Atmel SmartConnect

AT14614: ATWINC1500 Wi-Fi Network Controller - P2P Mode

APPLICATION NOTE

Introduction

This example application note demonstrates the steps to execute the ATWINC1500 Wi-Fi® P2P or Wi-Fi Direct mode application using the Atmel® SAM D21 Xplained Pro board as host MCU.

The following topics are covered:

- Organization of demo application
- Information about target boards
- Flow of demo application
- Step-by-step execution of the API

Features

- ATWINC1500 host MCU driver architecture
- ATWINC1500 internal architecture
- Application description with code snippets
- Events handled in the Wi-Fi callback function with appropriate structures used for each event
- Steps to execute the P2P or Wi-Fi Direct mode application demo using SAM D21 Xplained Pro board and ATWINC1500

Table of Contents

1. Application Overview

This section describes Wi-Fi P2P (Wi-Fi Direct™) terminologies used in the appnote and connection process is explained.

What is P2P or Wi-Fi Direct?

The term P2P refers to "peer-to-peer" networking. A peer-to-peer network allows computer hardware and software to function without the need for special server devices. P2P is an alternative to client-server network design. They are said to form a peer-to-peer network of nodes.

Wi-Fi Direct is not an IEEE standard, but a Wi-Fi Alliance technical specification called "Wi-Fi Peer-to-Peer (P2P) Specification". It allows Wi-Fi devices to connect to each other and form groups usually one-to-one, but also one-to-many. Wi-Fi Direct devices negotiate their roles in the connection. one of them assumes the traditional role of Access Point (AP) called *Group Owner (GO)* and the other devices, including non-Wi-Fi Direct enabled devices connect to the *AP/GO* as clients in station mode.

Wi-Fi Direct has some basic requirements such as,

- IEEE 802.11g supported WLAN supported devices
- RSN(WPA) with AES-CCMP encryption security type
- Wi-Fi Protected Setup (WPS)
- WME (WMM) WLAN multimedia support
- QoS (Quality of Service)
- OFDM data rate for management frames

The P2P device use the global MAC address as a *Device ID*. During the discovery and negotiation process, temporary local MAC address used for all frames within a group. *Action frames* and additional *Information Elements (IE)* are used to transport the details of the protocol.

What is P2P Group Owner and Autonomous GO (P2P GO) ?

The P2P device implementing AP-such as functionality in P2P group referred to as a P2P group owner (P2P GO) and device acting as client are know as P2P client. In Autonomous Group Owner (Auto GO) mode, by default P2P device start as a group owner.

In another method group owner will be decided based on intent value in the negotiation process. Higher intent value device becomes a group owner and lower value device act as a P2P client. A peer-to-peer network allows WLAN devices to directly communicate with each other. A legacy WLAN client device also connects with P2P group owner device acting as AP.

What is Intent Value ?

Its numerical value 0 to 15 is used in the P2P group formation. In order to agree on the device that acts as P2P GO, P2P devices send a numerical parameter, the GO Intent value, within the three-way handshake, and the device declaring the highest value becomes the P2P GO. To prevent conflicts when two devices declare the same GO Intent, a tie-breaker bit is included in the GO Negotiation Request, which is randomly set every time a GO Negotiation Request is sent.

What is P2P Client ?

The P2P capable WLAN device with lower intent value than the other P2P peer device act as P2P client. It decides in the negotiation process.

What is Wi-Fi Protected Setup or Wi-Fi Simple Configuration (WSC) ?

WPS is method used to get the credentials from AP by station to make the connection between the WLAN devices. This method works only on WPA2-PSK security configuration. In P2P connection

process, WPS method is used to connect with two P2P devices. In this method, AP or router acts as a *Registrar* and WLAN client or station acts as *Enrollee*.

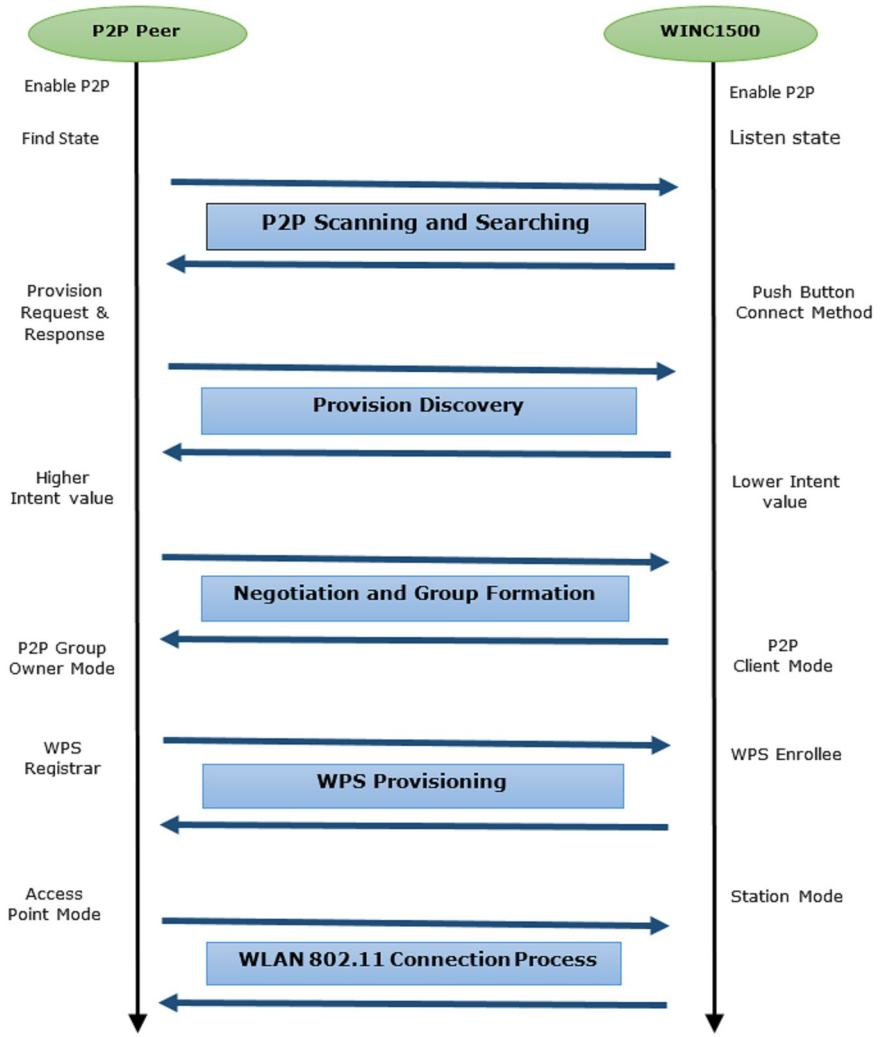
What is Enrollee ?

