SMART ARM-based Microcontroller

AT15347: SAM L22 Thermostat IoT Node

APPLICATION NOTE

Introduction

This application note explains the thermostat application (an IoT node) based on Atmel® | SMART SAM L22 and demonstrates the salient features of SAM L22 device in an embedded cloud design for the Internet of Things using Wi-Fi® connectivity.

This application demonstrates the temperature sensing capability, cloud connectivity using CryptoAuthentication™ security, and touch segment LCD for user interface.

This demonstration uses SAM L22 Xplained Pro Evaluation Kit, ATWINC1500 Xplained Pro Kit, and the I/O1 Xplained Pro Kit.

The Touch SLCD1 Xplained Pro has five mutual capacitive QTouch® buttons integrated on the segment LCD. It is included in the SAM L22 Xplained Pro evaluation kit.

Features

- Application parameters accessible and controllable from the Proximetry cloud
- Wi-Fi connectivity through the ATWINC1500 Xplained Pro extension board
- Implemented with Tickless FreeRTOS (version 8.0.1)
- Node authentication by onboard ATECC508A
- Maintain timestamp using internal RTC
- Segment LCD displays the ambient temperature and the desired temperature setpoint
- QTouch buttons for navigating the LCD display
- Automatic switching to external super capacitor for RTC backup domain

Table of Contents

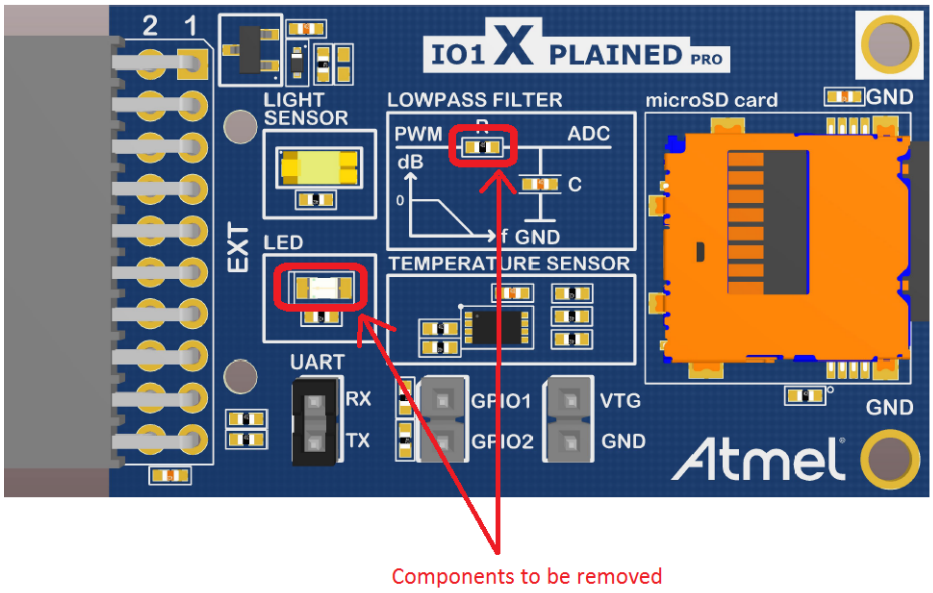
1. Prerequisites

- **Hardware**
  - Atmel | SMART SAM L22 Xplained Pro (ATSAML22-XPRO-B)
  - Atmel TSLCD1 Xplained Pro extension board
  - Atmel ATWINC1500 Xplained Pro extension board
  - I/O1 Xplained Pro extension board
  - One Micro USB type-B cable
- **Software**
  - Atmel Studio 7 (Version: 7.0.634 or later)
  - Atmel Software Framework - 3.29 or later
- **General**
  - Wi-Fi Hotspot (with internet connectivity)
  - A registered user account in Proximetry cloud space

Note:

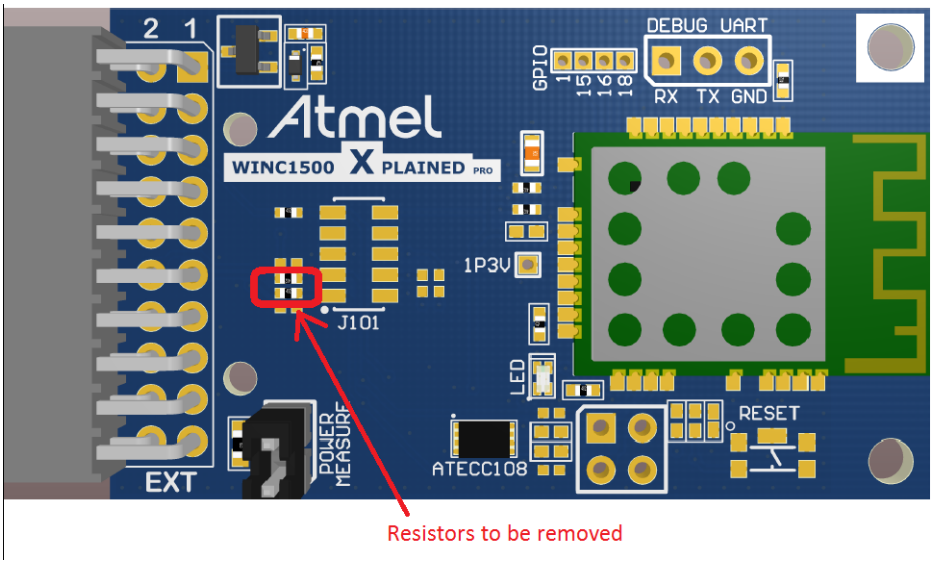
1. A few hardware conflicts may exist between I/Os used on I/O1 Xplained Pro and the PTC lines connected in TSLCD1 Xplained Pro. As a workaround, remove the resistor and the LED highlighted in the following picture on the I/O1 Xplained Pro. The components must be replaced for future use of the I/O1 Xplained Pro.

Figure 1-1. I/O1 Xplained Pro Modification



2. The ATWINC1500 Xplained Pro board has an ATECC108 device mounted on it which shares the same I2C lines and address as ATECC508A. To avoid the conflicts between the devices, remove the zero ohm resistor shown in the image. This disconnects ATECC108 chip from the ATWINC1500 Xplained Pro.

Figure 1-2. ATWINC1500 Xplained Pro Modification



3. Add-on board is used in this demo to drive the fan. Refer section Demo Hardware Set-up and Appendix for more details about this add-on board. Jumper position in this add-on board should be connected as follows:

Jumper Options	Description
2 - 1	If external power supply is used
2 - 3	If power should be supplied from SAM L22 Xplained Pro - Default position

4. VBAT SELECT jumper on SAM L22 Xplained Pro should be connected as follows:

Jumper Options	Description
PB03	CAP - If backup is needed for RTC - Default position
PB03	VCC - If backup is not needed for RTC

5. The ATECC508A chip on the SAM L22 Xplained Pro kit must be provisioned before using it in the actual project. Information about provisioning ATECC508A is available in Security and Authentication section.

2. IoT Node

Internet of things has become a buzzword in electronics industry. Atmel powers the IoT by focusing on edge nodes, a category that includes everything from key fobs, fitness bands, and home appliances to industrial equipment and infrastructures for smart cities.

Atmel powers the edge nodes and end devices that form the link between individual devices and the gateways that connect to the cloud. Atmel helps makers and designers with all the basic building blocks, from embedded processing solutions, connectivity to sensors, security, software, and tie them all together with a rich ecosystem of design tools and development partners.

This thermostat demo features sensing, embedded processing, internet connectivity, and security aspects which altogether makes it an IoT node.

2.1. Cloud Connectivity (From Sensor to Server)

Atmel has partnered with market-leading cloud providers to offer end-to-end solutions for IoT designers using Atmel wireless connectivity products. Each of the partner offer a distinct solution with unique features and support various Atmel devices. This allows developers to easily design the perfect solution for their Internet of Things (IoT) requirements.

