

# SVCameraDevice SDK接口说明

## 版本历史

版本号	作者	修改日期	修改说明	备注
V1.00	吴俊成	2021-11-17	正式发布	
-				

## SDK接口函数

SDK对外的所有的操作都集中 **SVCameraDevice** 类中.

该类现在提供一个设备查询接口和5个操作CameraDevice设备的接口.

1. **List<SVCameraDeviceInfo> enumDevice**  
枚举出所有CameraDevice设备.
2. **boolean openDevice(SVCameraDeviceInfo)**  
打开指定设备.
3. **String getSoftwareVersion()**  
读取软件版本.
4. **String getCameraModel()**  
读取摄像头模组名称
5. **boolean Zoom(byte zoom\_type, int step)**  
放大缩小, zoom\_type指定类型. step指定缩放的步进值.
6. **boolean Move(byte direction, int step)**  
移动摄像头. direction 指定移动的方向. step指定移动步长.

## 辅助类

针对每一个API接口,额外定义了参数的取值范围类.

调用该API接口时,可以使用辅助类里定义的取值范围.

以 **Move** 接口为例, 为了开发人员方便使用,direction的取值范围定义了一个列表.

```
public class DIRECTION_TYPE {  
    static public final byte DIRECTION_UP = 0;  
    static public final byte DIRECTION_DOWN = 1;  
    static public final byte DIRECTION_LEFT = 2;  
    static public final byte DIRECTION_RIGHT = 3;  
}
```

里面定义了调用 **Move** 接口可以使用的方向名称.如果要控制摄像头左移,建议写为:

```
controller.Move(SVCameraDevice.DIRECTION_TYPE.DIRECTION_LEFT, 2);
```

同样. **Zoom** 接口,也定义了相关的辅助定义

```
public class ZOOM_TYPE {  
    static public final byte ZOOM_TYPE_IN = 0;  
    static public final byte ZOOM_TYPE_OUT = 1;  
    static public final byte ZOOM_TYPE_RESET = (byte) 0xFF;  
}
```

## 使用方法

使用伪码如下:

```
//枚举出所有SVCameraDevice设备  
List<SVCameraDeviceInfo> lst = SVCameraDevice.enumDevice(this);  
  
SVCameraDeviceInfo device = lst.get(0);  
  
if(device == null) {  
    Log.e(TAG, "no device found!");  
    return;  
}  
  
//创建一个SVCameraDeviceController实例.  
SVCameraDevice controller = new SVCameraDevice(this);  
  
//打开指定SVCameraDevice设备.  
controller.openDevice(device);  
  
//ZOOM  
controller.Zoom(SVCameraDevice.ZOOM_TYPE.ZOOM_TYPE_IN, 2);  
  
//Move.  
controller.Move(SVCameraDevice.DIRECTION_TYPE.DIRECTION_RIGHT, 2  
);  
  
//取得版本号.  
String strModel = controller.getCameraModel();  
Log.e(TAG, strModel);  
  
String strSWVersion = controller.getSoftwareVersion();  
Log.e(TAG, strSWVersion);
```

## 系统要求

SVCameraDeviceSDK 要求Android版本在 Android 5.0(API Level >= 21)以上.  
并且需要用户授权"USB设备访问权限".