Abstract

This document describes the implementation of SAM L11’s innovative features to demonstrate an ultra low-power and secure LoRa® Node.

- Application requirements
- How to build and load the application on a SAM L11 target device
- Technical solution description and key SAM L11 features used to build the demonstration

The demonstration source codes are available with this application note.
# Table of Contents

Abstract........................................................................................................................................................................... 1

1. Introduction........................................................................................................................................................................... 3

2. Hardware and Software Requirements.......................................................................................................................... 8
   2.1. Hardware Requirements................................................................................................................................................... 8
   2.2. Software Requirements................................................................................................................................................... 11

3. Demonstration Setup............................................................................................................................................................ 13
   3.1. Hardware Setup............................................................................................................................................................... 13
   3.2. Network Setup................................................................................................................................................................. 13
   3.3. Software Setup................................................................................................................................................................. 14

4. Demonstration Description.................................................................................................................................................... 21
   4.1. Applications Overview....................................................................................................................................................... 21
   4.2. SAM L11 LoRa Demonstration: Secure Project.................................................................................................................. 23
   4.3. SAM L11 LoRa Demonstration: Non-Secure Project......................................................................................................... 27
   4.4. Power Consumptions of the SAM L11 LoRa Node............................................................................................................... 31

5. System Resources and Software Project Configuration.................................................................................................. 32
   5.1. Secure Project................................................................................................................................................................. 33
   5.2. Non-Secure Project......................................................................................................................................................... 36

6. Appendix................................................................................................................................................................................ 42
   6.1. The Things Network Setup.............................................................................................................................................. 42
   6.2. Cayenne............................................................................................................................................................................ 45
   6.3. Hardware Modifications...................................................................................................................................................... 50

The Microchip Web Site.......................................................................................................................................................... 53

Customer Change Notification Service.................................................................................................................................. 53

Customer Support..................................................................................................................................................................... 53

Microchip Devices Code Protection Feature......................................................................................................................... 53

Legal Notice.............................................................................................................................................................................. 54

Trademarks............................................................................................................................................................................... 54

Quality Management System Certified by DNV.................................................................................................................. 55

Worldwide Sales and Service.................................................................................................................................................. 56
1. **Introduction**

The different elements of the demonstration environment are based on a typical Internet of Things (IoT) network as shown in the following figure:

**Figure 1-1. IoT Network**

Each of these elements represents a specific part of an IoT network:

1. **The Things Network (TTN)** – A service for IoT networking exclusively on LoRa communications. It builds a large network with gateways, based on LoRaWAN™ protocol, to increase the number of existing LoRa applications and users. This protocol allows multiple features suitable for IoT, low-battery usage, long range, low bandwidth, and low-noise attenuation. Refer to *The Things Network Setup* for additional information about TTN and device registration.

2. **Cayenne** – A front-end web site made to simplify the creating and developing of LoRa-enabled IoT solutions. It enables different features, such as Data visualization, SMS and email alerts, triggers, and remote monitoring. The figure below shows the Cayenne dashboard of the LoRa Node on a smartphone:
Figure 1-2. Cayenne Dashboard

Refer to the chapter Cayenne for additional information about Cayenne configuration.

3. **The Gateway** – Enables the SAM L11 LoRa Node (SAM L11 + RN2483 or RN2903 wireless LoRa modules) to connect to the Cloud (The Things Network). Many gateways exist for LoRa communication. The demonstration should work with every LoRa gateway solution made for 868 MHz or 915 MHz, which are compliant with TTN network. Refer to the following web site for additional information about The Things Gateway (TTGateway): https://www.thethingsnetwork.org/docs/gateways/

4. **The SAM L11 LoRa Node** – The main part of the demonstration environment is based on the Microchip SAM L11 microcontroller connected with the LoRa Click Board™ and the IO1 Xplained Pro extension board. The following figure shows how the system makes use of the ARM® Trustzone® for ARM® Cortex®-M devices to store the LoRa keys and the RN2xx3 interface library into a Secure application, and manage interactions between the Secure and Non-Secure.
The following figure shows detailed information on the peripherals used with the SAM L11 LoRa Node:
Data Flash is configured as a secure memory to store the LoRa keys, and the RN2xx3 interface library is stored in the secure part of the Flash. There are three keys for an Over-The-Air Activation (OTAA) procedure:

- The Application EUI (AppEUI), that is, the application identifier
- The Device EUI (DevEUI), that is, the end device identifier
- The AppKey used to derive keys for security (for example, encryption)

The SERCOM 2 in UART mode and SERCOM 0 in UART mode are used to communicate with the RN2483 or the RN2903 wireless LoRa module and the EDBG console.

