Application Note

Firmware Update Procedure for WINC Module

Prerequisites

- **Hardware Prerequisites**
  - Supported Xplained Pro Evaluation Kit
  - Atmel WINC1500 **or** WINC3400 extension
  - USB Micro Cable (TypeA / MicroB)

- **Software Prerequisites**
  - Atmel Studio 7
  - Firmware update project

Introduction

This application note provides useful information to perform firmware update for any WINC IoT module.

The following topics will be covered:

- Firmware update procedure
- Root certificate update procedure

Table of Contents

Icon Key Identifiers

Icons are used to identify different assignment sections and reduce complexity. These icons are:

INFO	Delivers contextual information about a specific topic
TIPS	Highlights useful tips and techniques
TO DO	Highlights objectives to be completed
RESULT	Highlights the expected result of an assignment step
WARNING	Indicates important information
EXECUTE	Highlights actions to be executed out of the target when necessary

1. Firmware Update Project

The WINC1500 or WINC3400 Wi-Fi module firmware update project can be retrieved through the Atmel Software Framework.

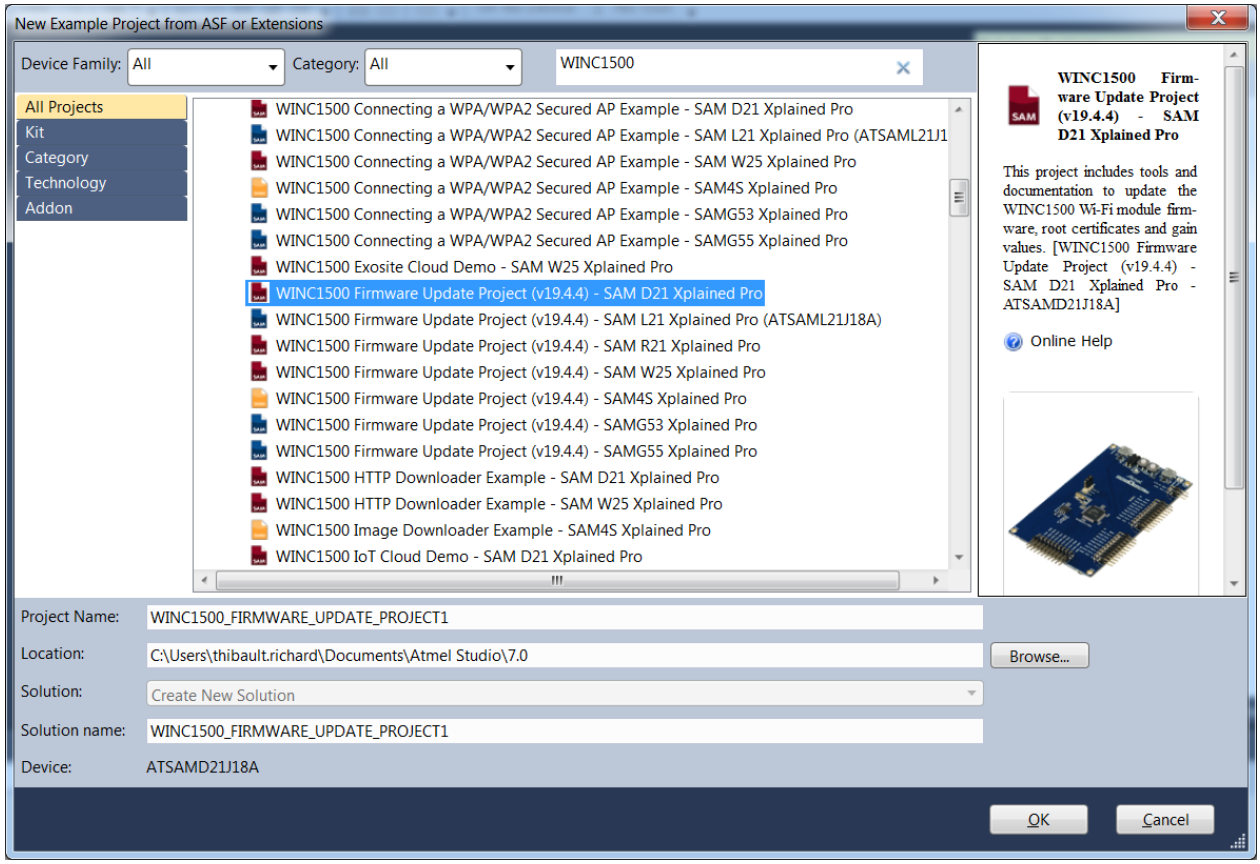
- The latest version of the Atmel Software Framework can be found on the Atmel Gallery web site (http://gallery.atmel.com/) or using the Atmel Studio Extension manager

1.1 Atmel Software Framework

Once the Atmel Software Framework is up-to-date, you can open and search for the Firmware Update Project from the *"New Example Project from ASF..."* menu in Atmel Studio.

INFO

WINC1500 Wi-Fi module related projects are all prefixed with "WINC1500" making it easier to filter the project in the dialog. Alternatively use "WINC3400" filter to find corresponding project.
The firmware update version is printed both in the project name and in the project description.

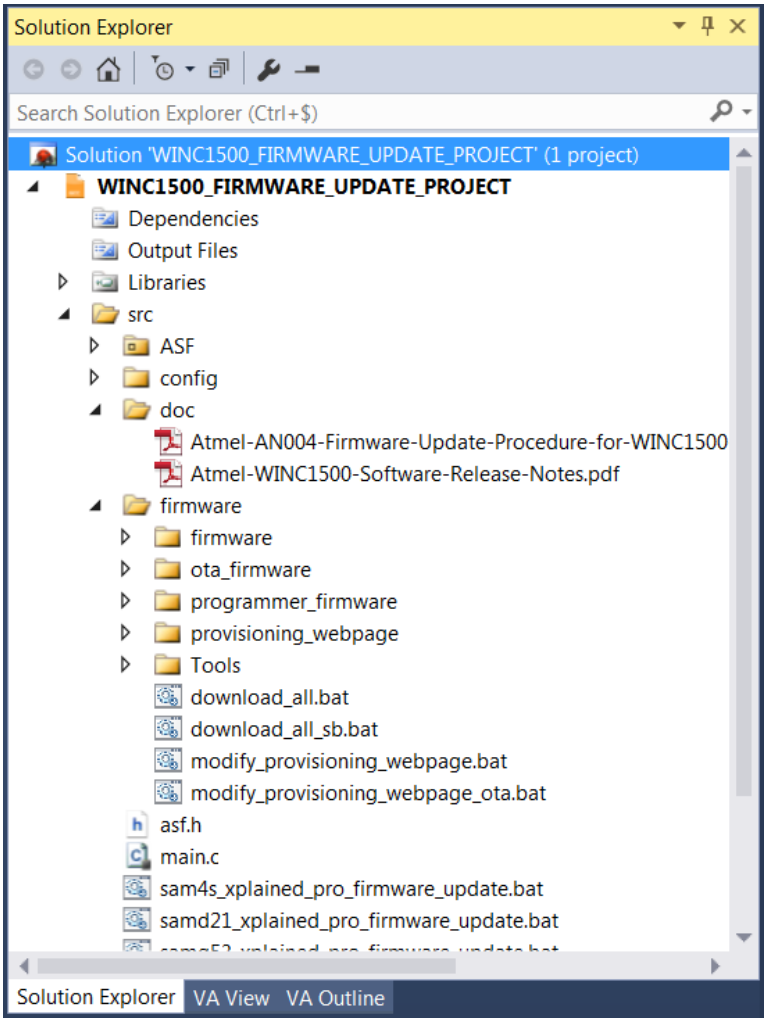


TO DO

Select the appropriate "*WINC Firmware Update Project (vxx.x.x)*" project corresponding to your Xplained Pro board and then press OK button to import firmware update project and related documentation.

