AN4152

PIC32CM LS00/LS60 Ultra Low-Power Secure LoRa Demonstration

Abstract

This document describes the implementation of the features of the PIC32CM LS00/LS60 to demonstrate an ultra-low power and secure LoRa® Node. It also covers the following topics:

- Application requirements.
- How to build and load the application on a PIC32CM LS00/LS60 target device.
- Technical solution description and the key features of the PIC32CM LS00/LS60 used to build the demonstration.

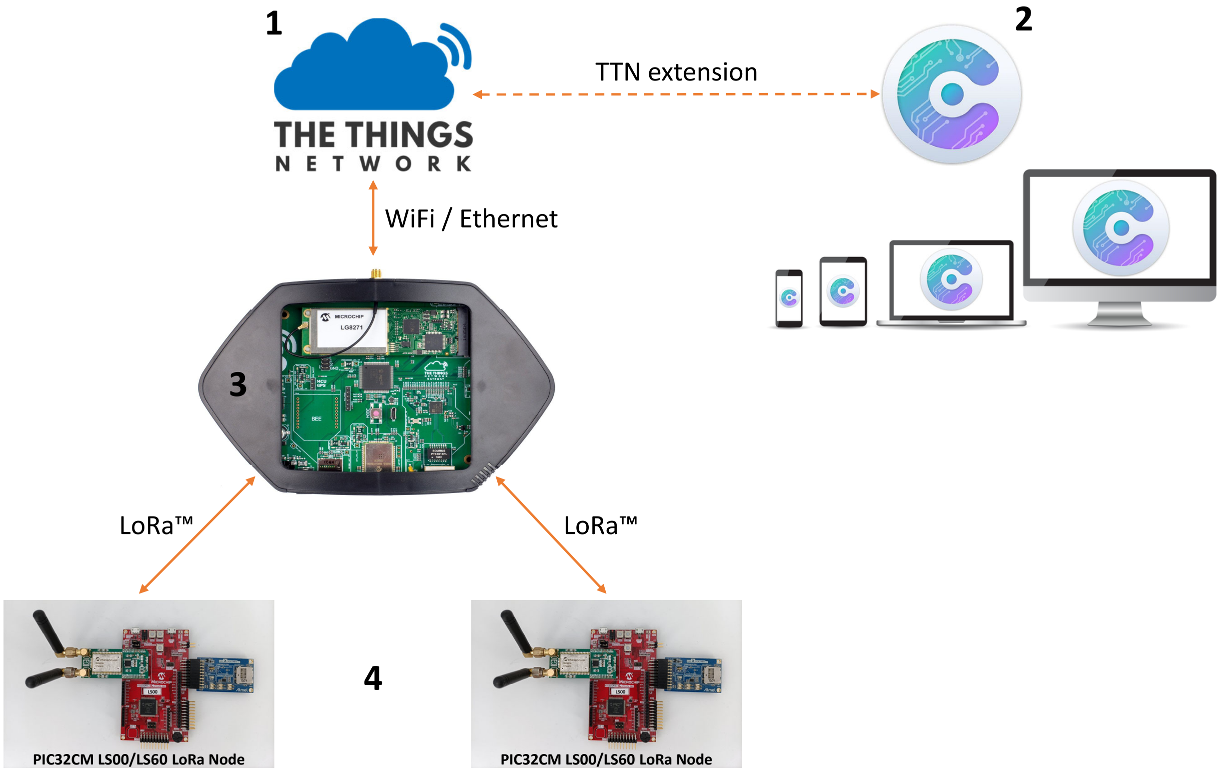
The demonstration source codes are available for download along with this document.

Table of Contents

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:

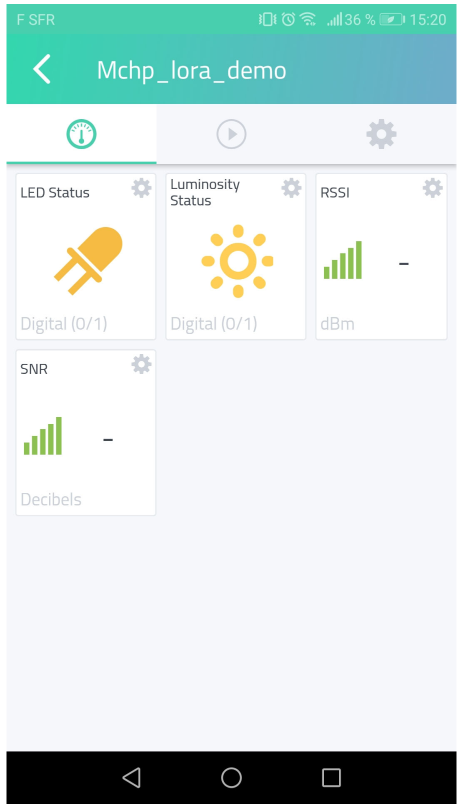
Figure 1-1. IoT Network



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:

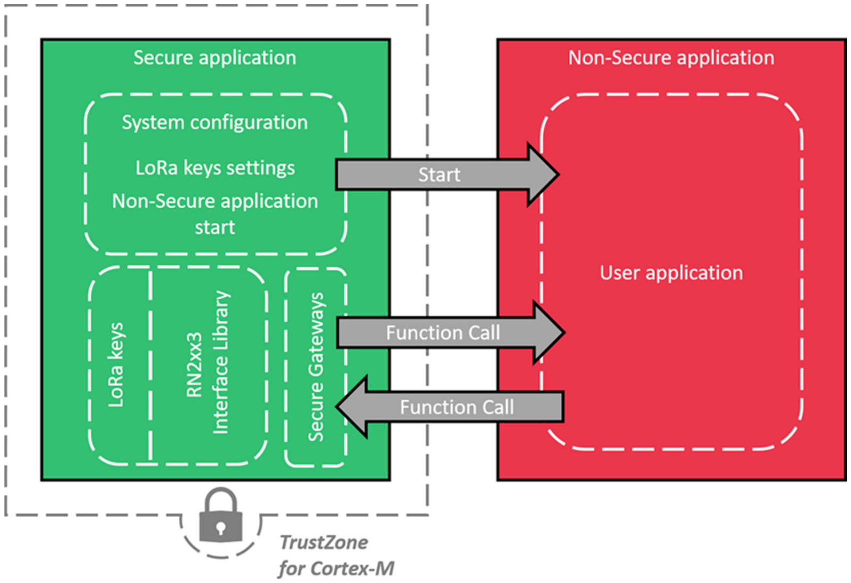
Figure 1-2. Cayenne Dashboard



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

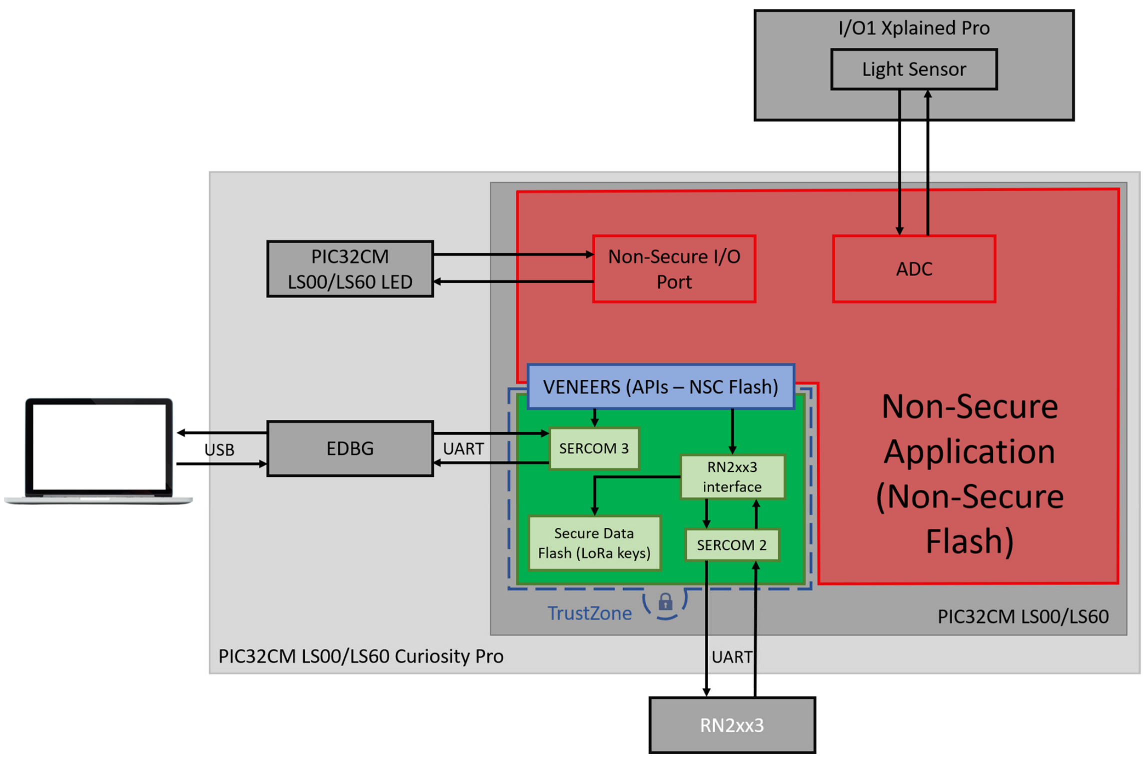
3. **The Gateway** – Enables the PIC32CM LS00/LS60 LoRa Node (PIC32CM LS00/LS60 + RN2483 or RN2903 wireless LoRa modules) to connect to the Cloud (The Things Network). Many gateways exist for LoRa communication. The demonstration must work with every LoRa gateway solution made for 868 MHz or 915 MHz, which are compliant with TTN network. Refer to the following web site for additional information about The Things Gateway (TTGateway): www.thethingsnetwork.org/docs/gateways/
4. **The PIC32CM LS00/LS60 LoRa Node** – The main part of the demonstration environment is based on the Microchip PIC32CM LS00/LS60 microcontroller connected with the LoRa Click Board™ and the I/O1 Xplained Pro extension board. The following figure shows how the system makes use of the Arm® TrustZone® for ARMv8-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.

Figure 1-3. Secure and Non-Secure



The following figure shows detailed information on the peripherals used with the PIC32CM LS00/LS60 LoRa Node:

Figure 1-4. LoRa Node Showing Peripherals



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. The following keys are used for the Over-The-Air Activation (OTAA) procedure:

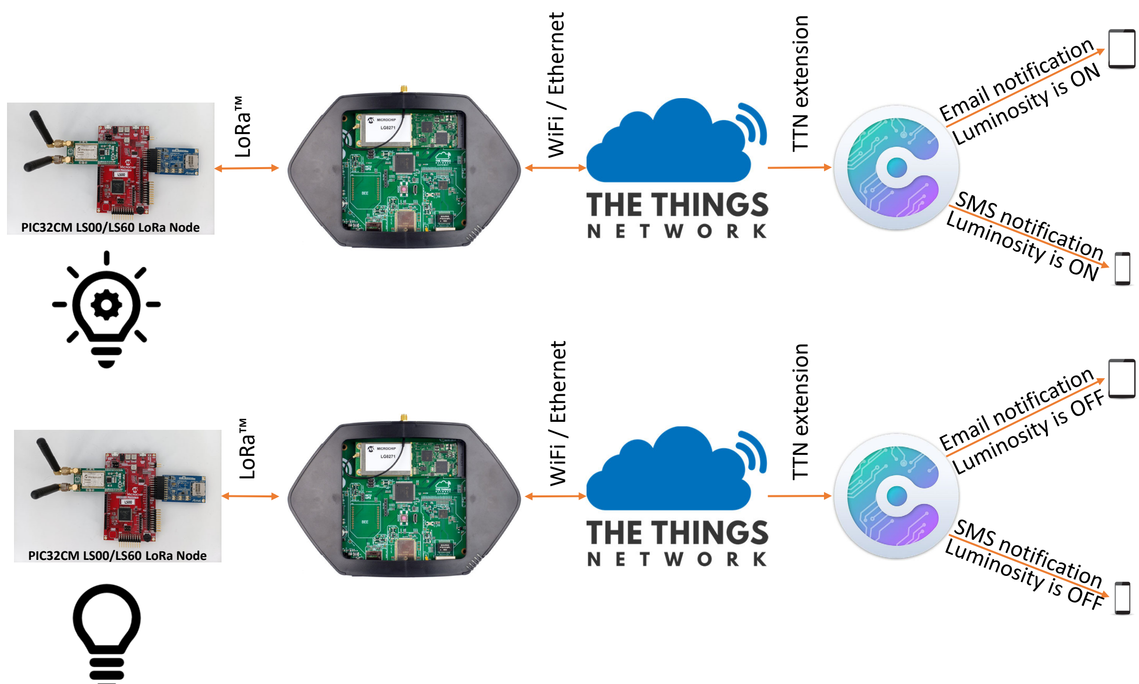
- The Application EUI (AppEUI), that is, the application identifier
- The Device EUI (DevEUI), that is, the end device identifier
- The AppKey used to derive keys for security, for example, encryption

The SERCOM 2 in UART mode and SERCOM 3 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 I/O1 Xplained Pro. The PIC32CM LS00/LS60 on-board LED is used as a control LED to know when the PIC32CM LS00/LS60 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. The secure library allows definition of the LoRa limitations of the demonstration and to reduce accesses to the Secure application from the Non-Secure application by providing a restricted set of APIs for the Non-Secure application. This application uses the PIC32CM LS00/LS60 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 PIC32CM LS00/LS60 LoRa Node is put in low-power mode (Standby mode for the PIC32CM LS00/LS60 and Sleep mode for the RN2483 or RN2903).

Figure 1-5. PIC32CM LS00/LS60 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:

- MPLAB® X IDE most up to date version
- MPLAB Code Configurator (MCC) for MPLAB Harmony v3 most up to date version
  - The csp package
  - The csp_apps_pic32cm_le_ls package
- MPLAB Data Visualizer (standalone or MPLAB X IDE plugin version)
- The PIC32CM LS00/LS60 DFP package
- Tera Term: osdn.net/projects/ttssh2/releases/

Hardware requirements for the European region:

- 1 x Microchip PIC32CM LS00/LS60 Curiosity Pro
- 1 x Microchip I/O1 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 PIC32CM LS00/LS60 Curiosity Pro
- 1 x Microchip I/O1 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)

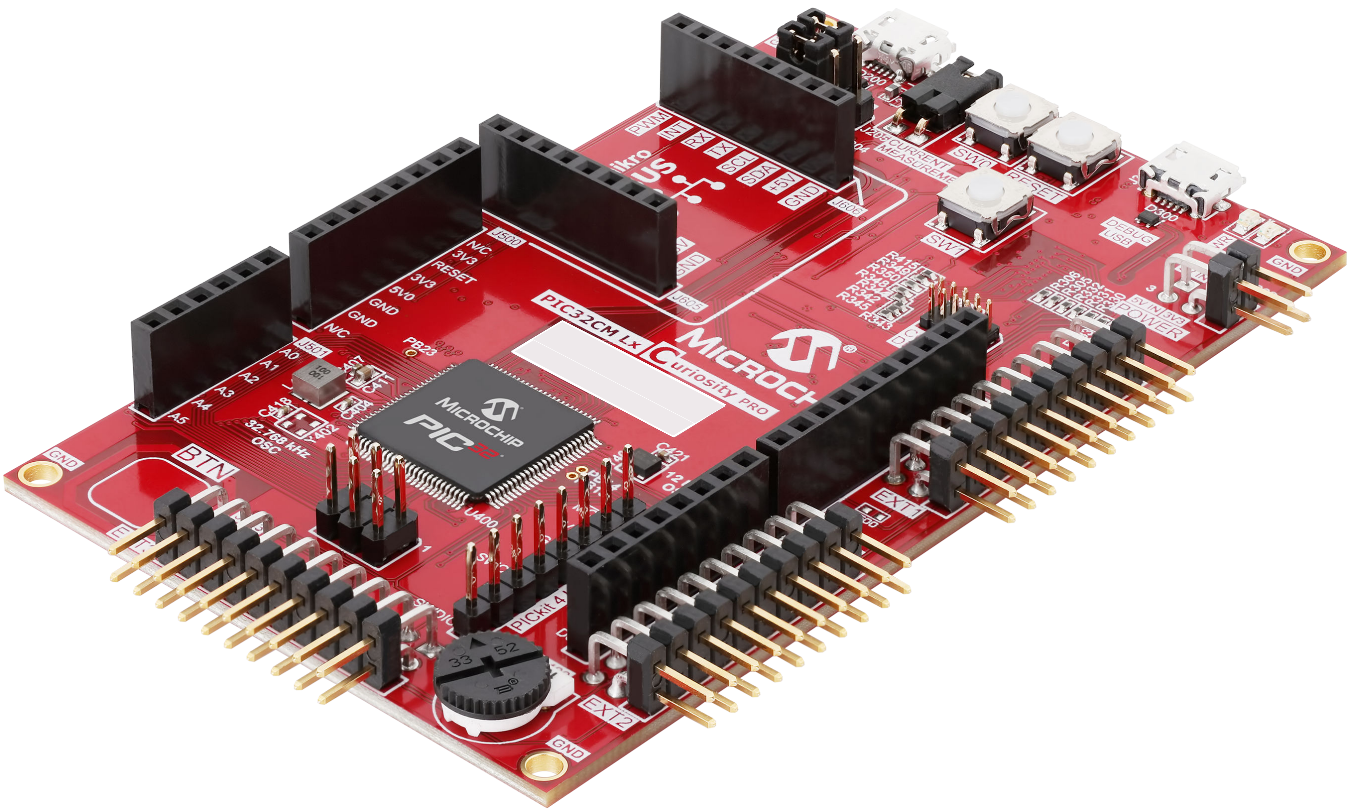
2.1 Hardware Requirements

2.1.1 Microchip PIC32CM LS00/LS60 Curiosity Pro

The Microchip PIC32CM LS00/LS60 Curiosity Pro evaluation kit is a hardware platform to evaluate the PIC32CM5164 LS00/LS60 100-pin microcontroller. Supported by the MPLAB X integrated development platform, the kit provides easy access to the features of the Microchip PIC32CM5164 LS00/LS60 and explains how to integrate the device in a custom design.

The Curiosity 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 Curiosity Pro extension kits offer additional peripherals to extend the features of the board, and ease the development of custom designs. The following figure illustrates the features of the PIC32CM LS00/LS60 Curiosity Pro board.

