

为帮助开发者快速调试无人机功能，Debug Tools提供对无人机硬件模块的参数访问和行为控制的接口调试工具。

包括：

- Key/Value：无人机支持的所有Key的调试工具
- Media Files：任务文件的上传、下载、查询、删除；视频文件的获取、删除、下载；以及健康检测
- Multi-Video Decoding：多视频解码
- NetMesh_DeviceList：组网完成后所有在线设备（无人机、遥控器等）的状态信息的数据上报
- LiveStreaming：视频实时码流播放
- Interaction-Center：
- Scenario Testing：遥控器配对；遥控器、IMU、指南针、云台校准；照片、视频录制
- FLIGHT_PARAMS：所有飞行参数设置、获取的调试工具
- VISION_PARAMS：前后、上下视觉避障异常信息展示；录制各方向视觉避障数据
- VISION_PARAMS1：
- VISION_ORIGINAL：
- VISION_TRACK：
- SKY_LINK：
- REPORT_CONTROL：
- SWITCH_SETTING：
- SYSTEM_UPDATE：
- IMAGE_TRANSMISSION：
- UPLOAD_DOWNLOAD_TEST：
- VIDEO_DECODE：

1. Key/Value

概述

KeyManager类提供了一组方法来访问硬件模块的参数和控制硬件模块的行为。通过KeyTools类提供的createKey方法创建AutelKey实例。然后通过KeyManager发起Key的Set、Get、Action、Listen或Report，来完成对无人机硬件模块的控制。

通过Key/Value调试模块，可以快速的对无人机硬件模块的参数和行为进行设置和读取。

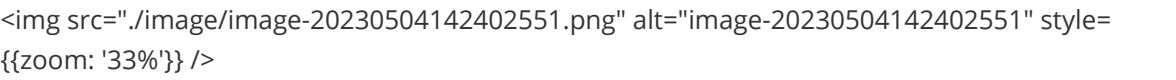
调试说明

下图为进入Key/Value调试页面，模块默认选择为Camera的截图：

1. 右侧**Camera**为当前选择的调试模块，可以点击切换其他模块。
2. 左侧**Key List**为当前选择的模块支持的所有Key的列表，可以翻页查找，也可以直接搜索查找。

目前支持调试的业务模块为：Camera、FLGHT_MISSION、FLIGHT CONTROL、BATTERY、GIMBAL、AIRLINK、CHANNEL_TYPE_NETMESH、AI_SERVICE、FLIGHT PARAMS、VISION、REMOTE CONTROLLER、COMMON UPLOAD、AI_TRACK、REMOTE_ID、NEST、SystemManager、MISSION MANAGER、ACCESSORIES PROXY、Autonomy。

