Android SDK API

Version: 0.8

Date: 2023/10/2

Category

Revision History

Revision	Author	Date	Description
0.1	Iris	2020/5/26	1. Initial version
0.2	Iris	2020/7/21	1. Add lamp ramp up ADC and lamp ADC among repeated times.
0.3	Iris	2020/9/16	1. Modify the flow chart of the Connected to the Device. 2. Add the section of other receiver.
0.4	Iris	2021/4/22	1. Add the flow chart of the all section. 2. Modify the description of the all section. 3. Remove the section of Lamp Ramp Up ADC and Lamp ADC among repeated times. Add to the section of Perform Scan – Normal, Perform Scan – Quick Set and Perform Scan – Manual. 4. Remove the section of other receiver. Add to the section of Perform Scan – Manual.
0.5	Iris	2021/7/2	1. Add the section of Warm-Up the device. 2. Modify the flow chart of the Connected to the Device and add 1-(b) to teach how to do the warm-up.
0.6	Iris	2022/1/4	1. Add the section of introduction and specification. 2. Add note to the required section. 3. Add corresponding API table in each section.
0.7	Iris	2023/3/31	1. The specification items add extend plus machines. 2. The Connected to the Device chapter removes the need to use Notify_IsEXTVersion() to notify the ISC SDK whether the device is an extended wavelength device.
0.8	Iris	2023/10/2	1. Add Calculate AU from a New Reference Scan topic.

Introduction

This document is to let users understand how to use the SDK provided by ISC for development. It introduces the entire APP process, including connection, scanning..., and provides how to call API for development. Corresponding with source code will make it easier to get started. If you want to know more about BLE callback, please refer to **ISC NIRScan Bluetooth Communications Data Sheet V1.3.pdf.**

Specification

The TIVA versions and functions supported by this SDK are as follows:

Standard Wavelength (900nm~1700nm)

TIVA	Function									
	Normal Scan	Quick Set Scan	Manual Scan	Maintain	Activation	Device Information	Device Status	Read Device Configuration	Add Device Configuration	Lock Button
<v2.1.0.67	X	X	X	X	X	X	X	X	X	X
v2.1.0.67~v2.3.2	V	X	X	X	X	V	V	V	X	X
v2.4.2	V	V	V	V	V	V	V	V	V	X
≧v2.4.3	V	V	V	V	V	V	V	V	V	V

Extend Wavelength (1350nm~2150nm)

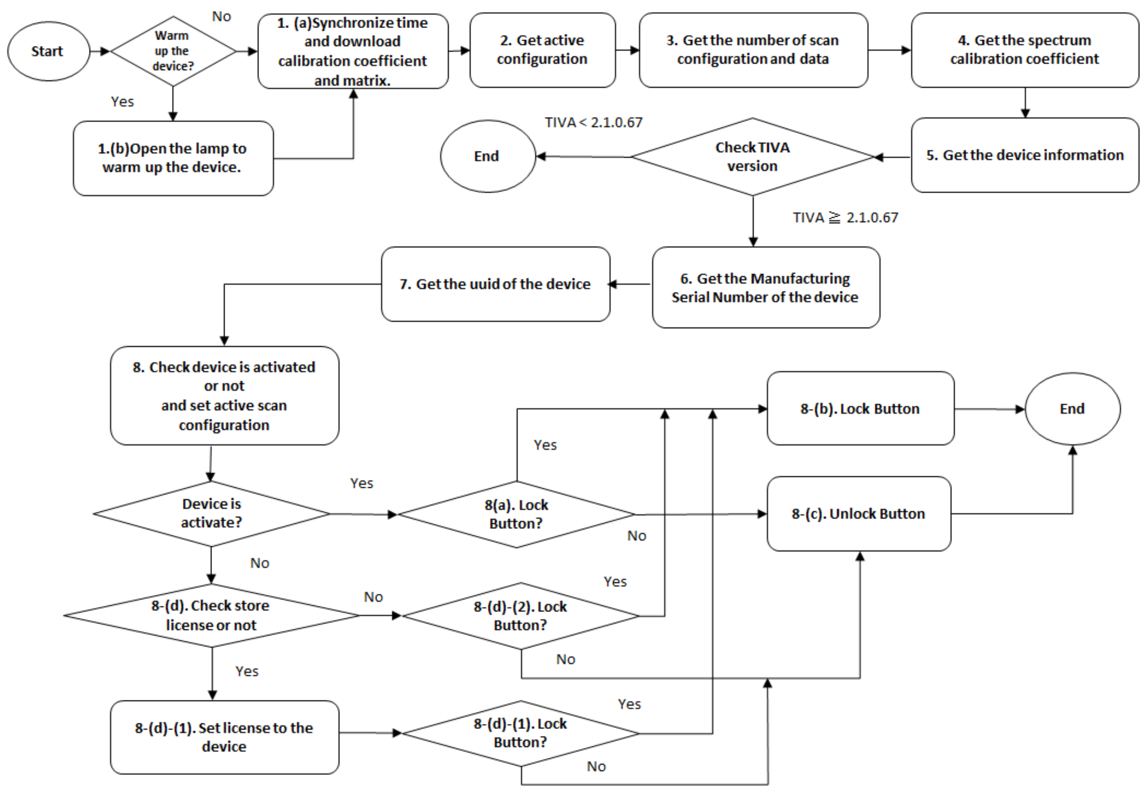
TIVA	Function									
	Normal Scan	Quick Set Scan	Manual Scan	Maintain	Activation	Device Information	Device Status	Read Device Configuration	Add Device Configuration	Lock Button
<v3.3.0	V	V	V	V	V	V	V	V	V	X
≧v3.3.0	V	V	V	V	V	V	V	V	V	V

Extend Plus Wavelength (1600nm~2400nm)

TIVA	Function									
	Normal Scan	Quick Set Scan	Manual Scan	Maintain	Activation	Device Information	Device Status	Read Device Configuration	Add Device Configuration	Lock Button
≧v5.0.1	V	V	V	V	V	V	V	V	V	V

Connected to the Device

ScanViewActivity.java



Start
Warm up the device?
No
Yes
1. (a)Synchronize time and download calibration coefficient and matrix.
1.(b)Open the lamp to warm up the device.
2. Get active configuration
3. Get the number of scan configuration and data
4. Get the spectrum calibration coefficient
5. Get the device information
Check TIVA version
TIVA < 2.1.0.67
End
TIVA ≧ 2.1.0.67
6. Get the Manufacturing Serial Number of the device
7. Get the uuid of the device
8. Check device is activated or not and set active scan configuration
Device is activate?
8(a). Lock Button?
8-(b). Lock Button
8-(c). Unlock Button
End
8-(d). Check store license or not
8-(d)-(2). Lock Button?
8-(d)-(1). Set license to the device
8-(d)-(1). Lock Button?