Figure 2-1. PIC32CM LS00/LS60 Curiosity Pro Board



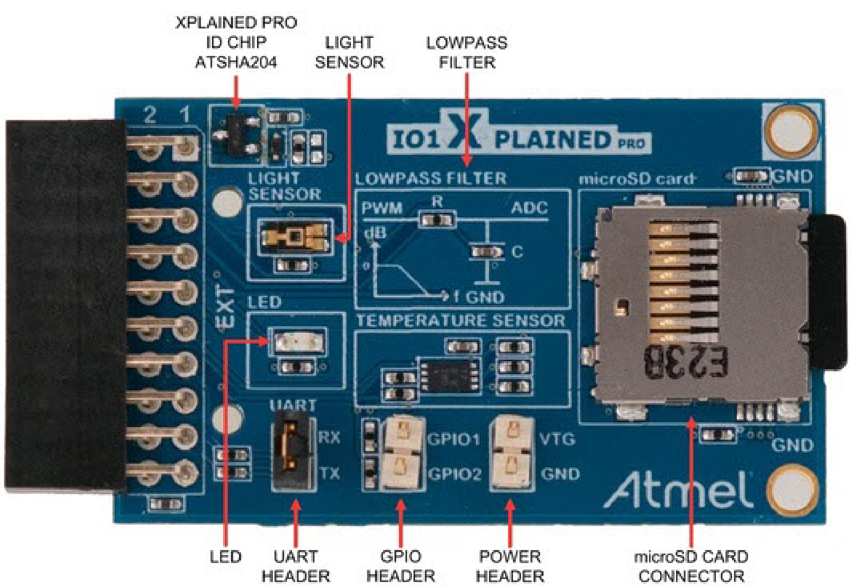
2.1.2 Microchip I/O1 Xplained Pro Extension Board

The Microchip I/O1 Xplained Pro extension board is a generic extension board for the Curiosity and Xplained Pro platform. It connects to any Xplained Pro standard extension header on any Curiosity or 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 Curiosity or 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
- **TWI**
  - AT30TSE758 temperature sensor with EEPROM

Figure 2-2. I/O1 Xplained Pro Extension Board

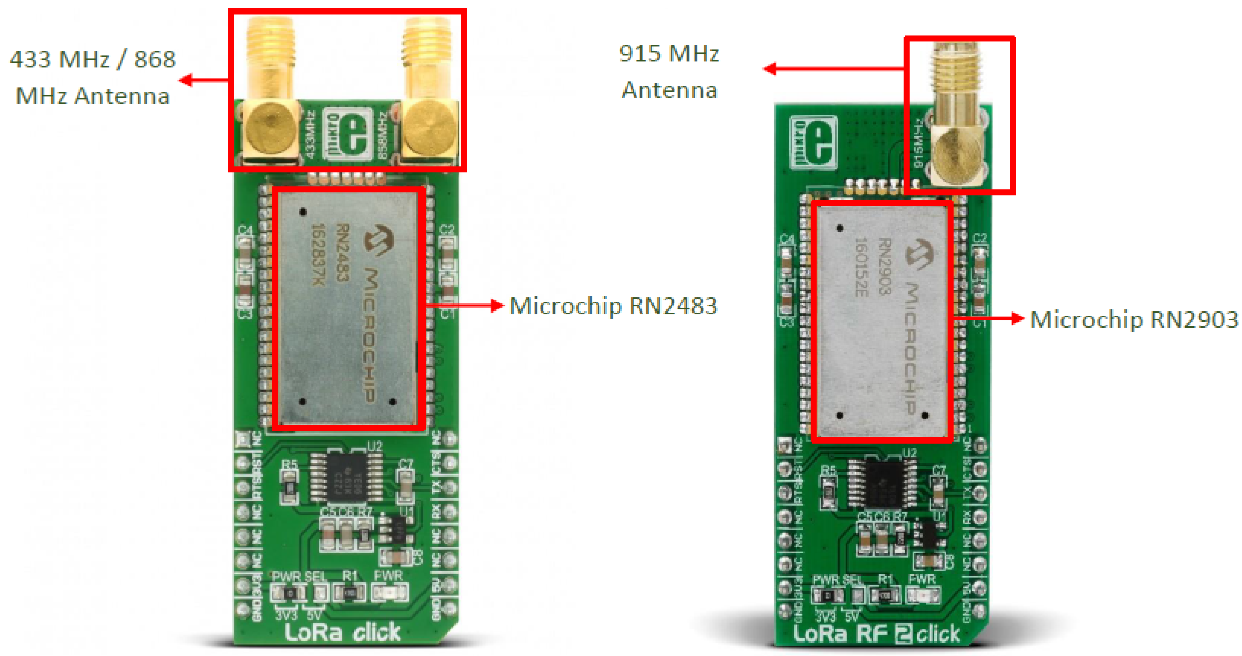


2.1.3 MikroElektronika LoRa Click Board (Includes Microchip RN2483 or RN2903)

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

Figure 2-3. RN2483 and RN2903 Boards



2.2 Software Requirements

2.2.1 MPLAB®X Integrated Development Environment

MPLAB X Integrated Development Environment (IDE) is an expandable, highly configurable software program that incorporates powerful tools to help you discover, configure, develop, debug, and qualify embedded designs or most of Microchip's microcontrollers and digital signals controllers. MPLAB X IDE works seamlessly with the MPLAB development ecosystem of software and tools.

Figure 2-4. MPLAB X IDE

Users can download MPLAB X IDE from the Microchip's website: www.microchip.com/mplab/mplab-x-ide.

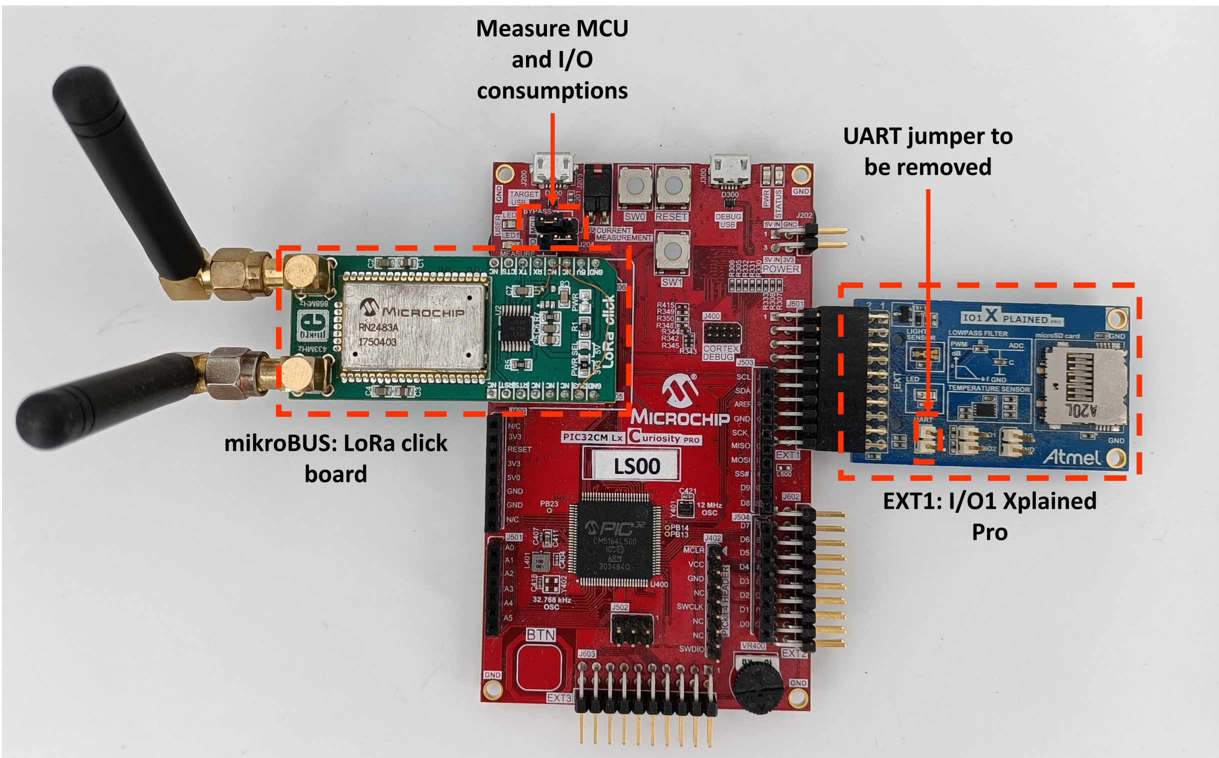
3. Demonstration Setup

The demonstration involves the following steps.

3.1 Hardware Setup

To setup the hardware, the user needs to perform these actions: Connect the I/O1 Xplained Pro extension board and LoRa click board to the PIC32CM LS00/LS60 Curiosity Pro board, and then set the board jumpers as shown in the following image. The demonstration hardware is now ready for evaluation.

Figure 3-1. Hardware Setup



Note: Ensure that the UART jumper on the I/O1 Xplained Pro board is removed, otherwise the PIC32CM LS00/LS60 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.

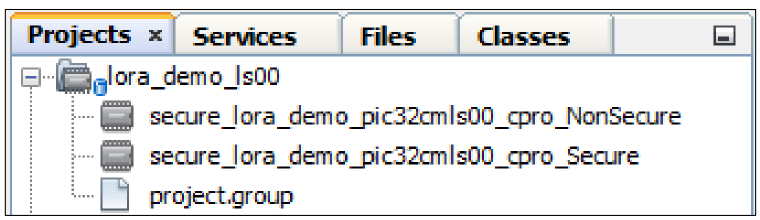
TTN-Compliant LoRa Gateway Setup
The following link provides a list of TTN compliant gateways and how to get started: www.thethingsnetwork.org/docs/gateways/.

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.

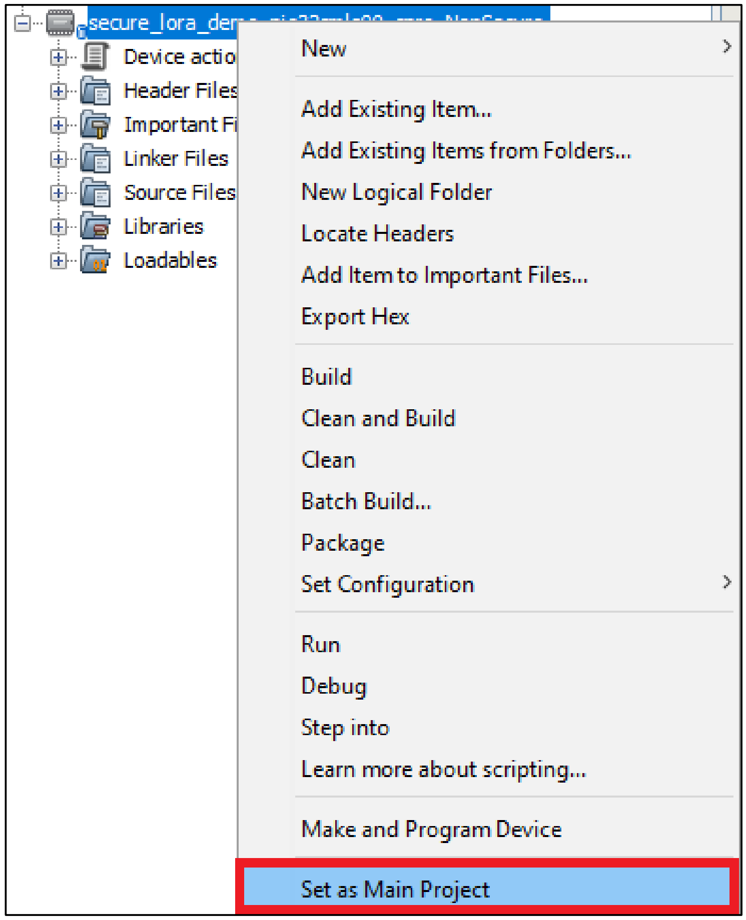
3.3 Software Setup

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

1. Open MPLAB X IDE.
2. Click **Projects** tab, and then expand the `lora_demo_ls00 (or lora_demo_ls60)` project group tree.
   Figure 3-2. MPLAB X Project Tree
   

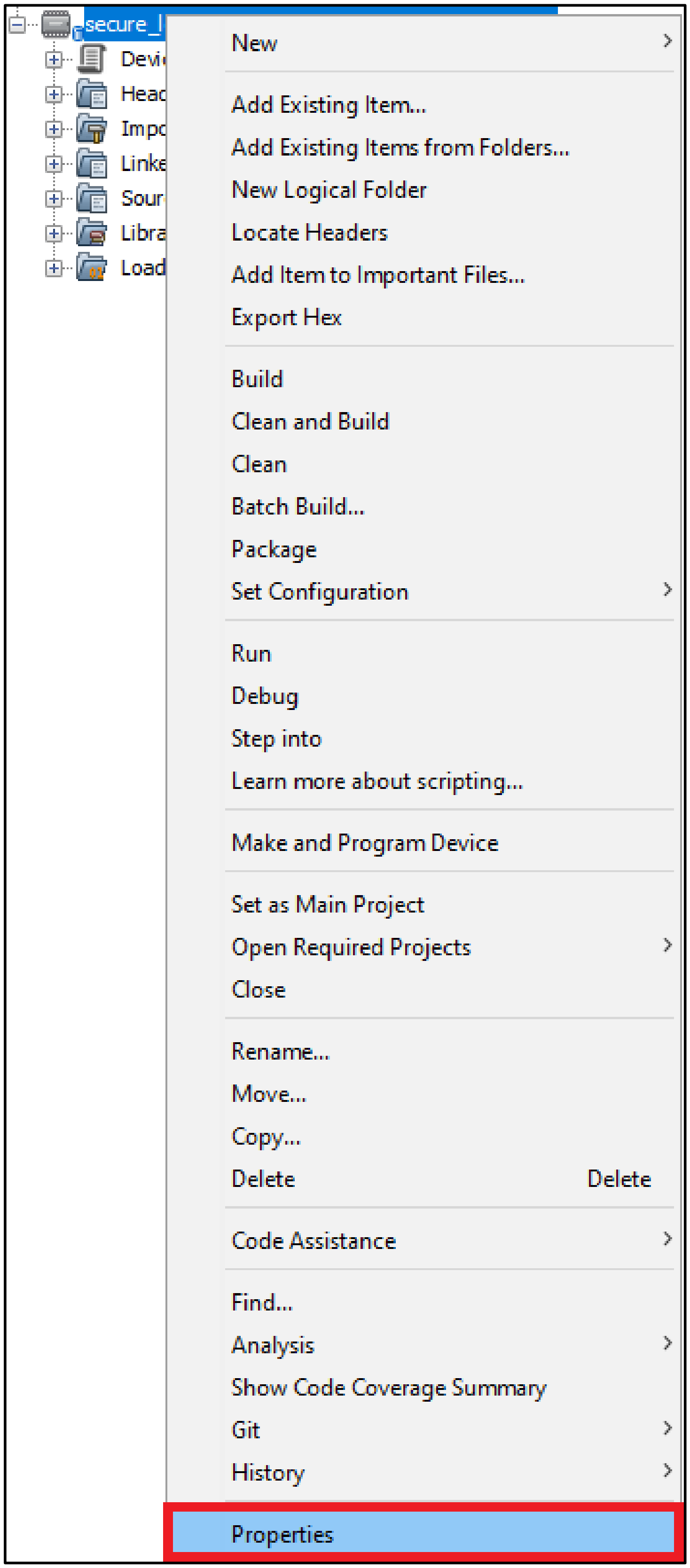
3. Double-click the `secure_lora_demo_pic32cmls00_cpro_NonSecure` project to open it.
4. Left-click the `secure_lora_demo_pic32cmls00_cpro_NonSecure` project, and then select **Set as Main Project.**.

Figure 3-3. Set NonSecure as Main Project



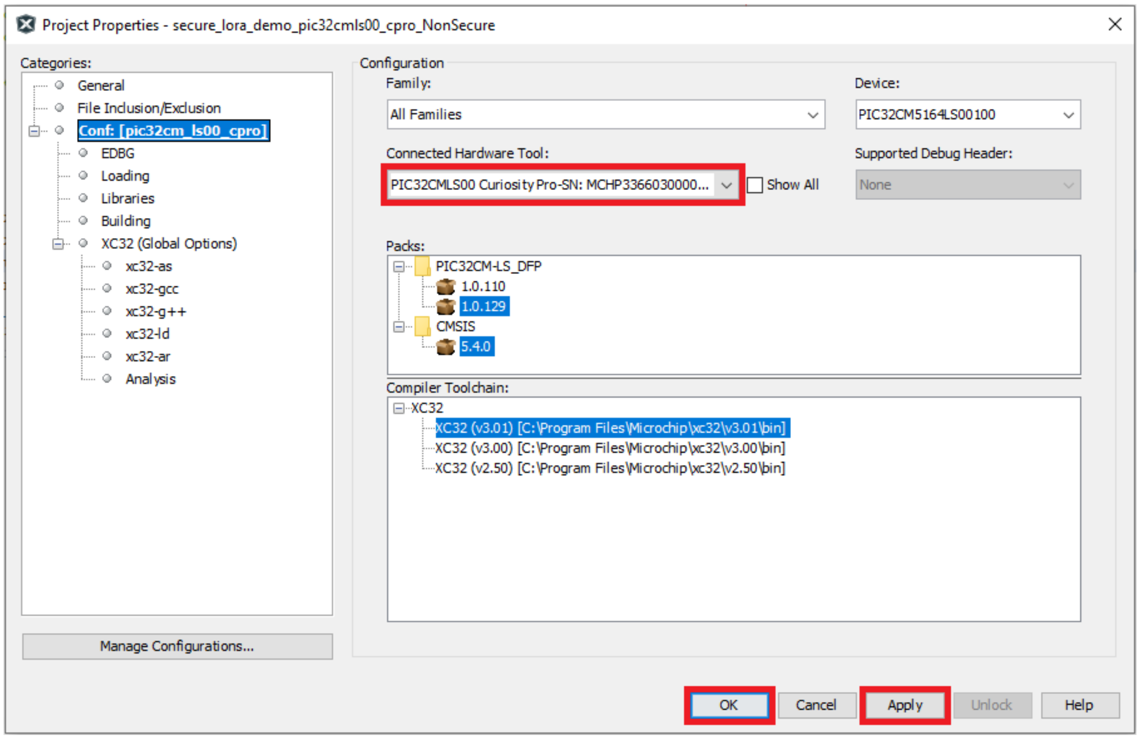
5. From the Solution Explorer panel, select *pic32cmls00_lora_demo_NonSecure* > *Properties*.

Figure 3-4. NonSecure Project Properties



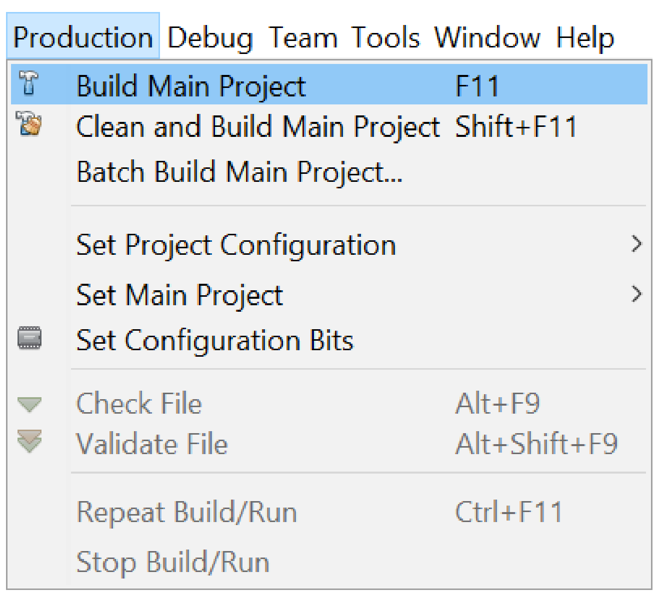
6. Select the connected board EDBG as the debugger/programmer.

Figure 3-5. NonSecure Project Properties Window



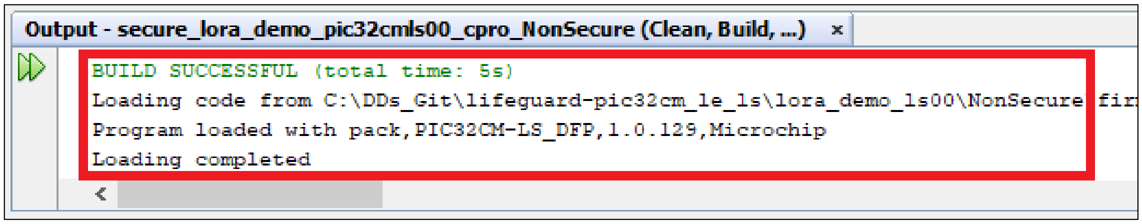
7. Click **Apply** to save the configuration, and then click **OK**.
8. From the toolbar, select *Production > Build Main Project* ( ), or F11 to build the demonstration.

Figure 3-6. Build Solution Menu



9. Check whether any error message is displayed on the MPLAB X IDE output window.

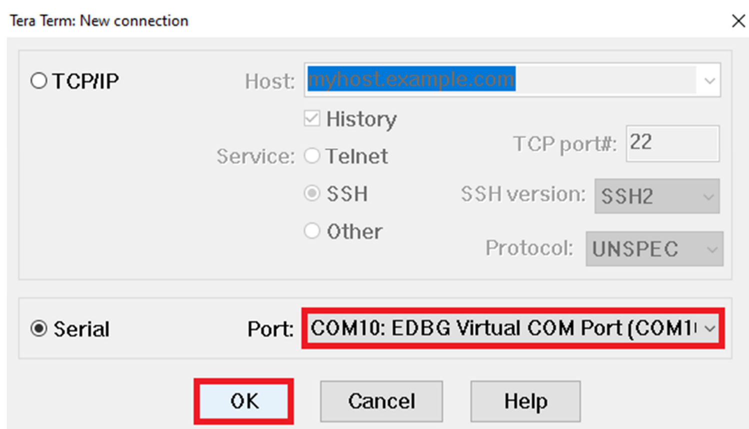
Figure 3-7. IDE Output Window



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

10. Flash the demonstration software on the hardware by clicking the ▷ button.
11. Open the Tera Term tool or any equivalent tool.
12. Choose the COM port number allocated to the PIC32CM LS00/LS60 Curiosity Pro, and then click **OK.**

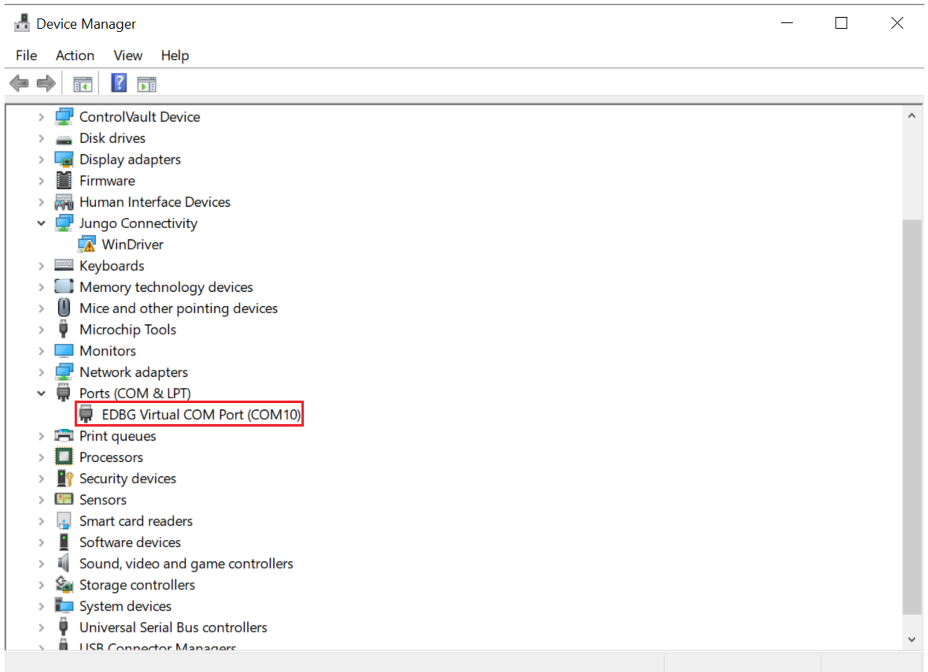
Figure 3-8. Tera Term New Connection Window



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

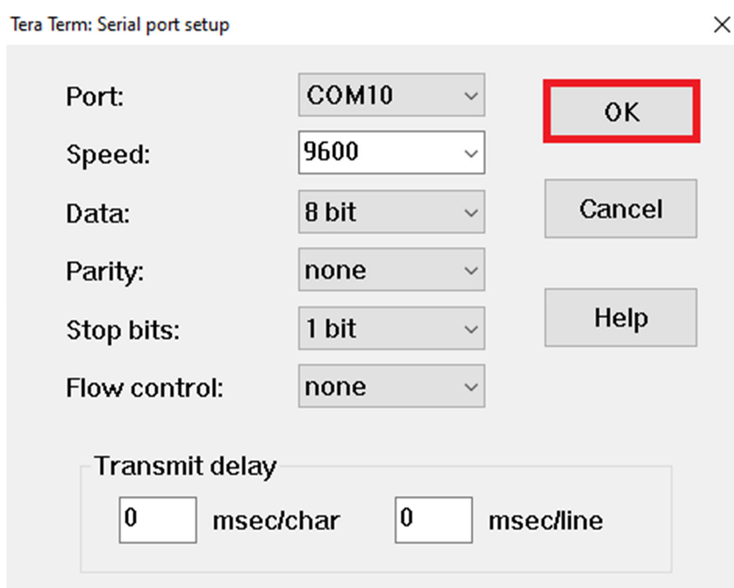
The Device Manager Window will be displayed, as shown below.

