
SONiX SNCAMDLL Specification

Released Version: v2.0

2016/1/5

SONiX TECHNOLOGY CO., LTD

Revision History

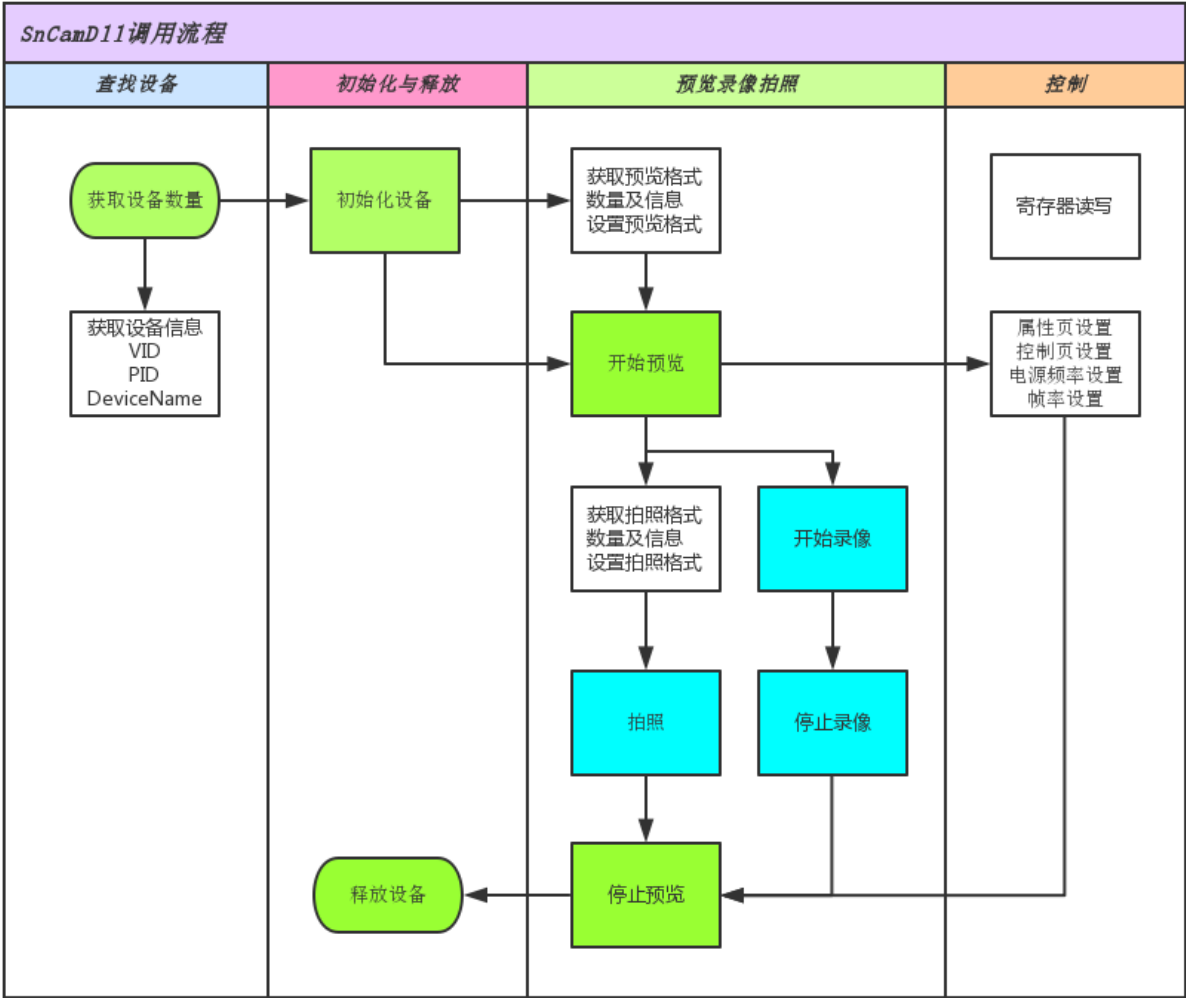
Revision	Data	Description	Author
0.01	2013/08/08	1.Draft Initial	yanzhe_chen
1.00	2013/08/09	1.Release	yanzhe_chen
2.00	2016/01/05	1.Update to adapt new dll	yanze_chen

