# **Application Note**

# IoT Application Examples for WINC1500 using SAM D21 Xplained Pro

**AN-007** 

# **Prerequisites**

- Hardware Prerequisites
  - Atmel SAM D21 Xplained Pro Evaluation Kit
  - Atmel WINC1500-MD1 extension
  - Atmel IO1 extension
  - USB Micro Cable (TypeA / MicroB)
- Software Prerequisites
  - Atmel Studio 6.2
  - Wi-Fi App Example codes

#### Introduction

This application note describes the Atmel WINC1500 Wi-Fi Network Controller to build state-of-the-art Internet Of Things (IOT) applications.

The following topics will be covered:

- How examples are organized.
- Target board information.
- Instruction for each example.



# **Table of Contents**

| Pre  | requis                         | sites  | 1  |  |  |  |  |  |
|------|--------------------------------|--|----|--|--|--|--|--|
| Intr | oducti                         | on   | 1  |  |  |  |  |  |
| Ico  | n Key                          | Identifiers  | 4  |  |  |  |  |  |
| 1.   | How The Examples Are Organized |  |    |  |  |  |  |  |
|      | 1.1                            | Basic Examples   | 5  |  |  |  |  |  |
|      | 1.2                            | Protocol Examples  | 5  |  |  |  |  |  |
|      | 1.3                            | Advanced Examples  | 6  |  |  |  |  |  |
| 2.   | Sour                           | ource Organization                                       |    |  |  |  |  |  |
| 3.   | How                            | How to Build & Execute                                   |    |  |  |  |  |  |
| 4.   | How                            | How to download new Firmware of WINC1500 Wi-Fi board?11  |    |  |  |  |  |  |
| 5.   | Example                        |  |    |  |  |  |  |  |
| ٠.   | 5.1                            | Basic Example: How to Get Chip ID                        |    |  |  |  |  |  |
|      | 5.2                            | Basic Example: How to Set Debug Level                    |    |  |  |  |  |  |
|      | 5.3                            | Basic Example: How to Get MAC Address                    |    |  |  |  |  |  |
|      | 5.4                            | Basic Example: How to Run STA Mode                       |    |  |  |  |  |  |
|      | 5.5                            | Basic Example: How to Run AP Mode                        |    |  |  |  |  |  |
|      | 5.6                            | Basic Example: How to Run P2P Mode                       |    |  |  |  |  |  |
|      | 5.7                            | Basic Example: How to Change Mode                        | 23 |  |  |  |  |  |
|      | 5.8                            | Basic Example: How to Scan APs                           | 24 |  |  |  |  |  |
|      | 5.9                            | Basic Example: How to Set Deep Sleep                     | 25 |  |  |  |  |  |
|      | 5.10                           | Basic Example: Security with WEP/WPA                     | 26 |  |  |  |  |  |
|      | 5.11                           | Basic Example: Connection to Enterprise Security Network | 28 |  |  |  |  |  |
|      | 5.12                           | Basic Example: Connection to Security WPS                | 30 |  |  |  |  |  |
|      | 5.13                           | Basic Example: How to Set Packet Monitoring              | 32 |  |  |  |  |  |
|      | 5.14                           | Basic Example: Get Signal Status                         | 34 |  |  |  |  |  |
|      | 5.15                           | Basic Example: AP Provision                              | 35 |  |  |  |  |  |
|      | 5.16                           | Basic Example: HTTP Provision                            | 37 |  |  |  |  |  |
|      | 5.17                           | Protocol Example: UDP (Server and Client)                | 39 |  |  |  |  |  |
|      | 5.18                           | Protocol Example: UDP Client                             | 42 |  |  |  |  |  |
|      | 5.19                           | Protocol Example: UDP Server                             | 43 |  |  |  |  |  |
|      | 5.20                           | Protocol Example: TCP Client                             | 44 |  |  |  |  |  |
|      | 5.21                           | Protocol Example: TCP Server                             |    |  |  |  |  |  |
|      | 5.22                           | Protocol Example: NTP Time Client                        |    |  |  |  |  |  |
|      | 5.23                           | Protocol Example: SMTP Send Email                        | 47 |  |  |  |  |  |



|    | 5.24             | Protocol Example: Location Client      | 49 |  |  |
|----|------------------|--|----|--|--|
|    | 5.25             | Advanced Example: Growl Notification   | 50 |  |  |
|    | 5.26             | Advanced Example: MQTT Chat            | 52 |  |  |
|    | 5.27             | Advanced Example: Weather Client       | 54 |  |  |
|    | 5.28             | Advanced Example: Wi-Fi Serial         | 55 |  |  |
|    | 5.29             | Advanced Example: OTA Firmware Upgrade | 57 |  |  |
| 6. | Conclusion       |  |    |  |  |
| 7  | Revision History |  |    |  |  |



# **Icon Key Identifiers**

**INFO** Delivers contextual information about a specific topic

TIPS Highlights useful tips and techniques

Info: 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. How The Examples Are Organized

This example package is consists of several example codes and projects. The Examples are organized different level of codes to explain WINC1500 API usage – from basic Wi-Fi operations, and then extends to advanced topic. Here's three level of category

- Basic Examples
- IOT Protocol Examples
- Advanced App Scenario

These example materials are delivered by Atmel Software Framework, or by Atmel FAE (such together with hands-on documentation, datasheets, application notes, software & tools).

#### 1.1 Basic Examples

These examples describe basic Wi-Fi operation in 'how-to' manner:

- How to read Chip ID(to identify WINC1500 H/W revision difference)
- How to adjust debug message level
- How to get MAC address of the Wi-Fi module
- How to start Wi-Fi in specific operation mode, such as:
  - STA Mode (Station mode, known as a Wi-Fi client)
  - AP mode (Access Point mode)
  - P2P mode (Peer-to-Peer mode, also known as Wi-Fi Direct®)
- How to switch mode between STA, AP and P2P modes during the runtime
- How to scan AP list that is nearby
- How to set deep sleep mode
- How to connect to secure Wi-Fi with using WEP/WPA/WPA2 security
- How to connect to enterprise security network
- How to connect to security WPS
- How to set packet monitoring
- How to get RF signal status by reading RSSI value
- How to set AP provision
- How to set HTTP provision

#### 1.2 Protocol Examples

After basic code examples, user may interest how to send and receive network packets. Here are protocol examples that can be extended for IoT application.

- UDP protocol example
  - Server and Client
  - Client
  - Server
- TCP protocol example
  - Client
  - Server
- NTP Time client retrieves network time for IoT application



- Send email send email from SMTP server
- Location client get the current location of the network provider using HTTP

## 1.3 Advanced Examples

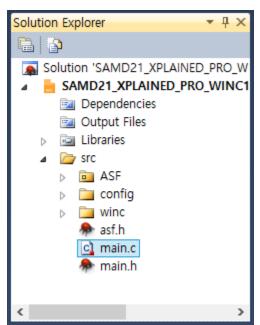
These examples demonstrate more complex example like

- Growl client demonstrates using RESTful API over SSL (essential for IoT application)
- MQTT Chat client demonstrate how to send and receive IoT information using MQTT protocol
- Weather client weather information for IoT application
- Wi-Fi serial useful for chatting or controlling a remote device
- OTA Firmware Upgrade WINC1500 Firmware upgrade via OTA server.

