Application Note

Exosite Cloud Connectivity using Atmel | SAM W25 Xplained Pro

AN-XXXXX

Prerequisites

- **Hardware Prerequisites**
  - Atmel | SMART SAM W25 Xplained Pro Board
  - Atmel I/O1 Xplained Pro extension
  - USB Micro Cable (TypeA / MicroB)

- **Software Prerequisites**
  - Atmel Studio 6.2 SP2
  - Atmel Studio Exosite demo project

Introduction

This document will demonstrate how to connect the SAM W25 Xplained PRO board to cloud backend services such as Exosite.

The following topics will be covered:

- Hardware overview for both SAM W25 Wi-Fi module and SAM W25 Xplained PRO evaluation board.
- How to connect to Exosite Cloud.
- How to create a dashboard to visualize the data.
- How to build and execute the Exosite demo

Table of Contents

Icon Key Identifiers

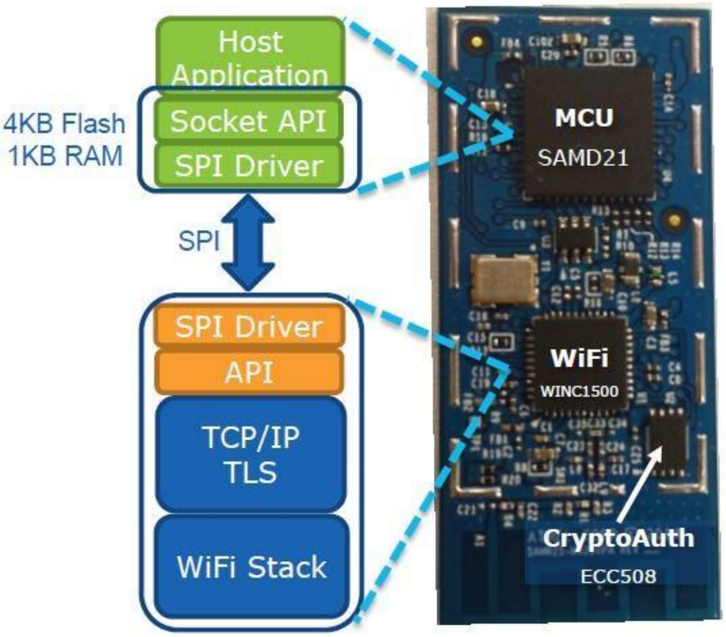
ℹ **INFO**	Delivers contextual information about a specific topic
✎ **TIPS**	Highlights useful tips and techniques
ℹ **Info:**	Highlights objectives to be completed
✓ **RESULT**	Highlights the expected result of an assignment step
! **WARNING**	Indicates important information
➔ **EXECUTE**	Highlights actions to be executed out of the target when necessary

1. Introduction.

1.1 Atmel | SMART SAM W25 Wi-Fi Module

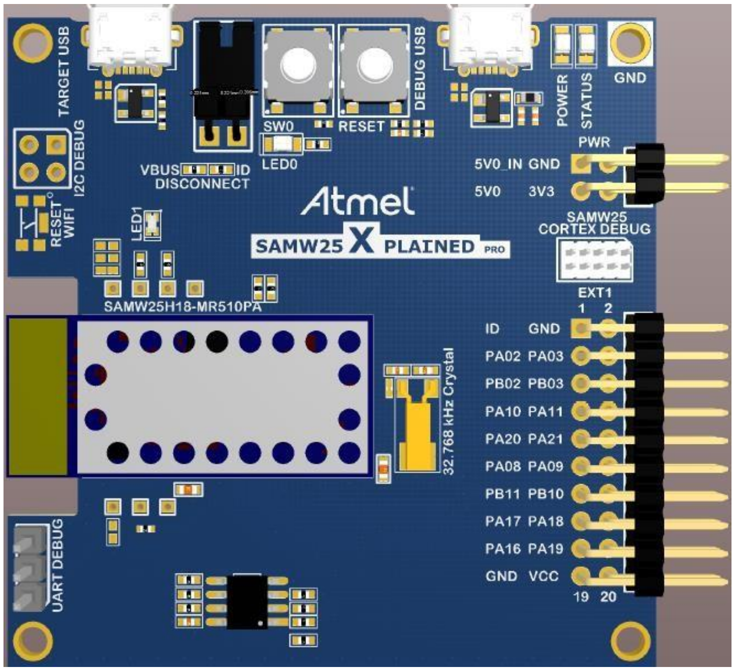
This highly integrated module offers the ideal solutions for designers seeking Wi-Fi connectivity. The Atmel | SMART SAM W25 Integrates TCP/IP stack on top of the Wi-Fi core and fully covers RF certifications such as FCC.

The SAM W25 Wi-Fi module is based on the industry-leading Atmel SmartConnect WINC1500 Wi-Fi core combined with the Atmel | SMART SAMD21 ARM Cortex M0+ based microcontroller. This turnkey system provides an integrated software solution with application and security protocols such as TLS and integrated network services (TCP/IP stack).



1.2 Atmel | SMART SAM W25 Xplained Pro Board

The SAM W25 Xplained Pro evaluation kit is a hardware platform to evaluate the SAM W25 Wi-Fi module. Supported by the Atmel Studio integrated development platform, the kit provides easy access to the features of the SAM W25 by including an on-board Embedded Debugger. Connection to any Xplained Pro Modules or external component is possible through available external connector, which make this board perfect for prototyping Wi-Fi low power applications.



- SAM W25 Wi-Fi Module
- One mechanical reset button
- One mechanical user pushbutton (wake-up, bootloader entry or general purpose)
- One yellow user LED
- 32.768kHz crystal
- USB interface, device and reduced host mode
- 8Mbit Serial Flash
- 1 Xplained Pro extension headers
- Embedded Debugger
  - Auto-ID for board identification in Atmel Studio
  - One yellow status LED
  - One green board power LED
  - Symbolic debug of complex data types including scope information
  - Programming and debugging
  - Data Gateway Interface: SPI, I²C, 4 GPIOs
  - Virtual COM port (CDC)
- USB powered

1.3 Atmel Studio 6

Atmel Studio 6 is the integrated development platform (IDP) for developing and debugging ARM® Cortex®-M processor-based and Atmel AVR® microcontroller (MCU) applications.

The Atmel Studio 6 IDP gives you a seamless and easy-to-use environment to write, build and debug your applications written in C/C++ or assembly code. Atmel Studio 6 supports all 8- and 32-bit AVR, the new SoC wireless family, Atmel | SMART SAM3 and SAM4 MCUs, and connects seamlessly to Atmel debuggers and development kits.

Download link: http://www.atmel.com/microsite/atmel_studio6/

Additionally, the IDP now includes two new features designed to further enhance your productivity: Atmel Gallery and Atmel Spaces:

- Atmel Gallery is an online apps store built in to Studio 6, allowing you to purchase both in-house and third-party development tools and embedded software.

  Link: http://gallery.atmel.com/

- Atmel Spaces is a collaborative workspace where you can securely share embedded design and track progress of projects with your peers.

  Link: http://spaces.atmel.com/gf/

INFO The Atmel Studio version used for this document is **Atmel Studio 6.2 with Service Pack 2**.

1.4 Atmel Software Framework (ASF)

The Atmel Software Framework (ASF) is a collection of embedded software for the Atmel megaAVR®, AVR XMEGA®, AVR UC3 Flash MCUs and ARM Cortex-M processor-based devices.

It simplifies the use of our MCUs by providing an abstraction to the hardware and high-value middleware. ASF is designed to be used for evaluation, prototyping, design and production phases.

ASF is integrated in the Atmel Studio IDP with a graphical user interface or available as standalone for GCC, IAR compilers. ASF can be downloaded for free.

ASF Standalone for GGC and IAR link: http://www.atmel.com/tools/AVRSOFTWAREFRAMEWORK.aspx

ASF Documentation: http://asf.atmel.com/docs/latest/

INFO The ASF version used for this document is **3.22.0**.