Figure 3-9. Windows Device Manager Window



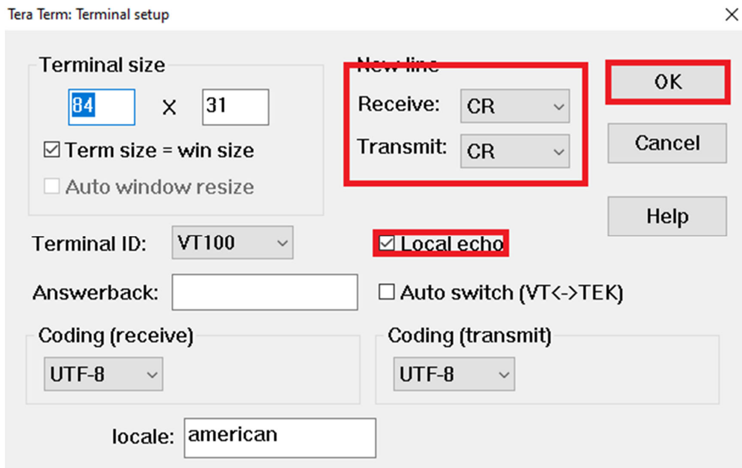
13. In the Tera Term Serial Port Setup dialogue, configure the serial interface using the parameters given below, and then click **OK.**

Figure 3-10. Tera Term Serial Port Setup



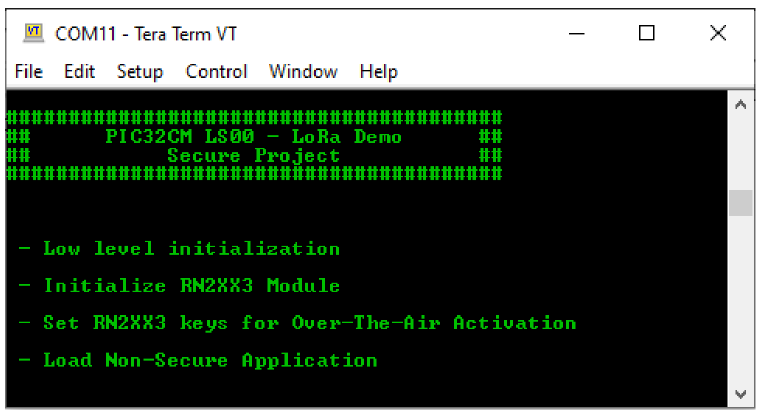
14. In the Tera Term Terminal Setup dialogue, choose values for Receive and Transmit, select “Local echo”, and then click **OK.**

Figure 3-11. Tera Term Terminal Setup



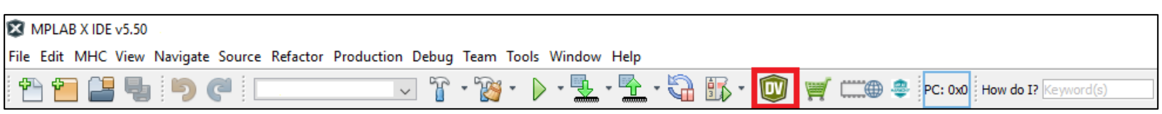
15. Reset the board by pressing the **RESET** button.
The demonstration will start by displaying the below messages on the terminal window. The displayed messages will depend on the different application keys have been provisioned on the device. Refer to Demonstration Description for additional information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

Figure 3-12. Terminal Window Displaying the Message



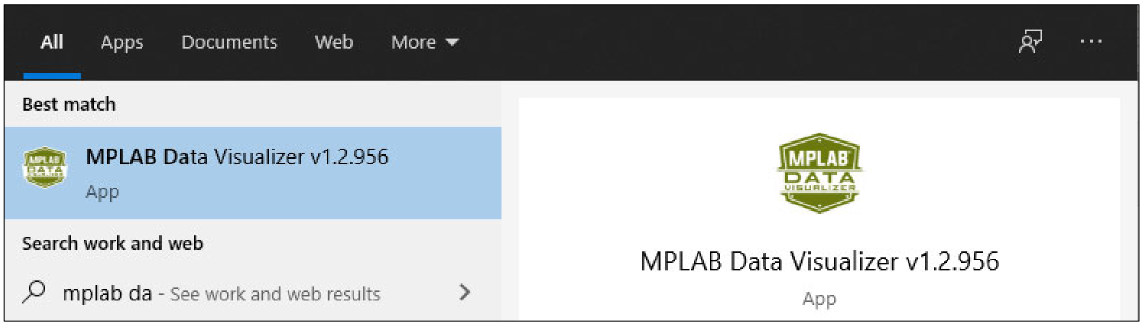
16. Open MPLAB Data Visualizer in MPLAB X IDE by clicking  (the Data Visualizer icon) on the toolbar.

Figure 3-13. MPLAB Data Visualizer Icon



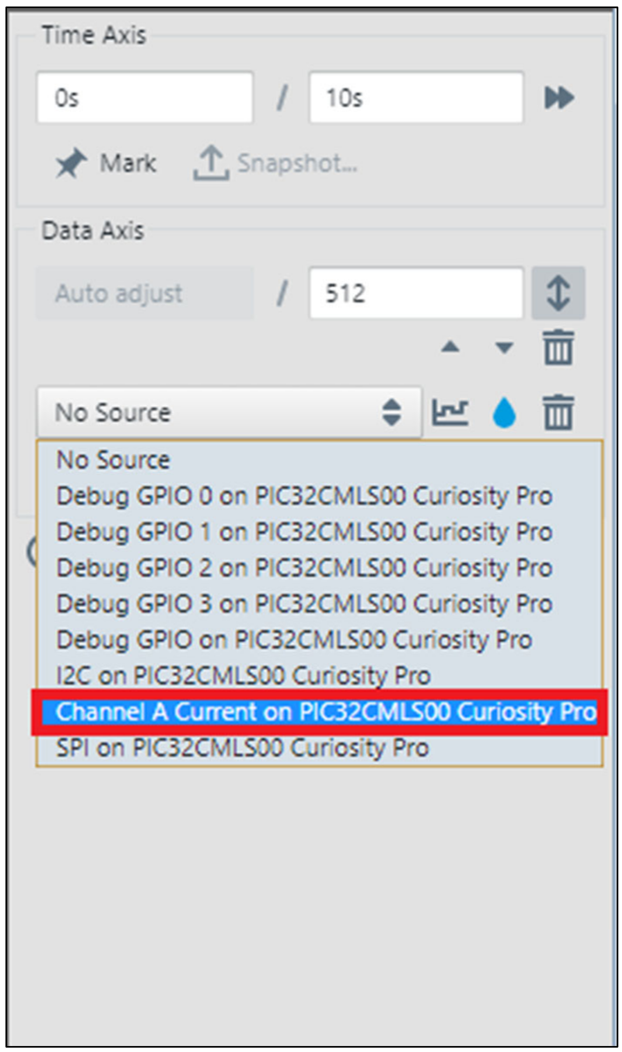
Note: The MPLAB Data Visualizer also has a standalone version that can be used instead of its MPLAB X IDE plugin version.

Figure 3-14. MPLAB Data Visualizer



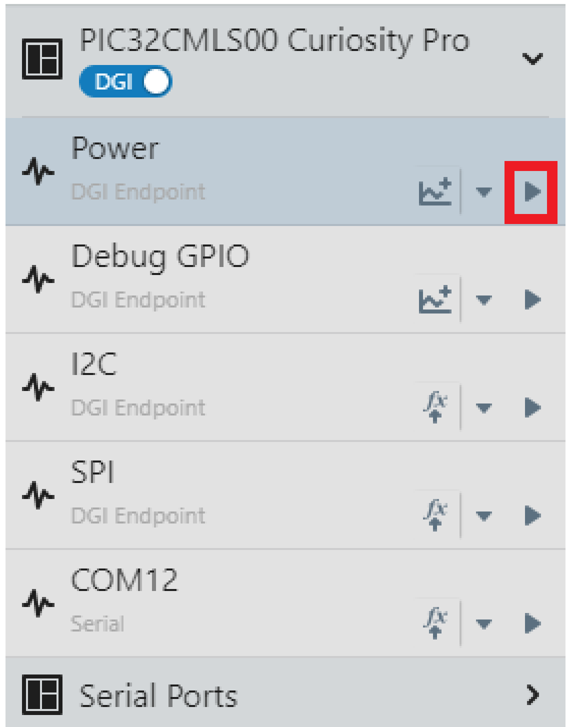
17. In the PIC32CMLS00 Curiosity Pro, select **Channel A Current** as the source of the opened time plot.

Figure 3-15. Time Plot Configuration: Channel A Current on the PIC32CMLS00 Curiosity Pro Selection



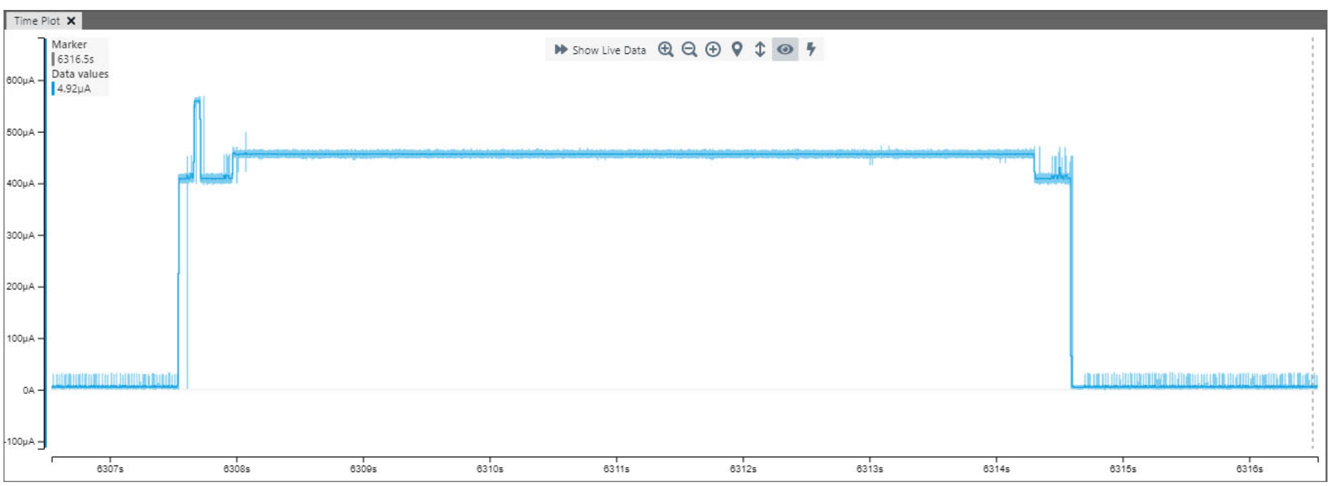
18. Click (play icon) in the Power box to display the power consumption details.

Figure 3-16. Power DGI Endpoint Display Start



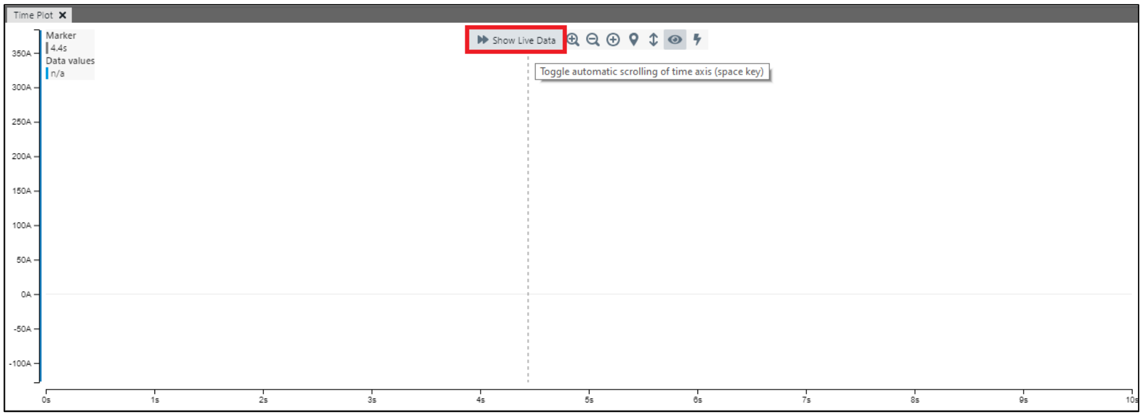
19. The Data Visualizer will display the PIC32CM LS00/LS60 LoRa Node power consumption details.

Figure 3-17. Data Visualizer Showing Power Consumption



Note: On the time plot window, if data is not appeared, click **Show Live Data**.

Figure 3-18. Time Plot Configuration: Show Live Data



4. Demonstration Description

4.1 Applications Overview

The user can evaluate the standard TrustZone for 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.

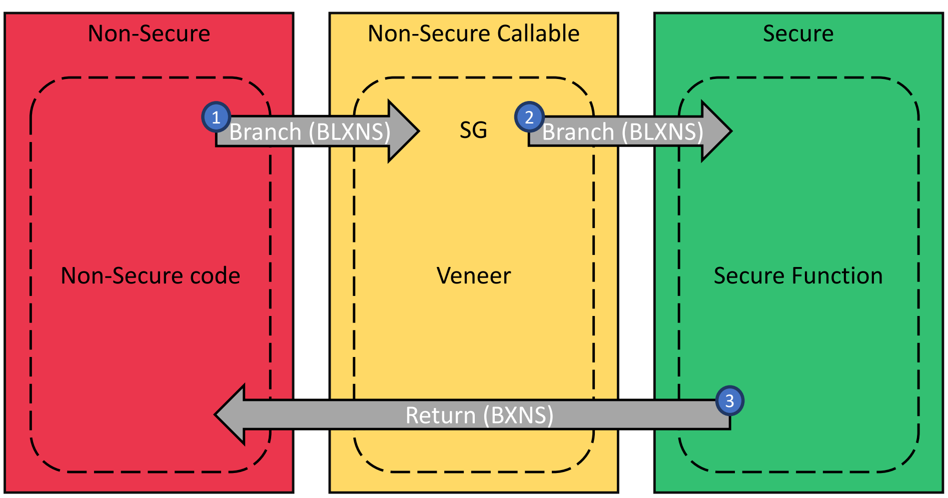
Figure 4-1. Non-Secure Application Code

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 PIC32CM LS00/LS60 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)

```
/* *********** 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_durartion" milliseconds
- PARAMETERS :  char* sleep_duration
- RETURN :      None
------------------------------------------------------ */
extern 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
------------------------------------------------------ */
extern 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
------------------------------------------------------ */
extern uint32_t nsc_rn2xx3_transmit_with_ACK(char* port ,char* payload,char* rx_port,char* rx_payload);
```

Figure 4-4. Gateway (Veneers)

```
/* -----------------------------------------------------
- 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
------------------------------------------------------ */
extern void nsc_periph_clock_init(uint32_t pch_id, uint32_t gclk_source);


/* *********** Secure PIC32CM LSxx Low_Power NSC functions *********** */

/* -----------------------------------------------------
- NAME :         void nsc_secure_enter_standby (void)
- DESCRIPTION : (Secure Gateway) Put the PIC32CM LSxx in Standby mode (Low-power)
- PARAMETERS :  None
- RETURN :      None
------------------------------------------------------ */
extern 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
------------------------------------------------------ */
extern void nsc_secure_console_puts (char *string);

/* ************ Secure PIC32CM LSxx Delay function ************ */

extern void nsc_SYSTICK_DelayMs ( uint32_t delay );
```

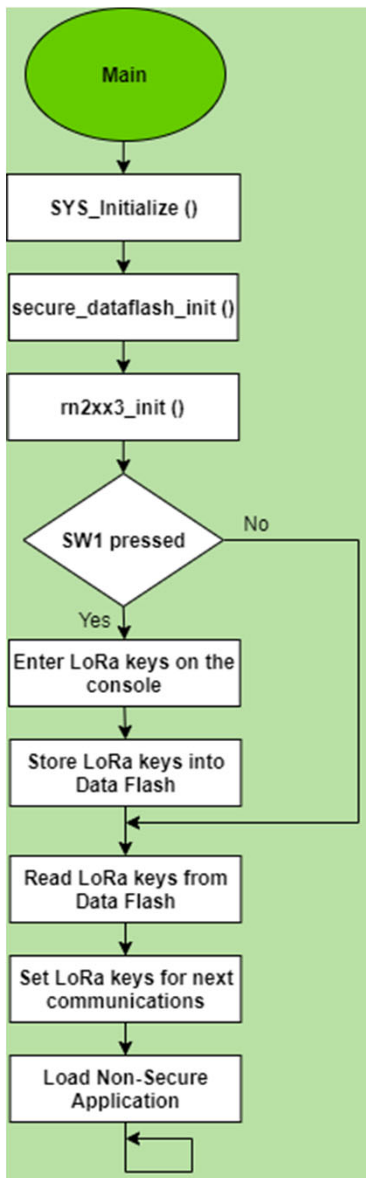
4.2 PIC32CM LS00/LS60 LoRa Demonstration: Secure Project

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 PIC32CM LS00/LS60, which are isolated from the Non-Secure software with the help of the Cortex-M23 TrustZone for ARMv8-M. It is in charge of performing the following tasks in the application:

- Low-level system settings.
- Providing the library and associated veneers (APIs) 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:

Figure 4-5. Secure Project Main Routine



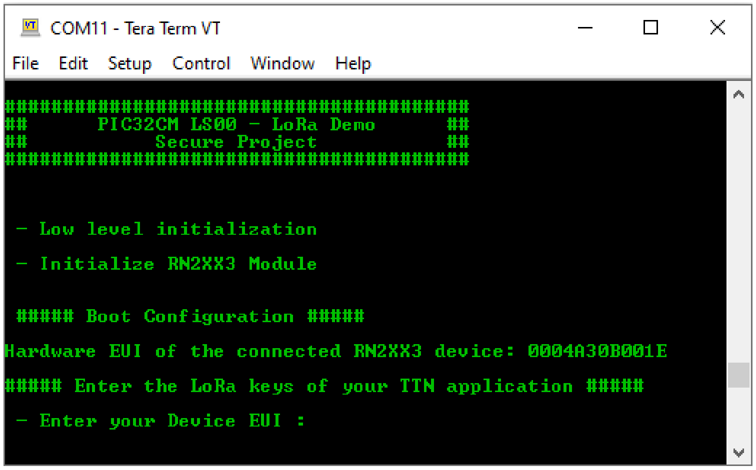
The above flowchart illustrates the different cases available in the Secure project. To start the key provisioning operation, the user must press the SW0 button during the Secure project start until the LoRa keys configuration screen appears for key provisioning. The LoRa keys to be provisioned must 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 the TTN and send data. These keys are accessible on the TTN after the device is registered.

