NXP Semiconductors
User's Guide

Document Number: SLN-RPK-NODE-UG
Rev. 0, 08/2018

Rapid IoT Prototyping Tool User's Guide

Contents

Recommended Computer Configuration

Rapid IoT Studio requires up-to-date computer and web browser configuration for successful experience.

Our engineer team has verified and tested the following computers, OS and Web browser versions.

Computer type	OS version	Web browser
Apple	Mac OS	Safari
PC	Windows 7 / 10	Edge 40.15063, Chrome 67.0.03396, Firefox 60.0.1
PC	Linux	Chrome

Table 1. Tested Computer Configurations

<u>Internet Explorer is not supported by the Rapid IoT Studio online tool.</u>

If your computer OS or Web browser version doesn't appear in this list, it may still be compatible, but our team didn't test this configuration so its functionality is not warranted.

Recommended Phone Configuration

SLN-RPK-NODE requires some iOS or Android phone App to demonstrate its full capabilities.

There are hundreds of combinations between phone models and OS versions requiring intensive efforts to validate them all.

Our engineer team has verified and tested the following phone models and OS versions

Phone type	Model	OS version
Apple	iPhone 6S / 7 / 7+ / X and iPad mini2	iOS 11.3
Apple	iPhone 7+ / X	iOS 11.4
Google	Nexus 5	Android 6
Google	Galaxy Tab and Nexus 9	Android 7
Google	Samsung A3, Huawei P6 and Nexus 6P	Android 8

Table 2. Tested Phone Configurations

<u>Windows and Blackberry phones are not supported by the Rapid IoT phone App.</u>

If your phone model or OS version doesn't appear in this list, it may still be compatible, but our team didn't test this configuration so its functionality is not warranted.

Usage Condition

SLN-RPK-NODE embeds numerous components, among which sensors, display and battery, qualified for different temperature grades so to obtain the best performances with your Rapid IoT prototyping kit, we recommend using the board at ambient temperature condition between +10°C/50°F and 30°C/90°F.

1. Introduction

NXP's Rapid IoT prototyping kit is a comprehensive, secure and optimized IoT end node solution with a user- friendly development environment that enables anyone to quickly take their idea to a proof-of-concept.

Rapid IoT (part number: SLN-RPK-NODE), embeds all the components required to prototype secure and connected end products. Its architecture is built upon two controllers:

- Kinetis K64F for the main application, powered by an ARM® Cortex®-M4 core
- Kinetis KW41Z for wireless connectivity, powered by an ARM® Cortex®-M0+ core

SLN-RPK-NODE can be used to evaluate the combination of K6x Kinetis K series and KW41Z/31Z/21Z Kinetis W series devices:

- User-program will be executed by the MK64FN1M0VMD12 microcontroller, clocked up to 120MHz and embedding 1,024KB of flash and 256KB of RAM memories, dual 16-bit analog-to-digital controllers (ADCs) sampling at up to 800k samples per second in 12-bit mode, numerous Serial interfaces as well as multiple timers and GPIOs.
- Wireless connectivity will be supported by the MKW41Z256VHT4 microcontroller, clocked up to 48MHz and embedding 512KB of flash and 128KB of RAM memories with simultaneous BLE v4.2 and Thread connectivity.

SLN-RPK-NODE includes:

- 10-axis motion sensing thanks to a combo accelerometer / magneto-meter
- gyroscope
- pressure sensor for altitude measurement
- environmental sensing via temperature/humidity, ambient light and air quality sensors
- display capabilities with low-power color screen
- authentication, identification
- user interfaces with LEDs, buzzer and touch plus push buttons
- additional memory for data storage
- rechargeable battery

The factory application includes USB and Bluetooth/Thread bootloaders to program your own firmware without external tool, and several IoT application use-cases leveraging components on board.

SLN-RPK-NODE is supported by a comprehensive and free-of-charge enablement suite from NXP and its partners, that includes:

- Rapid IoT Studio, an online IDE with visual drag-and-drop style programming
- Hardware design files
- Software source-code
- MCUXpresso development tools
- Documentation
- Training material

SLN-RPK-NODE is also compatible with the MikroElektronika Hexiwear tools, including the Docking Station, offering debug capabilities and a broad range of expansion possibilities via the Click™ boards.

2. Get Started with Rapid IoT

2.1. Out-of-the-box-experience

You just received your Rapid IoT Prototyping kit and you probably wonder where to start?

Figure 1. Rapid IoT Package overview

Let's begin opening the box and reviewing together the kit content.

The Rapid IoT unit comes with a printed Quick Start Guide, a SIM tool and a micro-USB cable.

Figure 2. Rapid IoT kit content

The printed Quick Start Guide provides useful information like the location and the numbering of the Touch or Push buttons as well as the menu architecture of the Factory application.

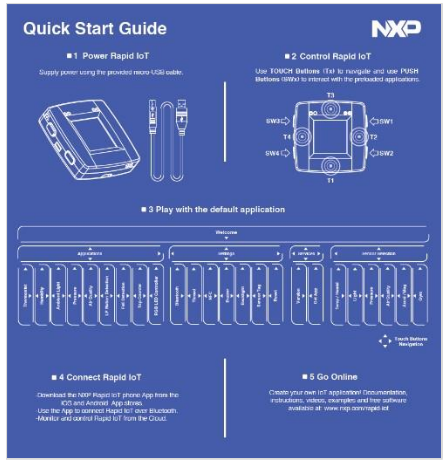


Figure 3. Printed Quick Stat Guide

The battery of your Rapid IoT probably ran out of energy so let's charge it first.

Connect one end of the provided micro-USB cable to your computer USB port or to any USB charger and the other end to the micro-USB connector of your Rapid IoT board.

Figure 4. Rapid IoT Power-on

The RGB LED and the White/Blue LEDs located in the top front will blink during few seconds to confirm that the preprogrammed Bootloaders loaded successfully (details in the bootloader chapter).

Figure 5. Bootloader Flash LEDs

While the factory application is loading the NXP and the Atmosphere logos will be displayed.

Figure 6. Logo Opening Screens

Then the Welcome menu with the Rapid IoT logo and the clock will show up on your Rapid IoT screen.

Figure 7. Factory Program Welcome Screen

Now helped with the printed Quick Start Guide you can navigate through the different menus with the Touch electrodes and change the options with the Push buttons.

The factory application is divided into four main categories: Application, Settings, Info and Sensors.

From the Application section, you can access mini-Apps that leverage the sensors onboard in different use-cases like: thermostat, barometer, air-quality, motion or fall detection, or color-light controller.

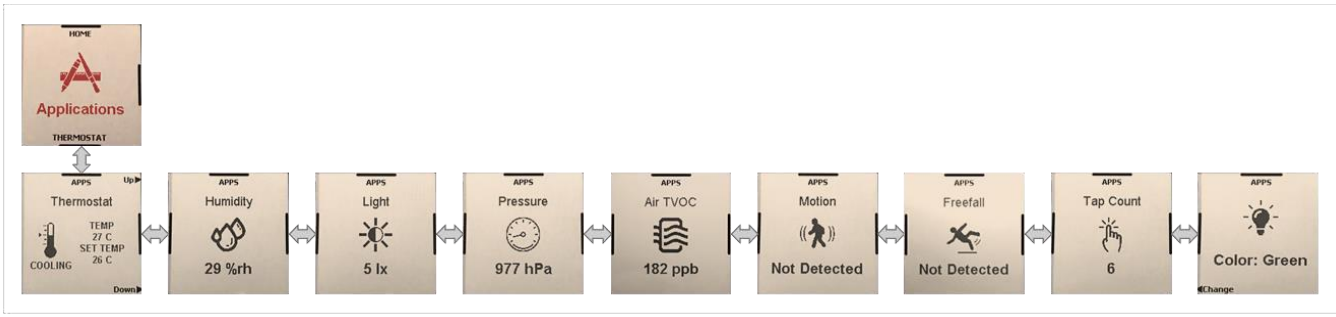


Figure 8. Factory Program Application Screens

From the Settings section, you can change the Wireless mode (Bluetooth or Thread), enable/disable the internal access to the NFC-tag or the Sensor-tag (Sensor data pushed through wireless), turn on/off the buzzer or the screen backlight, or restart your Rapid IoT board.

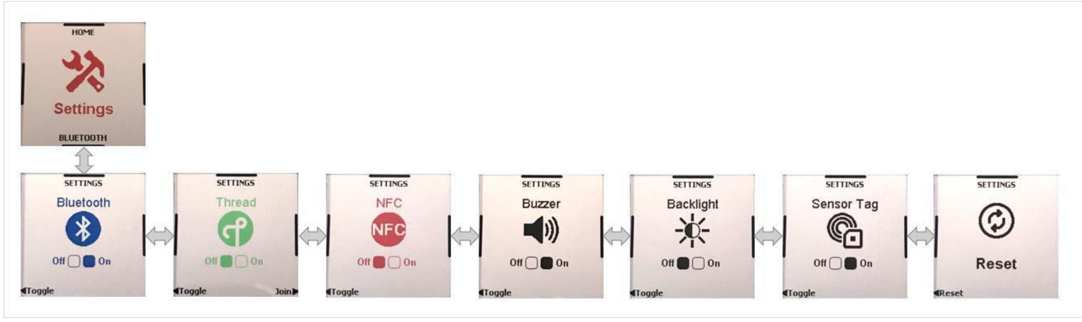


Figure 9. Factory Program Settings Screens

From the Info section, you can verify the version of the firmware programmed in your board or scan the QR code to quickly download the Android or iOS Phone App to monitor/control your Rapid IoT over Bluetooth.

Figure 10. Factory Program Info Screens

From the Sensors section, you can enable/disable each sensor onboard to create your custom monitoring profile and extend your Rapid IoT battery lifetime.

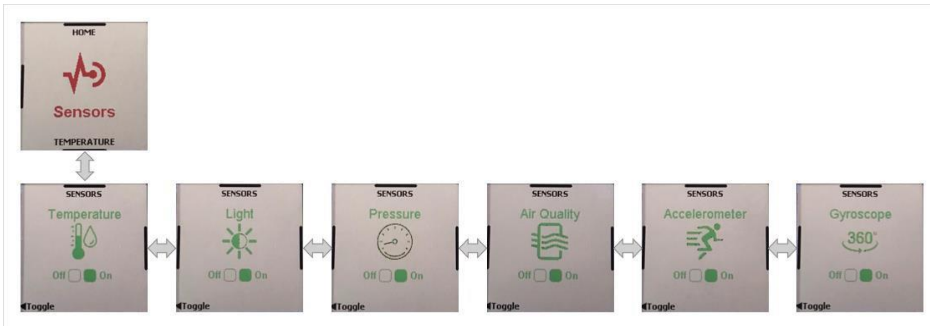


Figure 11. Factory Program Sensor Screens

By default, the application is configured to stream all the sensor data (Sensor tag) over Bluetooth radio to an Android or iOS Phone/Tablet. Thanks to the dedicated phone App the sensor data will be automatically pushed through WiFi or Cellular connectivity to the Cloud.

IMPORTANT: The firmware programmed in factory is a public version that can be broadly distributed. Rapid IoT board will be detected independently from the user-account used to login to the Phone App.

Now you must download and install the NXP Rapid IoT Android or iOS phone App from the Google or Apple App store.

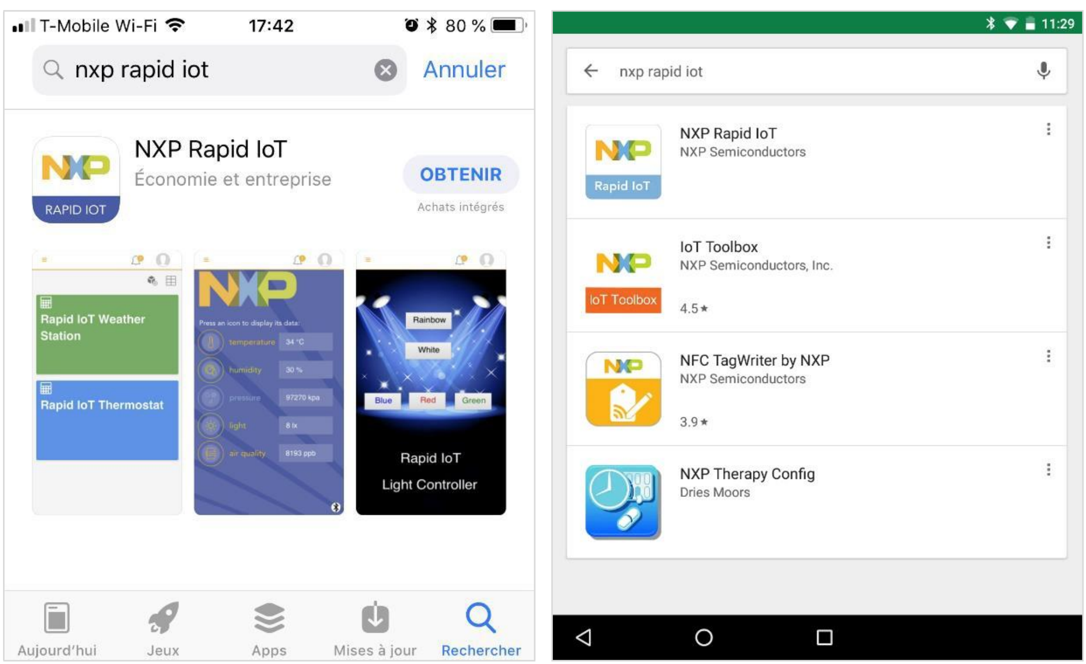


Figure 12. Rapid IoT Android/iOS App Download and Install

Then launch the Rapid IoT phone App from your phone/tablet.

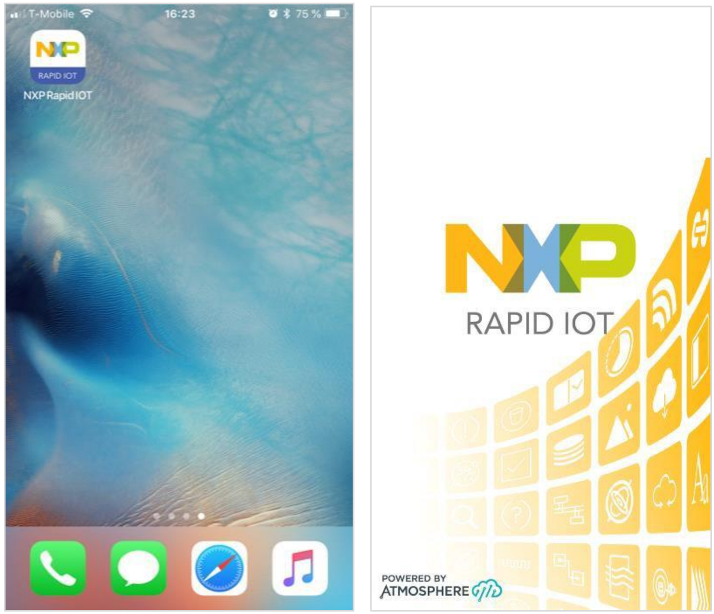


Figure 13. Launch Rapid IoT phone App

You will need a free NXP account to connect to the Rapid IoT phone App.

If you already have an account press LOGIN NXP SSO and you will be automatically redirected to the NXP Sign In page (NXP employees use your NXP WBI).