In the Non-Secure project, the ADC is configured to measure data received from the light sensor of the IO1 Xplained Pro. The SAM L11 on-board LED is used as a control LED to know when the SAM L11 is sending data.

The Non-Secure application uses the secure library provided by the Secure project to set up a secure low-power LoRa application. This application uses the SAM L11 LoRa Node to send luminosity status to the Cloud to inform the user (with text messages and emails) of the changes, allowing the user to take measures according to the information received. If no luminosity change is measured on the ADC, the SAM L11 LoRa Node is put in low-power mode (Standby mode for the SAM L11 and Sleep mode for the RN2483 or RN2903).
Figure 1-5. SAM L11 LoRa Node Set Up

Refer to 4. Demonstration Description for additional information about the Secure and Non-Secure applications.
2. **Hardware and Software Requirements**

Software Requirements:
- Atmel Studio 7 (build 1931 or later)
- SAM L11 DFP version 1.0.81
- Tera Term: [https://osdn.net/projects/ttssh2/releases/](https://osdn.net/projects/ttssh2/releases/)

Hardware requirements for the European region:
- 1 x Microchip SAM L11 Xplained Pro (dm320205)
- 1 x Microchip IO1 Xplained Pro extension (ATIO1-XPRO)
- 1 x MikroElektronika LoRa click Board (includes Microchip RN2483 wireless LoRa module)
- 1 x 868 MHz antenna
- 1 x 868 MHz Gateway (compliant with TTN network)
- 1 x Micro USB cable (type-A/Micro-B)

Hardware requirements for the North American region:
- 1 x Microchip SAM L11 Xplained Pro (dm320205)
- 1 x Microchip IO1 Xplained Pro extension (ATIO1-XPRO)
- 1 x MikroElektronika LoRa click Board (includes Microchip RN2903 wireless LoRa module)
- 1 x 915 MHz antenna
- 1 x 915 MHz Gateway (compliant with TTN )
- 1 x Micro USB cable (type-A/Micro-B)

2.1 **Hardware Requirements**

2.1.1 **Microchip SAM L11 Xplained Pro**

The Microchip SAM L11 Xplained Pro evaluation kit is a hardware platform to evaluate the ATSAML11E16A microcontroller. Supported by the Atmel Studio integrated development platform, the kit provides easy access to the features of the Microchip ATSAML11E16A and explains how to integrate the device in a custom design.

The Xplained Pro MCU series evaluation kits include an onboard Embedded Debugger which overcome the need of external tools to program or debug the onboard microcontroller. The Xplained Pro extension kits offer additional peripherals to extend the features of the board, and ease the development of custom designs. The figure below illustrates the features of the SAM L11 Xplained Pro board.
2.1.2 Microchip IO1 Xplained Pro Extension

The Microchip IO1 Xplained Pro extension board is a generic extension board for the Xplained Pro platform. It connects to any Xplained Pro standard extension header on any Xplained Pro MCU board.

The extension board uses the following functions on the standard Xplained Pro extension header to enhance the feature set of the Xplained Pro MCU boards.

- **SPI**
  - MicroSD card connector
  - 2 GB microSD card included
- **PWM**
  - LED control
  - PWM → Low pass filter → ADC
- **ADC**
  - PWM → Low pass filter → ADC
  - Light sensor
- **UART**
  - Loopback interface via pin header
2.1.3 MikroElektronika LoRa Click Board (Includes Microchip RN2483 or RN2903)

The MikroE LoRa click board is a LoRa RF technology-based SRD transceiver, which operates at 433 MHz or 868 MHz in Europe (with embedded RN2483) or at 915 MHz in North America (with embedded RN2903). This click board is LoRaWAN Class A compliant, and provides a long-range spread spectrum communication with high interference immunity. The module used on this click board is a fully certified LoRa Sub GHz. The RN2483 is compliant with the European RED directive assessed radio modem, the RN2903 is certified for both FCC and IC requirements. These boards are combined with the advanced and simple command interface which enables an easy integration and reduced development time.