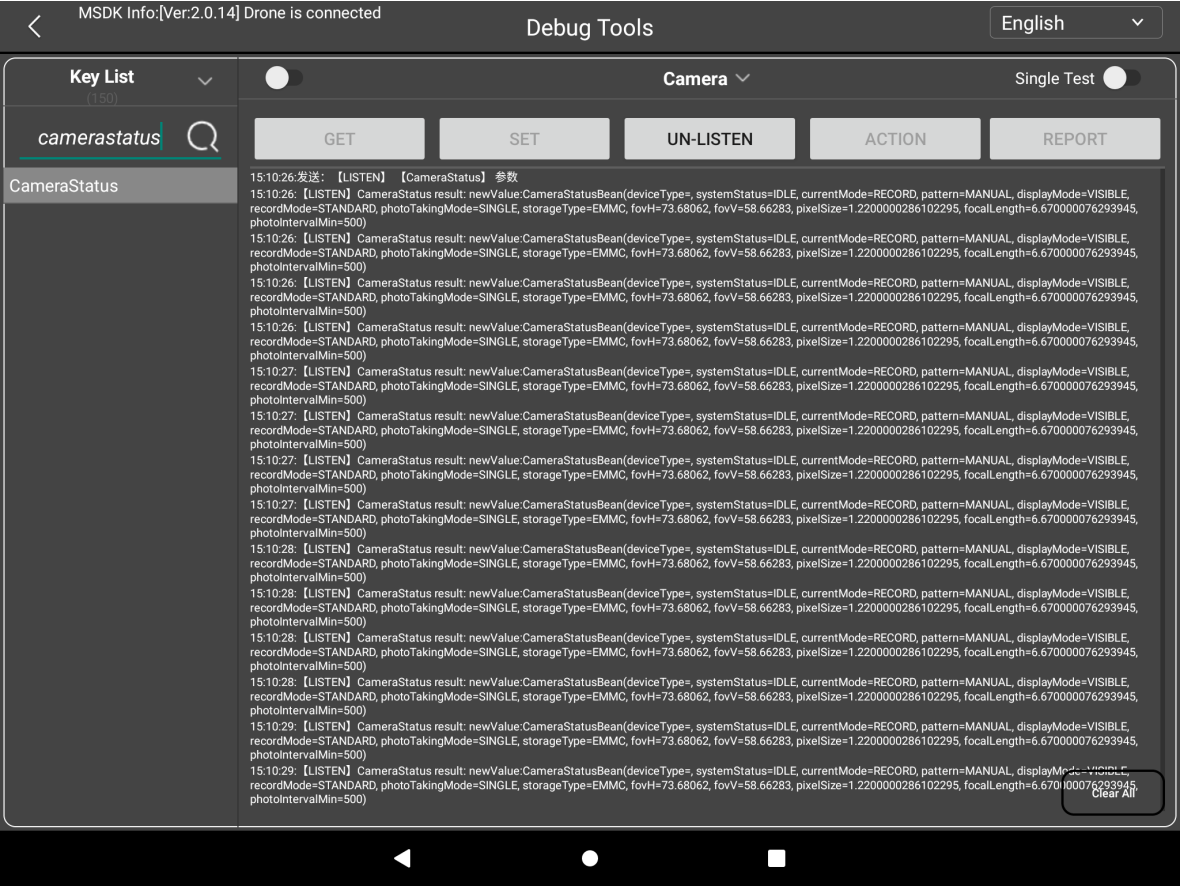
下图为点击Camera，进行模块切换的截图：



Listen方法

例如：获取相机状态，在Key List中查找并选中CameraStatus，由于CameraStatus只支持监听，右侧的Get、Set、Action、Report会灰化不可点，点击按钮Listen，下方就会展示CameraStatus的参数打印。同时Listen按钮会变成UnListen，点击UnListen可取消监听，停止参数打印。

下图为Listen成功，打印数据的截图：



调用方法示例：

```

fun addCameraStatusListen() {
    val keyCameraStatus: AutelKey<CameraStatusBean> =
        KeyTools.createKey(CameraKey.KeyCameraStatus)

    DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.listen(
        keyCameraStatus, cameraStatusCallback)
}

private val cameraStatusCallback = object :
    CommonCallbacks.KeyListener<CameraStatusBean> {
        override fun onValueChange(oldValue: CameraStatusBean?, newValue:
            CameraStatusBean) {

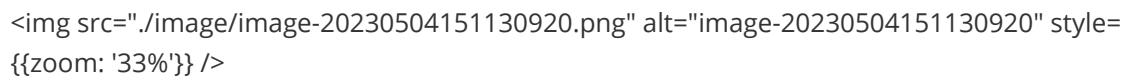
        }
    }
}

```

Get方法

例如：获取相机录像保存数量，在Key List中查找并选中CameraRecordNumber，由于CameraStatus只支持Get和Set，右侧Listen、Action、Report会灰化不可点，点击按钮Get，下方会展示CameraRecordNumber的参数打印。

下图为Get方法调用成功，打印结果的截图：

The image is a placeholder for a screenshot showing the result of a successful Get method call, specifically displaying the CameraRecordNumber parameter.

调用方法示例：

```

fun getCameraRecordNumber() {
    val keyCameraRecordNumber: AutelKey<Int> =
        KeyTools.createKey(CameraKey.KeyCameraRecordNumber)

    DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.getValue(
        keyCameraRecordNumber, object :
            CommonCallbacks.CompletionCallbackWithParam<Int> {
                override fun onSuccess(t: Int?) {
                    TODO("Not yet implemented")
                }

                override fun onFailure(error: IAutelCode, msg: String?) {
                    TODO("Not yet implemented")
                }
            })
}

```

Set方法

例如：设置录像开关，在Key List中查找并选中CameraRecordEnable，由于CameraRecordEnable只支持Get、Set，右侧的Listen、Action、Report会灰化不可点，点击按钮Set进行参数设置，下方会展示CameraRecordEnable的请求成功或失败的打印结果

下图为Set方法设置和调用成功的截图：

调用方法示例：

```
fun setRecordEnable(open: Boolean) {  
  
    DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.setValue(  
        KeyTools.createKey(CameraKey.KeyCameraRecordEnable), open,  
        object : CommonCallbacks.CompletionCallback {  
            override fun onSuccess() {  
  
            }  
  
            override fun onFailure(code: IAutelCode, msg: String?) {  
  
            }  
  
        }  
    )  
}
```

Action方法

例如：重置相机，在Key List中查找并选中CameraReset，由于CameraReset只支持Action，右侧的Get、Set、Listen、Report会灰化不可点，点击按钮Action，下方会展示CameraReset的请求成功或失败的打印结果

下图为调用CameraReset成功，打印结果的截图：

调用方法示例：

```
fun cameraReset() {  
    val key = KeyTools.createKey(CameraKey.KeyCameraReset)  
  
    DeviceManager.getDeviceManager().getFirstDroneDevice()?.getKeyManager()?.performAction(key, null, object : CommonCallbacks.CompletionCallbackWithParam<Void> {  
        override fun onSuccess(t: Void?) {  
  
        }  
  
        override fun onFailure(code: IAutelCode, msg: String?) {  
        }  
    })  
}
```