SONiX SNCAMDLL Specification

Released Version: v2.0

2016/1/5

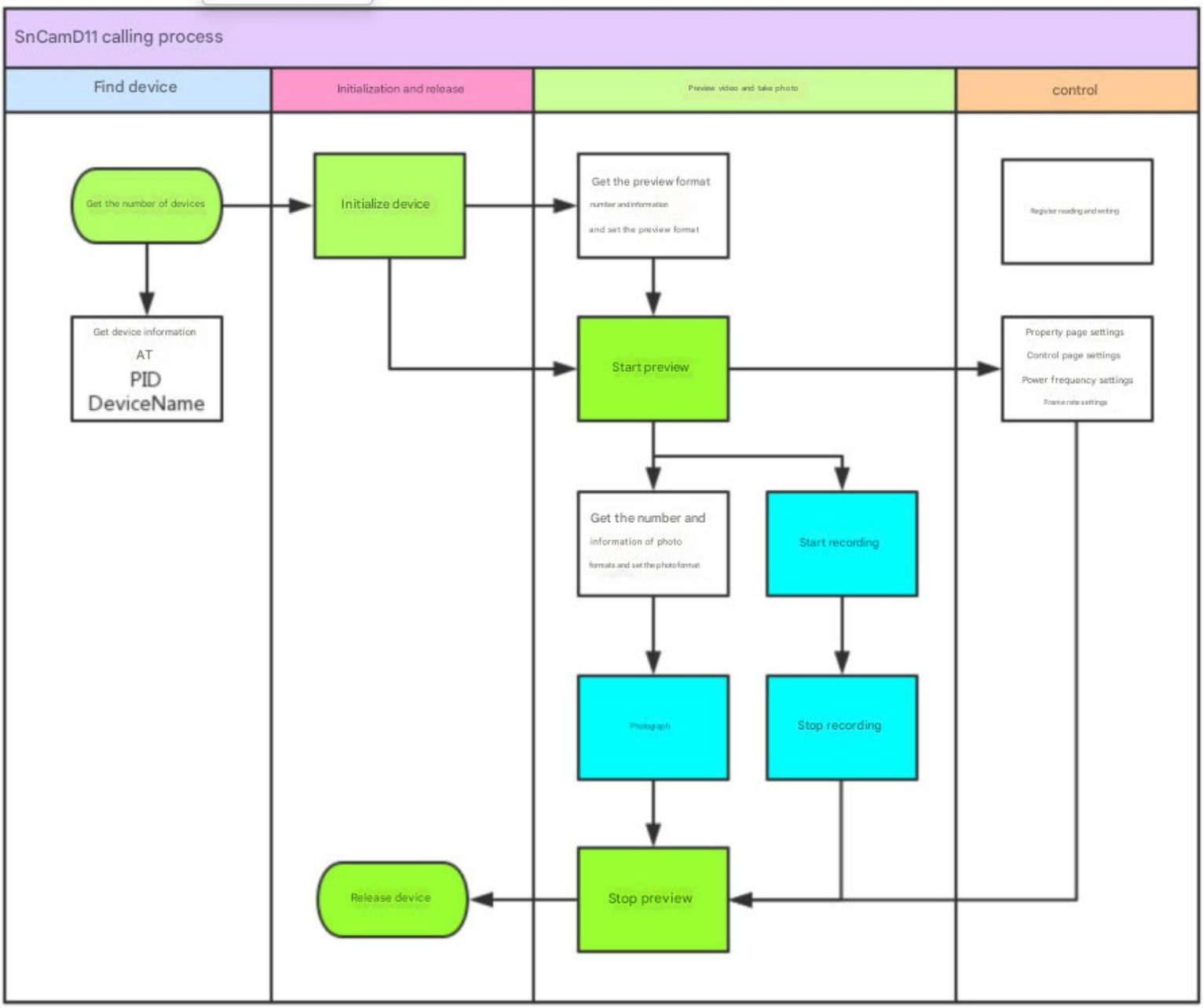
SONiX TECHNOLOGY CO., LTD

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

Table of contents

Highlight text



The blue ones are the projects that must be previewed before they can be executed.

1. Parameters

1. Function return value

The return values of all functions in the dynamic link library are in the following enumeration:

```
typedef enum tagDS_CAMERA_STATUS
{
 STATUS_OK = 1,                                  // Action successful
 STATUS_INTERNAL_ERROR = 0,                      // Internal error
 STATUS_NO_DEVICE_FIND = -1,                     // No camera found
 STATUS_NOT_ENOUGH_SYSTEM_MEMORY = -2,           // Not enough system memory
 STATUS_HW_IO_ERROR = -3,                        // Hardware IO error
 STATUS_PARAMETER_INVALID = -4,                  // Invalid parameter
 STATUS_PARAMETER_OUT_OF_BOUND = -5,             // Parameter out of bounds
 STATUS_FILE_CREATE_ERROR = -6,                  // Failed to create file
 STATUS_FILE_INVALID = -7,                       // Invalid file format
 STATUS_NO_RESOLUTION_FOUND=-8,                  // This format is not supported
 STATUS_NO_CAM_INIT=-9                           // Not initialized
}DS_CAMERA_STATUS;
```

2. Image Format

Still image and dynamic image formats are used to set the photo taking and preview image formats.