- UART - Communicate with RX, TX, RTS and CTS pins
2.2 Software Requirements

2.2.1 Atmel Studio 7 Integrated Development Platform

Atmel Studio 7 is the integrated development platform (IDP) for developing and debugging Atmel ARM® Cortex™-M processor-based and Atmel AVR® microcontroller applications.

The Atmel® Studio 7 IDP gives a seamless and easy-to-use environment to write, build, and debug user applications written in C/C++ or assembly code. Atmel Studio 7 supports all 8-bit and 32-bit AVR, the new SoC wireless family, SAM microcontrollers, and connects seamlessly to Atmel debuggers and development kits.

Figure 2-4. Atmel Studio 7

Atmel Studio 7 is available for download from the following location: http://www.microchip.com/avr-support/atmel-studio-7.
2.2.2 Atmel START
Atmel START is a web-based software configuration tool for starting a new embedded development on Microchip SAM and AVR microcontrollers. Starting from either a new project or an example project, Atmel START enables you to select and configure a set of software components from the SAM Advanced Software framework to tailor your embedded application in a usable and optimized manner. Atmel START supports code project generation for Atmel Studio 7, IAR Embedded Workbench®, Keil μVision®, or generic makefile generation.

Figure 2-5. Atmel START
3. Demonstration Setup
Demonstration involves the following steps:

3.1 Hardware Setup
To setup hardware, users need to perform these actions: Connect the IO1 Xplained Pro extension board and LoRa click board to the SAM L11 Xplained Pro board, and then set the board jumpers as shown in the below image. The demonstration hardware is now ready for evaluation.

Figure 3-1. Hardware Setup

Note: Ensure that the UART jumper on the IO1 Xplained Pro board is removed, otherwise the SAM L11 will not be able to communicate with the RN2483 or RN2903 module.

3.2 Network Setup
The Things Network (TTN) Setup
Refer to 7.1 The Things Network for additional information about TTN setup and device registration if TTN is not setup.
TTN-Compliant LoRa Gateway Setup
The following link provides a list of TTN compliant gateways and how to get started: https://www.thethingsnetwork.org/docs/gateways/.

Cayenne Setup
Refer to 7.2 Cayenne for additional information about device registration on Cayenne, payload format, and Cayenne intelligent features if Cayenne is not setup.

The network is now setup.

3.3 Software Setup
To open and load the demonstration project on the target hardware, follow these steps:

1. Open Atmel Studio 7.0.
2. Open the demonstration project: From File > Open > Project/Solution.
3. Browse and select the SAM_L11_LoRa_Demo.atsln file, which comes along with the demonstration package.
4. When the project is loaded under Atmel Studio 7.0 IDE, select the EDBG board as the debugger/programmer tool.
5. From the Solution Explorer panel, select Secure_project > Properties.
6. Select Tool from the properties window.
7. Select the connected board EDBG as the debugger/programmer.
8. Select SWD as interface.
9. Click  to save the configuration.

10. Click  or F7 to build the demonstration.

11. Check whether any error message is appeared in the IDE Output window.

   **Figure 3-4. IDE Output Window**

   ![IDE Output Window]

   **Important:** Ensure that the Secure project is compiled if any error occurs during the compilation.

12. Flash the demonstration software on the hardware by clicking  (Ctrl+Alt+F5).

13. Open the Tera Term tool or any equivalent tool.

14. Choose the COM port number allocated to the SAM L11 Xplained Pro.
Figure 3-5. Tera Term New Connection Window

Note: The COM port number can be retrieved from Windows Devices Manager.

Figure 3-6. Windows Device Manager Window

15. Configure the serial interface using the parameters shown below.
16. Reset the board by pressing **RESET**.
   The demonstration will start by displaying the below messages on the terminal window. The displayed messages will depend if the different application keys have been provisioned on the device. Refer to **Demonstration Description** for additional information on demonstration behavior.