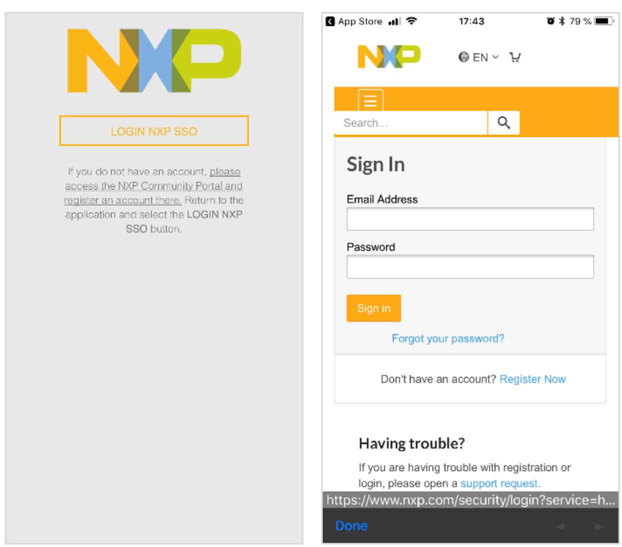


Figure 14. Sign-in/up with Rapid IoT phone App

If you don’t have an account yet simply select the link please access the NXP Community Portal and register an account there or the link Register Now from the NXP Sign In page.

Enter the requested information (First/Last Name, Email, Password, Country and Company Name). Check the box to confirm that you have read and accept the Term of Use and Privacy Policy and the box to confirm that you are not a Robot.
Finalize your registration by pressing the button Register.

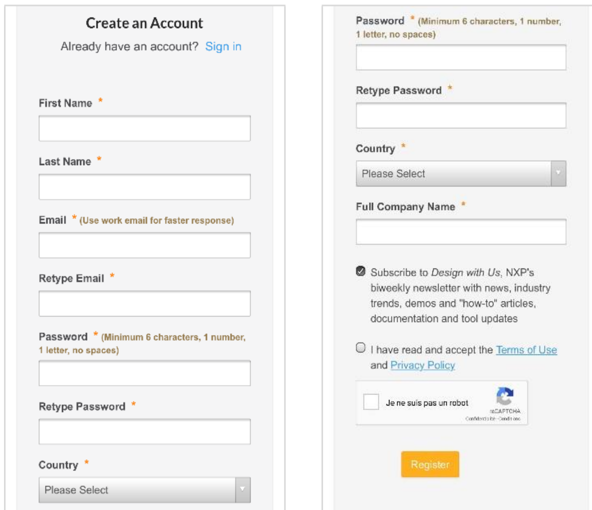


Figure 15. Create a free NXP account

Visit the mailbox used during the registration and open the email sent by:
nxpupdt1@contact.nxp.com

Click on the enclosed link to verify your email.
NOTE: Check the folder Junk/Spam/Clutter, if the confirmation email doesn’t show up in your inbox.

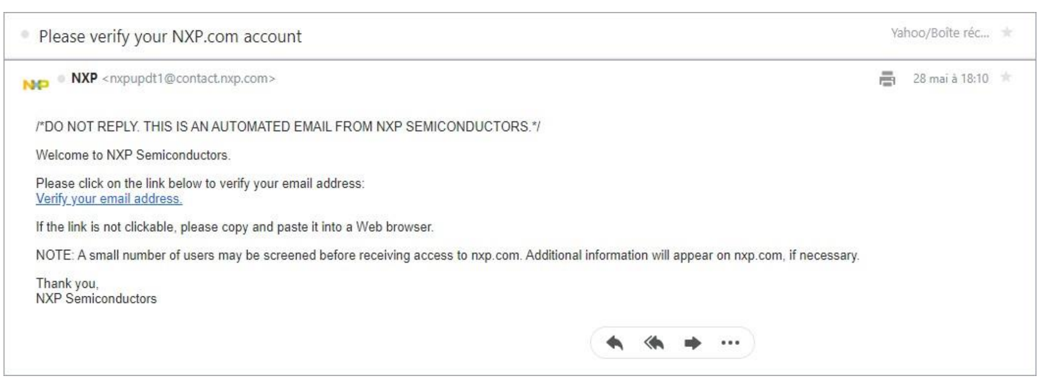


Figure 16. NXP Confirmation Email

You are now ready to login with your NXP account.
Press the button LOGIN NXP SSO.
Enter you Email Address and Password and press Sign In.

At the first connection to the Rapid IoT phone App with your SSO account a window with the User Agreement will appear.
Check the box Accept Term of Use and Privacy Policy and press SUBMIT

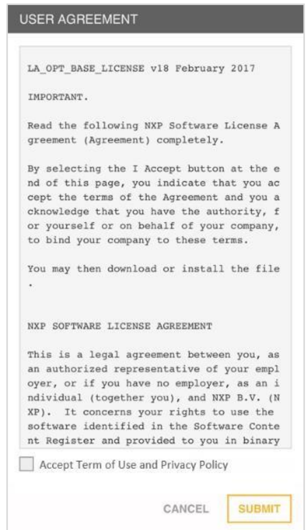


Figure 17. Accept NXP Term of Use and Privacy Policy

The phone App will automatically launch the Dashboard view.

Open the left panel by selecting the icon located in the top left corner.
Select the Device view to pair your equipment with your Rapid IoT board.

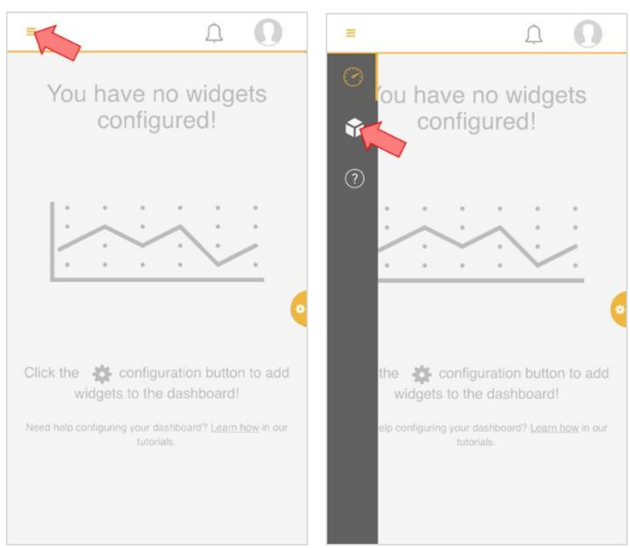


Figure 18. Open the Device View

Select the icon located in the top right corner to scan the Bluetooth devices in the range.

Figure 19. Connect New Device

Each Rapid IoT board has a unique MAC address printed on a sticker at the back of the casing. The phone App indicates the list of the Bluetooth devices detected with the type of Application programmed in their memory as well as the last four Hexadecimal numbers of their MAC address.

Select your Rapid IoT board and press PROVISION.

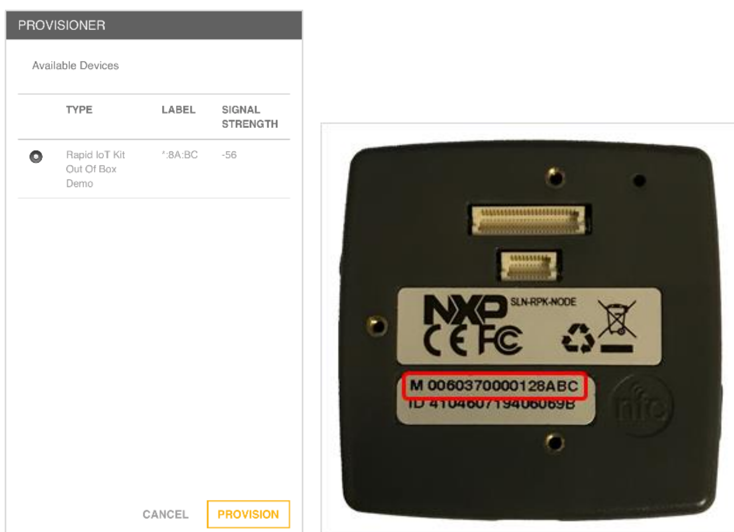


Figure 20. Select and Provision your Rapid IoT board

The phone App will automatically launch the phone App interface for the Out of the Box Demo.
It takes approx. 5s for the first sensor value to appear on the phone App.
Control the color of the RGB LED remotely by selecting one of the options in the phone App.

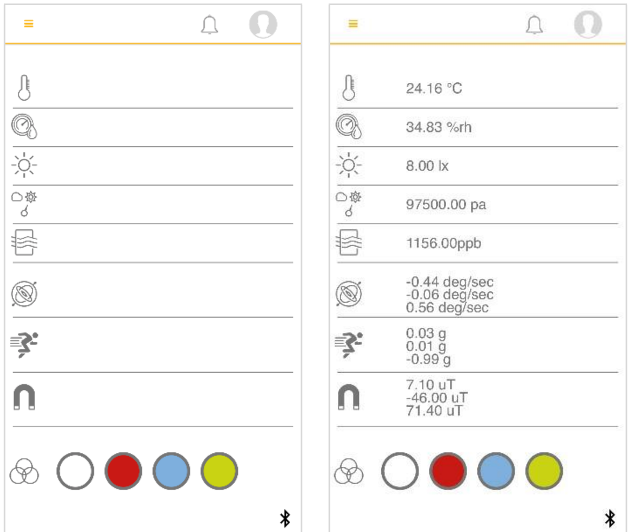


Figure 21. Phone App View

Congratulation you now successfully monitor and control your Rapid IoT with the phone App over Bluetooth Low-Energy (BLE)!!

We will now monitor your Rapid IoT board from the Cloud.

To visualize your Rapid IoT data remotely you need another Android/iOS terminal with the phone App installed or you can simply use the web browser of your computer.

The instruction below will apply for a computer access.

Open your web browser and visit the dedicated Rapid IoT Studio portal at: http://rapid-iot-studio.nxp.com.

Login with the same NXP account used to connect to the phone App.

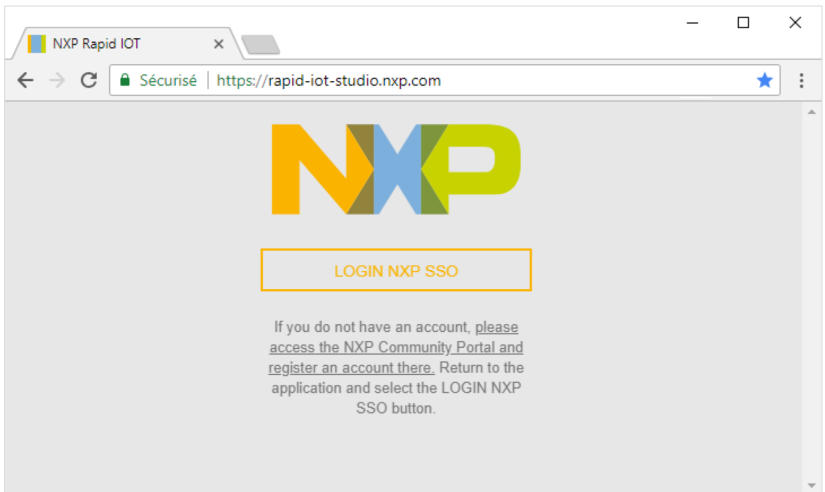


Figure 22. Computer access

The Dashboard view will open automatically.
Open the right panel by selecting the icon located in the right edge.

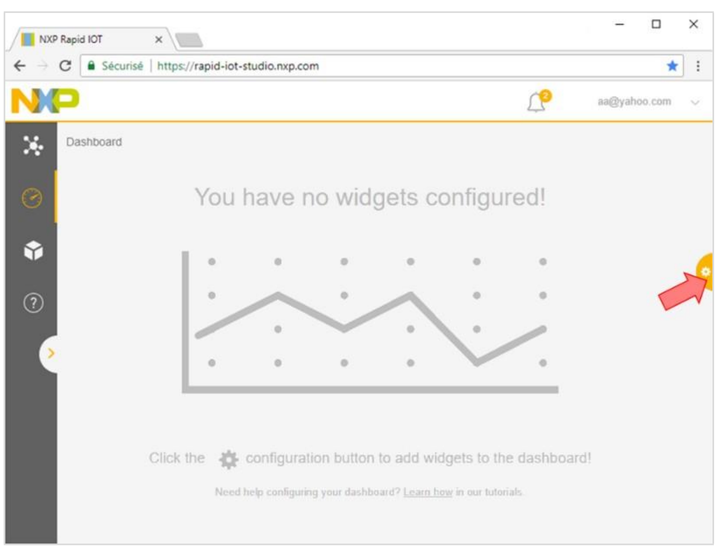


Figure 23. Dashboard view

Choose the option Data Graph to display from the Cloud the sensor value in a graphic.

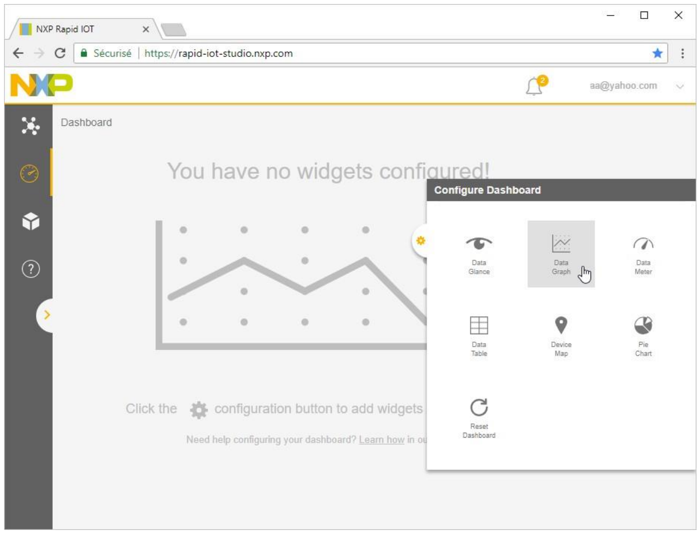


Figure 24. Create new Cloud Data Graph

A blank graphic will now be added to your Dashboard.
Select the icon Configure located in the top right corner to setup your graphic.

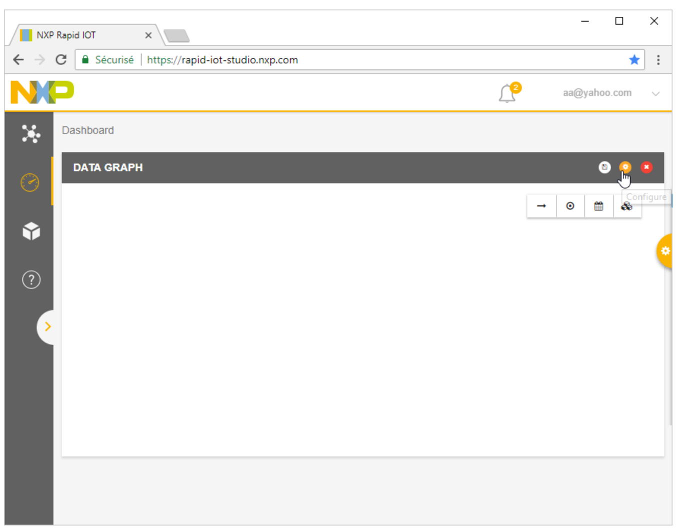


Figure 25. Configure Cloud Data Graph

First give a name to your graphic (ex. Sensors) and press Next.