1.2 Project Overview

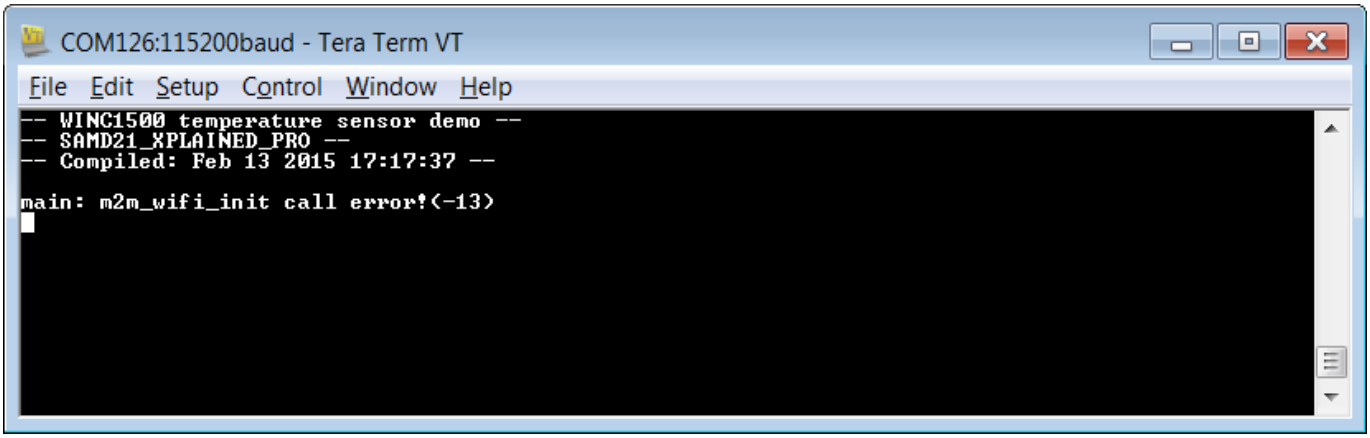
The firmware update project appears as a regular Atmel Studio ASF project:



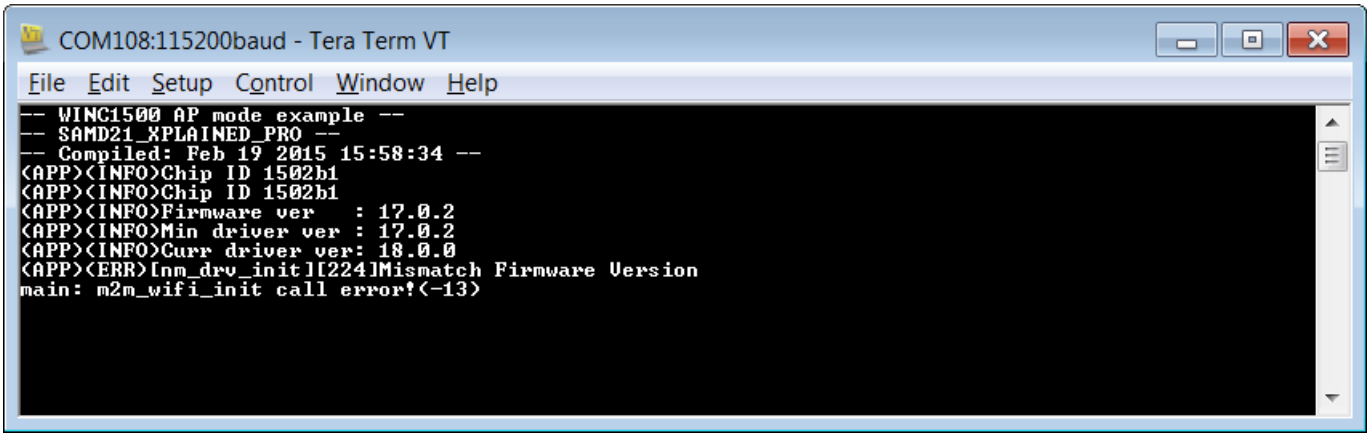
- **/src/doc folder** – Contains the following documentation:
  - Firmware update guide.
  - WINC1500 firmware and host driver release note.
- **/src/firmware folder** – Contains the new WINC firmware as well as:
  - The “download_all.bat” script to perform WINC firmware update using WINC built-in UART.
  - The “modify_provisioning_webpage.bat” to update the WINC firmware image with a customer HTTP provisioning webpage.
- **/src folder** – Contains update scripts to perform WINC firmware update using a serial bridge.

INFO The following update procedure takes care of updating the WINC firmware as well as TLS/SSL root certificates and radio gain tables.

When Atmel provides an update for the WINC module, it usually combines the Wi-Fi software API for the host MCU and a binary firmware for the WINC module. To ensure the compatibility between the Wi-Fi software API on the host MCU and the WINC module, a major/minor/patch version number is used and verified at the Wi-Fi driver init. When a version mismatch is detected at startup, the Wi-Fi driver initialization fails and the `m2m_wifi_init()` function returns an error code:



Setting the `CONF_WIFI_M2M_DEBUG` to 1 in the `conf_winc.h` configuration file of your Atmel Studio project will provide more information about the version mismatch error:



In this scenario a firmware update with the appropriate firmware version number is expected. Each ASF release is tied to one particular WINC driver/firmware release.

The firmware update can be done by connecting a Windows computer to:

- Host MCU UART (typically the EDBG COM port when using an Xplained Pro board). The host MCU is connected to the WINC via SPI, thus acting as a serial bridge.
- The WINC built-in UART.

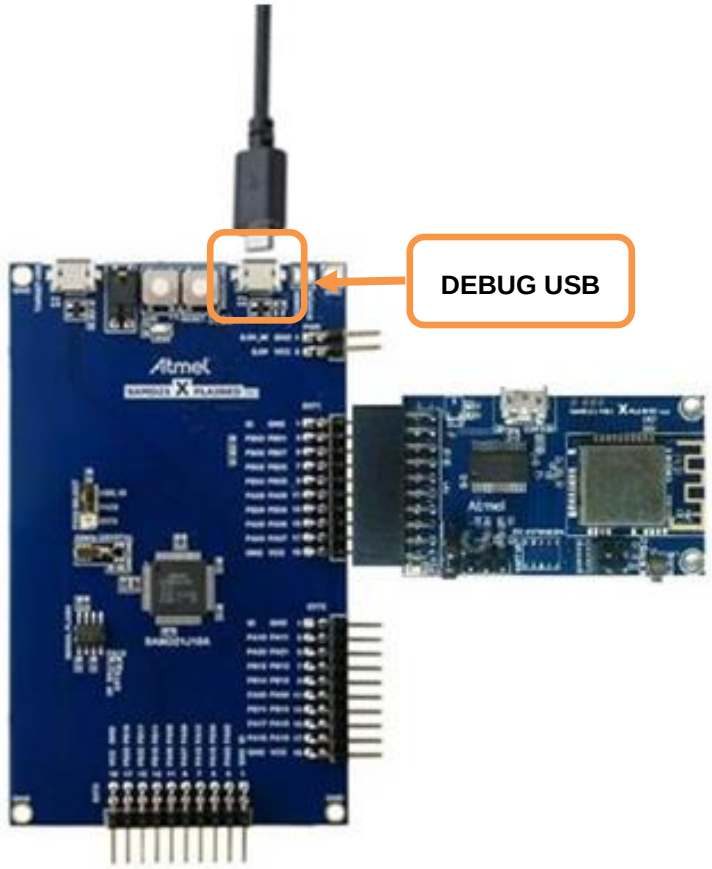
2. Firmware Update via Serial Bridge

Prior to running any update script, you must ensure that the hardware is setup as required. Below description is using a WINC1500 module, but procedure remains exactly the same for a WINC3400 module.

