT_HOST_FAIL);
        SDKLog.d(TAG, "connectMountPointFail:connect host failed ");
    }

});

```