



REALTEK

Realtek Start Guide P2P User Interface

Date: 2012/04/06

Version: 0.6

Start Guide P2P User Interface

Files:

All the necessary files are available in the “WiFi_Direct_User_Interface” folder of Realtek software package.

Definitions:

- (1) **P2P Device:** WFA P2P certified device that is capable of acting as both a P2P Group Owner and a P2P Client.
- (2) **P2P Group Owner (GO):** An “AP(Access Point)-like” entity that may provide and use connectivity between P2P Clients
- (3) **P2P Client:** A P2P Device that is connected to a P2P Group Owner.

1 Install P2P user interface

1.1 Using P2P user interface needs the following files:

- (1) wpa_supplicant_hostapd-0.8_rtw_20111118.zip
- (2) p2p_hostapd.conf
- (3) wpa_0_8.conf

And in the folder wpa_supplicant_hostapd, it can't be run without these files.

1.2 After checking these files, we could insert command install.sh to install P2P_UI, it also takes p2p_hostapd.conf , wpa_0_8.conf in this folder and install wpa_supplicant, hostapd in this folder.

1.3 We marked DHCP client/server functionalities as default because it's quite different depending on platforms. We implemented it for Fedora in this P2P user interface. You could also validate p2p connection by setting static IP.

1.4 After installation complete, we would use P2P_UI

2 P2P_UI

2.1 Launch the P2P_UI with interface (eg. wlan0)

```
#./P2P_UI wlan0
```


PBC: Push Button

```

*****
P2P UI TEST v0.5
*****
* Enable: [Enable/Device]
* Intent: 1
* Status: P2P_STATE_LISTEN
* Role: P2P_ROLE_DISABLE
* WPS method: P2P_NO_WPSINFO
* PIN code: 12345670
* Device name: "Realtek DMP Device"
* Peer device address: 00:00:00:00:00:00
* Peer interface address: 00:00:00:00:00:00

e) Wi-Fi Direct Enable/Disable
i) Intent ( The degree to be Group Owner/SoftAP )
a) Scan Wi-Fi Direct devices
m) Peer device address you want to test
p) Provision discovery
c) Input PIN codes
w) WPS method
n) Group owner negotiation
x) Start wpa supplicant/hostapd
h) Set operation channel          | t) Set SoftAP ssid
r) Get Current P2P Role           | s) Get Current P2P Status
d) Set device name                | l) Set Listen channel
f) Reflash Current State          | q) Quit
*****

* NO  DEVICE NAME      BSSID          GO  CONFIG METHOD
* [01] Realtek DMP Device 00:E0:4C:00:E0:11
* [02] Realtek DMP Device 00:1F:1F:D8:49:2F      *
*****

*insert cmd:

```

- 3.4 Set peer device address (insert cmd: m).
 - 3.4.1 From scan list, we could choose one P2P device to start the P2P connection.
 - 3.4.2 Input the number of device that we want to do P2P handshake.
- 3.5 Issue provision discovery frame (insert cmd: p).
 - 3.5.1 The purpose for issuing the provision discovery frame is to set/get the WPS PIN CODE or WPS PBC to/from peer P2P device.
 - 3.5.2 In version 0.5, the following procedures of P2P handshake (including set WPS method, group owner negotiation, start wpa_supplicant or hostapd and DHCP client/server) is combined in this step.
 - 3.5.3 Select the way to get the WPS configuration method we want to use.
 - [0]: We input the PIN code that displays on peer device screen.
 - [1]: Our screen display PIN code and input this PIN code on the peer device
 - *If we use this WPS configure method, we should input PIN code at first (see 6.4) or use default PIN code: 12345670.
 - [2]: Both P2P devices will use the WPS PBC
 - [3]: We input the PIN code that printed on peer device WPS label (like sticker).

3.5.9 If wpa_supplicant using WPS connection success, it will show wpa state like the following:

3.5.10 After WPS connection successful, user interface will run DHCP client automatically. That time, two P2P devices connected successfully, we could try to ping each other.

4.5.1 If it's PIN code and peer device want us to input PIN code on their display, then we will show the following information and we need to input peer's PIN code,

4.5.2 If it's PIN code and peer P2P device wants to input our PIN code on their device (or input the PIN code that displays on our label), then we need insert command "c" as accepted.

4.5.3 If it's PBC, then we need insert command "b" as accepted.

6.3.1 The purpose for issuing the provision discovery frame is to set/get the WPS PIN CODE or WPS PBC to/from peer P2P device.

6.3.2 Select the way to get the WPS configuration method we want to use.

[0]: We input the PIN code that displays on peer device screen.

[1]: Our screen display PIN code and input this PIN code on the peer device

*If we use this WPS configure method, we should input PIN code at first (see 6.4) or use default PIN code: 12345670.

[2]: Both P2P devices will use the WPS PBC

[3]: We input the PIN code that printed on peer device WPS label (like sticker).