4.2.1 Keys Not Provisioned

This section describes how key provisioning is realized during boot configuration. If LoRa keys are stored in the Data Flash, skip this section and proceed to Keys Provisioned.

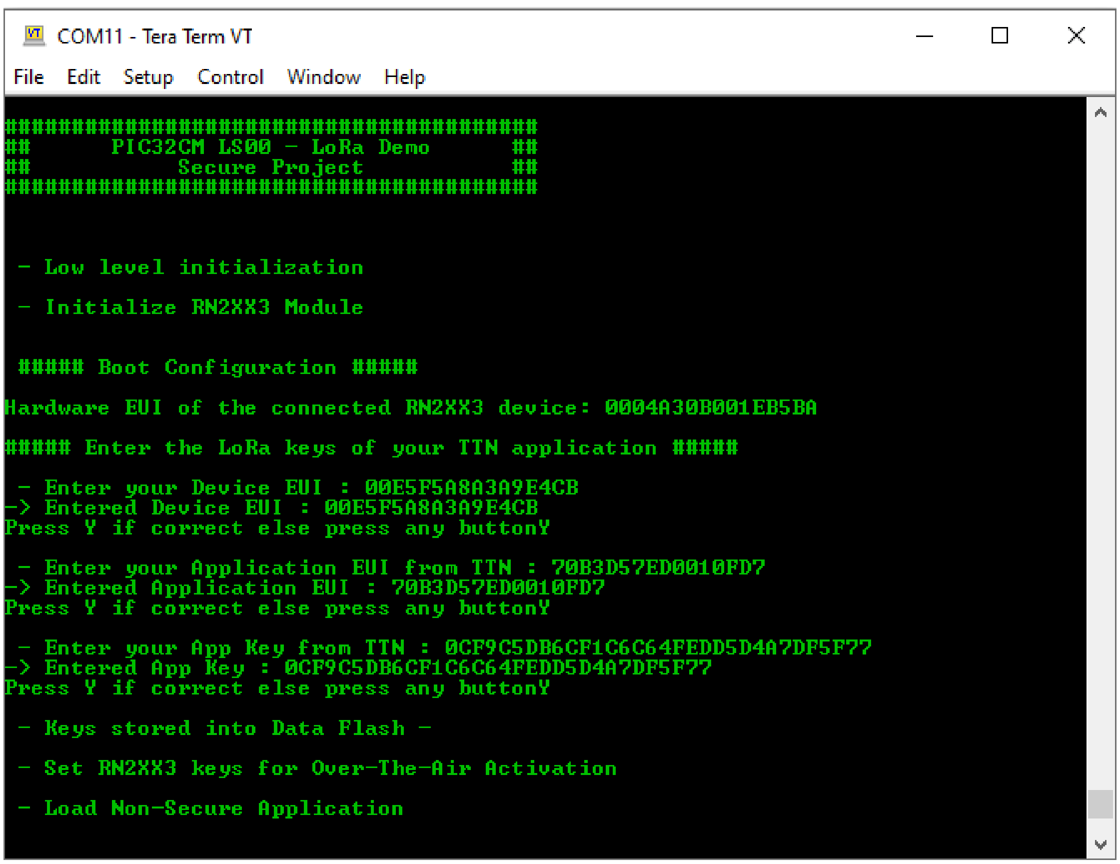
The following figure shows the messages displayed by the console after the LoRa keys configuration is entered by pressing the SW0 button after resetting the PIC32CM LS00/LS60 LoRa Node.

Figure 4-6. Console Displaying the Key Provisioning Entry Message



The user can enter DevEUI, AppEUI, and AppKey keys to be stored in Data Flash as shown in the following figure.

Figure 4-7. Console Displaying the Key Provisioning Process

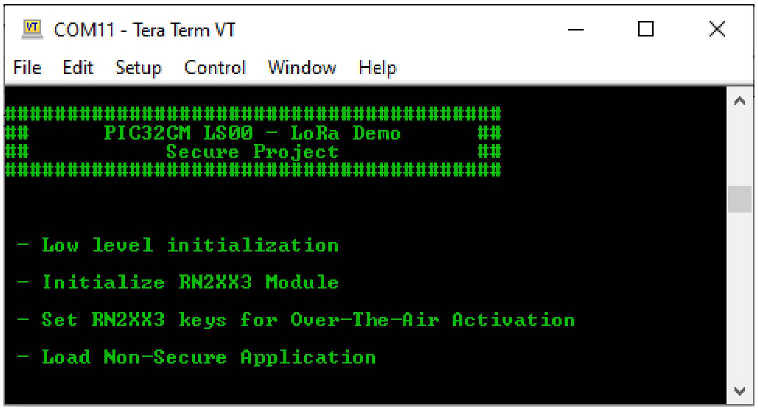


Keys are stored into the Data Flash and are set in the RN2483 or RN2903 module for the OTAA procedure.

4.2.2 Keys Provisioned

The following image shows the information displayed by the console after a PIC32CM LS00/LS60 reset when the keys are provisioned:

Figure 4-8. Console Showing Standard Boot Message



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, 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.

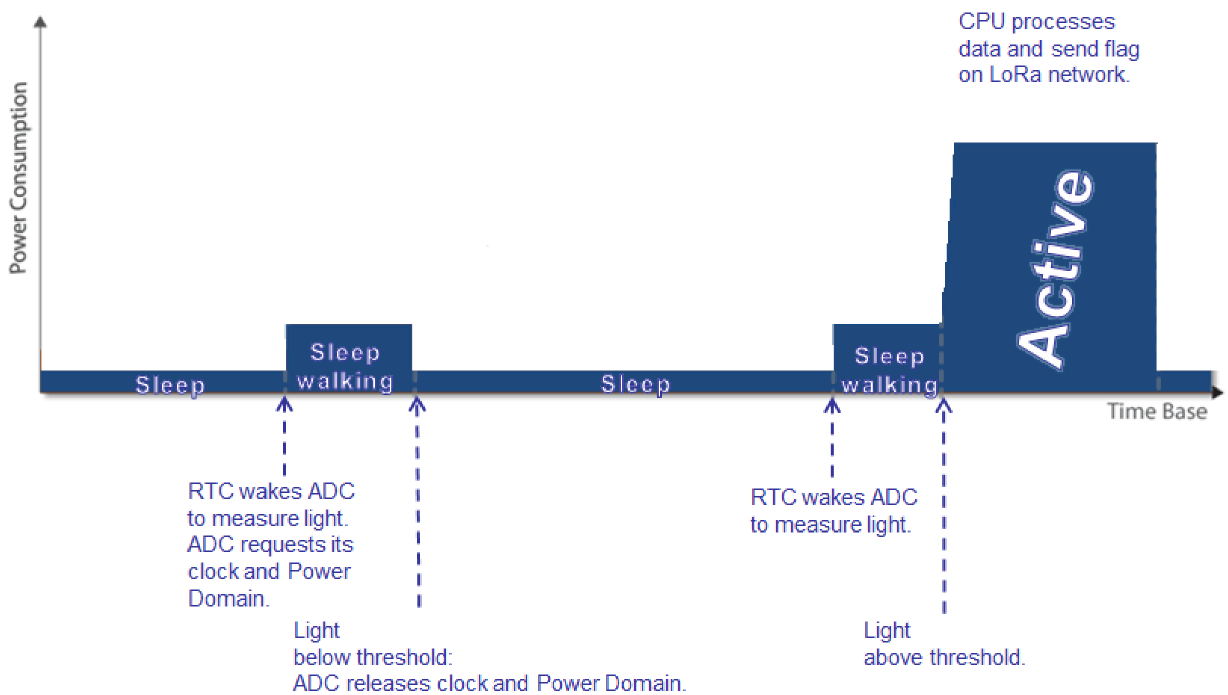
4.3 PIC32CM LS00/LS60 LoRa Demonstration: Non-Secure Project

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 PIC32CM LS00/LS60 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 PIC32CM LS00/LS60 and the RN2483 or RN2903, the devices are placed in low-power mode when an ambient luminosity change is not detected.

When the PIC32CM LS00/LS60 is in Standby mode, it uses SleepWalking feature to reduce the power consumption. SleepWalking gives the capability for the PIC32CM LS00/LS60 to wake up temporarily and asynchronously a peripheral without waking up the CPU.

The following figure illustrates the demonstration behavior and the benefits of SleepWalking on power consumption:

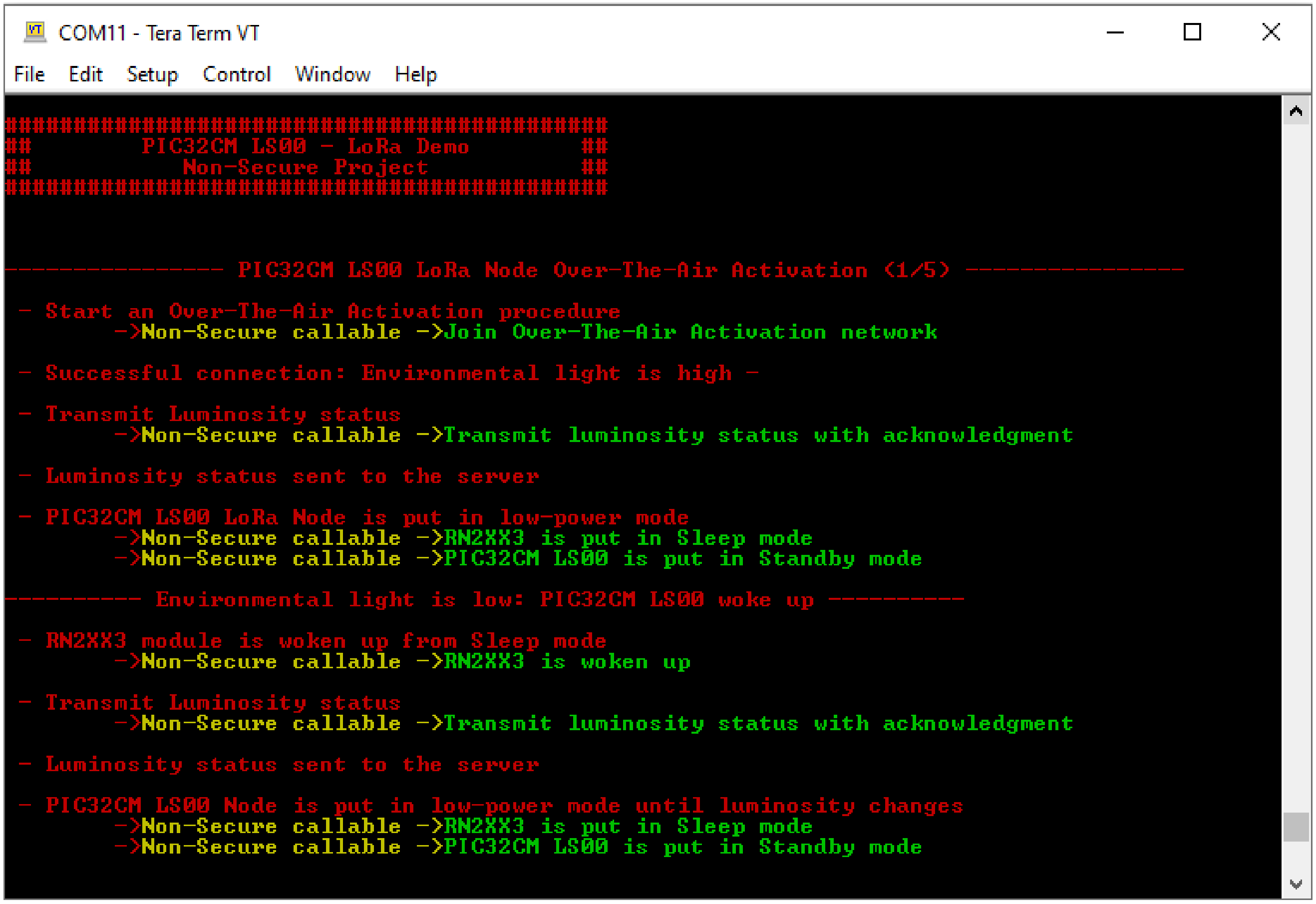
Figure 4-9. SleepWalking on Power Consumption



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 PIC32CM LS00/LS60 to remain in Standby mode for a long time.

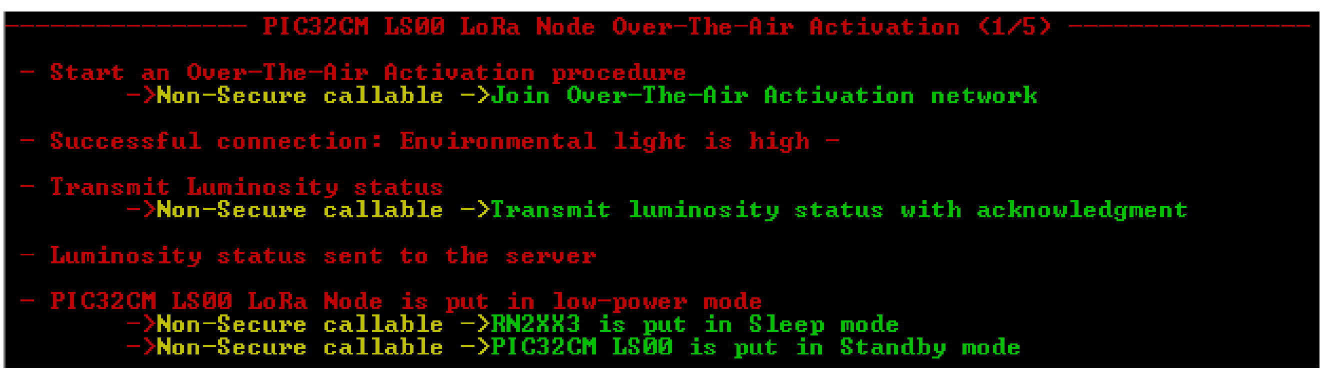
The following figure shows the message displayed on the console by the Non–Secure application in successful connection case:

Figure 4-10. Non-Secure Application Console



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, and then the RN2483 or RN2903 module and the PIC32CM LS00/LS60 are placed in low-power mode:

Figure 4-11. 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 PIC32CM LS00/LS60 are put in Low-Power mode:

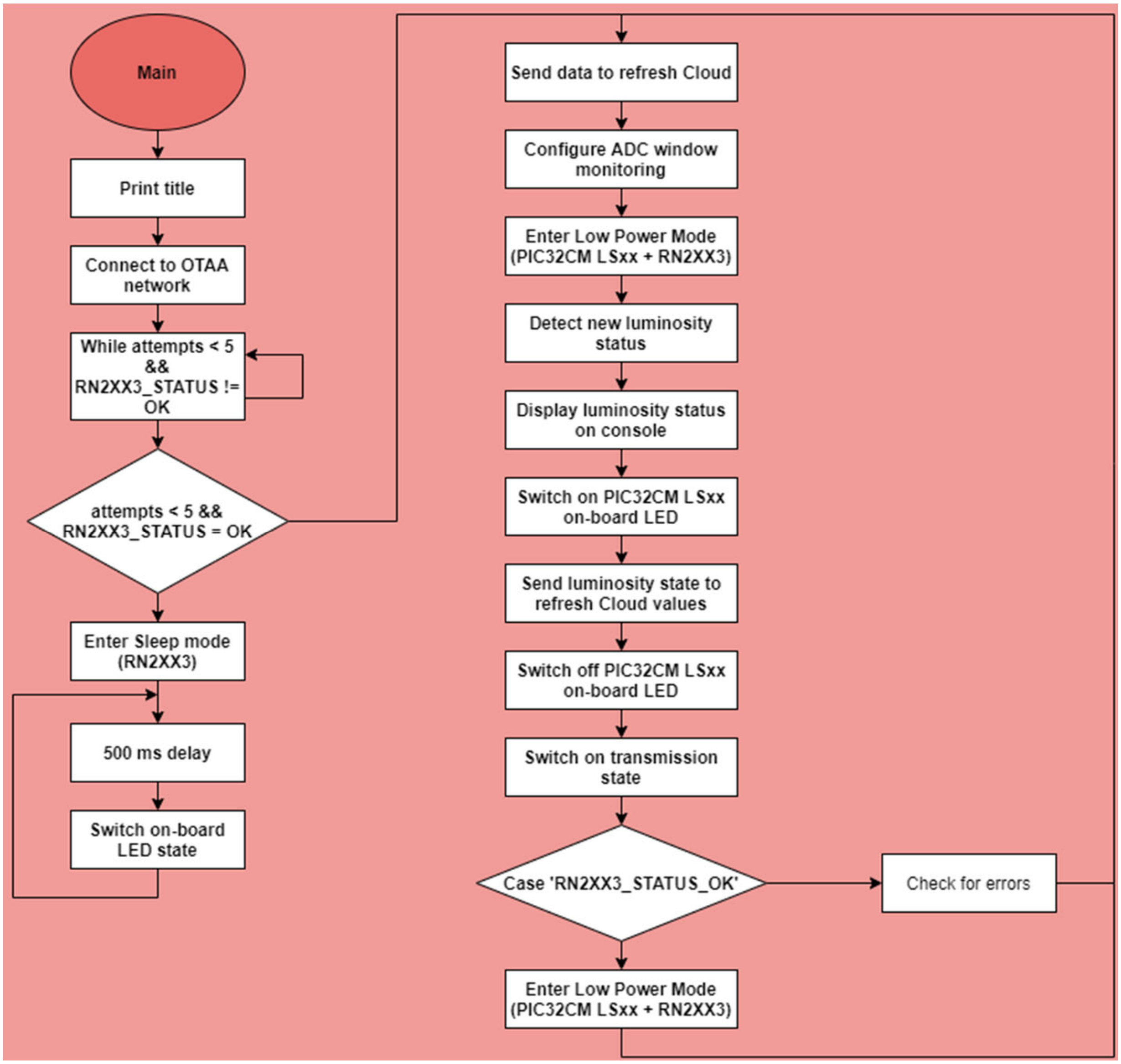
Figure 4-12. LoRa Transaction in Active Mode Message

```
---------- Environmental light is low: PIC32CM LS00 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

 - PIC32CM LS00 Node is put in low-power mode until luminosity changes
        ->Non-Secure callable ->RN2XX3 is put in Sleep mode
        ->Non-Secure callable ->PIC32CM LS00 is put in Standby mode
```

Note: Sometimes the luminosity status does not appear on the Cayenne dashboard, in that case, users must refresh their browser.

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

Figure 4-13. Non-Secure Application Flow Chart

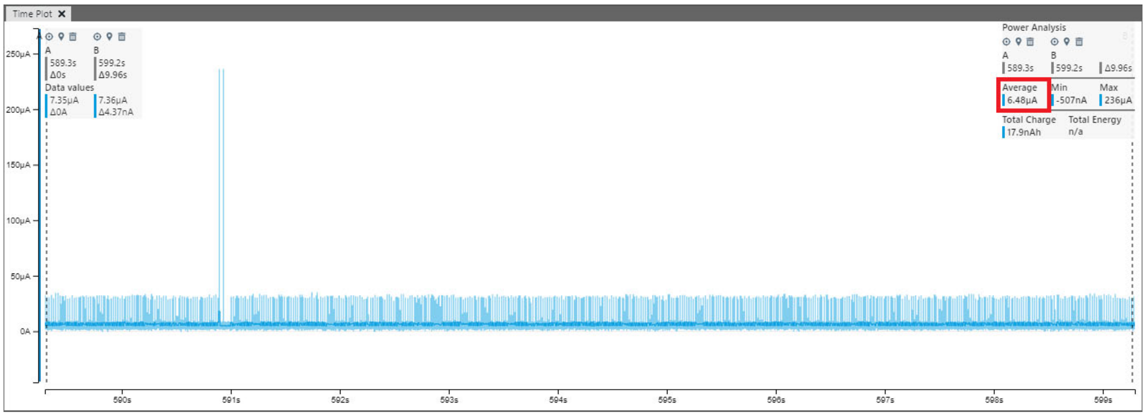


4.4 Power Consumptions of the LoRa Node

When the application is running, the dynamic current consumption of the whole application can be measured with the MPLAB Data Visualizer plugin from MPLAB X IDE (or it's standalone version). 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 I/O1 Xplained Pro extension board

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

Figure 4-14. PIC32CM LS00 IoT Node Power Consumption in Standby Mode



Note: The measured power consumption in Standby mode is a bit less than 6.5 µA.

Note: Equivalent power consumption are expected and measured on the PIC32CM LS60 LoRa IoT Node. To reduce the power consumption, some modifications are made on the RN2483 or RN2903 module and the I/O1 Xplained Pro extension board. Refer to the Hardware Modifications section for additional information about these modifications.