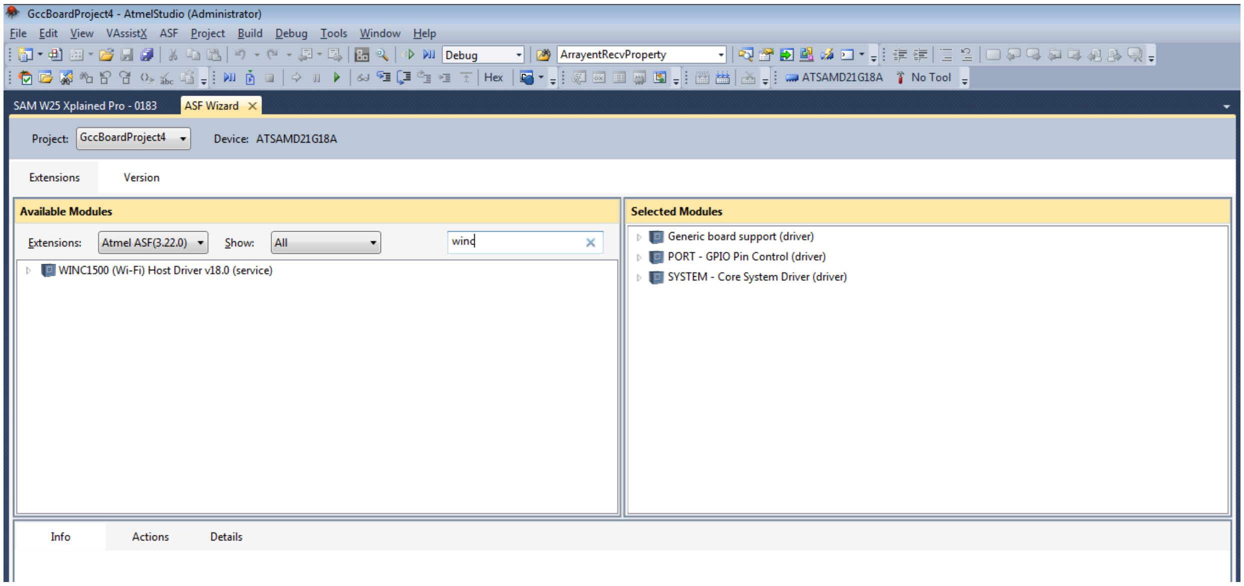
1.5 Atmel SmartConnect WINC1500 Host Driver

WINC1500 Host driver package contains the latest WINC1500 firmware.

1) Atmel Studio → Menu → ASF → ASF Wizard.

2) Write “winc” in the search window then you can find the WINC1500 Host Driver

Select the WINC1500 and proceed with download and installation.



GccBoardProject4 - AtmelStudio (Administrator)

Project: GccBoardProject4   Device: ATSAMD21G18A

Available Modules

Extensions: Atmel ASF(3.22.0)   Show: All   winc

- WINC1500 (Wi-Fi) Host Driver v18.0 (service)

Selected Modules

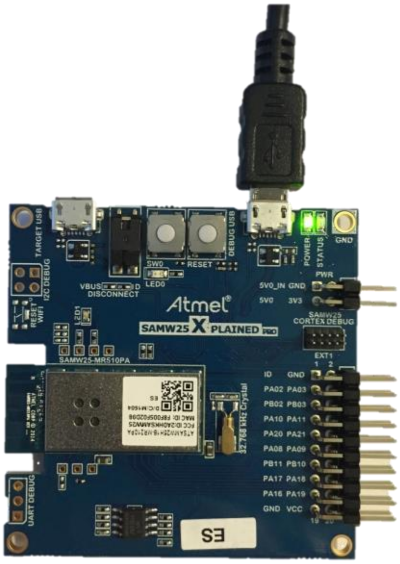
- Generic board support (driver)
- PORT - GPIO Pin Control (driver)
- SYSTEM - Core System Driver (driver)

1.6 How to download new Atmel SmartConnect WINC1500 Firmware

This section demonstrates how to upgrade the firmware of the WINC1500 chip on the SAM W25 Xplained Pro board. This is a basic operation to download firmware. It uses the following hardware:

- The SAM W25 Xplained Pro
- USB cable connected to the DEBUG USB PORT

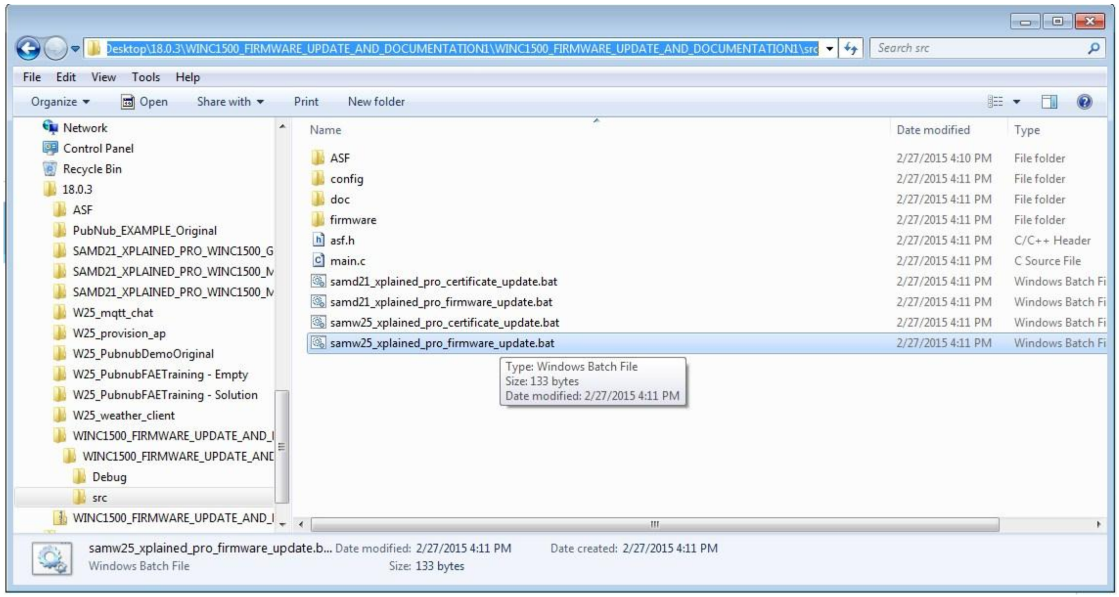
WARNING Do not connect SAM W25 VIRTUAL COM PORT to any terminal program such as Tera term or Putty because the firmware upgrade process will fail. Do not connect IO/1 extension at this point.



STEP 1: Firmware update to WINC1500 Board

1) Open *WINC1500_FIRMWARE_UPDATE_AND_DOCUMENTATION/src* project directory in Windows Explorer and look for samw25_xplained_pro_firmware_update.bat

2) Launch the batch file. ***\src\samw25_xplained_pro_firmware_update.bat*** as shown below


File Edit View Tools Help
Organize Open Share with Print New folder
Search src
Network
Control Panel
Recycle Bin
18.0.3
ASF
PubNub_EXAMPLE_Original
SAMD21_XPLAINED_PRO_WINC1500_G
SAMD21_XPLAINED_PRO_WINC1500_N
SAMD21_XPLAINED_PRO_WINC1500_N
W25_mqtt_chat
W25_provision_ap
W25_PubnubDemoOriginal
W25_PubnubFAETraining - Empty
W25_PubnubFAETraining - Solution
W25_weather_client
WINC1500_FIRMWARE_UPDATE_AND_I
WINC1500_FIRMWARE_UPDATE_ANI
Debug
src
WINC1500_FIRMWARE_UPDATE_AND_I

Desktop\18.0.3\WINC1500_FIRMWARE_UPDATE_AND_DOCUMENTATION1\WINC1500_FIRMWARE_UPDATE_AND_DOCUMENTATION1\src

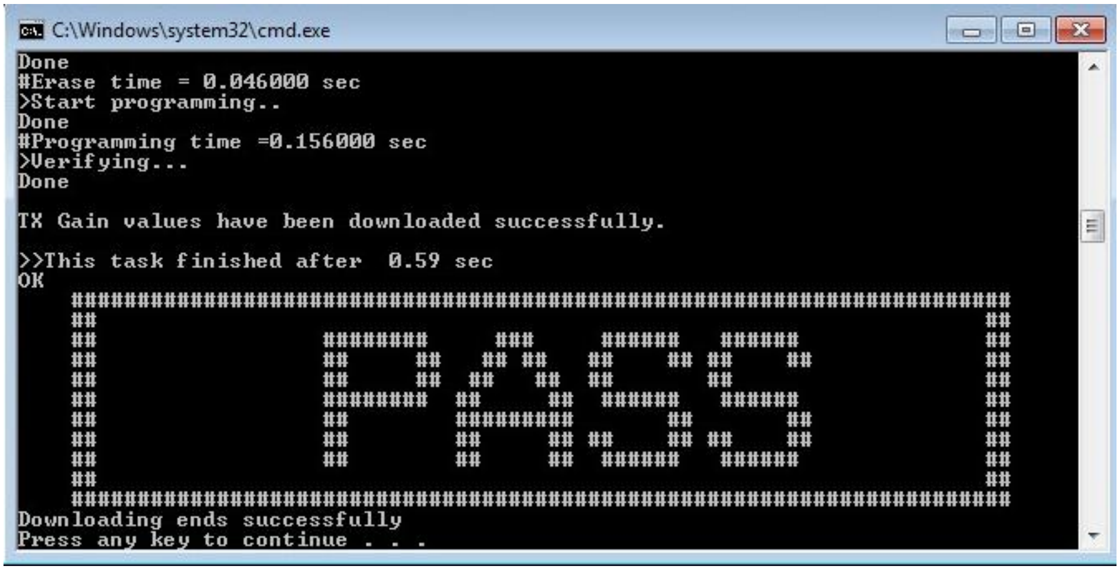
Name	Date modified	Type
ASF	2/27/2015 4:10 PM	File folder
config	2/27/2015 4:11 PM	File folder
doc	2/27/2015 4:11 PM	File folder
firmware	2/27/2015 4:11 PM	File folder
asf.h	2/27/2015 4:11 PM	C/C++ Header
main.c	2/27/2015 4:11 PM	C Source File
samd21_xplained_pro_certificate_update.bat	2/27/2015 4:11 PM	Windows Batch Fi
samd21_xplained_pro_firmware_update.bat	2/27/2015 4:11 PM	Windows Batch Fi
samw25_xplained_pro_certificate_update.bat	2/27/2015 4:11 PM	Windows Batch Fi
samw25_xplained_pro_firmware_update.bat	2/27/2015 4:11 PM	Windows Batch Fi