Figure 3-8. Terminal Window Displaying the Message

```
SAM L11- LoRa Demo
Secure Project

- Low level initialization
- Initialize RN2XX3 Module
- Set RN2XX3 keys for Over-The-Air Activation
- Load Non-secure Application
```
17. To start the Data Visualizer to measure the power consumption of the LoRa Node, follow these steps: Open Data Visualizer on Atmel Studio by clicking Tools > Data Visualizer.

*Figure 3-9. Atmel Studio*

18. Select SAM L11 Xplained Pro and click Connect.

*Figure 3-10. DGI Control Panel*

19. Select Power, and then click Start.

*Figure 3-11. DGI Control Panel*

20. The Data Visualizer will display the SAM L11 LoRa power consumption details.
Figure 3-12. Data Visualizer Showing Power Consumption
4. Demonstration Description

4.1 Applications Overview

The user can evaluate the standard TrustZone for the Armv8-M mechanism by executing the application. This Non-Secure application code uses the secure library provided by the Secure application code to build a smart lighting network with low-power based on LoRaWAN protocol.

Figure 4-1. Non-Secure Application Code

The following color schemes are used to display messages in the terminal window:

- Red: Non-Secure
- Yellow: Non-Secure Callable
- Green: Secure

When system initialization is performed, and Non-Secure application code is started, access to the Secure library from the Non-Secure application is done through standard function calls to a predefined set of secure APIs. These secure APIs are defined by the secure code and stored in a Non-Secure Callable region of the SAM L11 device. They constitute a set of secure gateways (veneers) that limit the access to the secure software.

Figure 4-2. Non-Secure to Secure function call mechanism
The following figures show an exhaustive list of secure gateways (veneers) provided by the Secure project:

**Figure 4-3. Secure Gateway (Veneers)**

```c
/* ************ Secure LoRa click-board NSC functions ************ */

/* -----------------------------------------------
   - NAME : void nsc_rn2xx3_break(void)
   - DESCRIPTION : (Secure Gateway) Send a break condition (0x55) to the rn2483
   - PARAMETERS : None
   - RETURN : None
 ----------------------------------------------- */

extern void nsc_rn2xx3_break(void);

/* -----------------------------------------------
   - NAME : void nsc_rn2xx3_sleep (char* sleep_duration)
   - DESCRIPTION : (Secure Gateway) Put the rn2483 module into sleep mode for "sleep_duration" milliseconds
   - PARAMETERS : char* sleep_duration
   - RETURN : None
 ----------------------------------------------- */

void nsc_rn2xx3_sleep (char* sleep_duration);

/* -----------------------------------------------
   - NAME : uint8_t nsc_rn2xx3_join_OTAA_network(void)
   - DESCRIPTION : (Secure Gateway) Set all parameters to configure and launch an OTAA (Over-The-Air Activation) connection
   - PARAMETERS : None
   - RETURN : rn2483_Status
 ----------------------------------------------- */

uint8_t nsc_rn2xx3_join_OTAA_network(void);

/* -----------------------------------------------
   - NAME : uint32_t nsc_rn2xx3_transmit(char* port ,char* payload, char* rx_port,char* rx_payload)
   - DESCRIPTION : (Secure Gateway) Send a command to the LoRa click-board
   - PARAMETERS : char* port
                  char* payload
                  char* rx_port
                  char* rx_payload
   - RETURN : rn2xx3_Transmission_Status
 ----------------------------------------------- */

uint32_t nsc_rn2xx3_transmit(char* port, char* payload, char* rx_port, char* rx_payload);

/* -----------------------------------------------
   - NAME : uint32_t nsc_rn2xx3_transmit_without_ACK(char* port ,char* payload)
   - DESCRIPTION : (Secure Gateway) Send a command to the LoRa click-board
   - PARAMETERS : char* port
                  char* payload
   - RETURN : rn2xx3_Transmission_Status
 ----------------------------------------------- */

uint32_t nsc_rn2xx3_transmit_without_ACK(char* port, char* payload);
```