1. When the device is connected, it will first check whether the user want to warm up the device.

   (a) Call the **ISCNIRScanSDK.SetCurrentTime()** to synchronize the time. After synchronize the time, will download calibration coefficient and calibration matrix automatically. Register **RefCoeffDataProgressReceiver** to see the progress of download calibration coefficient. Register **CalMatrixDataProgressReceiver** to see the progress of download calibration matrix. Register **RefDataReadyReceiver** to get the calibration coefficient and matrix.

   Note: In our SDK, should pass calibration coefficient and calibration matrix parameter to ISCNIRScanSDK library. Refer to the code under RefDataReadyReceiver :

```
refCoeff = intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_REF_COEF_DATA);
refMatrix = intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_REF_MATRIX_DATA);
ArrayList<ISCNIRScanSDK.ReferenceCalibration> refCal = new ArrayList<>();
refCal.add(new ISCNIRScanSDK.ReferenceCalibration(refCoeff, refMatrix));
ISCNIRScanSDK.ReferenceCalibration.writeRefCalFile(mContext, refCal);
```

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.SetCurrentTime() (UUID: 0x4348410C-444C-5020-4E49-52204E616E6F 0x4348410F-444C-5020-4E49-52204E616E6F 0x43484111-444C-5020-4E49-52204E616E6F)	Synchronize the time->Download calibration coefficient->Download calibration matrix	RefCoeffDataProgressReceiver (UUID: 0x43484110-444C-5020-4E49-52204E616E6F)	Download calibration coefficient
		CalMatrixDataProgressReceiver (UUID: 0x43484112-444C-5020-4E49-52204E616E6F)	Download calibration matrix
		RefDataReadyReceiver	Get the calibration coefficient and matrix

(b) Call the **ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.ON)** to open the lamp to warm up the device then set **Lamp_Info to LampInfo.WarmDevice**. Register **ReturnSetLampReceiver** to do the next step.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.ON) (UUID: 0x43484144-444C-5020-4E49-52204E616E6F)	Open the lamp to warm up the device	ReturnSetLampReceiver	Notify the completion of the lamp setting, according to lamp info to do next

2. Call the **ISCNIRScanSDK.GetActiveConfig()** to request to get active configuration from the device. Register **GetActiveScanConfReceiver** to get the index of active configuration.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetActiveConfig() (UUID: 0x43484118-444C-5020-4E49-52204E616E6F)	Request to get active configuration	GetActiveScanConfReceiver (UUID: 0x43484118-444C-5020-4E49-52204E616E6F)	Get the index of active configuration

3. Call the **ISCNIRScanSDK.GetScanConfig()** to request to get the number of scan configuration and data from the device. Register **ScanConfSizeReceiver** to get the number of scan configuration. Register **ScanConfReceiver** to get the scan configuration.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetScanConfig() (UUID: 0x43484113-444C-5020-4E49-52204E616E6F 0x43484114-444C-5020-4E49-52204E616E6F)	Get number of scan configuration ->Get scan configuration data	ScanConfSizeReceiver	Get the number of scan configuration
		ScanConfReceiver (UUID: 0x43484115-444C-5020-4E49-52204E616E6F)	Get the scan configuration

4. Call the **ISCNIRScanSDK.GetSpectrumCoef()** to request to get the spectrum calibration coefficient. Register **SpectrumCalCoefficientsReadyReceiver** to get the spectrum calibration coefficient.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetSpectrumCoef() (UUID: 0x4348410D-444C-5020-4E49-52204E616E6F)	Request to get the spectrum calibration coefficient	SpectrumCalCoefficientsReadyReceiver (UUID: 0x4348410E-444C-5020-4E49-52204E616E6F)	Get the spectrum calibration coefficient

Note: In our APP, should pass spectrum calibration coefficient to AddScanConfigViewActivity.java:

passSpectrumCalCoefficients = SpectrumCalCoefficients;

5. Call the **ISCNIRScanSDK.GetDeviceInfo()** to request to get the device information including the device of model name, serial number, HW version and TIVA version. Register **DeviceInfoReceiver** to get the device information. It is necessary to determine the device is standard, extend or extend plus wavelength according to TIVA version. If the device is a standard wavelength, the FW level is obtained with **GetFWLevelStandard(Tivarev)**. If the device is a extension wavelength, the FW level is obtained with **GetFWLevelEXT(Tivarev)**. If the device is a extension plus wavelength, the FW level is obtained with **GetFWLevelEXTPLUS(Tivarev)**. Should call **InitParameter()** to initialize the maximum and minimum wavelengths.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetDeviceInfo() (UUID: 0x00002A29-0000-1000-8000-00805F9B34FB 0x00002A24-0000-1000-8000-00805F9B34FB 0x00002A25-0000-1000-8000-00805F9B34FB 0x00002A27-0000-1000-8000-00805F9B34FB 0x00002A26-0000-1000-8000-00805F9B34FB 0x00002A28-0000-1000-8000-00805F9B34FB)	Request to get the device information	DeviceInfoReceiver	Get the device information
Notify_IsEXTVersion()	Notifying the ISC SDK that device is a standard wavelength or an extended wavelength	X	X
GetFWLevelStandard(Tivarev)	Get standard wavelength FW level	X	X
GetFWLevelEXT(Tivarev)	Get extend wavelength FW level	X	X
InitParameter()	Initialize the maximum and minimum wavelengths	X	X

If the TIVA version≧2.1.0.67 for the device, continue with the following steps.

6. Call the **ISCNIRScanSDK.GetMFGNumber()** to request to get the Manufacturing Serial Number of the device. Register **ReturnMFGNumReceiver** to get the Manufacturing Serial Number.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetMFGNumber() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get the Manufacturing Serial Number	ReturnMFGNumReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get the Manufacturing Serial Number

7. Call the **ISCNIRScanSDK.GetUUID()** to request to get the uuid from the device. Register **GetUUIDReceiver** to get the uuid.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetUUID() (UUID: 0x00002A23-0000-1000-8000-00805F9B34FB)	Request to get the uuid	GetUUIDReceiver	Get the uuid

8. Call the **ISCNIRScanSDK.ReadActivateState()** to request to get whether the device is activate. Register **ReturnReadActivateStatusReceiver** to get whether the device is activated. Call the **ISCNIRScanSDK.SetActiveConfig()** to set the active scan configuration.