Type: Windows Batch File
Size: 133 bytes
Date modified: 2/27/2015 4:11 PM

samw25_xplained_pro_firmware_update.b... Date modified: 2/27/2015 4:11 PM Date created: 2/27/2015 4:11 PM
Windows Batch File Size: 133 bytes

This .bat file will program the SAM D21 MCU with the appropriate application and after will start the process to upgrade the firmware on the WINC1500 Wi-Fi chip.

RESULT SUCCESS firmware download. The following message will be displayed on the command window.

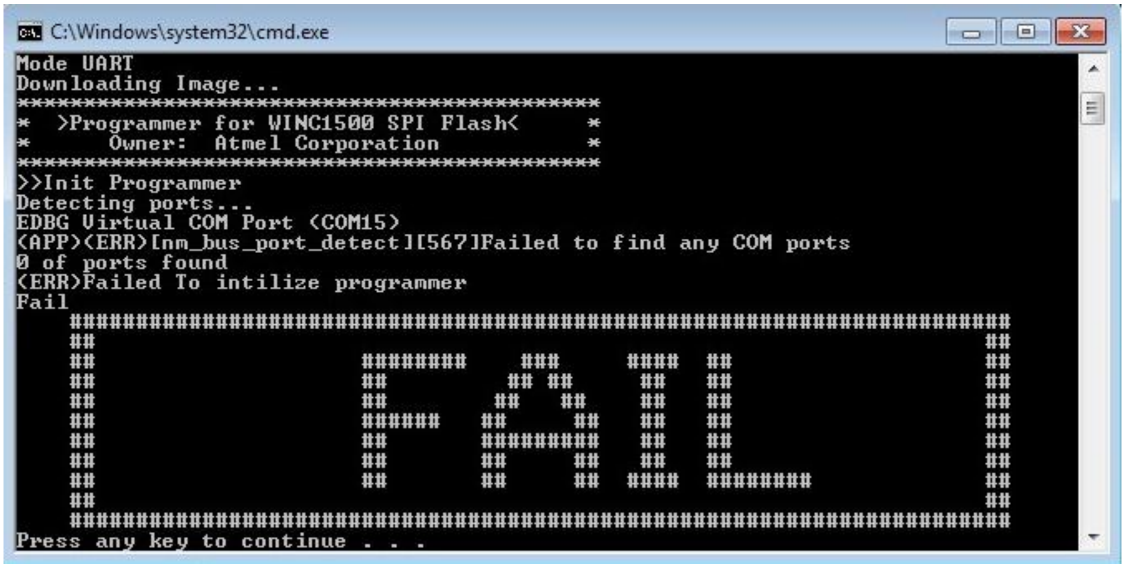


```
Done
#Erase time = 0.046000 sec
>Start programming..
Done
#Programming time =0.156000 sec
>Verifying...
Done

TX Gain values have been downloaded successfully.

>>This task finished after  0.59 sec
OK
    ########################################################################
    ##                                                                    ##
    ##                 ########     ###     ######   ######               ##
    ##                 ##     ##   ## ##   ##    ## ##    ##              ##
    ##                 ##     ##  ##   ##  ##       ##                    ##
    ##                 ########  ##     ##  ######   ######               ##
    ##                 ##        #########       ##       ##              ##
    ##                 ##        ##     ## ##    ## ##    ##              ##
    ##                 ##        ##     ##  ######   ######               ##
    ##                                                                    ##
    ########################################################################
Downloading ends successfully
Press any key to continue . . .
```

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



```
Mode UART
Downloading Image...
*********************************************
*  >Programmer for WINC1500 SPI Flash<      *
*      Owner:  Atmel Corporation           *
*********************************************
>>Init Programmer
Detecting ports...
EDBG Virtual COM Port (COM15)
(APP)(ERR)[nm_bus_port_detect][567]Failed to find any COM ports
0 of ports found
(ERR)Failed To intilize programmer
Fail
    ########################################################################
    ##                                                                    ##
    ##                    ########    ###     ####  ##                    ##
    ##                    ##         ## ##     ##   ##                    ##
    ##                    ##        ##   ##    ##   ##                    ##
    ##                    ######   ##     ##   ##   ##                    ##
    ##                    ##       #########   ##   ##                    ##
    ##                    ##       ##     ##   ##   ##                    ##
    ##                    ##       ##     ##  ####  ########              ##
    ##                                                                    ##
    ########################################################################
Press any key to continue . . .
```

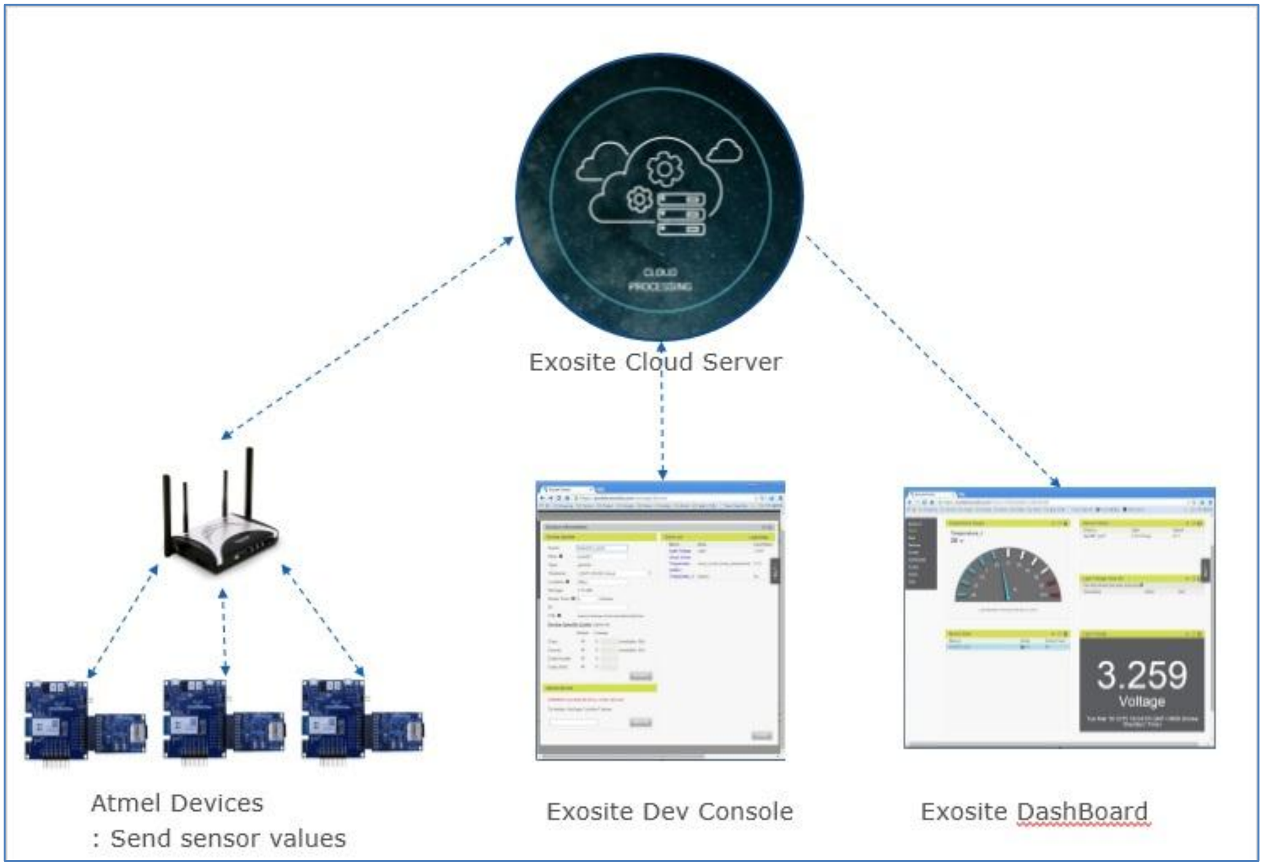
If you see a fail message as above, you must check your device is connected in the DEBUG USB port and also that no terminal software is currently using the com port of the SAM W25 board.