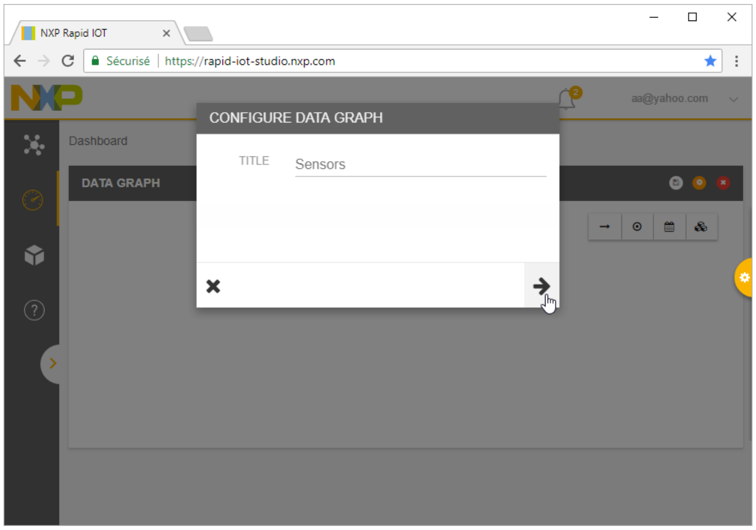


Figure 26. Name the Cloud Data Graph

Select the Rapid IoT application connected to the Cloud over Bluetooth via the phone App (ex. Out Of the Box Demo) and press Next.

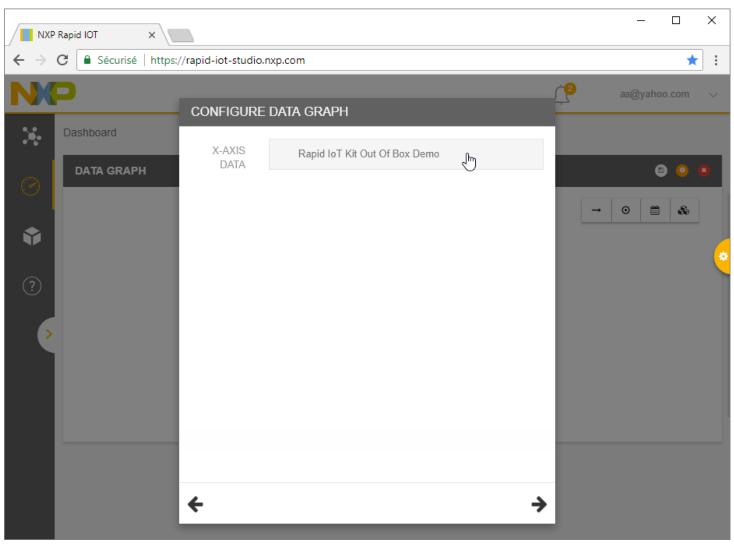


Figure 27. Select Cloud Data

Select the X-Axis data for your graphic (ex. _timestamp) and press Next.

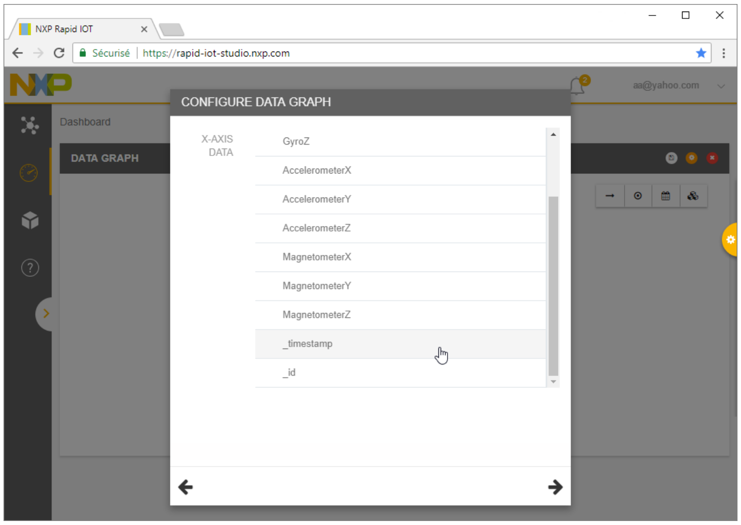


Figure 28. Configure X-Axis for Data Graph

Now select the Y-Axis data or Sensors that you want to display (multiple selection is allowed) and press Submit.

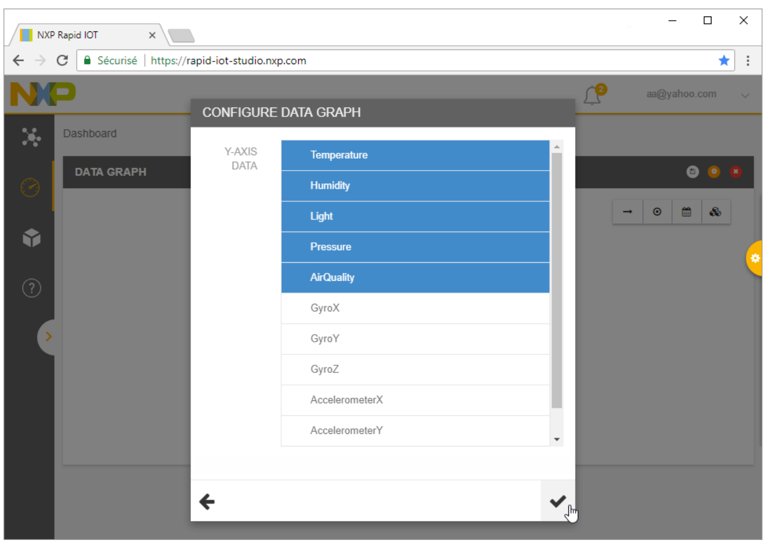


Figure 29. Configure Y-Axis/Sensor for Data Graph

The graphic named Sensors in your Dashboard will now display and update automatically the value of the selected sensors from the Cloud.

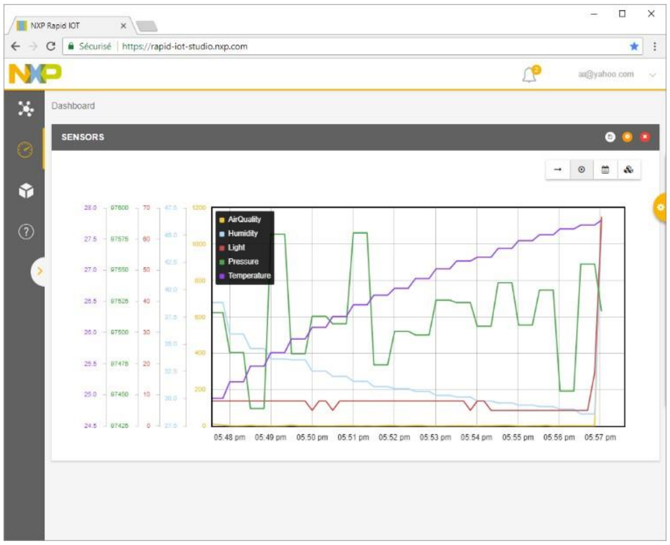


Figure 30. Sensor Graph Data from the Cloud

Congratulation you now successfully monitor your Rapid IoT from the Cloud!!

2.2. Get started with Rapid IoT Studio

2.2.1. Registration and Connection

Visit the dedicated Rapid IoT Studio portal at: http://rapid-iot-studio.nxp.com

You will need a free NXP account to access Rapid IoT Studio.

If you already have an account press LOGIN NXP SSO and you will be automatically redirected to the NXP Sign In page (NXP employees use your NXP WIB).

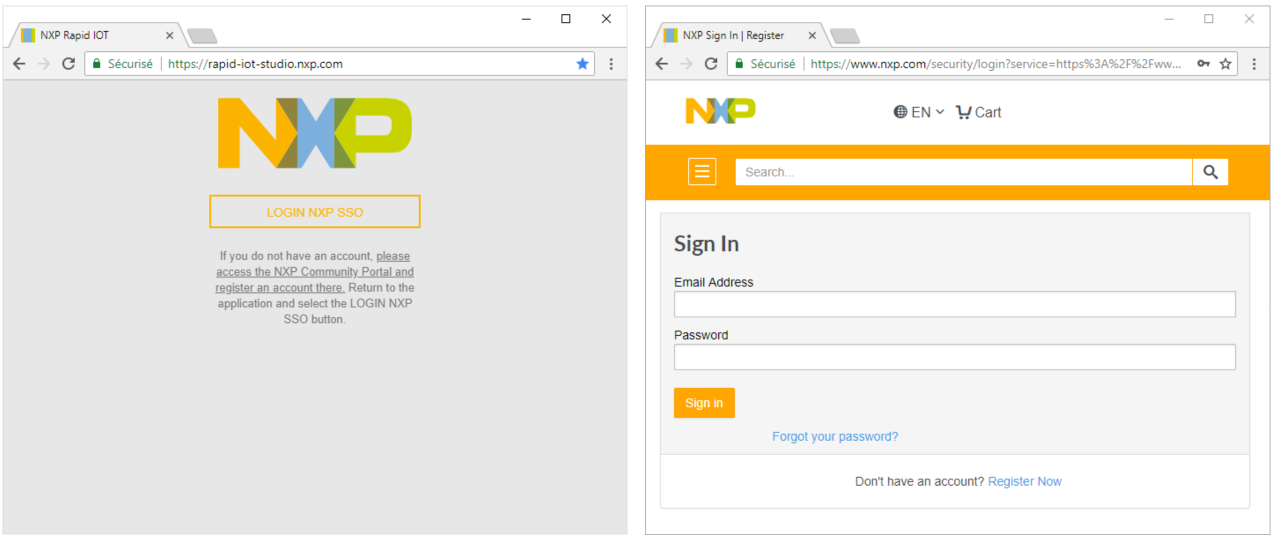


Figure 31. Sign-in/up with Rapid IoT Studio portal

If you don't have an account yet simply select the link please access the NXP Community Portal and register an account there or the link Register Now from the NXP Sign In page.

Enter the requested information (First/Last Name, Email, Password, Country and Company Name).
Check the box to confirm that you have read and accept the Term of Use and Privacy Policy and the box to confirm that you are not a Robot.
Finalize your registration by pressing the button Register.

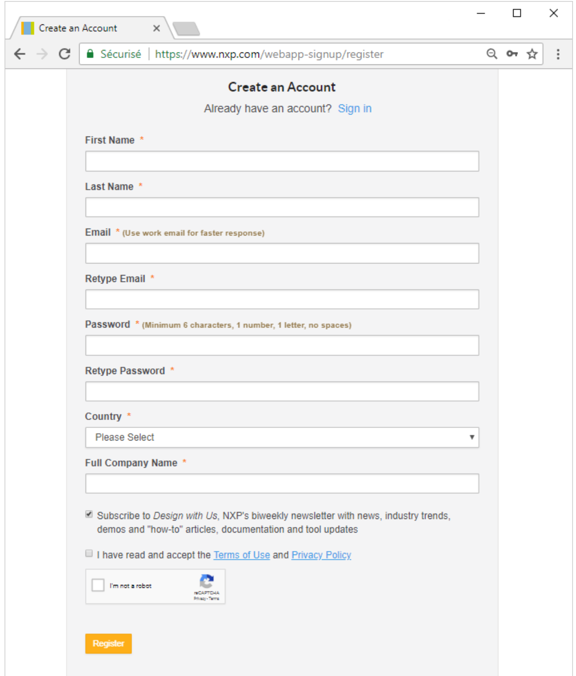


Figure 32. Create a free NXP account

Visit the mailbox used during the registration and open the email sent by:
nxpupdt1@contact.nxp.com

Click on the enclosed link to verify your email.
NOTE: Check the folder Junk/Spam/Clutter, if the confirmation email doesn't show up in your inbox.

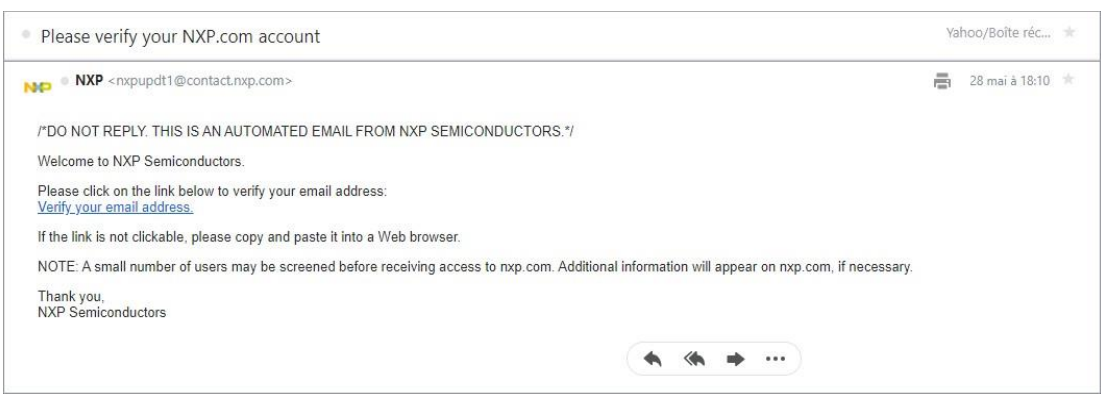


Figure 33. Confirmation Email

You are now ready to login with your NXP account.
Press the button LOGIN NXP SSO.
Enter you Email Address and Password and press Sign In.

At the first connection to the Rapid IoT phone App with your SSO account a window with the User Agreement will appear.
Check the box Accept Term of Use and Privacy Policy and press SUBMIT.

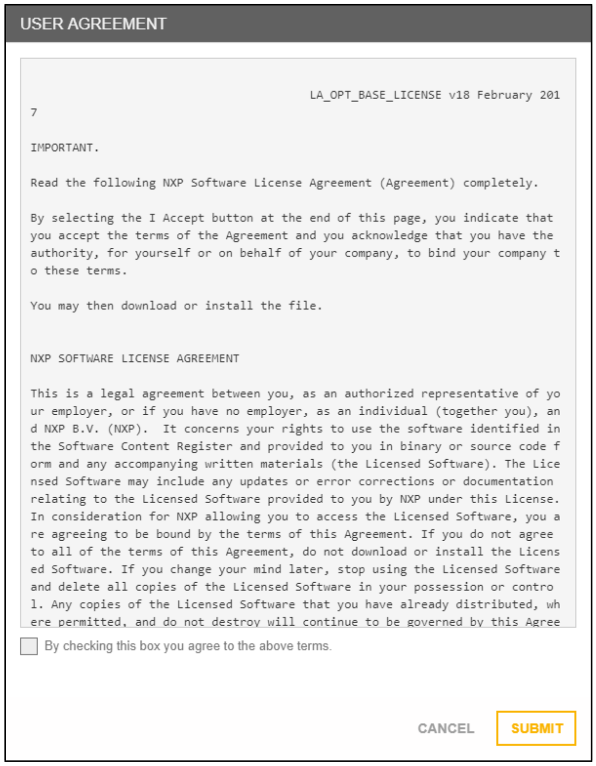


Figure 34. Accept Term of Use and Privacy Policy

Congratulation, you are now officially connected to Rapid IoT Studio!!

2.2.2. Open a Rapid IoT Studio Example

We will now open, build and program a Rapid IoT Studio example.

IMPORTANT: Applications build/programmed with Rapid IOT Studio are private applications. User will be able to connect their board over BLE only using the same user-account to login to the phone App.

