Atmel SmartConnect

AT12265: ATWINC1500 Wi-Fi Network Controller - HTTP Provision Mode

APPLICATION NOTE

Introduction

This application note explains how to build the state-of-art Internet of Things (IoT) applications using the Wi-Fi® HTTP Provision Mode with the Atmel® ATWINC1500 Wi-Fi Network Controller.

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

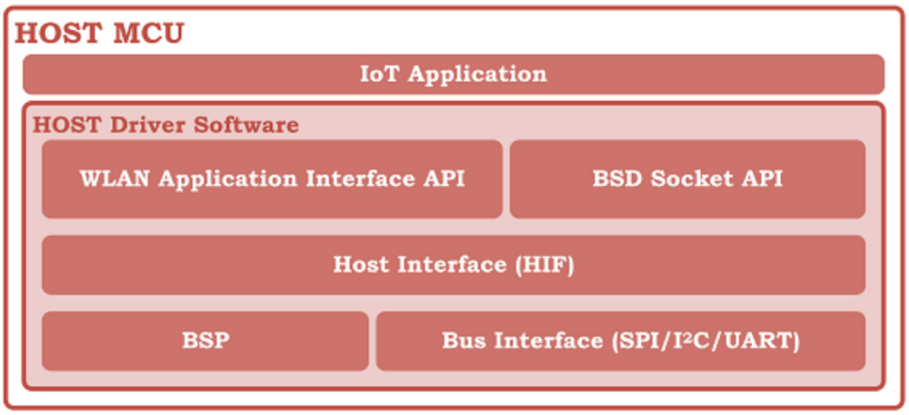
- ATWINC1500 host MCU driver architecture
- ATWINC1500 internal architecture
- Application description with code snippet
- Events handled in the Wi-Fi callback function with appropriate structures used for each events
- Steps to execute the HTTP provision mode application demo using SAM D21 Xplained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

Table of Contents

1. Host Driver Architecture

The ATWINC1500 host driver software is a C library. It provides the host MCU application with necessary APIs to perform WLAN and socket operations. It shows the architecture of the ATWINC1500 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.