❗ **WARNING** Do not connect SAM W25 VIRTUAL COM PORT terminal program such as tera term or terminal window while programming WINC1500.

2. Overview

2.1 Adding connectivity to the Exosite Cloud Service

The goal of this document is to show you how to connect the SAM W25 to the cloud, in this particular example we will be using the cloud service provided by Exosite (http://www.exosite.com/)



The connectivity architecture of this document is shown as above.

We will have the SAM W25 boards publishing temperature and light data to the Exosite cloud service and then we will use the Exosite dev console to see the data and also to send a command to the SAM W25 to control the LED.

Also, we can create our own dashboard or use a default dashboard service to visualize the temperature and light data.

2.2 Exosite Basic HTTP API

Exosite is offering an HTTP rest API, more details about the API can be found as below:

http://docs.exosite.com/http/

This is the HTTP-based API for writing to and reading from Exosite.

In order to access to an API to make calls to read and write something in the Exosite, you need the device's Client Interface Key (CIK).

2.3 Exosite client module API

We are providing a module (composed of exosite_example.c and exosite_example.h files) that is handling the HTTP client to use the services provided by the Exosite HTTP API.

The APIs for this project are composed mainly of three functions as below:

- exosite_example_init()
- exosite_example_activiate()
- exosite_example_read_write()

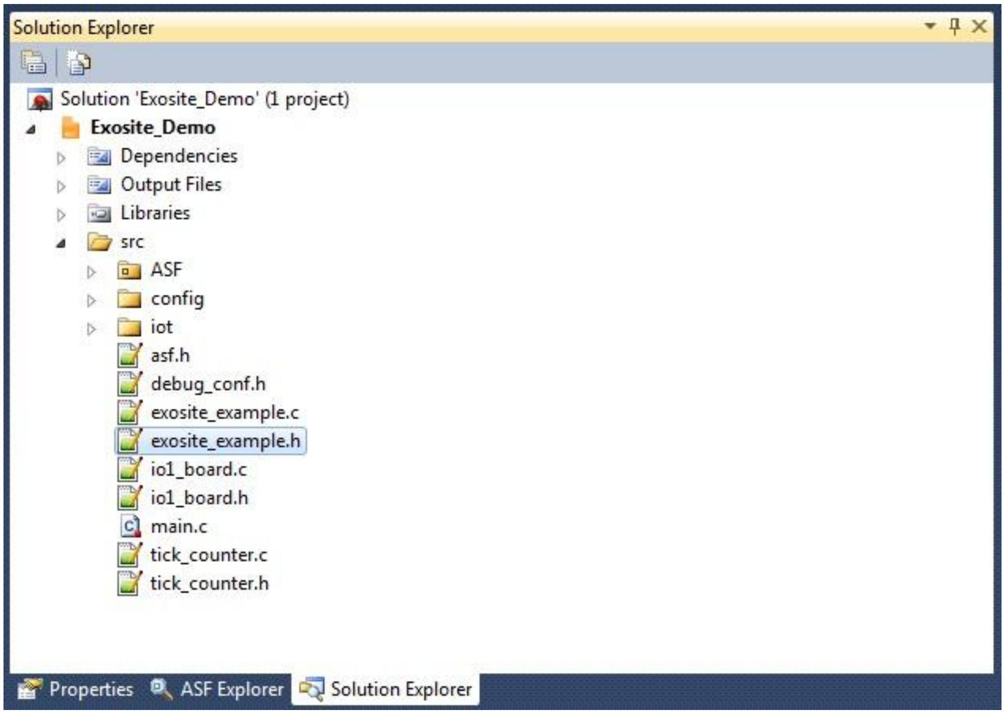
3. Getting started

3.1 Exosite_Demo Source Organization

1. If you get a zip file named *W25_Exosite.zip,* copy this file to your work space (ex. Desktop or C:\Atmel) and unzip it. This unzipped directory contains all the files needed in this application,
   - Open the directory where you just unzipped and open the following file

     Exosite_Demo.atsln

2. An Atmel Studio Project will be started and you can find the file “main.c” in this project. Then you will see all the features of this application described in the next section.

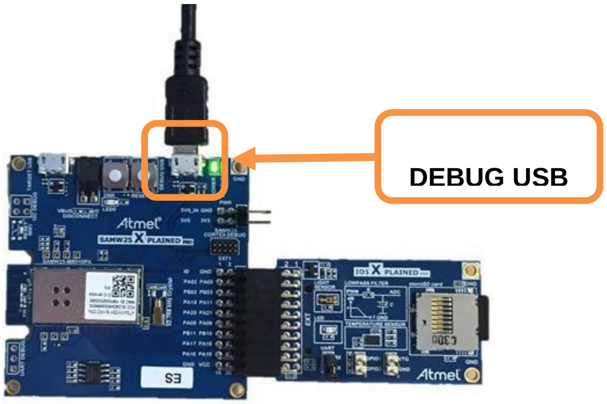


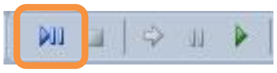
```
Solution 'Exosite_Demo' (1 project)
  Exosite_Demo
    Dependencies
    Output Files
    Libraries
    src
      ASF
      config
      iot
      asf.h
      debug_conf.h
      exosite_example.c
      exosite_example.h
      io1_board.c
      io1_board.h
      main.c
      tick_counter.c
      tick_counter.h
```

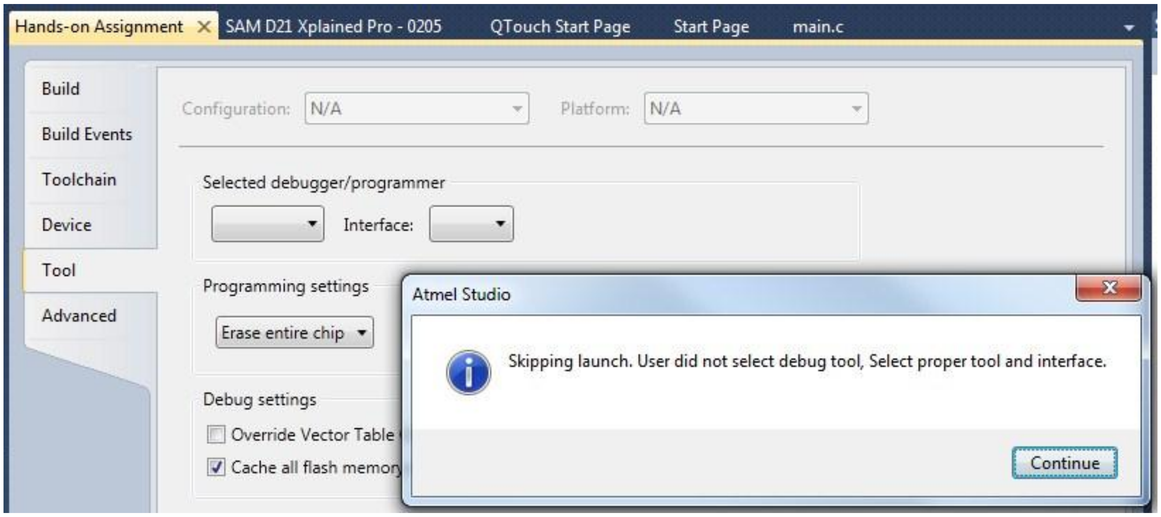
3. This application will show the following features :
   - Sets up Wi-Fi provisioning by using SoftAP
   - Makes Exosite demo application connect to Exosite cloud.
   - Receives LED commands from the cloud and turns LEDs on/off.
   - Periodically send temperature and light data to the cloud.