The Dashboard view will open automatically.
Open the left panel by selecting the icon located in the left edge.

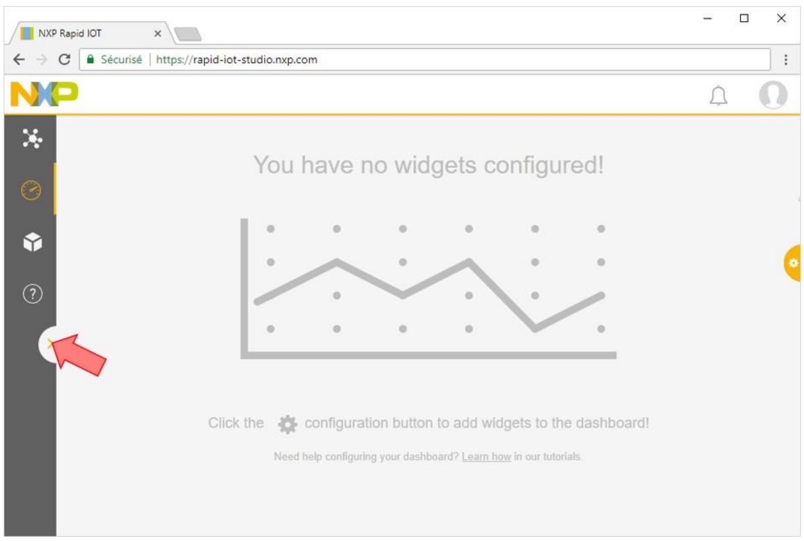


Figure 35. Dashboard view

Select the option Studio to launch the online programming interface.

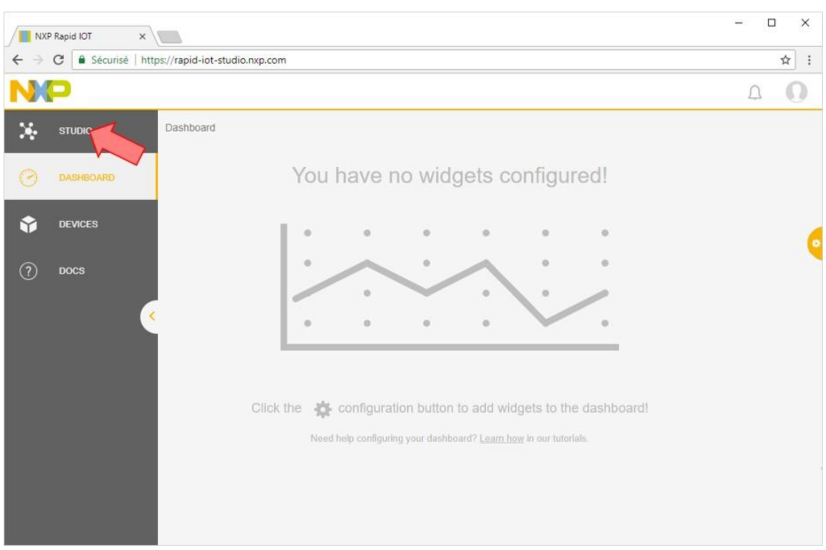


Figure 36. Studio Shortcut

Select the option EXAMPLES to display the project examples available online.

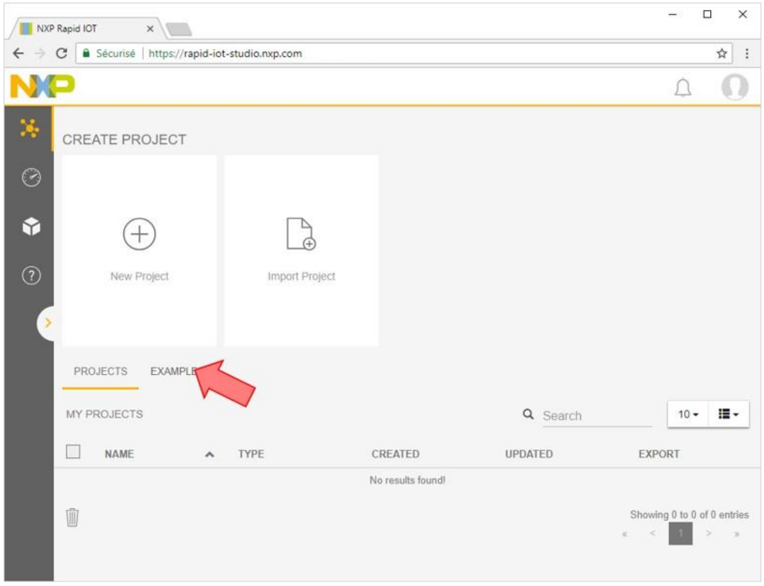


Figure 37. Example Tab

Let’s begin by opening a simple blinky example.
Select Rapid IoT Blinking an LED from the project example list.

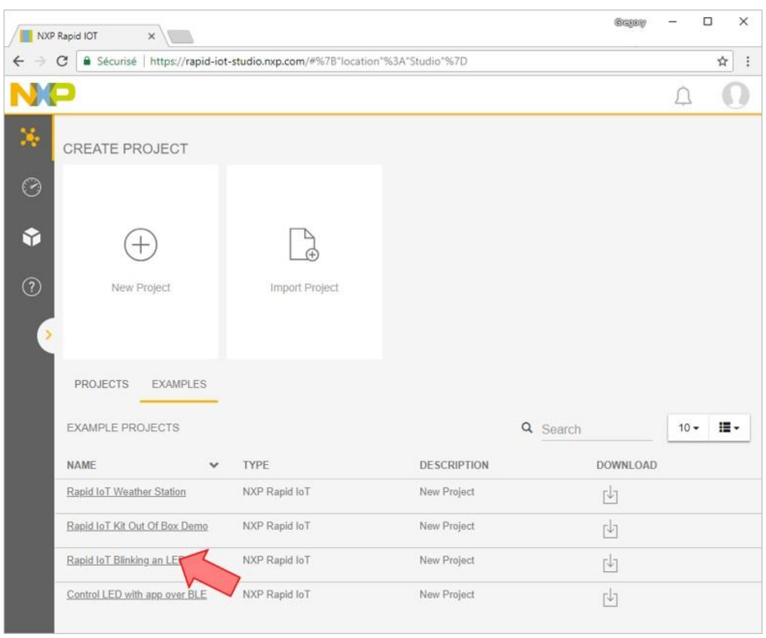


Figure 38. Blinky Example

The Embedded view will automatically open.

The project Blinking an LED requires few Elements: one interval, one RGB LED and one Display.

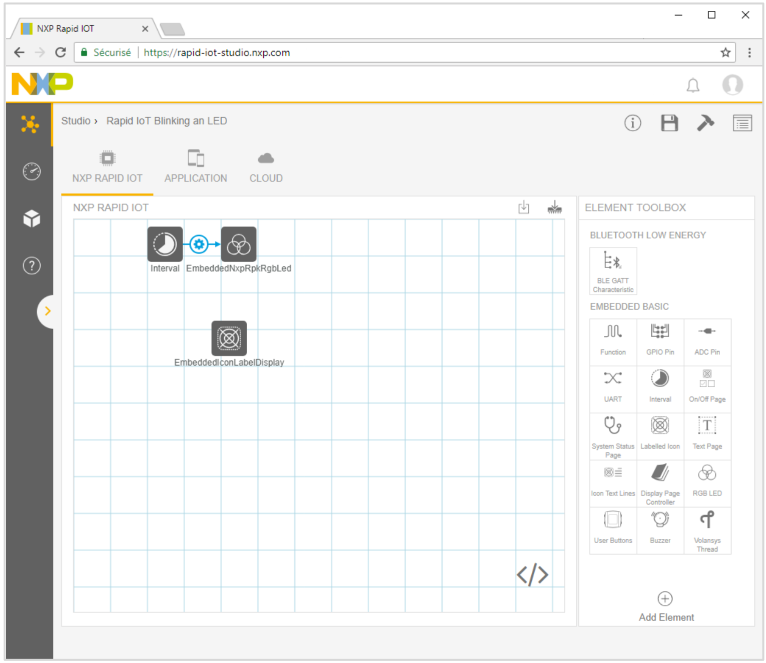


Figure 39. Embedded View for Blinking an LED Project Example

You can review the parameters of each Element by selecting them from the workspace.

For example, the Interval Element emulates a timer configured with a period of 1,000 millisecond.

The Interval Element is connected to the RGB LED that it toggles Red.

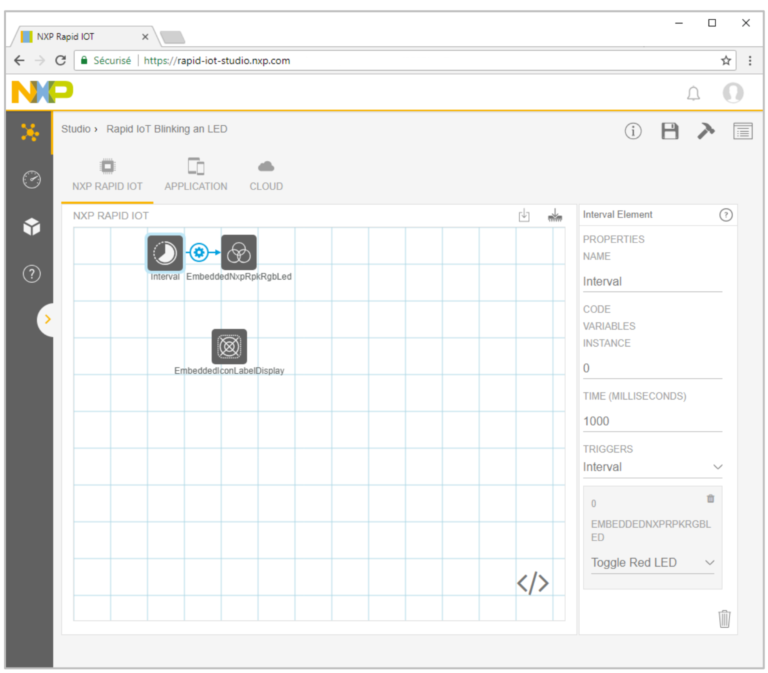


Figure 40. Interval Element

The Element EmbeddedIconLabelDisplay will display on the Rapid IoT screen an LED icon subtitled with the text Blinking an LED.

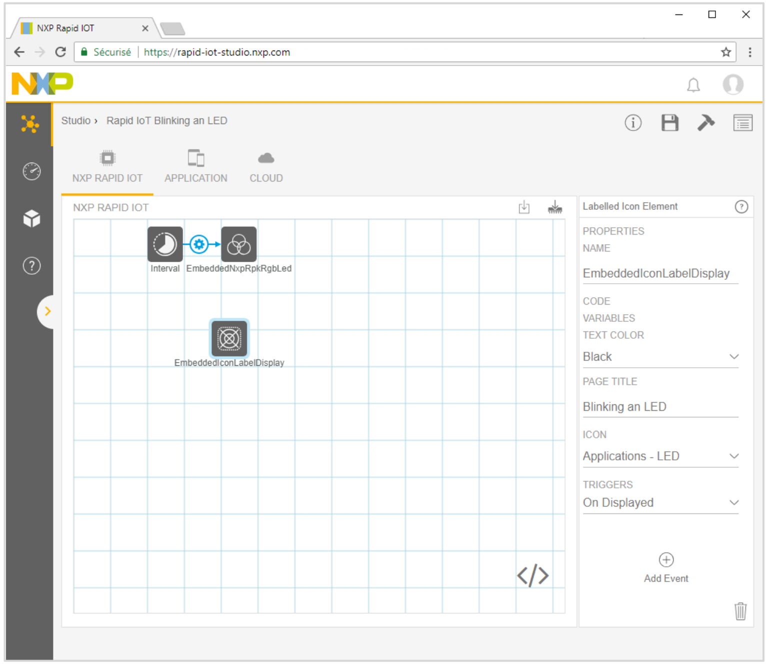


Figure 41. Display Element

Before launching any Build or Programming, please make sure to save first your project. Select the icon Save from the toolbar located in the top right corner to save your modifications.

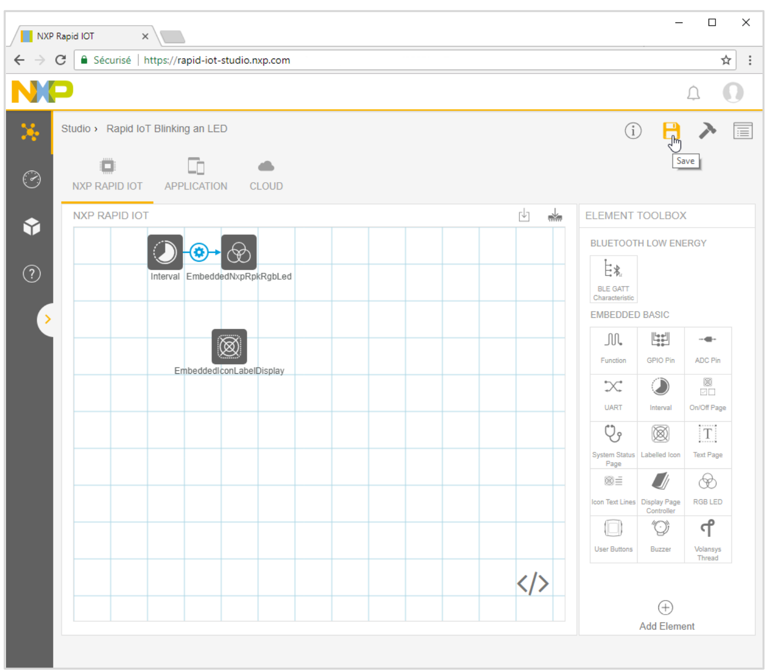


Figure 42. Save Project

Select the icon Compile from the toolbar located in the top right corner to compile your project.

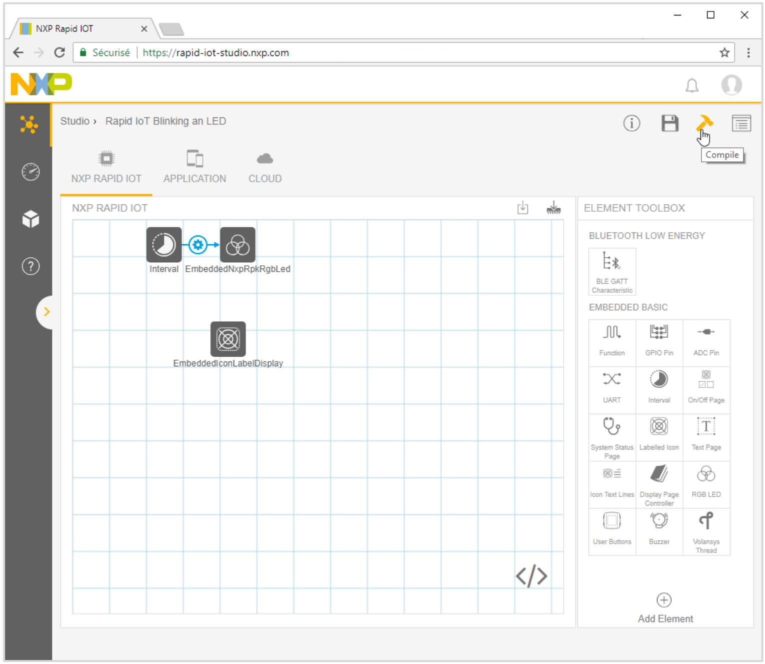


Figure 43. Compile Project

The workspace top orange bar will fill-in to indicate the progression of the compilation. The first time a project takes approx. 30 seconds to compile.