目录

SONiX SNCAMDLL Specification	0
Revision History	1
调用流程	1
一、参数	1
1、函数返回值	1
2、图像格式	1
3、录像格式	1
4、拍照格式	2
5、属性	2
6、控制	2
7、属性、控制标志位	3
8、电源频率	3
二、设备相关	4
1、获取设备数量	4
2、获取设备信息	4
3、设备初始化	4
4、设备释放	4
三、视频相关	5
1、回调函数	5
2、预览	5
3、拍照	6

4、录像	7
四、控制相关	8
1、设备属性	8
2、设备控制	8
3、帧率	9
4、电源频率设置	10
五、硬件 I/O	11
DSP 寄存器读写	11

调用流程



绿色为必须执行项目，只有设备初始化成功后才可以进行后续操作

白色为可选择执行项目

蓝色为必须开启预览之后才能执行的项目

一、参数

1、函数返回值

动态链接库中所有函数返回值均在以下枚举中：

```
typedef enum tagDS_CAMERA_STATUS
{
    STATUS_OK = 1,                // 动作成功
    STATUS_INTERNAL_ERROR = 0,    // 内部错误
    STATUS_NO_DEVICE_FIND = -1,   // 没有发现相机
    STATUS_NOT_ENOUGH_SYSTEM_MEMORY = -2, // 没有足够系统内存
    STATUS_HW_IO_ERROR = -3,      // 硬件IO错误
    STATUS_PARAMETER_INVALID = -4, // 参数无效
    STATUS_PARAMETER_OUT_OF_BOUND = -5, // 参数越界
    STATUS_FILE_CREATE_ERROR = -6, // 创建文件失败
    STATUS_FILE_INVALID = -7,      // 文件格式无效
    STATUS_NO_RESOLUTION_FOUND = -8, // 不支持此格式
    STATUS_NO_CAM_INIT = -9        // 没有初始化
}DS_CAMERA_STATUS;
```

2、图像格式

静态图像与动态图像格式，用以设置拍照和预览图像格式。

```
typedef enum tagDS_COLORSPACE
{
    COLORSPACE_YUY2,
    COLORSPACE_MJPG,
    COLORSPACE_RGB24,
    COLORSPACE_I420
}DS_COLORSPACE;
```

3、录像格式

录像文件是否压缩

```
typedef enum tagDS_VIDEOFORMAT
{
    VIDEOFORMAT_avi,                // AVI
    VIDEOFORMAT_asf                  // ASF
}
```

本数据为松翰科技股份有限公司专有之财产，未经书面同意不准透露、使用，亦不准复印或转变成任何其他形式使用。

The information contained herein is the exclusive property of SONIX and shall not be distributed, reproduced or disclosed in whole or no in part without prior written permission of SONIX.

```
}DS_VIDEOFORMAT;
```

4、拍照格式

拍设照片保存为 jpeg 或者 bmp

```
typedef enum tagDS_PICTUREFORMAT
{
    PICTUREFORMAT_JPG,           // JPEG
    PICTUREFORMAT_BMP           // BMP
}DS_PICTUREFORMAT;
```

