Atmel SmartConnect

AT12264: ATWINC1500 Wi-Fi Network Controller - Station Mode

APPLICATION NOTE

Introduction

This application note describes the Wi-Fi Station Mode demo with the Atmel® ATWINC1500 Wi-Fi® Network Controller to build state-of-the-art Internet of Things (IoT) applications.

The following topics are covered:

- Organization of demo application
- Information about Target board
- Flow of the demo application and step-by-step execution of the API's

Features

- ATWINC1500 Host driver architecture
- ATWINC1500 internal architecture
- WLAN connection process with different security methods supported by ATWINC1500
- Events handled in the Wi-Fi callback function with appropriate structures used for each events
- Steps to run the station mode application demonstration using SAM D21 XPRO board using ATWINC1500

Table of Contents

1. Overview

This Wi-Fi Station mode application primarily demonstrates a configuration with Wi-Fi credentials (such as, SSID (Service Set Identifier) and Pass-phrase or Password) in ATWINC1500 host application driver. These credential are used to connect with desired access point using the ATWINC1500-XSTK. After the successful connection establishment with AP (Access Point), Connection can be verified with basic ping operation from any of the network nodes. For ping operation, ping-free apps from Android device or command prompt ping tool can be used from the PC.

ATWINC1500 has additional remote configuration facility such as AP provisioning, HTTP provisioning modes applications to configure the Wi-Fi credentials SSID and Paraphrase.

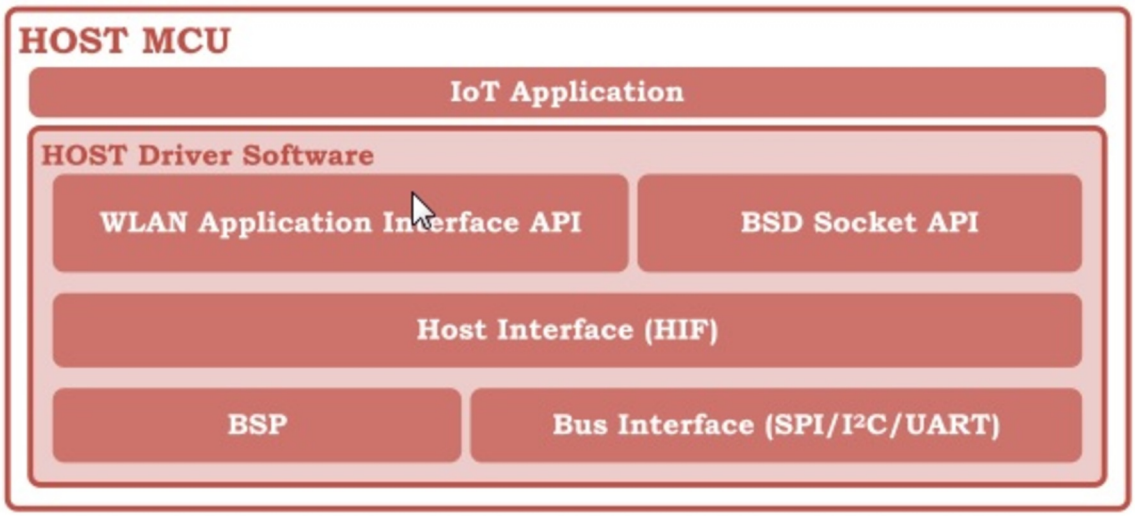
- AP credentials configured in the host driver application `main.h` file of ATWINC1500 station demo application will be used for connection with AP
- ATWINC1500 will initialize the station mode application and scan operation will be initiated to find the desired AP in the BSS area
- ATWINC1500 host application will parse the scan results to find the SSID and security methods of the access points in the BSS area to finds the configured AP
- Connection process will be initiated, when the matching AP is found with the with desired credentials configuration

Figure 1-1. Sample Demo Setup

2. Host Driver Architecture

ATWINC host driver software is a C library which provides the host MCU application with necessary APIs to perform necessary WLAN and socket operations. It shows the architecture of the WINC host driver software which runs on the host MCU. The components of the host driver are described in ATWINC1500 Wi-Fi Network Controller - Software Design Guide.