The WPS protocol defines, enrollee as a device seeking to join a wireless network. The enrollee detects the registrar device by using scan results and starts the provisioning process. The credentials obtained by the WPS provisioning process are used to connects with desired AP.

What is Registrar ?

The WPS protocol defines registrar, a device with the authority to issue and revoke access to a network; it may be integrated into a wireless access point (AP), or provided as a separate device. Registrar adds the WPS IE (Information Element) to advertise the WPS capability in the beacon frame.

Figure 1-1. P2P Connection Process



1.1. Device Discovery or P2P Find

The ATWINC1500 can only act as P2P client device and can be connected with another P2PGO device only. Before any connection can be established, P2P devices have to find each other. For this they alternately listen and send *probe requests* with additional *P2P information elements* on so-called *social*

channels, which are channels 1, 6 and 11 in the 2.4GHz band. Unconnected P2P devices and *Group Owners* reply to them with *probe response* frames which also include *P2P information elements* describing the device and group characteristics. *Group Owners* respond on behalf of devices which are part of their group, but clients may choose not be discoverable while they are connected to a group.

The ATWINC1500 starts as a P2P peer and will be in listen mode in the specified channel. In *listen mode* ATWINC1500 will not send any *probe requests* to get the information about the other P2P device. But in listen mode ATWINC1500 can able to respond for the *probe requests* from the other P2P peer by *probe response* frame with device name, supported P2P connection methods, listen and operating channel, *intent* value etc. Once other P2P peer received the probe response, it will display the ATWINC1500 device name as a supported P2P peer device list.

1.2. Provisional Discovery

Provisional discovery request and response frames are used to notify the connection method to other P2P peers. The ATWINC1500 P2P peer starts as a P2P client with response to the provisional discovery request. The P2P client initiates the WPS provisioning process to get the credentials from the group owner peer.

Note:

1. ATWINC1500 supports only *PushButton* configuration method to connect with other peer.
2. ATWINC1500 supports P2P client mode only. Connecting two ATWINC1500 in P2P mode is not supported.

1.3. Group Formation

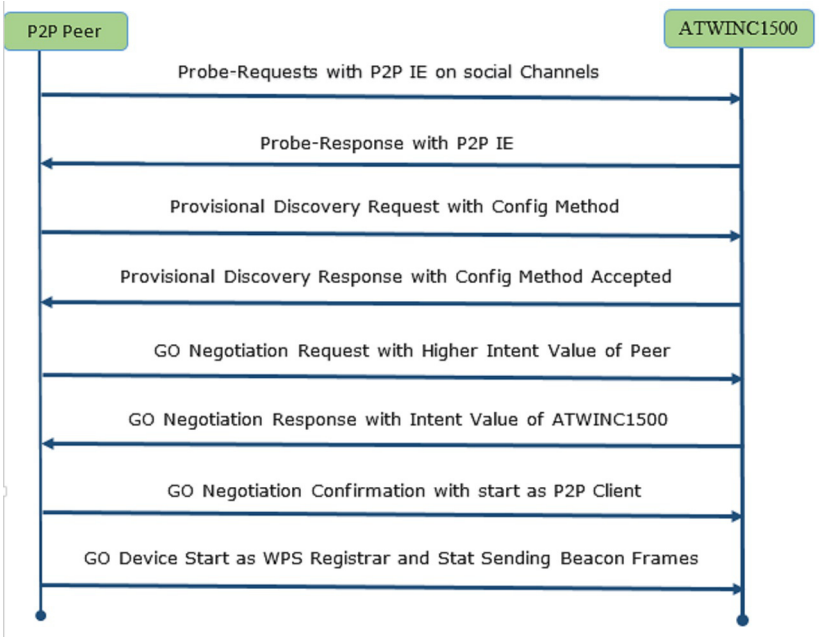
The P2P supported WLAN device acts as a either *P2P Group Owner (GO)* or *P2P Client*. The role of the WLAN device decides according to the intent value by group negotiation process.