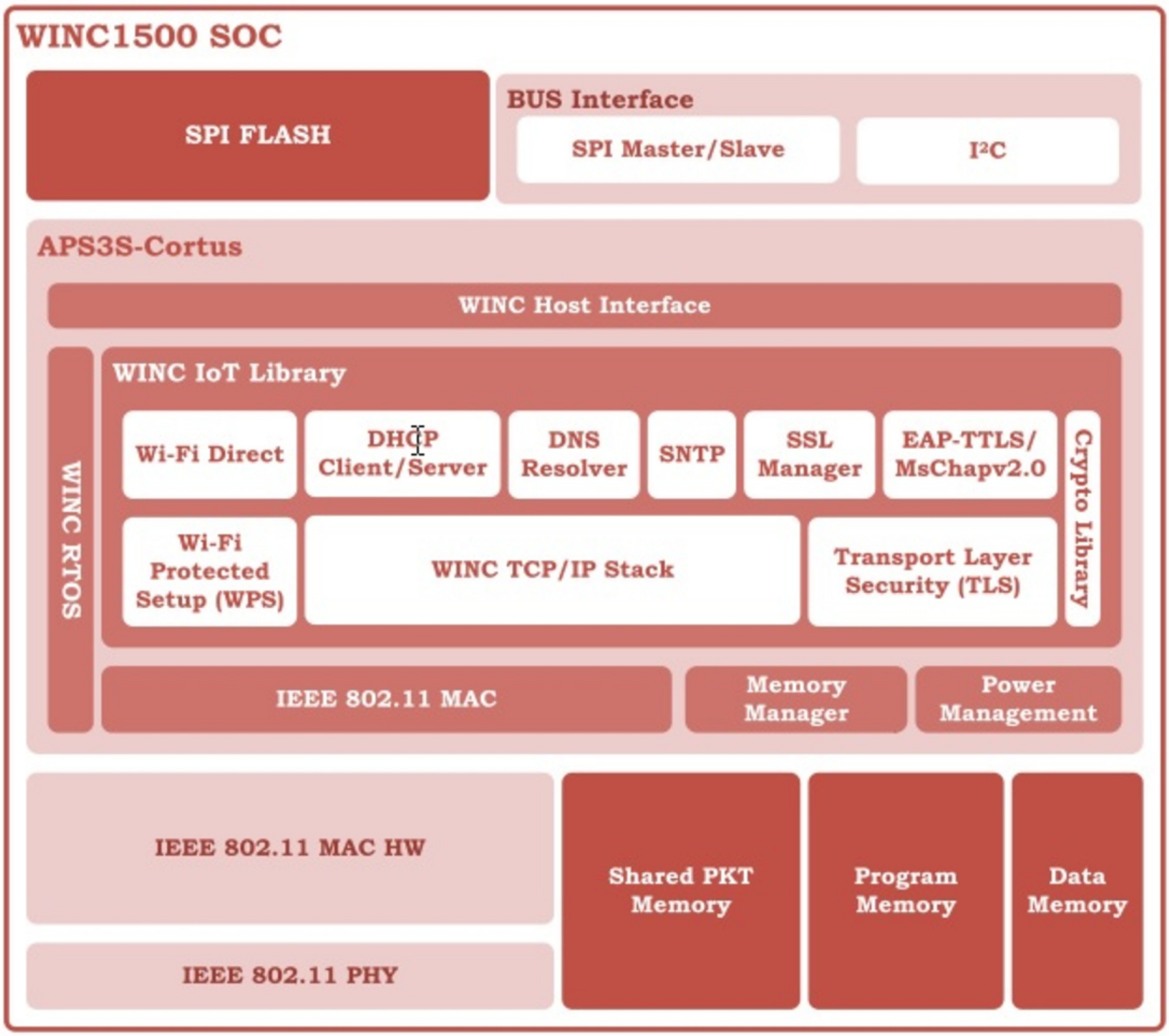
Figure 1-1. Host Driver Architecture



2. ATWINC1500 System Architecture

ATWINC1500 has a built-in Wi-Fi, IEEE®-802.11 physical layer and RF front end, and ASIC has an embedded APS3S-Cortus 32-bit CPU to run the ATWINC1500 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 2-1. ATWINC System Architecture



3. Application Overview

What is Wi-Fi Provisioning?

Wi-Fi provisioning is the process of connecting a new Wi-Fi device (station) to a Wi-Fi network. The provisioning process involves loading the station with the network name (often referred to as SSID) and its security credentials.

