CC3100 P2P Application

## **CC3100 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\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'

CC3100 P2P Application 2

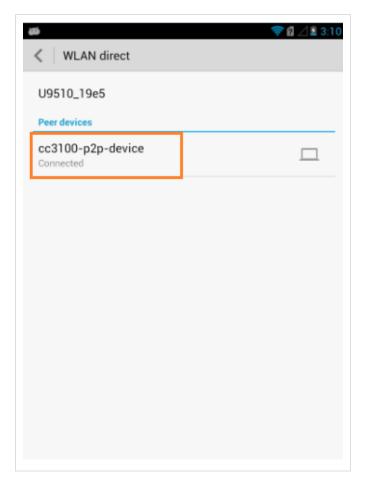
### Usage

 Connect the board to PC and configure the terminal-program for seeing the logs. CC3100 & CC3200 Terminal Setting <sup>[1]</sup> has detailed instructions for configuring the terminal-program

- Build and run the application
- Start remote P2P device (ex: Android device)

**Note:** 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 'Iperf' client on the remote P2P device and connect on 'PORT\_NUM'
- The Iperf version used in testing was: iperf-2.0.9-win64 (running on windows 10)

```
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'.

CC3100 P2P Application

## Limitations/Known Issues

None

### References

 $[1] \ http://processors.wiki.ti.com/index.php/CC3100\_\&\_CC3200\_Terminal\_Setting$ 

## **Article Sources and Contributors**

 $\textbf{CC3100 P2P Application} \quad \textit{Source}: \\ \textbf{http://processors.wiki.ti.com/index.php?oldid=233494} \quad \textit{Contributors}: \\ \textbf{A0131814, A0132173, A0221015, Codycooke, Malokyle, SarahPathillarian Contributors}: \\ \textbf{A0131814, A0132173, A0221015, A02210$ 

# **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 Image: P2P\_1.png Source: http://processors.wiki.ti.com/index.php?title=File: P2P\_1.png License: unknown Contributors: Codycooke