The ATWINC1500 device intent value is set lower value *(Intent attribute is 0-15)* to become a P2P client in the negotiation process. In the negotiation process higher intent value P2P peer become the *group owner*. In this process *Negotiation request, response, and confirmation* frames are exchanged between the P2P peers. If the configuration method or any of the P2P parameters are not supported group formation fails in the negotiation process.

After the successful *Group formation* ATWINC1500 starts as a P2P client and other peer starts as P2P GO or group owner. These frame exchange happens only on operating channel of the P2P devices.

The *GO* device acts as softAP and start sending the beacon frames. The P2P client device acts as WLAN client device (station mode). The *SSID* of the softAP is standardized to be "DIRECT-xy..." with xy being random characters or numbers and any postfix.

Figure 1-2. P2P Group Formation Process



1.4. WPS Provisioning and Connection Process

The *provisioning* phase begins. The client connects to the *GO Device* to exchange credentials using the *WPS* specification protocol. This process includes an exchange of eight *EAP (Extensible Authentication Protocol)* messages. To allow the connection, normally the user has to enter a PIN code or push a button on the device. In ATWINC1500 P2P configuration by default *PushButton* method is configured.

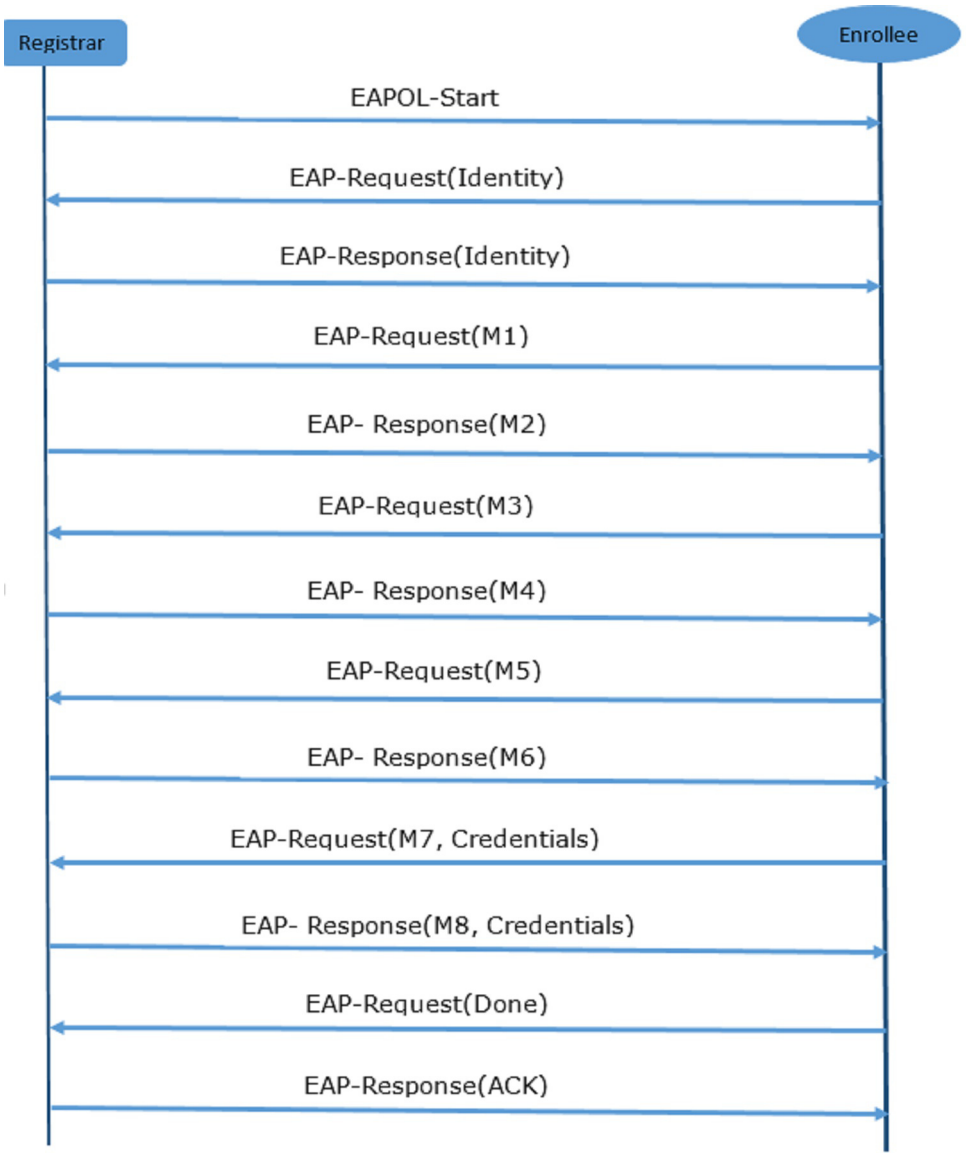
As per the P2P protocol, group owner act as a *Registrar* and P2P client act as a *Enrollee* device. Registrar starts the AP mode and Enrollee connects with AP in open Wi-Fi connection. After the open connection, enrollee look for the enabled *selected registrar* bit in the WPS IE of the registrar. When it founds in the IE of the registrar beacon frame, enrollee connects with registrar in OPEN connection mode. After the open connection WPS *EAP-Start* frame to initiated from the registrar to starts WPS provisioning process as shown.