2.1 Hardware Setup

INFO The update procedure requires that WINC module is attached on EXT1 of the Xplained Pro kit.

Connect the WINC IoT module on EXT1 of the SAM Xplained Pro board. Then plug a micro USB cable from your laptop to the debug USB port of the Xplained Pro kit as displayed below:

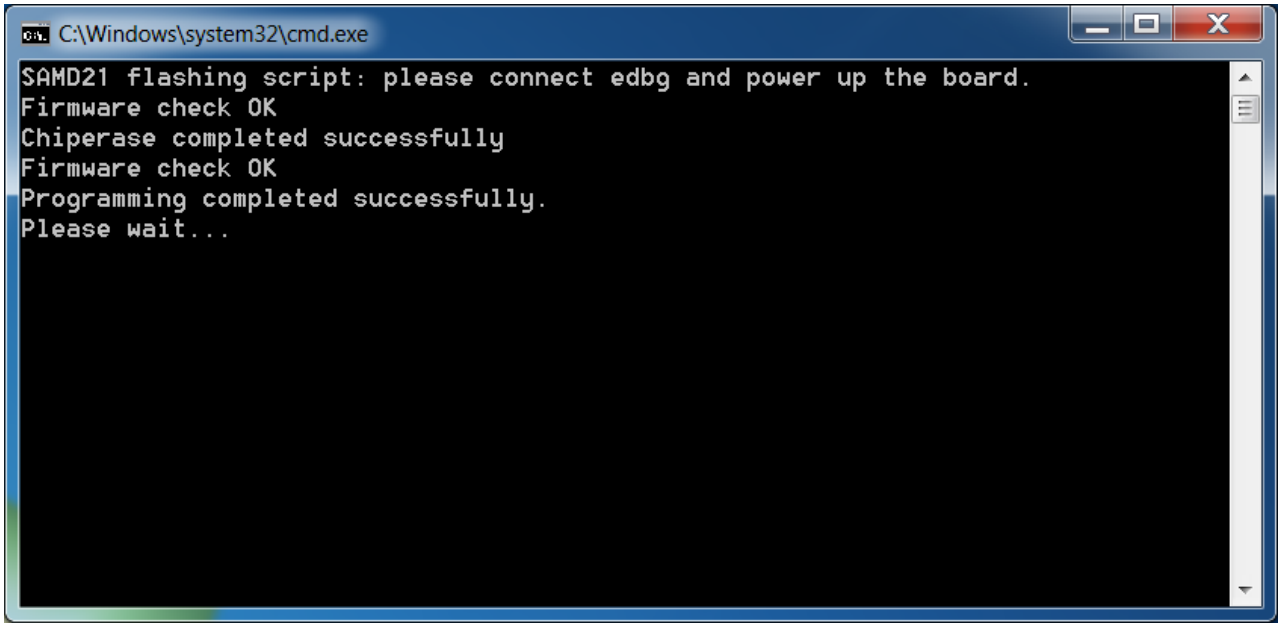


2.2 Batch Script

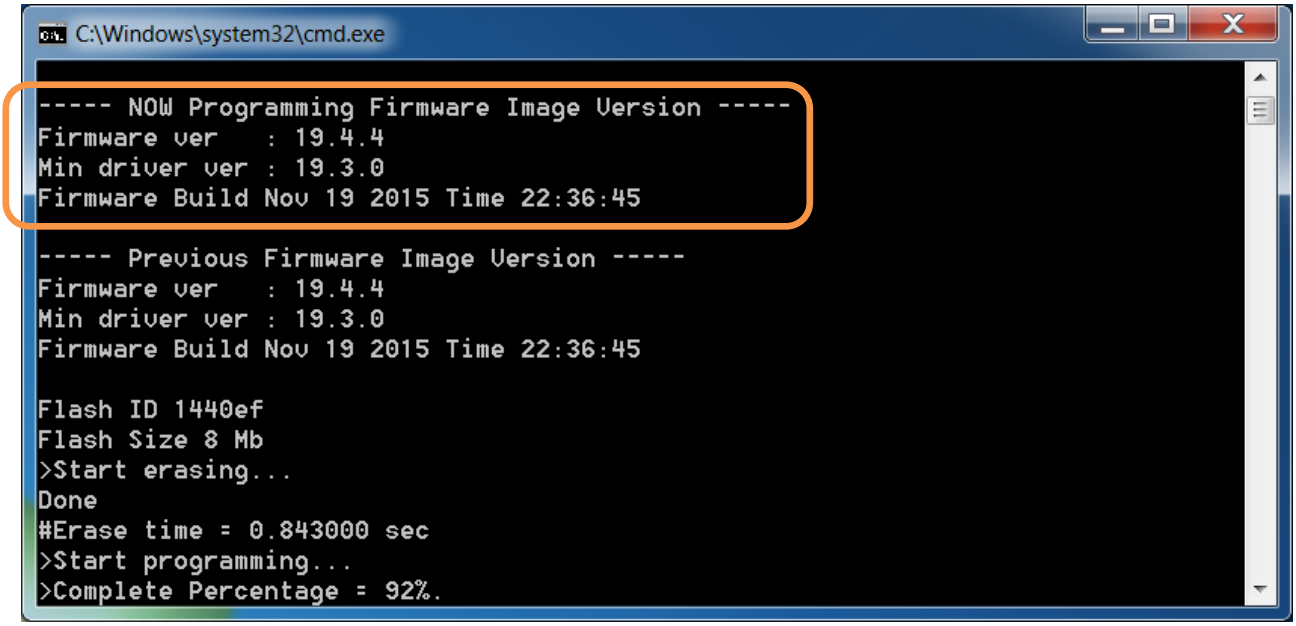
In the src folder of the "*WINCXXXX_FIRMWARE_UPDATE_PROJECT"* you will find a list of batch (.bat) script files used to trigger a firmware update.

- Ensure that the SAM Xplained Pro board is connected to your laptop via the debug USB port. The virtual EDBG COM port of the board is now listed in the device manager.
- Run the *sam_xplained_pro_firmware_update.bat* script that correspond to your board.

As a first stage, the batch script will program a serial bridge binary on the host MCU to redirect data from the computer (EDBG virtual COM port) to the WINC chip (via SPI).
The serial bridge also performs the WINC power up sequence, thus ensuring that the WINC bootloader is in the appropriate state to start a firmware update.