5. System Resources and Software Project Configuration

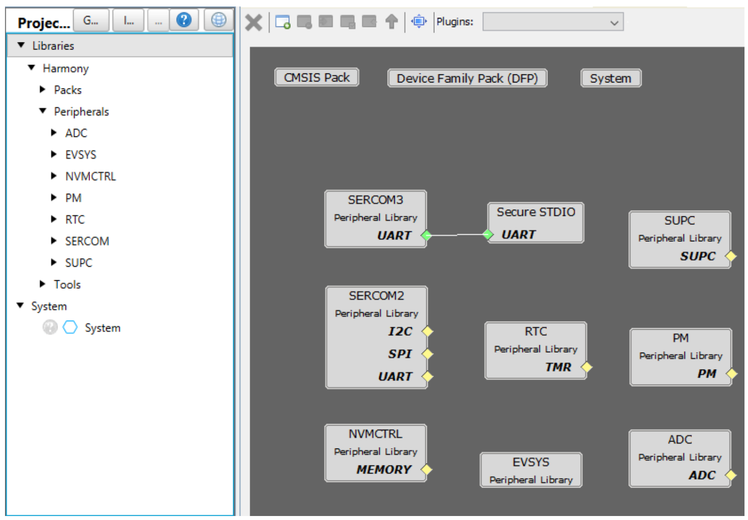
The demonstration application is built on MPLAB Harmony v3. After opening the demonstration project under MPLAB X IDE, the configuration used to generate the project under the MPLAB Code Configurator for Harmony v3 can be viewed by clicking the MCC icon, which is available in the MPLAB X IDE toolbar.

5.1 MPLAB Code Configurator (MCC) for MPLAB Harmony v3 View

5.1.1 MCC Project Setup

The Secure and Non-Secure projects are built on top of the following drivers and middleware:

Figure 5-1. MCC Project Graph and Active Components

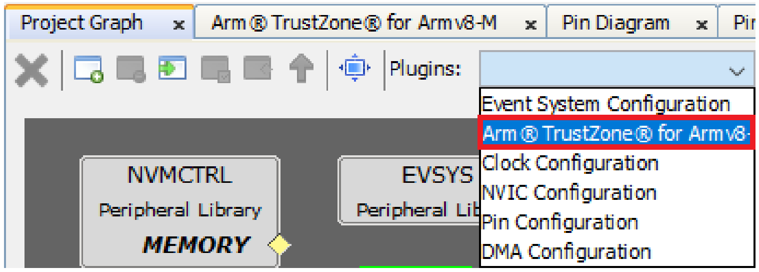


- **SERCOM3 (with Secure STDIO):** Used for displaying application output information on the terminal.
- **SERCOM2:** Interface for the RN2483 or RN2903 USART. This is used to send and receive commands and data to and from the RN2483 or RN2903 module.
- **ADC:** Used to measure environmental light from the I/O1 Xplained Pro that embeds a light sensor.
- **RTC:** Used to generate an event 10s after the start of the timer count.
- **NVMCTRL:** Used to generate an event 10s after the start of the timer count.
- **EVSYS:** Used to generate an event 10s after the start of the timer count.

Note: The Secure SYSTICK is also enabled to generate delays for synchronization in some commands. For additional information, go through the 'System' in the Project Graph (or Active Components) window.

The Project Graph and Active Components views show all the peripherals used in the IoT node application, including Secure and Non-Secure applications. The memory configuration and the security attribution for peripherals are assigned through the Arm® TrustZone® for Armv8-M tool, which is available in the MCC toolbar: select *Plugins > Arm® TrustZone® Armv8-M*.

Figure 5-2. MCC TrustZone® for Armv8-M Tool



The Arm® TrustZone® for Armv8-M manager sets the memory configuration of the PIC32CM LS00/LS60 device and the configuration of the peripherals.

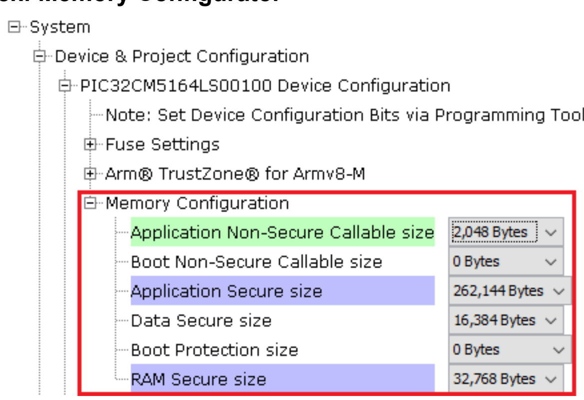
5.1.1.1 Arm® TrustZone® for Armv8-M Manager: Memory Configuration

The memory configuration of the device is set as follows for the PIC32CM LS00/LS60 IoT Node demonstration:

Figure 5-3. Arm® TrustZone® for Armv8-M Manager: Memory Configurator

The memory configuration is also available by following these steps: In the Project Graph Window, click and expand *System > Device & Project Configuration > Memory Configurator*.

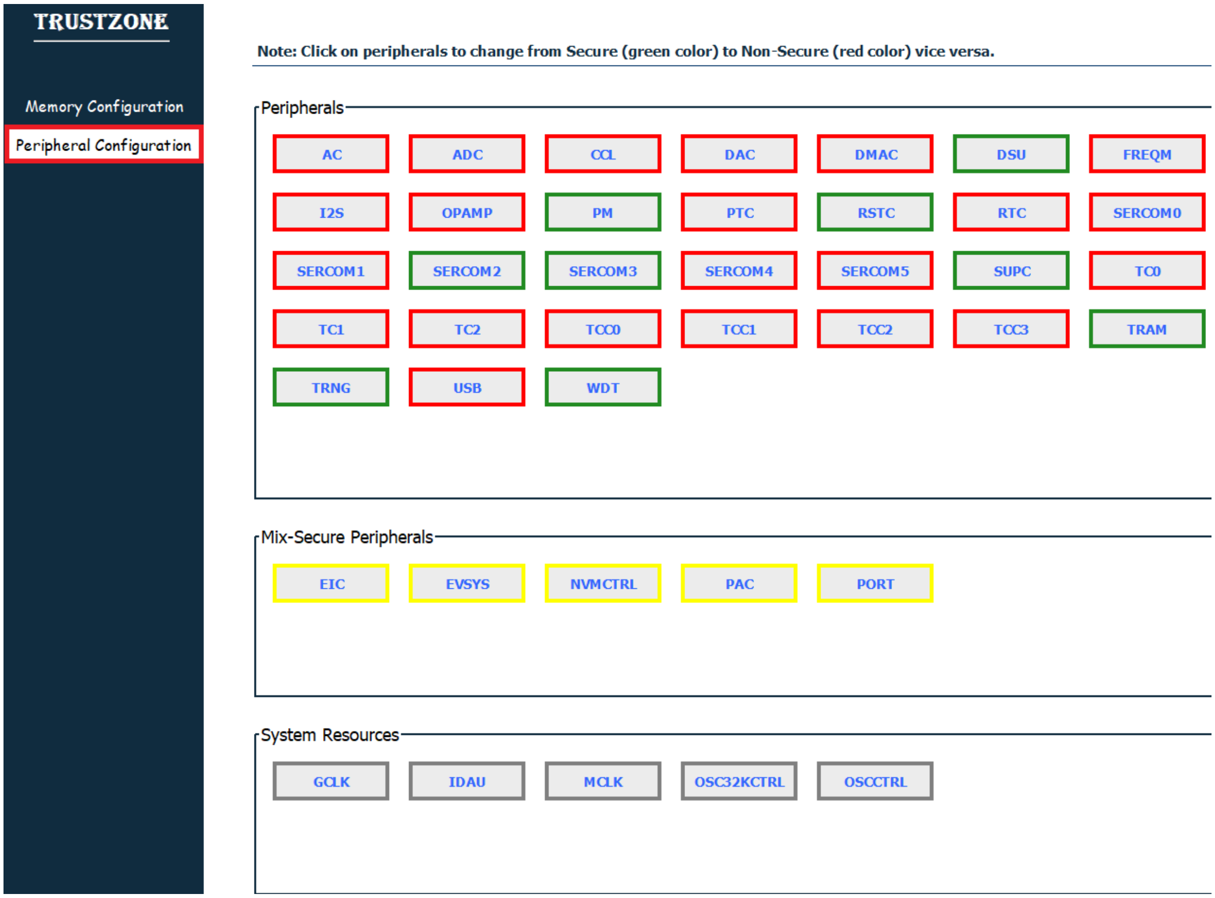
Figure 5-4. MCC System Block: Memory Configurator



5.1.1.2 Arm® TrustZone® for Armv8-M Manager: Peripheral Configuration

The peripherals configuration of the device is set as follows for the PIC32CM LS00/LS60 LoRa IoT Node demonstration:

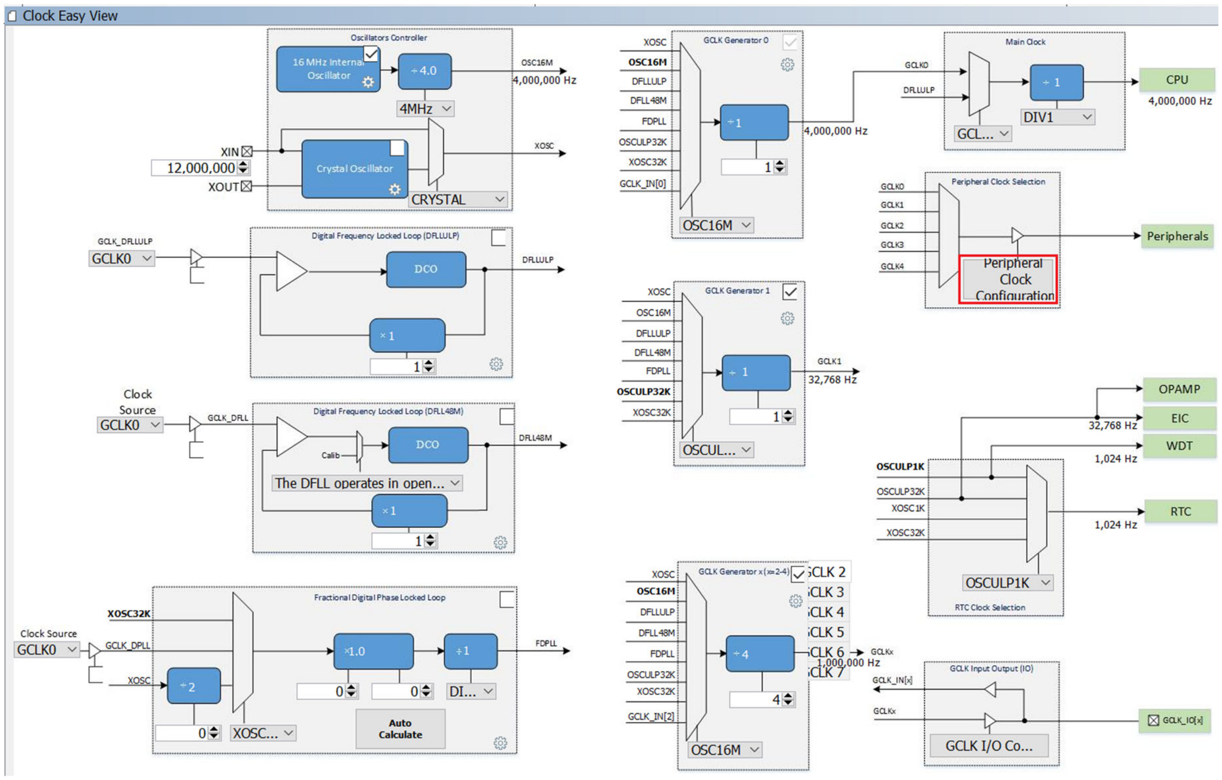
Figure 5-5. Arm® TrustZone® for Armv8-M Manager: Peripheral Configuration



5.1.2 MCC Project Clock Configuration

The clock configuration is available in the toolbar:

Figure 5-6. MCC Project Clock Configuration



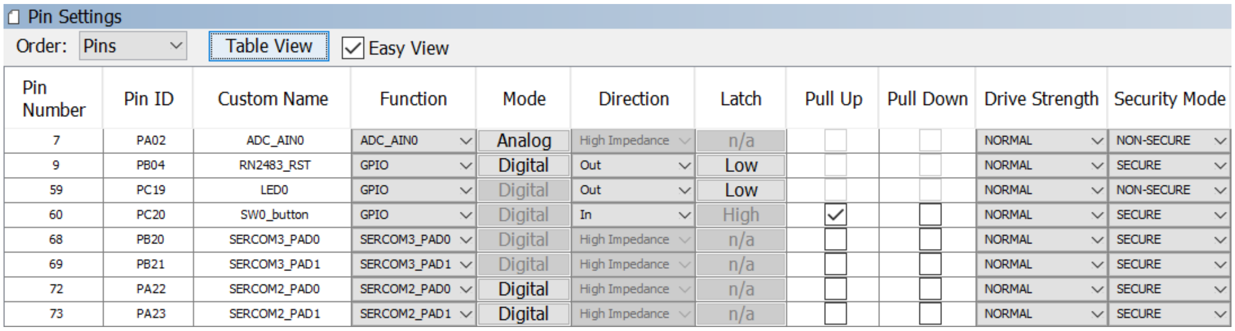
- The OSC16M is configured to run at 4 MHz and feeds the Generic Clock Generator 0 (GCLK0) and GCLK2. GCLK0 runs at 4 MHz and GCLK2 runs at 1 MHz.
- The OSCULP32K is configured to provide a 32 kHz source clock to the GCLK1.
- The GCLK is used to route oscillators to the peripherals. The GCLK0 provides a 4 MHz source clock to the CPU, EVSYS, and active SERCOMs. The GCLK1 provides a 32 kHz clock to SERCOMs' slow clocks and GCLK2 clocks the ADC peripheral.

Note: For additional information on peripherals source clocks, select **Peripheral Clock Configuration** block, as highlighted in the figure above.

5.1.3 MCC Project Pin Configuration

The pin configuration is accessible in the MCC toolbar. The following pins are set for the LoRa IoT Node application:

Figure 5-7. Pin Settings



Pin Settings

Order: Pins | Table View | Easy View

Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength	Security Mode
7	PA02	ADC_AIN0	ADC_AIN0	Analog	High Impedance	n/a			NORMAL	NON-SECURE
9	PB04	RN2483_RST	GPIO	Digital	Out	Low			NORMAL	SECURE
59	PC19	LED0	GPIO	Digital	Out	Low			NORMAL	NON-SECURE
60	PC20	SW0_button	GPIO	Digital	In	High	✓		NORMAL	SECURE
68	PB20	SERCOM3_PAD0	SERCOM3_PAD0	Digital	High Impedance	n/a			NORMAL	SECURE
69	PB21	SERCOM3_PAD1	SERCOM3_PAD1	Digital	High Impedance	n/a			NORMAL	SECURE
72	PA22	SERCOM2_PAD0	SERCOM2_PAD0	Digital	High Impedance	n/a			NORMAL	SECURE
73	PA23	SERCOM2_PAD1	SERCOM2_PAD1	Digital	High Impedance	n/a			NORMAL	SECURE

- The PA02 and PC19 pins are set as Non-Secure pins, and they drive the ADC AIN0 Input and the LED0.
- The others are set as secure pins:
  - The PB04 pin is set as a GPIO Output Low to drive the RN2483 (or RN2903) reset pin.
  - The PC20 pin drives the on-board switch button SW0.
  - The PB20 and PB21 pins drive the Rx and Tx SERCOM3 pins to display information (and receive characters) to/from a terminal.
  - The PB22 and PB23 pins drive the SERCOM2 Rx and Tx pins to establish the communication channel between the RN2483 (or RN2903) board and the PIC32CM LS00/LS60 microcontroller.

5.2 Generated Projects Source Code

The Secure and Non-Secure project source codes are generated by the MCC using the previous configuration:

Figure 5-8. Secure Project Architecture

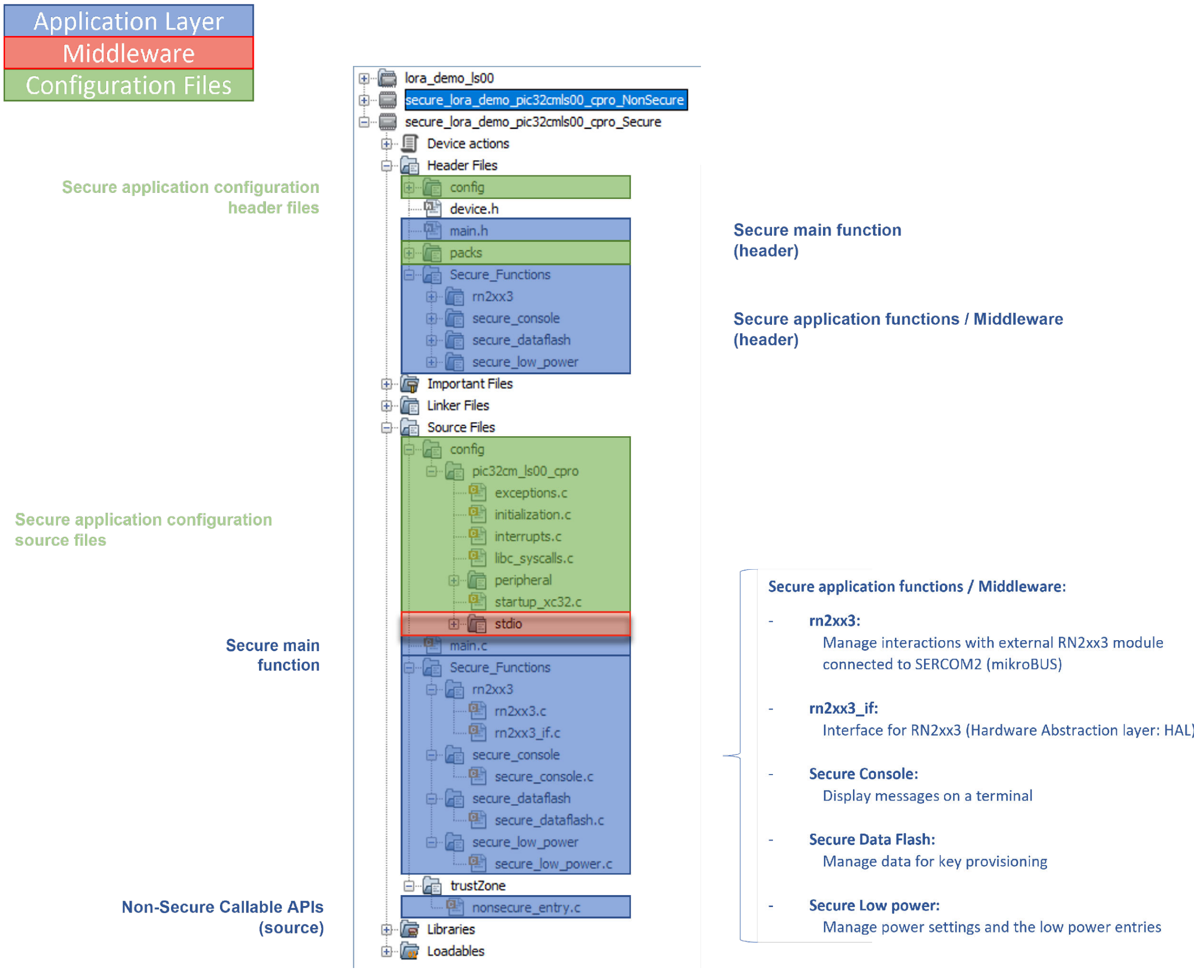
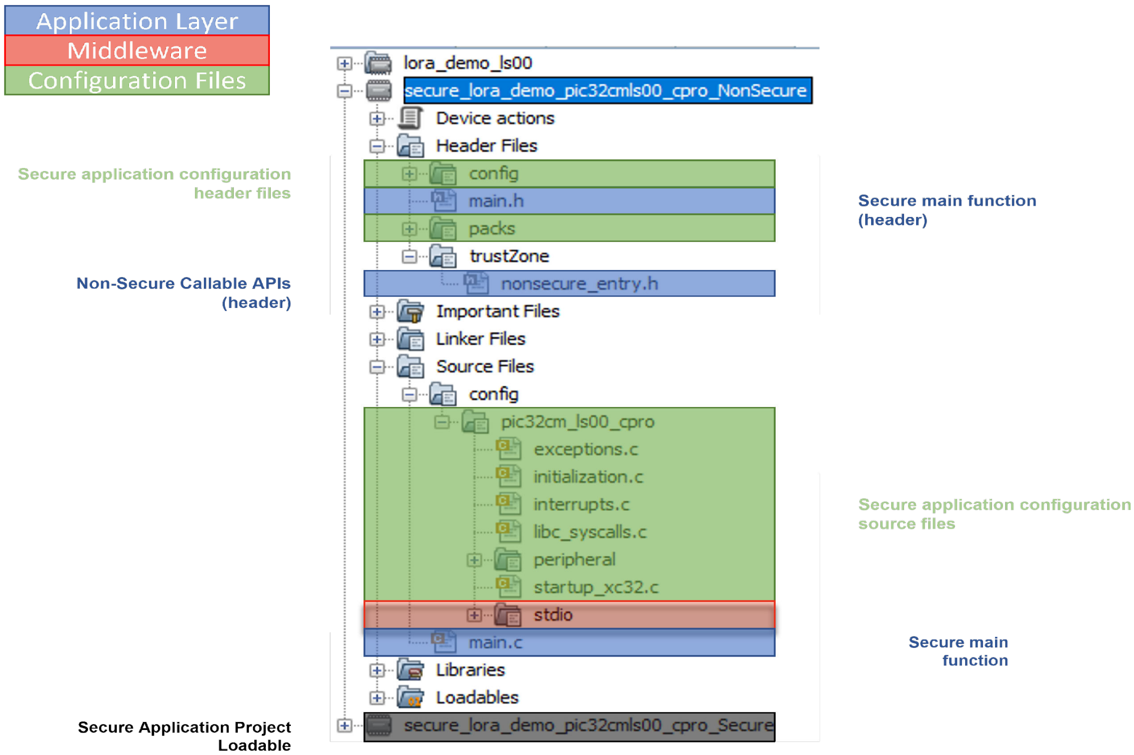