WPS provisioning process completes after the *M1-M8* frame exchange. As part of the M8 frame, AP (Registrar) connection credentials are exchanged between the enrollee and registrar. At the end of the M8 WPS frame, *EAP-Fail and De-Authentication* frames sends from the registrar to disconnect with enrollee. When WPS provisioning process completed, P2P client peer acts as a station and other Group owner peer act as AP. Using the credentials exchanged in the WPS provisioning process, enrollee starts the station and AP connection process proceeds with the WPA2-PSK 4-way handshake key authentication process.

When joining an existing group, or to speed up the provisioning phase later, devices can send *Provision Discovery request/response* frames before starting the group negotiation. Otherwise the *GO Negotiation* may fail. The WPS provisioning process to be restarted.

After that, the normal *RSN (WPA2) 4-way handshake* begins to exchange the encryption keys, where the *GO* assumes the role of *authenticator* and the client is the *supplicant*. The client starts the DHCP client to obtain the IPv4 address from the *GO Device*.

Figure 1-3. WPS Provisioning Process



2. Application Description

This section elaborates the ATWINC1500 host driver P2P mode application in detail. P2P connection process will be explained with respect to the ATWINC1500 API's sequence.

2.1. Wi-Fi Host Driver Initialization

- System Initialization of SAM D21 Xplained Pro board consists of MCU's clock initialize, hardware events and external hardware interfaces.

```
/* Initialize the board. */
    system_init();
```

- Configuration of console UART interface used for debug log output. Debug log level value can be set using `M2M_LOG_LEVEL` macro in the `nm_debug.h` file.

```
/* Initialize the UART console. */
    configure_console();
    printf(STRING_HEADER);
```

- The BSP driver initialization will follow the ATWINC1500 bring-up sequence. The sequence of chip enable and reset pin of ATWINC1500 is followed. Refer the `nm_bsp_init` API definition.

```
/* Initialize the BSP. */
    nm_bsp_init();
```

- The Wi-Fi host driver initialization starts with the API `m2m_wifi_init()` and the structure `tstrWifiInitParam`. The Wi-Fi initialization sequence configures the SPI communication interface and external interrupt with respect to the host MCU peripherals. Apart from this, the Wi-Fi host application layer callback function and the `wifi_cb()` function is registered during the initialization sequence.
- When the SPI interface initialization is complete, host MCU reads chip-id of the ATWINC1500. It indicates that WLAN module is ready for initialization process which needs the module reset and waits for the confirmation from the module.
- The Wi-Fi application callback function indicates the events such as connect and disconnect status. The function operates by obtaining the IP address from the DHCP server with respect to the DHCP request from the ATWINC1500 station. In SoftAP mode, DHCP server providing IP address to the connected WLAN client device will also be notified by the event.

```
/* Initialize Wi-Fi parameters structure. */
    memset((uint8_t *)&param, 0, sizeof(tstrWifiInitParam));

    /* Initialize Wi-Fi driver with data and status callbacks. */
    param.pfAppWifiCb = wifi_cb;
    ret = m2m_wifi_init(&param);
    if (M2M_SUCCESS != ret) {
        printf("main: m2m_wifi_init call error!(%d)\r\n", ret);
        while (1) {
        }
    }
```

2.2. Configuring P2P Device Name

The purpose of the P2P device name to list the P2P device in the find list, identify the MAC address of the particular device etc. The scanning process P2P device name is shared using probe request and

response frames. The `MAIN_WLAN_DEVICE_NAME` macro and `m2m_wifi_set_device_name()` API used to modify the P2P device name.

```
/** P2P mode Settings */
       #define MAIN_WLAN_DEVICE_NAME    "WINC1500_P2P" /* < P2P Device Name in main.h file*/
       /* Set device name to be shown in peer device. */
    ret = m2m_wifi_set_device_name((uint8_t *)MAIN_WLAN_DEVICE_NAME,
strlen(MAIN_WLAN_DEVICE_NAME));
    if (M2M_SUCCESS != ret) {
        printf("main: m2m_wifi_set_device_name call error!\r\n");
    }
```

2.3. Enabling P2P Mode in ATWINC1500

The function `m2m_wifi_p2p()` is used to start the ATWINC1500 in P2P client mode. ATWINC1500 host application driver provides the listen channel information as argument of this function. To start the P2P mode, parameters required such as P2P device name, listen channel, connection method, P2P device MAC address, etc. ATWINC1500 starts the P2P peer in listen mode. In the P2P configuration, social channel 1,6, or 11 is used. To configure the channel, modify the macro in `main.h` file. The channel value is enumeration type `tenuM2mScanCh`, declared in the `m2m_types.h`.

```
#define MAIN_WLAN_CHANNEL        M2M_WIFI_CH_6 /* < P2P Channel number */
/* Bring up P2P mode with channel number. */
    ret = m2m_wifi_p2p(MAIN_WLAN_CHANNEL);
    if (M2M_SUCCESS != ret) {
        printf("main: m2m_wifi_p2p call error!\r\n");
    }
       printf("P2P mode started. You can connect to %s.\r\n", (char *)MAIN_WLAN_DEVICE_NAME);
```

2.4. Wi-Fi Host Driver Event and Callback Handling

All the Wi-Fi host driver event is handled in the `m2m_wifi_handle_events()`. The HIF (Host communication Interface) layer API's is used to monitor the external interrupt which registered using host MCU configuration.

```
while (1) {
        /* Handle pending events from network controller. */
        while (m2m_wifi_handle_events(NULL) != M2M_SUCCESS) {
        }
    }
```

The ATWINC1500 triggers the external IRQ, the host interface ISR layer reads the ATWINC1500 control register to identify the type of event which triggered the external interrupt. Depending on the event, if any data is available in the ATWINC1500, registered callback functions are called with appropriate data.