For customer's IoT application, these examples are useful to how to use WINC1500 APIs and implement a feature for IoT application.

# 2. Source Organization

Most of examples are simple as possible. It is consists of main.c and main.h.



Info: Some examples may have additional source file. But the structure is similar across the samples.

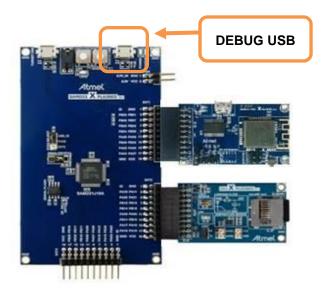
#### 3. How to Build & Execute

**EXECUTE** Build the solution (F7) and ensure you get no errors:

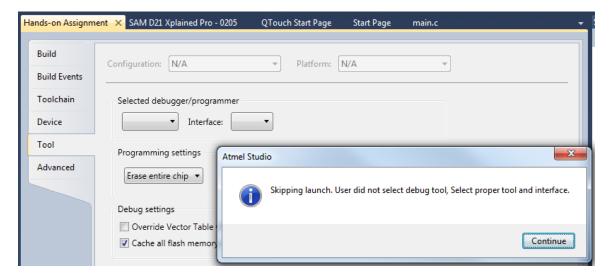
Info: Program the SAM D21 Xplained Pro.

 Connect the WINC1500 Wi-Fi extension and the IO1 extension to the SAM D21 Xplained Pro as displayed:



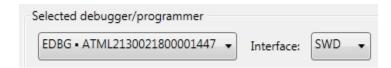


- Connect the SAM D21 Xplained Pro board to your PC using DEBUG USB connector.
- Program the application by clicking on the Start Debugging and Break icon:
- You will be asked to select your debug tool:

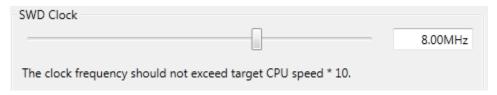




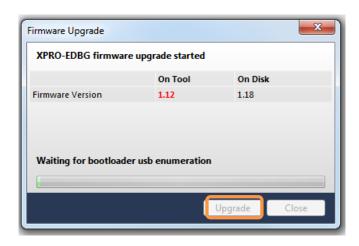
Select EDBG and SWD (Serial Wire Debug) as Interface:



Set SWD clock to 8 MHz to speed up programming:



- Click again on the Start Debugging and Break icon:
- The application will be programmed in the SAM D21 embedded flash and breaks at main function. MI Click on Continue to execute the application:
- Result: You may be asked to upgrade your EDBG firmware. If so, click on Upgrade:



- $\textbf{WARNING} \quad \text{Upgrade operation may take a few minutes, please } \underline{\textbf{wait}} \text{ for the operation to complete.}$
- RESULT The IOT application is now programmed and running.

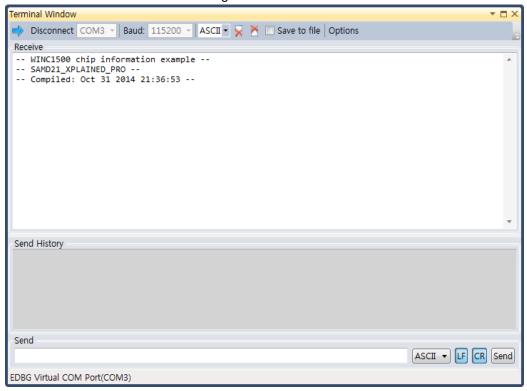
Open the SAMD21 EDBG DEBUG USB serial COM port, with the following settings:



- 115200 bauds,
- 8 bit data,
- no parity,
- one stop bit
- and no flow control

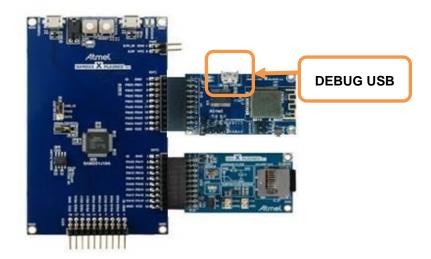
Open a serial terminal tool to see the result. You can use your preferred serial terminal, such as PuTTY, Tera Term, etc. You can also use terminal window plug-in in Atmel studio. You can install it through below menu.

Menu > Tools > Extension Manager



All the examples are using the same setup.





Connect the WINC1500 Xplained Pro board to your PC using DEBUG USB connector. You can get firmware debug information.

Open the WINC1500 DEBUG USB serial COM port, with the following settings:

- 115200 bauds,
- 8 bit data,
- no parity,
- one stop bit
- and no flow control

Open a serial terminal tool to see the result. (WINC1500 WIFI Debug information)

```
(1)Chip ID = 1502b1
(1)Flash ID = c21320c2, Size = 4 MBit
(1)Working Image offset = 0x3000 Rollback = 0x40000
(2)(M2M)(Efuse) successfully loaded from bank 1.
(2)EFUSE:MAC
(2)(M2M)MAC\_ADDR = F8:F0:05:20:AE:BF
(3)NMI M2M SW VERSION 17.0
(3)NMI MIN DRV VERSION 17.0
(3)Built at Sep 23 2014 13:09:07
(3)__ROM_FIRMWARE__
(4)(M2M)LOAD SEC
(6)(M2M)1000 400 2f000 2fc00 38000
(7)(M2M)Wifi Connect
(7)(M2M)SSID
                : NW01
(7)(M2M)AUTH
                : WPA-Personal
(7)(M2M)PSK
                : nmisemi2
(8)(M2M)Channel : 256
```



#### 4. How to download new Firmware of WINC1500 Wi-Fi board?

This guide demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to upgrade firmware of the WINC1500 Wi-Fi module. This is a basic operation to download firmware software. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header
- Connect the SAMD21(DEBUG USB PORT) to the USB port on your PC.



**TIPS** Do not connect SAMD21 VIRTUAL COM PORT via Terminal program.



serial\_bridge.c : Initialize SAMD21 and WINC1500 and enter to download mode.

- Build the program and download it into the board.
   \tools\Tools\samd21\_xplained\_pro\_serial\_bridge\samd21\_xplained\_pro\_serial\_bridge.atsln
- 2) Start the application.