(a) Device is activated : Call the **SetDeviceButtonStatus()** to check whether user want to lock button.

(b) Call the **ISCNIRScanSDK.ControlPhysicalButton(ISCNIRScanSDK.PhysicalButton.Lock)** to lock device button.

(c) Call the **ISCNIRScanSDK.ControlPhysicalButton(ISCNIRScanSDK.PhysicalButton.Unlock)** to unlock device button.

(d) Device is not activated : Check whether store license in the app. The shared preferences key is **ISCNIRScanSDK.SharedPreferencesKeys.licensekey**.

(1) Have license : Call the **ISCNIRScanSDK.SetLicenseKey(data)** to set the license. Register **RetrunActivateStatusReceiver** to get the license is valid or not. Call the **SetDeviceButtonStatus()** to check whether user want to lock button.

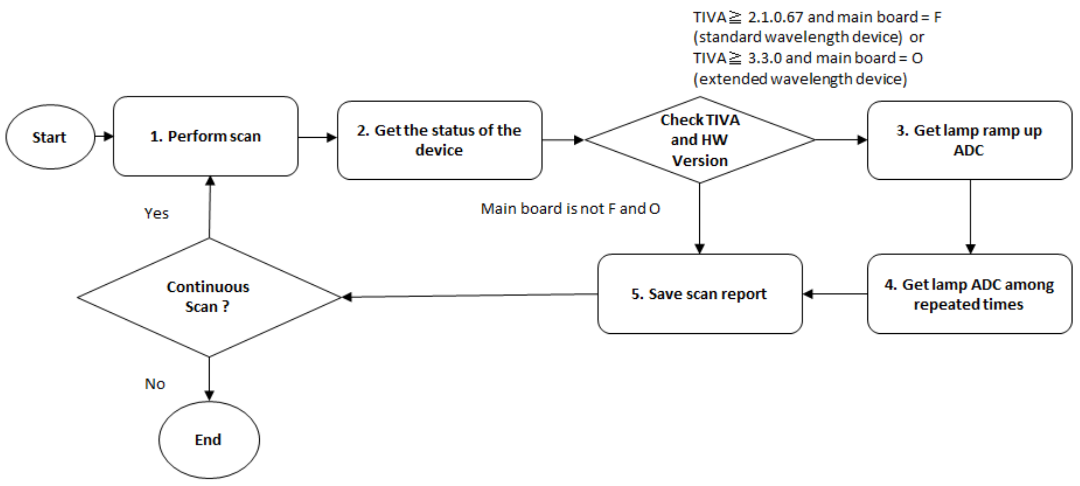
(2) Not have license : Call the **SetDeviceButtonStatus()** to check whether user want to lock button.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.ReadActivateState() (UUID: 0x43484130-444C-5020-4E49-52204E616E6F)	Request to get whether the device is activate	RetrunReadActivateStatusReceiver	Get whether the device is activated
ISCNIRScanSDK.SetActiveConfig() (UUID: 0x43484118-444C-5020-4E49-52204E616E6F)	Set the active scan configuration	X	X

SetDeviceButtonStatus()	Check whether user want to lock button	X	X
ISCNIRScanSDK.ControlPhysicalButton(ISCNIRScanSDK.PhysicalButton.Lock) (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Lock device button	X	X
ISCNIRScanSDK.ControlPhysicalButton(ISCNIRScanSDK.PhysicalButton.Unlock) (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Unlock device button	X	X
ISCNIRScanSDK.SetLicenseKey(data) (UUID: 0x43484130-444C-5020-4E49-52204E616E6F)	Set the license	RetrunActivateStatusReceiver (UUID: 0x43484131-444C-5020-4E49-52204E616E6F)	Get the license is valid or not

Perform Scan – Normal

ScanViewActivity.java

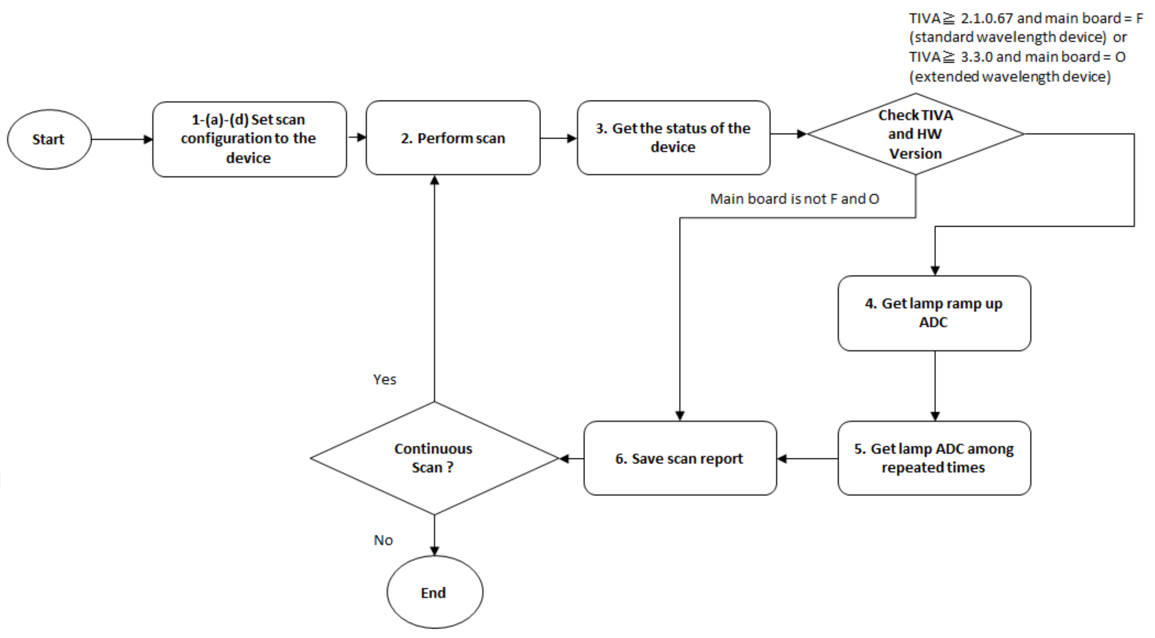

TIVA≧ 2.1.0.67 and main board = F (standard wavelength device) or TIVA≧ 3.3.0 and main board = O (extended wavelength device)
Start
1. Perform scan
2. Get the status of the device
Check TIVA and HW Version
3. Get lamp ramp up ADC
Main board is not F and O
Yes
Continuous Scan ?
5. Save scan report
4. Get lamp ADC among repeated times
No
End

1. Call the **PerformScan(long delaytime)** to scan the sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()** to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan result.
2. Call the **ISCNIRScanSDK.GetDeviceStatus()** to request to get the status of the device including battery capacity, system temperature, humidity, total lamp time, the byte of the device status and the byte of the error status. Register **GetDeviceStatusReceiver** to get the device status.
3. Call the **ISCNIRScanSDK.GetScanLampRampUpADC()** to request to get lamp ramp up ADC. Register **ReturnLampRampUpADCReceiver** to get lamp ramp up ADC.
4. Call the **ISCNIRScanSDK.GetLampADCAverage()** to request to get lamp ADC among repeated times. Register **ReturnLampADCAverageReceiver** to get lamp ADC among repeated times.
5. Call the **writeCSV(Scan_Spectrum_Data)** to save the scan report.