Host MCU Wi-Fi application driver handles the event with various categories such as:

- `m2m_wifi_cb` handles all the Wi-Fi configuration and connection events.
- `m2m_ip_cb` handles all the socket, and network application event callbacks.
- `m2m_ota_cb` handles all the *Over The Air* firmware upgrade events.

2.5. Wi-Fi Callback Function

Wi-Fi callback function is called depending on the success or failure state of the connection status and DHCP request confirmation.

```
static void wifi_cb(uint8_t u8MsgType, void *pvMsg)
{
   switch (u8MsgType) {
```

```
    case M2M_WIFI_RESP_CON_STATE_CHANGED:{
     tstrM2mWifiStateChanged *pstrWifiState =
                 (tstrM2mWifiStateChanged *)pvMsg;
     ..............................
    case M2M_WIFI_REQ_DHCP_CONF:{
     ..............................
}
```

The Wi-Fi callback function is called in the various scenarios. List of events handled during the `wifi_cb()` function are provided in the following table.

Table 2-1. Wi-Fi Callback Events

Wi-Fi Callback Events	Structure used for the Events	Comments
M2M_WIFI_RESP_SCAN_DONE	tstrM2mScanDone	Scan complete notification response for requested Scan command
M2M_WIFI_RESP_SCAN_RESULT	tstrM2mWifiscanResult	Response for the requested Scan results command
M2M_WIFI_RESP_CON_STATE_CHANGED	tstrM2mWifiStateChanged	WLAN connection state whether station or SoftAP mode
M2M_WIFI_RESP_CURRENT_RSSI	char *	Response to M2M_WIFI_REQ_CURRENT_RSSI with the RSSI value
M2M_WIFI_RESP_CONN_INFO	tstrM2MConnInfo	Connected AP information response
M2M_WIFI_RESP_PROVISION_INFO	tstrM2MProvisionInfo	Received provisioning information from the HTTP web page
M2M_WIFI_RESP_ETHERNET_RX_PACKET	char *	Receiving 802.3 type ethernet packet in bypass mode
M2M_WIFI_REQ_DHCP_CONF	tstrM2MIPConfig	Response indicating that IP address obtained and Netmask, Gateway, DNS addresses of the network
M2M_WIFI_RESP_IP_CONFLICT	unsigned int	Response indicating a conflict in obtained IP address. The user should re attempt the DHCP request
M2M_WIFI_RESP_GET_SYS_TIME	tstrSystemTime	Response of the time of day from network
M2M_WIFI_RESP_WIFI_RX_PACKET	tstrM2MWifiRxPacketInfo	Indicate that a packet was received in monitor mode
M2M_WIFI_RESP_DEFAULT_CONNECT	tstrM2MDefaultConnResp	Response for the connection information in default connect

Note: To set the static IP address, refer the FAQ. To get the gateway, DNS, and netmask address - refer the FAQ.

3. Steps to Run the P2P Mode Application

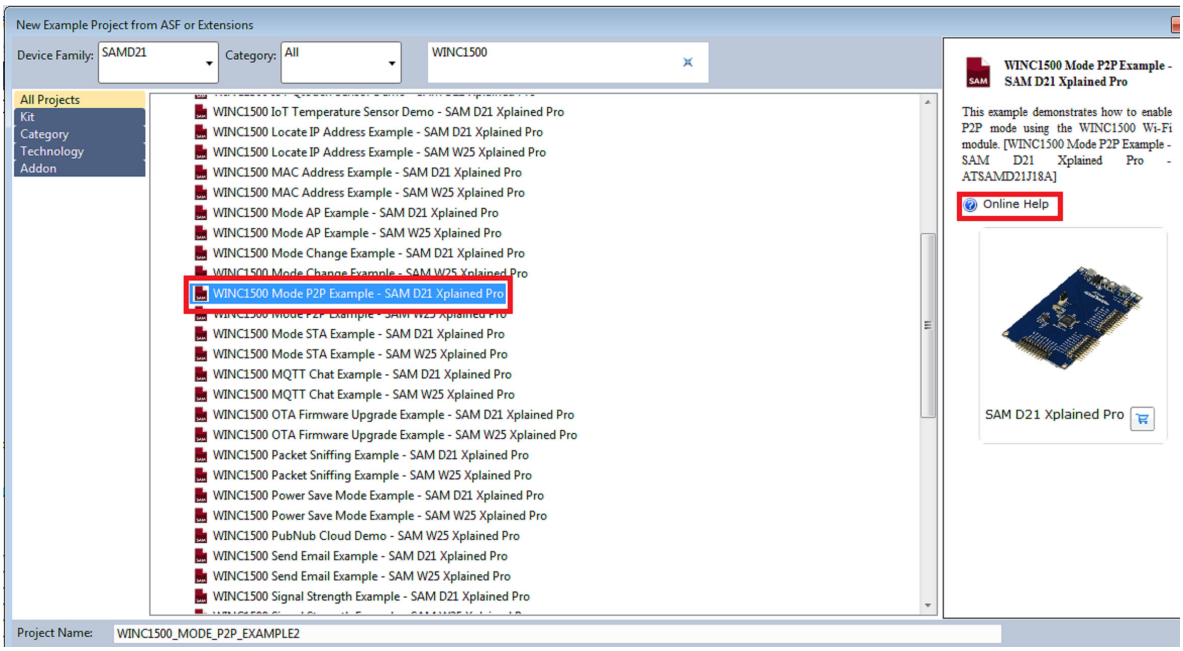
This topic provides the steps to run the P2P mode application using SAM D21 XPlained Pro board with ATWINC1500 WLAN module.