The cloud solution and Atmel Wi-Fi solution can easily be combined with broad portfolio of microcontroller technology to form an IoT solution. The Thermostat application has a sensor to server connectivity which uses the following building blocks:

- Temperature sensor (AT30TSE75x on I/O1 Xplained Pro)
- Embedded processing (ATSAML22N18A)
- Security (Node autentication using ATECC508A)
- Wireless Connectivity (ATWINC1500 Xplained Pro)
- Cloud Server (Atmel-Cloud service from Proximetry)

Connectivity Solution

Wi-Fi is a key technology that enables devices to connect each other, to a WAN, or simply to the Internet in order to provide remote monitoring and to control a system.

Cloud Partner

Proximetry offers cloud services with features such as remote device and data management solutions for IoT solution vendors. Several of Atmel's products can be integrated with the Proximetry cloud. The cloud service is also scalable to very large deployments and traffic loads. To access the device from Proximetry cloud, the user must register with Proximetry.

Following steps explains registration procedure.

1. A user must register and create an account on the Proximetry website (https://atmelcloud.proximetry.com/Login).
2. After registering, an activation code will be sent from Proximetry to user's registered e-mail. This activation code should be used in firmware while connecting to the cloud.
3. Every device should have an activation code associated with it to connect to the cloud.
4. By default, user corresponding to device's activation code can edit the control parameters from the cloud. If required, this editing permission can be extend to all.
5. Registered user who has a Proximetry account can view devices connected to cloud. However, to edit the control parameters, the user must have access permissions.

Each cloud partner have their specific methods to establish the connection and communicate with the cloud. An association with the respective cloud partner is required to communicate with that particular cloud space. In general, this association can be a library or source files or example project from respective cloud partner which enables that user to communicate with that cloud space. For more information about Proximetry cloud space, refer http://www.proximetry.io. Atmel has also partnered with the other cloud providers which is listed in the webpage http://www.atmel.com/products/wireless/cloud-partners/.

2.2. Security and Authentication

Networks and devices that lack authentication can be tricked into giving up access credentials or executing malicious code. The end node devices are most vulnerable points in the IoT, especially in cost-sensitive markets where omitting security might be seen as a viable way to cut costs. Device spoofing and “man-in-the-middle” (MITM) attacks are designed to gain access to a network using software that mimics an IoT device. Even networks with weak authentication are subject to MITM attacks in which the attacker eavesdrops on transmissions and steals access credentials. Authentication using Atmel CryptoAuthentication devices ensures a secure IoT end node. The new ATECC508A CryptoAuthentication IC employs ultra-secure hardware-based cryptographic key storage and cryptographic countermeasures which are more secure than software-based key storage.

The ATECC508A includes an EEPROM array that can be used to store up to 16 keys, certificates, miscellaneous read/write, read-only or secret data, consumption logging, and security configurations. Access to the various sections of memory can be restricted in a variety of ways and then the configuration can be locked.

The ATECC508A implements a complete asymmetric (public/private) key cryptographic signature solution based upon Elliptic Curve Cryptography and the ECDSA signature protocol. The device features hardware acceleration for the NIST standard P256 prime curve and supports the complete key life cycle from high quality private key generation, to ECDSA signature generation, ECDH key agreement, and ECDSA public key signature verification. The device is designed to securely store multiple private keys along with their associated public keys and certificates. The signature verification command can use any stored or an external ECC public key. Public keys stored within the device can be configured to require validation via a certificate chain to speed-up subsequent device authentications.

In this Thermostat Demo application, the ATECC508A device is used for mutual authentication of node and cloud.

The steps used by a cloud server (the host in this case) to authenticate a node are as follows:

1. End Node (ATECC508A) builds certificates
2. End node sends device certificate to cloud
3. End node sends signer certificate to cloud
4. Cloud server verifies end node certificate chain
5. Cloud server sends random challenge to end node
6. End Node signs random challenge using its private key securely contained in the ATECC508A. The ATECC508 signs the random challenge internally and returns the signature. The Private Key is not exposed.
7. Cloud server verifies whether the signature is valid

After cloud verifies end node, the end node will verify the authenticity of the cloud server and steps for latter is as follows:

1. Cloud server rebuilds certificates

2. End node receives host device certificate
3. End node receives host signer certificate
4. End node verifies host certificate chain
5. End node send random challenge to host
6. Cloud server signs random challenge using its private key
7. End node verifies random challenge
8. If verification completes successfully, the connection to the cloud is allowed

Note: The ATECC508A on the SAM L22 Xplained Pro must be provisioned before it can be used for Authentication. The steps for provisioning are explained in the application note *ATECC508A Node Authentication Example Using Asymmetric PKI* which is listed in the References section. The hex file which can be used for provisioning is available in the zip file associated with this application note.

3. Demo Hardware Set-up

Figure 3-1. Thermostat Demo Block Diagram

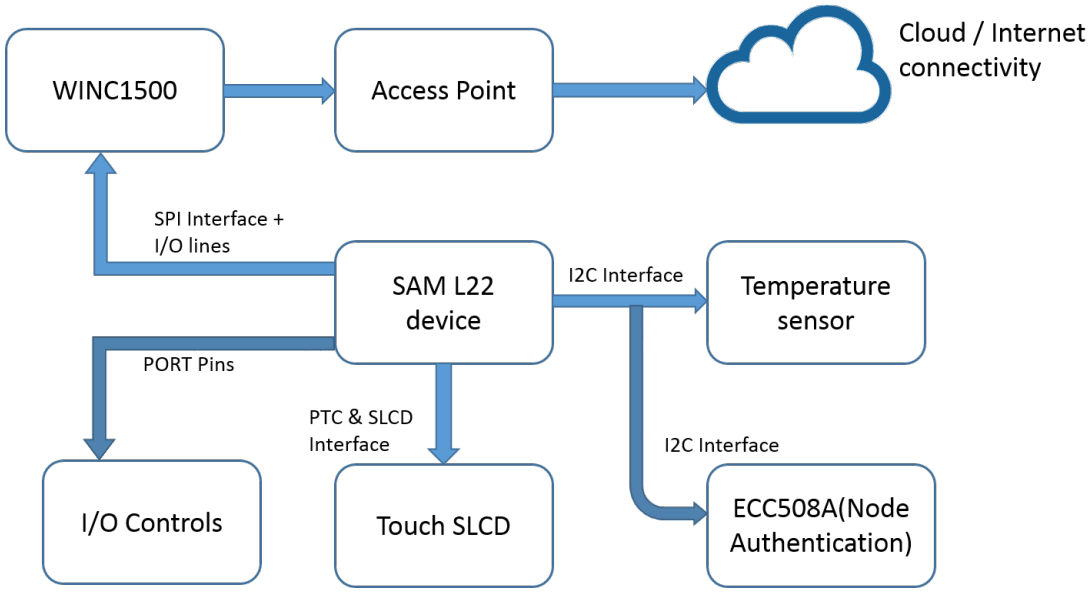
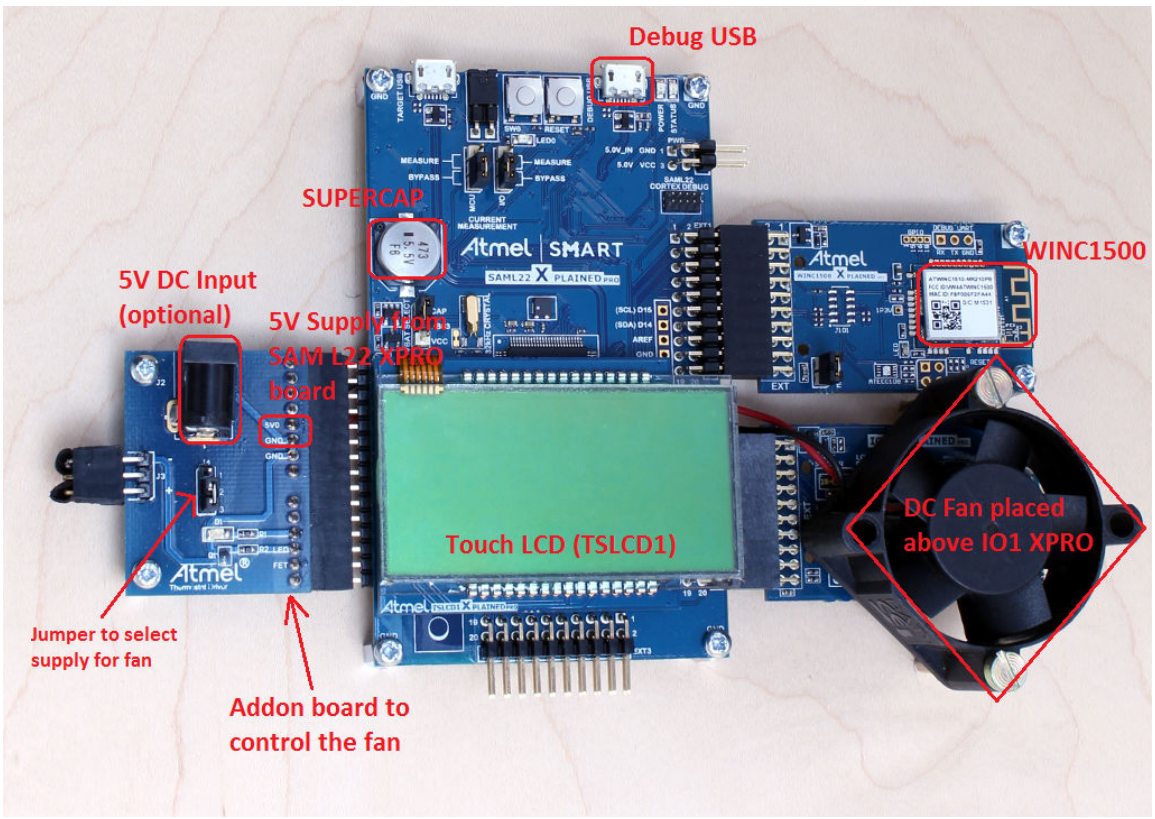