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

3. Video format

Whether the video file is compressed

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

```
}DS_VIDEOFORMAT;
```

4. Photo format

Take pictures and save them as jpeg or bmp

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

5. Attributes

There are 10 camera property settings in total.

```
typedef enum tagDS_CAMERA_PROPERTY
{
 PROPERTY_Brightness = 0,              // Brightness
 PROPERTY_Contrast = 1,                // Contrast
 PROPERTY_Hue = 2,                     // Hue
 PROPERTY_Saturation = 3,              // saturation
 PROPERTY_Sharpness = 4,               // Clarity
 PROPERTY_Gamma = 5,                   // Gamma
 PROPERTY_ColorEnable = 6,             // Black and white
 PROPERTY_WhiteBalance = 7,            // White balance
 PROPERTY_BacklightCompensation = 8,   // Backlight contrast
 PROPERTY_Gain = 9                     // Gain
} DS_CAMERA_PROPERTY;
```

6. Control

Camera preview control total 7 items

```
typedef enum tagDS_CAMERA_CONTROL
{
 CONTROL_Pan = 0,                      // Panorama
 CONTROL_Tilt = 1        ,             // Tilt
 CONTROL_Roll = 2        ,             //Mirror flip
 CONTROL_Zoom= 3         ,             // Zoom
 CONTROL_Exposure = 4        ,         // Exposure
 CONTROL_Iris = 5        ,             // Aperture
 CONTROL_Focus= 6                      // Focus
```

```
} DS_CAMERA_CONTROL;
```

7. Attributes and control flags

Property control flag, marking the corresponding property of the camera or controlling whether to process automatically.

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

8. Power frequency

50Hz, 60Hz power frequency switching.

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

2. Equipment related

1. Get the number of devices

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS GetCameraCount(OUT LONG* CameraCout);
* Function:        GetCameraCount
*       Description: Get the number of cameras (must be called)
* Parameters: Number of Cameras
* Return :         Return Status
```

2. Get device information

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS GetCameraInfo(IN BYTE CamNum,
                                                   OUT CHAR** VID,
                                                   OUT CHAR** PID,
                                                   OUT CHAR** DeviceName);
* Function:        GetCameraInfo
*       Description: Get Camera information
* Parameters: Camera ID, VID, PID, device name
* Return :         Return Status
```

3. Device initialization

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraInit(BYTE CamNum,
                                                DS_SNAP_PROC pCallbackFunction,
                                                HWND hWndDisplay);
* Function:        CameraInit
*       Description: Initialize Camera (must be called)
* Parameters: Camera number, callback function, preview window handle (NULL if no preview is required)
* Return :         Return Status
```

4. Equipment release

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraUnInit(BYTE CamNum);
* Function:        CameraUnInit
*       Description: Release Camera (must be called)
* Parameters: Camera ID
* Return :         Return Status
```

3. Video related

1. Callback function

The callback function extracts the original data and data length of each frame when the video preview is turned on. It needs to be defined before the device is initialized.

```
typedef int (CALLBACK* DS_SNAP_PROC)(BYTE *pImageBuffer, LONG BufferLength);
* Function:          DS_SNAP_PROC
*
      Description: Callback function (must be defined)
* Parameters: Preview image data per frame, data length
* Return :           Passive call
```

2. Preview

ÿ Get the number of preview formats

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS GetPreviewResolutionCount(IN BYTE CamNum,
                                                               OUT LONG* ResolutionCount);
* Function:          GetPreviewResolutionCount
*
      Description: Get the number of preview formats supported by the Camera
* Parameters: Camera number, format number
* Return :           Return Status
```

ÿ Get preview format information

```
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: Get the corresponding preview format information
* Parameters: Camera ID, Format ID, Color Space, Width, Height
* Return :           Return Status
```

ÿ Start preview

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPlay(BYTE CamNum);
* Function:          CameraPlay
*
      Description: Start preview
* Parameters: Camera ID
* Return :           Return Status
```

ÿ Stop preview

DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStop(BYTE **CamNum);**

* Function: CameraStop

* Description: Stop preview

* Parameters: Camera ID

* Return : Return Status

ÿ Preview format changes

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

* Function: CameraSetVideoFormat

* Description: Set the preview format

* Parameters: Camera ID, Format ID

* Return : Return Status

3. Take photos

ÿ Get the number of photo formats

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

* Function: GetStillResolutionCount

* Description: Get the number of photo formats supported by the Camera

* Parameters: Camera number, format number

* Return : Return Status

ÿ Get photo format information

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: Set the photo format

* Parameters: Camera ID, Format ID

* Return : Return Status

ÿ Set the photo format

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

* Function: CameraSetStillFormat

* Description: Set the photo format

* Parameters: Camera ID, Format ID

* Return : Return Status

ÿ Take a photo and get the image through the device StillPin. If the photo is not taken within 10 seconds, it will return failure.

