

避障系统使用教程

1. 概述

避障系统，主要是检测飞机的飞行范围内是否有障碍物，发现障碍物后会启用刹停安全保护机制。当前避障系统支持6向全面避障，主要分为前向、后向、上向、下向、左向、右向等方向避障。

2. 避障接口设置获取

避障开关设置

主要分为**水平避障开关**、**上方避障开关**、**下方避障开关**等三组开关，支持设置和获取。

水平避障开关：VisionKey.KeyHorizontalObstacleAvoidance

上方避障开关：VisionKey.KeyTopObstacleAvoidance

下方避障开关：VisionKey.KeyBottomObstacleAvoidance

以水平避障开关设置为例。

```
val key = KeyTools.createKey(VisionKey.KeyHorizontalObstacleAvoidance)
getManager()?.setValue(key, open, object : CommonCallbacks.CompletionCallback {
    override fun onSuccess() {

    }

    override fun onFailure(code: IAutelCode, msg: String?) {

    }
})
```

避障告警距离

主要分为**水平避障告警距离**、**上方避障告警距离**、**下方避障告警距离**等三组告警距离，支持设置和获取。

水平避障告警距离：VisionKey.KeyHorizontalWarningDistance

上方避障告警距离：VisionKey.KeyTopWarningDistance

下方避障告警距离：VisionKey.KeyBottomWarningDistance

以水平避障告警距离设置为例。

```

val key = KeyTools.createKey(VisionKey.KeyHorizontalWarningDistance)
getKeyManager()?.setValue(key, warningDistance, object :
CommonCallbacks.CompletionCallback {
    override fun onSuccess() {

    }

    override fun onFailure(code: IAutoCode, msg: String?) {

    }
})

```

避障刹车距离

主要分为**水平避障刹车距离**、**上方避障刹车距离**、**下方避障刹车距离**等三组刹车距离，支持设置和获取。

水平避障刹车距离：VisionKey.KeyHorizontalBrakeDistance

上方避障刹车距离：VisionKey.KeyTopBrakeDistance

下方避障刹车距离：VisionKey.KeyBottomBrakeDistance

以水平避障刹车距离设置为例。

```

val key = KeyTools.createKey(VisionKey.KeyHorizontalBrakeDistance)
getKeyManager()?.setValue(key, brakeDistance, object :
CommonCallbacks.CompletionCallback {
    override fun onSuccess() {

    }

    override fun onFailure(code: IAutoCode, msg: String?) {

    }
})

```

3. 视觉雷达图上报

飞机飞行过程中，会实时上报当前的避障距离信息列表。列表中包含了6向的避障距离信息。可以根据某方向的实时避障距离和避障告警距离、刹车距离来比较，然后实时提示用户当前的飞行安全状态。

```

val keyWarning = KeyTools.createKey(VisionKey.KeyReportEmergency)
getKeyManager()?.listen(keyWarning, object :
CommonCallbacks.KeyListener<List<VisionRadarInfoBean>> {
    override fun onValueChange(oldValue: List<VisionRadarInfoBean>?, newValue:
List<VisionRadarInfoBean>) {

    }
})

```

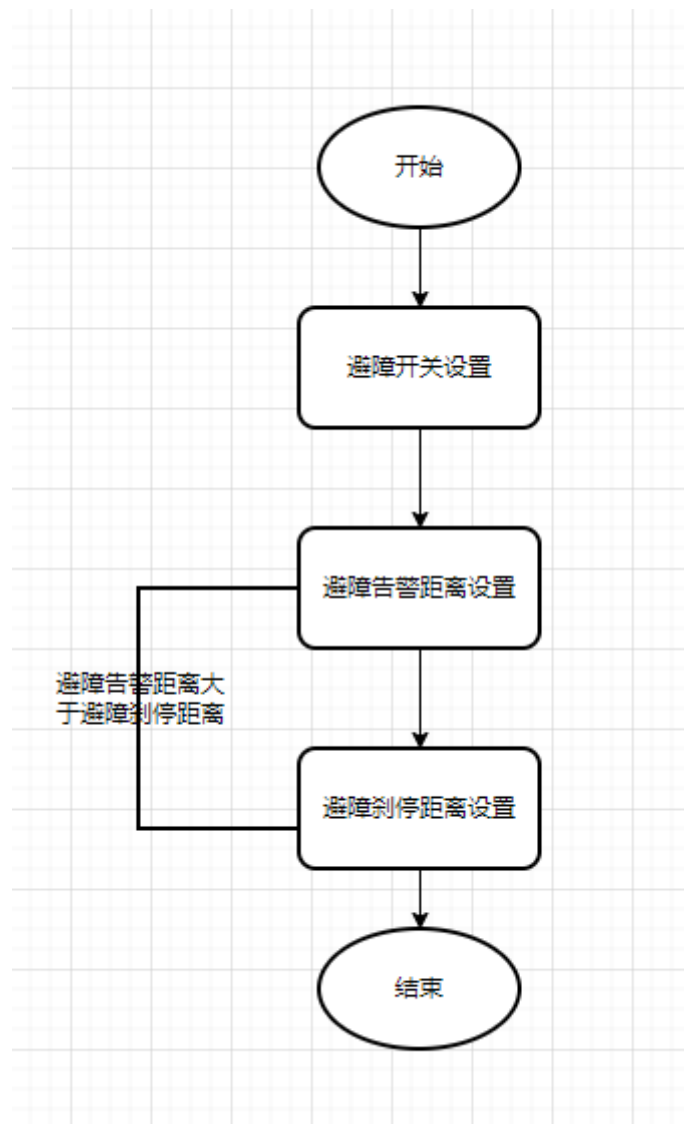
视觉雷达信息

```
data class VisionRadarInfoBean(  
    var timeStamp: Long = 0, //时间戳  
    var position: VisionSensorPositionEnum = VisionSensorPositionEnum.FRONT, //传  
    感器位置  
    var distances: List<Float>? = null //传感器与障碍物的多个距离  
)
```

视觉传感器位置

```
enum class VisionSensorPositionEnum(var value: Int) {  
    FRONT(0),          //前  
    REAR(1),           //后  
    BOTTOM(2),         //下  
    RIGHT(3),          //右  
    LEFT(4),           //左  
    TOP(5),            //上  
}
```

4. 接口调用顺序



5. 视觉雷达图上报示例

```
val keywarning = KeyTools.createKey(VisionKey.KeyReportEmergency)
getKeyManager()?.listen(keywarning, object :
CommonCallbacks.KeyListener<List<VisionRadarInfoBean>> {
    override fun onValueChange(oldValue: List<VisionRadarInfoBean>?, newValue:
List<VisionRadarInfoBean>) {
        for (value in newValue) {
            if (value.position == VisionSensorPositionEnum.TOP) {
                val distanceList = value.distances
                if (distanceList != null) {
                    if (distanceList.min() < getTopWarningDistance()) {
                        ToastUtils.showToast("上方存在障碍物，请谨慎飞行")
                    }
                }
            }
        }
    }
})
```