Figure 3-2. Demo Unit



4. Demonstrated Functionalities

The following features of the SAM L22 are explained and demonstrated in this application note.

- Segment LCD Module
- Peripheral Touch Controller (PTC)
- SERCOM SPI Module (ATWINC1500)
- SERCOM I2C Module (Temperature Sensor and CryptoAuthentication chip)
- RTC
- Automatic power switching
- PORT (I/O controls)
- TC (PTC Time base and RTOS Time base)

4.1. Cloud Functionality

The cloud connectivity is an essential feature of this application. The Proximetry cloud has a default dashboard for every device connected to it. This dashboard can be customized based on the application requirements such as number of control parameters and statistics information. The dashboard specific to this demo will show the graph of room temperature which is updated once every 5 sec (AirSync Interval) in the cloud. The thermostat device should send synchronization message to the cloud at regular interval to intimate to the cloud that device is active. This time interval is called Periodic Airsync Reporting Interval (AirSync Interval) and this can be configured in the firmware.

Apart from displaying the room temperature and setting target temperature, automatic mode is another configurable parameter. If automatic mode is turned off, then the user can control the fan and LEDs available on the demo kit manually. Otherwise, SAM L22 firmware controls the fan according to the pre-set and actual room temperature values.

4.2. Wi-Fi Functionality

To connect a Wi-Fi access point, thermostat requires the network credentials. There are two methods to connect a Wi-Fi access point. The first method is to include the credentials in the firmware. This is the default method used for network connection. The other method is to use a provisioning mode. If the default network is not available, provisioning mode will be activated. This generates a Wi-Fi hotspot from the ATWINC1500. To initiate a connection, connect phone/tablet/laptop to the Wi-Fi hotspot offered by the thermostat device. When a laptop/phone is connected to the thermostat access point, the configuration page can be accessed at www.atmelconfig.com using a web browser.

To switch the network, web provisioning can be used. The module can be configured in an access point mode by pressing SW0 button on the SAM L22 Xplained Pro for more than 3 seconds. The network credentials can be supplied to the thermostat using these methods.

4.3. LCD and Touch Buttons

Using AT30TSE758 available on the I/O1 Xplained Pro, SAM L22 reads the ambient temperature and displays it on the segment LCD. At the same time, the SET temperature is also displayed on the segment LCD. This can be controlled using the QTouch buttons, indicated by an up and down arrow.

4.4. Timestamp Using RTC

The RTC in SAM L22 device maintains the time and date. When powered-on first time, the time and date will display a preconfigured time and date. The values can be changed by tapping the QTouch button, 'enter' on the segment LCD. Using the arrows allow the user to set the time and date. Confirm by tapping 'enter' one more time. The time and date will be retained only during power failure. If the reset button is pushed, the time and date will be lost and the user must reconfigure.

The RTC date and time will be retained when power fails. Whenever hardware reset (reset push button) is issued, the code is reset to initial condition. If power is lost for a short time and board is powered again, RTC can retain the time in the clock, provided a hardware reset is not issued to device.

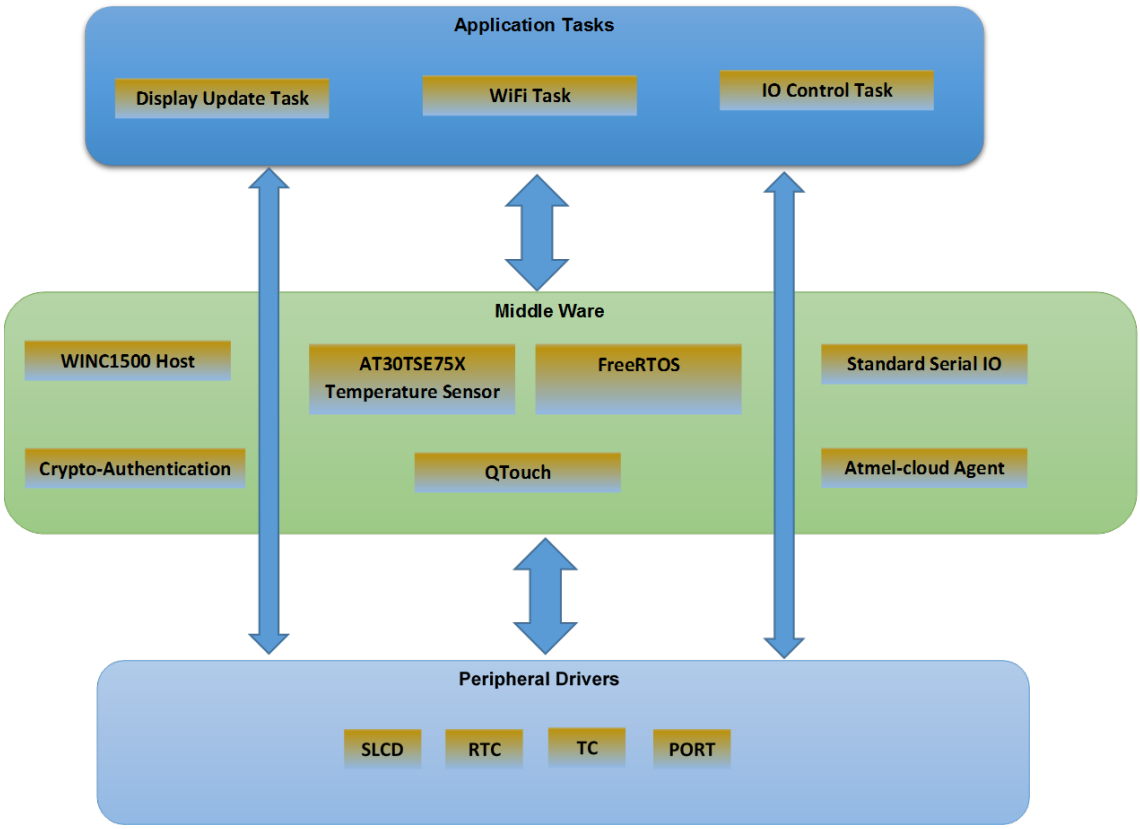
4.5. Battery Backup Option

The super capacitor mounted on the SAM L22 Xplained Pro allows the system to retain the time and date for approximately 24 hours in the backup domain. If configured, the Supply controller automatically switches to backup power domain during a power failure. If a larger super capacitor or battery is connected to the external battery supply pin, the backup time can be extended beyond 24 hours.

