Release Date	Document Version	Note
2012/8/21	1.0	First version
2012/9/14	1.1	Refine statement
2012/10/01	1.2	Support Android ICS & JB
2012/10/16	1.3	Refine statement
2013/08/09	1.4	Refinement

1. Introduction

Wi-Fi Alliance had announced the Miracast (Wi-Fi Display developed by Wi-Fi Alliace – WFD) in May 2012 and the specification had been finalized in the end of August 2012.

This document is not prepared for Android 4.2(and newer), They had their own structure implementing Miracast.
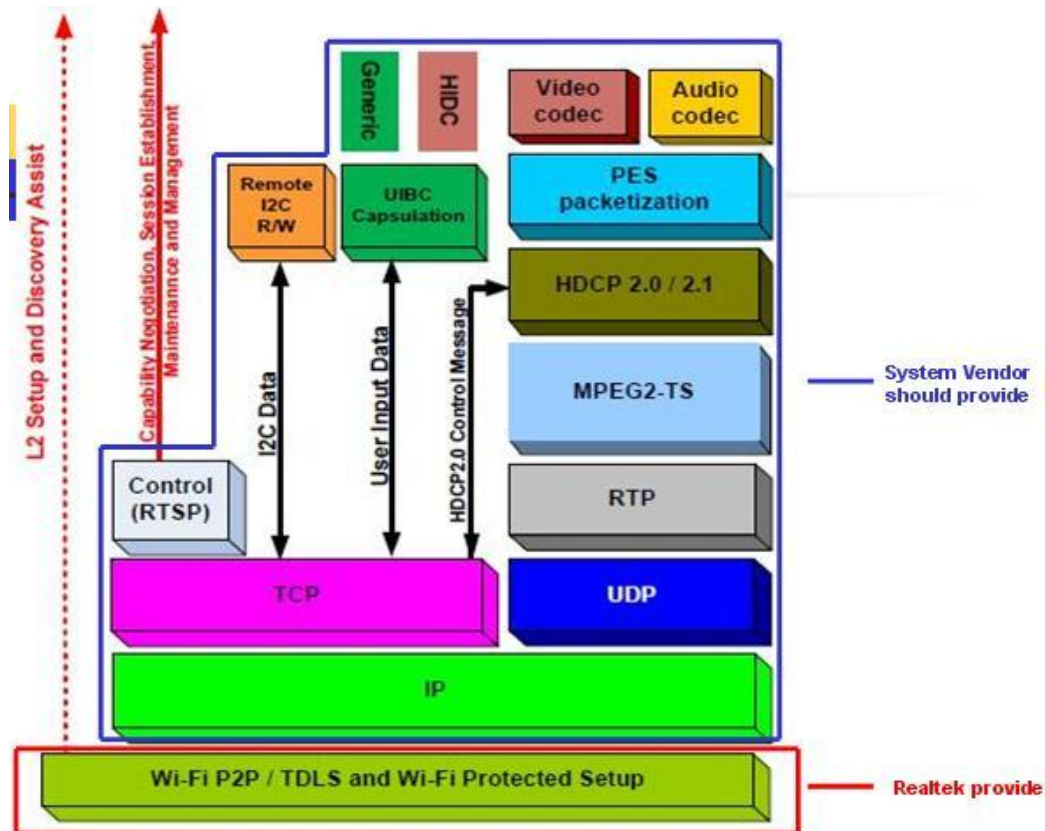
This picture is captured from the Miracast specification. The parts surrounded by Blue line should be provided/implemented by system vendor. Realtek will provide the parts surrounded by Red line.

The Realtek software package had included the Wi-Fi driver and wpa_supplicant for

Miracast feature. The Wi-Fi driver and wpa_supplicant had been modified to support Miracast on Andorid 4.0(ICS) / Android 4.1 (Jelly Bean) platforms or Linux embedded systems. This document will provide the guideline how to enable/process the Miracast these platforms by using the Realtek Wi-Fi solution.