During the update process, the batch script will output the firmware version being programmed onto the WINC as well as the previously installed firmware version:



After several seconds the following message should appear, meaning that the WINC1500 Wi-Fi module update procedure is complete:

RESULT The WINC chip firmware, TLS/SSL root certificates and gain table have been successfully updated.

3. Firmware Update via Built-in UART

Prior to running any update script, you must ensure that the hardware is setup as required. Below description is using a WINC1500 module, but procedure remains exactly the same for a WINC3400 module.

3.1 Hardware Setup

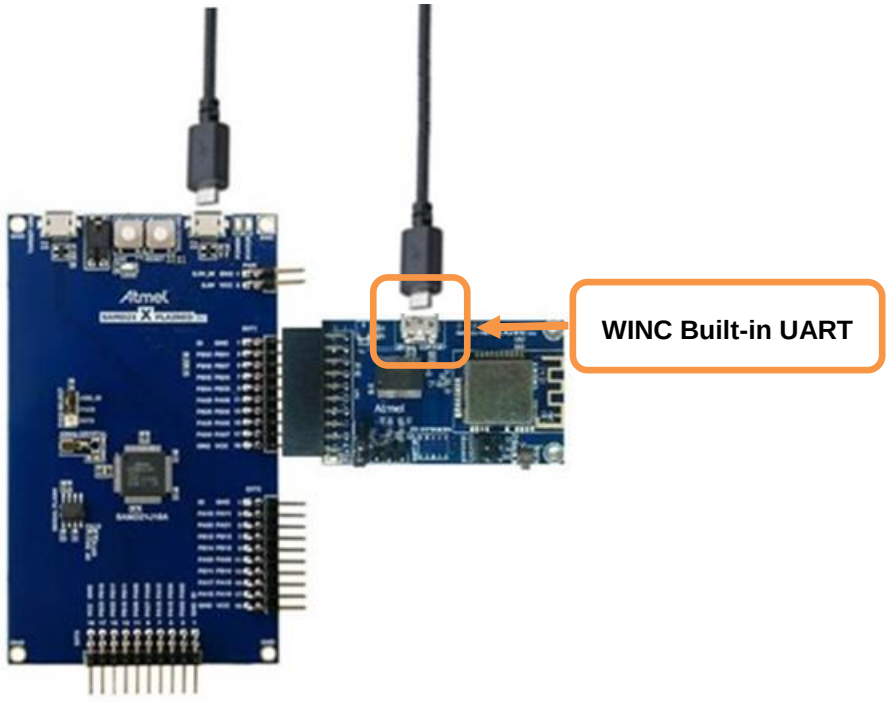
To perform a firmware update using the WINC built-in UART, it is mandatory that the WINC chip is in the right bootloader state. To do so, a host MCU must power up the WINC chip and then perform the reset sequence as defined in the datasheet. This can be done very easily from the host MCU by calling the `m2m_bsp_init()` function.

```
int main(void)
{
	/* Initialize the board. */
	sysclk_init();
	board_init();

	/* Initialize the WINC BSP and perform reset sequence. */
	nm_bsp_init();

	while(1) {
		/* WINC firmware is ready for the built-in UART update. */
	}
}
```

The WINC chip is now ready to perform a firmware update via the WINC built-in UART. Simply connect your PC to the WINC built-in UART.



INFO

Depending on the WINC module version, it may feature a micro USB plug which is connected to the WINC built-in UART via an FTDI module. In this case, the PC must have the latest FTDI driver installed to see the corresponding virtual serial COM port.

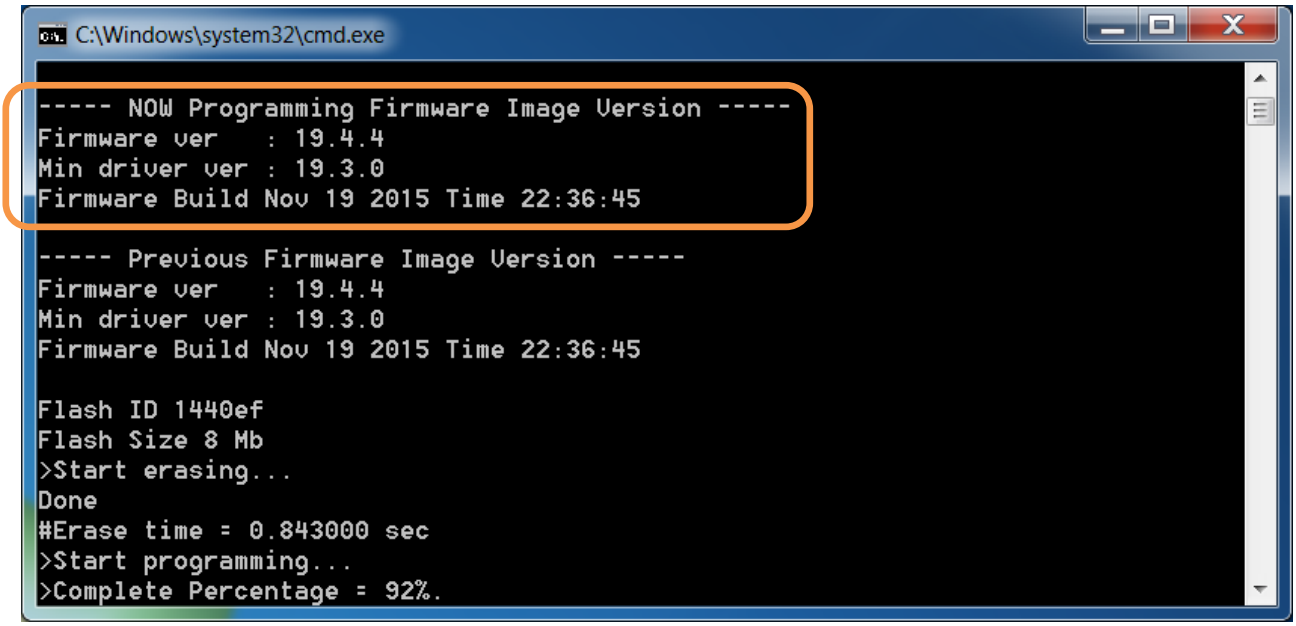
Alternatively, if no micro USB plug is present; RX, TX and GND UART signals are available to connect to a third-party serial to USB converter which in turns connects to the PC.

3.2 Batch Script

In the src/firmware folder of the "*WINCXXXX_FIRMWARE_UPDATE_PROJECT"* is located the download_all.bat script which is used to trigger the built-in firmware update.

- Ensure that the host MCU is powered up and that the WINC built-in UART is connected to your laptop via a serial to USB converter.
- In a Windows shell, run the command "*download_all.bat UART"* to start the firmware update.