5、属性

摄像头属性设置共 10 项。

```
typedef enum tagDS_CAMERA_PROPERTY
{
    PROPERTY_Brightness = 0,      // 亮度
    PROPERTY_Contrast = 1,        // 对比度
    PROPERTY_Hue = 2,             // 色调
    PROPERTY_Saturation = 3,      // 饱和度
    PROPERTY_Sharpness = 4,       // 清晰度
    PROPERTY_Gamma = 5,           // 伽马
    PROPERTY_ColorEnable = 6,     // 黑白
    PROPERTY_WhiteBalance = 7,    // 白平衡
    PROPERTY_BacklightCompensation = 8, // 逆光对比
    PROPERTY_Gain = 9             // 增益
} DS_CAMERA_PROPERTY;
```

6、控制

摄像头预览控制共 7 项

```
typedef enum tagDS_CAMERA_CONTROL
{
    CONTROL_Pan = 0,              // 全景
    CONTROL_Tilt = 1,             // 倾斜
    CONTROL_Roll = 2,             // 镜像翻转
    CONTROL_Zoom = 3,             // 缩放
    CONTROL_Exposure = 4,         // 曝光
    CONTROL_Iris = 5,             // 光圈
    CONTROL_Focus = 6             // 焦点
}
```

```
} DS_CAMERA_CONTROL;
```

7、属性、控制标志位

属性控制标志位，标志摄像头相应属性或控制是否自动处理。

```
typedef enum tagDS_PROPERTY_FLAGS
{
    PROPERTY_FLAGS_Auto = 1,           // 自动
    PROPERTY_FLAGS_Manual = 2         // 手动
} DS_PROPERTY_FLAGS, DS_CONTROL_FLAGS;
```

8、电源频率

50Hz、60Hz 电源频率切换。

```
typedef enum tagDS_POWER_LINE
{
    POWER_LINE_50Hz = 1,
    POWER_LINE_60Hz = 2
} DS_POWER_LINE;
```


二、设备相关

1、获取设备数量

DT_API SNCAMDLL_API DS_CAMERA_STATUS GetCameraCount(OUT LONG* CameraCout);

- * Function: GetCameraCount
- * Description: 获取Camera数量（必须调用）
- * Parameters: Camera数量
- * Return : 返回状态

2、获取设备信息

DT_API SNCAMDLL_API DS_CAMERA_STATUS GetCameraInfo(IN BYTE CamNum,
OUT CHAR** VID,
OUT CHAR** PID,
OUT CHAR** DeviceName);

- * Function: GetCameraInfo
- * Description: 获取Camera信息
- * Parameters: 相机编号, VID, PID, 设备名称
- * Return : 返回状态

3、设备初始化

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraInit(BYTE CamNum,
DS_SNAP_PROC pCallbackFunction,
HWND hWndDisplay);

- * Function: CameraInit
- * Description: 初始化Camera（必须调用）
- * Parameters: 相机编号, 回调函数, 预览窗口句柄（无需预览则NULL）
- * Return : 返回状态

4、设备释放

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraUnInit(BYTE CamNum);

- * Function: CameraUnInit
- * Description: 释放Camera（必须调用）
- * Parameters: 相机编号
- * Return : 返回状态

三、视频相关

1、回调函数

回调函数在开启视频预览时提取出每帧图像原始数据和数据长度，需要在设备初始化前定义。

```
typedef int (CALLBACK* DS_SNAP_PROC)(BYTE *pImageBuffer, LONG BufferLength);
```

- * Function: DS_SNAP_PROC
- * Description: 回调函数（必须定义）
- * Parameters: 每帧预览图像数据，数据长度
- * Return : 被动调用

2、预览

➤ 获取预览格式数量

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS GetPreviewResolutionCount(IN BYTE CamNum,  
                                                                OUT LONG* ResolutionCount);
```

- * Function: GetPreviewResolutionCount
- * Description: 获取Camera支持的预览格式数量
- * Parameters: 相机编号，格式数量
- * Return : 返回状态

➤ 获取预览格式信息

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS GetPreviewResolutionInfo(IN BYTE CamNum,  
                                                                IN BYTE ResolutionNum,  
                                                                OUT DS_COLORSPACE* piColorspace,  
                                                                OUT LONG* plWidth,  
                                                                OUT LONG* plHeight);
```