**Figure 4-4. Secure Gateway (Veneers)**

```c
/* -----------------------------------------------
   - NAME : void nsc_periph_clock_init(uint32_t pch_id, uint32_t gclk_source)
   - DESCRIPTION : (Secure Gateway) Change peripheral clock settings
   - PARAMETERS : uint32_t pch_id
                  uint32_t gclk_source
   - RETURN : None
 ----------------------------------------------- */

void nsc_periph_clock_init(uint32_t pch_id, uint32_t gclk_source);
```
4.2 **SAM L11 LoRa Demonstration: Secure Project**

The Secure project executed at product startup manages the critical aspect of the application. It is stored and runs from the secure memories of the SAM L11, which are isolated from the Non-Secure software with the help of the Cortex-M23 TrustZone. It is in charge of performing the following tasks in the application:

- Low-level system settings
- Providing the library and associated veneers (API) for the RN2483 or RN2903 module (SERCOM) to the Non-Secure project
- Store the application keys from the RN2483 or RN2903, and TTN
- Loading the Non-Secure application
- Manage the serial communication with the Host computer and the LoRa click-board

The following is the flowchart of the Secure project main routine:

```c
/* *********** Secure SAML11_Low_Power NSC functions *********** */

/* ----------------------------------------------- */
- NAME :   void nsc_secure_enter_standby (void)
- DESCRIPTION : (Secure Gateway) Put the SAM L11 in Standby mode (Low-power)
- PARAMETERS : None
- RETURN :   None
------------------------------------------------------- */

void nsc_secure_enter_standby (void);

/* *********** Secure console NSC functions *********** */

/* ----------------------------------------------- */
- NAME :   void nsc_secure_console_puts (char *string)
- DESCRIPTION : (Secure Gateway) Write a string to the console
- PARAMETERS : char * string
- RETURN :   None
----------------------------------------------- */

void nsc_secure_console_puts (char *string);
```
The above flowchart illustrates the different cases available in the Secure project. To start the Key provisioning operation, the user must press the SW0 button during the Secure project start until the LoRa Keys configuration screen appears for key provisioning. The LoRa keys to be provisioned must be in line with the user’s TTN setup, and are stored in the secure memory region of the product Data Flash. After the provisioning sequence, the keys will be retrieved from the secure memory region of the product Data Flash during each application startup, until the product Data Flash is cleared (Full chip erase). These keys will be used by the application to join to TTN and send data. These keys are accessible on TTN after the device is registered.

4.2.1 Keys Not Provisioned
This section describes how key provisioning is realized during boot configuration. If the LoRa keys are already stored in the Data Flash, skip this part and then proceed to Keys Provisioned.

The following figure shows the messages displayed by the console after the LoRa keys configuration is entered by pressing the SW0 button after resetting the SAM L11 LoRa Node.
The user can enter DevEUI, AppEUI, and AppKey keys to be stored in Data Flash as shown in the below image.

Figure 4-7. Console Displaying the key provisioning entry Message
4.2.2 Keys Provisioned

The following image shows the information displayed by the console after a SAM L11 reset when the keys are provisioned:

Keys are stored into the Data Flash and are set in the RN2483 or RN2903 module for the OTAA procedure.
The low-level settings are configured in the Secure project, which controls performance level, clock sources, clock domains, wait states, and the TrustZone manager. Peripherals like SERCOM are used to communicate with the RN2483 or RN2903, the console, and Data Flash for keys storage.

The RN2483 or RN2903 module is initialized in the Secure project and the Over-The-Air Activation (OTAA) LoRa keys are read from the Data Flash and set for joining TTN and next transmissions.

4.3 **SAM L11 LoRa Demonstration: Non-Secure Project**

The Non-Secure application makes use of the secure library provided by the Secure project to set up the secure low-power LoRa application. This application uses the SAM L11 LoRa Node to send the luminosity status to the Cloud to inform the user (using text messages or emails) of the changes. To reduce the overall application power consumption on both the SAM L11 and the RN2483 or RN2903, the devices are placed in Low-Power mode when an ambient luminosity change is not detected.

When SAM L11 is in Standby mode, it uses a key feature called SleepWalking with Dynamic Power Gating (Dynamic SleepWalking) to reduce the power consumption. SleepWalking gives the capability for the SAM L11 to wake up temporarily and asynchronously a peripheral without waking up the CPU. With Dynamic Power Gating, the SAM L11 can turn on or off power of certain peripherals dynamically whether they are used or not.