6.3.3 After 6.3.2, we issue provision discovery request and also declaring which WPS method we want to use(this information is inside provision discovery frame)

6.4 Input PIN codes (insert cmd: c).

6.4.1 This step is optional because the P2P_UI already provides the default PIN Code

6.4.2 The P2P_UI will check the validation of the inputted PIN code. If we want use [1] in 6.3.2, it's recommended that we input PIN code before issuing provision discovery

6.4.3 If we select [0] in 6.3.2, we input PIN code that displays on the screen of peer P2P device.

6.4.4 We don't need to input PIN code if you select the [2] in 6.3.2

6.4.5 If we select [3] in 6.3.2, we should input the PIN code which printed on the label of peer P2P device.

6.5 Set WPS configuration method (insert cmd: w).

6.5.1 This command is used to inform the P2P driver that the P2P_UI already received the WPS PIN code or WPS PBC. The WPS PIN code and WPS PBC should be acquired by using the command 6.3 and 6.4.

6.5.2 We will set WPS configuration method based on 6.3 we issued.

If you select the [0] in 6.3, we will choose the [1] here.

If you select the [1] in 6.3, we will choose the [2] here.

If you select the [2] in 6.3, we will choose the [3] here.

If you select the [3] in 6.3, we will choose the [1] here.

6.6 Issue group owner negotiation (insert cmd: n)

6.6.1 It will do the group owner negotiation handshake. If our intent is lower than peer device, the P2P_UI would start wpa_supplicant automatically and act as P2P client.

6.6.2 Else, if our intent is higher than peer device, the P2P_UI would start hostapd automatically and act as P2P GO.

6.6.3 For group owner negotiation procedure, it could be issued by one of two P2P devices.

6.7 DHCP

6.7.1 If wpa_supplicant reported “wpa_state=COMPLETED”, it means P2P connection is established successfully and P2P_UI will start dhclient automatically to try to get the IP from the P2P GO.

6.7.2 If hostapd reported “dot11RSNAStatsSTAAddress=XX:XX:XX:XX:XX:XX”, it means the P2P connection is established successfully and the P2P_UI will start DHCP server automatically.

6.8 For the other P2P device (passive connected), it could do re-flash current state to check whether P2P device connected using P2P handshake.

7 Optional commands.

The following are some optional commands that could set or get information from drivers.

7.1 Set operation channel (insert cmd: h)

7.1.1 It is optional command because the P2P_UI will have the rule to determine the default value for the operation channel. However, this information will be ignored if this P2P device is the P2P client after the group owner negotiation handshake. Also, if hostapd is running, we could not change this value, same as 7.2 and 7.5

7.1.2 The P2P_UI will use the value of “channel” in /hostapd.conf as default value.

7.1.3 This operation channel is the channel number which the P2P GO should stay if the local P2P device is the GO.

7.1.4 Notice that the value should be the same as the value of “channel” in /hostapd.conf

7.2 Set softAP ssid (insert cmd: t)

7.2.1 It's optional because the P2P_UI will use the content of “ssid” in /hostapd.conf as default

7.2.2 This command will inform other drivers the ssid string if this P2P device is the GO.

7.2.3 Notice that the value should be the same as the content of “ssid” in /hostapd.conf

7.3 Get current P2P role (insert cmd: r)

You can check the P2P role which the driver is acting as.

- 0: Disable
- 1: P2P Device
- 2: P2P Client
- 3: P2P GO

7.4 Get current P2P status (insert cmd: s)

You can check the current P2P status that the driver stands for.

- 0: P2P_STATE_NONE
- 1: P2P_STATE_IDLE
- 2: P2P_STATE_LISTEN
- 3: P2P_STATE_SCAN
- 4: P2P_STATE_FIND_PHASE_LISTEN
- 5: P2P_STATE_FIND_PHASE_SEARCH
- 6: P2P_STATE_TX_PROVISION_DIS_REQ
- 7: P2P_STATE_RX_PROVISION_DIS_RSP
- 8: P2P_STATE_RX_PROVISION_DIS_REQ
- 9: P2P_STATE_GONEGO_ING
- 10: P2P_STATE_GONEGO_OK
- 11: P2P_STATE_GONEGO_FAIL
- 12: P2P_STATE_RECV_INVITE_REQ
- 13: P2P_STATE_PROVISIONING_ING
- 14: P2P_STATE_PROVISIONING_DONE

7.5 Set device name (insert cmd: d)

Set the device name which will be seeing by the other P2P devices. In Default we get this name from /hostapd.conf.

7.5.1 It's optional because the P2P_UI will use the content of "ssid" in /hostapd.conf as default

7.5.2 This command will set the P2P device name which will be seeing by other P2P devices.

7.5.3 Notice that the value should be the same as the content of "ssid" in /hostapd.conf

7.6 Set listen channel (insert cmd: l)

Set the listen channel where the device will stay in the listen state, the listen channel will set channel 1, 6, 11 only, as the spec. described.

7.7 Reflash current state (insert cmd: f)

It will reflash current state and other information that P2P_UI maintains.

7.8 Quit (insert cmd: q)

Quit P2P user interface, P2P functionality will be disabled.