A blue popup window will indicate when the compilation ended up successfully.

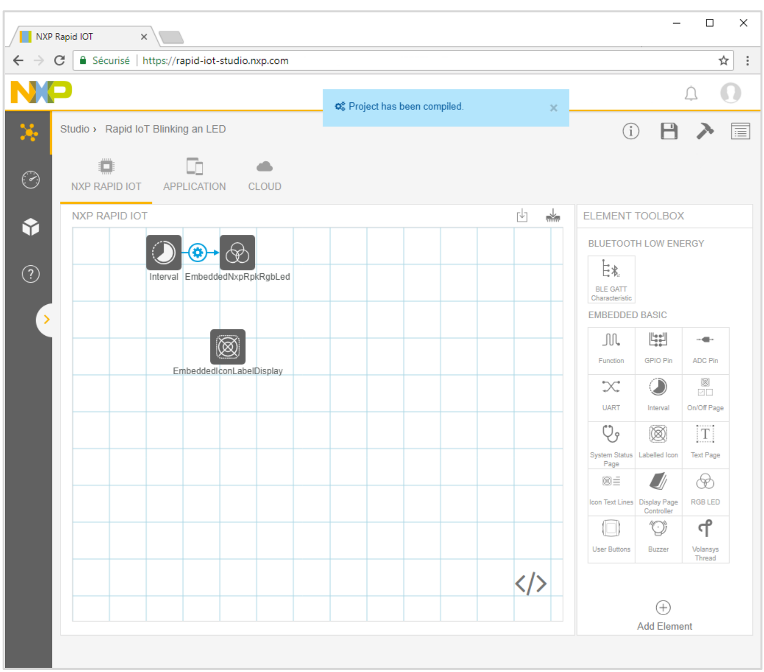


Figure 44. Compilation Successful

Select the icon ProgramFirmware from the Embedded View toolbar to build your project.
The file Rapid IoT Blinking an LED firmware.bin will be automatically downloaded by your web browser.

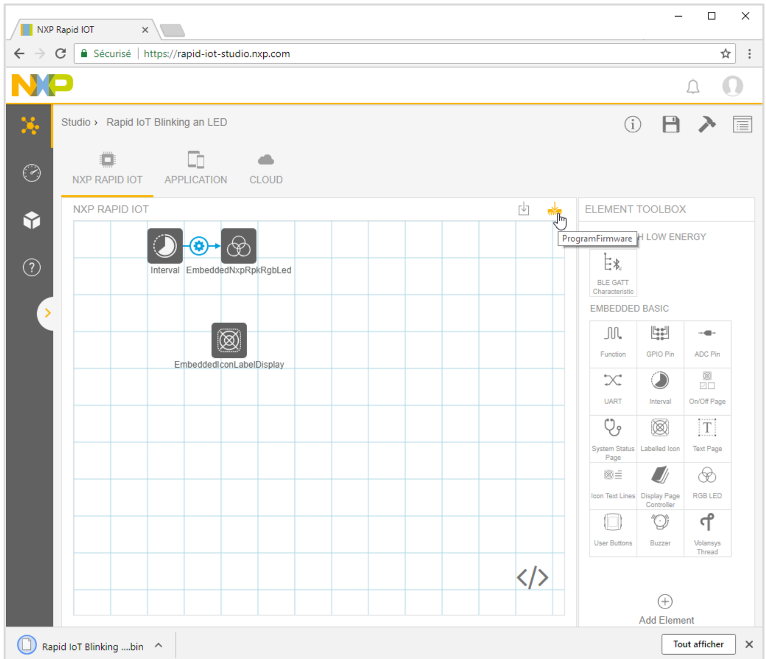


Figure 45. Build, Generate and Download Firmware

SLN-RPK-NODE Application and Wireless MCUs are pre-programmed in factory with a Bootloader to easily update their application through the onboard USB connector. To reprogram K64F internal flash with the new Rapid IoT Blinking an LED application, simply follow those steps:

- Connect one end of the provided USB cable to the computer and the other end to the micro USB type-B connector of the SLN-RPK-NODE,
- Keep SW3 button pressed while pushing shortly SW5/Reset button,
- Wait 1-2s for RGB LED to blink Green then release SW3 button.

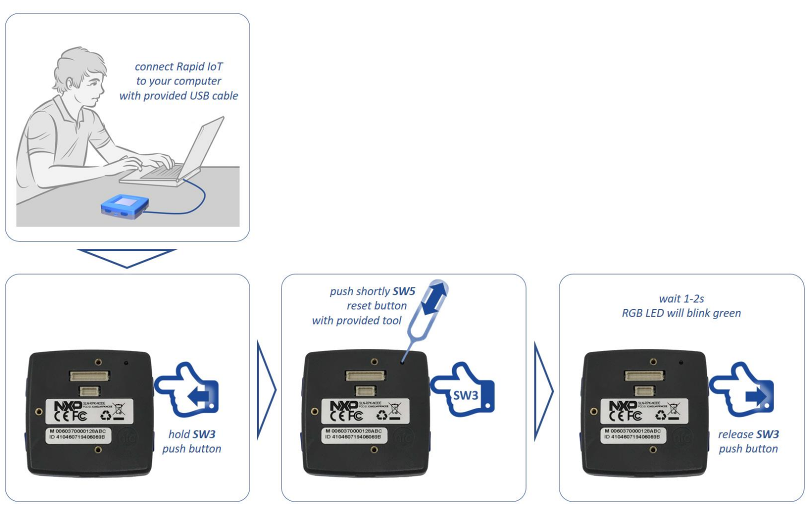


Figure 46. Instructions for USB Mass Storage Device Programming

RGB LED will blink green and your computer will detect a new Mass Storage drive and automatically install the appropriate drivers

- From your computer file explorer, drag-n-drop or copy-paste into the Mass Storage drive the file Rapid IoT Blinking an LED firmware.bin generated by Rapid IoT Studio available from your download folder

Figure 47. Instructions for pushing a new application through USB

Bootloader will automatically identify the MCU target to reprogram thanks to the binary file signature.
RGB LED will blink purple during download and blink blue during serial flash programming.
RGB LED will blink green during K64F internal flash (re)programming with the new application (read from Serial Flash memory) and automatically reset, when ready.

Figure 48. USB Programming LED

RGB LED	BLUE LED	WHITE LED
PURPLE BLINK	OFF	OFF
BLUE BLINK	OFF	OFF
GREEN BLINK	OFF	OFF

Table 3. K64F USB Programming LED Sequence

Congratulation, you have now officially compiled and programmed a Rapid IoT Studio example on your board!!

Your project is stored online by Rapid IoT Studio.
You can access it from any computer with an internet connection logging-in with your user-account.

But you can also save your project configuration Offline.
From the Studio Project Manager view select the icon Export next to your project.
The file Rapid IoT Blinking an LED.atmo will be automatically downloaded by your web browser.

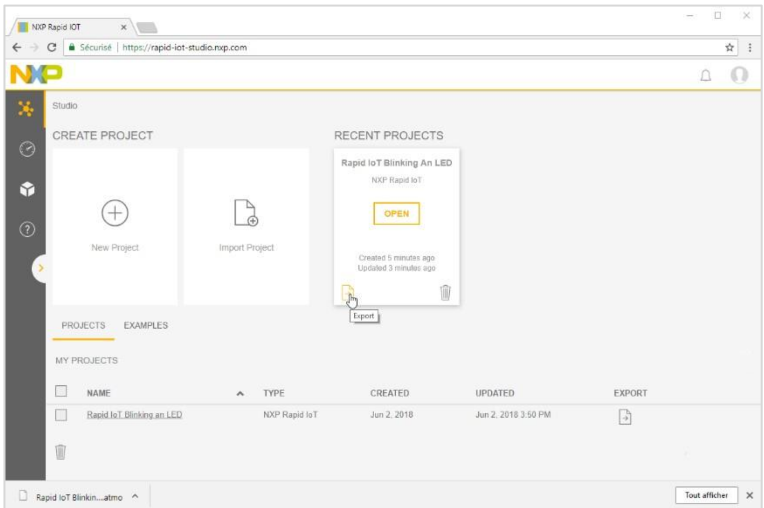


Figure 49. Save/Export Project Offline

Then you can easily import your project to Rapid IoT Studio.

From the Studio Project Manager view select the icon Import Project.
Select the file Rapid IoT Blinking an LED.atmo from the browser window and press OK.
Your project will now be automatically added to your Online library.

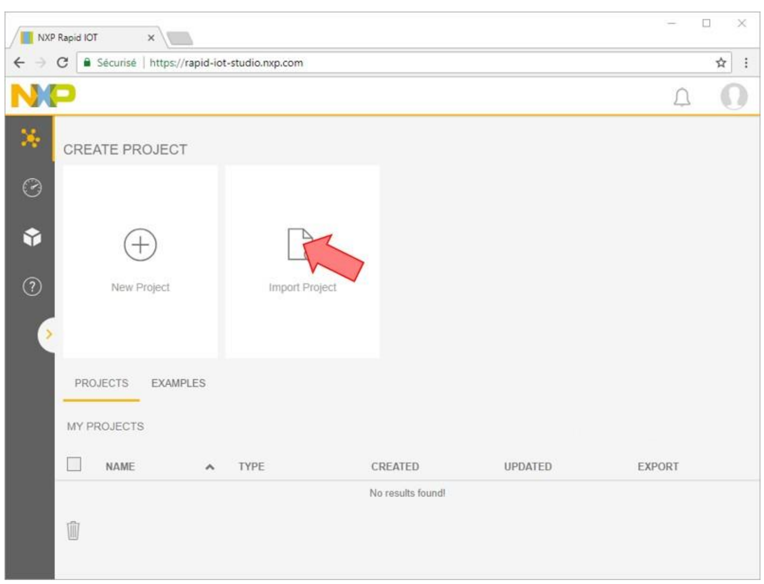


Figure 50. Import Project Online

You can also export your Rapid IoT Studio project to debug it with MCUXpresso IDE.
Open your project and select the icon Download from the Embedded View toolbar.
The file Rapid IoT Blinking an LED source.zip will be automatically downloaded by your web browser.

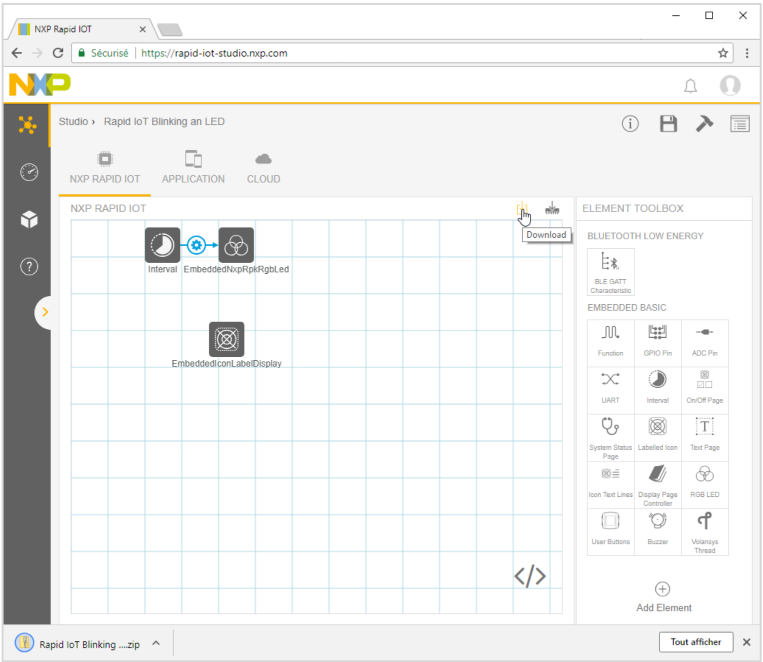


Figure 51. Export Project to MCUXpresso

2.3. Get Started with MCUXpresso Tool suite

2.3.1. MCUXpresso IDE

MCUXpresso IDE brings developers an easy-to-use Eclipse-based development environment for NXP's microcontrollers based on Arm® Cortex®-M cores. It offers advanced editing, compiling and debugging features with the addition of MCU-specific debugging views, code trace and profiling, multicore debugging, and integrated configuration tools. Its debug connections support every NXP evaluation boards including the Docking station of the Rapid IoT Prototyping tool with industry-leading open-source and commercial debug probes from ARM®, P&E Micro® and SEGGER®

To download for free NXP MCUXpresso IDE go online at: www.nxp.com/MCUXpresso
Select from the PRODUCTS tab MCUXpresso IDE.
Go to DOWNLOADS tab and select the LATEST VERSION of the tool.

You will be asked to sign-in/up with a free NXP user-account.

When MCUXpresso installer download completes, double click on the executable, follow the install instruction and keep the default options.

Launch MCUXpresso IDE and define the Workspace location where you will copy and store your projects (default C:\MCUXpresso.Workspace) and press OK.

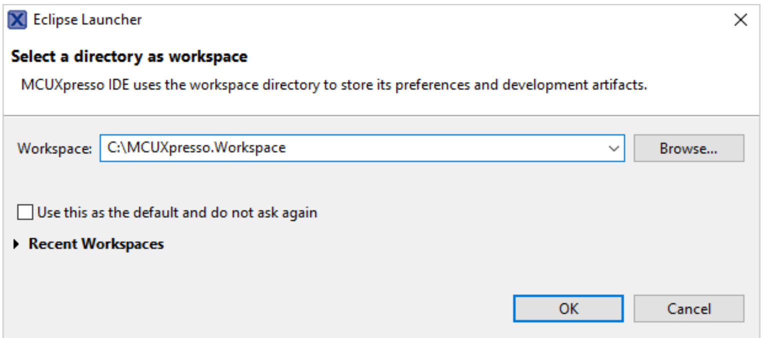


Figure 52. MCUXpresso IDE Workspace

Congratulation you successfully installed the free NXP MCUXpresso IDE on your computer!!

2.3.2. Software Development Kit (SDK)

MCUXpresso SDK is a comprehensive software enablement package designed to simplify and accelerate application development with NXP's microcontrollers based on Arm® Cortex®-M cores. The MCUXpresso SDK includes production-grade software with integrated RTOS (optional), integrated stacks and middleware, reference software, and more. It is available in custom downloads based on user selections of MCU, evaluation board, and optional software components.

To download for free Rapid IoT SDK package go online at: www.nxp.com/rapid-iot
Select from the PRODUCTS tab Rapid IoT prototyping kit.
Go to SOFTWARE & TOOLS tab and select the latest version of the Rapid IoT SDK package.

You will be asked to sign-in/up with a free NXP user-account.

When the download of the compressed file completes, extract it in your MCUXpresso Workspace on your computer HDD.

2.3.3. Import Rapid IoT SDKs

The Rapid IoT SDK is a composite SDK. It features two SDK target specific SDKs, for each target MCU in the Rapid IoT device.

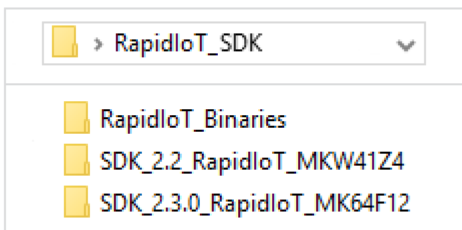


Figure 53. Rapid IoT SDK package

Before building the Rapid IoT SDK example projects, the target SDKs need to be imported into MCUXpresso IDE by drag and dropping each target SDK into the “Installed SDKs” window from MCUXpresso IDE.

