CC31xx P2P Application

Overview

This is a sample application demonstrating how CC3100 can be connected to a P2P device. This application starts a TCP server on port 5001 and waits for P2P device to connect and send data on it.

Different P2P roles supported by CC3100 are:

- **SL_P2P_ROLE_GROUP_OWNER**: CC3100 will be configured in 'Group-Owner' mode
- **SL_P2P_ROLE_CLIENT**: CC3100 will be configured in 'Client' mode
- **SL_P2P_ROLE_NEGOTIATE**: CC3100 will negotiate with remote device for 'Client/Group-Owner' role.

CC3100 can be configure in below modes to initiate negotiation:

- **SL_P2P_NEG_INITIATOR_ACTIVE**: CC3100 will perform discovery - Once the remote device is found, it sends the negotiation request immediately
- **SL_P2P_NEG_INITIATOR_PASSIVE**: CC3100 will perform discovery - Once the remote device is found, CC3100 waits for it to start negotiation
- **SL_P2P_NEG_INITIATOR_RAND_BACKOFF**: CC3100 will perform discovery - Once the remote device is found, it triggers a random timer (1-7 seconds) and waits for the remote device to negotiate. On timer expiry, CC3100 starts negotiation itself

Supported security types used during P2P negotiation are:

- **SL_SEC_TYPE_P2P_PBC**
- **SL_SEC_TYPE_P2P_PIN_DISPLAY**
- **SL_SEC_TYPE_P2P_PIN_KEYPAD**

CC3100 can be configured in **any_p2p** mode as well - When configured, CC3100 will perform discovery and connect to the first found device using security type **SL_SEC_TYPE_P2P_PBC**

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**

Application details

This application configures the device with following settings:

- P2P role negotiate (SL_P2P_ROLE_NEGOTIATE)
- P2P negotiation initiation active (SL_P2P_NEG_INITIATOR_ACTIVE)
- P2P device listens on channel 11 and P2P device's operation channel is set to 6
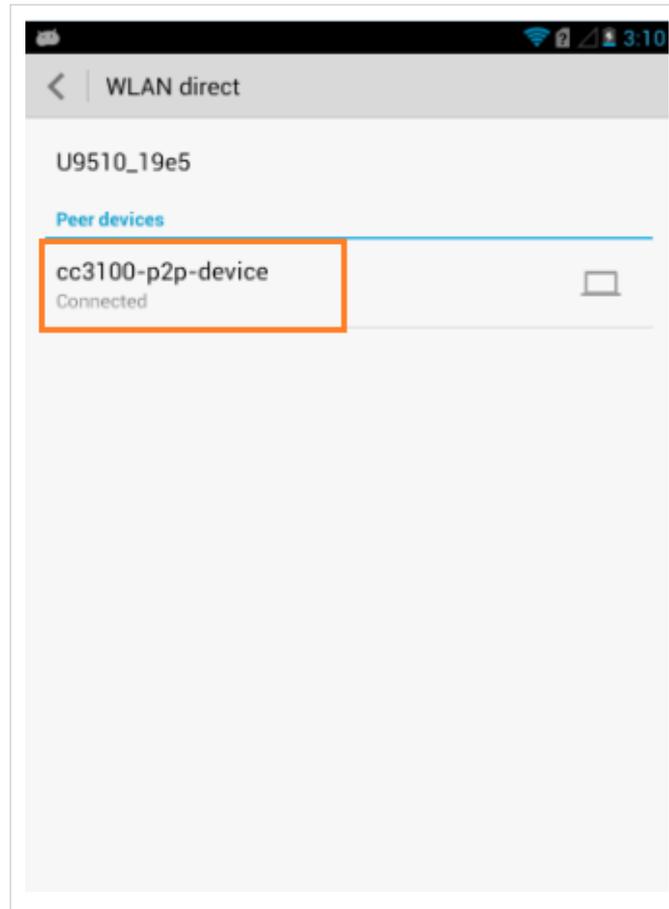- P2P connect security type 'PBC'

Usage

- Connect the board to PC and configure the terminal-program for seeing the logs - [| CC31xx_&_CC32xx_Terminal_Setting_Wiki [1]] has detailed instructions for configuring the terminal-program
- Build and run the application
- Start remote P2P device

```
CC3100 will be visible as P2P_DEVICE_NAME (which is set in the sample application) to the remote P2P device
```

- Connect to CC3100. On successful connection, CC3100's IP address will be displayed on the terminal-program's console

- Open an 'Ipef' client on the remote P2P device and connect on 'PORT_NUM'

```
Iperf.exe -c <DEST_IP_ADDR> -p <PORT_NUM> -i 1
```

- See the self explanatory logs on the terminal-program's console. On success, below message will be displayed on the terminal

Note: : User needs to reconfigure the device in 'Station-Mode' for executing other sample applications. Refer function configureSimpleLinkToDefaultState in this example's main.c for configuring the device in 'Station-Mode'.

Limitations/Known Issues

None

References

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

Article Sources and Contributors

CC31xx P2P Application *Source*: http://processors.wiki.ti.com/index.php?oldid=184928 *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
Image:P2P_1.png *Source*: http://processors.wiki.ti.com/index.php?title=File:P2P_1.png *License*: unknown *Contributors*: Codycooke