During the update process, the batch script will output the firmware version being programmed onto the WINC as well as the previously installed firmware version:

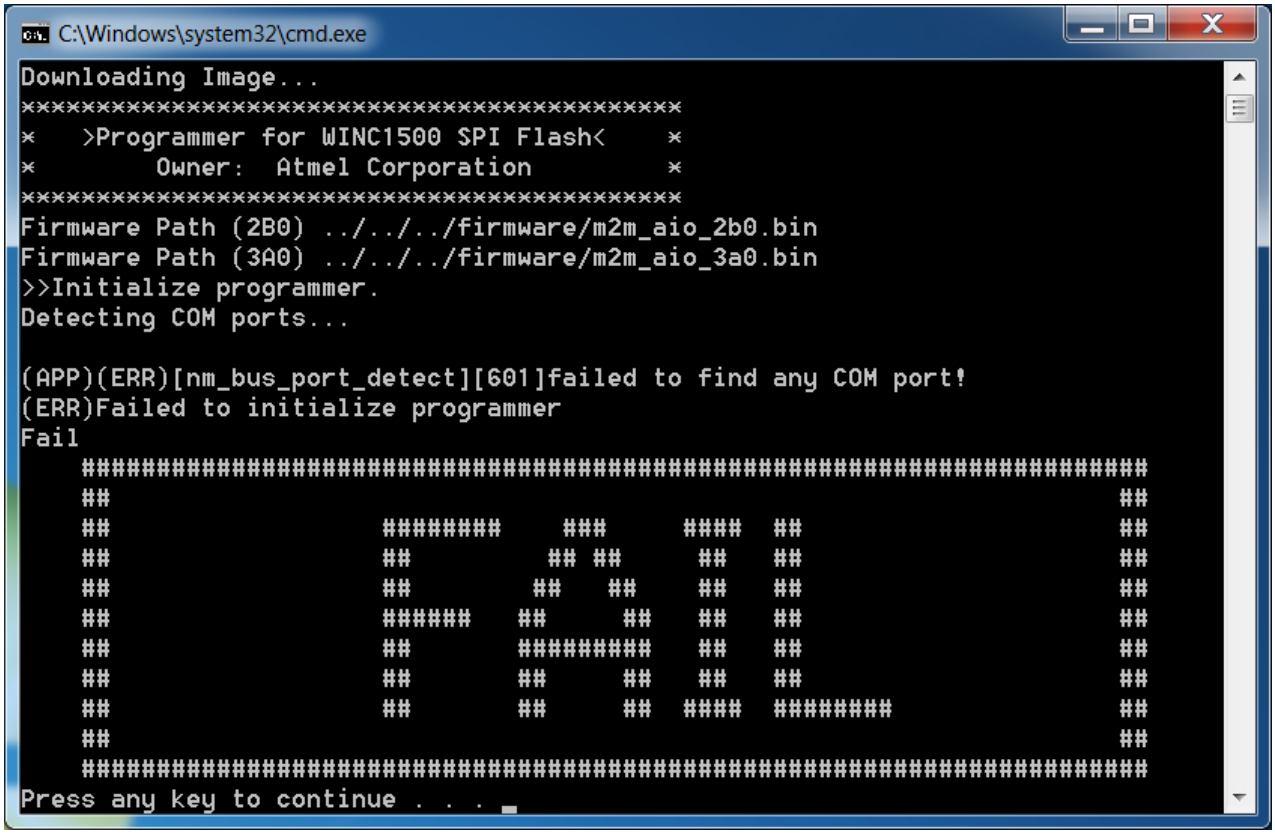


After several seconds the following message should appear, meaning that the WINC1500 Wi-Fi module update procedure is complete:

RESULT The WINC chip firmware, TLS/SSL root certificates and gain table have been successfully updated.

4. Troubleshoot Firmware Update

1. Failed to find any COM port.



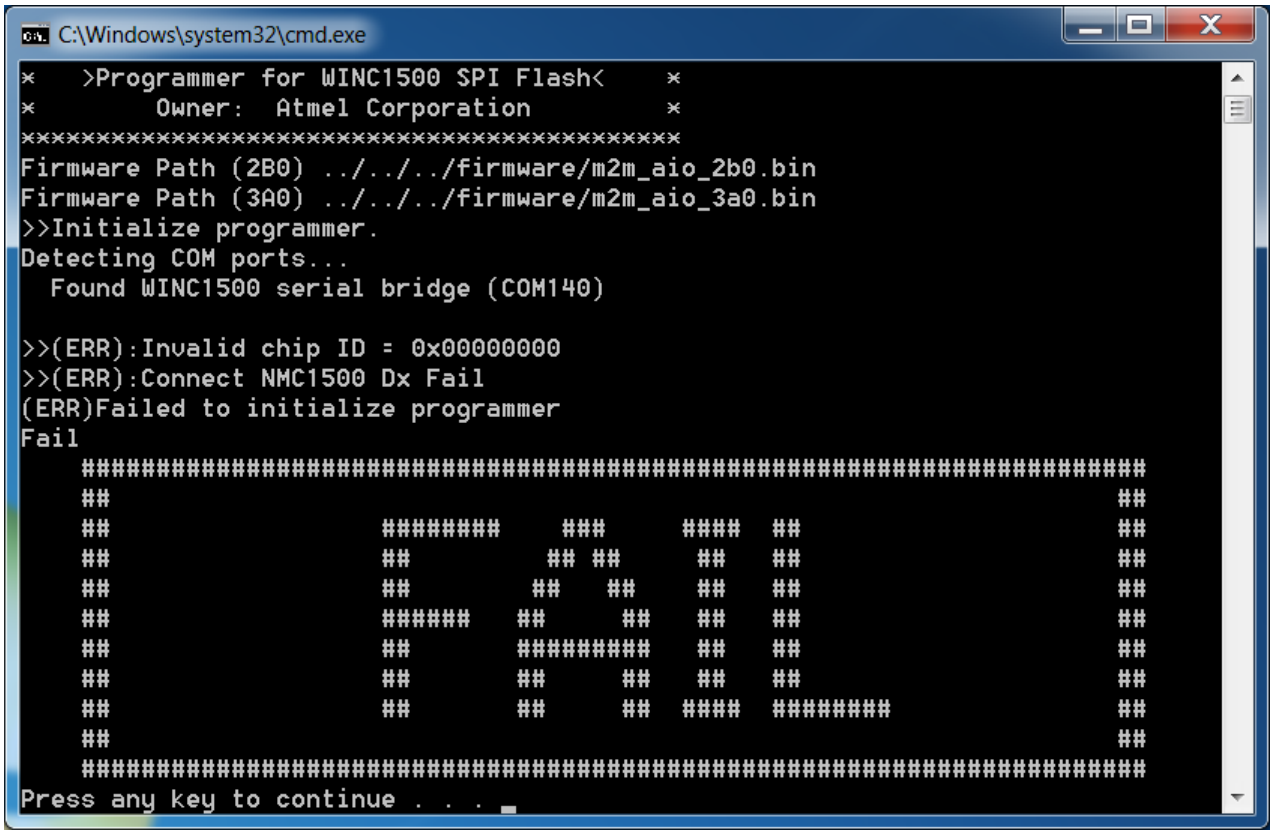
The image downloader tool used to perform a serial bridge or a buit-in UART firmware update will try to look for available COM ports using Windows API. It will then try to match each COM port name with “EDBG” string or a port number “(COM” string. If one of the two conditions is true, the program will then try to send a 0x12 char on the UART line. The other side is then expected to answer 0x5A for a built-in UART update or 0x5B for a serial bridge update.