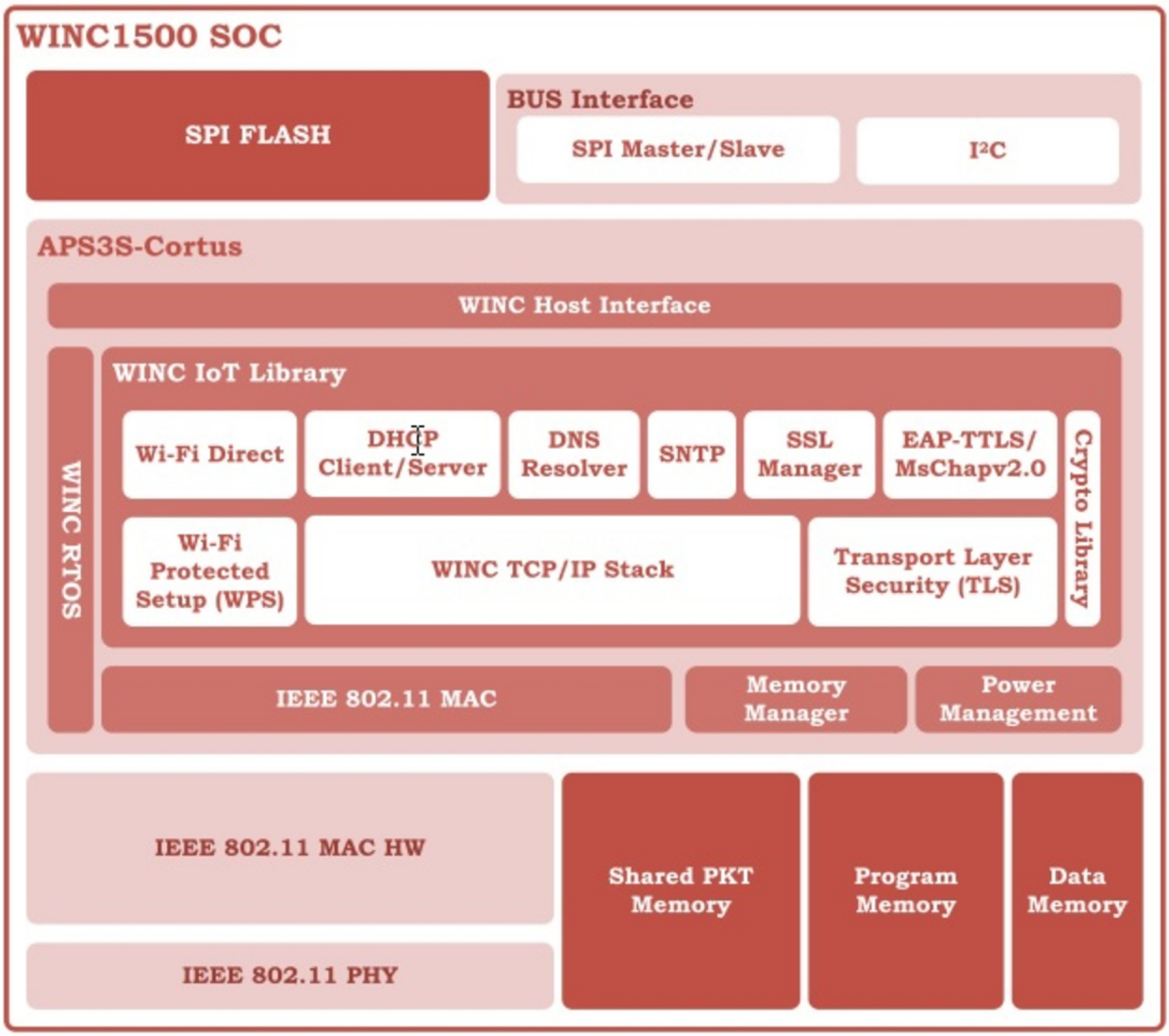
Figure 2-1. Host Driver Architecture



3. ATWINC1500 System Architecture

In addition to its built-in Wi-Fi IEEE®-802.11 physical layer and RF front end, the ATWINC ASIC has an embedded APS3S-Cortus 32-bit CPU to run the ATWINC firmware. The firmware comprises the Wi-Fi IEEE-802.11 MAC layer and embedded protocol stacks which offload the host MCU. The components of the system are described in the sub-sections of ATWINC1500 Wi-Fi Network Controller - Software Design Guide.

Figure 3-1. ATWINC System Architecture



4. Application Description

This section describes the ATWINC1500 host driver station mode application in detail. WLAN connection process is explained with respect to the ATWINC1500 API's sequence.

What is Wi-Fi Station?

In IEEE 802.11 (Wi-Fi) terminology, a station (STA) is a device that has the capability to use the 802.11 protocol. For example, a station may be a laptop, a desktop PC, PDA, access point, or Wi-Fi phone. A STA may be fixed, mobile, or portable. Generally in wireless networking terminology, a station also calls wireless client and node. With a station also being referred as transmitter or receiver based on its transmission characteristics. IEEE 802.11-2007 formally defines station as: Any device that contains an IEEE 802.11-conformant media access control (MAC) and physical layer (PHY) interface to the wireless medium (WM). WLAN station can operate in Infrastructure mode and Ad-Hoc or Peer-to-Peer mode.

What is BSS?

The basic service set (BSS) provides the basic building-block of an 802.11 wireless LAN. In infrastructure mode, a single AP together with all associated stations (STAs) is called a BSS. This must not to be confused with the coverage of an access point, known as the basic service area (BSA). The access point acts as a master to control the stations within that BSS; the simplest BSS consists of one access point and one station.

What is SSID?

Each BSS or ESS (Enhanced Service set - one or more interconnected BSS with LAN) is identified by a service set identifier (SSID) - a series of 0 to 32 octets. It is used as a unique identifier for a wireless LAN. Since this identifier must often be entered into devices manually by a human user, it is often a human-readable string and commonly called the *network name*.

What is BSSID?

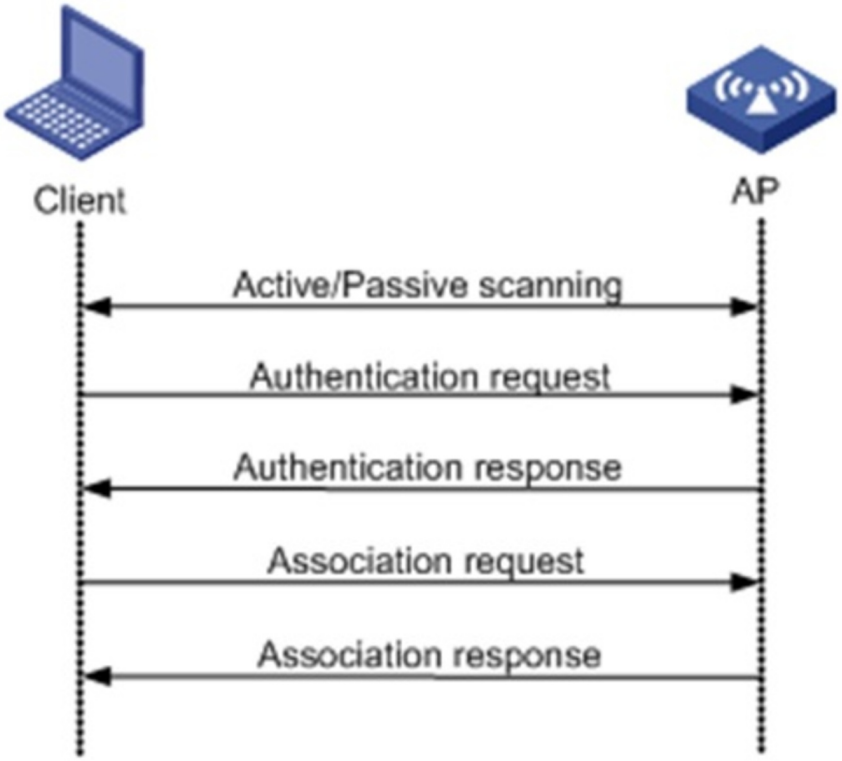
Each BSS is uniquely identified by a basic service set identification (BSSID). For a BSS operating in infrastructure mode, the BSSID is the MAC address of the wireless access point (WAP) generated by combining the 24-bit Organization Unique Identifier (the manufacturer's identity) and the manufacturer's assigned 24-bit identifier for the radio chip-set in the WAP. The BSSID is the formal name of the BSS and is always associated with only one BSS.