**RESULT** The application is now programmed and running.

3) Launch the batch file. \tools\download\_all\_sb\_samd21\_xplained\_pro.bat





SUCCESS firmware download. Following information will be displayed on the command window.

```
SAMD21 flashing script: please connect edbg and power up the board.
Firmware check OK
Chiperase completed successfully
Firmware check OK
Programming completed successfully.
Mode UART
Can not find image builder path...
Downloading Image...
  >Programmer for WINC1500 SPI Flash<
     Owner: Atmel Corporation
************
>>Init Programmer
Detecting ports...
EDBG Virtual COM Port(COM4)
(APP)(INFO)WINC1500 Serial Bridge Found
Avail port COM4
1 of ports found
>Start erasing...
#Erase time = 4.633000 sec
>Start programming..
Done
#Programming time =6.661000 sec
Image downloaded successfully.
>>This task finished after 11.42 sec
##
                                                          ##
##
                ########
                           ###
                                 ######
                                         ######
                                                          ##
##
                ##
                     ##
                          ## ##
                                 ##
                                      ## ##
                                             ##
                                                          ##
##
                ##
                      ## ##
                             ##
                                 ##
                                        ##
                                                          ##
##
                ####### ##
                              ## ######
                                         ######
                                                          ##
##
                ##
                        ########
                                      ##
                                                          ##
##
                ##
                              ## ##
                                      ## ##
                                             ##
                                                          ##
##
                ##
                        ##
                              ## ######
                                         ######
                                                          ##
Downloading ends successfully
```





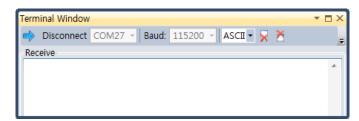
FAILED firmware download. Following information will be displayed on the command window.

```
SAMD21 flashing script: please connect edbg and power up the board.
Firmware check OK
Chiperase completed successfully
Firmware check OK
Programming completed successfully.
Mode UART
Can not find image builder path...
Downloading Image...
  >Programmer for WINC1500 SPI Flash<
     Owner: Atmel Corporation
************
>>Init Programmer
Detecting ports...
EDBG Virtual COM Port(COM4)
(APP)(ERR)[nm_bus_port_detect][567]Failed to find any COM ports
0 of ports found
(ERR)Failed To intilize programmer
Fail
##
                 ########
                           ###
                                       ##
                                                        ##
##
                 ##
                          ## ##
                                   ##
                                       ##
                                                        ##
##
                 ##
                          ##
                                   ##
                                       ##
                                                        ##
##
                 ######
                                   ##
                                       ##
                                                        ##
                         ##
##
                                       ##
                                                        ##
##
                  ##
                                   ##
                                       ##
                                                        ##
##
                 ##
                         ##
                               ##
                                  ####
                                       ########
                                                        ##
                                                        ##
```

If you meet the failed message, you must check your device assembled well.



WARNING You must check terminal program. Do not connect SAMD21 VIRTUAL COM PORT via Terminal program.



Retry firmware download again.



# 5. Example

## 5.1 Basic Example: How to Get Chip ID

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to retrieve the chip information of the Wi-Fi module. This is a basic operation to identify which HW version is used. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and retrieve information.

- 4) Build the program and download it into the board.
- 5) Start the application.



#### **RESULT**

The application is now programmed and running. Following information will be displayed on the terminal window. In the below result, you can see the chip ID of ID1502b1 and RF revision ID of 4. User must be aware of which version of WINC1500 module which is used.

```
-- WINC1500 chip information example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:--
Chip ID: 1502b1
RF Revision ID: 4
Done.
```



TIPS

WINC1500 behavior and corresponding log messages can be different according to the revision.



#### 5.2 Basic Example: How to Set Debug Level

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to configure debug level and print debugging messages of the Wi-Fi module. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and print debugging messages according to the debug level.

1) Enable debug mode by configuring the below code in **config\conf\_winc.h**:

```
/** Debug Options */
#define CONF_WIFI_M2M_DEBUG (1)
```

This example demonstrates setting debug level by M2M\_DEBUG\_LEVEL(...) macro to choose debug level.

```
for (int8_t i = 0; i < 5; i++) {
    /* Set debug level. */
    M2M_DEBUG_LEVEL(i);

    printf("\r[set debug level : %d]\r\n", (unsigned int)i);

    /* Display debug information. */
    printf("\r");
    M2M_PRINT("test message.\n");
    M2M_ERR("test message.\n");
    M2M_INFO("test message.\n");
    M2M_REQ("test message.\n");
    M2M_DBG("test message.\n");
    printf("\r\n");
}</pre>
```

- 2) Build the program and download it into the board.
- 3) Start the application.



**RESULT** 



```
-- WINC1500 debug level example --
-- SAMD21 XPLAINED PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
[set debug level : 0]
test message.
[set debug level : 1]
test message.
(APP)(ERR)[main][190]test message.
[set debug level : 2]
test message.
(APP)(ERR)[main][190]test message.
(APP)(INFO)test message.
[set debug level : 3]
test message.
(APP)(ERR)[main][190]test message.
(APP)(INFO)test message.
(APP)(R)test message.
[set debug level : 4]
test message.
(APP)(ERR)[main][190]test message.
(APP)(INFO)test message.
(APP)(R)test message.
(APP)(DBG)[main][193]test message.
Done.
```



TIPS

For a debugging session, user must turn on the debug option, and analyze the situation. Engineering team also requires the logs to review.



## 5.3 Basic Example: How to Get MAC Address

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to retrieve the MAC Address of the Wi-Fi module. The example uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



- 1) Build the program and download it into the board.
- 2) Start the application.

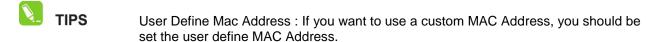


**RESULT** 

The application is now programmed and running. Following information will be displayed on the terminal window.

| WINC1500 MAC ADDRESS example      |  |
|-----------------------------------|--|
| SAMD21_XPLAINED_PRO               |  |
| Compiled: xxx xx xxxx xx:xx:xx    |  |
| OTP MAC Address : XX:XX:XX:XX:XX  |  |
| (0r)                              |  |
| JSER MAC Address : XX:XX:XX:XX:XX |  |





TIPS In this example result, you can see the OTP MAC Address or USER MAC Address.



## 5.4 Basic Example: How to Run STA Mode

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to behave as a station. The example uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and connect to AP as a station mode.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"
```

- 2) Build the program and download it into the board.
- 3) Start the application.



**RESULT** 

```
-- WINC1500 station mode example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
Connecting to XXXXXX.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
```



#### 5.5 Basic Example: How to Run AP Mode

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to behave as an AP. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and act as an AP.

1) Configure below code in main.h for AP information.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_OPEN

#define MAIN_WLAN_CHANNEL (6)
```

- 2) Build the program and download it into the board.
- 3) Start the application.



#### **RESULT**

```
-- WINC1500 AP mode example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
AP mode started. You can connect to XXXXXX.
Station connected
Station IP is xxx.xxx.xxx
```

- Info: The WINC1500 supports AP mode operation with the following limitations:
  - 1. Only ONE associated station is supported. After a connection is established with a station, further connections are rejected.
  - 2. OPEN and WEP security modes.
  - 3. The device could not work as a station in this mode (STA/AP Concurrency is not supported).