5. Firmware Implementation

The following diagram illustrates the firmware architecture.

Figure 5-1. Firmware Architecture



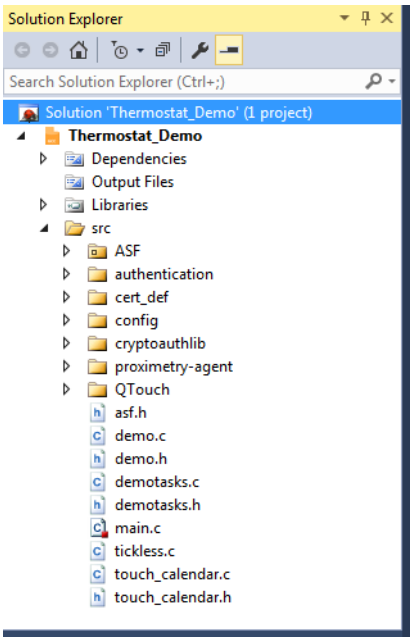
Tickless FreeRTOS: To reduce the power consumption, it is a common practice to place the MCU in low power mode (sleep mode) when running in idle task. RTOS will enter and exit the low power mode based on the available idle time. In this Tickless FreeRTOS idle task, RTOS tick is suppressed and device is placed in deeper sleep mode and wakes up after the configured time. Timer Counter with higher period (relatively larger period than RTOS tick period) is configured to wake the device after entering the sleep mode in idle task. For more information about Tickless FreeRTOS, refer http://www.freertos.org/low-power-tickless-rtos.html

QTouch® Library: Atmel QTouch Library makes it simple for developers to embed capacitive touch button, slider, wheel functionality into general purpose Atmel SMART | ARM® and AVR® microcontroller applications. The royalty- free QTouch Library provides several library files for each device and supports different numbers of touch channels, enabling both flexibility and efficiency in touch applications. For more information about QTouch, refer http://www.atmel.com/products/touchsolutions/bsw/default.aspx

5.1. Project Contents

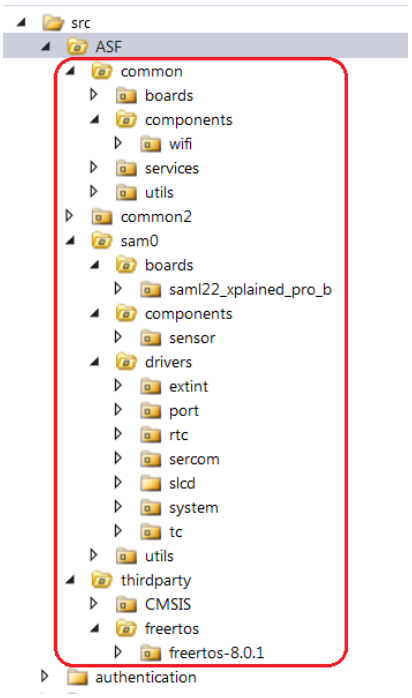
The Atmel Studio project corresponding to this thermostat demo project is available as a zip file along with this application note. The project consists of modules and application files as shown in the following image.

Figure 5-2. Files and Folder Structure of the Project



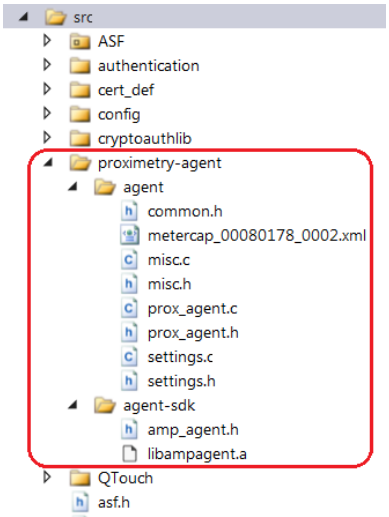
1. Application source files
   - `main.c`
   - `tickless.c` - Used for tickless FreeRTOS
   - `demo.c` - Wireless functionality
   - `demotasks.c` - Tickless FreeRTOS tasks
   - `touch_calendar.c` - RTC and touch functionality
2. Atmel Software Framework (ASF) module contains drivers and services for device peripherals, middleware and components used in the project. FreeRTOS kernel and port files, ATWINC1500 driver, Temperature sensor driver, SERCOM drivers are a few other important source files.

Figure 5-3. ASF Files



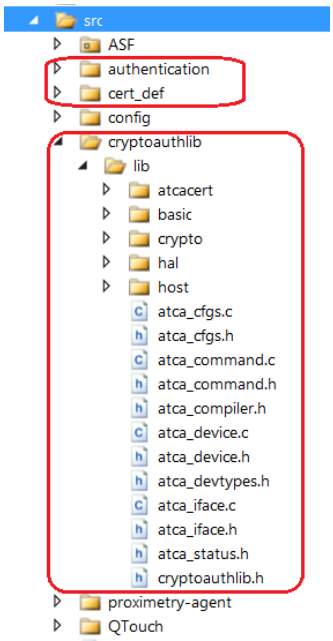
3. Proximetry-agent module: Contains library and source files for Proximetry cloud interface.

Figure 5-4. Proximetry- Agent Library and Files



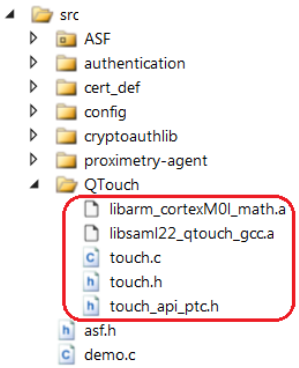
4. Cryptoauthentication module: Contains source files for security chip (ECC508) interface.

Figure 5-5. Authentication Module Files



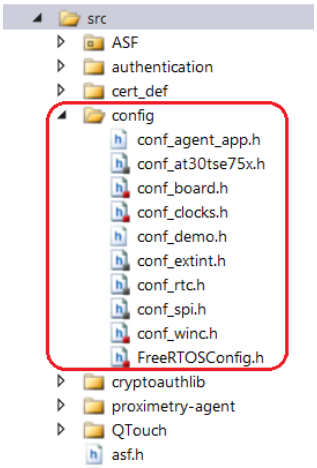
5. QTouch module: Contains library and source file for capacitive touch interface in TSLCD.

Figure 5-6. QTouch Files



6. Configuration header files
   The config folder has all configuration files which contains #defines related to this demo. Information about some key configuration files and #defines in them are as follows:

Figure 5-7. Configuration Files



6.1. **`conf_demo.h`**

This file contains macros for Enable authentication, Tick Timer, Touch Timer, IO control pins and Wi-Fi Access point details. Enable authentication macro is used to enable/disable authentication process. If this macro is disabled, then firmware will skip the authentication process and connect to the cloud.

6.2. **`conf_agent_app.h`**

This file contains macros for device model no., device name, default SSID / Passphrase (of Wi-Fi network to which device has to connect), web provisioning enable, default server IP, server port number, AirSync Interval and default activation code

Activation code sent by proximetry should be defined for the macro *DEF_ACTIVATION_CODE*. Unsecure and secure cloud communication uses different server port. If authentication is enabled, then server port 5454 is used else port 5050 is used.

