Freescale Semiconductor
User's Guide

Document Number: WFDSINKAPI
Rev. L5.0.0_1.0.0-ga, 06/2015

i.MX Android™ Wi-Fi Display Sink API Introduction

Contents

1 Overview....1
2 Requirements....2
3 API Description....2
4 Examples....3

1 Overview

i.MX Android™ L5.0.0_1.0.0-ga release includes support for Wi-Fi Display Sink, which allows a device to act as Wi-Fi Display Sink to render the audio and video content from Wi-Fi Source Device. The application framework provides access to the Wi-Fi Display Sink functionality through the i.MX Android Wi-Fi Display Sink APIs. These APIs let applications configure the device as a Wi-Fi Display Sink device, which can be connected with Wi-Fi Display Source, and render the RTSP streaming from Wi-Fi Display Source.

Using the i.MX Android Wi-Fi Display Sink APIs, an Android application can perform the following:

- Scan for other Wi-Fi Display Source devices.
- Get self peer name and MAC address.
- Set the device name which is displayed in other Wi-Fi Display devices.
- Set the display surface to render the RTSP streaming from the Wi-Fi Display Source Device.
- Start or stop the RTSP streaming between Wi-Fi Display Source and Sink.
- Disconnect Wi-Fi display connection from P2P layer.
- Send the input event to the Wi-Fi Display Source device, which known as UIBC.

2 Requirements

- i.MX Android L5.0.0_1.0.0-ga release
- Realtek 8723AS Wi-Fi SDIO module
- Atheros AR6233 Wi-Fi SDIO module

3 API Description

Below is a description of the API exported by com.fsl.wfd.WfdSink.

3.1 Constants

The following table lists the constants.

Constant string name	Constant value
WFD_DEVICE_LIST_CHANGED_ACTION	com.fsl.wfd.DEVICE_LIST_CHANGED
WFD_DEVICE_CONNECTED_ACTION	com.fsl.wfd.DEVICE_CONNECTED
WFD_THIS_DEVICE_UPDATE_ACTION	com.fsl.wfd.DEVICE_THIS_UPDATE
EXTRA_DEVICE_LIST	deviceList
EXTRA_CONNECTION_CHANGED	connectionChanged

3.2 Methods

The following table lists the methods.

Method	Description	Parameter
void startSearch()	This function launches the Wi-Fi P2P scanning. After the function is called, the Wi-Fi P2P peer discovery starts. When Wi-Fi P2P peers are discovered, it sends the WFD_DEVICE_LIST_CHANGED_ACTION broadcast to deliver the peers' information.	-
void setDeviceName(String name)	This function sets the Wi-Fi P2P device name. Call this function to rename the device.	name: the name used to identify a Wi-Fi display device.
String getDeviceName()	This function retrieves the device name.	-
String getDeviceMacAddress()	This function gets the MAC address.	-
void stopSearch()	This function stops the Wi-Fi P2P from scanning peers. Call this function to quit the WFD sink function after the Wi-Fi P2P connection is interrupted.	-
void startRtsp(Surface surface)	This function starts the RTSP streaming when the Wi-Fi P2P network is connected.	surface: it represents the display area.

Table continues on the next page...

Method	Description	Parameter
void stopRtsp()	This function stops RTSP streaming.	-
void stopRtp()	This function stops RTP sessions.	-
void disconnect()	This function stops the P2P connection.	-
void sendKeyEvent(int KeyCode, KeyEvent event)	This function sends the key event through UIBC. If Sink and Source negotiate to support UIBC, call this interface to send the key event. Get this event from the key action listener of the activity.	-
void sendTouchEvent(MotionEvent event)	This function sends the touch event through UIBC. If Sink and Source negotiate to support UIBC, call this interface to send the touch event. Get this event from the activity's view.	-
void setMiracastMode(int mode)	This function sets the role of the Wi-Fi Display. Set WifiP2pManager.MIRACAST_SINK when the connection is established. Set WifiP2pManager.MIRACAST_DISABLED when the connection is no longer available.	mode: WifiP2pManager.MIRACAST_SINK the connection is standalone and is not disturbed by other P2P device. WifiP2pManager.MIRACAST_DISABLED the P2P connection is normal.
void sendScrollEvent(MotionEvent event)	This function sends the MotionEvent of the mouse wheel from the onGenericMotionEvent(MotionEvent).	-

4 Examples

A demo application is provided in myandroid/package/apps/fsl_imx_demo/WfdSink to show how to use the WFD Sink APIs.

When writing Android.mk, add these two variables:

- Include fsl.imx as static Java library.
- Disable the ProGuard function.

```
LOCAL_STATIC_JAVA_LIBRARIES := fsl.imx
LOCAL_PROGUARD_ENABLED := disabled
```

4.1 Registering a broadcast receiver

Before the Wi-Fi Display Sink can function properly, the WFD_DEVICE_LIST_CHANGED_ACTION, WFD_THIS_DEVICE_UPDATE_ACTION, and WFD_DEVICE_CONNECTED_ACTION broadcasts should be registered to receive the Wi-Fi P2P network state.