What is SoftAP?

SoftAP is an abbreviated term for "Software enabled Access Point". This is a software that enables a computer that is not designed to be a router into a wireless access point. It is often used interchangeably with the term "virtual router".

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. An STA may be fixed, mobile, or portable. In wireless networking terminology a station is also called wireless client and node. It is 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.

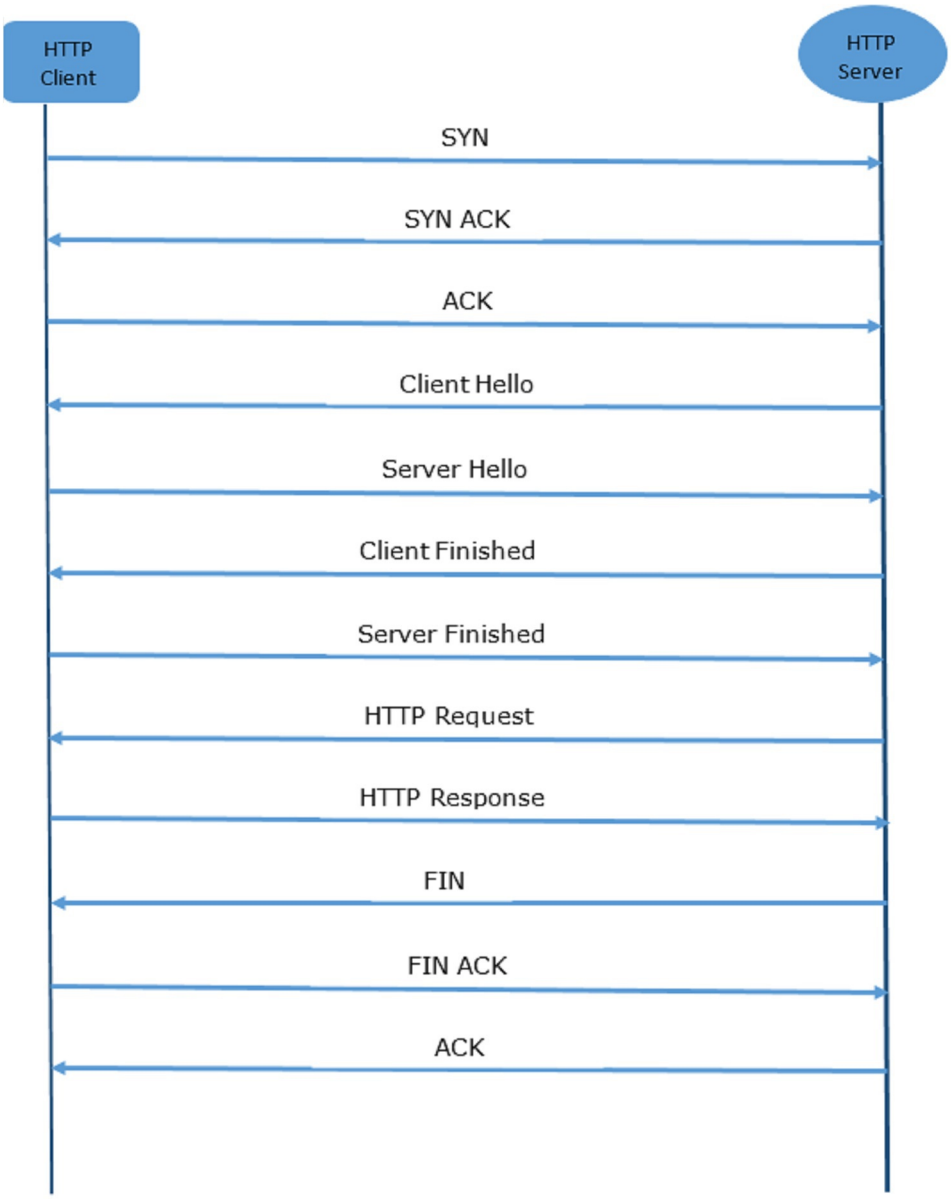
What is WINC1500 HTTP Provisioning?