The following figure illustrates the demonstration behavior and the benefits of Dynamic SleepWalking on power consumption:
The ADC is woke up every ten seconds by the RTC to convert the ambient luminosity level and compare the result with a preset threshold value. Due to the SleepWalking feature, the CPU will be woken up only if the threshold is exceeded. This allows the SAM L11 to remain in Standby mode for a long time.

The following figure shows the message displayed on the console by the Non–Secure application in successful connection case:
For this example, the LoRa Node tries and succeeds to join the OTAA network. The starting ADC value is used for the first transmission to refresh the cloud values. Then the RN2483 or RN2903 module and the SAM L11 are placed in low-power mode:

**Figure 4-12. OTAA succeeded Message**

Low luminosity is detected by the ADC, and the RN2483 or RN2903 module is woke up from Sleep mode to send the status of ambient luminosity. Luminosity status is transmitted to the cloud, which transmits back an acknowledge to the RN2483 or RN2903 module. Until high luminosity is detected, the RN module and SAM L11 are put in low-power mode:
Figure 4-13. LoRa Transaction in Active Mode Message

- Environmental light is high: SAM L11 woke up
- RN2XX3 module is woken up from Sleep mode
  → Non-Secure callable → RN2XX3 is woken up
- Transmit Luminosity status
  → Non-Secure callable → Transmit luminosity status with acknowledgment
- Luminosity status sent to the server
- SAM L11 LoRa Node is put in low-power mode until luminosity changes
  → Non-Secure callable → RN2XX3 is put in Sleep mode
  → Non-Secure callable → SAM L11 is put in Standby mode;

Note: Sometimes the status of Luminosity does not appear on the Cayenne dashboard. In that case, users need to refresh their browser.

The following figure illustrates the flow chart of the Non–Secure application:

Figure 4-14. Non-Secure Application Flow Chart
4.4 Power Consumptions of the SAM L11 LoRa Node

When the application is running, the dynamic current consumption of the whole application can be measured with the Data Visualizer tool from Atmel Studio. The current consumption of the application will vary according to the ambient light intensity. This variation is due to the analog light sensor technology implemented on the IO1 Xplained Pro extension board.

When the last status of ambient luminosity is low, the Data Visualizer will display the following screen:

Figure 4-15. Node Power consumption (ambient luminosity low)

![Power consumption graph for low luminosity](image)

When the last status of ambient luminosity is high, the Data Visualizer displays the following screen:

Figure 4-16. Node Power consumption (ambient luminosity high)

![Power consumption graph for high luminosity](image)

**Note:** To reduce the power consumption, some modifications are made on the RN2483 or RN2903 module and the IO1 Xplained Pro extension board. Refer to the Hardware Modifications section for additional information about these modifications.
5. **System Resources and Software Project Configuration**

The demonstration application is built on Atmel START ASF4. After opening the demonstration project under Atmel Studio, the configuration used to generate the project under Atmel START can be viewed by clicking , which is available on the top of the solution explorer windows:

*Figure 5-1. Solution Explorer Window*

---

**Note:** Clicking the Atmel Start button under Atmel Studio requires internet access as the local project setting will reload on the remote Atmel Start Server frontend.
5.1 Secure Project

5.1.1 Project Architecture
The secure project is built on top of the following ASF4 drivers and middleware:

**Figure 5-2. Secure Project Architecture**

- **FLASH**: Used for key provisioning in Data Flash.
- **RN2XX3_USART_IF**: Interface for the RN2483 or RN2903 USART. This is used to send and receive commands and data to and from the RN2483 or RN2903 module.
- **DELAY_MS**: Used for delaying some commands for synchronization.
- **TRUSTZONE_MANAGER**: Used for managing the product TrustZone for the Cortex-M.
- **CONSOLE**: Used for displaying application output information on the terminal.

A detailed information on component configuration can be found by clicking on each module under Atmel Start.

5.1.2 System Clock Configuration
The System Clock Configuration is accessible by clicking the Atmel START Clock tab. The clock configuration for the application is shown in the following figure:
Three different clock domains are configured in the system due to Generator 0, Generator 1, and Generator 2.