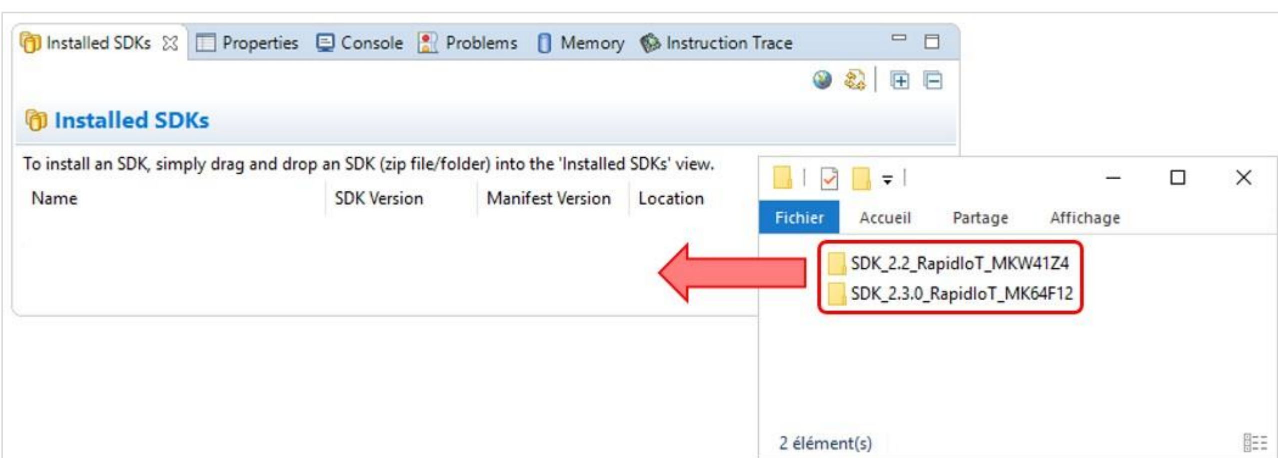


Figure 54. Rapid IoT target SDK install for MCUXpresso IDE

For each package a confirmation window will pop-up, select OK to validate.

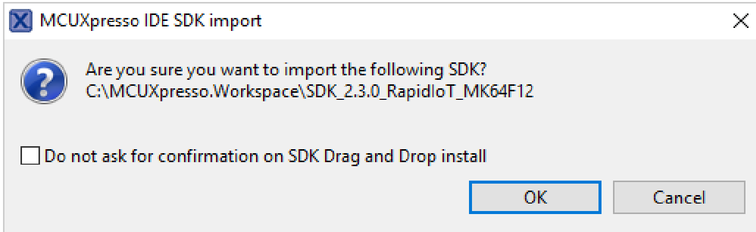


Figure 55. MCUXpresso SDK import confirmation window

Congratulation you successfully installed and configured Rapid IoT SDK on MCUXpresso IDE.

2.3.4. Import Rapid IoT Projects

The Rapid IoT SDKs require to import manually the projects unlike classical MCUXpresso SDKs. This feature (as well as the documentation) will be upgraded during the periodic updates.

To import Rapid IoT projects into MCUXpresso IDE, follow those steps:

From the Quickstart Panel, select the option Import project(s) from file system…

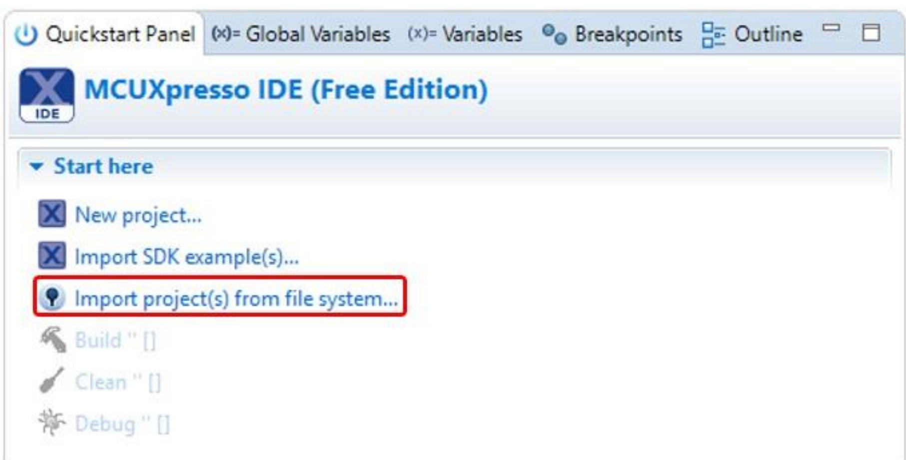


Figure 56. MCUXpresso Quickstart Project Import from File System

Enter (or browse to) the file system location of the intended target SDK package.

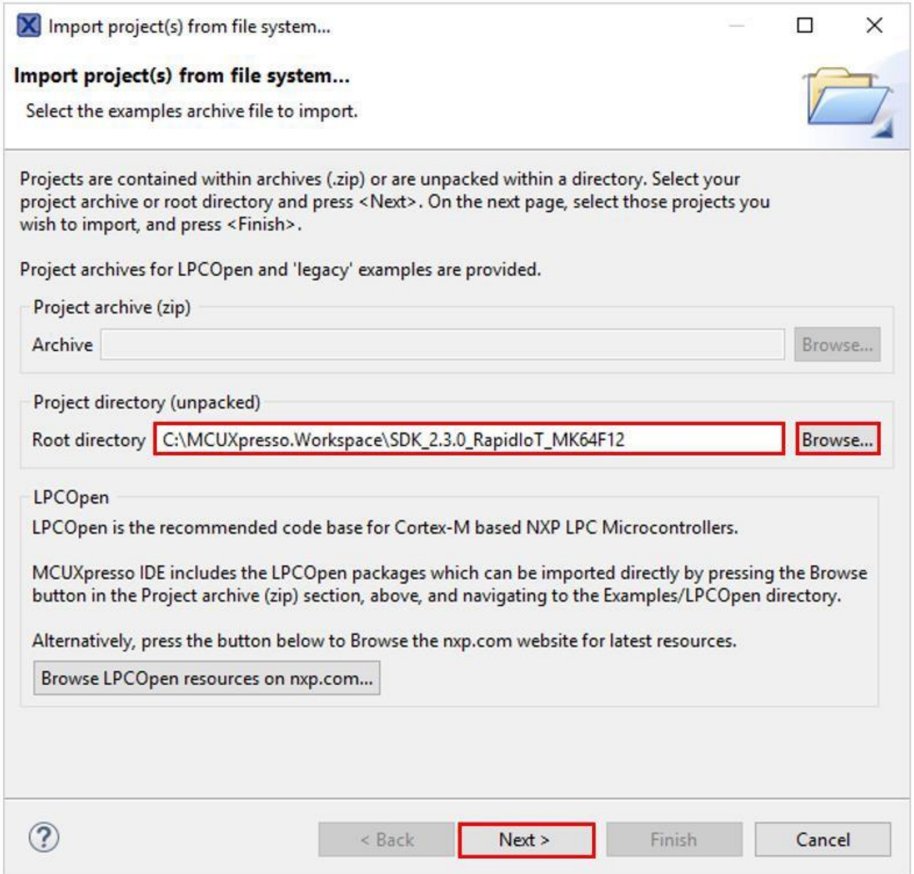


Figure 57. MCUXpresso root directory selection

Select the projects by pushing Select All button (NOTE: the RapidIot_Base project must be selected if the Atmosphere_Project is selected) and make sure that the box Copy projects into workspace is unchecked, then press the button Finish.

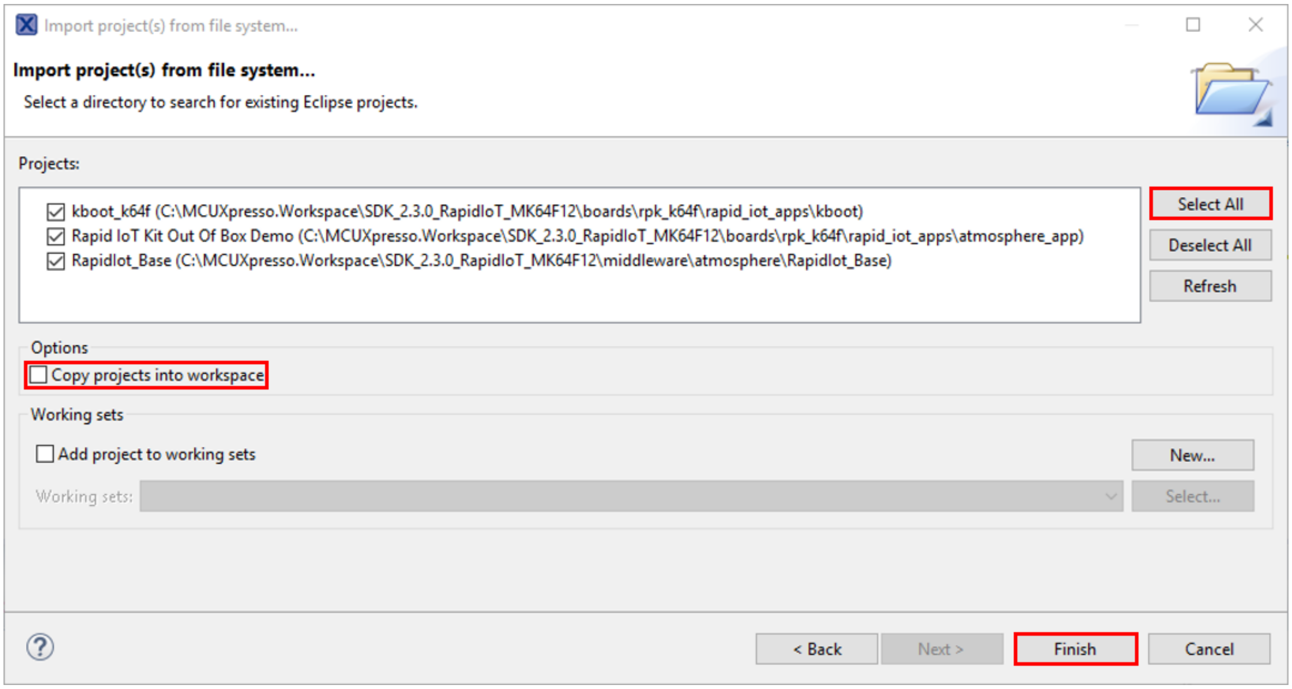


Figure 58. MCUXpresso project import selection

The projects will now appear in the Project Explorer window.

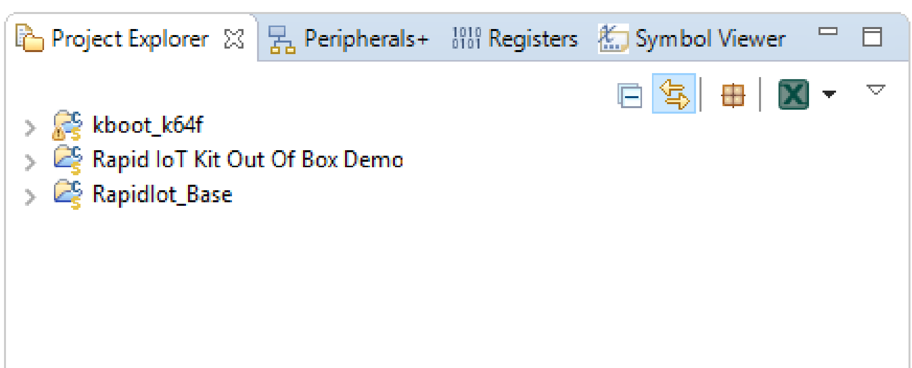


Figure 59. MCUXpresso Project Explorer

2.3.5. Build a Rapid IoT Project

Select from the Project Explorer window the project that you want to compile.
NOTE: You must build first the RapidIot_Base project before building the Atmosphere_Project.

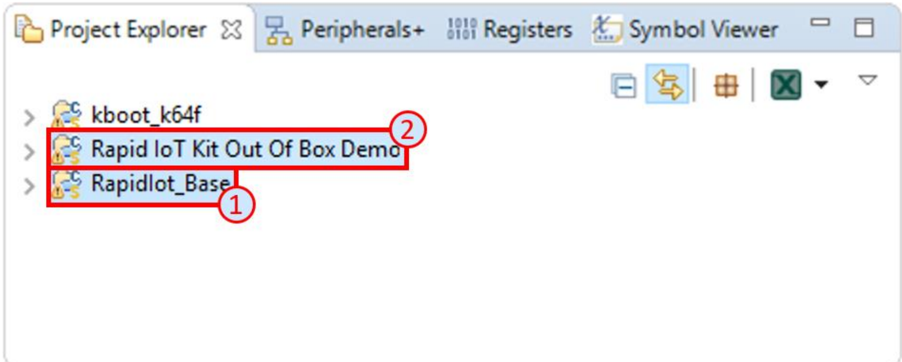


Figure 60. Project Selection

From the Quickstart Panel, select the option Build ‘Atmosphere_Project’ [Debug].

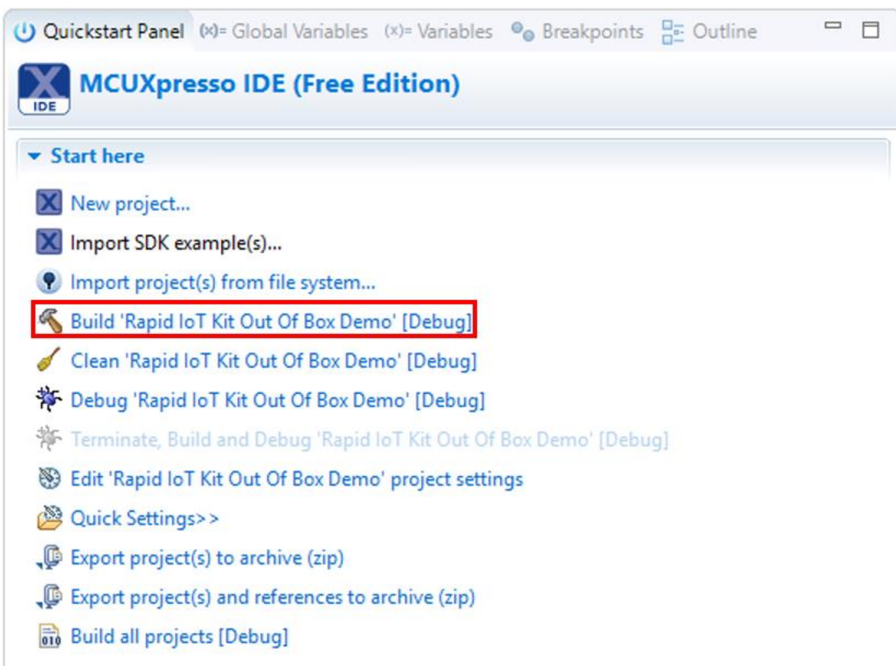


Figure 61. Project Build

Wait for the Console window to confirm that the compilation ended up successfully (approx. 2min for the first time to compile the Atmosphere_Project).