- * Function: GetPreviewResolutionInfo
- * Description: 获取相应预览格式信息
- * Parameters: 相机编号，格式编号，色彩空间，宽，高
- * Return : 返回状态

➤ 开始预览

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPlay(BYTE CamNum);
```

- * Function: CameraPlay
- * Description: 开始预览
- * Parameters: 相机编号
- * Return : 返回状态

➤ 停止预览

本数据为松翰科技股份有限公司专有之财产，未经书面同意不准透露、使用，亦不准复印或转变成任何其他形式使用。

The information contained herein is the exclusive property of SONIX and shall not be distributed, reproduced or disclosed in whole or no in part without prior written permission of SONIX.

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStop(BYTE CamNum);

- * Function: CameraStop
- * Description: 停止预览
- * Parameters: 相机编号
- * Return : 返回状态

➤ 预览格式变更

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSetVideoFormat(BYTE CamNum,
BYTE ResolutionNum);

- * Function: CameraSetVideoFormat
- * Description: 设置预览格式
- * Parameters: 相机编号, 格式编号
- * Return : 返回状态

3、拍照

➤ 获取拍照格式数量

DT_API SNCAMDLL_API DS_CAMERA_STATUS GetStillResolutionCount(IN BYTE CamNum,
OUT LONG* ResolutionCount);

- * Function: GetStillResolutionCount
- * Description: 获取Camera支持的拍照格式数量
- * Parameters: 相机编号, 格式数量
- * Return : 返回状态

➤ 获取拍照格式信息

DT_API SNCAMDLL_API DS_CAMERA_STATUS GetStillResolutionInfo(IN BYTE CamNum,
IN BYTE ResolutionNum,
OUT DS_COLORSPACE* piColorspace,
OUT LONG* plWidth,
OUT LONG* plHeight);

- * Function: CameraSetStillFormat
- * Description: 设置拍照格式
- * Parameters: 相机编号, 格式编号
- * Return : 返回状态

➤ 设置拍照格式

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSetStillFormat(BYTE CamNum,
BYTE ResolutionNum);

- * Function: CameraSetStillFormat
- * Description: 设置拍照格式
- * Parameters: 相机编号, 格式编号
- * Return : 返回状态

➤ 拍照, 通过设备 StillPin 获取图像, 若 10 秒内拍照未成功则返回失败

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSnapShot(BYTE CamNum,
DS_PICTUREFORMAT uiformat,
WCHAR* wszPictureFile
);
```

```
* Function:      CameraSnapShot
* Description:    拍照
* Parameters:     相机编号，BMPorJPEG，图片文件名称（包括图片路径、文件名）
* Return :        返回状态
```

4、录像

➤ 开始录像，需要先开启预览。

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStartCapture(BYTE CamNum,
DS_VIDEOFORMAT uiformat,
WCHAR* wszCaptureFile);
```

```
* Function:      CameraStopCapture
* Description:    开始录像
* Parameters:     相机编号
* Return :        返回状态
```

➤ 停止录像

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStopCapture(BYTE CamNum);
```

```
* Function:      CameraStopCapture
* Description:    停止录像
* Parameters:     相机编号
* Return :        返回状态
```

四、控制相关

1、设备属性

➤ 获取属性范围

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPropertyGetRange(IN BYTE CamNum,
                                                             IN DS_CAMERA_PROPERTY uiProperty,
                                                             OUT LONG* pMin, OUT LONG* pMax,
                                                             OUT LONG* pSteppingDelta,
                                                             OUT LONG* pDefault,
                                                             OUT DS_PROPERTY_FLAGS* pCapsFlags);
```

- * **Function:** CameraPropertyGetRange
- * **Description:** 获取属性页项目范围、步进、默认值、是否自动
- * **Parameters:** 相机编号, 属性值, 最小值, 最大值, 步进, 是否自动
- * **Return :** 返回状态

➤ 获取属性值

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPropertyGet(BYTE CamNum,
                                                         DS_CAMERA_PROPERTY uiProperty,
                                                         OUT LONG* IValue,
                                                         OUT DS_PROPERTY_FLAGS* Flags);
```