Wi-Fi AP Provision mode primarily demonstrates on how to configure the credentials (such as, SSID and Passphrase) in ATWINC1500 remotely. The configured credentials are used to connect with a desired access point. This demo uses any WLAN client supported devices to configure the credentials and verify the connection using simple ping operation. For the ping operation, ping-free apps can be used.

Remote configuration facilities such as AP provisioning and HTTP provisioning modes are available. In HTTP provisioning mode, the HTTP page is used to configure the credentials. The HTTP server is running in the WINC firmware and the HTTP page is also stored in the WINC flash memory.

Figure 3-1. Demo Setup

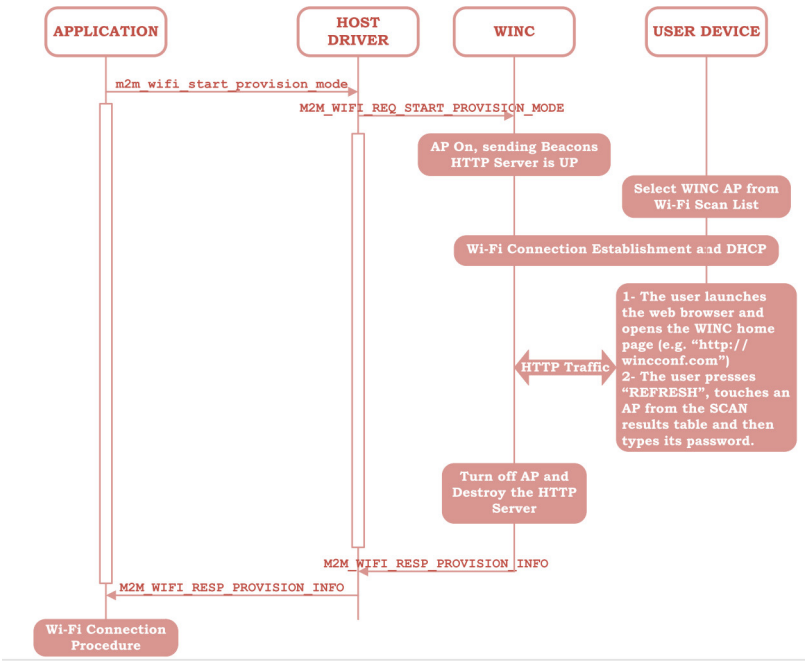
Figure 3-2. HTTP Protocol Sequence



4. Application Description

This chapter describes the ATWINC1500 host driver HTTP provision mode application in detail. The WLAN connection process is explained with respect to the ATWINC1500 API's sequence.

Figure 4-1. ATWINC1500 HTTP Provisioning Process



4.1. 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 completed, host MCU reads chip-id of ATWINC1500. It indicates that the WLAN module is ready for the initialization process which needs the module to be reset and waits for the confirmation from the module.

- The purpose of the Wi-Fi application callback function is to indicate 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 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) {
        }
    }
```

4.2. Configuring SSID Using MAC Address and Device Name

ATWINC1500 softAP SSID and device name are appended by the last two bytes of MAC address. The function `set_dev_name_to_mac` is used derive the SSID and device name from MAC address. Every ATWINC1500 module contains a MAC address stored in OTP. The user application can modify the MAC address stored in the OTP.

The same MAC address is used for framing the SSID of ATWINC softAP. SoftAP device name is set using function `m2m_wife_set_device_name()` and `gacDeviceName` array variable as argument.