4.1. Open Connection Process

The following sequence is followed by ATWINC1500 during the WLAN connection process.

- Scanning
- Authentication Request and Response
- Association Request and Response

Figure 4-1. Generic Connection Process

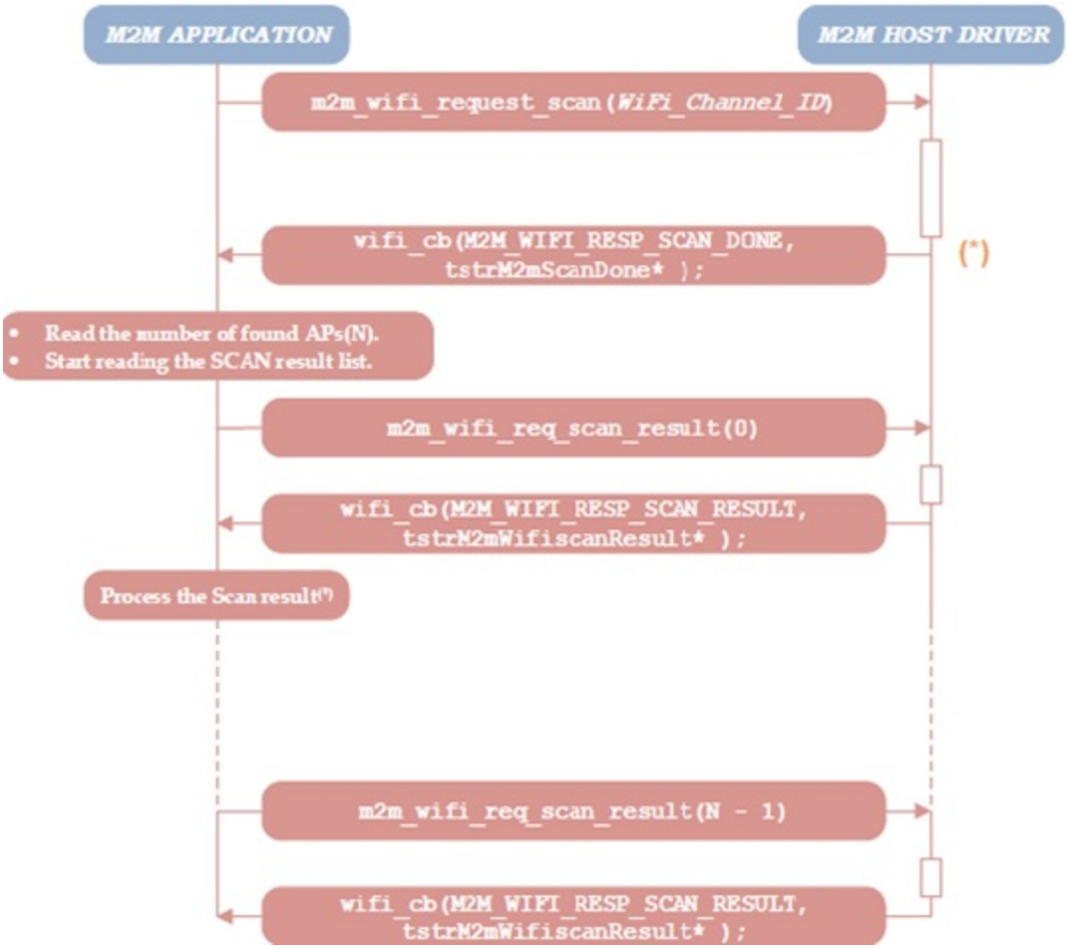


4.1.1. Scanning Process

WLAN scanning includes the following process:

1. ATWINC1500 WLAN module will send the Probe-Request broadcast frames in all the WLAN channels to scan the access points available nearby.
2. Access points available nearby will send the Probe-Response frames with their SSID, security information, channel with other information elements.
3. ATWINC1500 WLAN module will make scan results from the received Probe-Response frames. From the available scan results, it will filter the access point having the matching SSID and security type access point.
4. When found the required access point, ATWINC1500 will send the specific Probe-Request with SSID and security type configured before starting the association process.