3.2 Program the SAM W25 Xplained Pro

- Connect the I/O1 extension to the SAM W25 Xplained Pro shown as below:



- Connect the SAM W25 Xplained Pro board to your PC by 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:



Configuration: N/A Platform: N/A

Selected debugger/programmer

Interface:

Programming settings

Erase entire chip

Debug settings

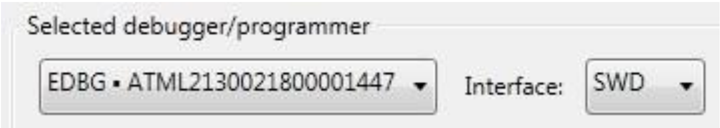
☐ Override Vector Table

☑ Cache all flash memory

Atmel Studio

Skipping launch. User did not select debug tool, Select proper tool and interface.

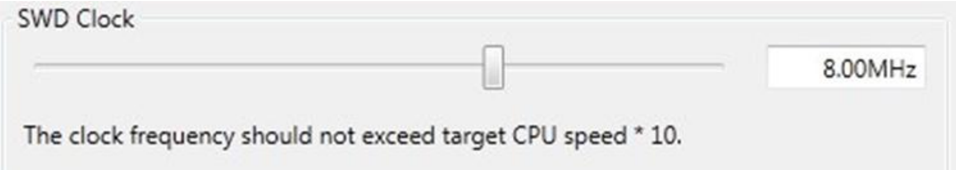
- Select EDBG and SWD (Serial Wire Debug) as Interface:



Selected debugger/programmer

EDBG • ATML2130021800001447 Interface: SWD

- Set SWD clock to 8MHz to speed up programming:

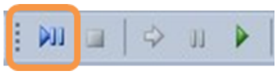


SWD Clock

8.00MHz

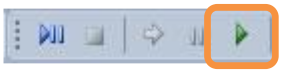
The clock frequency should not exceed target CPU speed * 10.

- Click again on the Start Debugging and Break icon:

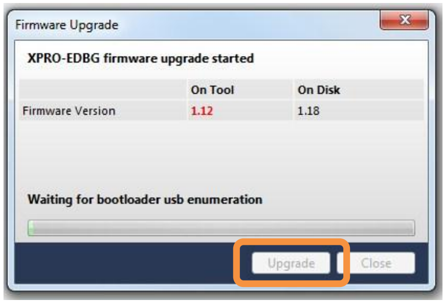


- The application will be programmed in the SAM D21 embedded flash and breaks at main function.

  Click on Continue to execute the application:



INFO You may be asked to upgrade your EDBG firmware. If so, click on Upgrade:



Firmware Upgrade

XPRO-EDBG firmware upgrade started

	On Tool	On Disk
Firmware Version	1.12	1.18

Waiting for bootloader usb enumeration

WARNING Upgrade operation may take a few minutes, please **wait** for the operation to complete.

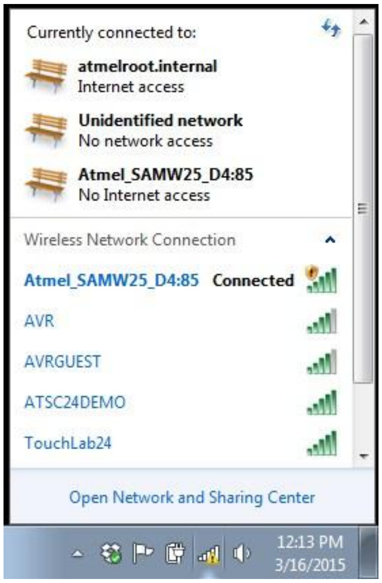
RESULT The Exosite demo application is now programmed and running.

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

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

3.3 AP provisioning

The SAM W25 module now should be in AP mode and be listed as a Wi-Fi network with the same name as shown in the below picture (Atmel_SAMW25_XX:XX). Notice that the last two bytes of the MAC address are appended to the SSID. Simply connect your PC/Smartphone to the module in AP mode.



Currently connected to:

atmelroot.internal
Internet access

Unidentified network
No network access

Atmel_SAMW25_D4:85
No Internet access

Wireless Network Connection

Atmel_SAMW25_D4:85 Connected

AVR

AVRGUEST

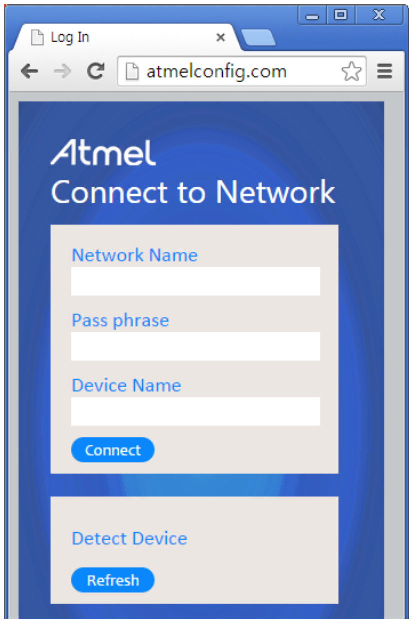
ATSC24DEMO

TouchLab24

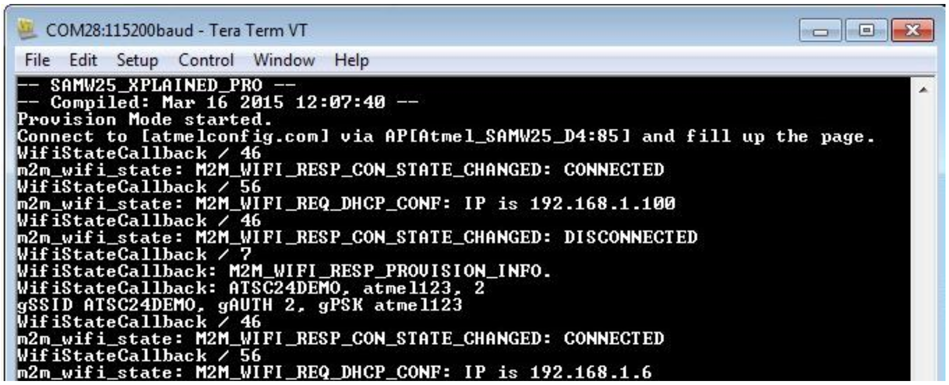
Open Network and Sharing Center

12:13 PM
3/16/2015

TO DO Once connected, just open your favorite web browser at the following address http://atmelconfig.com and provide the required Network Name (SSID) and Passphrase (Device Name can be blank) fields of the Wi-Fi AP the SAM W25 is supposed to connect.



RESULT Enter your SSID and Passphrase. Now your SAM W25 module is provisioned and connected to the provided Wi-Fi AP.



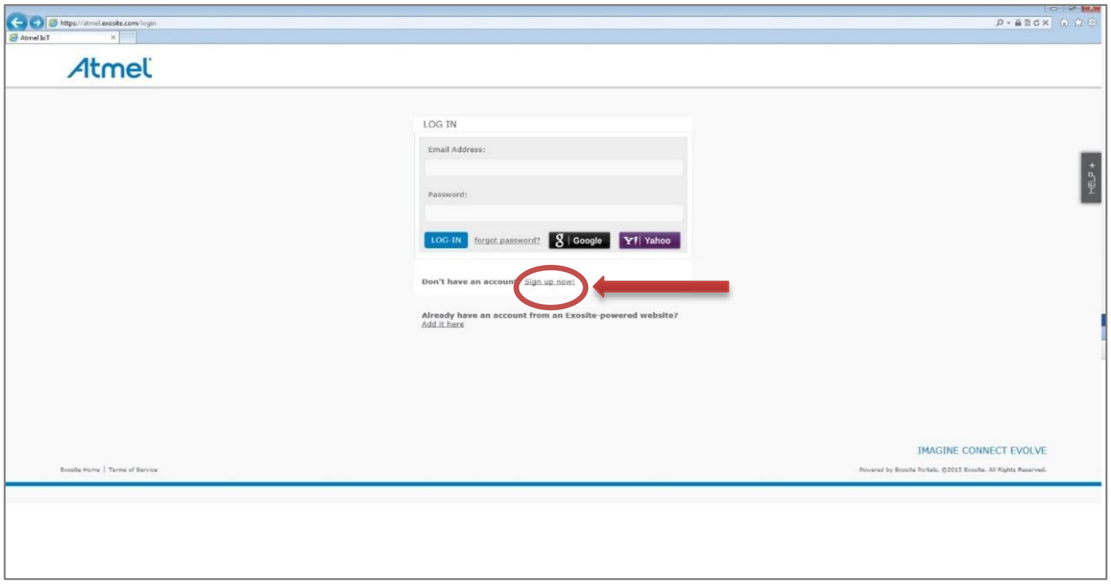
```
-- SAMW25_XPLAINED_PRO --
-- Compiled: Mar 16 2015 12:07:40 --
Provision Mode started.
Connect to [atmelconfig.com] via AP[Atmel_SAMW25_D4:85] and fill up the page.
WifiStateCallback / 46
m2m_wifi_state: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
WifiStateCallback / 56
m2m_wifi_state: M2M_WIFI_REQ_DHCP_CONF: IP is 192.168.1.100
WifiStateCallback / 46
m2m_wifi_state: M2M_WIFI_RESP_CON_STATE_CHANGED: DISCONNECTED
WifiStateCallback / 7
WifiStateCallback: M2M_WIFI_RESP_PROVISION_INFO.
WifiStateCallback: ATSC24DEMO, atmel123, 2
gSSID ATSC24DEMO, gAUTH 2, gPSK atmel123
WifiStateCallback / 46
m2m_wifi_state: M2M_WIFI_RESP_CON_STATE_CHANGED: CONNECTED
WifiStateCallback / 56
m2m_wifi_state: M2M_WIFI_REQ_DHCP_CONF: IP is 192.168.1.6
```

Now that the SAM W25 is connected to the AP with internet connectivity, it will immediately connect to the Exosite messaging service and will start sending the temperature and light data.