**EXECUTE** Now AP mode is ready. you can use smart phone to connect to the WINC1500 that is running as AP mode.





#### 5.6 Basic Example: How to Run P2P Mode

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to behave as a P2P device. P2P is also known as Wi-Fi Direct®. This demo uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and act as a P2P device.

1) Configure below code in main.h for P2P information.

```
#define MAIN_WLAN_DEVICE_NAME "WINC1500_P2P"
#define MAIN_WLAN_CHANNEL (6)
```

- 2) Build the program and download it into the board.
- 3) Start the application.



**RESULT** 

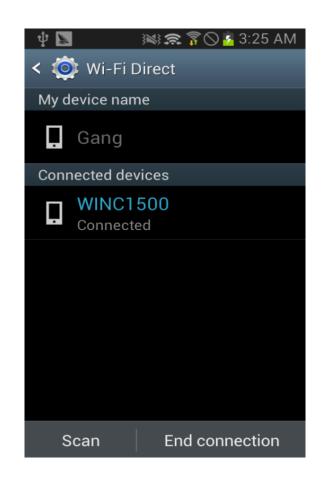
The application is now programmed and running. Following information will be displayed on the terminal window

```
-- WINC1500 P2P mode example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
P2P mode started. You can connect to XXXXXX.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
```



**EXECUTE** After running P2P mode, Open your smart phone's Wi-Fi Direct menu, then scan and connect to WINC1500 device.







#### 5.7 Basic Example: How to Change Mode

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to bring up as a station, an AP or a P2P mode. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c : Initialize the WINC1500. For the first time, the device acts as station mode, then switch to AP mode or P2P mode. For the details in each mode, refer to the MODE\_STA, MODE\_AP and MODE\_P2P example.

1) Configure below code in main.h for AP information.

2) Configure below code in main.h for P2P information.

```
#define MAIN_WLAN_DEVICE_NAME "WINC1500_P2P"
#define MAIN_WLAN_P2P_CHANNEL (6)
```

- 3) Build the program and download it into the board.
- 4) Start the application.



## **RESULT**

```
-- WINC1500 mode change example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
AP mode, start
AP mode, end
P2P mode, start
P2P mode, end
```



### 5.8 Basic Example: How to Scan APs

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to how to scan AP as a station. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and scan AP until defined AP is founded.

1) Configure below code in the main.h for AP to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"
```

- 2) Build the program and download it into the board.
- 3) Start the application.



#### **RESULT**

```
-- WINC1500 AP scan example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxxx xx:xx:xx --
[1] SSID:DEMO_AP1
[2] SSID:DEMO_AP2
[3] SSID:DEMO_AP
Found DEMO_AP
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx
```



#### 5.9 Basic Example: How to Set Deep Sleep

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to check the PS(PowerSave) mode. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1



main.c: Initialize the WINC1500 and set PS Mode and the get RSSI for the connected AP.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"
```

Configure below code in the config main.h for PowerSave mode.

```
#define MAIN_PS_SLEEP_MODE "M2M_PS_MANUAL"
```

- 3) Build the program and download it into the board.
- 4) On the computer, open and configure a terminal application as the follows.
- 5) Start the application
- 6) In the terminal window, the following text should appear:



RESULT

```
-- WINC1500 PowerSave mode example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx
RSSI for the current connected AP (-xx)
```



#### 5.10 Basic Example: Security with WEP/WPA

This example demonstrates how to connect WINC1500 Wi-Fi device to AP with WEP, WPA Security. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and connect to AP using Security mode WEP, WPA.

- 1) Assemble the devices and connect to USB debug cable from PC.
- 2) Prepare an AP that supports WEP and WPA/WPA2 Security and configure Wi-Fi Security. For more information, please refer to AP manufacturer's manual.
- 3) To test WPA Security, Use case 2 in main function and modify MAIN\_WLAN\_PSK in source file main.h.

```
#define MAIN_WLAN_PSK "12345678"
```

4) To test WEP Security, Use case 1 in main function and select wep64\_parameters or wep128\_parameters as security parameters. Modify MAIN\_WLAN\_DEVICE\_NAME, MAIN\_WLAN\_WEP\_KEY\_INDEX and MAIN\_WLAN\_WEP\_KEY\_40 or MAIN\_WLAN\_WEP\_KEY\_104 in source file main.h.

```
#define MAIN_WLAN_DEVICE_NAME "DEMO_AP"

#define MAIN_WLAN_WEP_KEY_INDEX 1

#define MAIN_WLAN_WEP_KEY_40 "1234567890"

#define MAIN_WLAN_WEP_KEY_104 "1234567890abcdef1234567890"
```

5) Run the application. If the device connected successfully, IP address which is assigned by DHCP will be displayed on the terminal program.





```
-- WINC1500 security connection with WEP,WPA security example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
Connecting to XXXXXXX.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
```



## 5.11 Basic Example: Connection to Enterprise Security Network

This example demonstrates how to connect WINC1500 Wi-Fi device to AP with WPA/WPA2 enterprise security. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and connect AP with Security WPA/PWA2 enterprise.

- 1) Assemble the devices and connect to USB debug cable from PC.
- 2) Prepare an AP that supports WPA/WPA2 enterprise security.

You need to know following things before configuring RADIUS server settings in the AP. Ask your network administrator to get there information and configure it in the AP.

- User name
- Password
- Name of wireless network
- Root certificate file
- 3) Download the root certificate to WINC1500 using RootCertDownload.bat file.
- 4) Modify MAIN\_WLAN\_802\_1X\_USR\_NAME, MAIN\_WLAN\_802\_1X\_PWD to the name and the password, respectively. Modify MAIN\_WLAN\_DEVICE\_NAME to wireless network name.

```
#define MAIN_WLAN_DEVICE_NAME "DEMO_AP"
#define MAIN_WLAN_802_1X_USR_NAME "atmeluser"
#define MAIN_WLAN_802_1X_PWD "12345678"
```

5) Build and run the application. If the device connected successfully, IP address which is assigned by DHCP will be displayed on the terminal program.





The application is now programmed and running. Following information will be displayed on the terminal window.

```
-- WINC1500 security connection with WPA/WPA2 enterprise example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
Connection successfully completed.
```

**WARNING** 

For using the security enterprise network, the root certificate must be installed. Download the root certificate using the root\_certificate\_downloader.(Refer to WINC1500 Software User Guide)

Info: 802.1x EAP(Extensible Authentication Protocol).