- * **Function:** CameraPropertyGet
- * **Description:** 获取属性页项目设置值
- * **Parameters:** 相机编号, 属性值, 设置值, 是否自动
- * **Return :** 返回状态

➤ 设置属性值

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPropertySet(BYTE CamNum,
                                                         DS_CAMERA_PROPERTY uiProperty,
                                                         IN LONG IValue,
                                                         IN DS_PROPERTY_FLAGS Flags);
```

- * **Function:** CameraPropertySet
- * **Description:** 设置属性页项目
- * **Parameters:** 相机编号, 属性值, 设置值, 是否自动
- * **Return :** 返回状态

2、设备控制

➤ 获取控制范围

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraControlGetRange(BYTE CamNum,
                                                             DS_CAMERA_CONTROL uiControl,
                                                             OUT LONG* pMin, OUT LONG* pMax,
                                                             OUT LONG* pSteppingDelta,
                                                             OUT LONG* pDefault,
                                                             OUT DS_CONTROL_FLAGS* pCapsFlags);
```

- * Function: CameraControlGetRange
- * Description: 获取控制页项目范围、步进、默认值、是否自动
- * Parameters: 相机编号，控制值，最小值，最大值，步进，是否自动
- * Return : 返回状态

➤ 获取控制值

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraControlGet(BYTE CamNum,
                                                         DS_CAMERA_CONTROL uiControl,
                                                         OUT LONG* IValue,
                                                         OUT DS_CONTROL_FLAGS* Flags);
```

- * Function: CameraControlGet
- * Description: 获取控制页项目设置值
- * Parameters: 相机编号，控制值，设置值，是否自动
- * Return : 返回状态

➤ 设置控制值

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraControlSet(BYTE CamNum,
                                                         DS_CAMERA_CONTROL uiControl,
                                                         IN LONG IValue,
                                                         IN DS_CONTROL_FLAGS Flags);
```

- * Function: CameraControlSet
- * Description: 设置控制页项目
- * Parameters: 相机编号，控制值，设置值，是否自动
- * Return : 返回状态

3、帧率

➤ 获取帧率

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraGetFrameSpeed(BYTE CamNum,
                                                            OUT DOUBLE* dFramerate);
```

- * Function: CameraGetFrameSpeed
- * Description: 获取当前帧率
- * Parameters: 相机编号，帧率
- * Return : 返回状态

➤ 设置预览帧率，需先开启图像

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSetFrameSpeed(BYTE CamNum,
```

IN DOUBLE dFramerate);

- * Function: CameraSetFrameSpeed
- * Description: 设置帧率
- * Parameters: 相机编号, 帧率
- * Return : 返回状态

4、电源频率设置

- 设置电源频率, 需先开启图像

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSetPowerLine(BYTE CamNum,
IN DS_POWER_LINE PowerLine
);

- * Function: CameraSetPowerLine
- * Description: 设置电源频率
- * Parameters: 相机编号, 电源频率
- * Return : 返回状态

五、硬件 I/O

DSP 寄存器读写

➤ 读

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraReadDSP(BYTE CamNum,  
LONG laddress, OUT LONG* lData);
```

- * Function: CameraReadDSP
- * Description: 读寄存器
- * Parameters: 相机编号, 寄存器地址, 值
- * Return : 返回状态

➤ 写

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraWriteDSP(BYTE CamNum,  
LONG laddress, IN LONG* lData);
```

- * Function: CameraWriteDSP
- * Description: 写寄存器
- * Parameters: 相机编号, 寄存器地址, 值
- * Return : 返回状态