3.4 Viewing the published data on Exosite portal Part

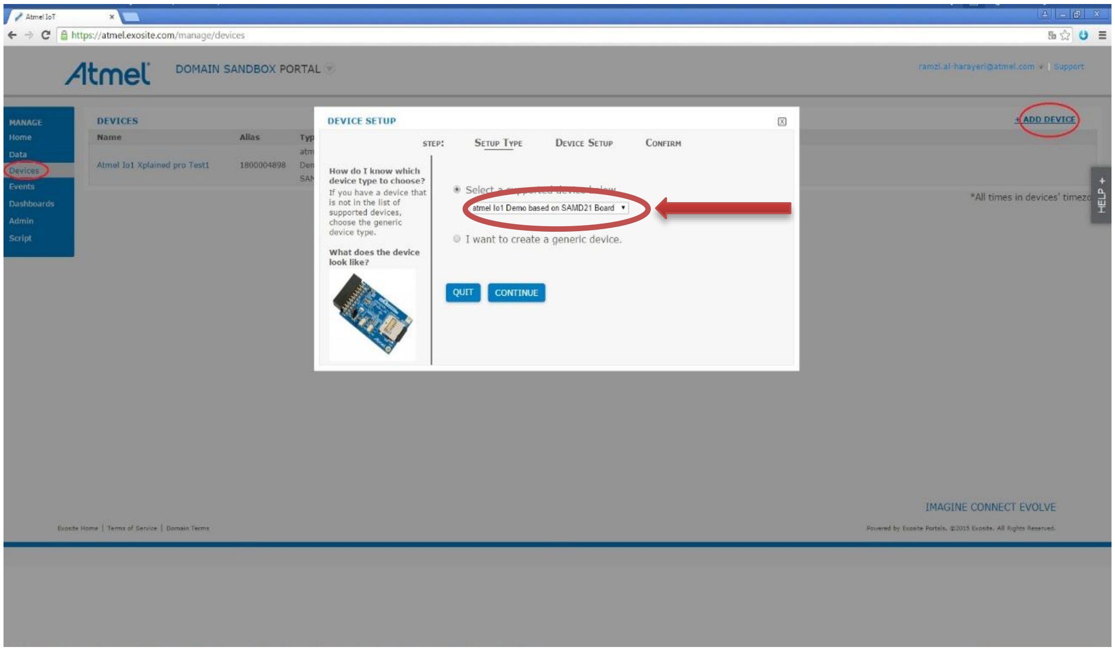
1) Open a free account on Exosite Portal

- Go to Exosite Portal web site http://atmel.exosite.com
- Work on the “Create an Account” and then log in


https://atmel.exosite.com/login
Atmel IoT
Atmel
LOG IN
Email Address:
Password:
LOG-IN
forgot password?
Google
Yahoo
Don't have an account? Sign up now!
Already have an account from an Exosite-powered website?
Add it here
HELP +
IMAGINE CONNECT EVOLVE
Exosite Home | Terms of Service
Powered by Exosite Portals. ©2015 Exosite. All Rights Reserved.

2) Register your device and input your device serial number on I/O1 extension.

- You can see the device setup window as below.
- Select Device -> ADD DEVICE -> Atmel I/O1 Demo based on SAMD21 Board



DEVICE SETUP

STEP: Setup Type | Device Setup | Confirm

How do I know which device type to choose?
If you have a device that is not in the list of supported devices, choose the generic device type.

What does the device look like?

● Select a supported device below
atmel Io1 Demo based on SAMD21 Board

○ I want to create a generic device.

QUIT | CONTINUE

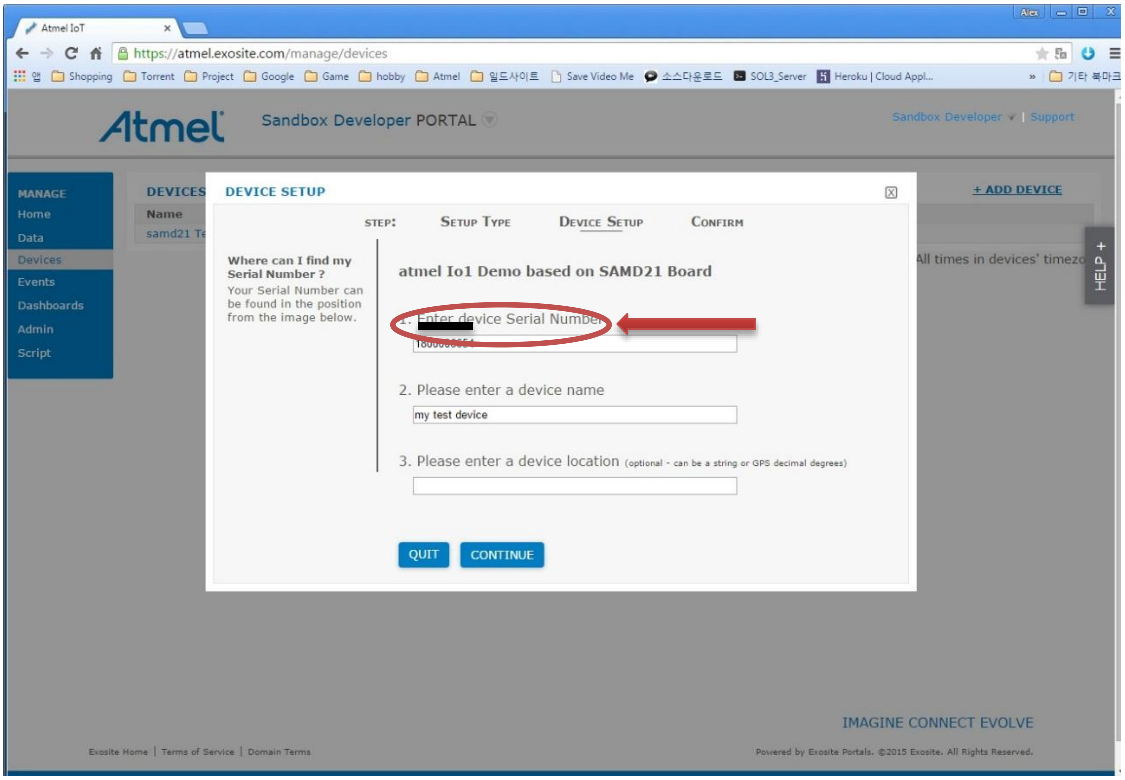
Name	Alias	Typ
Atmel Io1 Xplained pro Test1	1800004898	atm Den SAM

*All times in devices' timezo

Exosite Home | Terms of Service | Domain Terms

Powered by Exosite Portals. ©2015 Exosite. All Rights Reserved.