Figure 4-2. Scan Operation Process

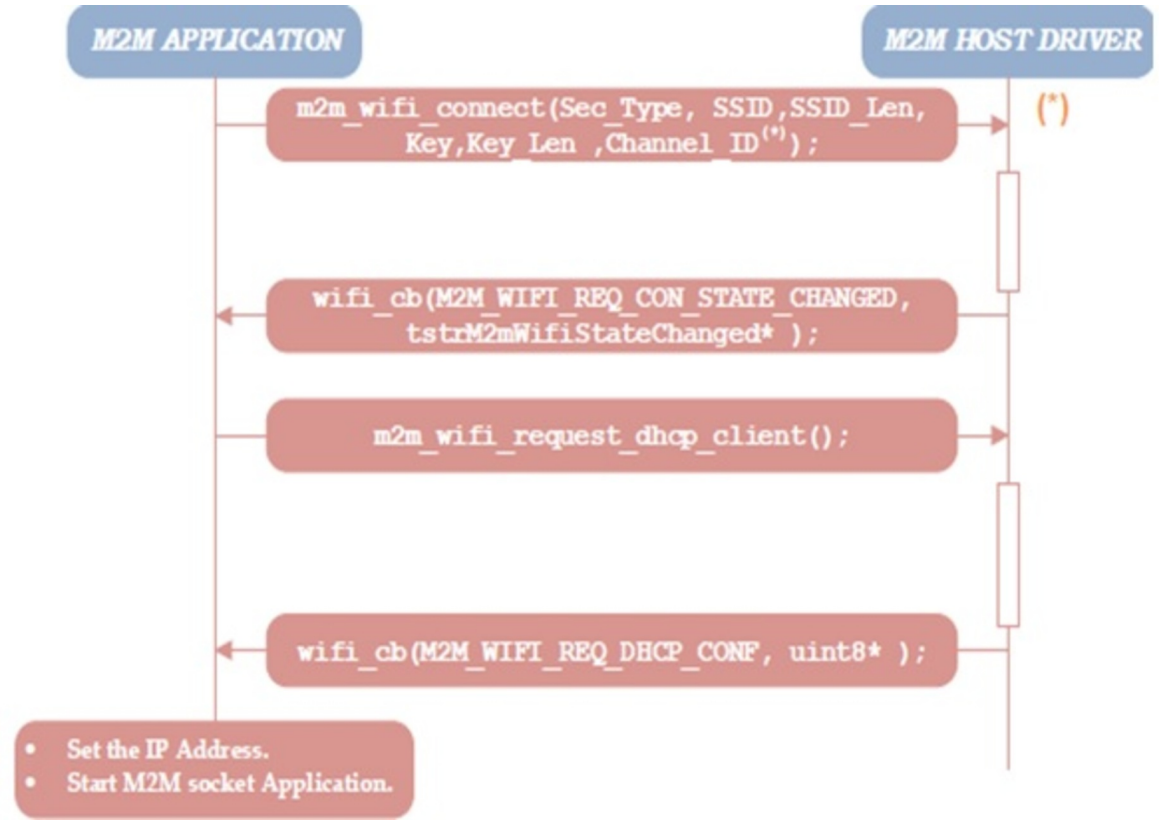


4.1.2. Authentication and Association Process

WLAN association process occurs in the following sequence.

1. Corresponding AP will respond for the specific probe request with BSS information in probe response frame. ATWINC1500 will start the association process with the confirmation from the access point.
2. Authentication request will be sent using the configured authentication method. Usually open authentication will be used. Except WEP shared security method. Access point responds with authentication success frame.
3. The initial purpose of the authentication frame is to validate the device type and verify that the requesting station has proper 802.11 capability to join the network.
4. This exchange is based on simple two-frame (Authentication Request and Response) called Open System.
5. Association request will be sent followed by the authentication success and AP will respond the association success.

Figure 4-3. WINC Association Flow



4.2. Wi-Fi Host Driver Initialization

- System Initialization of SAM D21 XPro board. In this MCU's clock, hardware, events, and external hardware if any will be initialized.

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

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

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

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

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

- Wi-Fi host driver initialization start with API `m2m_wifi_init()` and structure `tstrWifiInitParam` filled with appropriate information. Wi-Fi Initialization sequence will configure the SPI communication interface and external interrupt with respect to the host MCU and ATWINC1500 pin connections. Apart from this Wi-Fi host application layer callback function `wifi_cb()` will also be registered with the initialization sequence.
- After successful initialization of communication interface between the ATWINC1500 and host MCU, able to read the ATWINC1500 chip-id which indicates that ATWINC1500 ready to accept the Wi-Fi

initialization and configuration commands of the WLAN module. In this process, ATWINC1500 WLAN module will be reset by the host MCU and wait for the module firmware to start confirmation.

- The purpose of the Wi-Fi application callback function is to indicate the events such as connect and disconnect status, obtaining the IP address from the DHCP server with respect to the DHCP request from the ATWINC1500 station. When ATWINC1500 act as softAP, providing the IP address to the WLAN client will also be indicated.

```
/* 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) {
        }
    }
```

4.3. Wi-Fi Callback Function

In station mode, the Wi-Fi callback function will be called whether success or failure 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:{
    ............................
}
```

Wi-Fi callback function will be called in the specified scenarios. Various type of events with respect to the `Wi-Fi_cb()` registered with respective ATWINC1500 firmware API's.

Table 4-1. 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

Wi-Fi Callback Events	Structure used for the Events	Comments
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.

4.4. ATWINC1500 Supported Connection Methods

In the following sections, ATWINC1500 supported Wi-Fi security based connection process is explained in detail.

4.4.1. No Security Method

In no security mode or open connection method, password is not required and connection is based on the open authentication type. Configure the AP credentials as mentioned. The `m2m_wifi_connect()` API will establish the connection between the desired AP with ATWINC1500 Wi-Fi client. Basic connection process is explained in the Open Connection Process on page 6 section.

```
/** No security information for Wi-Fi connection */
#define MAIN_WLAN_SSID  "AtmelWifi" /**< Destination SSID */
#define MAIN_WLAN_AUTH  M2M_WIFI_SEC_OPEN /* < Security manner */
```