| 802.1x EAP<br>Type∉<br>Features /<br>Benefits∉ | MD5 ↓<br>↓<br>Message<br>Digest 5 ₽ | TLS ↓<br>↓<br>Transport Level<br>Security₽             | TTLS ↓<br>↓<br>Tunnel<br>Transport<br>Level Security∉ | PEAP ↓<br>↓<br>Protected<br>Transport<br>Level Security₽ | FAST ↓<br>↓<br>Flexible<br>Authentication via<br>Secure Tunneling↔ | LEAP ↓<br>↓<br>Lightweight<br>Extensible<br>Authentication<br>Protocol ₽ |
|--|-------------------------------------|--|---|--|--|--|
| Client-side<br>certificate<br>required         | No ₽                                | Yes ₽  | No∉   | No₽  | No↓<br>(PAC) ₽   | No÷  |
| Server-side<br>certificate<br>required         | No₽                                 | Yes ₽  | No₽   | Yes ₽  | No↓<br>(PAC) ₽   | No÷  |
| WEP Key<br>Manage                              | No₽                                 | Yes ₽  | Yes ₽   | Yes ₽  | Yes ₽  | Yes ₽  |
| Rogue AP<br>Detection                          | No₽                                 | No₽  | No₽   | No₽  | Yes ₽  | Yes ₽  |
| Provider∉                                      | MS ₽                                | MS ₽   | Funk ₽  | MS ₽   | Cisco ₽  | Cisco ₽  |
| Authentication<br>property∻                    | One way₽                            | Mutual₽  | Mutual₽   | Mutual₽  | Mutual₽  | Mutual₽  |
| Difficulty building                            | User<br>environment∂                | Difficulties(Due to the client certificate to build) ↔ | Normal₽   | Normal₽  | Normal₽  | Normal ₽   |
| Wi-Fi Security∉                                | Failure₽                            | Very High ₽  | High ₽  | High₽  | High₽  | High (When the use of strong passwords) ↔                                |



## 5.12 Basic Example: Connection to Security WPS

This example demonstrates how to connect WINC1500 Wi-Fi device to AP with WPS Security. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1



main.c: Initialize the WINC1500 and connect AP using WPS.

- 1) Assemble the devices and connect to USB debug cable from PC.
- 2) On the computer, open and configure a terminal application as the follows.
- 3) Prepare an AP that supports Wi-Fi Protected Setup(WPS)
- 4) To test WPS button method, Use case 1 in main function.
- 5) Configure below code in the main.h for WPS push button feature.

```
#define MAIN_WPS_PUSH_BUTTON_FEATURE "true"
```

- 6) Press WPS button at the AP. (For more information, please refer to AP product documentation)
- 7) Run the application. The WINC1500 will be connected to the AP automatically without security information.



## **RESULT**

```
-- WINC1500 security connection with Wi-Fi Protected Setup(WPS) example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
SW0 button pressed
Device is connecting using WPS Push Button option
Wi-Fi request WPS
SSID : xxxxxx, authtyp : x pw : xxxxxxxx
Request Wi-Fi connect
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
```

- 8) To test WPS PIN method, Use case 2 in main function.
- 9) Configure below code in the main.h for WPS PIN number and WPS push button feature.

```
#define MAIN_PIN_NUMBER "12345670"
#define MAIN_WPS_PUSH_BUTTON_FEATURE "true"
```



- 10) Enter WPS PIN number in the AP setup menu and start the AP. ( For more information, please refer to AP product documentation )
- 11) Run the application. The WINC1500 will be connected to the AP automatically without security information.



## **RESULT**

```
-- WINC1500 security connection with Wi-Fi Protected Setup(WPS) example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Wi-Fi request WPS
SSID : xxxxxx, authtyp : x pw : xxxxxxxx
Request Wi-Fi connect
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
```



### 5.13 Basic Example: How to Set Packet Monitoring

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to monitor all wireless data packets. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1



main.c: Initialize the WINC1500 and then configure to monitoring parameters and start the packet monitoring.

1) Please set the below MAC address in the main.c that you want to monitor.

```
static const uint8_t SRC_MAC_ADDR[6] = \{0x78, 0xF7, 0xBE, 0xFA, 0x92, 0x5A\}; static const uint8_t DST_MAC_ADDR[6] = \{0x78, 0xF7, 0xBE, 0xFA, 0x92, 0x5A\};
```

2) Activate one of the following codes in the start\_packet\_monitoring function.

```
memcpy(wifi_monitor_conf.au8SrcMacAddress, SRC_MAC_ADDR,
    sizeof(SRC_MAC_ADDR));
memcpy(wifi_monitor_conf.au8DstMacAddress, DST_MAC_ADDR,
    sizeof(DST_MAC_ADDR));
```

- 3) Build the program and download it into the board.
- 4) On the computer, open and configure a terminal application as the follows.
- 5) Start the application
- 6) In the terminal window, the following text should appear:





```
-- WINC1500 packet monitoring example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
Packet monitoring started.
Channel: 11
FrameType : 0xFF
FrameSubtype : 0xFF
Source MAC address : 00:00:00:00:00
Destination MAC address : 78:F7:BE:FA:92:5A
monitoring_cb() MONITOR PACKET u8FrameType:0x00, u8FrameSubtype:0xD0
monitoring_cb() SRC Mac address : 00:26:66:1A:08:5C
monitoring_cb() DST Mac address : 78:F7:BE:FA:92:5A
monitoring_cb() MONITOR PACKET u8FrameType:0x00, u8FrameSubtype:0x50
monitoring_cb() SRC Mac address : 00:26:66:C6:00:2A
monitoring_cb() DST Mac address : 78:F7:BE:FA:92:5A
monitoring_cb() MONITOR PACKET u8FrameType:0x08, u8FrameSubtype:0x88
monitoring_cb() SRC Mac address : 00:26:66:1A:08:5D
monitoring_cb() DST Mac address : 78:F7:BE:FA:92:5A
```



#### 5.14 Basic Example: Get Signal Status

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to check the signal strength such as RSSI. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and get RSSI for the connected AP.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"
```

- 2) Build the program and download it into the board.
- 3) Start the application.



#### **RESULT**

```
-- WINC1500 signal statistics example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:--
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx
RSSI for the current connected AP (-48)
```



#### 5.15 Basic Example: AP Provision

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to start Provision Mode. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and start Provision Mode until one of various APs is selected.

- 1) Build the program and download it into the board.
- 2) Start the application.

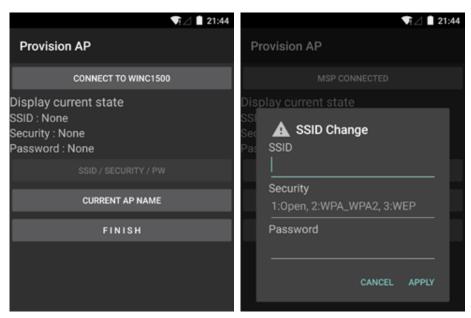


**RESULT** 

```
-- WINC1500 AP Provision example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
AP Provision mode started.
On the android device, connect to WINC1500_PROVISION_AP then run setting app.
socket_cb: Ready to listen.
```

- 3) Install provision\_ap.apk in the source package to your android device. You can also build the android application source and install it.
- 4) Connect your android device to WINC1500.
- 5) Launch the android application to configure AP, press the Connect button then SSID button will be available.
- 6) Input connection info, then press Apply button.





7) WINC1500 will be connected to the AP which you configured.

```
Wi-Fi connected. IP is xxx.xxx.xxx socket_cb: Client socket is created.