In some cases, Wi-Fi routers may block the UDP transfers to these ports (due to network and security settings). In this case, debug console may look fine with proper messages but still device will not be connected to cloud. In this case, using access points which does not block these transfers is recommended. Eg. mobile phone as hotspot with mobile data enabled could be a useful workaround.

6.3. **`conf_winc.h`**

This file contains configuration macros for SERCOM and ATWINC1500

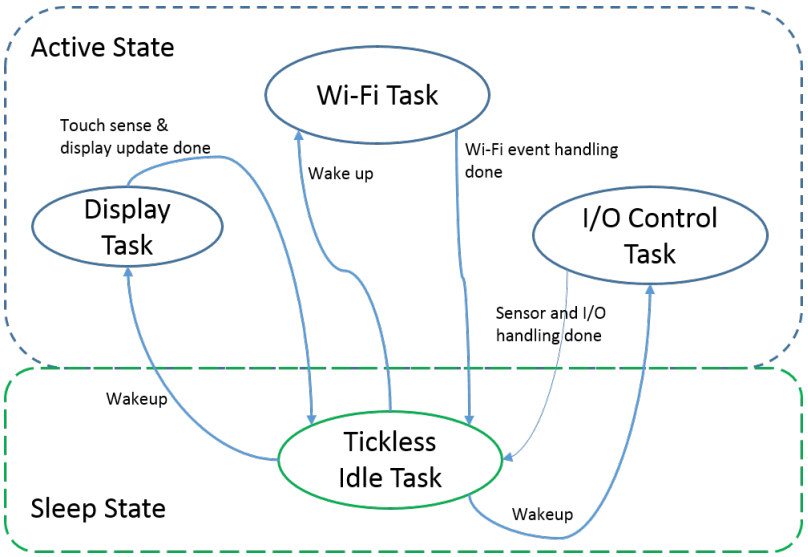
6.4. **`FreeRTOSConfig.h`**

This file contains macros for all FreeRTOS related configurations such as enabling Tickless mode, minimal task stack size, CPU clock, Task priority. Refer FreeRTOS getting started application note in reference section for more details on FreeRTOS.

Apart from these above mentioned configuration files, basic configuration files such as clock, board and interface related files are also present in configuration folder in the project.

5.2. Application Tasks

Figure 5-8. Application Tasks

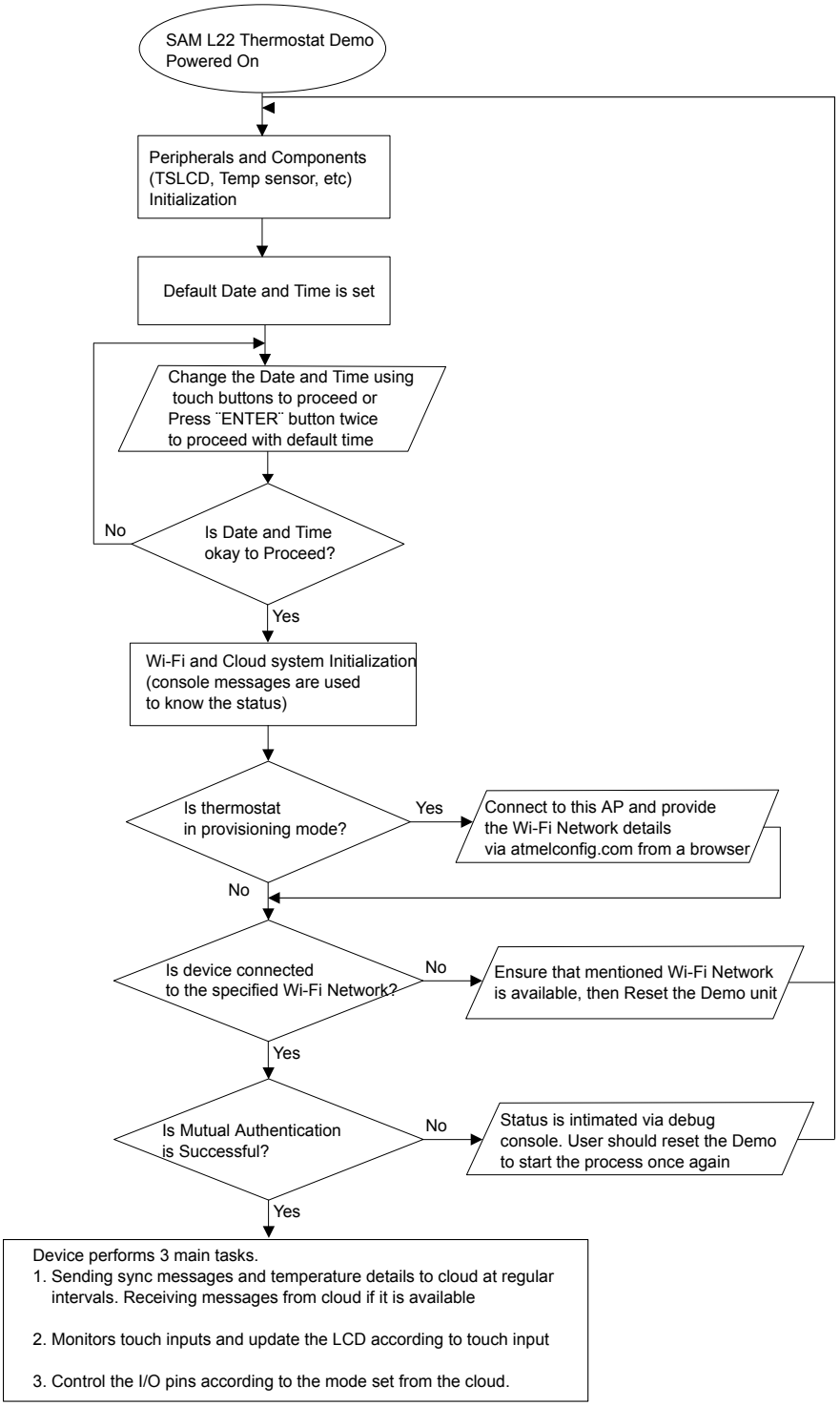


1. **Wi-Fi Task:**
   Wi-Fi task handles the Wi-Fi events and also communicate with Proximetry-cloud agent. If web provisioning is enabled, this task also monitors the entry condition and put the device in provisioning mode if that conditon is met. In this case, entry condition is button press for predefined time. The SW0 button on SAM L22 Xplained pro is used for this purpose.
2. **Display Task:**
   Display task set the default display on LCD for first time and waits till the calendar is configured. After it is configured, this task updates the room temperature on LCD display and monitors user touch inputs to update LCD display for valid touch inputs on integrated Qtouch buttons. If there is a valid touch, this task will update the LCD display and control parameters accordingly. Valid touch can be either to increase / decrease set temperature or to calendar mode. If there is a change in set temperature, this task will update the LCD display and control parameter variable with a new value. This new value will be communicated to cloud in IoCtrlTask.
3. **I/O Control Task:**
   This task reads the room temperature and controls I/O pins used to turn on/off fan, heater and cooler. In the demo unit, a DC fan is used to demonstrate real application scenario. So I/O pin associated with fan will control a DC fan. For I/O pins associated with heater and cooler, LEDs present in SAM L22 Xplained Pro and add-on board are used for demonstration purpose. There are two modes available in this thermostat application, automatic mode and manual mode. Mode selection can be done from Proximetry cloud once device is connected successfully with cloud. By default, the application starts in automatic mode. In automatic mode, I/Os (For fan, heater and cooler) will be controlled by firmware based on the set and room temperature. If application is in manual mode, then I/Os are controlled manually from the Proximetry cloud.

5.3. Demo Execution Sequence / Flow

The demo execution flow is illustrated in the following flow chart.

Figure 5-9. Steps to Execute the Demo



5.4. Functions Used in the Firmware

This section describes function calls and definitons used in each application file. This provides an overview of functions used in the application.