3.1. Getting Started ASF ATWINC1500 P2P Mode Demo

This topic provides the steps for demonstrating ATWINC1500 projects using Atmel Studio ASF example applications.

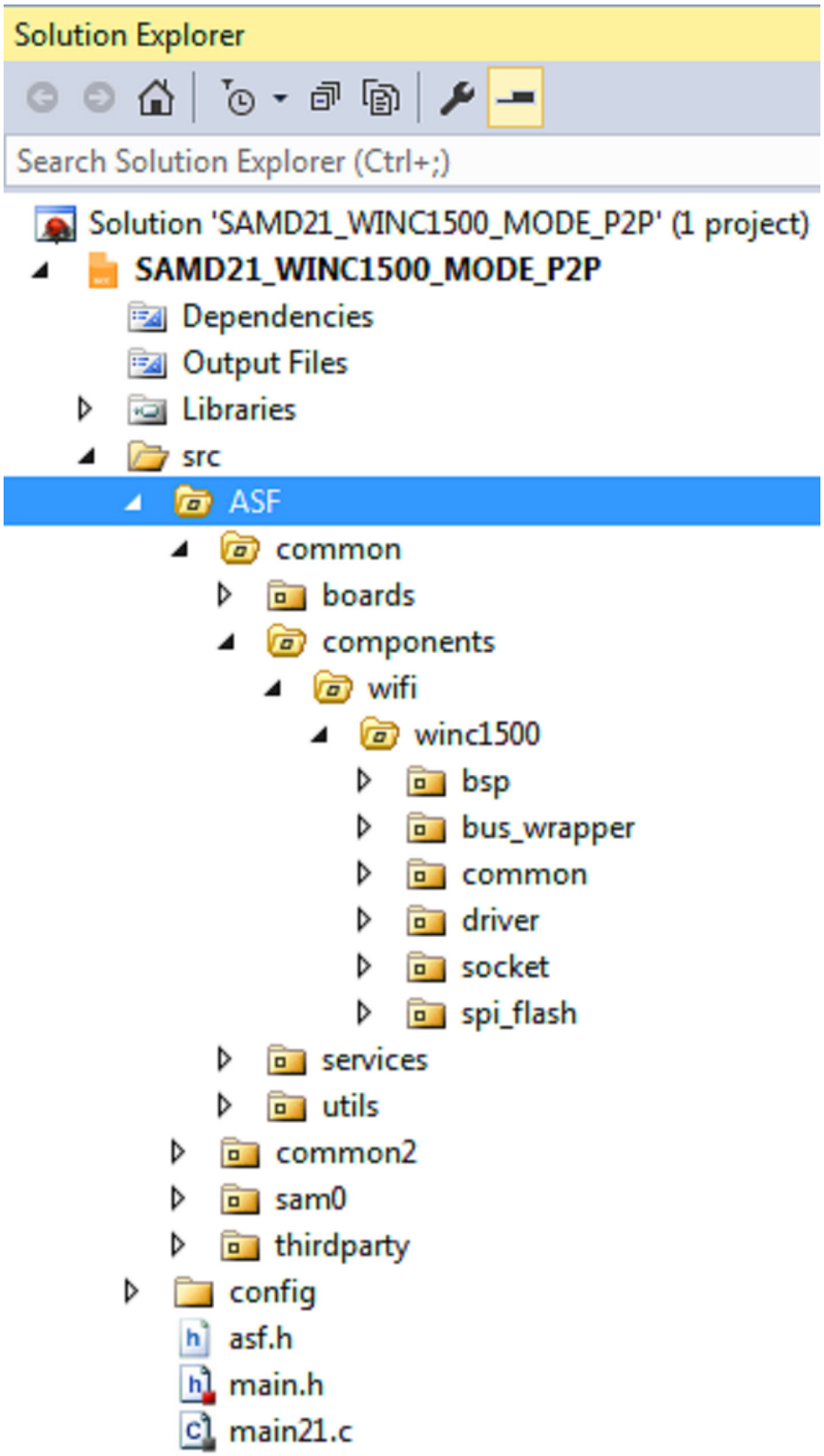
1. Open Atmel Studio 7. Go to **File** > **New** > **Example Projects**.
2. Search for ATWINC1500 sample application for other MCU.
3. Select the Wi-Fi P2P Mode Example `WINC1500_MODE_P2P_EXAMPLE` project for SAM D21 and open it.

Figure 3-1. Atmel Studio ATWINC1500 Project Creation



The directory structure for AP provision mode application is as follows.