Disable to AP

Connecting to XXXXXX.

wifi_cb: CONNECTED

Wi-Fi connected. IP is xxx.xxx.xxx
```



# 5.16 Basic Example: HTTP Provision

## 5.16.1 HTTP Provisioning

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to start Provision Mode. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and start Provision Mode until one of various AP is selected.

1) Configure below code in main.h for provision information.

- 2) Build the program and download it into the board.
- 3) Start the application.



# **RESULT**

```
-- WINC1500 HTTP Provision example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Provision Mode started.
Connect to [atmelconfig.com] via AP[WINC1500_xx:xx] and fill up the page
```

- Connect your mobile device to WINC1500.
- 5) Browse the webpage (atmelconfig.com) to setup AP, fill up the page then press Connect.
- 6) WINC1500 will be connected to the AP what you configured.





```
wifi_cb: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx.xxx
wifi_cb: M2M_WIFI_RESP_PROVISION_INFO:
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx
```

# 5.16.2 Replace the logo in the provision page

Here is the way to change the logo image on the provision page.

- Replace the logo image in firmware release source directory.
   \tools\firmware\wifi\_v111\src\nmi\_m2m\source\http\Server\config\atmel-logo.png
- 2) Modify style.css in the same directory.

- 3) Re-download the firmware to load the new html page.(Refer to chapter 4)
- 4) Connect Provision AP again.





# 5.17 Protocol Example: UDP (Server and Client)

This program demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to test UDP socket. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the Wi-Fi module and test UDP server and client.

1) Configure below code in the main.h for AP information to be connected.

- 2) Build the program and download it into the board both device A and device B.
- 3) Start the application of device A and device B.





```
-- WINC1500 UDP example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
m2m_wifi_state: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
m2m_wifi_state : M2M_WIFI_REQ_DHCP_CONF : IP is xxx.xxx.xxx.xxx
m2m_wifi_socket_handler: bind success!
main: message sent
wifi_nc_data_callback: received app message
UDP test Complete!
```





```
-- WINC1500 UDP example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
m2m_wifi_state: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
m2m_wifi_state : M2M_WIFI_REQ_DHCP_CONF : IP is xxx.xxx.xxx.xxx
m2m_wifi_socket_handler: bind success!
main: message sent
wifi_nc_data_callback: received app message
UDP test Complete!
```



# 5.18 Protocol Example: UDP Client

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to test UDP socket. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the Wi-Fi module and test UDP server.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"

#define MAIN_WIFI_M2M_PRODUCT_NAME "NMCTemp"

#define MAIN_WIFI_M2M_SERVER_IP ØxFFFFFFFF

#define MAIN_WIFI_M2M_SERVER_PORT (6666)

#define MAIN_WIFI_M2M_REPORT_INTERVAL (1000)
```

- 2) Build the program and download it into the board.
- 3) Start the application.



# **RESULT**

```
-- WINC1500 UDP client example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED : CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF : IP is xxx.xxx.xxx
main: message sent
. . .
main: message sent
UDP client test Complete!
```



# 5.19 Protocol Example: UDP Server

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to test UDP socket. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the Wi-Fi module and test UDP client.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"

#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK

#define MAIN_WLAN_PSK "12345678"

#define MAIN_WIFI_M2M_PRODUCT_NAME "NMCTemp"

#define MAIN_WIFI_M2M_SERVER_IP ØxFFFFFFFF

#define MAIN_WIFI_M2M_SERVER_PORT (6666)

#define MAIN_WIFI_M2M_REPORT_INTERVAL (1000)
```

- 2) Build the program and download it into the board.
- 3) Start the application.



# **RESULT**

```
-- WINC1500 UDP server example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED : CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF : IP is xxx.xxx.xxx
socket_cb: bind success!
socket_cb: received app message.(1)
. . .
socket_cb: received app message.(10)
UDP server test Complete!
```



# 5.20 Protocol Example: TCP Client

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to test TCP client. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the Wi-Fi module and test TCP client.

1) Configure below code in main.h for AP information to be connected.

- 2) Build the program and download it into the board.
- 3) Start the application.



# **RESULT**

```
-- WINC1500 TCP client example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
m2m_wifi_state: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx
socket_cb: connect success!
socket_cb: send success!
socket_cb: recv success!
TCP Client Test Complete!
```



# 5.21 Protocol Example: TCP Server

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to test TCP server. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the Wi-Fi module and test TCP server.

1) Configure below code in main.h for AP information to be connected.

- 2) Build the program and download it into the board.
- 3) Start the application.



## **RESULT**

```
-- WINC1500 TCP server example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx
socket_cb: bind success!
socket_cb: listen success!
socket_cb: accept success!
socket_cb: recv success!
socket_cb: send success!
TCP Server Test Complete!
close socket
```



# 5.22 Protocol Example: NTP Time Client

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to retrieve time information from time server. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the chip and retrieve info.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"
#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK
#define MAIN_WLAN_PSK "12345678"
```

- 2) Build the program and download it into the board.
- 3) Start the application.



RESULT

The application is now programmed and running. Following information will be displayed on the terminal window.

```
-- WINC1500 time client example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx
m2m_ip_resolve_handler : DomainName pool.ntp.org
socket_cb: The GMT time is xx:xx:xx
```

**WARNING** If the server connection is unstable may not be operated normally.



# 5.23 Protocol Example: SMTP Send Email

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to send email for SMTP server. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the chip and send email.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN WLAN SSID
                              "DEMO AP"
#define MAIN WLAN AUTH
                              M2M WIFI SEC WPA PSK
#define MAIN WLAN PSK
                              "12345678"
                              "<sender@gmail.com>"
#define MAIN_SENDER_RFC
                              "<recipient@gmail.com>"
#define MAIN_RECIPIENT_RFC
#define MAIN_TO_ADDRESS
                              "recipient@gmail.com"
                              "sender@gmail.com"
#define MAIN FROM ADDRESS
                              "12345678"
#define MAIN_FROM_PASSWORD
```

- 2) Build the program and download it into the board.
- 3) Start the application.



# **RESULT**

The application is now programmed and running. Following information will be displayed on the terminal window.

```
-- WINC1500 send email example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
wifi_cb: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
wifi_cb: M2M_WIFI_REQ_DHCP_CONF: IP is xxx.xxx.xxx
Host IP is 173.194.72.108
Host Name is smtp.gmail.com
Recipient email address is recipient@gmail.com
main: Email was successfully sent.
```



For using the Gmail, the root certificate must be installed. Download the root certificate using the root\_certificate\_downloader.(Refer to WINC1500 Software User Guide)



- **WARNING** If the server connection is unstable may not be operated normally.
- TIPS

  Limitations/Known issues
  1. Email is sent to only 1 recipient.
  2. Only plain text Email is supported.



#### 5.24 **Protocol Example: Location Client**

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to get location of the network provider. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the WINC1500 and get location information.

4) Configure below code in main.h for AP information to be connected.

```
#define MAIN WLAN SSID
                              "DEMO AP"
#define MAIN WLAN AUTH
                              M2M WIFI SEC WPA PSK
#define MAIN WLAN PSK
                              "12345678"
```

- 5) Build the program and download it into the board.
- 6) Start the application.



### RESULT

The application is now programmed and running. Following information will be displayed on the terminal window.