Figure 5-9. Non-Secure Project Architecture

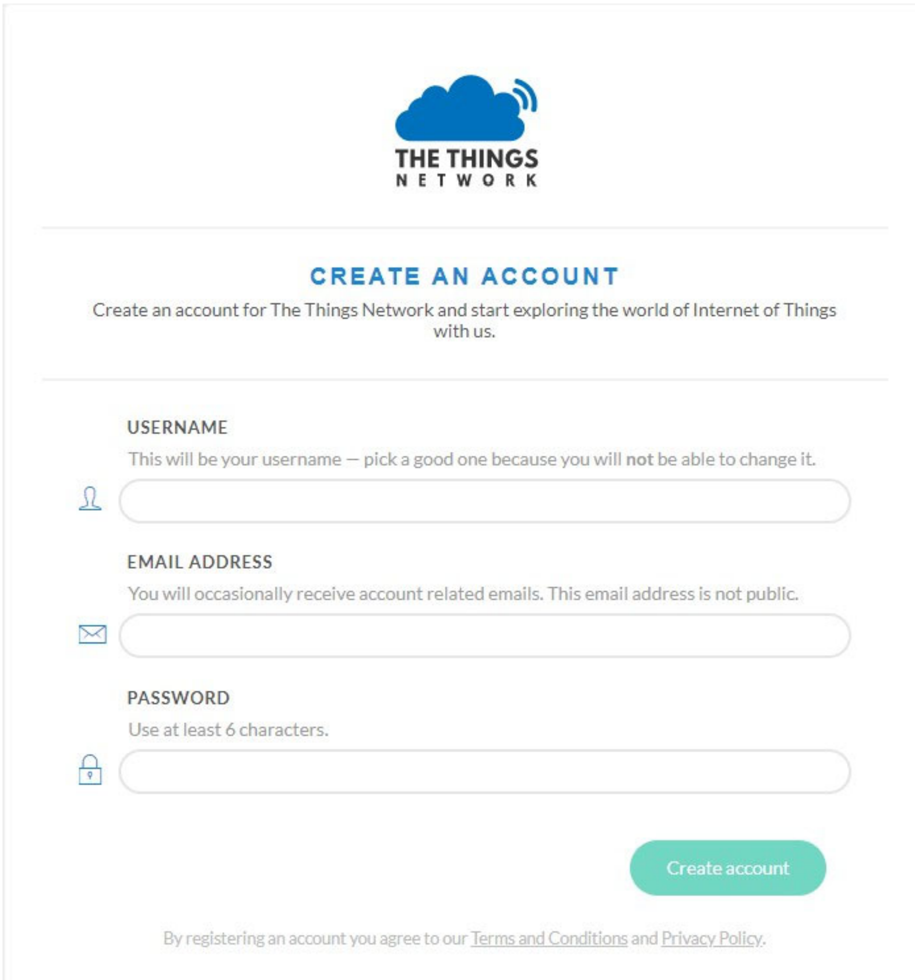


5.2.1 Creating the Application

To create an application, follow these steps:

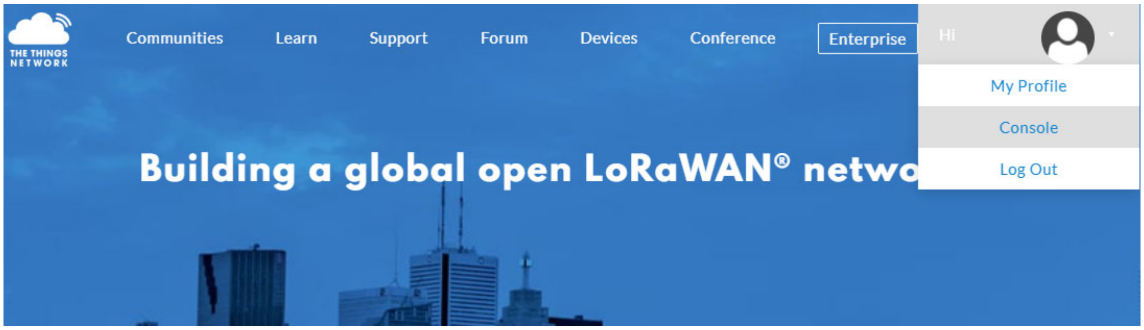
1. Before configuring the gateway, users must create the application and register the devices.
2. Create a user account on the TTN. The TTN can be downloaded from the following location: www.thethingsnetwork.org/. The TTN main window is shown below:

Figure 5-10. The Things Network Main Page



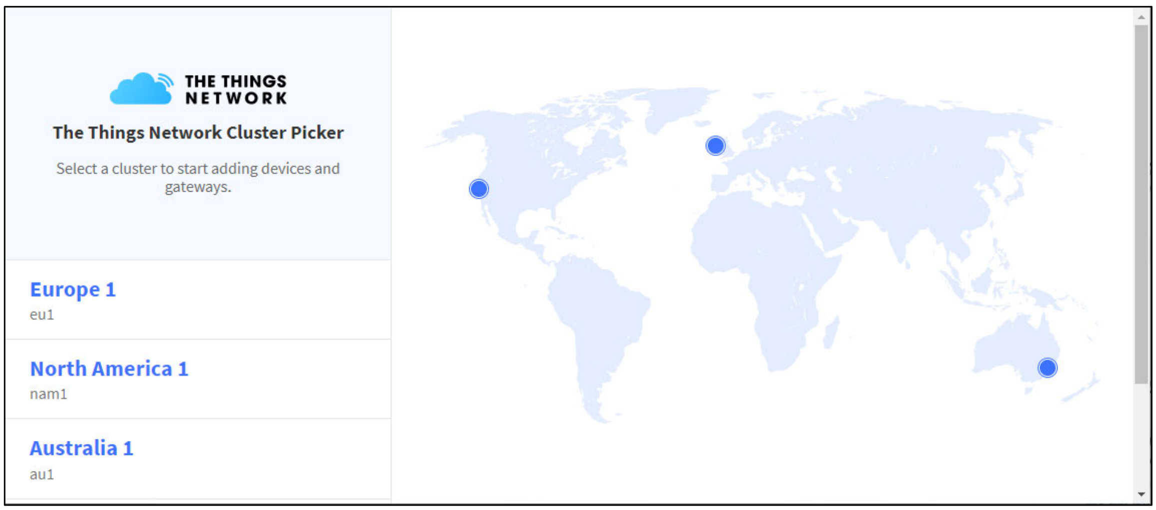
3. Enter the Username, Email Address, Password, and then Click **Create account**.
4. After creating the TTN account, follow the steps to create an application.
5. The TTN window will display the newly created account.

Figure 5-11. TTN Showing a Newly Created Account



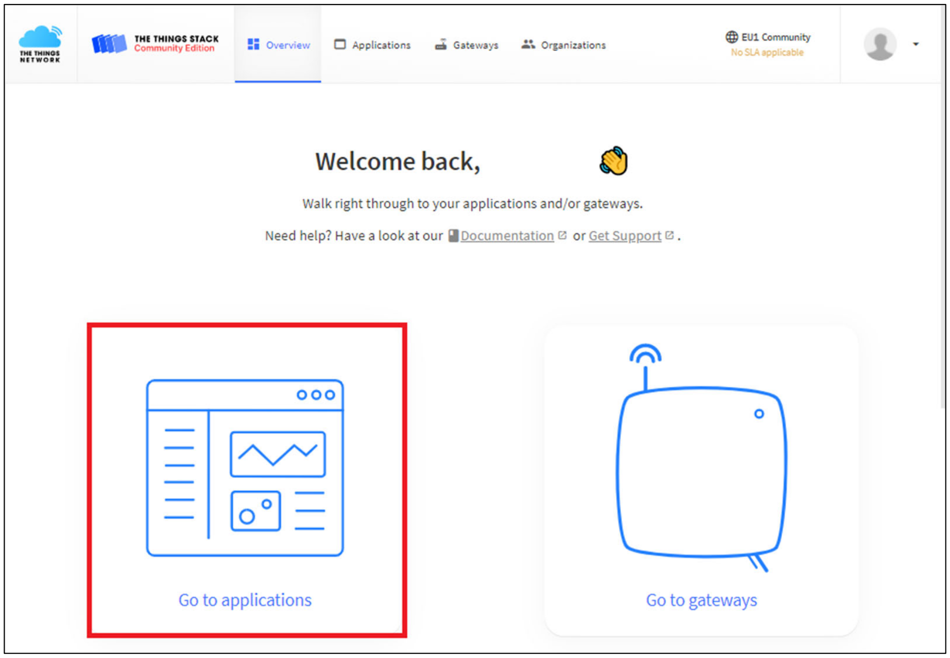
6. Click **Console**.
7. Select the application cluster (Europe1 cluster is used for this example).

Figure 5-12. TTN Cluster Selection



8. The TTN user console will be displayed as shown below. Click **Go to applications**.

Figure 5-13. TTN Console Window

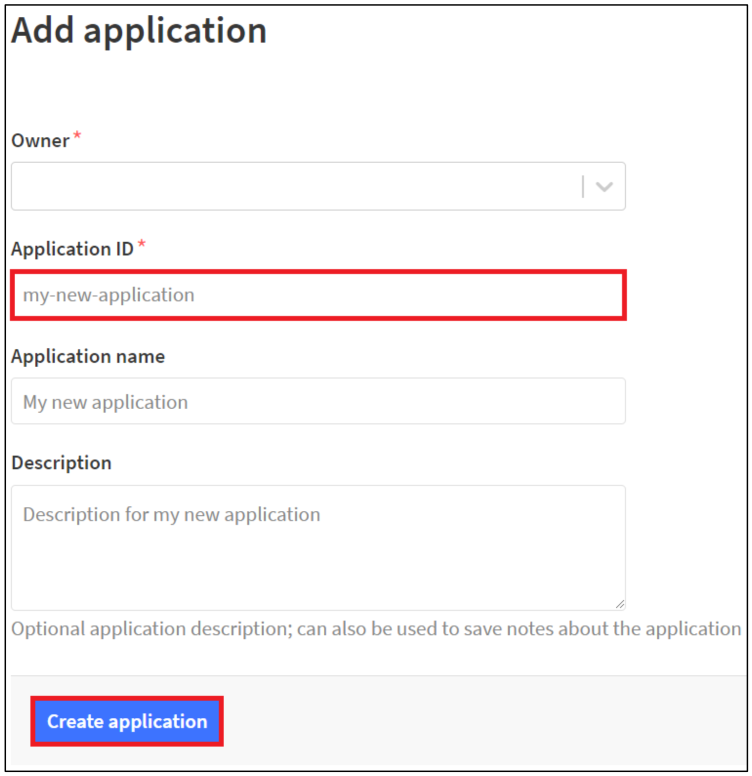


9. Click **add application**. The Add Applications window will be displayed.

Figure 5-14. Add Applications Area

10. Enter user-specific information for Application ID, Application name, and Description.

Figure 5-15. Add Applications Window



11. Click **Create application** to add the TTN application.

6. Appendix

6.1 The Things Network Setup

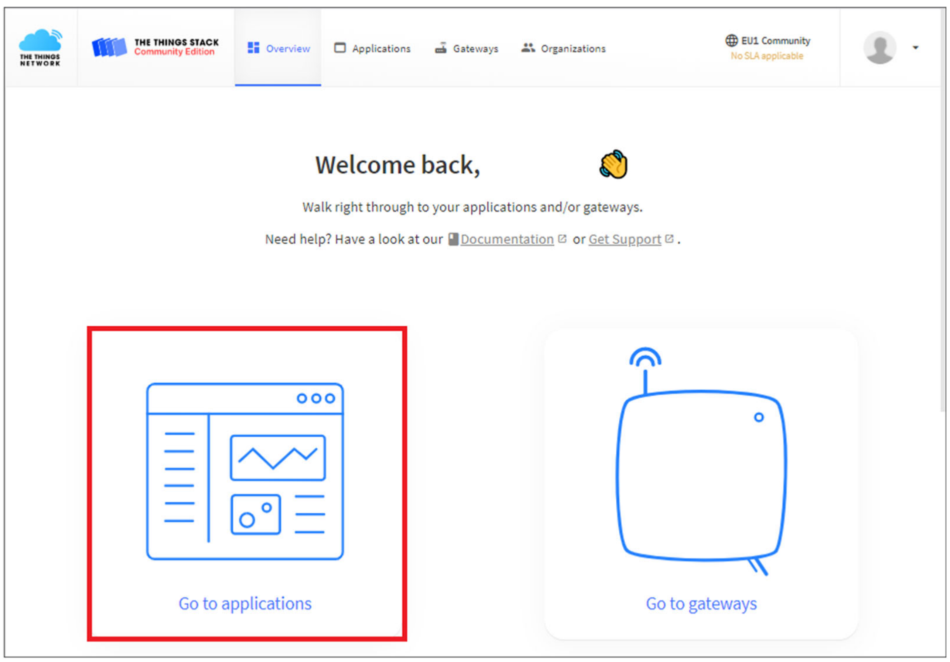
The Things Network is a service for IoT networking, that is, building a large network with gateways for LoRa communication, based on the LoRaWAN protocol, to increase the number of existing LoRa applications and users. This protocol allows multiple features that are perfect for IoT, such as low-battery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

6.1.1 Device Registration

After the application is created, the device can be registered in the new application using these steps:

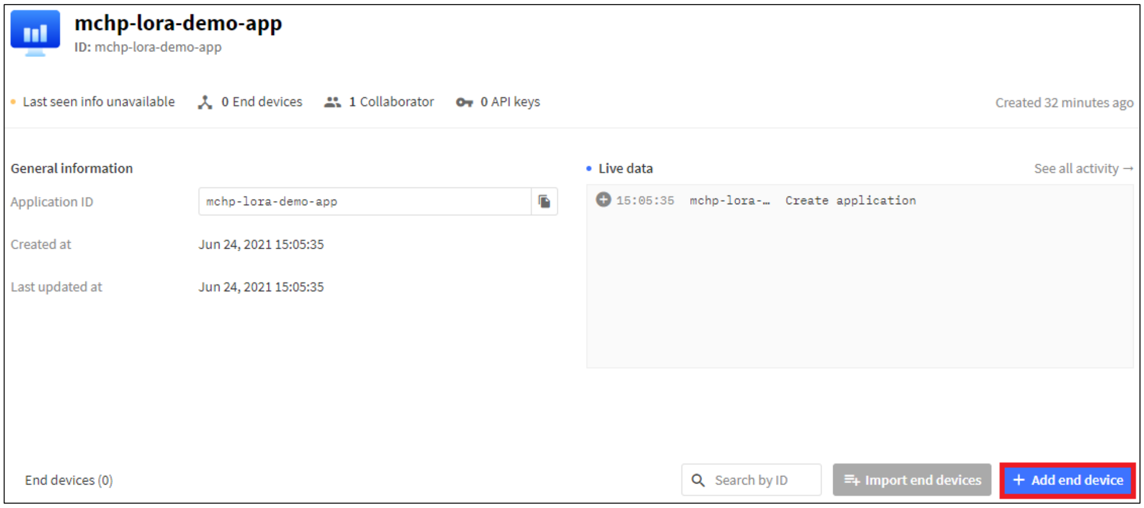
1. From the TTN console, click **Go to applications**.

Figure 6-1. TTN Console: Go To Applications



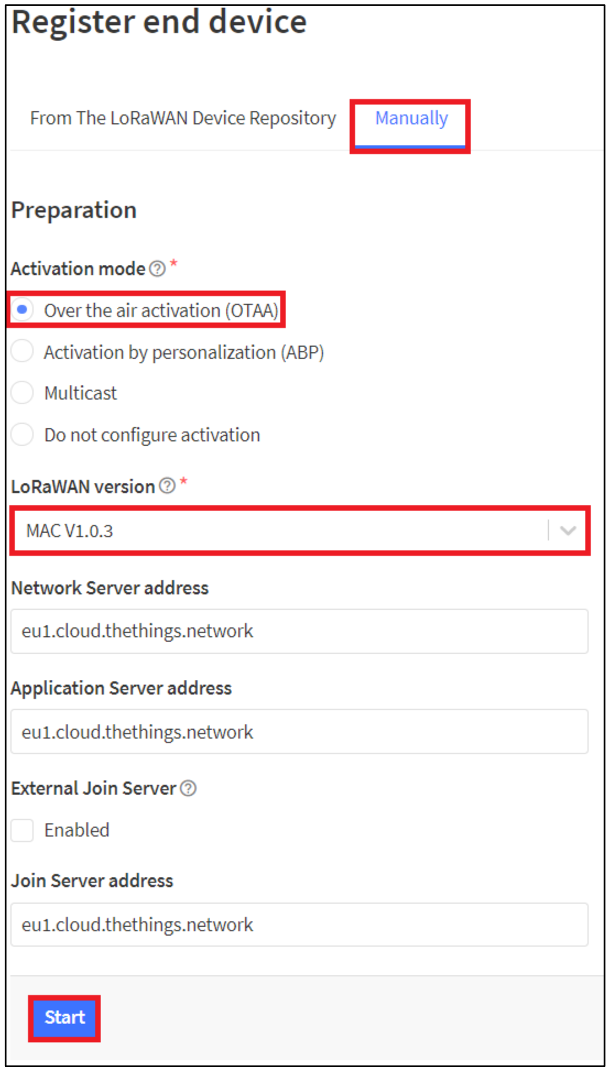
2. Click on the application to have an overview, and then click **Add end device**.

Figure 6-2. TTN Application: Add End Device



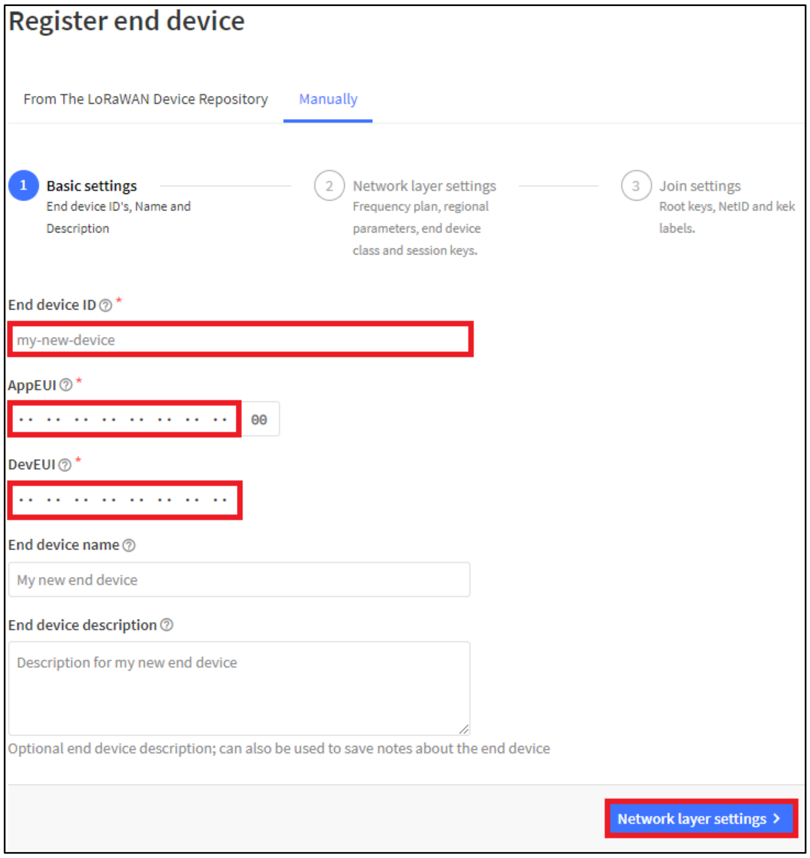
3. In the Register end device dialogue, click the **Manually** tab, and select the Over The Air Activation (OTAA) and the firmware version of the connected RN2483/RN2903 LoRaWAN device (MAC v1.0.3 for this example), and then click **Start**.