API	Process	Receiver	Receiver Description
PerformScan(long delaytime)	Set delay time to scan sample to avoid BLE hang.	X	X
ISCNIRScanSDK.StartScan() (UUID: 0x4348411D-444C-5020-4E49-52204E616E6F 0x43484127-444C-5020-4E49-52204E616E6F)	Trigger the scan event	ScanDataReadyReceiver (UUID: 0x43484128-444C-5020-4E49-52204E616E6F)	Get the scan result
ISCNIRScanSDK.GetDeviceStatus() (UUID: 0x00002A19-0000-1000-8000-00805F9B34FB 0x43484101-444C-5020-4E49-52204E616E6F 0x43484102-444C-5020-4E49-52204E616E6F 0x43484109-444C-5020-4E49-52204E616E6F 0x43484103-444C-5020-4E49-52204E616E6F 0x43484104-444C-5020-4E49-52204E616E6F)	Request to get the status of the device	GetDeviceStatusReceiver	Get the device status
ISCNIRScanSDK.GetScanLampRampUpADC() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ramp up ADC	ReturnLampRampUpADCReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ramp up ADC
ISCNIRScanSDK.GetLampADCAverage() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ADC among repeated times	ReturnLampADCAverageReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ADC among repeated times
writeCSV(Scan_Spectrum_Data)	Save the scan report	X	X

Perform Scan – Quick Set

ScanViewActivity.java


TIVA≧ 2.1.0.67 and main board = F (standard wavelength device) or TIVA≧ 3.3.0 and main board = O (extended wavelength device)
Start
1-(a)-(d) Set scan configuration to the device
2. Perform scan
3. Get the status of the device
Check TIVA and HW Version
Main board is not F and O
4. Get lamp ramp up ADC
5. Get lamp ADC among repeated times
6. Save scan report
Continuous Scan ?
Yes
No
End

1. Set the scan configuration to the device. **(Note: This only set the configuration to the device memory and generating the scan patterns accordingly.)**

(a) Call the **ChangeScanConfigToByte()** and write quick set UI settings to **ISCNIRScanSDK. ScanConfigInfo write_scan_config**. It will return the byte array of the scan config (return **EXTRA_DATA**).

(b) Call the **ISCNIRScanSDK.ScanConfig(EXTRA_DATA,ISCNIRScanSDK.ScanConfig.SET)** to set the configuration to the device. Register **WriteScanConfigStatusReceiver** to verify whether the scan configuration was set successfully.

(c) Check whether the set scan configuration in the device memory is valid. Call the **ISCNIRScanSDK.ReadCurrentScanConfig()** to get the current device configuration. Register **ReturnCurrentScanConfigurationDataReceiver** to get the byte array of current configuration of the device.

(d) Call the **Compareconfig(intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_CURRENT_CONFIG_DATA))** to compare whether the configuration set by the device and the configuration set by the user in the quick set are the same.

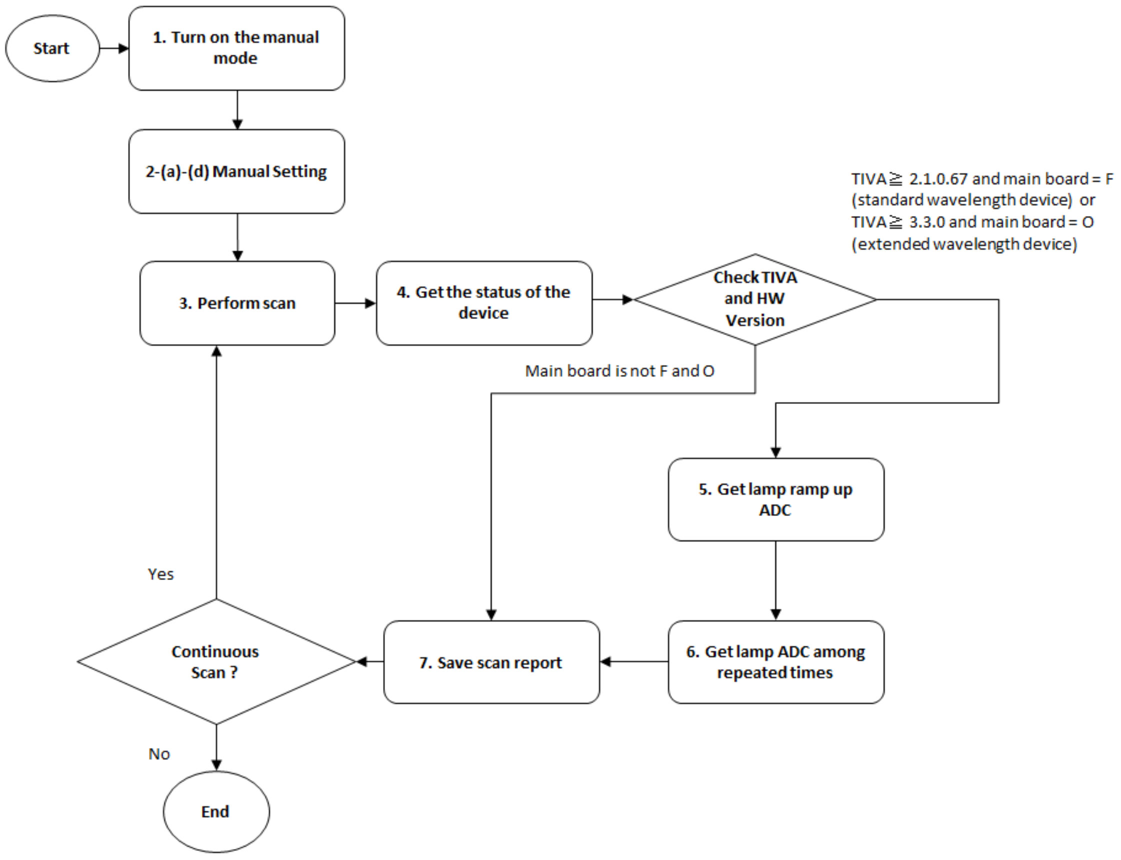
2. Call the **PerformScan(long delaytime)** to scan the sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()** to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan result.
3. Call the **ISCNIRScanSDK.GetDeviceStatus()** to request to get the status of the device including battery capacity, system temperature, humidity, total lamp time, the byte of the device status and the byte of the error status. Register **GetDeviceStatusReceiver** to get the device status.
4. Call the **ISCNIRScanSDK.GetScanLampRampUpADC()** to request to get lamp ramp up ADC. Register **ReturnLampRampUpADCReceiver** to get lamp ramp up ADC.
5. Call the **ISCNIRScanSDK.GetLampADCAverage()** to request to get lamp ADC among repeated times. Register **ReturnLampADCAverageReceiver** to get lamp ADC among repeated times.
6. Call the **writeCSV(Scan_Spectrum_Data)** to save the scan report.