```
/* Starting the wifi connect process using the AP's Credentials*/
printf("Connecting to %s.\r\n", (char *)str_ssid);
m2m_wifi_connect((char *)MAIN_WLAN_SSID,
                 strlen((char *)MAIN_WLAN_SSID),
                 MAIN_WLAN_AUTH, NULL,
                 M2M_WIFI_CH_ALL);
```

4.4.2. Security Methods Supported in ATWINC1500

The following table summarizes the ATWINC1500 security methods with respect to the WLAN security methods.

Table 4-2. Security Methods Supported

ATWINC1500 security type macro	ATWINC1500 security manners	WLAN security methods	Comments
MAIN_WLAN_AUTH	M2M_WIFI_SEC_OPEN	No Security	WLAN Open Connection
	M2M_WIFI_SEC_WEP	WEP 40 Bit - Open	WLAN WEP security in Open authentication mode
		WEP 104 Bit - Open	
		WEP 40 Bit - Shared	WLAN WEP security in Shared authentication mode
		WEP 104 Bit - Shared	
	M2M_WIFI_SEC_WPA_PSK	WPA-PSK/TKIP	WPA Personal security methods in Open authentication mode
		WPA-PSK/AES	
		WPA-PSK/TKIP or AES	
		WPA2-PSK/TKIP	
		WPA2-PSK/AES	
		WPA2-PSK/TKIP or AES	
	M2M_WIFI_SEC_802_1X	EAP-TTLS/ MSCHAPv2	WPA Enterprises Security methods will work in Open and Shared Authentication Modes

4.4.3. Demonstrating WEP Security Mode

The configuring WEP security method parameters with respect to the AP configuration on ATWINC1500 as explained. The `m2m_wifi_connect()` API will be used to establish the connection between the desired AP and the ATWINC1500 Wi-Fi client.

4.4.3.1. WEP Key Format

Supported WEP key values are Alphanumeric or Hexadecimal. An alphanumeric character is 'a' through 'z', 'A' through 'Z', and '0' through '9'. A hexadecimal digit is '0' through '9' and 'A' through 'F'. There are two types of WEP security key formats available 64-bit key and 128-bit key.

The 64-bit WEP data encryption method allows for a five-character (40-bit) input. Additionally, 24 factory-set bits are added to the 40-bit input to generate a 64-bit encryption key. The 24 factory-set bits are not user-configurable. This encryption key will be used to encrypt/decrypt all data transmitted via the wireless interface. Some vendors refer to the 64-bit WEP data encryption as 40-bit WEP data encryption since the user-configurable portion of the encryption key is 40-bits wide.

The 128-bit WEP data encryption method consists of 104 user-configurable bits. Similar to the 40-bit WEP data encryption method, the remaining 24-bits are factory set and not user configurable. Some

vendors allow passphrases to be entered instead of the cryptic hexadecimal characters to ease encryption key entry.

When configured for 40-bit encryption, 802.11 products typically support up to four WEP Keys. Each 40-bit WEP Key is expressed as five sets of two hexadecimal digits (0-9 and A-F). For example, "12 34 56 78 90" is a 40-bit WEP Key.

When configured for 128-bit encryption, 802.11 products typically support four index WEP Keys but in ATWINC1500 the user can only set any one of the index keys. The 128-bit WEP Key is expressed as 13 sets of two hexadecimal digits (0-9 and A-F). For example, "12 34 56 78 90 AB CD EF 12 34 56 78 90" is a 128-bit WEP Key.

Table 4-3. WEP Key Sample Formats

Encryption Key Size	Number of Hexadecimal Digits	Example of Hexadecimal Key Content
64-bit (24+40)	10	4C72F08AE1
128-bit (24+104)	26	4C72F08AE19D57A3FF6B260037

4.4.3.2. WEP Security Mode of Operation

In the connection process, the mentioned sequence will be followed by ATWINC1500. In WINC there is no provision for configuring authentication type whether its OPEN or SHARED. SHARED authentication or OPEN authentication type will be decided according to the beacon frame information from the AP. WINC firmware will parse from beacon frame and proceed accordingly.

```
/** security information for Wi-Fi connection */
#define MAIN_WLAN_DEVICE_NAME           "AtmelWifi" /**< Destination SSID */
#define MAIN_WLAN_AUTH                  M2M_WIFI_SEC_WEP /* < Security manner */
#define MAIN_WLAN_WEP_KEY_INDEX         1 /**< WEP key index */
#define MAIN_WLAN_WEP_KEY_40            "1234567890" /**< 64 bit WEP key. In case
of WEP64, 10 hexadecimal (base 16) characters (0-9 and A-F) ) */
#define MAIN_WLAN_WEP_KEY_104           "1234567890abcdef1234567890" /**< 128 bit
WEP key. In case of WEP128, 26 hexadecimal (base 16) characters (0-9 and A-F) ) */
```

In WEP security mode, structure `tstrM2mWifiWepParams` must be filled with WEP key index (1 to 4), size of the key (40 or 104 bit key), WEP key password string. This structure will be passed as an argument of `m2m_wifi_connect()` API to initiate the connection to the AP.

```
/** Security parameters for 64 bit WEP Encryption @ref m2m_wifi_connect */
tstrM2mWifiWepParams wep64_parameters = { MAIN_WLAN_WEP_KEY_INDEX,
                                          sizeof(MAIN_WLAN_WEP_KEY_40),
                                          MAIN_WLAN_WEP_KEY_40};