- Insert the S/N written in the back of I/O1 X Plained pro



DEVICE SETUP

STEP: Setup Type | Device Setup | Confirm

Where can I find my Serial Number ?
Your Serial Number can be found in the position from the image below.

atmel Io1 Demo based on SAMD21 Board

1. Enter device Serial Number
[redacted]

2. Please enter a device name
my test device

3. Please enter a device location (optional - can be a string or GPS decimal degrees)

QUIT | CONTINUE

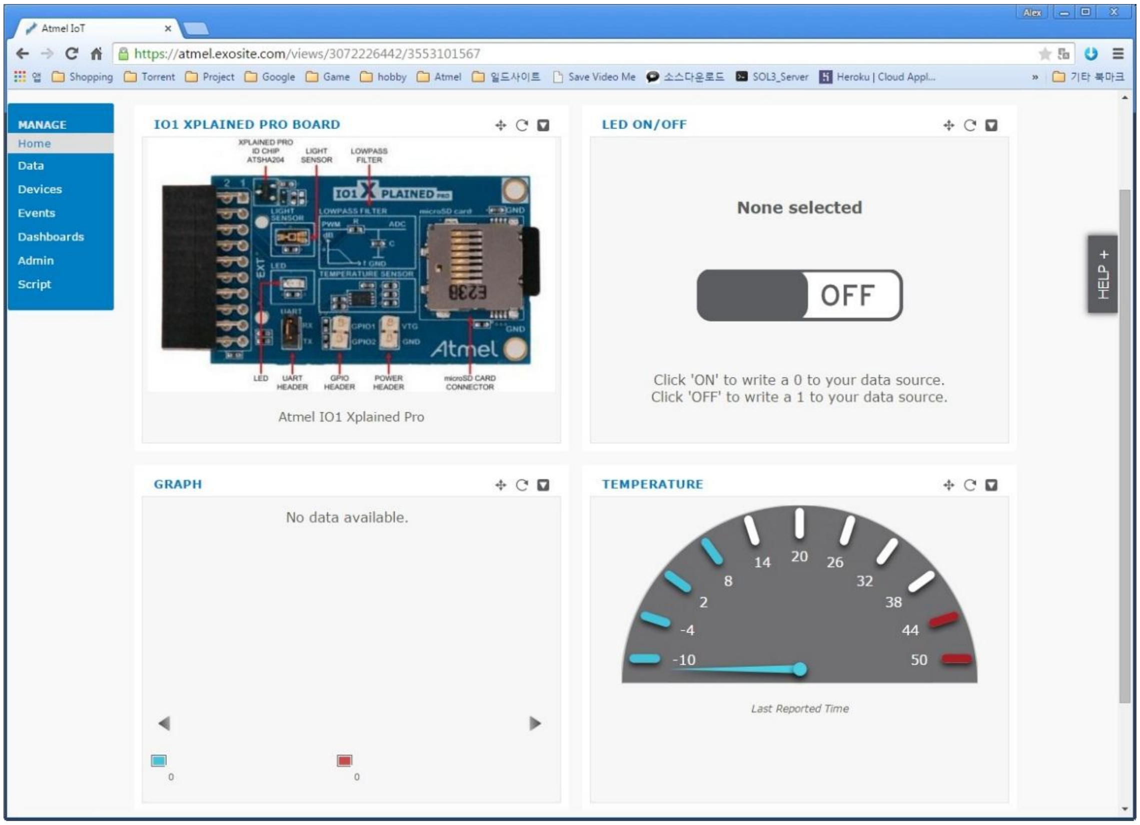
Name
samd21 Te

All times in devices' timezo

Exosite Home | Terms of Service | Domain Terms

Powered by Exosite Portals. ©2015 Exosite. All Rights Reserved.

3) And then the DashBoard of my device will be shown.



None selected

Click 'ON' to write a 0 to your data source.
Click 'OFF' to write a 1 to your data source.

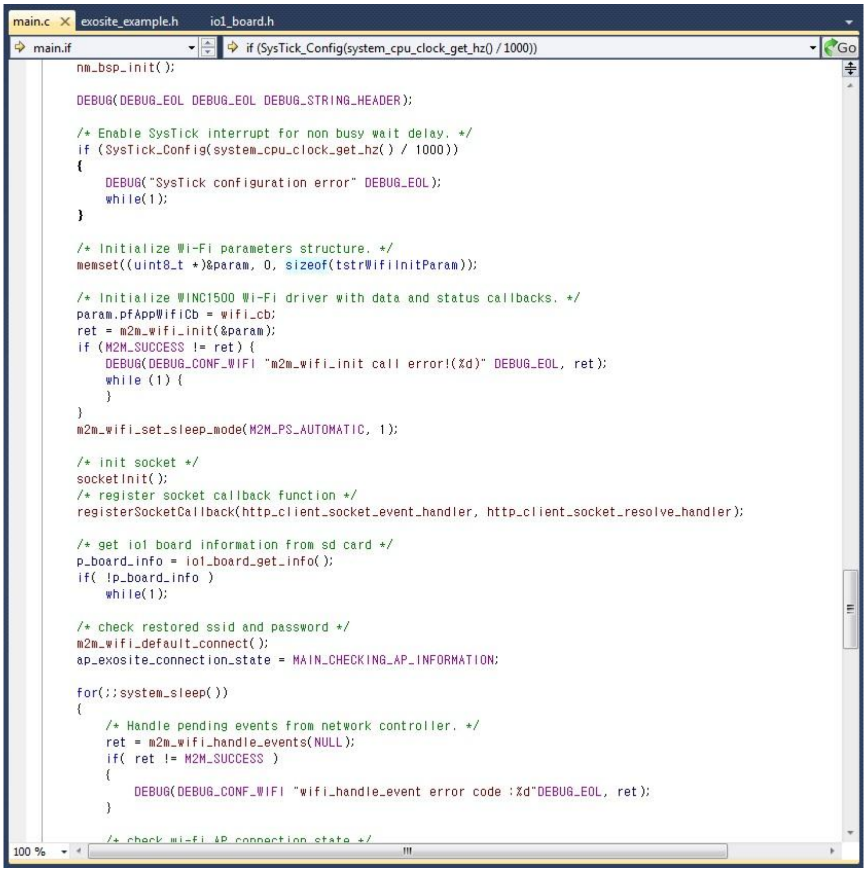
No data available.

4. Programming Guide for SAM W25

For this application, you can start with “*main.c file.*”

The above file has all the codes needed to do the system initialization by controlling the components such as uart, buttons, led’s, BSP, temperature sensor, Wi-Fi driver and socket.

The codes are basically to be seen before the “*for(;;system_sleep())*” sentence as below;



```
nm_bsp_init();

DEBUG(DEBUG_EOL DEBUG_EOL DEBUG_STRING_HEADER);

/* Enable SysTick interrupt for non busy wait delay. */
if (SysTick_Config(system_cpu_clock_get_hz() / 1000))
{
    DEBUG("SysTick configuration error" DEBUG_EOL);
    while(1);
}

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

/* Initialize WINC1500 Wi-Fi driver with data and status callbacks. */
param.pfAppWifiCb = wifi_cb;
ret = m2m_wifi_init(&param);
if (M2M_SUCCESS != ret) {
    DEBUG(DEBUG_CONF_WIFI "m2m_wifi_init call error!(%d)" DEBUG_EOL, ret);
    while (1) {
    }
}
m2m_wifi_set_sleep_mode(M2M_PS_AUTOMATIC, 1);

/* init socket */
socketInit();
/* register socket callback function */
registerSocketCallback(http_client_socket_event_handler, http_client_socket_resolve_handler);

/* get io1 board information from sd card */
p_board_info = io1_board_get_info();
if( !p_board_info )
    while(1);

/* check restored ssid and password */
m2m_wifi_default_connect();
ap_exosite_connection_state = MAIN_CHECKING_AP_INFORMATION;

for(;;system_sleep())
{
    /* Handle pending events from network controller. */
    ret = m2m_wifi_handle_events(NULL);
    if( ret != M2M_SUCCESS )
    {
        DEBUG(DEBUG_CONF_WIFI "wifi_handle_event error code :%d"DEBUG_EOL, ret);
    }

    /* check wi-fi AP connection state */
```

4.1 Initialize Exosite

The first step is to initialize the Exosite client module by calling the function "*Exosite_example_init()*". This function receives one parameter which is a function pointer to call back the http module. The role of this function is to initialize a timer_module for the http_module and also is to initialize the http_module. We need to add the following code just before the "*for(;;system_sleep())*" inside *main()* function

```
/* Exosite init code here  */
ret = Exosite_example_init(main_http_client_callback);
```

The "*main_http_client_callback()*" will be registered by Exosite_example_init() and will receive and process all socket events.