API	Process	Receiver	Receiver Description
ChangeScanConfigToByte()	Convert scan Config to byte array. Return value is EXTRA_DATA	X	X
ISCNIRScanSDK.ScanConfig(EXTRA_DATA,ISCNIRScanSDK.ScanConfig.SET) (UUID: 0x43484142-444C-5020-4E49-52204E616E6F)	Set the configuration to the device	WriteScanConfigStatusReceiver (UUID: 0x43484143-444C-5020-4E49-52204E616E6F)	Verify whether the scan configuration was set successfully
ISCNIRScanSDK.ReadCurrentScanConfig() (UUID: 0x43484140-444C-5020-4E49-52204E616E6F)	Get the current device configuration	ReturnCurrentScanConfigurationDataReceiver (UUID: 0x43484141-444C-5020-4E49-52204E616E6F)	Get the byte array of current configuration of the device
Compareconfig(intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_CURRENT_CONFIG_DATA))	Compare whether the configuration set by the device and the configuration set by the user in the quick set are the same	X	X
PerformScan(long delaytime)	Set delay time to scan sample to avoid BLE hang.	X	X
ISCNIRScanSDK.StartScan() (UUID: 0x4348411D-444C-5020-4E49-52204E616E6F	Trigger the scan event	ScanDataReadyReceiver (UUID: 0x43484128-444C-5020-4E49-52204E616E6F)	Get the scan result

0x43484127-444C-5020-4E49-52204E616E6F)			
ISCNIRScanSDK.GetDeviceStatus() (UUID: 0x00002A19-0000-1000-8000-00805F9B34FB 0x43484101-444C-5020-4E49-52204E616E6F 0x43484102-444C-5020-4E49-52204E616E6F 0x43484109-444C-5020-4E49-52204E616E6F 0x43484103-444C-5020-4E49-52204E616E6F 0x43484104-444C-5020-4E49-52204E616E6F)	Request to get the status of the device	GetDeviceStatusReceiver	Get the device status
ISCNIRScanSDK.GetScanLampRampUpADC() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ramp up ADC	ReturnLampRampUpADCReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ramp up ADC
ISCNIRScanSDK.GetLampADCAverage() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ADC among repeated times	ReturnLampADCAverageReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ADC among repeated times
writeCSV(Scan_Spectrum_Data)	Save the scan report	X	X

Perform Scan – Manual

ScanViewActivity.java


Start
1. Turn on the manual mode
2-(a)-(d) Manual Setting
3. Perform scan
4. Get the status of the device
Check TIVA and HW Version
TIVA≧ 2.1.0.67 and main board = F (standard wavelength device) or TIVA≧ 3.3.0 and main board = O (extended wavelength device)
Main board is not F and O
5. Get lamp ramp up ADC
6. Get lamp ADC among repeated times
7. Save scan report
Continuous Scan ?
Yes
No
End

1. Turn on the manual mode will support to set lamp on or off, scan PGA and scan repeats.
2. Manual Setting

(a) Call the **ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.ON)** to turn on the lamp. Register **ReturnSetLampReceiver** to know the set is completed.
(b) Call the **ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.OFF)** to turn off the lamp. Register **ReturnSetLampReceiver** to know the set is completed.
(c) Call the **ISCNIRScanSDK.SetPGA(pga)** to set scan PGA. Register **ReturnSetPGAReceiver** to know the set is completed.
(d) Call the **ISCNIRScanSDK.SetScanRepeat(scan_repeat)** to set scan repeats. Register **ReturnSetScanRepeatsReceiver** to know the set is completed.

3. Call the **PerformScan(long delaytime)** to scan the sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()** to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan result.
4. Call the **ISCNIRScanSDK.GetDeviceStatus()** to request to get the status of the device including battery capacity, system temperature, humidity, total lamp time, the byte of the device status and the byte of the error status. Register **GetDeviceStatusReceiver** to get the device status.
5. Call the **ISCNIRScanSDK.GetScanLampRampUpADC()** to request to get lamp ramp up ADC. Register **ReturnLampRampUpADCReceiver** to get lamp ramp up ADC.
6. Call the **ISCNIRScanSDK.GetLampADCAverage()** to request to get lamp ADC among repeated times. Register **ReturnLampADCAverageReceiver** to get lamp ADC among repeated times.
7. Call the **writeCSV(Scan_Spectrum_Data)** to save the scan report.

Close the manual mode as follows or reference to ChangeLampState():

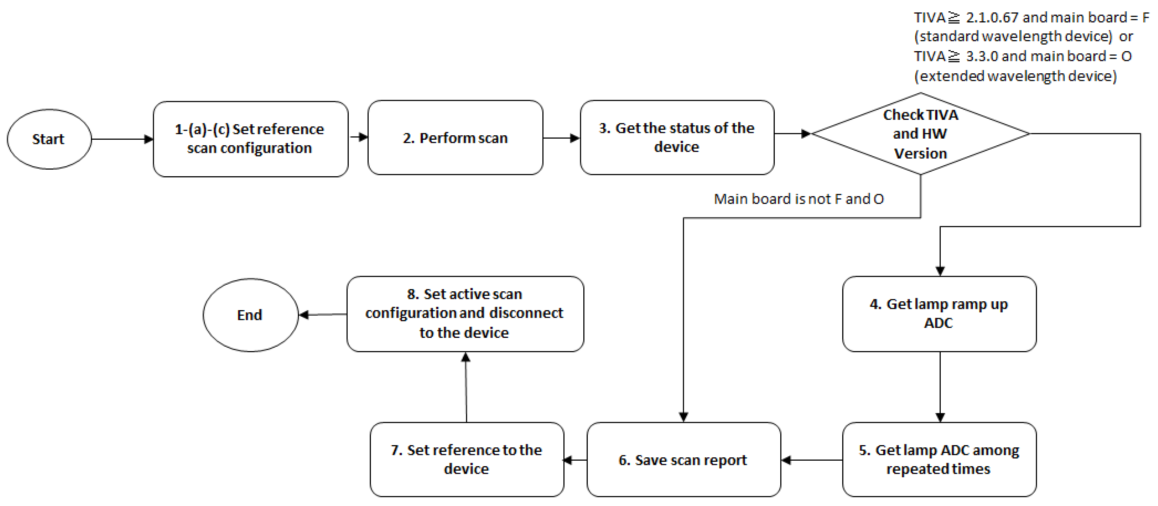
1. If the lamp is turned on, should close the lamp(**ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.OFF)**).
2. Change the lamp state to auto(**ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.AUTO)**).