1. **main.c** file

Function	Description
system_init();	For clock and board initialization. This is standard system initialization function from ASF template.
configure_console();	To configure debug console via EDBG SERCOM Module. Defined in the main.c file itself.
atcab_init(&cfg_ateccx08a_i2c_default);	ATECC508 device initialization routine
at30tse_init();	Temperature sensor initialization routine
slcd_init();	Segment LCD initialization routine
configure_tc();	Timer configurations for touch acquisitions
touch_sensors_init();	Touch sensors initialization (PTC Module initialization)
rtc_init();	RTC Module initialization
demo_init();	Wireless and Cloud initialization happens in this function. This is defined in demo.c file.
demotasks_init();;	Creating FreeRTOS tasks to run the application. Function is defined in demotasks.c file and only function call is made from main.c file.
vTaskStartScheduler();	FreeRTOS API (FreeRTOS scheduler taking the control to run the tasks)

2. **demotasks.c** file

Function	Description
demotasks_init()	Creating different FreeRTOS tasks to run the application. This function is called from main.c before calling the RTOS Scheduler.
vDisplayTask(void *params)	Updates the LCD display based on user inputs and parameter changes.
vWiFiTask(void *params)	Responds to Wi-Fi events and handle communication with Atmel-cloud agent.
vIoCtrlTask(void *params)	Reads the room temperature and controls IO pins used for indicating Fan/Heater/ Cooler activation.

3. **touch_calendar.c** file

Function	Description
rtc_date_time_set(void)	Function is used to set date and time. Date and time is set with the help of touch buttons on LCD.
set_default_display(void)	Set the default display for the first time. Then another function will be updating the LCD on valid touch inputs

Function	Description
`thermostat_io_update(void)`	This function controls the IO pin based on the room temperature and update the status of IO pin to cloud.
`touch_sense_update_display(void)`	Check for valid user touch and update the display accordingly.

4. **`demo.c`** file

Function	Description
`demo_init(void)`	Initialize hardware for ATWINC1500, perform Wi-Fi module level initialization, sockets and it's callback initilaization, connect to predefined Wi-Fi network and perform cloud initialization (using Proximetry cloud library).
`m2m_wifi_socket_handler(SOCKET sock, uint8_t u8Msg, void *pvMsg)`	Callback function for the socket event.
`m2m_wifi_state(uint8_t u8MsgType, void *pvMsg)`	State machine implementation function for Wi-Fi functionality.
`command_process_cloud(uint8_t cmd, uint8_t dev_type)`	This function has the complete state machine implemented for mutual authentication. This will be called to initiate authentication process.
`setup_prox_agent_socket()`	Function setup the proximetry agent UDP sockets.

5. **`tickless.c`** file

Function	Description
`vPortSetupTimerInterrupt(void)`	This function initialize and start the timer for FreeRTOS tick.
`vPortSuppressTicksAndSleep(TickType_t xExpectedIdleTime)`	Function to configure timer for sleep and calculate time slept.

5.5. Low Power Design Considerations

SAM L microcontroller series are ultra-low-power devices that are specifically designed for low power applications and battery powered products. The Low power features used in this thermostat applications are as follows:

- **Tickless FreeRTOS**
  This application is based on FreeRTOS and hence tickless FreeRTOS is used for low power consumption. Device will operate in Standby mode in Idle task.
- **Low Power Mode of ATWINC1500**
  The ATWINC1500 module can be put into sleep mode independently and this feature is used in the application. Whenever Wi-Fi communication is not required in application, ATWINC1500 is put into sleep mode and wakes before the Wi-Fi communication begins.
- **Efficient Regulator Mode**

Internal regulator has two modes - LDO and BUCK. By selecting appropriate regulator mode, power consumption can be optimized. Switching (BUCK) mode is power efficient and is utilized in this application.

- **SLCD Peripheral**
  Low power waveform mode is a built-in feature available in SLCD module and this is useful in low power applications. This feature can be used in application if LCD has Type-B waveform support. Since TSLCD1 Xplained Pro supports this feature, it is used in this application.

6. Steps to Build and Run the Demo

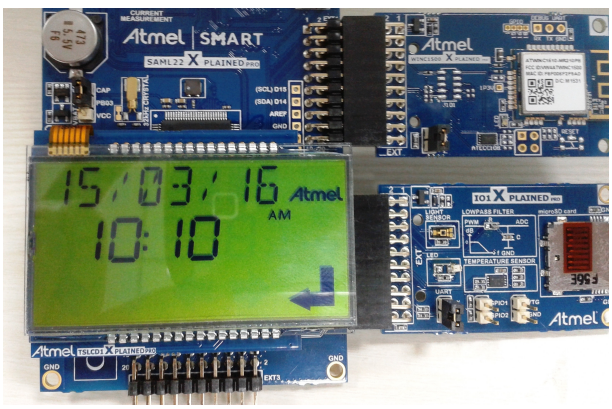
The steps to build and run the demo on SAM L22 Xplained Pro kit are as follows.

1. The Hardware set-up should be connected in such a way that Xplained Pro extension boards are connected in SAM L22 Xplained Pro as follows:
   - TSLCD Xplained Pro in EXT5
   - I/O1 Xplained Pro in EXT2
   - ATWINC1500 in EXT1

   Add-on board drives the 5V fan using I/O pin from MCU. This board is not mandatory for executing the demo application. Refer to section Demo Hardware Set-up and Appendix.
2. When the set-up is ready, connect the Debug USB port to PC. Wait until the EDBG enumeration completes successfully.
3. Open a serial session for EDBG Virtual COM with any serial communication utility with following settings, 115200 baud, 8 data bits, 1 stop bit, no parity and no flow control.
4. Launch the Atmel Studio. Navigate to > **File** > **Open** > browse to the location of demo project > Select the solution or project file.
5. In the Project properties > **Tools** tab > Select this EDBG in the Tool settings.
6. Before compiling the project, set the #defines for **SSID** and **password** of target Wi-Fi network and **cloud activation code** in `conf_agent_app.h` file.
7. Compile the Project and download the code to the SAM L22 Device.
8. Check for the user interface messages in debug console. Ensure that predefined access point (with internet connectivity) is active to have successful wireless connectivity.
9. Following images shows default screen on the SLCD and steps to complete the date and time setting. The Wi-Fi communication will be started only if date and time is set successfully. If needed, this step can be skipped which means application starts with default date and time.
   The default calendar setting displayed on the SLCD is as follows.
   - Date is displayed in DD/MM/YY format.
   - Time is displayed in 24hour format (AM /PM indication is also available).
   - "Atmel" Segement is turned ON.
   - "ENTER" QTouch button is active and the respective segment is visible.

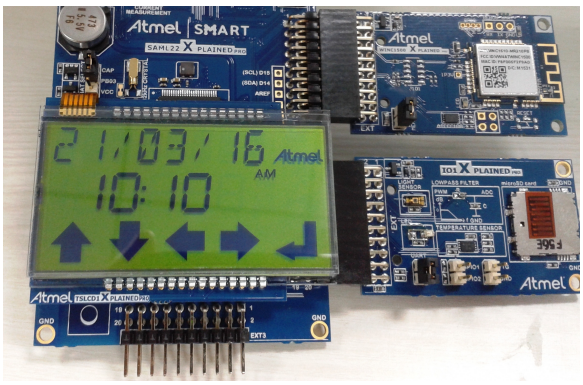
   To configure the date and time, press the "ENTER" button (touch button) available in lower - right corner of the SLCD.