```
static sint8 gacDeviceName[] = "WINC1500_00:00";
static uint8 gau8MacAddr[] = {0xf8, 0xf0, 0x05, 0x45, 0xD4, 0x84};
    /* Get the Mac address and setting the device */
       m2m_wifi_get_otp_mac_address(mac_addr, &u8IsMacAddrValid);
    if (!u8IsMacAddrValid) {
        m2m_wifi_set_mac_address(gau8MacAddr);
    }

    m2m_wifi_get_mac_address(gau8MacAddr);

    set_dev_name_to_mac((uint8_t *)gacDeviceName, gau8MacAddr);
    set_dev_name_to_mac((uint8_t *)gstrM2MAPConfig.au8SSID, gau8MacAddr);
    m2m_wifi_set_device_name((uint8_t *)gacDeviceName, (uint8_t)m2m_strlen((uint8_t
*)gacDeviceName));
```

4.3. Starting HTTP Provision Mode

The HTTP Provisioning mode is a special mode used to configure the AP credentials remotely. To start the HTTP provision mode, use the special API `m2m_wifi_start_provision_mode()` and the structure `tstrM2MAPConfig` with all requiredparameters. This combined API starts the ATWINC1500 as a softAP and initializes the HTTP network application to trigger the HTTP server.

ATWINC1500 uses the device name specified by the application as SSID of the softAP.

When the WLAN client is connecting to the softAP of ATWINC1500, it redirects to the provisioning web page *atmelconfig.com*. Enter the AP credentials such as SSID and passphrase.

Fill the required credentials and press the **Connect** button.. The HTTP provision page sends the details to the WINC1500 HTTP server.

```
#define MAIN_M2M_AP_SEC                    M2M_WIFI_SEC_OPEN /*No security Mode*/
#define MAIN_M2M_AP_WEP_KEY                "1234567890"      /*WEP security Key*/
#define MAIN_M2M_AP_SSID_MODE              SSID_MODE_VISIBLE /*SSID hidden mode disable*/
#define MAIN_HTTP_PROV_SERVER_DOMAIN_NAME  "atmelconfig.com" /*Credential configuration web
page*/
```

```
#define MAIN_M2M_DEVICE_NAME                "WINC1500_00:00"  /*Device Name*/
static CONST char gacHttpProvDomainName[] = MAIN_HTTP_PROV_SERVER_DOMAIN_NAME;

static tstrM2MAPConfig gstrM2MAPConfig = {
    MAIN_M2M_DEVICE_NAME, 1, 0, WEP_40_KEY_STRING_SIZE, MAIN_M2M_AP_WEP_KEY,
(uint8)MAIN_M2M_AP_SEC, MAIN_M2M_AP_SSID_MODE};
       /*DHCP server IP and SoftAP mode IP address.*/
    gstrM2MAPConfig.au8DHCPServerIP[0] = 0xC0; /* 192 */
    gstrM2MAPConfig.au8DHCPServerIP[1] = 0xA8; /* 168 */
    gstrM2MAPConfig.au8DHCPServerIP[2] = 0x01; /* 1 */
    gstrM2MAPConfig.au8DHCPServerIP[3] = 0x01; /* 1 */
       set_dev_name_to_mac((uint8_t *)gstrM2MAPConfig.au8SSID, gau8MacAddr);
    m2m_wifi_start_provision_mode((tstrM2MAPConfig *)&gstrM2MAPConfig, (char
*)gacHttpProvDomainName, 1);
    printf("Provision Mode started.\r\nConnect to [%s] via AP[%s] and fill up the page.\r\n",
MAIN_HTTP_PROV_SERVER_DOMAIN_NAME, gstrM2MAPConfig.au8SSID);
```

HTTP provisioning page is stored in the internal memory of ATWINC1500. This is a custom provisioning web page provided by Atmel, where the customers can modify the web page with certain limitations. A customer can change the attributes such as custom image, color, fonts. But the size of web page cannot be modified and extra tabs cannot be added..

Note: Third argument of the `m2m_wifi_start_provision_mode()` is used to enable or disable the redirect feature of the HTTP provision web page.

4.4. Wi-Fi Host Driver Event and Callback Handling

The Wi-Fi host driver events are handled by running the `m2m_wifi_handle_events()` in the infinite loop. HIF (Host communication Interface) layer API's are used to monitor the ATWINC1500 external interrupt. These event callback functions are registed using the 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, the host interface ISR layer will read the ATWINC1500 control register. It identifies the type of event which triggered the external interrupt. If there is any data available in the ATWINC1500 buffer, the corresponding event and registered callback function will be called. The host MCU Wi-Fi application driver handles different categories events 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_ip_cb()` handles all the socket, and network application event callbacks
- `m2m_ota_cb()` handles all the *Over The Air* firmware upgrade events

4.5. Handling Provisioning Information

When the ATWINC1500 HTTP server receives the credentials from the provisioning web page, it triggers the `M2M_WIFI_RESP_PROVISION_INFO` event. The credential parameters such as SSID and passphrase are parsed in `wifi_cb()`. These parameters are passed in to the `m2m_wifi_connect()` function using the structure `pstrProvInfo`.