How to fix it

- Make sure WINC COM port is listed in the device manager.
- Make sure WINC COM port is not opened by any other application. For verification, try to open and close the COM port with a terminal application.
- Cheap USB cable (serial bridge) or cheap serial to USB converter (built-in UART) can introduce garbage on the UART line thus failing the detection of the WINC COM port. Try a different cable.
- When performing a built-in UART update it is expected that the WINC bootloader is in a particular state that can only be achieved after doing a clean power up and reset sequence. Hence, before doing a firmware update always ensure that a clean power up and reset sequence has been made.
- Make sure that no other extension board (ex: IO1...) is connected to the Xplained Pro board while performing the firmware update.
- Make sure the project path is not exceeding Windows maximum 260 characters path length

INFO It is possible to force the firmware update tool to use a specified COM port number. For example to use COM56, run the script like this: "*download_all.bat UART 56"*

2. **Failed to initialize programmer: invalid chip ID.**

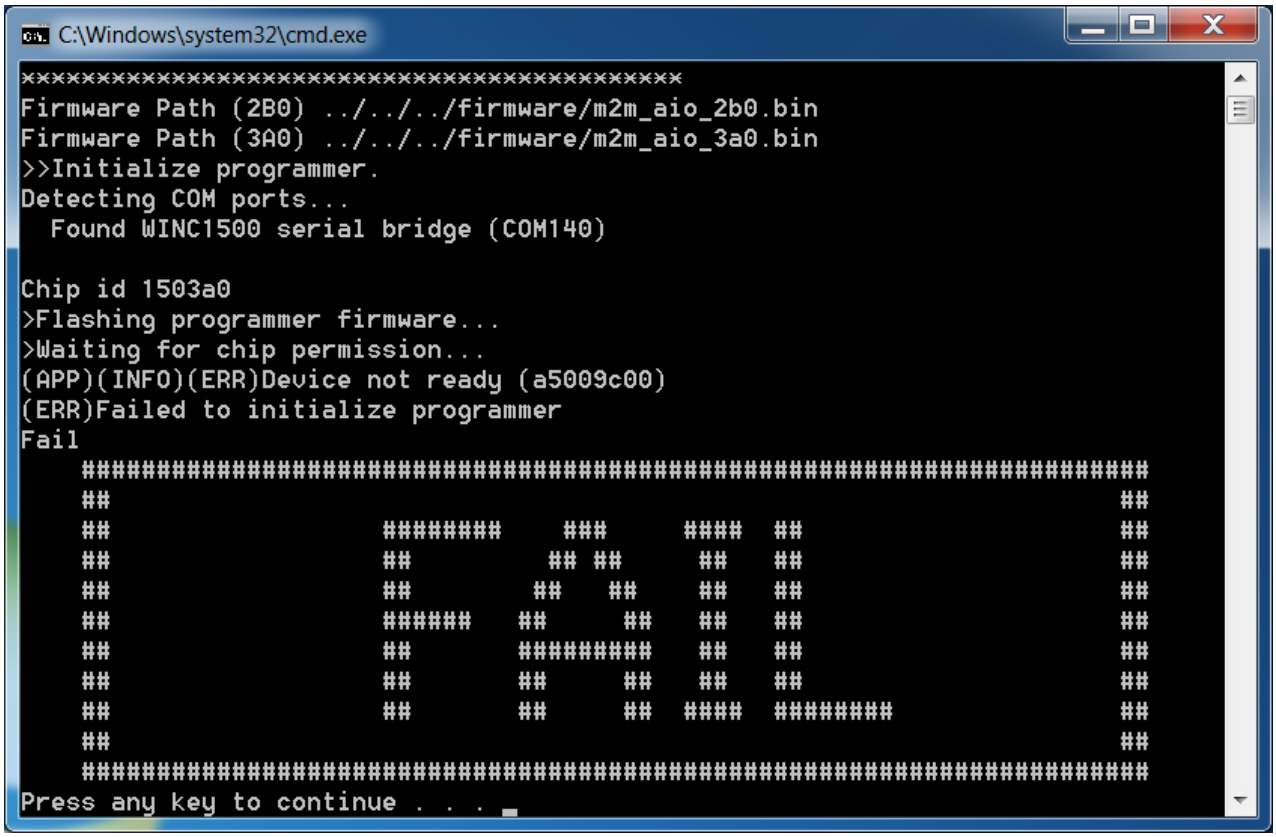


The above error typically happens when there is garbage or noise on the UART line preventing from reading the correct chip ID value.

How to fix it

- Try connecting the PC and the WINC module with a different cable. A clean power up and reset sequence of the WINC is necessary to start over with the WINC bootloader in the appropriate state.

3. **Failed to initialize programmer: waiting for chip permission.**



After printing the correct chip ID of the WINC, the firmware update tool programs a small binary (programmer firmware) to assist with WINC flash programming. At this stage the image downloader will change the UART baudrate from 115200 to 500000 to speed up the actual transfer of the firmware image. Once the baudrate change is made, the chip permission is verified to ensure the UART connection is reliable. Failing at this stage means that the current setup does not support such a high baudrate.

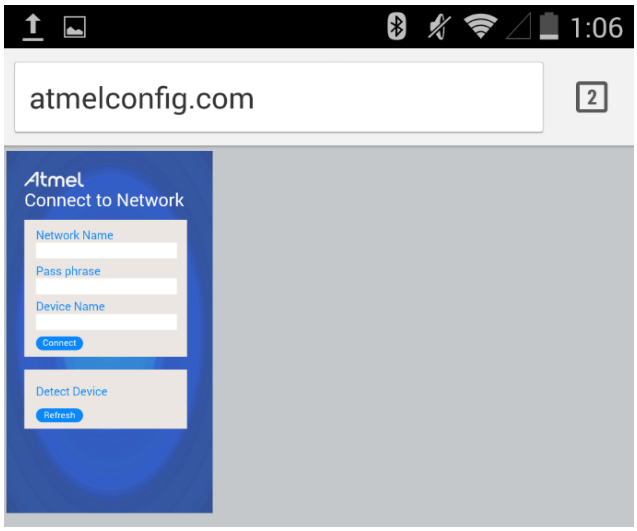
How to fix it

- It is recommended to try connecting the PC and the WINC module with a different cable. Also a clean power up and reset sequence of the WINC is necessary to start over with the WINC bootloader in the appropriate state.

5. Modify Provisioning Webpage

The WINC device features a provisioning webpage mode that can be used to help a user with entering credentials to connect the WINC device to the desired Access Point. The HTTP server and the actual HTML provisioning webpage is embedded in the WINC firmware, hence it cannot be modified from the host MCU. However, the firmware update project can be used to modify the HTML code and generate a new firmware image with the updated webpage.

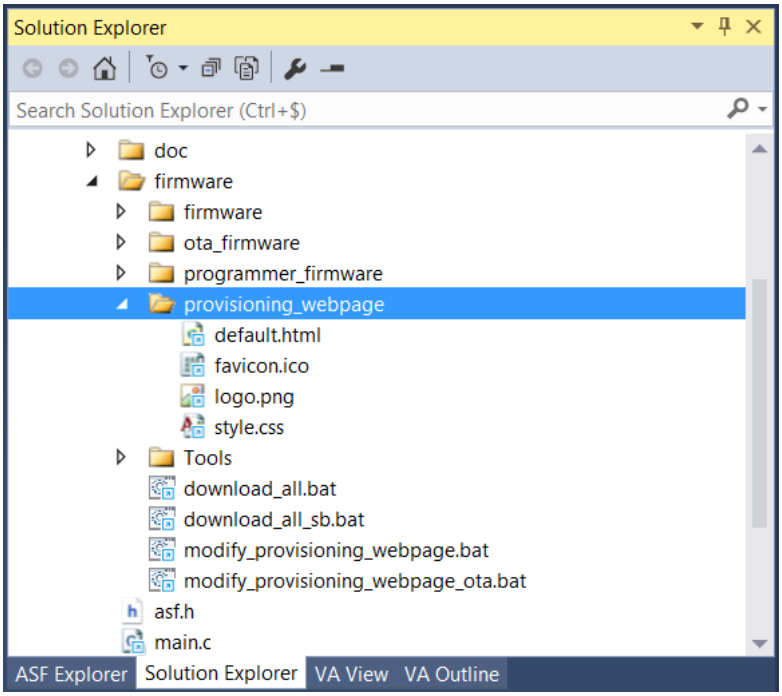
The host MCU can enter the web provisioning mode by calling the `m2m_wifi_start_provision_mode()` API.