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.ON) (UUID: 0x43484144-444C-5020-4E49-52204E616E6F)	Turn on the lamp	ReturnSetLampReceiver	Know the set is completed
ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.OFF) (UUID: 0x43484144-444C-5020-4E49-52204E616E6F)	Turn off the lamp	ReturnSetLampReceiver	Know the set is completed
ISCNIRScanSDK.SetPGA(pga) (UUID: 0x43484146-444C-5020-4E49-52204E616E6F)	Set scan PGA	ReturnSetPGAReceiver	Know the set is completed
ISCNIRScanSDK.SetScanRepeat(scan_repeat) (UUID: 0x43484147-444C-5020-4E49-52204E616E6F)	Set scan repeats	ReturnSetScanRepeatsReceiver	Know the set is completed
PerformScan(long delaytime)	Set delay time to scan sample to avoid BLE hang.	X	X
ISCNIRScanSDK.StartScan() (UUID: 0x4348411D-444C-5020-4E49-52204E616E6F 0x43484127-444C-5020-4E49-52204E616E6F)	Trigger the scan event	ScanDataReadyReceiver (UUID: 0x43484128-444C-5020-4E49-52204E616E6F)	Get the scan result
ISCNIRScanSDK.GetDeviceStatus() (UUID: 0x00002A19-0000-1000-8000-00805F9B34FB 0x43484101-444C-5020-4E49-52204E616E6F 0x43484102-444C-5020-4E49-52204E616E6F	Request to get the status of the device	GetDeviceStatusReceiver	Get the device status

0x43484109-444C-5020-4E49-52204E616E6F 0x43484103-444C-5020-4E49-52204E616E6F 0x43484104-444C-5020-4E49-52204E616E6F)			
ISCNIRScanSDK.GetScanLampRampUpADC() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ramp up ADC	ReturnLampRampUpADCReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ramp up ADC
ISCNIRScanSDK.GetLampADCAverage() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ADC among repeated times	ReturnLampADCAverageReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ADC among repeated times
writeCSV(Scan_Spectrum_Data)	Save the scan report	X	X
ChangeLampState()	Leaving the manual scan page to restore the lamp state to auto	X	X

Update Reference to the Device

ScanViewActivity.java


TIVA≧ 2.1.0.67 and main board = F (standard wavelength device) or TIVA≧ 3.3.0 and main board = O (extended wavelength device)
Start
1-(a)-(c) Set reference scan configuration
2. Perform scan
3. Get the status of the device
Check TIVA and HW Version
Main board is not F and O
4. Get lamp ramp up ADC
5. Get lamp ADC among repeated times
6. Save scan report
7. Set reference to the device
8. Set active scan configuration and disconnect to the device
End

1. Set reference scan configuration.

   (a) Call the **ISCNIRScanSDK.SetReferenceParameter(int MINWAV,int MAXWAV)** to set the reference configuration to the device. Register **WriteScanConfigStatusReceiver** to get back and check whether the set scan configuration in the device memory is valid.

   Note: In our APP, Need to set the reference set flag to Compareconfig(byte EXTRA_DATA[]) to determine that the reference config needs to be compared now.

   ```
   reference_set_config = true;
   ```

(b) Call the **ISCNIRScanSDK.ReadCurrentScanConfig()** to get the current device configuration. Register **ReturnCurrentScanConfigurationDataReceiver** to get the byte array of current configuration of the device.

(c) Call the **Compareconfig(intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_CURRENT_CONFIG_DATA))** to compare whether the configuration set by the device and the configuration set by the reference are the same.

2. Call the **PerformScan(long delaytime)** to scan the reference sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()**to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan results.
3. Call the **ISCNIRScanSDK.GetDeviceStatus()** to request to get the status of the device including battery capacity, system temperature, humidity, total lamp time, the byte of the device status and the byte of the error status. Register **GetDeviceStatusReceiver** to get the device status.
4. Call the **ISCNIRScanSDK.GetScanLampRampUpADC()** to request to get lamp ramp up ADC. Register **ReturnLampRampUpADCReceiver** to get lamp ramp up ADC.
5. Call the **ISCNIRScanSDK.GetLampADCAverage()** to request to get lamp ADC among repeated times. Register **ReturnLampADCAverageReceiver** to get lamp ADC among repeated times.
6. Call the **writeCSV(Scan_Spectrum_Data)** to save the scan report. In our APP, will change configuration name to Reference.
7. Call the **ISCNIRScanSDK.SaveReference()** to set the reference to the device.

Note: In our APP, will pop out the confirm dialog and should set reference flag.

saveReference = true;

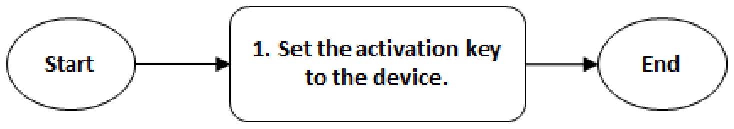
8. After finish saving the reference to the device, call the **ISCNIRScanSDK.ScanConfig(ActiveConfigByte,ISCNIRScanSDK.ScanConfig.SET)** to set active scan configuration to the device and disconnect.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.SetReferenceParameter(int MINWAV,int MAXWAV) (UUID: 0x43484142-444C-5020-4E49-52204E616E6F)	Set the reference configuration to the device	WriteScanConfigStatusReceiver (UUID: 0x43484142-444C-5020-4E49-52204E616E6F)	Get back and check whether the set scan configuration in the device memory is valid
ISCNIRScanSDK.ReadCurrentScanConfig() (UUID: 0x43484140-444C-5020-4E49-52204E616E6F)	Get the current device configuration	ReturnCurrentScanConfigurationData Receiver (UUID: 0x43484141-444C-5020-4E49-52204E616E6F)	Get the byte array of current configuration of the device
Compareconfig(intent.getByteArrayExtra(ISCNIRScanSDK.EXTRA_CURRENT_CONFIG_DATA))	Compare whether the configuration set by the device and the configuration set by the reference are the same	X	X
PerformScan(long delaytime)	Set delay time to scan sample to avoid BLE hang.	X	X