- **Generator 0**: The main clock domain. It provides a switchable 4 MHz clock to the core and the peripherals used in Active mode for the application. Generator 0 is automatically switched off by the system when entering Standby mode (static clock gating).
- **Generator 1**: The slow clock domain. It provides an always ON 32 kHz clock source used by the RTC to run in Standby mode.
- **Generator 2**: It provides an ON DEMAND 1 MHz clock source for the Secure and Non-Secure peripherals.

### 5.1.3 PINMUX Configuration

The PINMUX Configurator is accessible by clicking the Atmel START PINMUX tab. The PINMUX Configurator allows the user to set the configuration of each pin of the SAM L11 device. When using Atmel START, most of the pins are configured automatically according to the project module configured in the DASHBOARD tab.
For additional information on pin assignment for the demonstration context, and detailed information on pin settings, click on each pin in the pin muxing table or on the SAM L11 pinout picture.

5.1.4 Generated Project Source Code: Secure Project

The previous sections of this document have described the application layer of the demonstration. For additional information on middleware and ASF4 can be found on Atmel START help and in the generated source code.

The demonstration project source code is accessible under the Solution Explorer window of Atmel Studio 7 and corresponds to the following application layers:
Figure 5-5. Generated Secure Project Source Code

5.2 Non-Secure Project

The Non-Secure project uses API and functionalities provided by the Secure application library. It does not use any specific ASF drivers, and they do not have access to clock configurations of the system. The Non-Secure project configuration under Atmel START is shown in the following figure:
5.2.1 Generated Non-Secure Project Source Code
The generated non-secure project source code details are shown in the below image:

5.2.2 PINMUX Configuration: Non-Secure Project
The PINMUX Configurator is accessible by clicking the Atmel START PINMUX tab. The PINMUX Configurator enables the user to set the configuration of each pin of the SAM L11 device. When using Atmel START, many pins are configured automatically according to the project module as configured in the DASHBOARD tab.
The PA07 pin is configured as a Non-Secure pin in the Non-Secure project, and this pin is used to configure the SAM L11 onboard LED.

**Note:** Only the pins defined as Non-Secure in the Secure Project TrustZone Manager can be handled in the PINMUX configurator of the Non-Secure project.

### 5.2.3 Creating Application

Follow these steps for creating an application:

1. Before configuring the gateway, users need to create the application and register the devices.
2. Create an user account on TTN. TTN can be downloaded from the following location: [https://www.thethingsnetwork.org/](https://www.thethingsnetwork.org/), see below image for TTN main window.
3. Enter User Name, Email Address, Password, and then Click **Create account**.
4. After creating the TTN account, follow these steps to create an application.
5. The TTN window will display the newly created account.
6. Click **CONSOLE**.

   **Figure 5-11. TTN Console Window**

7. Click **APPLICATIONS**, and then click **add application** on the top right.

   **Figure 5-12. Add Applications area**
8. Add Applications window will be displayed. Enter details for Application ID, Description with user-specific information, Application EUI, and server information in the Handler registration field according to the region where the application will be set.

**Figure 5-13. Add Applications Window**

![Add Application Window](image)

**Note:** Application EUI will be generated by TTN.

9. Click **Add application** to add the TTN application.
6. Appendix

6.1 The Things Network Setup
The Things Network is a service for IoT networking, that is, building a large network with gateways for LoRa communication, based on the LoRaWAN protocol, to increase the number of existing LoRa applications and users. This protocol allows multiple features that are perfect for IoT, such as low-battery usage, long range, low bandwidth, and low-noise attenuation.

6.1.1 Device Registration
Once the application is created, the device can be registered in the new application using these steps:

1. From the TTN console, click APPLICATIONS.
2. Click on the application to have an overview, and then click register device.

Figure 6-1. Application Window

3. Enter information for Device ID, Device EUI, App Key, and App EUI, and then Click Register to register the device in the TTN Application.
Figure 6-2. Device Register Window

Note:
1. The Device EUI can be generated by TTN or set by the user. Click on the pencil icon, below Device EUI, to switch between choices.
2. The AppKey can be generated either by TTN or set by the user. Click on the pencil icon, below App Key, to switch between choices.

6.1.2 Cayenne Integration
The user must create a link between the TTN application and the Cayenne dashboard. After creating the link, the device must be registered with Cayenne. Follow these steps to register the device with Cayenne.