/** Security parameters for 128 bit WEP Encryption @ref m2m_wifi_connect */
tstrM2mWifiWepParams wep128_parameters = { MAIN_WLAN_WEP_KEY_INDEX,
                                           sizeof(MAIN_WLAN_WEP_KEY_104),
                                           MAIN_WLAN_WEP_KEY_104};
```

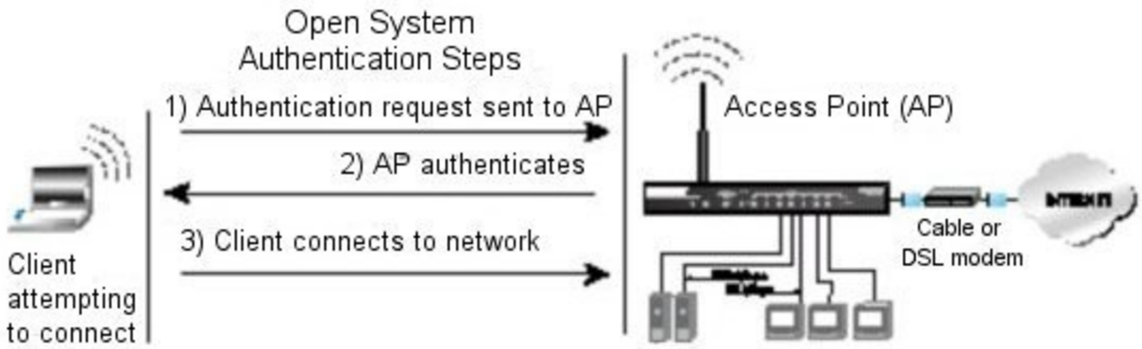
```
 /* Strating the wifi connect process using the AP's Credentials*/
            printf("Connecting to %s.\r\n", (char *)str_ssid);
        m2m_wifi_connect((char *)MAIN_WLAN_DEVICE_NAME,
                         strlen((char *)MAIN_WLAN_DEVICE_NAME),
                         MAIN_WLAN_AUTH,
                         &wep64_parameters,
                         M2M_WIFI_CH_ALL);
```

WEP security method with open authentication type connection process will be same as the Open Connection Process on page 6.

4.4.3.3. Open WEP Security Method

1. In WEP security method, OPEN authentication mode Authentication frame request will be sent with configured authentication method and empty challenge text field of frame to AP.
2. AP will respond to the authentication frame immediately when the challenge text filed is empty for Open authentication type.

Figure 4-4. WEP Open Authetication Sequence



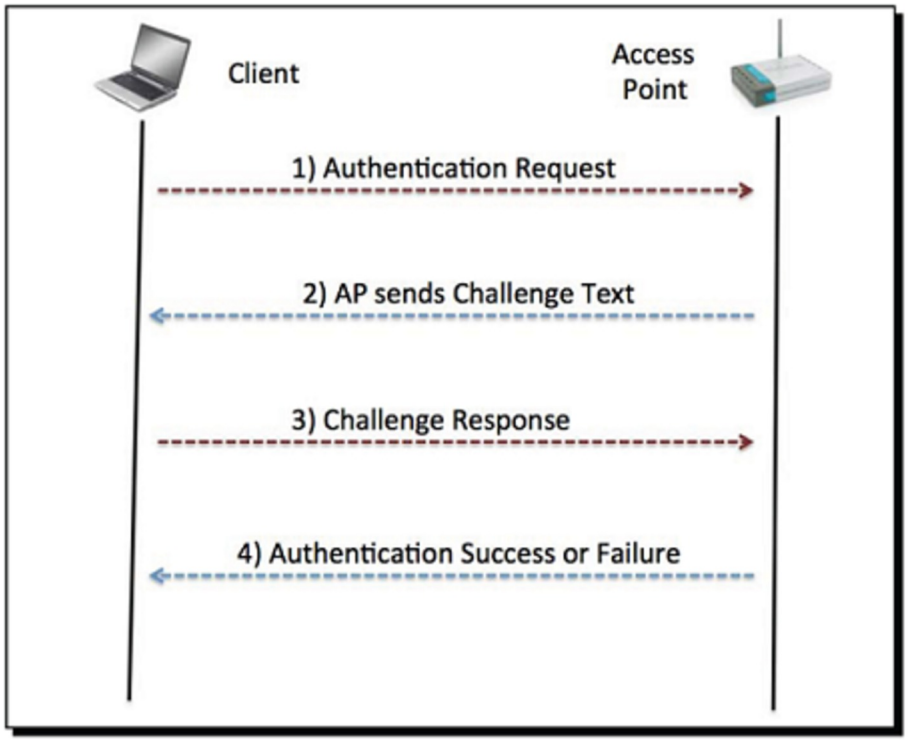
3. In WEP OPEN security method, key verification will occur only during the time of data transfer. The association sequence key will not be verified.

4.4.3.4. Shared WEP Security Method

This section elaborates the WEP shared mode operation,

1. Totally four authentication request and response frames will be exchanged in the WEP shared key authentication process.
2. In WEP security method, Shared authentication mode - Authentication frame request will be sent with configured authentication method and empty challenge text field to AP.
3. If authentication type is shared key, AP will respond for the authentication frame immediately with challenge text.

Figure 4-5. WEP Shared Key Authentication Sequence



4. Station will encrypt the challenge text and will be sent to the AP as a challenge text response.
5. If the encrypted challenge text will be decrypted again in the AP and will be verified with the original WEP shared key.
6. When WEP key matches, then AP will send the authentication response with success.
7. If the WEP does not match, the failure message will be sent and association process will be terminated.

4.4.4. WPA/WPA2 Security Connection Process

In WPA/WPA2 security mode, passphrase should be configured as mentioned with AP SSID and connection is based on the open authentication type. The `m2m_wifi_connect()` API will be used to establish the connection between the desired AP with ATWINC1500 Wi-Fi client. Basic connection process explained in the Open Connection Process on page 6 section.

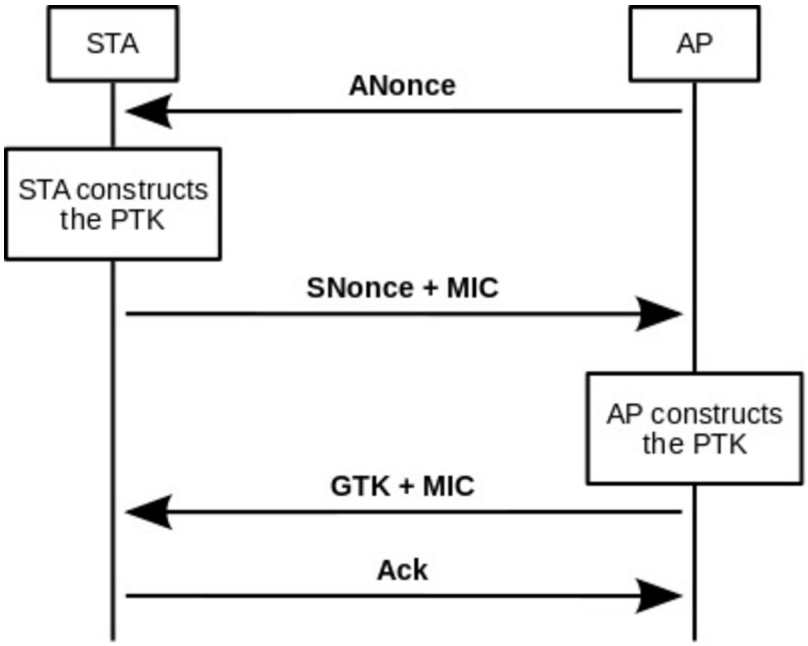
```
/** WPA/WPA2 security information for Wi-Fi connection */
#define MAIN_WLAN_SSID    "atmelwifi" /* < Destination SSID */
#define MAIN_WLAN_AUTH    M2M_WIFI_SEC_WPA_PSK /* < Security manner */
#define MAIN_WLAN_PSK     "1234567890" /* < Password for Destination SSID */
```