```
static void wifi_cb(uint8_t u8MsgType, void *pvMsg)
{
    switch (u8MsgType) {
    case M2M_WIFI_RESP_PROVISION_INFO:
    {
        tstrM2MProvisionInfo *pstrProvInfo = (tstrM2MProvisionInfo *)pvMsg;
        printf("wifi_cb: M2M_WIFI_RESP_PROVISION_INFO.\r\n");
        if (pstrProvInfo->u8Status == M2M_SUCCESS) {
```

```
            m2m_wifi_connect((char *)pstrProvInfo->au8SSID, strlen((char *)pstrProvInfo-
>au8SSID), pstrProvInfo->u8SecType,
                    pstrProvInfo->au8Password, M2M_WIFI_CH_ALL);
```

Note: In HTTP provisioning application, the security type of the AP must be configured using WPA2-PSK/CCMP only. Other security methods are not supported.

4.6. HTTP Provision Mode Limitations

The current implementation of the HTTP Provisioning has the following limitations:

- AP mode supports OPEN and WEP security types only
- The AP can only support a single associated station. Further connect attempts will be rejected.
- Concurrency (simultaneous STA/P2P and AP mode) is not supported. Before activating the AP mode, the host MCU application should disable the mode currently running.
- Provisioning uses AP mode with open security. No Wi-Fi security nor application level security (e.g. TLS) is used and therefore the AP credentials entered by the user are sent on the clear and can be seen by eavesdroppers.
- The ATWINC Provisioning home page is a static HTML page. No server-side scripting is allowed in ATWINC HTTP server.
- Only APs with WPA-personal security (passphrase based) and no security (Open network) can be provisioned
- WEP and WPA-Enterprise AP's cannot be provisioned
- The Provisioning is responsible for delivering the connection parameters to the application and the connection procedure.
- The validity of the connection parameters is the responsibility of the application

5. How to Run the HTTP Provision Mode Application

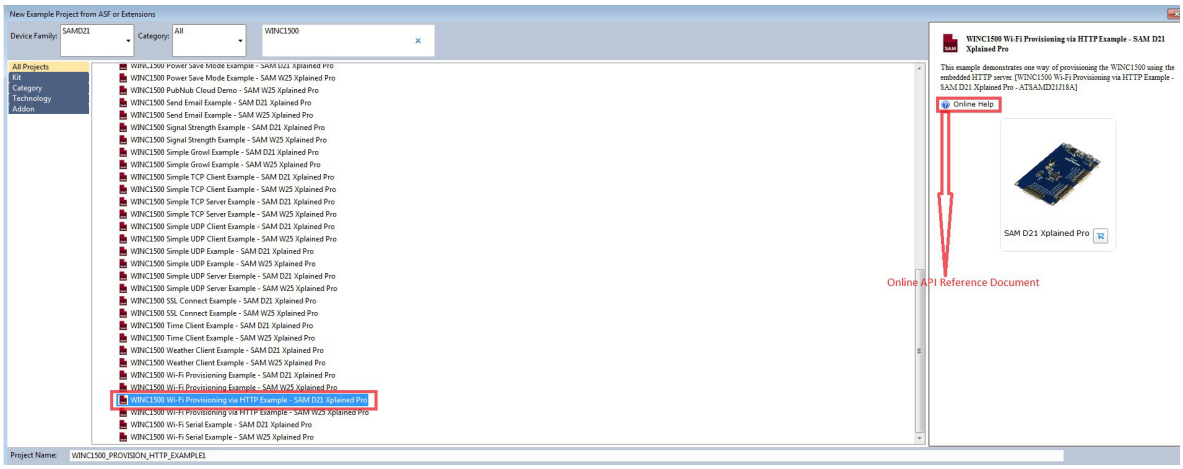
This topic elaborates the steps to run the HTTP provisioning mode application using SAM D21 XPlained Pro board with ATWINC1500 WLAN module.

5.1. Getting Started ASF ATWINC1500 HTTP Provision Mode Demo

To get started with ATWINC1500 projects using Atmel Studio ASF example applications,

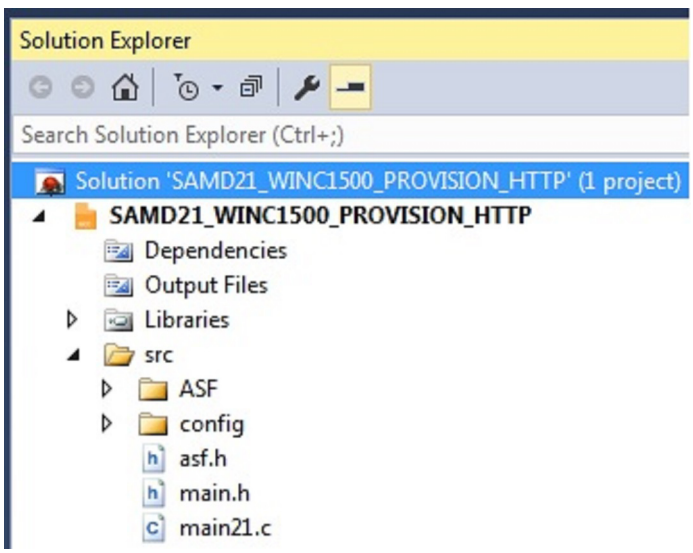
1. Open Atmel Studio 7. Go to **File** > **New** > **Example Projects**.
2. Search for ATWINC1500 sample application for other MCU.
3. Select the Wi-Fi HTTP Provisioning Example `WINC1500_PROVISION_HTTP_EXAMPLE` project for SAM D21 and open it.

Figure 5-1. Atmel Studio ATWINC1500 Project Creation



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

Figure 5-2. HTTP Provision Mode Directory Structure

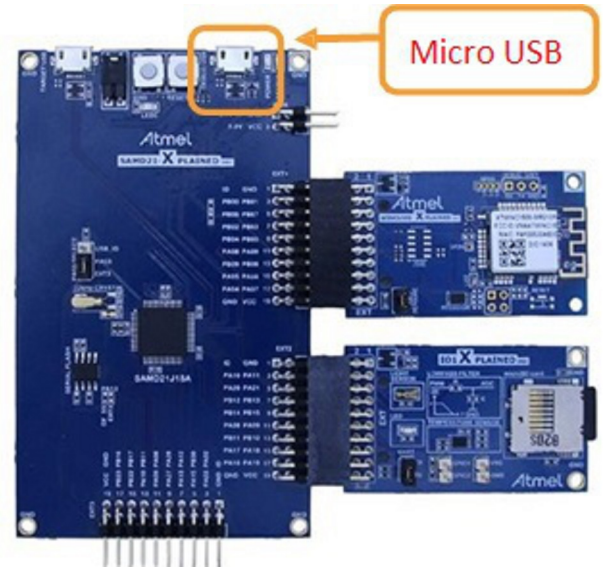


5.2. Programming SAM D21 Xplained Pro

To download the firmware from the PC to the 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 5-3. SAM D21 Xplained Pro Board with ATWINC1500



2. Connect the USB cable to the 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 the latest firmware into the ATWINC1500 module, follow the steps specified in the Quick Start Guide.

5.3. Executing HTTP Provision Mode Application

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

The example uses the following hardware:

- The SAM D21 Xplained Pro
- The ATWINC1500 on EXT1 header
- The 802.11 b/g/n supported AP or router
- Android mobile or any WLAN client device

Figure 5-4. Demo Setup

1. In HTTP provision mode demo, ATWINC1500 starts as a softAP using open security mode (no security method).It broadcasts the beacon frames with SSID `WINC1500_00:00`. The following macros are defined in the `main.h` file.