1. From the TTN console, click APPLICATIONS.
2. Select the application to have an overview, and then click Payload Formats.

Figure 6-3. Application Window

3. In the Payload Format overview, click Payload Format, and then select Cayenne LPP instead of Custom.
4. When finished, click Save.
5. Click Integrations.
6. In Integrations Overview, click **add integration** and then select Cayenne LPP.

7. Select a Process ID for the Cayenne integration, and then select default key in the Access Key box.

8. Select Add Integration to link the TTN Application with Cayenne.
6.2 Cayenne

Cayenne is a front end web site that simplify the action of creating and developing of LoRa-enabled IoT solutions. It allows features such as, Data visualization, SMS, email alerts, triggers, and remote monitoring. Cayenne is available on multiple platforms, such as IOS, Android, and Windows.

Cayenne is available for download form the following location: https://mydevices.com/

Figure 6-8. Cayenne Main Page
The user must create a free account on Cayenne. After creating the account, users need to follow these steps to setup Cayenne for the demonstration.

6.2.1 Registering Device
To register the devices on Cayenne, follow these steps:

1. Log on using the newly created Cayenne account, and then click LoRa.

   Figure 6-9. Cayenne Main Page

2. Scroll down to the LoRa section, and then click The Things Network.
3. Scroll down and click **Cayenne LPP**.
4. Enter *DevEUI* of the device registered in TTN application, change the device name if required, and then click **Add device**.
5. The device is registered in Cayenne.

6.2.2 Configure Triggers
The Trigger feature enables automation of the LoRa application using ‘If or Then’ statements, which are based on real-time data and actions. Follow these steps to build the automation of the LoRa:

1. To configure the widgets that are on the Cayenne dashboard, click wheel icon on the widget block.

**Figure 6-12. Widget Block**

*Note:* The widget appears only after the first connection of the node on the network.
2. Configure Digital Input (2) with the different parameters, as shown in the below image.

Figure 6-13. Widget Configuration Window

3. After widgets are configured, click Trigger which is available under Add new.

Figure 6-14. Add New Trigger section

4. Drag and drop the device into the if box, and then click setup notification in the then box.
5. When *if* and *then* boxes are filled, reproduce the following triggers.

**Figure 6-16. Reproducing the Triggers**

**Note:** Notifications through text message or emails can be chosen by selecting Send Text messages or Send Email options, or by adding a custom recipient (mobile number or email address).

6. Click *Save* to save the triggers.

7. The triggers are now configured in Cayenne. The board can be reset to have the demonstration running.

### 6.3 Hardware Modifications

To reduce the consumption of the global demonstration, some modifications are applied to the MikroElektronika LoRa click board and the IO1 Xplained Pro extension board.

#### 6.3.1 MikroElektronika LoRa Click Board

The embedded voltage regulator and power LED populated on the Mikroelektronika LoRa click board are not used by our application, and it can be removed to reduce the overall application power consumption.
This embedded voltage regulator can be removed and bypassed using a wire connected from MikroBus 3.3V pin to the regulator output soldering pad. All modifications made on RN2483 or RN2903 module are shown in the image below. The embedded regulator output voltage pin is connected to the 3.3 voltage pin.

**Figure 6-17. RN2483A Showing Modifications**

### 6.3.2 IO1 Xplained Pro Extension Board

The power consumption of the overall application can be reduced by isolating the light sensor on the IO1 Xplained Pro board, that is removing all components on the board except light sensor and the associated resistor.

The following figure illustrates the IO1 Xplained Pro board.
Figure 6-18. IO1 Xplained Pro Board
The Microchip Web Site

Microchip provides online support via our web site at http://www.microchip.com/. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user’s guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip’s customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.


Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Heldo, JukeBlox, KeeLoq, Kleer, LANCheck, LINK MD, maXSylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.


SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.
Quality Management System Certified by DNV

ISO/TS 16949
Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.
<table>
<thead>
<tr>
<th>AMERICAS</th>
<th>ASIA/PACIFIC</th>
<th>ASIA/PACIFIC</th>
<th>EUROPE</th>
</tr>
</thead>
</table>