

CC31xx Connection Policy

Overview

This sample application demonstrates how to use/enable the various connection policies of CC3100 device.

WLAN profiles are designed to allow connections to pre-defined stored access-points. A profile provides the information required to connect to a given AP. This information includes the SSID, security-type and security-keys of an AP. Each profile refers to a certain AP. The profiles are stored on the non-volatile memory, and therefore are preserved during device reset.

The following APIs are available for handling profiles:

- "sl_WlanProfileAdd" - Used for adding a new profile. SSID and security info shall be provided, where returned value refers to the stored index (out of the 7 available)
- "sl_WlanProfileDel" - Used for deleting a certain stored profile, or for deleting all at once. Index shall be the input parameter
- "sl_WlanProfileGet" - Used for retrieving information of certain stored profile. Index shall be the input parameter

WLAN connection policy defines four options to connect the SimpleLink device to a given AP:

- **Auto:** The device attempts to connect to an AP from the stored profiles based on priority. Up-to 7 profiles are supported and upon a connection attempt, the device selects the highest priority profile. If several profiles are within the same priority, decision will be made based on the security type (WPA2 -> WPA -> OPEN). If the security-type is also the same, selection will be based on received signal's strength.
- **Fast:** The device attempts to connect to the last connected AP. In this mode "probe request" is not transmitted prior to "authentication request", as both the SSID and channel are known.
- **Auto SmartConfig:** If this policy is set, CC3100 will automatically start the SmartConfig process after booting-up, if no command is received by host within 2 seconds.
- **Any P2P:** This policy can be used to connect to any visible WiFi-Direct device in P2P mode - When configured in this mode, CC3100 will perform discovery and connect to the first found device using security type 'SL_SEC_TYPE_P2P_PBC'. Refer **P2P** sample application for more details

The default connection policy of CC3100 device is **Auto** and **Auto Smartconfig**.

The configuration is performed by using the following API: "sl_WlanPolicySet". For additional information about these APIs, please refer to the doxygen API manual in the latest SDK release.

```

/* Reset policy settings */
sl_WlanPolicySet(SL_POLICY_CONNECTION , SL_CONNECTION_POLICY(0,0,0,0,0), 0, 0);

/* Enable auto connect (connection to stored profiles according to priority) */
/* Connection should first be established to the higher (secured) profile */
sl_WlanPolicySet(SL_POLICY_CONNECTION , SL_CONNECTION_POLICY(1,0,0,0,0), 0, 0);

/* Enable fast connect (connection to last connected AP) */
sl_WlanPolicySet(SL_POLICY_CONNECTION , SL_CONNECTION_POLICY(1,1,0,0,0), 0, 0);

/* Enable Auto Smartconfig policy */
sl_WlanPolicySet(SL_POLICY_CONNECTION , SL_CONNECTION_POLICY(0,0,0,0,1), 0, 0);

```

[Return to CC31xx & CC32xx Home Page](#)

[Return to CC31xx Sample Applications](#)

Application details

This application initializes CC3100 device and adds two profiles, one, a unsecured AP profile with priority 6 and the other WPA-2 secured AP profile with priority 7. It then enables 'Auto' connect policy. The device shall connect to the available AP of highest priority (in this case, **Secured**) profile.

Application deletes all the profiles and sets the policy to 'Auto SmartConfig' and restarts the device. Upon reset, the device waits for 2 seconds for command(s) from Host-MCU, and if no commands are received, it enters **SmartConfig** mode.

Application deletes all the profiles, connects to the open AP, set the connection policy to 'Fast Connect' and restarts the device. Upon reset, the device will connected to the last connected AP.

Note: This wiki page is only applicable for **CC3100-SDK v1.0.0** and upward releases. For documentation on older SDKs' examples, refer corresponding file in `<cc3100-sdk-installation-location>\cc3100-sdk\docs\examples\`

Source Files briefly explained

i. main - Initializes the device, adds profiles and sets connection policies

Usage

- Connect the board to a Windows-PC and configure the terminal-program for seeing the logs - [[CC31xx_&_CC32xx_Terminal_Setting_Wiki](#)^[1]] has detailed instructions for configuring the terminal-program
- Open **main.c** and change **UNSEC_SSID_NAME**, **SEC_SSID_NAME** and **SEC_SSID_KEY** as per your access-points properties
- Build and run the application using CCS
- See the self explanatory logs on the terminal-program's console.

```
The application adds both secured (priority 7) and unsecured AP
profiles sockets(priority 6) and sets the connection policy to Auto
The device tries to connect to AP - Upon connection attempt, it
selects the highest priority profile. If several profiles are within
the same priority, decision will be made based on security type (WPA2
-> WPA -> OPEN)
The application then connects to the open AP, enables FAST policy
and restarts the device
On reset, the device tries to establish connection w/ the last
connected AP, which in our case is the open AP
The application sets the 'Auto SmartConfig' policy and resets the
device
```

- Launch the SmartConfig application on a Smartphone and complete the provisioning process
- On success, below message will be displayed on the terminal

Note: References to **restart** in above sections mean, putting & waking-up the device in & from hibernate respectively.

Limitations/Known Issues

- None

References

[1] http://processors.wiki.ti.com/index.php/CC31xx_&_CC32xx_Terminal_Setting

Article Sources and Contributors

CC31xx Connection Policy *Source:* <http://processors.wiki.ti.com/index.php?oldid=194463> *Contributors:* A0131814, A0132173, A0221015, Codycooke, Malokyle

Image Sources, Licenses and Contributors

File:Cc31xx cc32xx return home.png *Source:* http://processors.wiki.ti.com/index.php?title=File:Cc31xx_cc32xx_return_home.png *License:* unknown *Contributors:* A0221015

File:Cc31xx return sample apps.png *Source:* http://processors.wiki.ti.com/index.php?title=File:Cc31xx_return_sample_apps.png *License:* unknown *Contributors:* A0221015