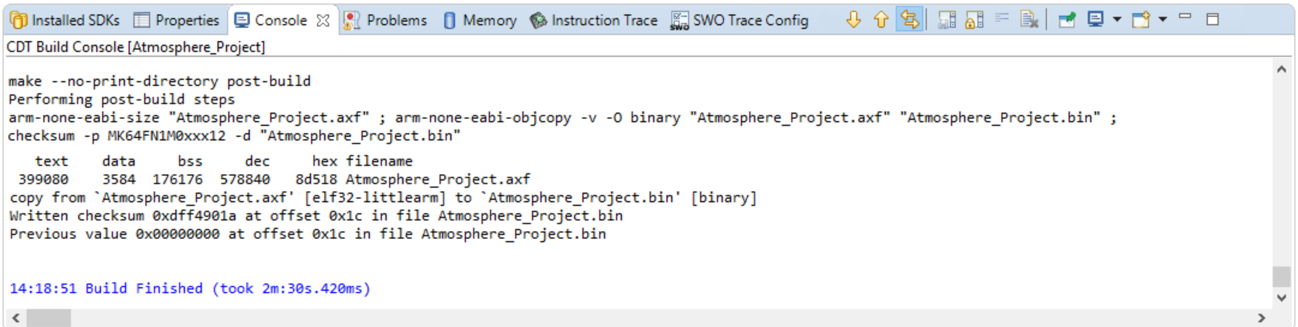


Figure 62. Console Window

Congratulation, you successfully built your Rapid IoT project with MCUXpresso IDE!!

2.3.6. Debug a Rapid IoT Project

Select from the Project Explorer window, the same project that you just compiled successfully.

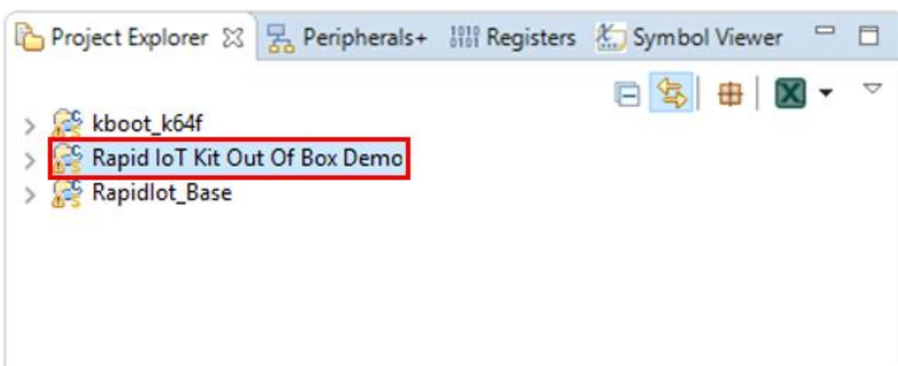


Figure 63. Project Selection

IMPORTANT: The Docking station shipped by MikroElektronika is preprogrammed in factory with an OpenSDA DAP-LINK application dedicated for the MikroElektronika Hexiwear platform. User should reprogram (once) the OpenSDA application of their Docking station before programming or debugging their Rapid IoT board.

To download for Docking station DAP-LINK for Rapid IoT go online at: www.nxp.com/rapid-iot
Select from the PRODUCTS tab Rapid IoT prototyping kit.
Go to SOFTWARE & TOOLS tab and select Docking Station DAP-LINK for SLN-RPK-NODE.
You will be asked to sign-in/up with a free NXP user-account.
When the download of the file completed, copy the Docking Station DAP-LINK firmware in your MCUXpresso Workspace on your computer HDD.

To reprogram the OpenSDA Application from the MikroElektronika Docking-station follow those steps (Rapid IoT board should not be mounted to the Docking station):

- Set the Power switch of the Docking station (red below) to OFF.
- Set the jumper switches of the Docking station to 11001000.
- Connect your Docking station to the USB port of your computer.
- Keep MK64 RESET button of the Docking station pressed, while setting the Power switch of the Docking station to ON, wait 2 seconds then release the MK64 RESET button.

Your computer will detect a new Mass Storage drive named MAINTENANCE and automatically install the appropriate drivers.

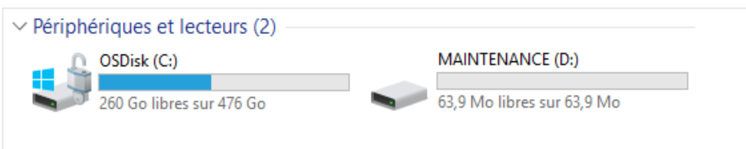


Figure 64. MAINTENANCE MSD drive

- From your computer file explorer, drag-n-drop or copy-paste into the MAINTENANCE drive the binary file k20dx_rapid_iot_if_crc_legacy_0x8000.bin.

Wait for the download to complete.

- Set the Power switch of the Docking station (red below) to OFF.
- Set the Power switch of the Docking station (red below) to ON.

Your computer should detect new Mass Storage drive named IOT-DAPLINK and automatically install the appropriate drivers.

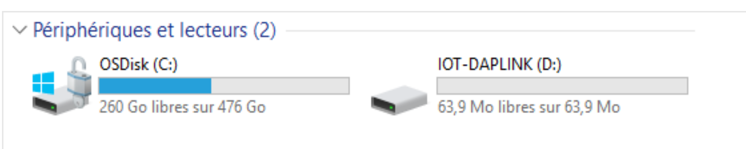


Figure 65. IOT-DAPLINK MSD drive

Congratulation your MikroElektronika Docking-station is now ready to program, serial-monitor and debug safely both MCU targets of your Rapid IoT board!!

IMPORTANT: Before connecting Rapid IoT to the MikroElektronika Docking-station or changing the Jumper Switches make sure to set the Power switch of the Docking station (red below) to OFF.

Plug your Rapid IoT to the Docking station aligning carefully the board to board connectors (yellow below).

Figure 66. Docking station connector

The OpenSDA circuitry embedded in the Mikroelektronika Hexiwear Docking-station (MIKROE-2094) can program, serial-monitor and debug alternately the K64F and the KW41Z target MCUs featured in the Rapid IoT.

Set the jumper switches of the Docking station to 11001000 to program or debug the Application/K64F controller.

Figure 67. Docking Station Jumper Switch configuration to debug K64F

Set the jumper switches of the Docking station to 00111000 to program or debug the Wireless/KW41Z controller.

Figure 68. Docking Station Jumper Switch configuration to debug KW41Z

From the Quickstart Panel, select the option Debug ‘Atmosphere_Project’ [Debug].

Figure 69. Project Debug

A configuration window should popup to ask you to select your debug Probe.

If your Docking station has been properly programmed with the DAP-LINK application for the Rapid IoT (see Docking station programming in the Hardware section), a CMSIS-DAP probe should be detected.

Otherwise check the box MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes and press Search again.

When your probe is properly detected, simply select it from the list of the Available attached probes, check the box Remember my selection (for this Launch configuration) to avoid the same window to popup at the next debug and press OK.

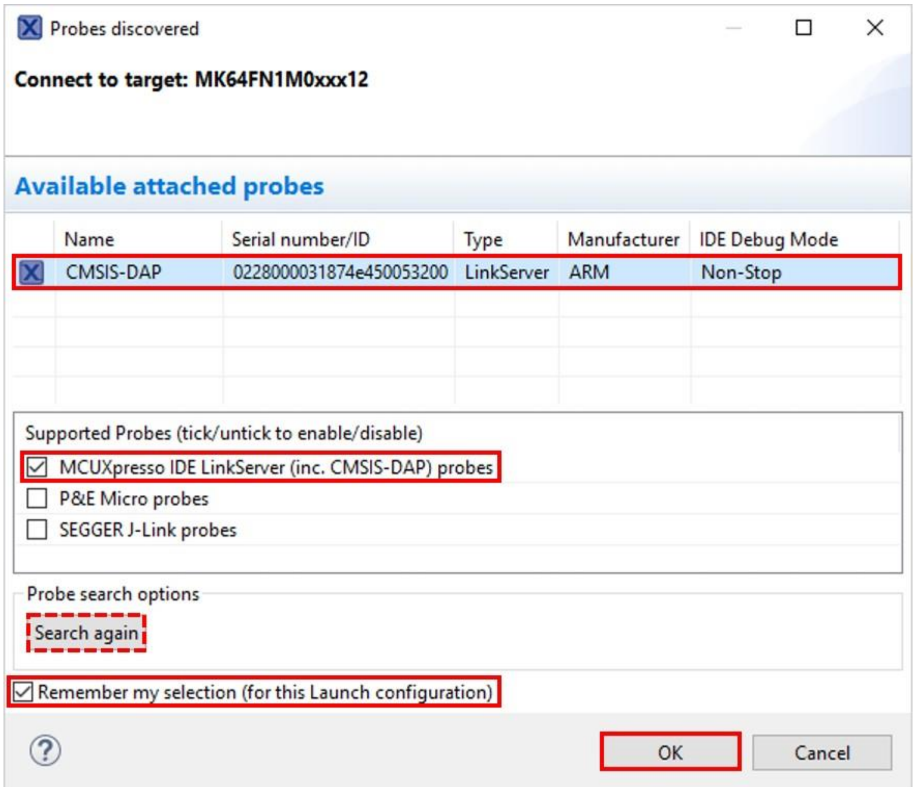


Figure 70. Probe Selection Window

Wait for MCUXpresso IDE to switch to the Debug view and press the Resume button from the toolbar.

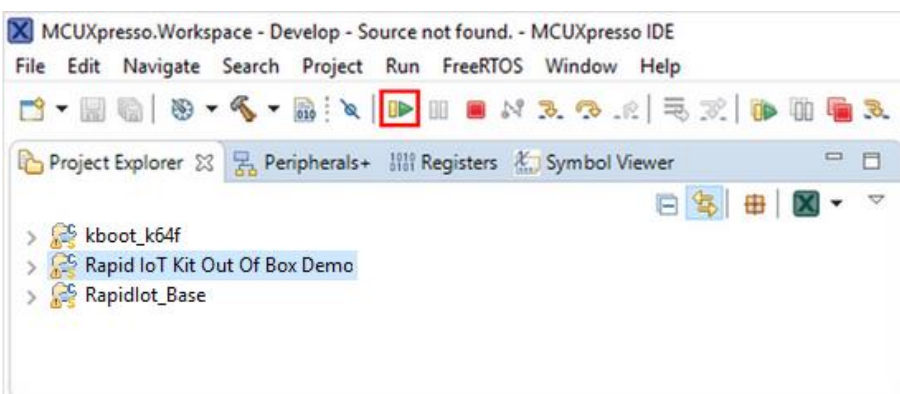


Figure 71. Debug View

Congratulation, you successfully debugged your Rapid IoT with MCUXpresso IDE!!

3. Rapid IoT Hardware Architecture

3.1. Hardware Overview

The features of the SLN-RPK-NODE hardware are as follows:

- MK64FN1M0VMD12 device in 144 BGA (Main MCU)
- MKW41Z512VHT4 device in 48 LQFN (Wireless MCU) with 2.4GHz chip RF antenna
- Accelero-/Magneto-meter FXOS8700CQ with digital output
- Gyroscope FXAS21002CQ with digital output
- Pressure barometric/altitude Sensor MPL3115A2 with digital output
- Temperature and Humidity Sensor with digital output
- Ambient Light Sensor with digital output
- Air Quality Sensor with digital output
- Low-power 1.28" color display with 176x176 resolution and SPI controller
- NOR Flash 128Mbit with SPI interface for recovery, update and data logging
- Security chip A1006 for anti-counterfeit
- NFC TAG I2C plus NT3H2211 with 2KB memory and flexible 13.56MHz antenna
- Four capacitive touch electrodes around display with Serial Touch Screen Controller
- RGB LED for user interface
- Blue and White LED for radio status
- Micro-buzzer for sound feedback and user interface
- Reset and four User Push Buttons
- Ultra-low Power PCF2123 Calendar RTC with Alarm function and SPI Interface
- Single Voltage Domains VDD3V2 with multiple power-supply schemes
- LiPo battery 240mAh with MC34671 fast USB battery charger
- Multi role USB interface with micro-B USB connector
- 50-pin Board to board connector compatible with Hexiwear Docking Station
- 20-pin Board to board connector for next-gen accessories

Figure below shows the block diagram of the SLN-RPK-NODE design:

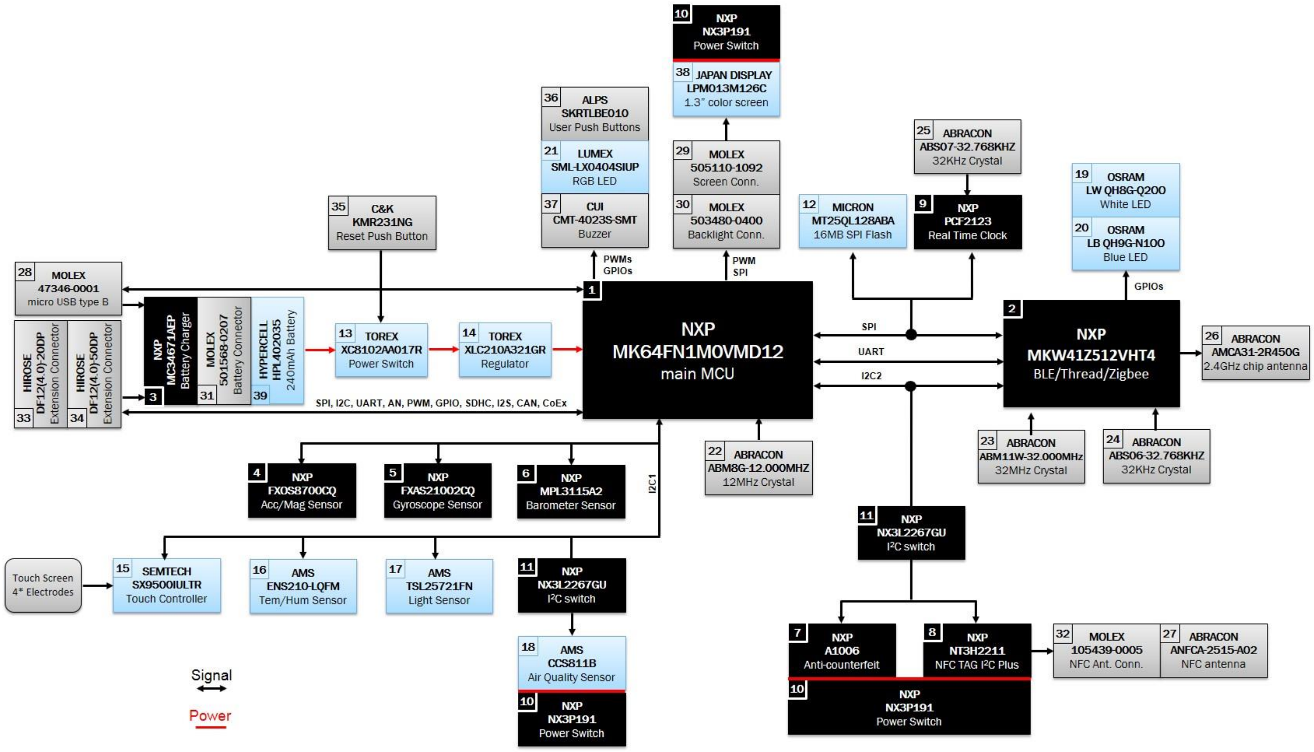


Figure 72. SLN-IOT-NODE block diagram

Reference	Supplier	Part-Number
1	NXP	MK64FN1M0VMD12
2	NXP	MKW41Z512VHT4
3	NXP	MC34671AEPR2
4	NXP	FXOS8700CQR1
5	NXP	FXAS21002CQR1
6	NXP	MPL3115A2R1
7	NXP	A1006UK/TA1NXZ
8	NXP	NT3H2211W0FHK
9	NXP	PCF2123BS/1,518
10	NXP	NX3P191UK,012
11	NXP	NX3L2267GM,115
12	MICRON	MT25QL128ABA1EW7-0SIT
13	TOREX	XC8102AA017R-G
14	TOREX	XCL210A321GR-G
15	SEMTECH	SX9500IULTRT
16	AMS	ENS210-LQFM
17	AMS	TSL25721FN
18	AMS	CCS811B-JOPR5K
19	OSRAM	LW QH8G-Q200-3K5L-1
20	OSRAM	LB QH9G-N100-35-1