```
#define MAIN_M2M_AP_SEC                 M2M_WIFI_SEC_OPEN
#define MAIN_M2M_AP_WEP_KEY             "1234567890"
```

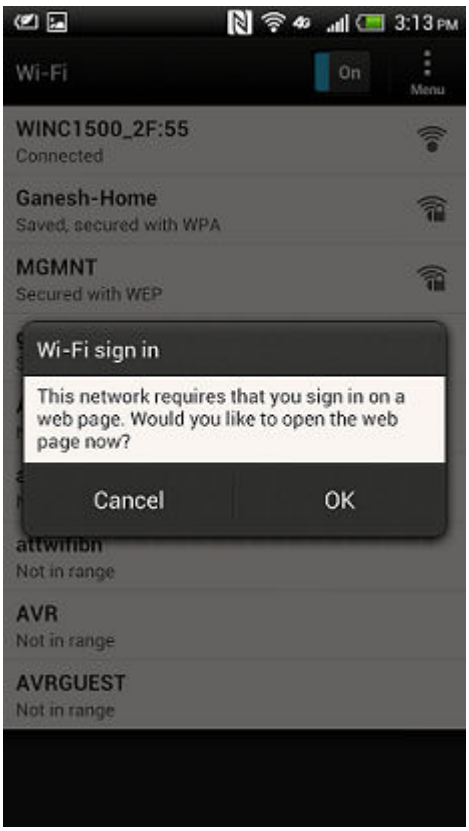
```
#define MAIN_M2M_AP_SSID_MODE            SSID_MODE_VISIBLE
#define MAIN_HTTP_PROV_SERVER_DOMAIN_NAME    "atmelconfig.com"
#define MAIN_M2M_DEVICE_NAME                 "WINC1500_00:00"
#define MAIN_MAC_ADDRESS                     {0xf8, 0xf0, 0x05, 0x45, 0xD4, 0x84}
```

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

Figure 5-5. Atmel Studio Debug Button

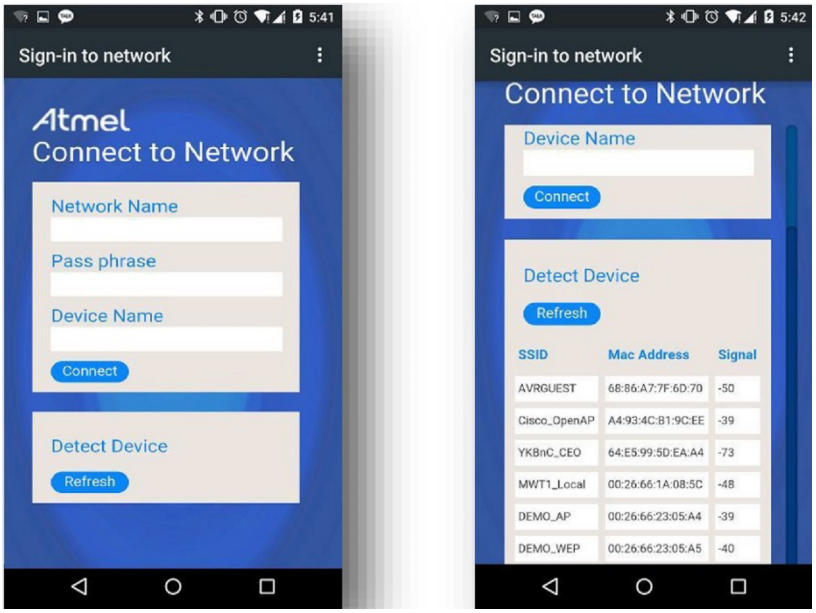
4. Run the application. Success or error messages appear in the serial port terminal.
5. ATWINC1500 softAP is available to connect with a mobile device or any WLAN client supported devices.
6. Connect to ATWINC1500 softAP as shown.
7. After the successful connection with ATWINC1500, the HTTP configuration web page is triggered automatically as shown.

Figure 5-6. HTTP Web Triggering After AP Connection



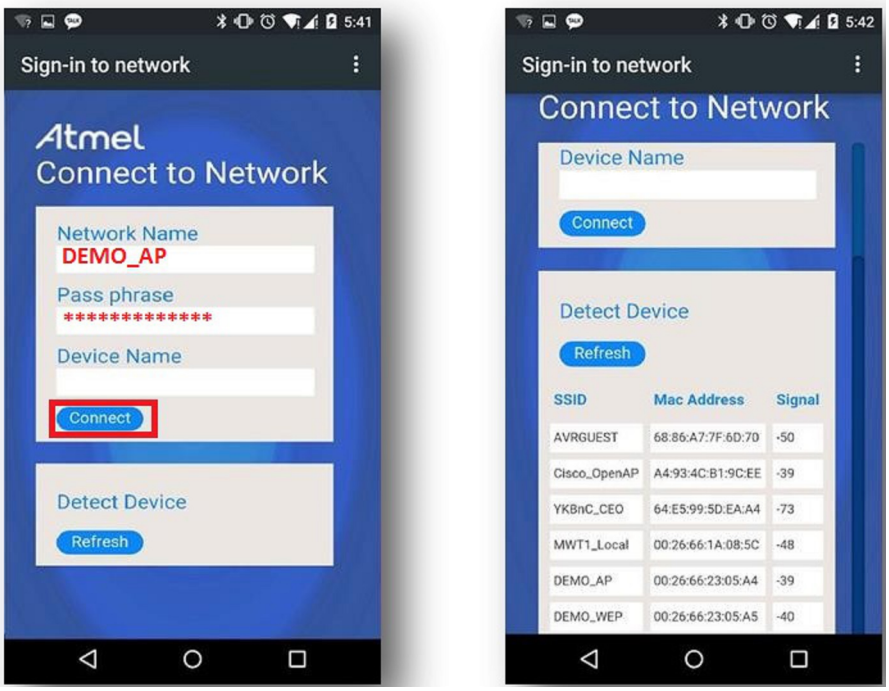
8. Press **OK**. The web page opens in the default browser as shown.