ISCNIRScanSDK.StartScan() (UUID: 0x4348411D-444C-5020-4E49-52204E616E6F 0x43484127-444C-5020-4E49-52204E616E6F)	Trigger the scan event	ScanDataReadyReceiver (UUID: 0x43484128-444C-5020-4E49-52204E616E6F)	Get the scan result
ISCNIRScanSDK.GetDeviceStatus() (UUID: 0x00002A19-0000-1000-8000-00805F9B34FB 0x43484101-444C-5020-4E49-52204E616E6F 0x43484102-444C-5020-4E49-52204E616E6F 0x43484109-444C-5020-4E49-52204E616E6F 0x43484103-444C-5020-4E49-52204E616E6F 0x43484104-444C-5020-4E49-52204E616E6F)	Request to get the status of the device	GetDeviceStatusReceiver	Get the device status
ISCNIRScanSDK.GetScanLampRampUpADC() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ramp up ADC	ReturnLampRampUpADCReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ramp up ADC
ISCNIRScanSDK.GetLampADCAverage() (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Request to get lamp ADC among repeated times	ReturnLampADCAverageReceiver (UUID: 0x4348410B-444C-5020-4E49-52204E616E6F)	Get lamp ADC among repeated times
writeCSV(Scan_Spectrum_Data)	Save the scan report	X	X
ISCNIRScanSDK.SaveReference() (UUID: 0x43484132-444C-5020-4E49-52204E616E6F)	Set the reference to the device	X	X
ISCNIRScanSDK.ScanConfig(ActiveConfigByte,ISCNIRScanSDK.ScanConfig.SET) (UUID: 0x43484142-444C-5020-4E49-52204E616E6F)	Set active scan configuration to the device	X	X

Activation Key Setting

ActivationViewActivity.java

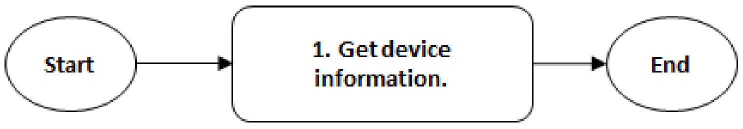


1. Call the **ISCNIRScanSDK.SetLicenseKey(data)** to set the activation key**(key length is 24)**. Register **ReturnActivateStatusReceiver** to check whether the activation key is set success.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.SetLicenseKey(data) (UUID: 0x43484130-444C-5020-4E49-52204E616E6F)	Set the activation key	ReturnActivateStatusReceiver (UUID: 0x43484131-444C-5020-4E49-52204E616E6F)	Check whether the activation key is set success

Device Information

DeviceInfoViewActivity.java



1. Call the **ISCNIRScanSDK.GetDeviceInfo()** to request to get the device information including the device of manufacturer, model name, serial number, HW version and TIVA version. Register **DeviceInfoReceiver** to get the device information.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetDeviceInfo() (UUID: 0x00002A29-0000-1000-8000-00805F9B34FB 0x00002A24-0000-1000-8000-00805F9B34FB 0x00002A25-0000-1000-8000-00805F9B34FB 0x00002A27-0000-1000-8000-00805F9B34FB 0x00002A26-0000-1000-8000-00805F9B34FB 0x00002A28-0000-1000-8000-00805F9B34FB)	Request to get the device information	DeviceInfoReceiver	Get the device information

Device Status

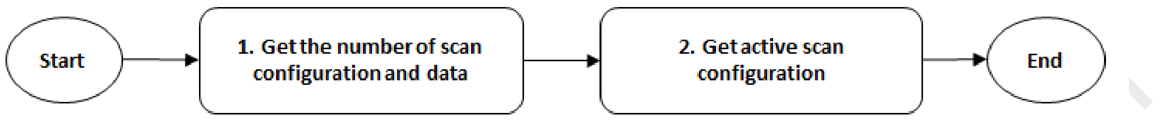
DeviceStatusViewActivity.java

1. Call the **ISCNIRScanSDK.GetDeviceStatus()** to request to get the status of the device including battery capacity, system temperature, humidity, total lamp time, the byte of the device status and the byte of the error status. Register **mStatusReceiver** to get the device status.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetDeviceStatus() (UUID: 0x00002A19-0000-1000-8000-00805F9B34FB 0x43484101-444C-5020-4E49-52204E616E6F 0x43484102-444C-5020-4E49-52204E616E6F 0x43484109-444C-5020-4E49-52204E616E6F 0x43484103-444C-5020-4E49-52204E616E6F 0x43484104-444C-5020-4E49-52204E616E6F)	Request to get the status of the device	mStatusReceiver	Get the device status

List Scan Configuration

ScanConfigurationsViewActivity.java

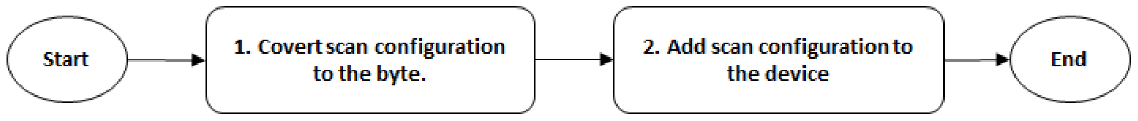

Start
1. Get the number of scan configuration and data
2. Get active scan configuration
End

1. Call the **ISCNIRScanSDK.GetScanConfig()** to request to get the number of scan configuration and scan configuration data. Register **ScanConfSizeReceiver** to get the number of scan configuration. Register **ScanConfReceiver** to get scan configuration data.
2. Call the **ISCNIRScanSDK.GetActiveConfig()** to request to get the active configuration index in the device. Register **GetActiveScanConfReceiver** to get active configuration index.