After enabling the web provisioning mode, the follow actions are expected:

1. User finds and connects WINC AP SSID with a smart phone (or tablet).
2. User opens a web browser and writes the WINC home page in the address bar.
3. If the HTTP redirect is enabled when calling the `m2m_wifi_start_provision_mode()` function, any web address the WINC home page will load automatically. Else user must enter the same URL as supplied in previous function call.
4. WINC home page (shown in picture above) will appear in the browser.
5. To scan for the list of Wi-Fi APs in the area, user can press “Refresh”.
6. Desired AP is then selected from the search list (by one click) and its name will appear automatically in the “Network Name” text box. Alternatively, user can simply edit the corresponding text box.
7. User must then enter the correct AP passphrase (for WPA/WPA2 personal security) in the “Pass Phrase” text box. If the AP is not secured (open network) the field should be left empty.
8. WINC device name may be optionally configured if desired by the user in the “Device Name” text box.
9. User presses “Connect” button so the WINC can turn off AP mode and start connecting to the provisioned AP.

The firmware update projects includes the HTML code used by the WINC and also the necessary scripts to generate a new WINC firmware image.



- **/src/firmware** – Contains the script to generate an updated WINC firmware image:
  - The “modify_provisioning_webpage.bat” script to generate a new WINC firmware image that includes the HTML code provided in the “provisioning_webpage” folder. This new firmware image will overwrite the default firmware image located in the “firmware” folder. Performing a firmware update procedure as described in the first section of this document will use the newly generated firmware image.
  - The “modify_provisioning_webpage_ota.bat” script to generate a new WINC firmware OTA image that includes the HTML code provided in the “provisioning_webpage” folder. This image is only usable to perform an OTA update. Output image will be stored in the “ota_firmware” folder.

- **/src/firmware/provisioning_webpage folder** – Contains the WINC HTML code:
  - “logo.png” - Logo image displayed at the top of the provisioning page.
  - “favicon.ico” - Icon that appears inside the browser tab displaying WINC provisioning page.
  - “default.html” - The default HTML file that appears in the browser when the user requests the WINC1500 Home Page.
  - “style.css” - Cascading Style Sheet (CSS) used for describing the look and formatting of the default.html contents.

5.1 Limitations

The provisioning webpage of the WINC can be modified but there are some points that must be put into consideration when applying modifications to the existing provisioning webpage design. Following will be the list of these limitations:

- The total size of the modified files listed in the “provisioning_webpage” folder **must not** exceed 7.9 KB.
- The user is expected to adjust his modifications within the already existing files and **must not** (add, remove or rename) any of these files.
- The WINC HTTP server expects to deal with the following (shall not be modified also):
  - HTML input element used to store AP name to connect to must have name “login”.

```
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Network Name<p>
<input type="text" name="login" value="" maxlength="32">
<p>Pass phrase</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
```

  - HTML input element used to store AP passphrase to connect to must have name “password”.

```
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Network Name<p>
<input type="text" name="login" value="" maxlength="32">
<p>Pass phrase</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
```

  - HTML input element used to store device name must have name “device-name”. (could be left blank by the user)

```
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Network Name<p>
<input type="text" name="login" value="" maxlength="32">
<p>Pass phrase</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
```

- HTML input button object with name “connect” to submit the above data to the WINC HTTP server.

```
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Network Name<p>
<input type="text" name="login" value="" maxlength="32">
<p>Pass phrase</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
```

- HTML input button object with name “refresh” to submit the scan request.

```
<div id="detect-device-container">
<form id="detect-device-holder" method="get" action="">
<p>Detect Device</p>
<input type="submit" name="refresh" value="Refresh" class="refresh-button">
<div id="buffer"></div>
</form>
</div>
```

- The HTML division under the name “detect-device-container” **must** exist as the WINC HTTP server will insert the scan results information inside this form.

```
<div id="detect-device-container">
<form id="detect-device-holder" method="get" action="">
<p>Detect Device</p>
<input type="submit" name="refresh" value="Refresh" class="refresh-button">
<div id="buffer"></div>
</form>
</div>
```

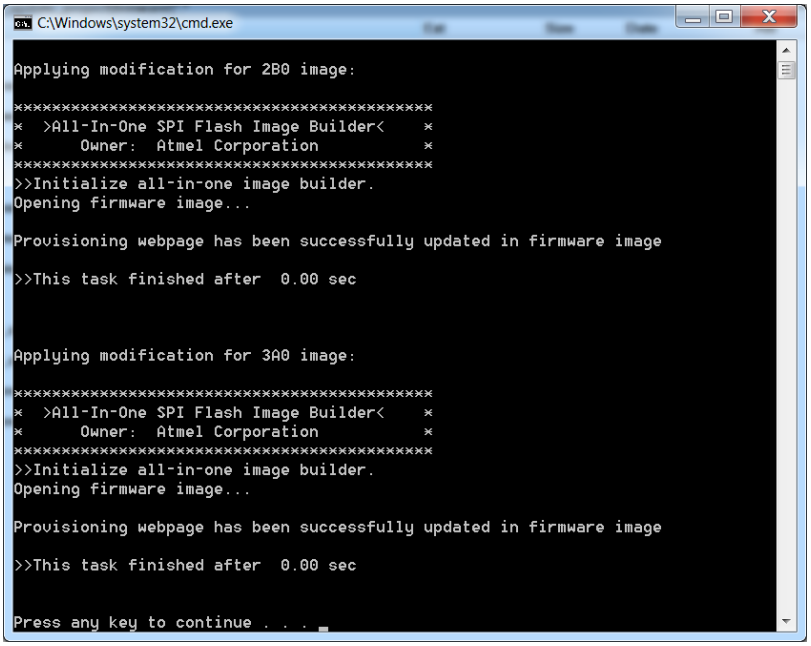
WARNING It’s highly recommended to keep “default.html” file with minimal changes. Failure to comply with the above tag names during modification may result in unexpected behavior.

5.2 Customization

As mentioned before the WINC provisioning webpage interface is customizable as long as the source files are modified with following the limitations described in previous section.

- “logo.png” file can be totally replaced with a custom logo to appear at the top of the page.
- “favicon.ico” file can also be replaced with a custom image to appear in the browser tab.
- “style.css” file can be modified to reflect a custom design.
- “default.html” file can be modified without changing the essential I/O elements described in previous section.

After modifying the HTML page files, generating the firmware image for local firmware update can be done easily calling the “modify_provisioning_webpage.bat” script. The output file will be located in the “firmware” folder.
Alternatively, generating the firmware image for an OTA firmware update can be done easily calling the “modify_provisioning_webpage_ota.bat” script. The output file will be located in the “ota_firmware” folder.