Figure 6-3. Register end Device: Preparation



4. Enter information for the End Device ID, AppEUI, and DevEUI, and then click **Network layer settings**.

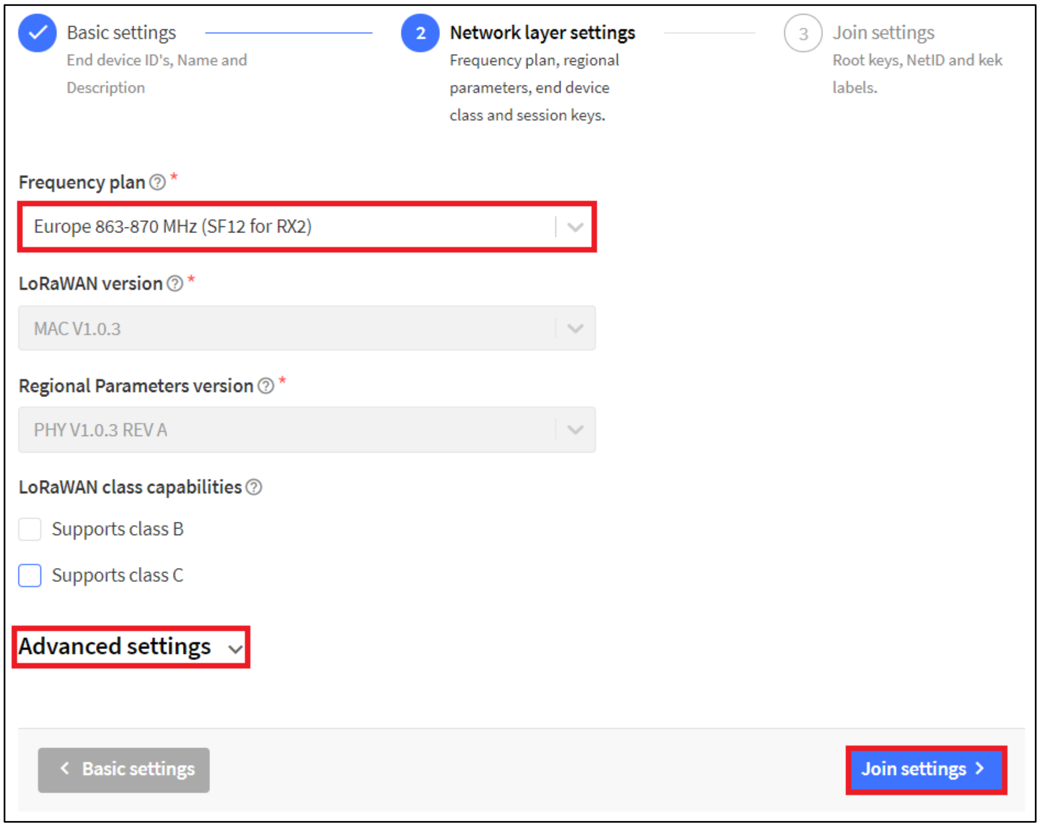
Figure 6-4. Register End Device: Basic Settings



Note: The DevEUI and AppEUI must be defined by the user. Fill the blank boxes below the AppEUI and DevEUI with key values.

5. Set the application’s Frequency plan, and then click **Join settings**.

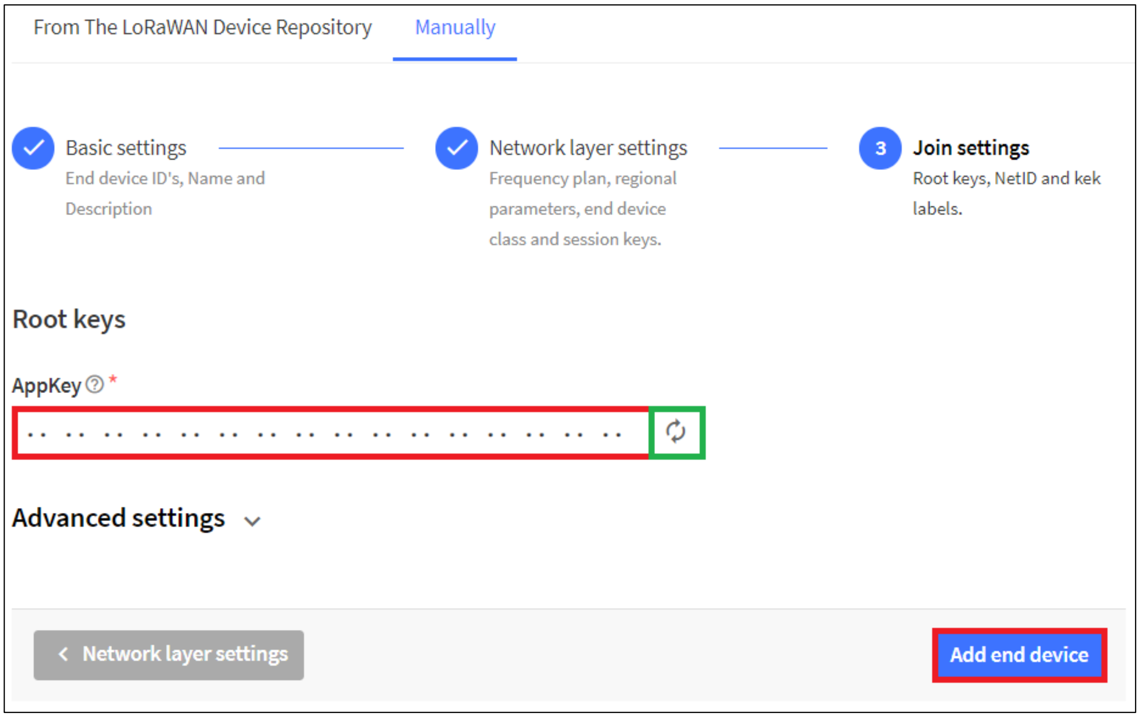
Figure 6-5. Register End Device: Network Layer Settings

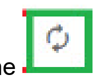


Note: The advanced settings drop-down menu can be used for the network layer configuration.

6. Enter information for the AppKey, and then click **Add end device**.

Figure 6-6. Register End Device: Join Settings



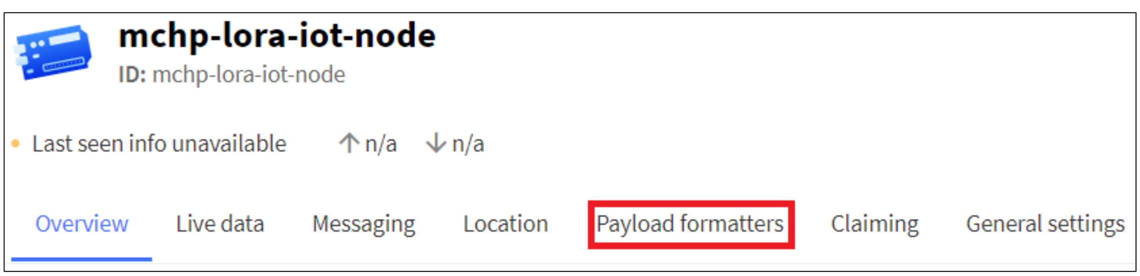
Note: The AppKey can be generated either by TTN or set by the user. Click on the  icon to switch between choices.

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.

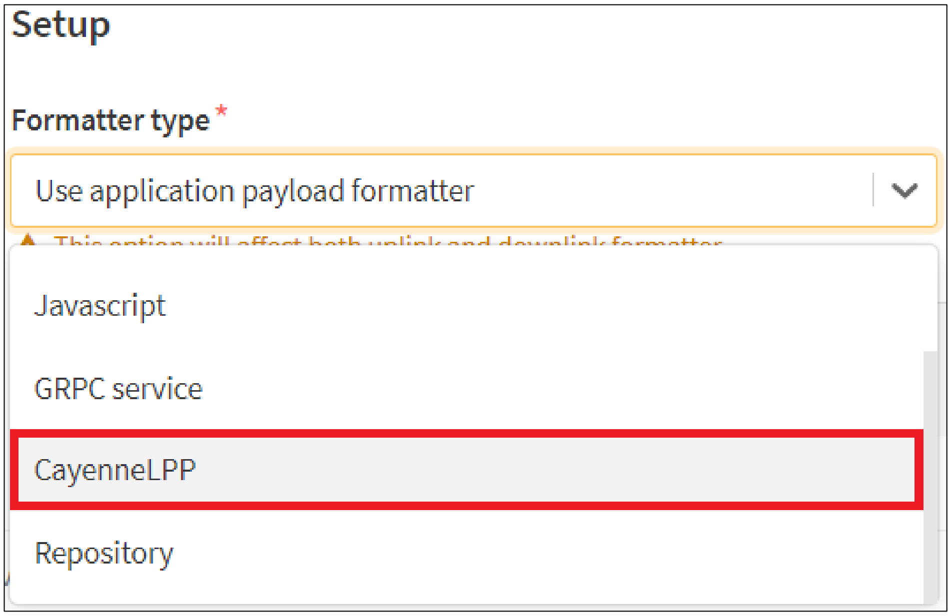
1. From the TTN console, click **Go to applications**.
2. Select the application end device to have an overview, and then click **Payload Formatters**.

   Figure 6-7. Payload Formatters



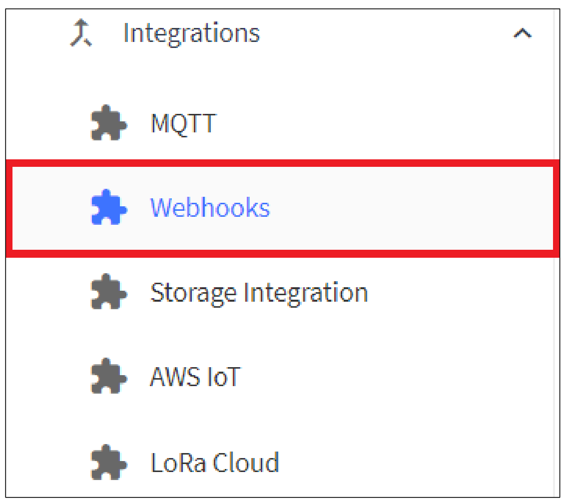
3. In the Payload Formatters Setup dialogue box, from the **Formatter type** drop-down menu, select **Cayenne LPP**.

Figure 6-8. Payload Formatters: Setup



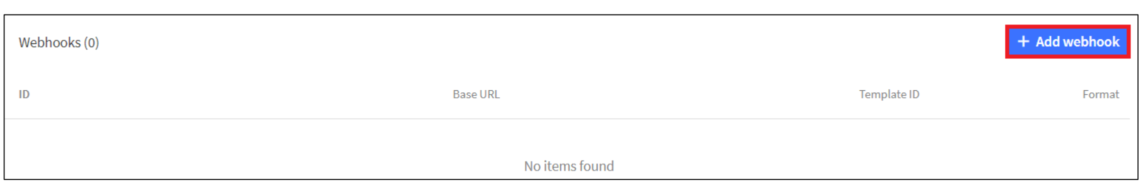
4. Under **Integrations** drop-down menu, select **Webhooks.**

Figure 6-9. Integrations: Webhooks



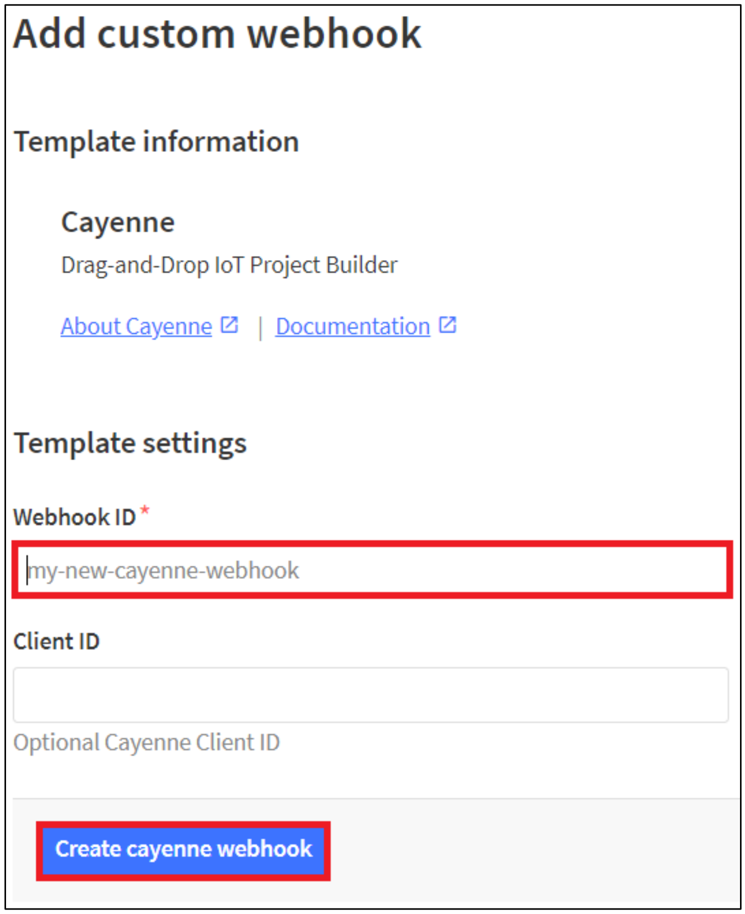
5. In the Webhooks, click **Add webhook.**

Figure 6-10. Integrations: Add Webhook



6. In the Add custom webhook dialogue, select a Webhook ID for the Cayenne integration, and then click **Create Cayenne Webhook**.

Figure 6-11. Integrations: Create Cayenne Webhook

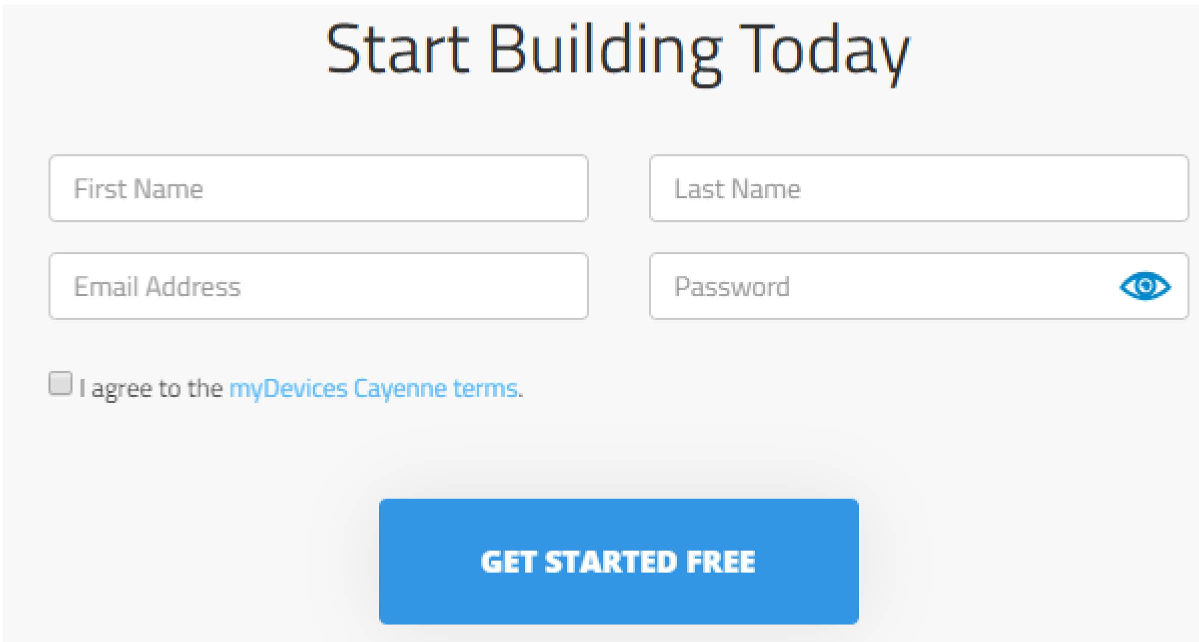


6.2 Cayenne

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/. The Cayenne main page is shown below:

Figure 6-12. Cayenne Main Page



The user must create a free account on Cayenne. After creating the account, users must follow these steps to setup Cayenne for the demonstration.

6.2.1 Registering Device

To register the devices on Cayenne, follow these steps:

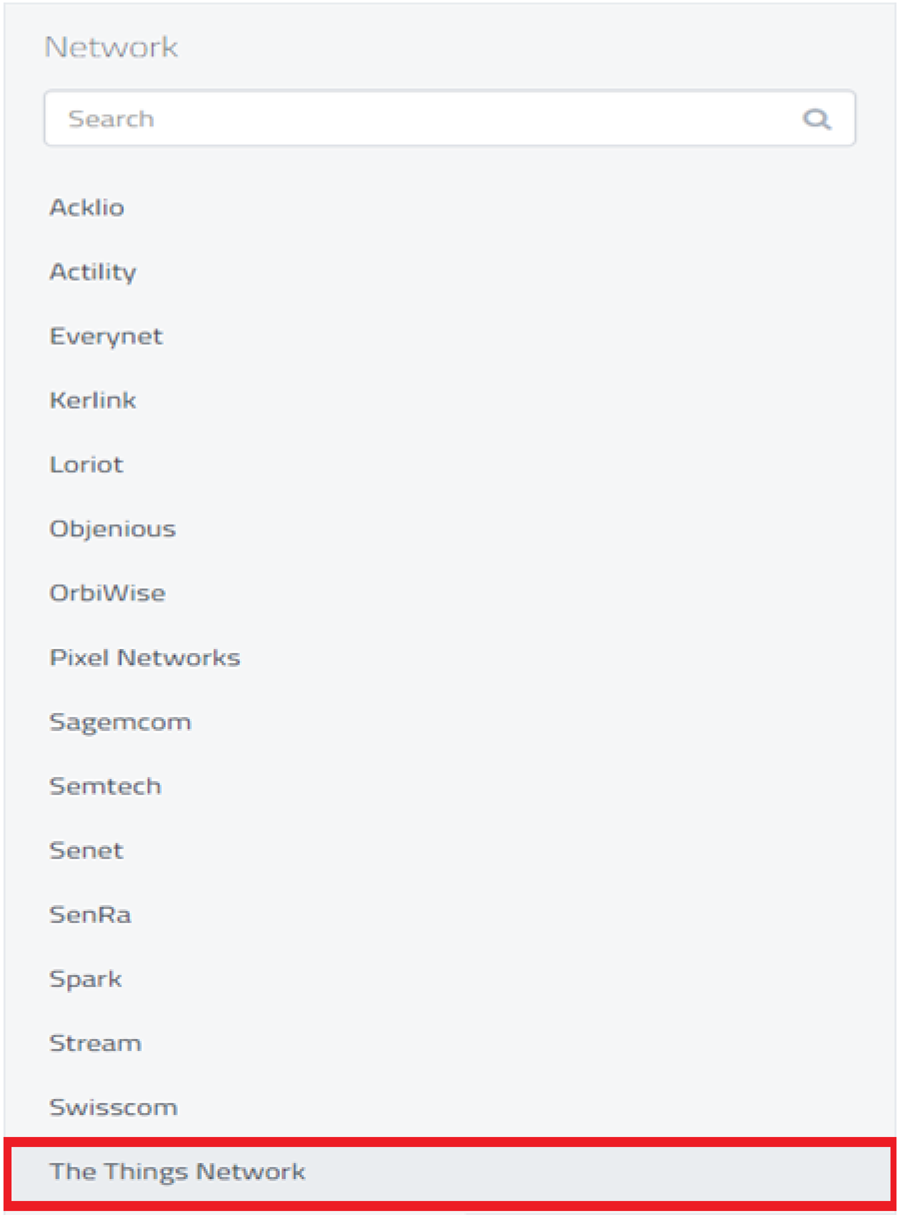
1. Log on using the newly created Cayenne account, and then click **LoRa**.