Figure 6-1. Startup



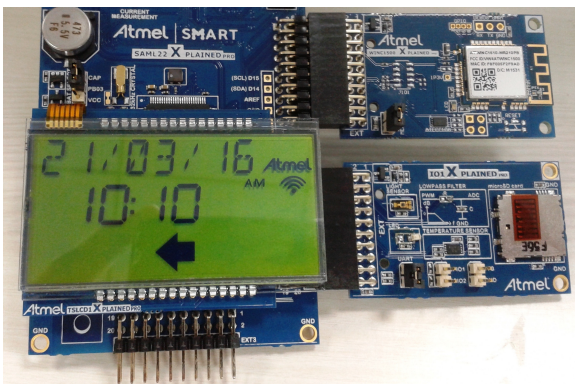
When "ENTER" button is pressed, all the 5 touch buttons become visible and active. Date segements will start to blink indicating that using "UP/DOWN" arrow touch buttons, date can be adjusted. After displaying 31, the Date will be rolled over to 1. This roll over feature is present for month, year and other segments also. Similarly RIGHT / LEFT buttons are used to navigate to month, year, time segments and can be set accordingly. After date and time is set, press the "ENTER" again to exit the calendar configuration. When date and time is configured, Wi-Fi comunication is started.

Figure 6-2. Configure the Calendar



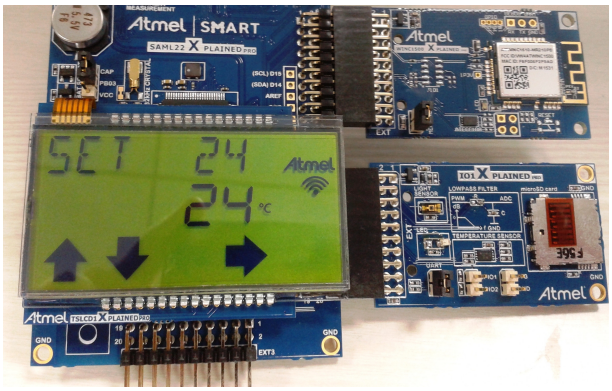
If the mentioned access point is available, the device will connect to that access point and debug console will reflect the corresponding message if debug is enabled. When Wi-Fi communication is successfully established, the Wi-Fi symbol on the LCD is turned ON to intimate the availability of the Wi-Fi connectivity. The calendar is displayed as shown in the following image.

Figure 6-3. Calendar Display



On pressing "LEFT" button, temperature parameters are shown in the LCD display. The first line shows set temperature and second line shows room temperature. The "UP/DOWN" buttons are used to increase or decrease the set temperature. The "RIGHT" button is used to navigate back to calendar display.

Figure 6-4. Temperature Display

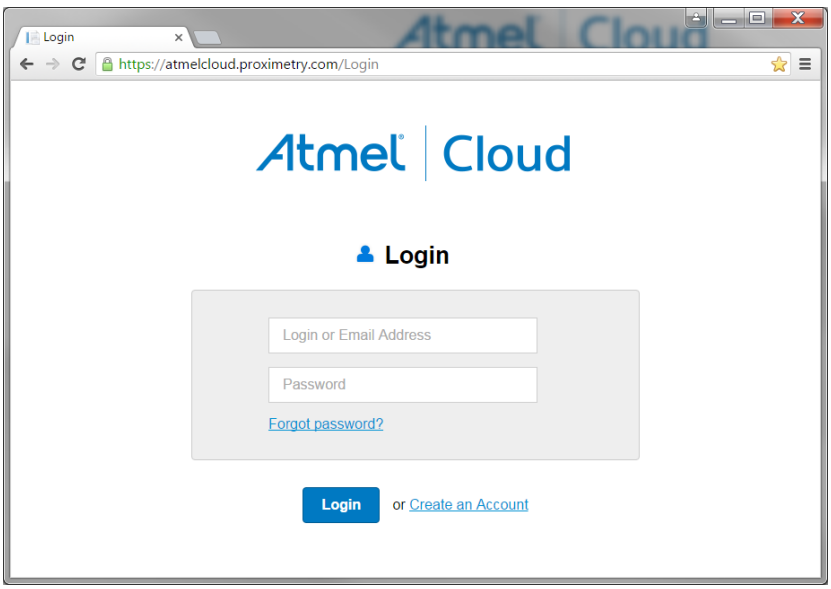


Debug console will display the sync messages at each AirSync Interval.

To access the device from cloud,

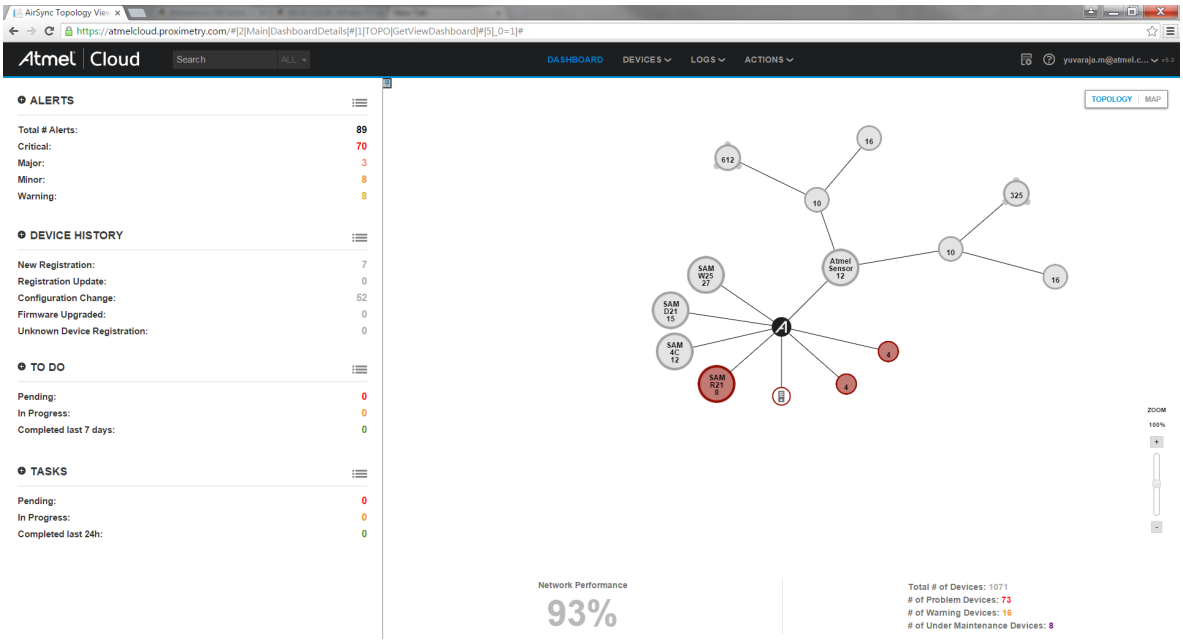
1. Using a PC/laptop, open the browser (preferably Google chrome), and go to Proximetry cloud link https://atmelcloud.proximetry.com/Login
2. New users must register and create an account. The login credentials are provided during registration.

   Figure 6-5. Proximetry Login Webpage



3. After logging in, default screen appears as follows:

Figure 6-6. Default Webpage after Logging in



4. The device can be found by searching using the keyword “Thermostat” in the Search bar. Select the device from the lists displayed based on the ATWINC1500 board MAC address. In the device dashboard, **Summary** tab is available in left hand side and statistics (graph)is displayed on the right hand side.
   Note: Device name in cloud for every device is unique. The full device name consists of predefined device name followed by last 3 bytes of ATWINC1500 board MAC Address. Example: Predefined device name is **SAM L22 Thermostat Demo** and ATWINC1500 MAC address is **F8F005F2F9AD**. The device name displayed in the cloud as **SAM L22 Thermostat Demo_F2F9AD**