API	Process	Receiver	Receiver Description
ISCNIRScanSDK.GetScanConfig() (UUID: 0x43484113-444C-5020-4E49-52204E616E6F 0x43484114-444C-5020-4E49-52204E616E6F)	Get the number of scan configuration->Get scan configuration data	ScanConfSizeReceiver	Get the number of scan configuration
		ScanConfReceiver (UUID: 0x43484115-444C-5020-4E49-52204E616E6F)	Get scan configuration data
ISCNIRScanSDK.GetActiveConfig() (UUID: 0x43484118-444C-5020-4E49-52204E616E6F)	Request to get the active configuration index	GetActiveScanConfReceiver (UUID: 0x43484118-444C-5020-4E49-52204E616E6F)	Get active configuration index

Add Scan Configuration to the Device

AddScanConfigViewActivity.java


Start
1. Covert scan configuration to the byte.
2. Add scan configuration to the device
End

1. Call the **ChangeScanConfigToByte()** and write UI settings to **ISCNIRScanSDK.WriteScanConfiguration(write_scan_config)**. It will return the byte array of the scan configuration (return **EXTRA_DATA**).

2. Call the **ISCNIRScanSDK.ScanConfig(EXTRA_DATA,ISCNIRScanSDK.ScanConfig.SAVE)** to set the configuration to the device. Register **WriteScanConfigStatusReceiver** to check whether the setting is successful.

API	Process	Receiver	Receiver Description
ChangeScanConfigToByte()	Transmit scan configuration to the byte that user set.	X	X
ISCNIRScanSDK.ScanConfig(EXTRA_DATA,ISCNIRScanSDK.ScanConfig.SAVE) (UUID: 0x43484142-444C-5020-4E49-52204E616E6F)	Set the configuration to the device	WriteScanConfigStatusReceiver (UUID: 0x43484143-444C-5020-4E49-52204E616E6F)	Check whether the setting is successful

Warm-Up the Device

HomeViewActivity.java

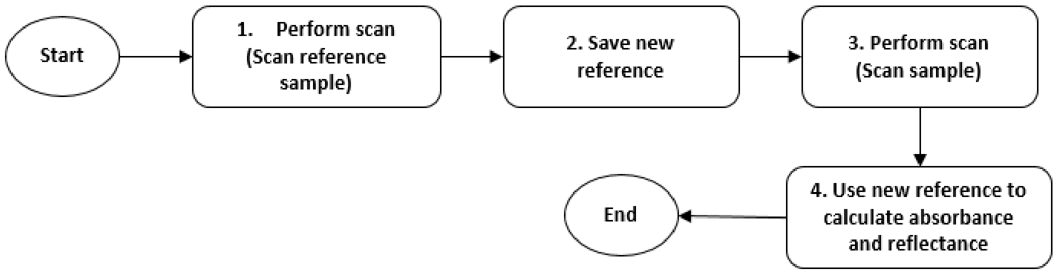
When the user clicks on the connection, the Intent will be used to pass the flag whether the user wants to warm up the device. See **newscanhIntent.putExtra("warmup",switch_Warmup.isChecked());**. After clicks on the connection, see the section of Connected to the Device 1-(b) . It will teach how to open the lamp to do the warm up the device.

ScanViewActivity.java

Warm-up time reaches three minutes, you can directly switch pages or scan, and the device will automatically turn off the lamp. See **ChangeLampState()**. If device have do the warm up, will call **ISCNIRScanSDK.ControlLamp(ISCNIRScanSDK.LampState.AUTO)** to close the lamp.

Warning: When the user sets connect with lamp on, must pay attention to the warm-up time and confirm that the device is warmed up and turn off the lamp before leaving.

Calculate AU from a New Reference Scan


Start
1. Perform scan (Scan reference sample)
2. Save new reference
3. Perform scan (Scan sample)
4. Use new reference to calculate absorbance and reflectance
End

1. Place the **reference sample** and call the **PerformScan (long delaytime)** to scan the reference sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()** to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan result.

2. The acquired spectral data(intensity) needs to be temporarily stored in **ReferenceIntensity**.

   For example :

```
public static List<Integer> ReferenceIntensity =   new ArrayList<Integer>();

ReferenceIntensity.clear();
Scan_Spectrum_Data = new
ISCNIRScanSDK.ScanResults(Interpret_wavelength,Interpret_intensity,Interpret_uncalibratedIntens
ity,Interpret_length);

for (index = 0; index < Scan_Spectrum_Data.getLength(); index++) {
      ReferenceIntensity.add(Scan_Spectrum_Data.getUncalibratedIntensity()[index]);
}
```

3. Place the **sample** and call the **PerformScan(long delaytime)** to scan the sample. The delay time is set to 300ms to avoid the BLE hang. In **PerformScan(long delaytime)**, call the **ISCNIRScanSDK.StartScan()** to trigger the scan event. You should register **ScanDataReadyReceiver** to get the scan result.

4. Use the stored **new reference sample signal intensity and sample signal intensity** to calculate absorbance and reflectance of the sample.

   For example :

```
Scan_Spectrum_Data = new
ISCNIRScanSDK.ScanResults(Interpret_wavelength,Interpret_intensity,Interpret_uncalibratedIntensit
y,Interpret_length);

mXValues.clear();
mIntensityFloat.clear();
mAbsorbanceFloat.clear();
mReflectanceFloat.clear();
mWavelengthFloat.clear();
mReferenceFloat.clear();
int index;
for (index = 0; index < Scan_Spectrum_Data.getLength(); index++) {
  mXValues.add(String.format("%.02f", ISCNIRScanSDK.ScanResults.getSpatialFreq(mContext,
  Scan_Spectrum_Data.getWavelength()[index])));

  mIntensityFloat.add(new Entry((float) Scan_Spectrum_Data.getWavelength()[index],(float)
  Scan_Spectrum_Data.getUncalibratedIntensity()[index]));

  mAbsorbanceFloat.add(new Entry((float) Scan_Spectrum_Data.getWavelength()[index],(-1) *
  (float) Math.log10((double) Scan_Spectrum_Data.getUncalibratedIntensity()[index] / (double)
  ReferenceIntensity.get(index))));

  mReflectanceFloat.add(new Entry((float) Scan_Spectrum_Data.getWavelength()[index],(float)
  Scan_Spectrum_Data.getUncalibratedIntensity()[index] / (float) ReferenceIntensity.get(index)));

  mWavelengthFloat.add((float) Scan_Spectrum_Data.getWavelength()[index]);

  mReferenceFloat.add(new Entry((float) Scan_Spectrum_Data.getWavelength()[index],(float)
  ReferenceIntensity.get(index)));

}
```

Note: The ScanDataReadyReceiver needs to judge whether the scan is to scan the reference sample or the sample to decide whether to go step2 or step4. In addition, the ReferenceIntensity can also be saved to the local file, and the local file can be read and loaded into the ReferenceIntensity before entering the sample scanning. In this demonstration, it should be noted that the scan configuration (including PGA) of the scanned reference sample must be the same as the scan configuration of the current scanned sample.