2. Configure wpa_supplicant for Miracast

In the software package, Realtek provided three wpa_supplicants. Please refer to the following wpa_supplicants by your platform

platform	wpa_supplicant
Android iCS	wpa_supplicant_8 in android_reference_codes_ICS_nl80211
Android JB 4.1	wpa_supplicant_8_jb_4.1_rtw_rxxxx.xxxxxxxx.tar.gz
Linux	wpa_supplicant_8_jb_4.2_rtw_rxxxx.xxxxxxxx.tar.gz

In order to support the Miracast in wpa_suppliacnt, some flags should be turn on manually before compiling it.

2.1: Function configuration file for wpa_supplicant

Before compiling the wpa_supplicant, we need to set the CONFIG_WFD flag to y and enable it in the .config file of wpa_supplicant folder to support the Miracast function.

# Enable Wi-Fi Display (Miracast)
CONFIG_WFD=y

Please also make sure the following flags had been defined and enabled as well in the .config file for Wi-Fi Direct functionality.
CONFIG_DRIVER_NL80211=y
CONFIG_CTRL_IFACE=y
CONFIG_P2P=y
CONFIG_AP=y
CONFIG_WPS=y
CONFIG_WPS2=y
P.S.: For wpa_supplicant_8_jb_4.2_rtw_rxxxx.xxxxxxxx.tar.gz, please disable CONFIG_WIFI_DISPLAY because it's for Android 4.2 (and newer) usage. That is:
CONFIG_WIFI_DISPLAY=y

After making sure all the necessary flags are defined and enabled, please re-compile it.

If a wpa_supplicant already exists on your system with supporting the Wi-Fi Direct functionality, you can search the "CONFIG_WFD" keyword in this released wpa_supplicant and merge it into your wpa_suppliacnt as well.

2.2 Setting configuration file for wpa_supplicant (WFD.conf file of Realtek software package)
In the setting configuration file, it can contain the Miracast pre-values when the wpa_supplicant is initialized.

These pre-values are:
device_name=Media Player v10
wfd_tcpport=554
wfd_device_type=1

The "device_name" is the human-readable string. Other Miracast device will show this name on its application/utility for your Miracast device.
The "wfd_tcpport" is the port number used by your Miracast RTSP engine. The default value is 554.
The "wfd_device_type" will specify the type of your Miracast device. If your Miracast device is the Source device, please assign 0 to it. If your Miracast device is Sink device, please assign 1 to it.

3. Configuration Wi-Fi driver with Miracast supporting
To make sure the following flags are enabled in the include/autoconf.h file of Wi-Fi driver folder.
#define CONFIG_IOCTL_CFG80211   1
    //#define RTW_USE_CFG80211_STA_EVENT /* Indecate new sta asoc through cfg80211_new_sta */
    #define CONFIG_CFG80211_FORCE_COMPATIBLE_2_6_37_UNDER
#define CONFIG_P2P   1
#define CONFIG_WFD   1
#define CONFIG_CONCURRENT_MODE   1

Please enable RTW_USE_CFG80211_STA_EVENT if you Linux kernel version is >= 3.2.0
After enabling driver flags, the Wi-Fi driver should be re-compiled to include the Miracast functionality.

4. Enable Miracast function

In this document, we use the command line to launch/operate the wpa_supplicant and wpa_cli application. It is a convenient way to experiment and check the result manually. In the Android system, all the steps described below should be integrated into the Android framework and the Android application should handle to parse the result strings returned by wpa_cli & wpa_supplicant.

Step1: Plug in the Wi-Fi dongle.
Step2: Insert the Wi-Fi driver which enables all necessary flags to support the Miracast.
Step3: Launch the wpa_supplicant
#> wpa_supplicant –iwlan0 –Dnl80211 –c ./WFD.conf –dd &
Step4: Launch the wpa_cli to open the command interface
#> wpa_cli
Step5: Enable the Miracast
#> driver wfd-enable
#> p2p_listen
Last Step: Disable the Miracast
#> driver wfd-disable

5. Start the Miracast connection
5.1 Search other Miracast devices
Step 1: Enable the Miracast
#> driver wfd-enable
Step 2: Search other Miracast devices for 10 seconds
#> p2p_find 10