Figure 3-2. P2P Mode Directory Structure

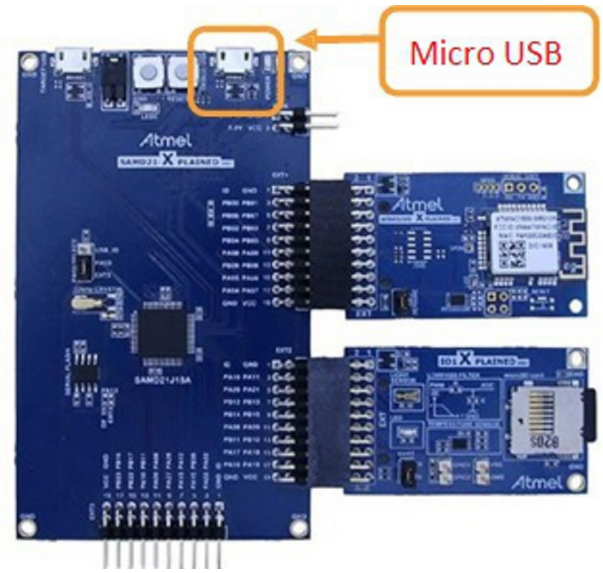


3.2. Programming the SAM D21 Xplained Pro

To dowload the firmware from PC to ATWINC1500, use the firmware update application provided in the ASF.

1. Connect the SAM D21 Xplained Pro board using the ATWINC1500 EXT1 header as shown.

Figure 3-3. SAM D21 Xplained Pro Board with ATWINC1500



2. Connect the USB cable to EDBG port of the SAM D21 Xplained Pro board.
3. Compile and program the ATWINC1500 ASF application using Atmel Studio.
4. To download or upgrade new firmware into ATWINC1500 module, follow the steps specified in the Quick Start Guide.

3.3. Executing the P2P mode Application

This example demonstrates execution of ATWINC1500 as a Wi-Fi P2P mode, using SAM D21 Xplained Pro board as host MCU.

This example uses the following hardware:

- Atmel SAM D21 Xplained Pro board
- ATWINC1500 on EXT1 header
- The Wi-Fi Direct or P2P supported WLAN device

Figure 3-4. Demo Setup

1. P2P mode demo, ATWINC1500 starts as a P2P peer device in listen mode. In listen mode ATWINC1500 cannot start the connection itself, but it responds for the other P2P peer invitation.

P2P listen mode channel can be specified by `MAIN_WLAN_CHANNEL` macro its defined in `main.h` file.

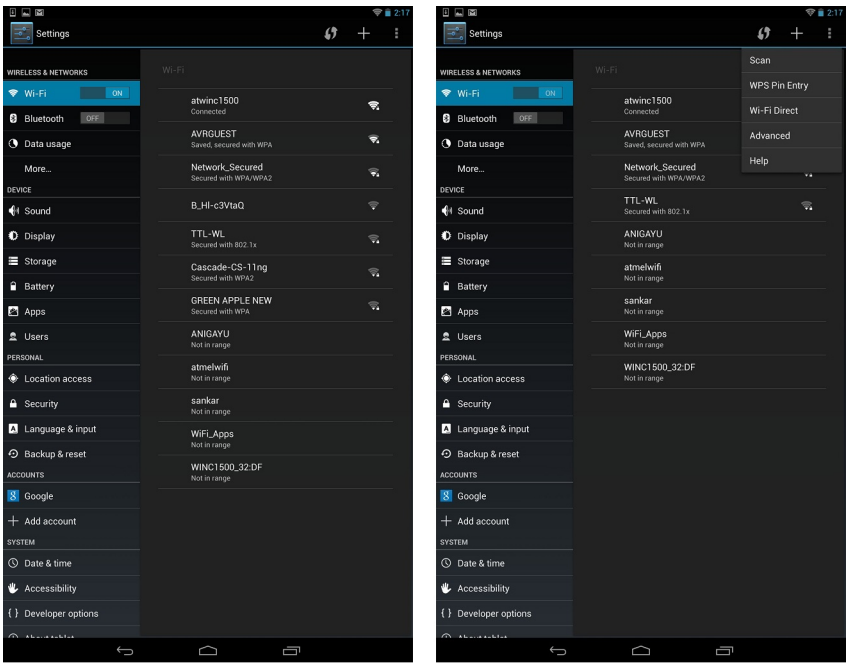
```
/** P2P mode Settings */
#define MAIN_WLAN_DEVICE_NAME    "WINC1500_P2P" /* < P2P Device Name */
#define MAIN_WLAN_CHANNEL        M2M_WIFI_CH_6 /* < P2P Channel number */
```

2. Open serial port terminal application with the COM port configuration 115200,8,none,1,none.
3. Compile and download the image into the SAM D21 XPRO board.