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraSnapShot(BYTE CamNum,
                                                    DS_PICTUREFORMAT uiformat,
                                                    WCHAR* wszPictureFile
                                                    );
* Function:        CameraSnapShot
*       Description: Take a photo
* Parameters: Camera number, BMP or JPEG, image file name (including image path and file name)
* Return :         Return Status
```

4. Video recording

ÿ To start recording, you need to enable preview first.

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStartCapture(BYTE CamNum,
                                                        DS_VIDEOFORMAT uiformat,
                                                        WCHAR* wszCaptureFile);
* Function:        CameraStopCapture
*       Description: Start recording
* Parameters: Camera ID
* Return :         Return Status
```

ÿ Stop recording

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraStopCapture(BYTE CamNum);
* Function:        CameraStopCapture
*       Description: Stop recording
* Parameters: Camera ID
* Return :         Return Status
```

4. Control Related

1. Device attributes

ÿ Get attribute range

```
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: Get the property page item range, step, default value, and whether it is automatic
* Parameters: Camera number, property value, minimum value, maximum value, step, whether automatic
* Return :        Return Status
```

ÿ Get attribute values

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPropertyGet(BYTE CamNum,
                                                       DS_CAMERA_PROPERTY uiProperty,
                                                       OUT LONG* lValue,
                                                       OUT DS_PROPERTY_FLAGS* Flags);
* Function:        CameraPropertyGet
*       Description: Get the property page item setting value
* Parameters: Camera number, property value, setting value, whether automatic
* Return :        Return Status
```

ÿ Set attribute values

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraPropertySet(BYTE CamNum,
                                                       DS_CAMERA_PROPERTY uiProperty,
                                                       IN LONG lValue,
                                                       IN DS_PROPERTY_FLAGS Flags);
* Function:        CameraPropertySet
*       Description: Set the property page item
* Parameters: Camera number, property value, setting value, whether automatic
* Return :        Return Status
```

2. Device Control

ÿ Obtaining control scope

```
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: Get the control page item range, step, default value, and whether it is automatic
* Parameters: Camera number, control value, minimum value, maximum value, step, whether automatic
* Return :           Return Status
```

ÿ Get control value

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraControlGet(BYTE CamNum,
                                                     DS_CAMERA_CONTROL uiControl,
                                                     OUT LONG* lValue,
                                                     OUT DS_CONTROL_FLAGS* Flags);
* Function:          CameraControlGet
*
      Description: Get the setting value of the control page item
* Parameters: Camera number, control value, setting value, whether automatic
* Return :           Return Status
```

ÿ Set control value

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraControlSet(BYTE CamNum,
                                                     DS_CAMERA_CONTROL uiControl,
                                                     IN LONG lValue,
                                                     IN DS_CONTROL_FLAGS Flags);
* Function:          CameraControlSet
*
      Description: Set the control page item
* Parameters: Camera number, control value, setting value, whether automatic
* Return :           Return Status
```

3. Frame rate

ÿ Get frame rate

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraGetFrameSpeed(BYTE CamNum,
                                                        OUT DOUBLE* dFramerate);
* Function:          CameraGetFrameSpeed
*
      Description: Get the current frame rate
* Parameters: Camera number, frame rate
* Return :           Return Status
```

ÿ To set the preview frame rate, you need to open the image first.

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

IN DOUBLE **dFramerate);**

* Function: CameraSetFrameSpeed
*
Description: Set the frame rate
* Parameters: Camera number, frame rate
* Return : Return Status

4. Power frequency setting

ÿ To set the power frequency, you need to turn on the image first.

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

* Function: CameraSetPowerLine
*
Description: Set the power frequency
* Parameters: Camera number, power frequency
* Return : Return Status

5. Hardware I/O

DSP register read and write

ÿ Read

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraReadDSP(BYTE CamNum,
                                                   LONG laddress, OUT LONG* lData);
* Function:          CameraReadDSP
*       Description: Read register
* Parameters: Camera number, register address, value
* Return :           Return Status
```

ÿ Write

```
DT_API SNCAMDLL_API DS_CAMERA_STATUS CameraWriteDSP(BYTE CamNum,
                                                    LONG laddress, IN LONG* lData);
* Function:          CameraWriteDSP
*       Description: Write register
* Parameters: Camera number, register address, value
* Return :           Return Status
```