```
/* Strating the wifi connect process using the AP's Credentials*/
printf("Connecting to %s.\r\n", (char *)str_ssid);
m2m_wifi_connect((char *)MAIN_WLAN_SSID,
                 strlen((char *)MAIN_WLAN_SSID),
                 MAIN_WLAN_AUTH, NULL,
                 M2M_WIFI_CH_ALL);
```

- After the open mode WLAN connection and the WPA-PSK 4-way key hand shake will occur and complete the connection process. In the 4-way key hand shake, passphrase will be shared and verified by the access point. The ATWINC1500 will establish the connection with desired AP.

- WPA/WPA2 will support for 8 to 63 characters passphrase or 64 character PSK (Pre Shared Key). PSK can be generated using software tools, it requires the SSID and passphrase of the AP configuration. PSK generate tools are available in web based tool and freewares.

Figure 4-6. WPA/WPA2-PSK 4-Way Key Handshake Process



4.5. Wi-Fi Host Driver Event and Callback Handling

All the Wi-Fi host driver events are handled in the `m2m_wifi_handle_events()` by running in the infinite loop. This function internally uses the HIF (Host Hardware Interface) layer API to monitor the ATWINC1500 external interrupt. This external interrupt is registered using host MCU external interrupt configuration.

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

When ATWINC1500 external interrupt occurs, host interface ISR layer reads the ATWINC1500 control register to identify the type of event which triggered the external interrupt by ATWINC1500 and reads the data related to the event as well from the ATWINC1500 buffer. After reading the complete data for event corresponding callback function will be triggered. Host MCU Wi-Fi driver handle three types of event categories such as `M2M_REQ_GRP_WIFI,M2M_REQ_GRP_IP,M2M_REQ_GRP_OTA`.

- `m2m_wifi_cb` handles all the Wi-Fi configuration and connection events.
- `m2m_i_cb`handles all the socket, and network application event callbacks.
- `m2m_ota_cb`handles all the *Over-The-Air* firmware upgrade events.

5. How to Run the Station Mode Application

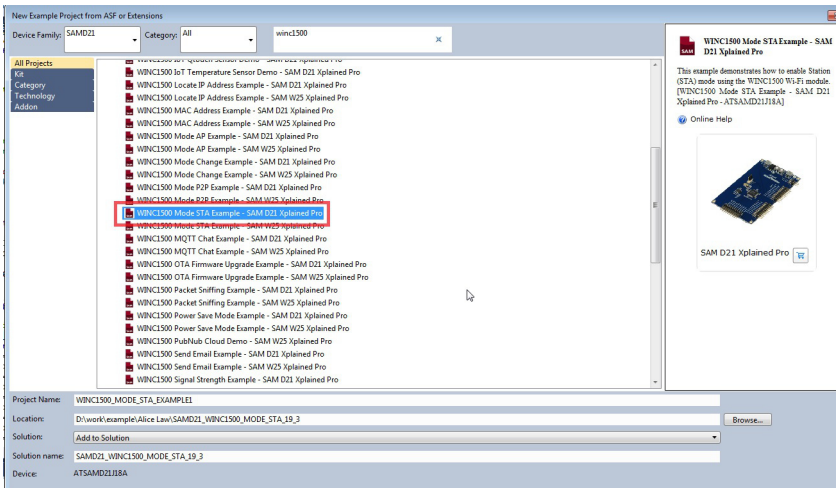
This topic elaborates the steps to run the station application using SAM D21 Xplained Pro board using ATWINC1500 WLAN module.

5.1. Getting Started ASF ATWINC1500 Station Mode Demo

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

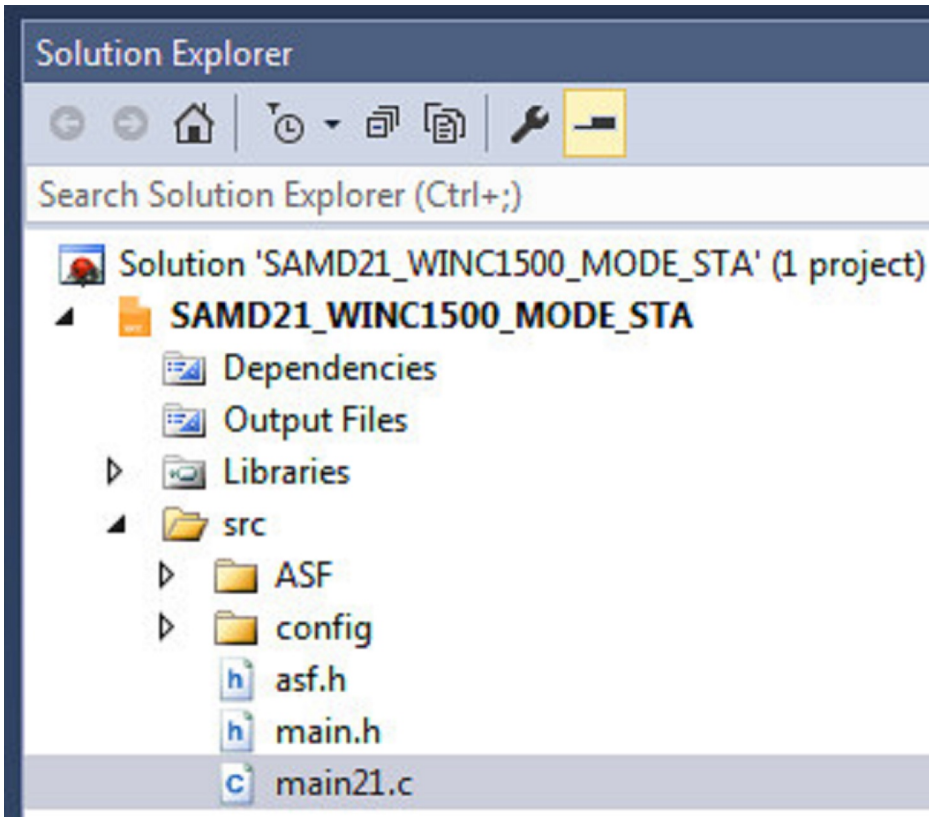