```
-- WINC1500 location client example --
-- SAMD21 XPLAINED PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
Connected
Request complete
Received response 200 data size 178
Location : {latitude},{longitude}
Disconnected reason:-104
```



## WARNING

If disconnect reason is equal to -ECONNRESET(-104), it means Server disconnected your connection due to the keep alive timeout. This is a normal operation.

This example obtains the location of your network provider not your current position.



**WARNING** If the server connection is unstable may not be operated normally.



# 5.25 Advanced Example: Growl Notification

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro.

This example transmits a notification from the WINC1500 device (based on a certain trigger) to a public remote server which in turn sends it a phone application.

The initiated notification from the WINC1500 device is directed to a certain subscriber on the server. The supported applications are PROWL (for iPhone notifications) and NMA (for ANDROID notifications).

It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize growl and send notification message.

- 1) Build the program and download it into the board.
- 2) Start the application.



**RESULT** 

```
-- WINC1500 simple growl example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Provision Mode started.
Connect to [atmelconfig.com] via AP[WINC1500_08:CA] and fill up the page.
```

- 3) Connect your mobile device to WINC1500.
- 4) Browse the webpage for setting AP, fill up the page then press Connect.
- 5) WINC1500 will be connected to the AP what you entered.
- 6) Growl message will be sent.

```
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx
wifi_cb: M2M_WIFI_RESP_PROVISION_INFO.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx
send Growl message
Growl CB: 20
```



This example supports sending GROWL notifications to the following servers.

- PROWL for iOS push notifications (https://www.prowlapp.com/)
- NMA for Android push notifications (http://www.notifymyandroid.com/).

In order to enable the GROWL application (for sending notifications) working, you need to set your own API key to represent your account. Create your own by following instructions:

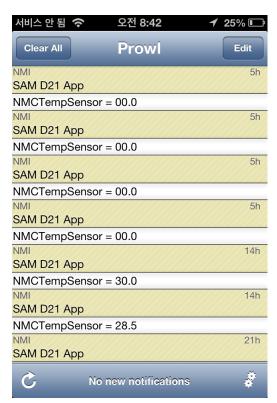
- Create a NMA account at http://www.notifymyandroid.com/ and create an API key. Copy the
  obtained key string in the file main.h in the macro NMA\_API\_KEY as the following.
- Create a PROWL account at https://www.prowlapp.com/ and create an API key. Copy the obtained API key string in the file main.h in the macro PROWL\_API\_KEY as the following.

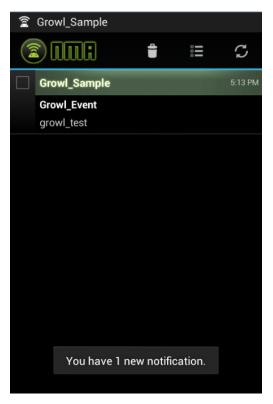
#define NMA\_API\_KEY "f8bd3e7c9c5c10183751ab010e57d8f73494b32da73292f6"
#define PROWL API KEY "117911f8a4f2935b2d84abc934be9ff77d883678"



For using the growl, the root certificate must be installed. Download the root certificate using the root\_certificate\_downloader.(Refer to WINC1500 Software User Guide)

Launch the Growl or NMA application to receive notification.







# 5.26 Advanced Example: MQTT Chat

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to chat using MQTT protocol. It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the board, connect to an MQTT broker and chat with the other devices.

mqtt.c, mqtt.h: Implementation of MQTT 3.1

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN_WLAN_SSID "DEMO_AP"
#define MAIN_WLAN_AUTH M2M_WIFI_SEC_WPA_PSK
#define MAIN_WLAN_PSK "12345678"
```

- 2) Build the program and download it into the board.
- 3) Start the application.
- 4) On the terminal window, enter the user name through the terminal window.

```
-- WINC1500 Wi-Fi MQTT chat demo --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Preparation of the chat has been completed.
```

- **WARNING** The user name should not contain a white space.
- 5) After chatting initialization completed, please enjoy the chat.





The Initialization may take longer than a few minutes according to the network environment.



**RESULT** 

Application is now successfully running. Below shows two user are talking to each other device connected to the same MQTT chat server.

```
Enter the user name (Max 64 characters)
User : demo_user
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
Preparation of the chat has been completed.
demo_user >> hi!
demo_user >> anybody there?
other_user >> I'm here
other_user >> hi
```

- WARNING Maximum message length should be shorter than 128-bytes
- **WARNING** If the server connection is unstable may not be operated normally.



#### 5.27 **Advanced Example: Weather Client**

This example demonstrates the use of the WINC1500 with the SAMD21 Xplained Pro board to retrieve weather information from weather server (openweathermap.org). It uses the following hardware:

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize the chip and retrieve info.

- 1) Build the program and download it into the board.
- 2) Start the application.



RESULT

The application is now programmed and running. Following information will be displayed on the terminal window.

```
-- WINC1500 weather client example --
-- SAMD21 XPLAINED PRO --
-- Compiled: xxx xx xxxx xx:xx:xx -
Provision Mode started.
Connect to [atmelconfig.com] via AP[WINC1500_08:CA] and fill up the page.
```

- 3) Connect your mobile device to WINC1500.
- 4) Browse the webpage for setting AP, fill up the page then press Connect.
- 5) WINC1500 will be connected to the AP what you entered.
- 6) The weather info will be printed.

```
Wi-Fi IP is xxx.xxx.xxx.xxx
wifi_cb: M2M_WIFI_RESP_PROVISION_INFO.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx
Host IP is 144.76.102.166
Host Name is openweathermap.org
City: Seoul
Temperature: 20.63
Weather Condition: sky is clear
```



WARNING If the server connection is unstable may not be operated normally.



#### 5.28 Advanced Example: Wi-Fi Serial

This example demonstrates how to emulate serial ports between two devices. It reads input data from serial interface and sends it via Wi-Fi connection and terminal window will print the messages which you typed or received. It can be useful for chatting or controlling a remote device. It uses the following hardware and you need to prepare two pairs of SAMD21 and WINC1500 boards.

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize devices and USART interface. Create TCP sockets, send/receive messages and print them on the terminal window.

1) Configure below code in main.h for AP information to be connected.

```
#define MAIN WLAN SSID
                              "DEMO AP"
#define MAIN_WLAN_AUTH
                              M2M_WIFI_SEC_WPA_PSK
                              "12345678"
#define MAIN WLAN PSK
```

- 2) Build the program and download it into the board.
- Start the application.
- 4) On the terminal window, the following text should appear:

```
-- WINC1500 Wi-Fi Serial example --
-- SAMD21_XPLAINED_PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
Wi-Fi connected.
Wi-Fi IP is xxx.xxx.xxx.xxx
socket_cb: bind success.
socket_cb: listen success.
```

5) Check IP address of each board and execute connection on one device by typing the below command on the terminal window with the other device's address. Use prefix "<<" to execute local commands.

```
<connect xxx.xxx.xxx.xxx
```

6) If connected, the following text should appear:



```
(Local device)
Connecting to [xxx.xxx.xxx] ...
Socket connect success.

(Remote device)
Socket accept success.
```

- 7) Type messages on the terminal window and you will see the sent/received messages.
- 8) You can control LED on the remote device by typing the following command. Use prefix ">>" to execute remote commands.

```
>>control ledon
(Or)
>>control ledoff
```



RESULT

Application reads data from serial interface.