The wpa_cli will show up all the devices which support Wi-Fi Direct and Miracast.

```
> p2p_find 10
OK
<3>P2P-DEVICE-FOUND 12:03:7f:10:85:fc p2p_dev_addr=02:03:7f:10:85:fc pr
i_dev_type=1-0050F204-1 name='Atheros Wi-Fi Direct' config_methods=0x0
dev_capab=0x21 group_capab=0x9
<3>P2P-DEVICE-FOUND c2:cb:38:06:ac:5d p2p_dev_addr=c2:cb:38:06:2c:5d pr
i_dev_type=1-0050F204-5 name='DIRECT-8Y7.1.4' config_methods=0x80 dev_c
apab=0x37 group_capab=0x49
<3>P2P-DEVICE-FOUND 00:e0:4c:00:d1:f4 p2p_dev_addr=00:e0:4c:00:d1:f4 pr
i_dev_type=1-0050F204-1 name='REALTEK-PC' config_methods=0x238c dev_cap
ab=0x21 group_capab=0x0 wfd_session_avail:1 wfd_rtsp_ctrlport=8554, wfd
_device=0
```

In this example, three P2P devices are discovered. However, only one P2P device with

00:e0:4c:00:d1:f4 MAC address supports the Miracast feature. If the P2P device supports the Miracast feature, the "wfd_session_avail", "wfd_rtsp_ctrlport", "wfd_device" will appear in its attribute information.

The "wfd_session_avail" will show up the status of WFD session of this P2P device. "wfd_session_avail:1" means this P2P device is ready for Miracast session. "wfd_session_avail:0" means the Miracast session of this P2P device is occupied and is not able to start the Miracast now.

The "wfd_rtsp_ctrlport" means the port number of RTSP engine is using now. A Miracast Sink device should fetch this port number and connect to it after having the successful P2P connection.

The "wfd_device" means the Miracast device type of this P2P device. "wfd_device=0" means this P2P device is Miracast Source device. "wfd_device=1" means this P2P device is Miracast Sink device.

Step 3: Connect to a Miracast device using wpa_cli

#> ./wpa_cli p2p_prov_disc 00:e0:4c:00:d1:f4 pbc //Send provision discovery to peer Miracast device

#> ./wpa_cli p2p_connect 00:e0:4c:00:d1:f4 pbc //Connect to peer Miracast device

Step 4: Check the link status

The "status" command can show up the information of current Wi-Fi link.

```
> status
bssid=00:e0:4c:00:d1:f4
ssid=DIRECT-Ng
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=CCMP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
p2p_device_address=00:02:72:b7:5e:0b
address=00:02:72:b7:5e:0b
```

We can use the "mode" information to know this Miracast device is P2P GO or P2P client. If this Miracast device is P2P client, the "mode" element will be "station". If this device is P2P GO, the "mode" will be "P2P GO".

When the "wpa_state" element is "COMPLETED", it means the P2P connection had been established successfully.

Step 5: Get the IP address for peer Miracast device

As the introduction described above, the Miracast Source device will start the RTSP

server on its TCP control port number and wait for the RTSP connection from the Miracast Sink device.

For the Miracast Sink device, it is necessary to know the IP address of Miracast Source device. After that, the Miracast Sink device is just able to connect to Miracast Source device to start the RTSP handshake.

Unfortunately, there is no a common way to get peer's IP address on all systems. For Linux system, it is possible to use the different DHCP client/DHCP Server applications. But there might be different ways to get the IP address for the peer's Miracast device.

On other Android systems, we can use the showlease command to know peer Miracast device if this Miracast device is P2P client.
#> showlease wlan0 // Assume we had a P2P connection on wlan0
Addr: 192.168.10.113
Net: 255.255.255.0
Leasetime: 604800
Renew: 0
Rebind: 0
Server: 192.168.10.10

If this Miracast device is P2P client, the IP address for Miracast Source device is 192.168.10.10 in this example. We can also check the "WifiP2pInfo.groupOwnerAddress" to know the IP address for Miracast source device.

If this Miracast device is P2P Group Owner, we can check the /data/misc/dhcp/dnsmask.leases file to know the IP address for peer Miracast Sink device.

For the Miracast Sink device, it can start the RTSP handshake after getting the IP address of Miracast Source device and control port number (by using the scan results of p2p_find).

For the Miracast Source device, it should just launch the RTSP server and wait for the connection on control port.