INFO The modify provisioning webpage script for the WINC1500 will generate two files; respectively one image for revision A (2b0) and one image for revision B (3a0).

Finally, push the firmware image modification by doing a firmware update as describe in first section of this document. OTA update can also be used to update the HTML page.

5.3 Example

A simple example will be described to illustrate the above sections as to how the WINC default HTTP provisioning webpage can be modified.

1. Text replacement in “default.html” file.

```
<br></br>
<p>Connect to Network<p>
</div>
<div id="scroll">
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Network Name<p>
<input type="text" name="login" value="" maxlength="32">
<p>Pass phrase</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
<div id="detect-device-container">
<form id="detect-device-holder" method="get" action="">
<p>Detect Device</p>
<input type="submit" name="refresh" value="Refresh" class="refresh-button">
<div id="buffer"></div>
</form>
</div>
```

```
<br></br>
<p>Wireless LAN<p>
</div>
<div id="scroll">
<div id="form-container">
<form id="form-holder" method="post" action="">
<p>Acess Point Name <p>
<input type="text" name="login" value="" maxlength="32">
<p>Password</p>
<input type="password" name="password" value="" maxlength="64">
<p>Device Name (Optional)<p>
<input type="text" name="device-name" value="" maxlength="48">
<input type="submit" name="connect" value="Connect" class="connect-button">
</form>
<div id="buffer"></div>
</div>
<div id="detect-device-container">
<form id="detect-device-holder" method="get" action="">
<p>Send Scan Request</p>
<input type="submit" name="refresh" value="Refresh" class="refresh-button">
<div id="buffer"></div>
</form>
</div>
```

2. Change in the style sheet “style.css” file.

```
#body {
        background: #C4C8CB repeat-x;
}

#log-in-form-container {
        width: 320px;
        height: 568px;
        background-image: radial-gradient(ellipse at center, #348ad5 0%,#348ad5 10%,#
        color: #0A88FB;
        font-family: Calibri, Candara, Segoe, "Segoe UI", Optima, Arial, sans-serif;
        font-size:18px;
}

#log-in-form-container p {
        padding:0;
        margin:0;
}

#logo-container {
        width: 256px;
        height: 58px;
        background: url('logo.png') no-repeat left center;
        color: White;
        font-family: "Segoe UI", Candara, Calibri, Segoe, Optima, Arial, sans-serif;
        font-size:28px;
        line-height: 70%;
        margin-left:28px;
        padding-top:30px;
}
```

```
#body {
        background: #0d8521 repeat-x;
}

#log-in-form-container {
        width: 320px;
        height: 600px;
        background-image: radial-gradient(ellipse farthest-corner at 470px 47px , #F
        color: #0A88FB;
        font-family: Calibri, Candara, Segoe, "Segoe UI", Optima, Arial, sans-serif;
        font-size:20px;
}

#log-in-form-container p {
        padding:0;
        margin:0;
}

#logo-container {
        width: 256px;
        height: 58px;
        background: url('logo.png') no-repeat left center;
        color: Black;
        font-family: "Segoe UI", Candara, Calibri, Segoe, Optima, Arial, sans-serif;
        font-size:28px;
        line-height: 70%;
        margin-left:28px;
        padding-top:60px;
}
```

```

#form-container {
        width: 252px;
        background-color: #EBE6E1;
        margin-left: 28px;
        padding-top:15px;
}
```

```

#form-container {
        width: 252px;
        background-color: #e7dd7b;
        margin-left: 28px;
        padding-top:15px;
}
```

```
.connect-button {
    background-position: 0px 0px;
    width: 73px;
    height: 22px;
    background-color: #0A88FB ;
    border: 2px solid transparent;
    border-radius: 50px;
    color: White;
    font-family: Candara, "Segoe UI", Calibri, Segoe, Optima, Arial, sans-serif;
    font-size: 14px;
    text-align: center;
    cursor: pointer;
    outline:0;
}
```

```
.connect-button {
    background-position: 0px 0px;
    width: 73px;
    height: 22px;
    background-color: #ee3b1b ;
    border: 2px solid transparent;
    border-radius: 50px;
    color: White;
    font-family: Candara, "Segoe UI", Calibri, Segoe, Optima, Arial, sans-serif;
    font-size: 14px;
    text-align: center;
    cursor: pointer;
    outline:0;
}
```

```
#detect-device-container {
        margin-left: 28px;
        width: 252px;
        background-color: #EBE6E1;
        margin-top: 20px;
}
```

```
#detect-device-container {
        margin-left: 28px;
        width: 252px;
        background-color: #e7dd7b;
        margin-top: 20px;
}
```

```
.refresh-button {
    background-position: 0px 0px;
    width: 73px;
    height: 22px;
    background-color: #0A88FB;
    border: 2px solid transparent;
    border-radius: 50px;
    color: White;
    font-family: Candara, "Segoe UI", Calibri, Segoe, Optima, Arial, sans-serif;
    font-size: 14px;
    text-align: center;
    cursor: pointer;
    outline:0;
}
```

```
.refresh-button {
    background-position: 0px 0px;
    width: 73px;
    height: 22px;
    background-color: #ee3b1b;
    border: 2px solid transparent;
    border-radius: 50px;
    color: White;
    font-family: Candara, "Segoe UI", Calibri, Segoe, Optima, Arial, sans-serif;
    font-size: 14px;
    text-align: center;
    cursor: pointer;
    outline:0;
}
```

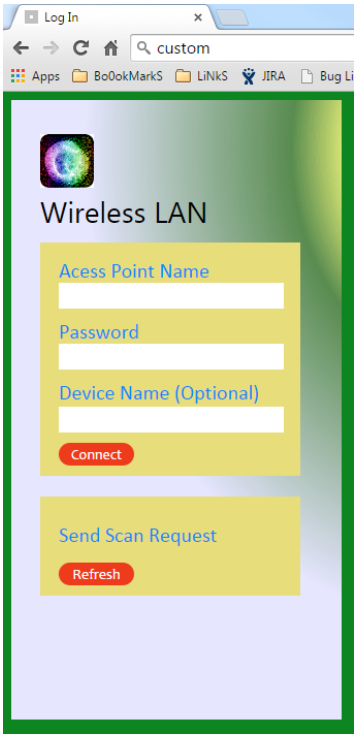
```
#log-in-form-container thead {
        font-weight: normal;
        color: #0A88FB;
        text-align: left;
        font-size:13px;
}
```

```
#log-in-form-container thead {
        font-weight: normal;
        color: #1af497;
        text-align: left;
        font-size:13px;
}
```

After generating and updating the WINC firmware image, the above modification results in the following HTTP provisioning page:

Before:

After:



6. Revision History

Doc. Rev.	Date	Comments
XXXXXE	03/2016	Added troubleshoot section and customizing webpage section.
XXXXXD	12/2015	Added support for both WINC1500 and WINC3400.
XXXXXC	02/2015	Updated firmware update procedure (batch file).
XXXXXB	12/2014	Updated firmware update procedure (serial bridge).
XXXXXA	11/2014	Initial document release.

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1)(408) 441-0311
Fax: (+1)(408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Road
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg.
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81)(3) 6417-0300
Fax: (+81)(3) 6417-0370