This example show how to create a broadcast receiver:

```
private final BroadcastReceiver mWifiP2pReceiver = new BroadcastReceiver()
{
    public void onReceive(Context context, Intent intent) {
          String action = intent.getAction();
```

```
          if (WfdSink.WFD_DEVICE_LIST_CHANGED_ACTION.equals(action)) {
              WifiP2pDevice[] devices;
              Parcelable[] devs;
              devs = intent.getParcelableArrayExtra(WfdSink.EXTRA_DEVICE_LIST);
              if (devs == null || devs.length == 0) {
                  return;
              }

              devices = new WifiP2pDevice[devs.length];
              for (int i=0; i<devs.length; i++) {
                   devices[i] = (WifiP2pDevice)devs[i];
              }

              mSourcePeers.clear();
              for (WifiP2pDevice device : devices) {
                   mSourcePeers.add(device.deviceName);
              }

           }
           if (WfdSink.WFD_DEVICE_CONNECTED_ACTION.equals(action)) {
               boolean connected =
               intent.getBooleanExtra(WfdSink.EXTRA_CONNECTION_CHANGE
               handleConnection(connected);
           }
           if (WfdSink.WFD_THIS_DEVICE_UPDATE_ACTION.equals(action)) {
               currentdevice = (TextView)findViewById(R.id.currentdevice);
               if (mThisName != mWfdSink.getDeviceName())
               mThisName = mWfdSink.getDeviceName();
               currentdevice.setText("SSID:" + mWfdSink.getDeviceName());
            }
       }
 };
```

This example show how to register the broadcast receiver:

```
IntentFilter intentFilter = new IntentFilter();
intentFilter.addAction(WfdSink.WFD_DEVICE_LIST_CHANGED_ACTION);
intentFilter.addAction(WfdSink.WFD_DEVICE_CONNECTED_ACTION);
intentFilter.addAction(WfdSink.WFD_THIS_DEVICE_UPDATE_ACTION);
registerReceiver(mWifiP2pReceiver, intentFilter);
```

4.2 Enabling discoverability

The API startSearch() of WfdSink should be called to start scanning Wi-Fi display source devices.

4.3 Connecting devices

The Wi-Fi P2P network broadcasts is received. The Wi-Fi P2P source device list can be obtained when WFD_DEVICE_LIST_CHANGED_ACTION is received.

The startRtsp() and stopRtsp() functions of WfdSink should be called according to WFD_DEVICE_CONNECTED_ACTION.

To ensure that the P2P Stack is exclusively used by the Wi-Fi Display Sink Application, use setMiracastMode to set the role of WifiDisplay.

To manage the Wi-Fi P2P network connection state:

```
private handleConnection(boolean connected) {
      if (connected) {
           mWfdSink.startRtsp(mSurfaceHolder.getSurface());
```

```
            mWfdSink.setMiracastMode(WifiP2pManager.MIRACAST_SINK);
        }
        else {
            mWfdSink.stopRtsp();
            mWfdSink.setMiracastMode(WifiP2pManager.MIRACAST_DISABLED);
        }
}
```

4.4 Ending Wi-Fi Display Sink

Ending Wi-Fi display sink includes two use cases: passive ending and active ending.

- Passive ending is initiated by the source side. In this instance, sink only needs to call stopRtsp() after receiving group removing intent.
- Active ending is initiated by the sink side. For example, if you press “back” or ”home” softkey, sink ends Wi-Fi display actively. In this instance, you should call disconnect() except stopRtsp(). The former removes group connection and the later ends the upper RTSP connection.

NOTE

The default Android WFD Source requires the RTSP streaming to start within 15 s after the Wi-Fi P2P connection is built. The startRtsp() method should be called immediately after receiving the broadcast WFD_DEVICE_CONNECTED_ACTION with EXTRA_CONNECTION_CHANGED to be true.

4.5 How to use sendKeyEvent, sendScrollEvent, and sendTouchEvent

The API sendKeyEvent , sendScrollEvent, and sendTouchEvent have one event parameter. These events should be obtained from the upper activity. For example:

```
    @Override
    public boolean onKeyDown(int keyCode, KeyEvent event) {
        if(keyCode == KeyEvent.KEYCODE_BACK) {
            mHandler.removeMessages(SET_UI_FLAG_HIDE_NAVIGATION);
            this.exitDialog();
        } else if (keyCode == KeyEvent.KEYCODE_VOLUME_DOWN
                || keyCode == KeyEvent.KEYCODE_VOLUME_UP) {
            mWfdSink.sendKeyEvent(keyCode, event);
            return true;
        }
        return super.onKeyDown(keyCode, event);
    }

       mSurfaceView = (SinkView)findViewById(R.id.sink_preview);
       SurfaceHolder holder = mSurfaceView.getHolder();
       holder.addCallback(this);
       holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
       mSurfaceView.setOnTouchListener(new OnTouchListener() {
               @Override
               public boolean onTouch(View v, MotionEvent event) {
                   Log.d(TAG, "sink view on touch event x=" + event.getX() + "y=" +
event.getY());
                   mWfdSink.sendTouchEvent(event);
                   return true;
               }
       });
    @Override
       public boolean onGenericMotionEvent(MotionEvent event) {
```

```
            if (event.isFromSource(InputDevice.SOURCE_MOUSE)) {
                switch (event.getAction()) {
                    case MotionEvent.ACTION_SCROLL:
                        mWfdSink.sendScrollEvent(event);
                        break;
                    case MotionEvent.ACTION_HOVER_MOVE:
                        mWfdSink.sendTouchEvent(event);
                        Log.i(TAG,"sendrolon GenerMotionEvent HoverMoved." +
event.toString());
                        break;
                }
            }
            return super.onGenericMotionEvent(event);
        }
```

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: WFDSINKAPI
Rev. L5.0.0_1.0.0-ga
06/2015