Figure 6-7. Summary Tab on Left Hand Side

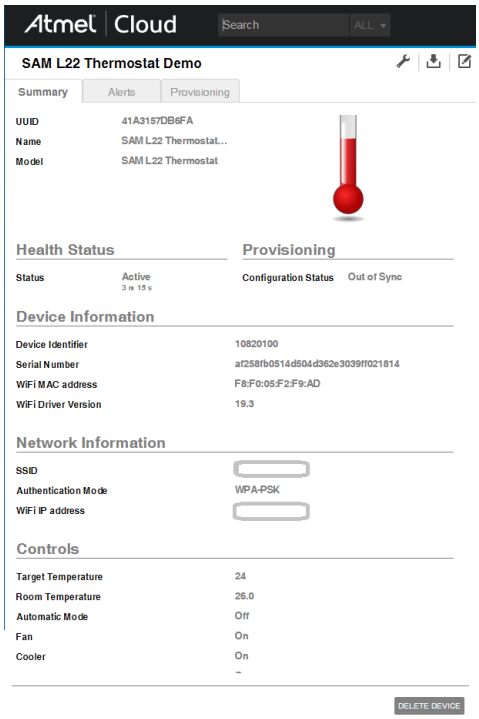
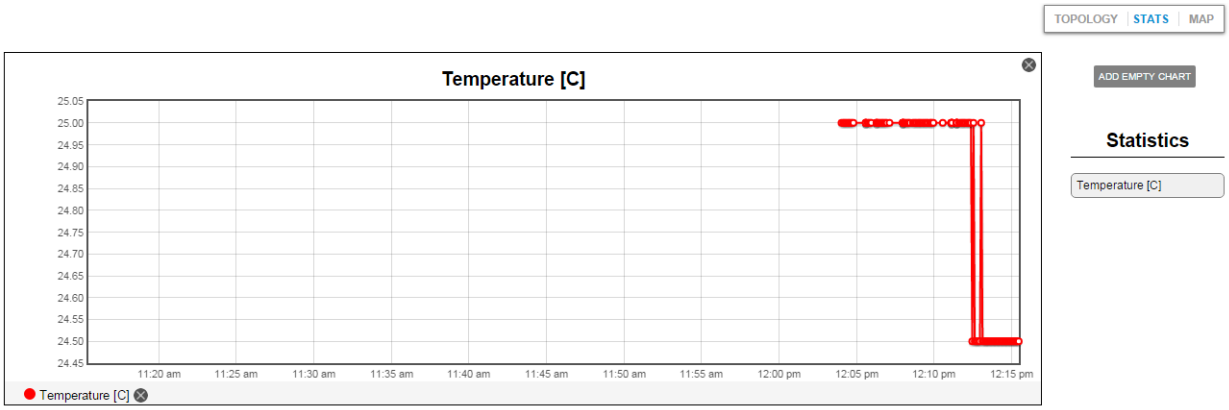
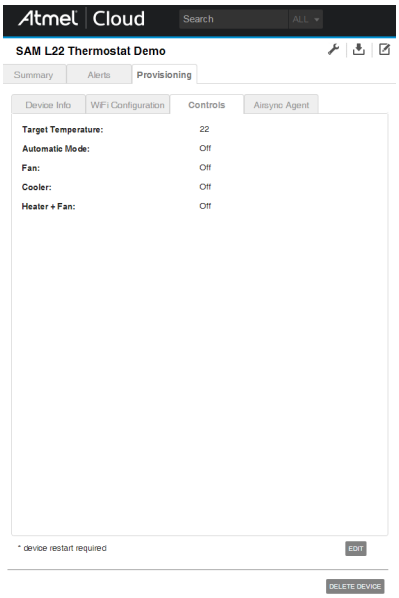


Figure 6-8. Statistics Tab on Right Hand Side



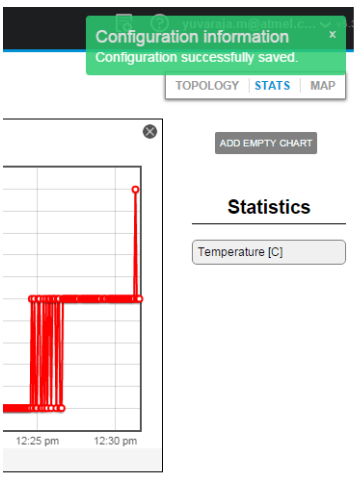
5. To edit the control parameters from cloud, click the **Provisioning** tab on left hand side. Select the **Controls** tab available inside **Provisioning** tab.

Figure 6-9. Provisioning Tab



6. To change the parameters, click **Edit** option available at the bottom of the **Controls** tab. Then edit the parameters (Target Temperature, Automatic mode etc.) and click **Save**. If the configuration is successfully saved, then a message will pop-up in the upper-right corner of the browser.

Figure 6-10. Configuration Change Success



7. Check if the change is reflected on the LCD. Depending on the cloud response and internet connectivity, there could be a delay of few seconds.

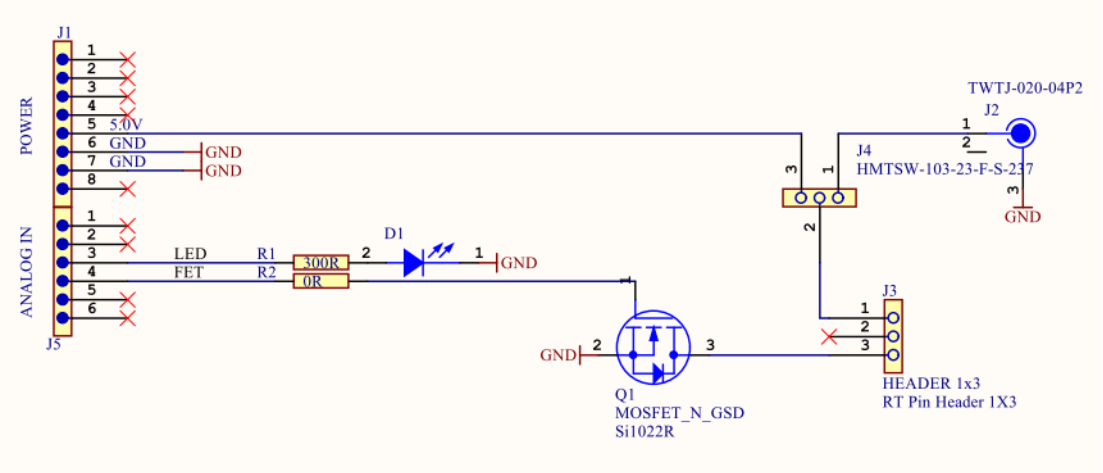
7. References

1. SAM L22 Device Datasheet: http://www.atmel.com/Images/Atmel-42402-SAM-L22_Datasheet.pdf
2. SAM L22 Xplained Pro Schematics: http://www.atmel.com/images/Atmel-42474-SAM-L22-Xplained-Pro_User-Guide.zip
3. SAM L22 Xplained Pro User Guide: http://www.atmel.com/Images/Atmel-42474-SAM-L22-Xplained-Pro_User-Guide.pdf
4. SAM L22 Getting Started Guide: http://www.atmel.com/Images/Atmel-42494-SAML22-Getting-Started-Guide_AT09886.pdf
5. I/O1 Xplained Pro User Guide: http://www.atmel.com/images/atmel-42078-io1-xplained-pro_user-guide.pdf
6. ATWINC1500 Xplained Pro User Guide: http://www.atmel.com/Images/Atmel-42388-ATWINC1500-Xplained-Pro_UserGuide.pdf
7. Atmel Cloud Partners: http://www.atmel.com/products/wireless/cloud-partners/
8. Atmel IoT Resources: http://www.atmel.com/iot/resources.aspx
9. ATECC508A Node Authentication example: http://www.atmel.com/Images/Atmel-8983-CryptoAuth-ATECC508A-Node-Example-Asymmetric-PKI-ApplicationNote.pdf
10. FreeRTOS Getting Started Application note: http://www.atmel.com/Images/Atmel-42382-http://www.atmel.com/Images/Atmel-42138-Getting-Started-with-FreeRTOS-on-SAM-D20-D21-R21-L21-L22_ApplicationNote_AT03664.pdf

8. Appendix

1. Schematics of the driver board for controlling the fan used in the Thermostat Demo

Figure 8-1. Driver Board Schematics



9. Revision History

Doc. Rev.	Date	Comments
42722A	05/2016	Initial document release.

Atmel | Enabling Unlimited Possibilities®

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