   Figure 6-13. Cayenne Main Page

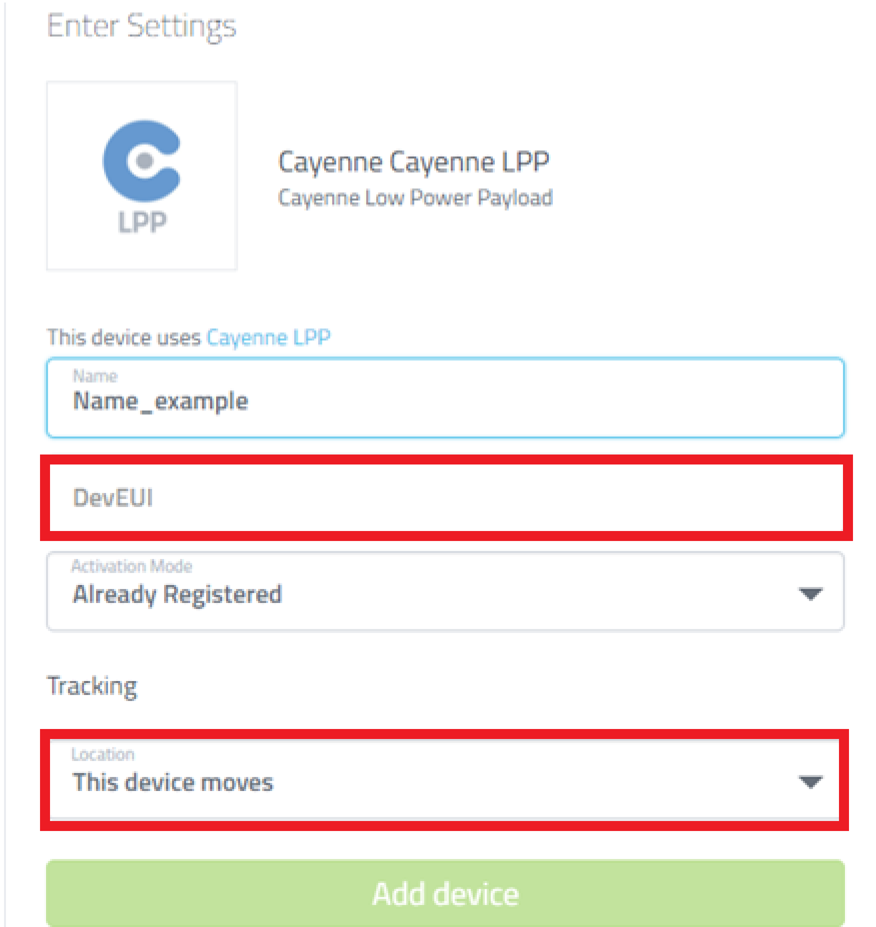
2. Scroll down to the LoRa section, and then click **The Things Network**.

Figure 6-14. LoRa Section



3. Scroll down and click **Cayenne LPP**.
4. Enter *DevEUI* of the device registered in the TTN application, change the device name if required, and then click **Add device**.

Figure 6-15. Cayenne Settings



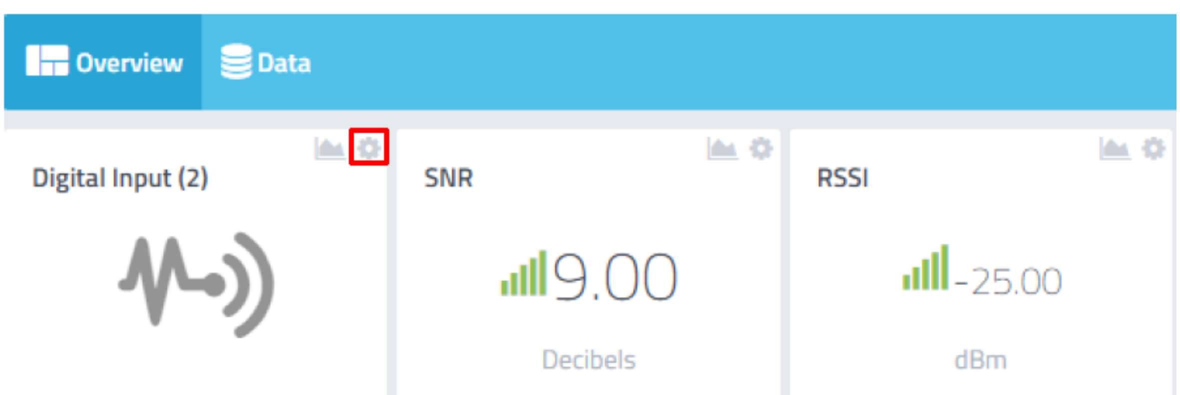
5. The device is registered in Cayenne.

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 displayed on the Cayenne dashboard, click  (wheel icon) on the widget. block.

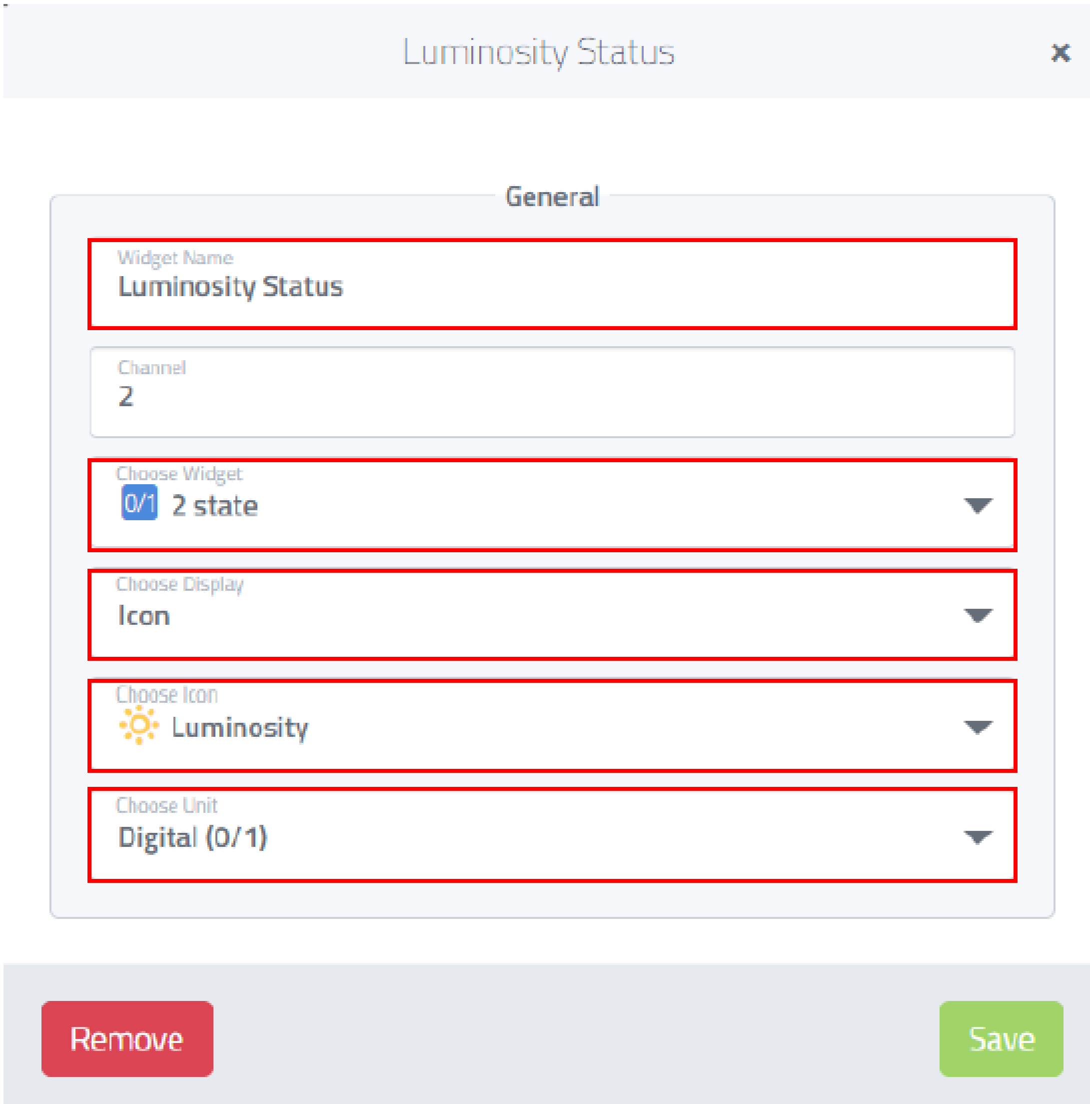
Figure 6-16. Widget Block



Note: The widget appears only after the first connection of the node on the network.

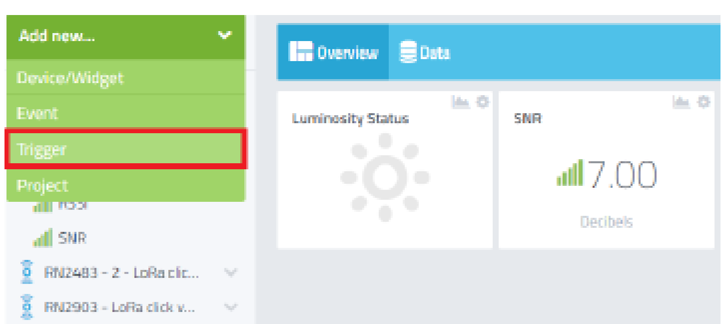
2. Configure Digital Input (2) with different parameters as shown in the following figure.

Figure 6-17. Widget Configuration Window



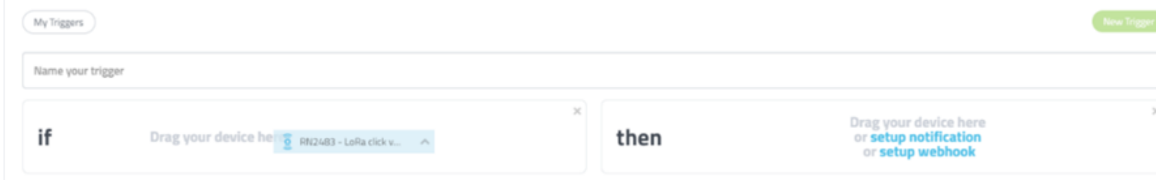
3. Click **Save.**
4. After widgets are configured, under Add new, click **Trigger**.

Figure 6-18. Add New Trigger section



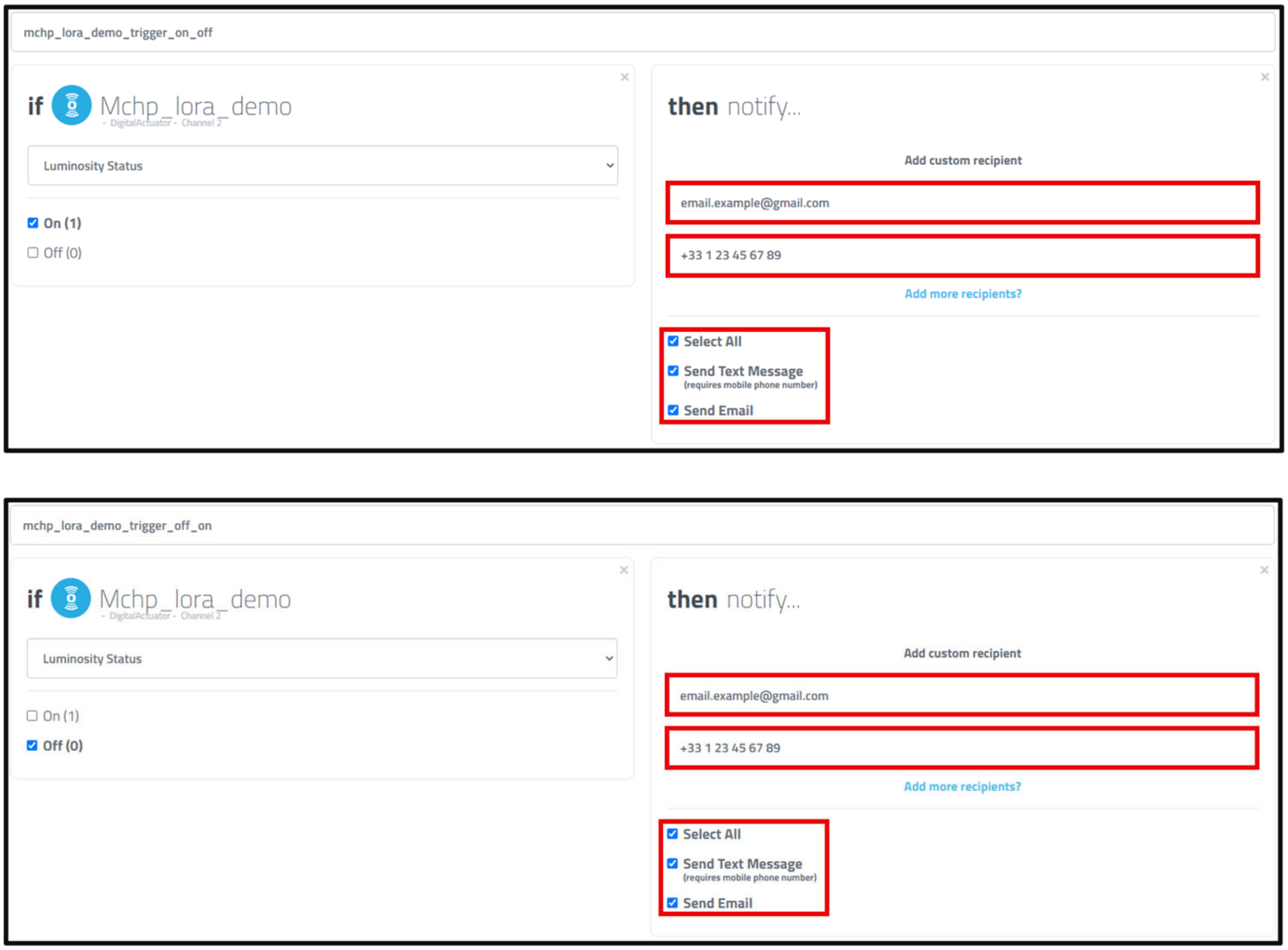
5. Drag and drop the device into the *if* box, and then click **setup notification** in the *then* box.

Figure 6-19. IF and Then Section



6. When *if* and *then* boxes are filled, reproduce the following triggers.

Figure 6-20. Reproducing the Triggers



Note: Notifications through text message or emails can be chosen by selecting *Select All*, *Send Text Message* or *Send Email* options, or under 'Add Custom Recipient' enter mobile number and email address.

7. Click **Save** to save the triggers.
8. The triggers are now configured in Cayenne. The board can be reset to have the demonstration running.

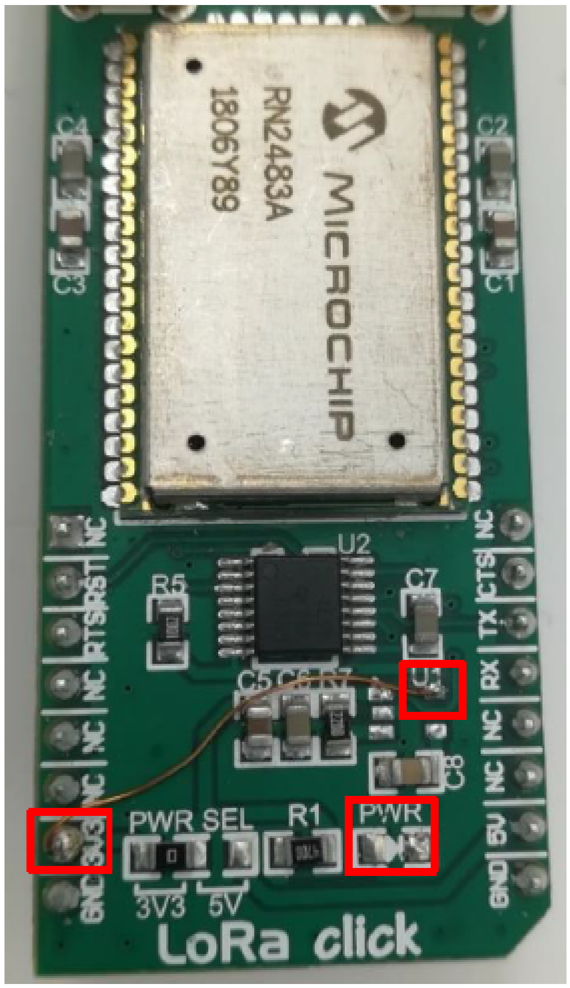
6.3 Hardware Modifications

To reduce the consumption of the global demonstration, some modifications are applied to the MikroElektronika LoRa click board and the I/O1 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 the MikroBus 3.3V pin to the regulator output soldering pad. All modifications made on the 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 6-21. RN2483A Showing Modificaitons

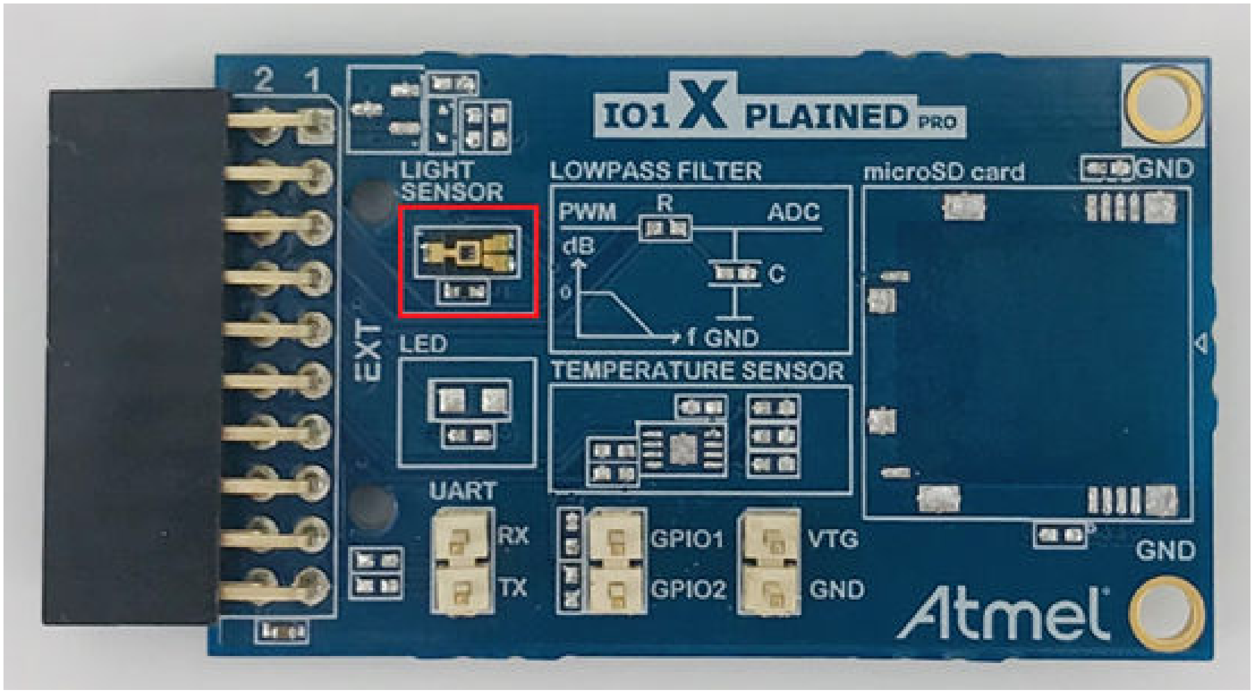


6.3.2 I/O1 Xplained Pro Extension Board

The power consumption of the overall application can be reduced by isolating the light sensor on the I/O1 Xplained Pro board, that is removing all components on the board except light sensor and the associated resistor.

The following figure illustrates the I/O1 Xplained Pro board.

Figure 6-22. I/O1 Xplained Pro Board



7. Revision History

Revision C - 04/2022
Removed all “Confidential NDA” verbiage from the document for this release.

Revision B - 01/2022
The following updates were made throughout this document.

- References to the MHC were updated to MCC
- Updated figure 5-2 with a new image and reworded the title

Revision A - 08/2021
This is the initial release of this document.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **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 (FAQs), technical support requests, online discussion groups, Microchip design partner 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

Product Change Notification Service

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

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: 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 specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "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

Trademarks

ISBN: 978-1-6683-0186-9

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733

China - Beijing
Tel: 86-10-8569-7000

China - Chengdu
Tel: 86-28-8665-5511

China - Chongqing
Tel: 86-23-8980-9588

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115

China - Hong Kong SAR
Tel: 852-2943-5100

China - Nanjing
Tel: 86-25-8473-2460

China - Qingdao
Tel: 86-532-8502-7355

China - Shanghai
Tel: 86-21-3326-8000

China - Shenyang
Tel: 86-24-2334-2829

China - Shenzhen
Tel: 86-755-8864-2200

China - Suzhou
Tel: 86-186-6233-1526

China - Wuhan
Tel: 86-27-5980-5300

China - Xian
Tel: 86-29-8833-7252

China - Xiamen
Tel: 86-592-2388138

China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444

India - New Delhi
Tel: 91-11-4160-8631

India - Pune
Tel: 91-20-4121-0141

Japan - Osaka
Tel: 81-6-6152-7160

Japan - Tokyo
Tel: 81-3-6880- 3770

Korea - Daegu
Tel: 82-53-744-4301

Korea - Seoul
Tel: 82-2-554-7200

Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

Malaysia - Penang
Tel: 60-4-227-8870

Philippines - Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan - Hsin Chu
Tel: 886-3-577-8366

Taiwan - Kaohsiung
Tel: 886-7-213-7830

Taiwan - Taipei
Tel: 886-2-2508-8600

Thailand - Bangkok
Tel: 66-2-694-1351

Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829

Finland - Espoo
Tel: 358-9-4520-820

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Garching
Tel: 49-8931-9700

Germany - Haan
Tel: 49-2129-3766400

Germany - Heilbronn
Tel: 49-7131-72400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Rosenheim
Tel: 49-8031-354-560

Israel - Ra'anana
Tel: 972-9-744-7705

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Padova
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Norway - Trondheim
Tel: 47-72884388

Poland - Warsaw
Tel: 48-22-3325737

Romania - Bucharest
Tel: 40-21-407-87-50

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Gothenberg
Tel: 46-31-704-60-40

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820