TIPS

User commands can be modified to execute various actions.



#### 5.29 **Advanced Example: OTA Firmware Upgrade**



This project is available for a specific H/W version of WINC1500 which has 4Mb flash memory.

This example demonstrates how to upgrade WINC1500 Firmware via OTA. It downloads WINC1500 firmware from OTA Download server. OTA Download server is a web server. You can upload a new firmware image and download it to your device. It uses the following hardware and you need to prepare two pairs of SAMD21 and WINC1500 boards and a web server.

- The SAMD21 Xplained Pro
- The WINC1500 on EXT1 header



main.c: Initialize devices and set the server information. It connects to the OTA Download Server.

- 1) Set your OTA Download server.
- Upload OTA firmware binary to the root folder in your server. (e.g.

```
http://192.168.0.137/m2m ota.bin)
```

3) Configure below code in main.h for AP information to be connected.

```
#define MAIN WLAN SSID
                              "DEMO AP"
#define MAIN_WLAN_AUTH
                              M2M_WIFI_SEC_WPA_PSK
                              "12345678"
#define MAIN_WLAN_PSK
#define MAIN OTA URL
                              "http://192.168.0.137/m2m_ota.bin"
```

- 4) Build the program and download it into the board.
- 5) Start the application.
- 6) The following text should appear on the terminal window:

```
-- WINC1500 OTA Firmware upgrade example --
-- SAMD21 XPLAINED PRO --
-- Compiled: xxx xx xxxx xx:xx:xx --
m2m_wifi_connect.
Wi-Fi connected
Wi-Fi IP is xxx.xxx.xxx.xxx.
```

When you get the IP address then OTA get started.

7) The following text should appear on the terminal window:



```
(1)Chip ID = 1502b1
(1)Flash ID = c21320c2, Size = 4 MBit
(1)Working Image offset = 0x3000 Rollback = 0x40000
(2)(M2M)(Efuse) successfully loaded from bank 1.
(2)EFUSE:MAC
(2)(M2M)MAC\_ADDR = xx:xx:xx:xx:xx
(3)NMI M2M SW VERSION 17.0
(3)NMI MIN DRV VERSION 17.0
(3)Built at Sep 23 2014 13:09:07
(3) ROM FIRMWARE
(4)(M2M)LOAD SEC
(6)(M2M)1000 400 2f000 2fc00 38000
(7)(M2M)Wifi Connect
(7)(M2M)SSID : NW01
(7)(M2M)AUTH : WPA-Personal
(7)(M2M)PSK : nmisemi2
(8)(M2M)Channel : 256
(8)Reset MAC
(9)>> Sleep clk src <= Int. osc
(9)>> wakeup_delay = 1500 us
(9)>> pds = [652 652 6526 1957 3 ]
(9)-----
(489)MAC State <3>
(494)MAC State <4>
(494)Join on 11
(494)>> sta_wait_join 179
(494)MAC State <5>
(494)MAC State <6>
(495)Init Auth.
(495)MAC State <7>
(495)MAC State <9>
(495)MAC State <10>
(496)MAC State <1>
(496)!@#$ Rate DN (MCS-5) !@#$
(496)Assoc Success.
(498)(M2M)WIFI Connected
(499)(DHCP)<- DISCOVER
(500)Tsf join
(510)Tsf join Done
(532)(DHCP)-> OFFER
(563)(DHCP)-> ACK
(563)(DHCP)Self IP
                          : "xxx.xx.xxx.xxx"
```

You can find firmware version and build time information.

```
(3)NMI M2M SW VERSION 17.0
(3)NMI MIN DRV VERSION 17.0
(3)Built at Sep 23 2014 13:09:07
```



8) If connected to OTA Server, the following text should appear on the terminal window:

```
(770)(M2M)(OTA) Invalidate RB Image
Start Writing.. at 40000 Size 204380
(809)STATS 0 0 5 25 0, err: 16 - stuck: 0
(809)!@#$ Rate DN (48.0 Mbps) !@#$
(910)!@#$ Rate DN (MCS-4) !@#$
(1079)(M2M)Verification OK 204380
(1080)(M2M)(OTA) Update image done successfully
(1080)(M2M)Socket 6 Closed
(1083)(M2M)(OTA) SWap image done successfully
(1084)(M2M)(OTA) Switching firmware done.
```

9) If success OTA firmware upgrading, the following text should appear on the terminal window:

```
Wi-Fi IP is xxx.xxx.xxx.xxx
OtaUpdateCb 1 0
OtaUpdateCb m2m_ota_switch_firmware start.
OtaUpdateCb 2 0
OTA Success. Press reset your board.
```



**RESULT** 

Application reads data from serial interface.

```
(1)Chip ID = 1502b1
(1)Flash ID = c21320c2, Size = 4 MBit
(1)Working Image offset = 0x3000 Rollback = 0x40000
(2)(M2M)(Efuse) successfully loaded from bank 1.
(2)EFUSE:MAC
(2)(M2M)MAC_ADDR = xx:xx:xx:xx:xx:xx
(3)NMI M2M SW VERSION 17.0
(3)NMI MIN DRV VERSION 17.0
(3)Built at Sep 23 2014 13:03:02
(3)__ROM_FIRMWARE__
```



# 6. Conclusion

This application note explains example of use of the Atmel WINC1500 Wi-Fi module along with the Wi-Fi Software API for the SAM D21 Xplained Pro board.

The following topics have been covered:

- Basic example of controlling WINC1500 module
- Simple protocol examples
- IoT application example

You have seen how the WIN1500 Wi-Fi module makes it easy to add Wi-Fi capabilities to a SAM D21 host MCU.



# 7. Revision History

| Doc. Rev. | Date    | Comments                 |
|-----------|---------|--------------------------|
| AN001     | 10/2014 | Initial document release |
| AN002     | 11/2014 | document release         |
| AN003     | 11/2014 | document release         |
| AN004     | 12/2014 | document release         |
| AN005     | 12/2014 | document release         |
| AN006     | 12/2014 | document release         |
| AN007     | 01/2015 | document release         |





# Enabling Unlimited Possibilities®

#### **Atmel Corporation**

1600 Technology Drive San Jose, CA 95110

**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

© 2012 Atmel Corporation. All rights reserved. / Rev.: XXXXA-10/14

Atmel<sup>®</sup>, Atmel logo and combinations thereof, Enabling Unlimited Possibilities<sup>®</sup>, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.