This function can be located in the main.c file or in a separate file as below;

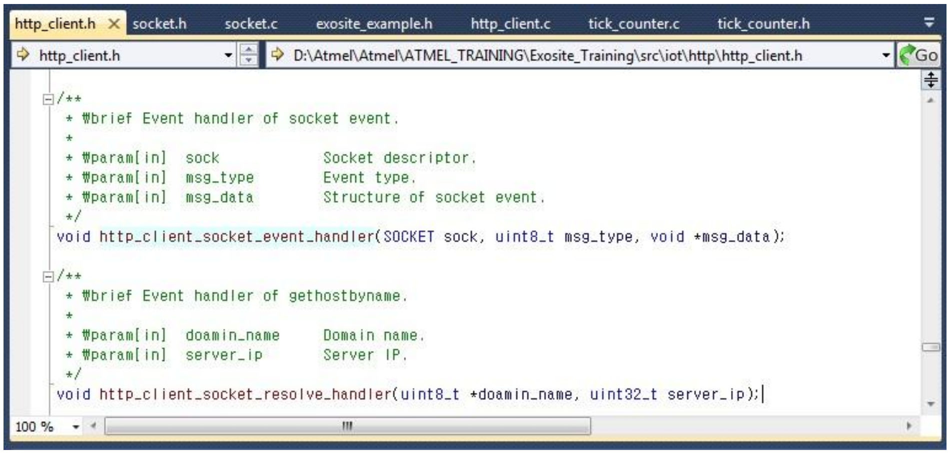
```
/**
 * \brief Callback to get the Data from socket.
 */
static void main_http_client_callback(struct http_client_module *module_list,
                                       int type,
                                       union http_client_data *data)
{
    Switch(type)
    {
      case HTTP_CLIENT_CALLBACK_SOCK_CONNECTED: break
      case HTTP_CLIENT_CALLBACK_REQUESTED: break;
      case HTTP_CLIENT_CALLBACK_RECV_RESPONSE: break;
      case HTTP_CLIENT_CALLBACK_DISCONNECTED: break;
    }
}
```

4.2 Handle socket callback

We registered the callback function for the sockets by calling the following function in *main():*

registerSocketCallback(http_client_socket_event_handler, http_client_socket_resolve_handler);

The above function receives two parameters. The first one is a function pointer to deliver socket messages from the socket. The other one is a function pointer that is used for DNS resolving functionalities. We can find those two parameters in “*http_client.h*” as below;



```
/**
 * ₩brief Event handler of socket event.
 *
 * ₩param[in]  sock           Socket descriptor.
 * ₩param[in]  msg_type       Event type.
 * ₩param[in]  msg_data       Structure of socket event.
 */
void http_client_socket_event_handler(SOCKET sock, uint8_t msg_type, void *msg_data);

/**
 * ₩brief Event handler of gethostbyname.
 *
 * ₩param[in]  doamin_name    Domain name.
 * ₩param[in]  server_ip      Server IP.
 */
void http_client_socket_resolve_handler(uint8_t *doamin_name, uint32_t server_ip);
```

4.3 Send Data to Exosite

In order to send the data from the temperature and light sensors on the I/O1 to the Exosite cloud, the function *“exosite_example_read_and_write()”* should be used as below;

```
/* publish the temp measurements every interval */
if (tick_counter_check_timer())
{
  Char send_buf[100];
  int dTemp = 0;
  int dLight = 0;

  /* prepare to sensor data in the I/o1 Board */
  io1_board_prepare_to_get_info();
  dTemp = io1_board_get_temperature();
  dLight = io1_board_get_lightvalue();
  sprint(send_buf,"degree=%d&voltage=%d", (int)dTemp, (int)dLight);
  if( exosite_example_read_and_write(send_buf, (char*)p_board_info->cik))
  .
}
```

With the above function, the data will be sent to the cloud continuously in regular intervals.

4.4 Receive data from Exosite

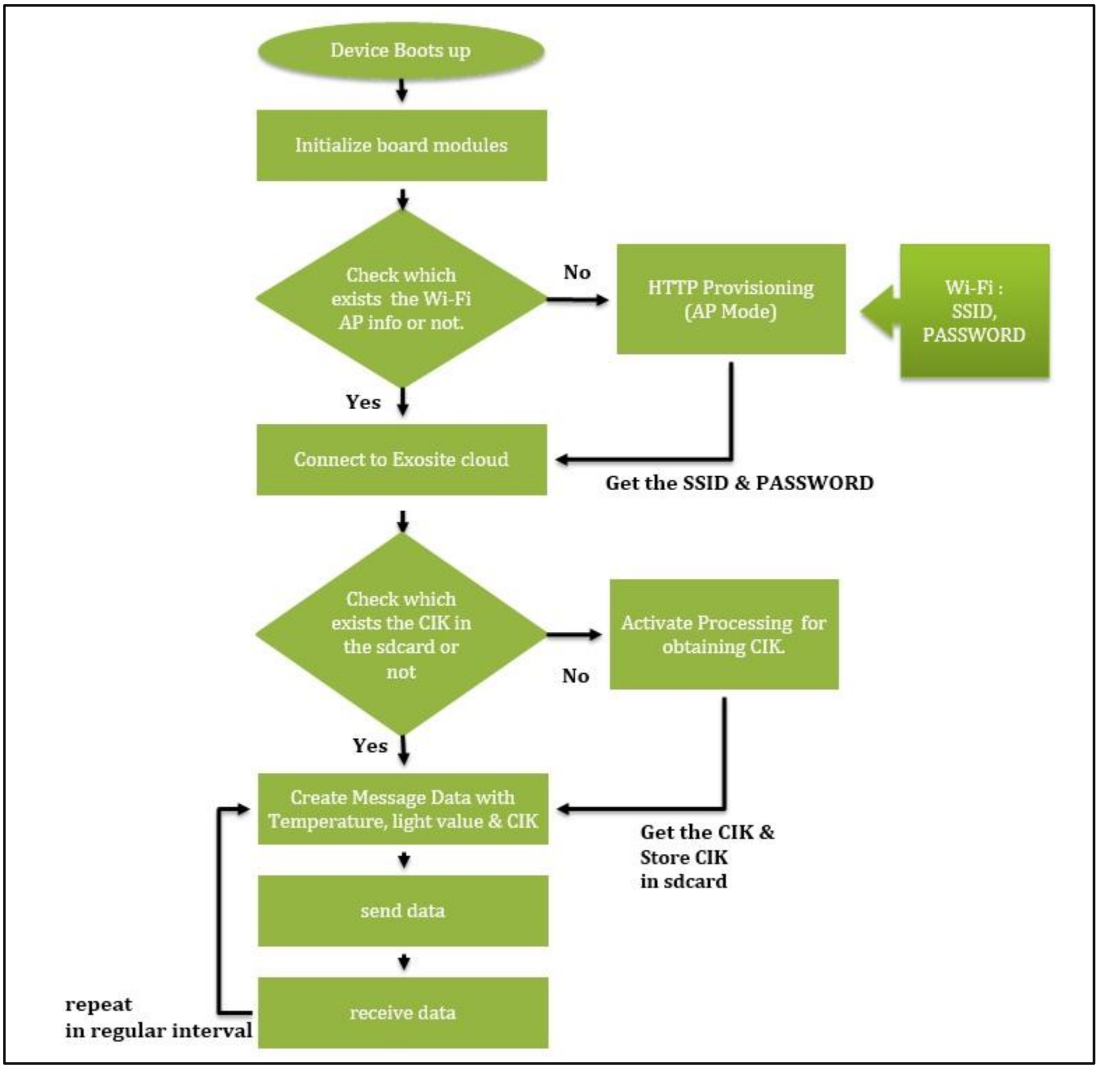
In order to receive the response messages from Exosite, we need to use "*main_http_client_callback()"* function as below;

```
/**
 * \brief Callback to get the Data from socket.
 */
static void main_http_client_callback(struct http_client_module *module_list,
                                      int type,
                                      union http_client_data *data)
{
      case HTTP_CLIENT_CALLBACK_RECV_RESPONSE:
        parsing_http_response_data( data->recv_response.response_code,
                                    data->recv_response.content,
                                    data->recv_response.content_length);
       break;
}
```

Then you can add the different codes to process the received messages without the *"parsing_http_response_data()"* function.

INFO This function is needed to receive the CIK from Exosite.

4.5 Flow Chart


Device Boots up
Initialize board modules
Check which exists the Wi-Fi AP info or not.
No
HTTP Provisioning (AP Mode)
Wi-Fi : SSID, PASSWORD
Yes
Connect to Exosite cloud
Get the SSID & PASSWORD
Check which exists the CIK in the sdcard or not
Activate Processing for obtaining CIK.
No
Yes
Create Message Data with Temperature, light value & CIK
Get the CIK & Store CIK in sdcard
send data
repeat in regular interval
receive data

‘

5. Conclusion

This document demonstrated the how to use the SAM W25 Wi-Fi module for Exosite cloud and how to connect to the cloud.

The following topics have been covered:

- SAM W25 Wi-Fi module presentation
- APIs of Exosite.
- Programming the SAM W25 with Exosite_Demo project
- Viewing dashboard in the Exosite web site
- Exchanging the data between SAM W25 and Exosite

Revision History

Doc. Rev.	Date	Comments
XXXXXA	05/2015	Initial document release

Enabling Unlimited Possibilities®

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

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.