Figure 5-7. ATWINC1500 Provisioning Page



9. Click **Refresh** to scan all the available APs or routers. These scan results are provided through the ATWINC1500 HTTP server to the provisioning page.
10. Enter the AP credentials to connect with the desired AP as shown.

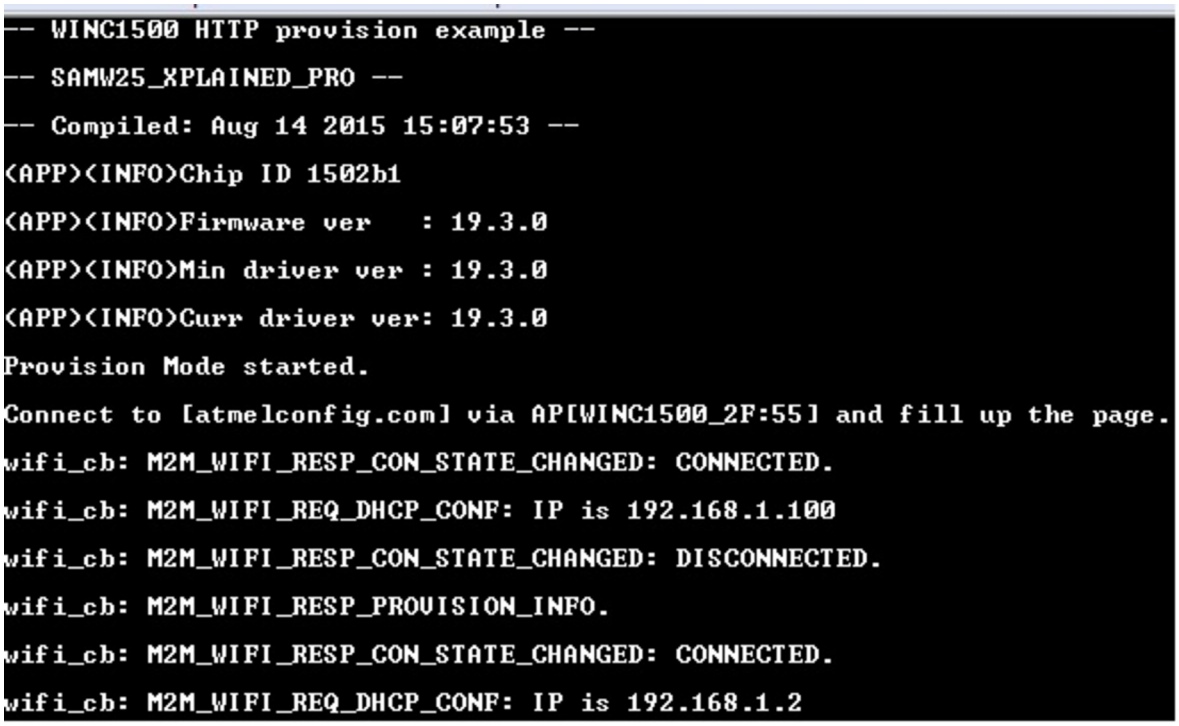
Figure 5-8.



11. Click **Connect** to transfer credentials to the ATWINC1500 HTTP server.
12. The ATWINC1500 HTTP server triggers the provisioning response event with provisioning information.
13. The host MCU handle the provisioning response event in the Wi-Fi callback function.
14. The AP credentials such as SSID, Security type, and passphrase are parsed.

15. Using AP's configuration, the ATWINC1500 application driver connects with the desired AP and obtains the IP address using the DHCP client request.
16. The HTTP provision mode terminal log displays the successful connection.

Figure 5-9. Provision Mode Log



6. Revision History

Doc Rev.	Date	Comments
42642A	03/2016	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**