Figure 3-5. Atmel Studio Debug Button

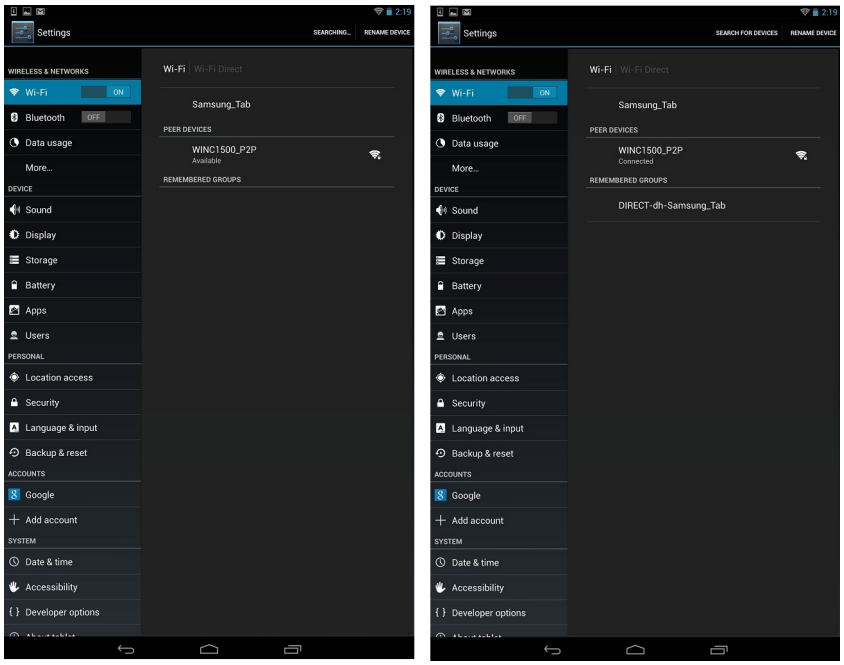
4. Run the application, success or error messages appears in the serial port terminal.
5. ATWINC1500 starts the P2P listen mode and other devices can initiate and connect to the ATWINC1500.
6. Any Wi-Fi Direct or P2P supported device, option is available in the **Wi-Fi home page** > **Options** > **Wi-Fi Direct** as shown.

Figure 3-6. P2P Mode Option in Android Device



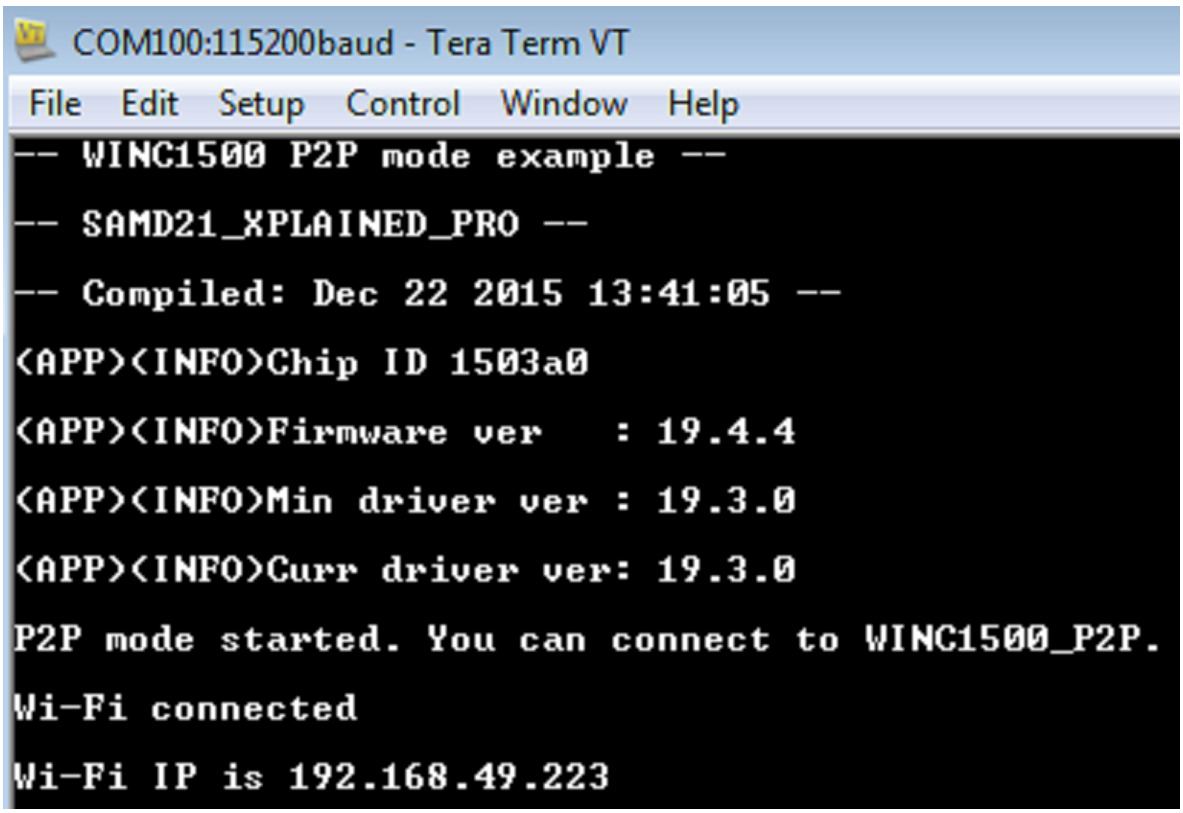
7. Enable the **Wi-Fi Direct** options, the P2P peer device starts active scanning and list the available P2P supported peer devices.
8. The P2P peer device lists the ATWINC1500 P2P device by the device name configured.
9. Initiate the P2P connection with ATWINC1500. This P2P connection is based on the PushButton configuration method configured by default in the ATWINC1500 device.
10. Connection between the P2P peer device and ATWINC1500 P2P device is established successfully after the sequence of protocol frame exchange.

Figure 3-7. ATWINC1500 Connected with P2P Peer



11. After the P2P connection ATWINC1500 sends the DHCP request to obtain the IP address and DHCP server of the P2P peer device provides the IP address.
12. Successful P2P connection logs are shown.

Figure 3-8. P2P Mode Terminal Log



4. Revision History

Doc Rev.	Date	Comments
42643A	08/2016	Initial document release.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**