- Open Atmel Studio 7 **File** > **New** > **Example Projects**
- In the search tab, enter *ATWINC1500* sample application for different MCU
- Select the Station mode `WINC1500_MODE_STA_EXAMPLE` project for SAM D21 and open the project

Figure 5-1. Atmel Studio ATWINC1500 Project Creation



- Station Mode Application Directory Structure is as follows

Figure 5-2. Station Mode Directory Structure



5.2. Programming SAM D21 XPRO and ATWINC1500

To program ATWINC1500 using serial bridge application on MUC and *image_downloader* from PC tool application.

1. Connect the SAM D21 XPRO board with ATWINC1500 EXT1 header as shown.

Figure 5-3. SAM D21 XPRO Board with ATWINC1500

2. Connect the USB cable to EDBG port of the SAM D21 XPRO board.

3. To download or upgrade new release firmware in to ATWINC1500 module, follow the steps specified in the Quick Start Guide.

5.3. Executing Station Mode Application

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

The example uses the following hardware:

- The SAM D21 Xplained Pro
- The ATWINC1500 on EXT1 header
- 802.11 b/g/n supported AP or Router

Figure 5-4. Demo Setup

Station mode demo requires the Wi-Fi credentials such as SSID and Security type to start the connection with access point. This demo is explained based on the WPA-PSK security method in OPEN authentication type. To set a security type `MAIN_WLAN_AUTH` macro value to be configured as mentioned. Corresponding AP's SSID should filled `MAIN_WLAN_SSID` macro as given. These macros are defined in the `main.h` file.

1. Configure the AP credentials SSID and password as mentioned.
2. This application demo based on the WPA/WPA2-PSK method. Configure the AP based on the supported security methods.

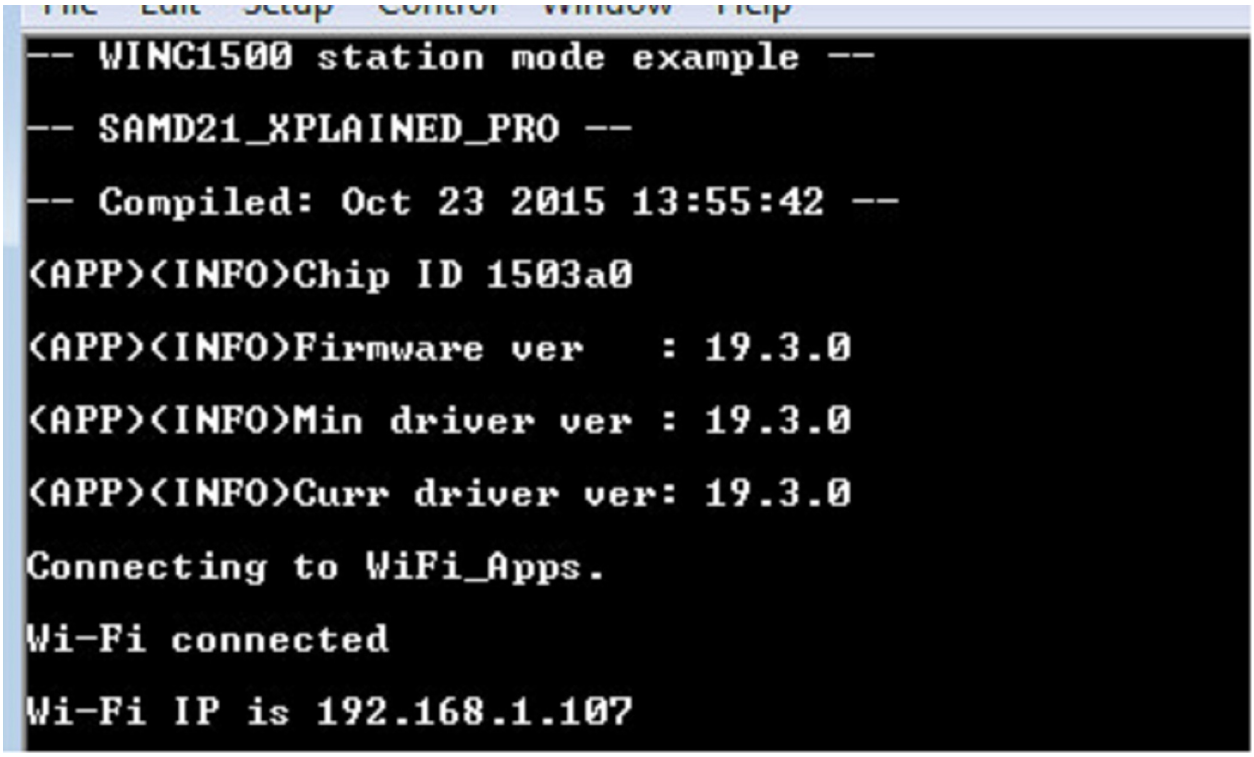
```
 /* AP configuration parameters*/
#define MAIN_WLAN_SSID       "AtmelWifi" /* Access Point SSID */
#define MAIN_WLAN_AUTH       M2M_WIFI_SEC_WPA_PSK    /* Security manner */
#define MAIN_WLAN_PSK        "1234567890" /* < Passphrase for Destination SSID
*/
```

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

Figure 5-5. Atmel Studio Debug Button

5. Run the application, success or error messages appear in the serial port terminal.

Figure 5-6. Station Mode Output Log



6. Revision History

Doc. Rev.	Date	Comments
42630A	01/2016	Initial document release.

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