Reference	Supplier	Part-Number
21	LUMEX	SML-LX0404SIUPGUSB
22	ABRACON	ABM8G-106-12.000MHz-T
23	ABRACON	ABM11W-32.000MHz-8-D1G
24	ABRACON	ABS06-32.768KHZ-T
25	ABRACON	ABS07-32.768kHz-7-1
26	ABRACON	AMCA31-2R450G-S1F-T
27	ABRACON	ANFCA-2515-A02
28	MOLEX	47346-0001
29	MOLEX	505110-1092
30	MOLEX	503480-0400
31	MOLEX	501568-0207
32	MOLEX	105439-0005
33	HIROSE	DF12(4.0)-20DP-0.5V(86)
34	HIROSE	DF12(4.0)-50DP-0.5V(86)
35	C&K	KMR231NG LFS
36	ALPS	SKRTLBE010
37	CUI	CMT-4023S-SMT
38	JAPAN DISPLAY	LPM013M126C
39	HYPERCELL	HPL402035-240mAh

Table 4. SLN-RPK-NODE build of material

The primary components and their placement on the hardware assembly are explained in the below figures:

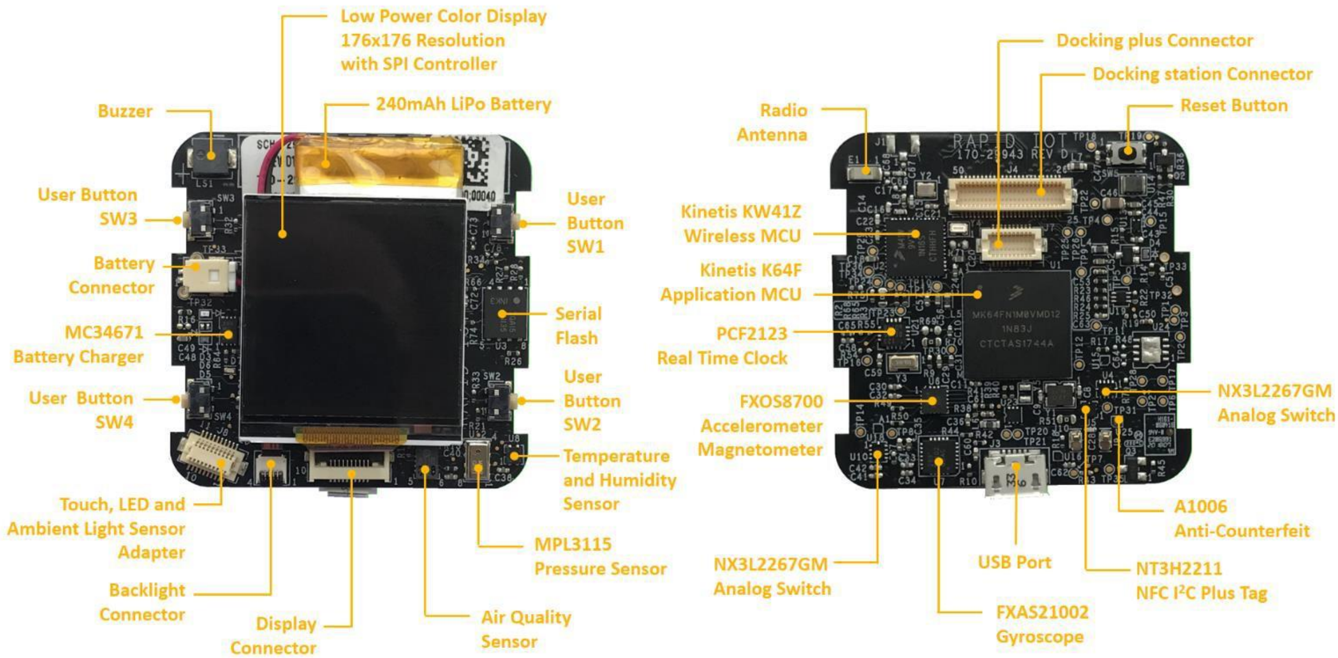


Figure 73. Rapid IoT main-board callout

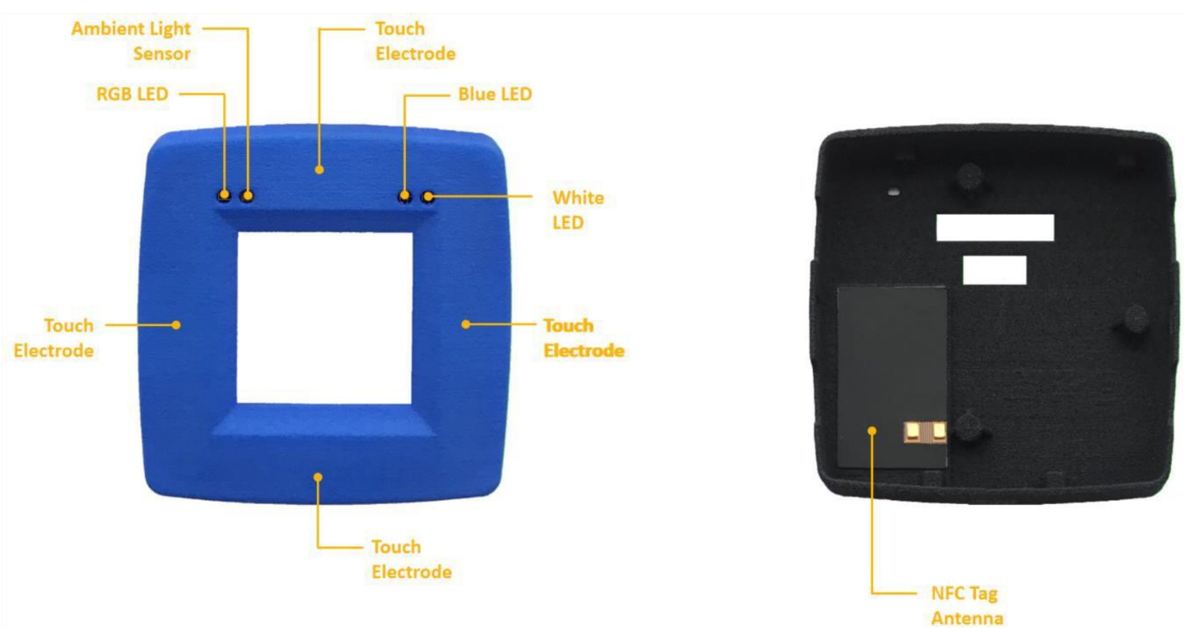


Figure 74. Rapid IoT touch-board and casing callout

3.2. Main Microcontroller

3.2.1. K64F features

The SLN-RPK-NODE features the MK64FN1M0VMD12 MCU, part of the Kinetis K6x family in 144 MAPBGA package capable to run up to 120MHz. The following table describes some of the features of the MK64FN1M0VMD12 MCU available on SLN-RPK-NODE hardware.

Features	Description
Performances	• 120 MHz ARM Cortex-M4 core with DSP instruction set, single precision floating point unit, single cycle MAC, and single instruction multiple data (SIMD) extensions • Up to four channel DMA for peripheral and memory servicing with reduced CPU loading and faster system throughput • Cross bar switch enables concurrent multi-master bus accesses, increasing bus bandwidth • Independent flash banks allowing concurrent code execution and firmware updating with no performance degradation or complex coding routines

Low-Power Capabilities	• 11 Low-Power modes with power and clock gating for optimal peripheral activity and recovery times.
	• Low-leakage wake-up unit. SLN-RPK-NODE can wake-up from Low-Leakage stop (LLS) and Very Low-Leakage stop (VLLS) modes with following sources:
	  ○ User Switch 1,
	  ○ KW41Z PTB0 I/O,
	  ○ FXOS8700CQ accelerometer interrupt,
	  ○ PCF2123 RTC interrupt.
	• Low-power timer for continual system operation in reduced power states

Flash and RAM Memory	• 1,024 KB Flash featuring fast access times, high reliability and four levels of security protection • 256 KB of RAM
Mixed-signal Capability	• 3 High-speed 16-bit ADC with configurable resolution accessible from the docking connector • 1 High-speed 16-bit ADC with configurable resolution used for battery level sensing

Timing and Control	• 1 Flex Timer dedicated to:
	  ○ RGB LEDs
	  ○ Display backlight
	  ○ Display VCOM
	• 1 Flex Timer dedicated to:
	  ○ Buzzer
	• 3 generic PWM outputs accessible from the docking connector
	• One low-power timer

Features	Description

Connectivity Interfaces	• Full-Speed USB device/host/on-the-go with device charge detect capability
	• Optimized charging current/time enabling longer battery life
	• 4 UART modules:
	  ○ UART0 dedicated to debug via the docking station
	  ○ UART2 and UART3 accessible from the docking connector
	  ○ UART4 is dedicated to inter-processor communication.
	• One Inter-IC Sound (I2S) serial interface for audio system interfacing accessible from the docking connector. It should be noted that the I2S interface is multiplexed with I2C2 bus
	• 3 DSPI modules:
	  ○ SPI0 accessible from the docking connector, with 3 distinct Chip Selects
	  ○ SPI1 dedicated to external NOR Flash and RTC (PCF2123)
	  ○ SPI2 dedicated to the Display
	• 3 I2C modules:
	  ○ I2C0 accessible from the docking connector
	  ○ I2C1 dedicated to the onboard sensors
	  ○ I2C2 dedicated to authentication chip (A1006) and NTAG (NT3H2211)
	• Secured digital host controller (SDHC) accessible from the docking connector
	• 1 FlexCAN module available on the connector J7

Reliability, Safety and Security	• Hardware encryption coprocessor for secure data transfer and storage. Faster than software • implementations and with minimal CPU loading. Supports a wide variety of algorithms - DES, 3DES, AES, MD5, SHA-1, SHA-256 • Memory protection unit provides memory protection for all masters on the cross-bar switch, • increasing software reliability • Cyclic redundancy check (CRC) engine validates memory contents and communication data, increasing system reliability • Independently-clocked COP guards against clock skew or code runaway for fail-safe applications, • such as the IEC 60730 safety standard for household appliances • Included in NXP product longevity program, with assured supply for a minimum of 10 years after launch

Table 5. Features of MK64FN1M0VMD12 used in SLN-RPK-NODE

3.2.2. K64F pin assignment

IC Pin	NET (schematic top)	K64 Function (Port)	IC Pin	NET (Schematic Top)	K64 Function (Port)
A1	SPI1_MISO	SPI1 (PTD7)	G1	-	VOUT33
A2	SPI1_MOSI	SPI1 (PTD6)	G2	VCC-USB	VREGIN
A3	SPI1_SCK	SPI1 (PTD5)	G3	DISP_BLIGHT	FTM3 (PTE12)
A4	SPI1_PCS0	SPI1 (PTD4)	G4	DISP_EXTMODE	GPIO (PTE11)
A5	RTC_INT	GPIO / LLWU_P12 (PTD0)	G5	VCC-MCU	VREFH
A6	UART3_RX	UART3 (PTC16)	G6	GND	VREFL
A7	NTAG_EN	GPIO (PTC12)	G7	GND	VSS
A8	RGB_R	FTM3 (PTC8)	G8	GND	VSS
A9	MB1_SPI_CS	SPI0 (PTC4)	G9	NTAG_I2C_EN	GPIO (PTB5)
A10	NC	NC	G10	MB1_INT	GPIO (PTB4)
A11	MB2_SPI_CS	SPI0 (PTC3)	G11	I2C0_SDA	I2C0 (PTB3)
A12	MB3_SPI_CS	SPI0 (PTC2)	G12	I2C0_SCL	I2C0 (PTB2)
B1	PRESSURE_INT1	GPIO (PTD12)	H1	USB_DP	USB0_DP
B2	ACCEL_RST	GPIO (PTD11)	H2	USB_DN	USB0_DM
B3	PRESSURE_INT2	GPIO (PTD10)	H3	GND	VSS
B4	UART2_TX	UART2 (PTD3)	H4	USER_SW4	GPIO (PTE28)
B5	USB_VBUS	GPIO (PTC19)	H5	VCC-MCU	VDDA
B6	GYRO_RST	GPIO (PTC15)	H6	GND	VSS
B7	I2C1_SDA	I2C1 (PTC11)	H7	GND	VSS
B8	MB_SPI_MISO	SPI0 (PTC7)	H8	GND	VSS
B9	DISP_EN	GPIO (PTD9)	H9	MB3_PWM	FTM2 (PTB1)
B10	NC	NC	H10	KW41_WAKE_UP	GPIO / LLWU_P5 (PTB0)
B11	ACCEL_INT1	GPIO / LLWU_P6 (PTC1)	H11	CHG_STATE	GPIO (PTA29)
B12	AMB_INT	GPIO (PTC0)	H12	NTAG_FD	GPIO (PTA28)
C1	DISP_DISP	GPIO (PTD15)	J1	GND	ADC0_DP1
C2	RTC_CLKOE	GPIO (PTD14)	J2	GND	ADC_DM1
C3	ACCEL_INT2	GPIO (PTD13)	J3	BAT_SENS	ADC0_SE21
C4	UART2_RX	UART2 (PTD2)	J4	KW41_UART_CTS	GPIO (PTE27)
C5	GYRO_INT2	GPIO (PTC18)	J5	DO_SWDCLK	SWD (PTA0)
C6	BAT_SENS_EN	GPIO (PTC14)	J6	AIR_WAKEN	GPIO (PTA1)
C7	I2C1_SCL	I2C1 (PTC10)	J7	AIR_RESETN	GPIO (PTA6)
C8	MB_SPI_MOSI	SPI0 (PTC6)	J8	AIR_QUALITY_EN	GPIO (PTA7)
C9	AIR_QUALITY_I2C_EN	GPIO (PTD8)	J9	I2S_TX_FS/I2C2_SDA	I2S0 or I2C2 (PTA13)
C10	NC	NC	J10	AIR_INTN	GPIO (PTA27)
C11	KW41_RST_FROM_K64	GPIO (PTB23)	J11	MB2_RST	GPIO (PTA26)
C12	DISP_SPI_SDI	SPI2 (PTB22)	J12	AUTH_RST	GPIO (PTA25)
D1	SD_CLK	SDHC0 (PTE2)	K1	GND	ADC1_DP1
D2	SD_D0	SDHC0 (PTE1)	K2	GND	ADC1_DM1
D3	SD_D1	SDHC0 (PTE0)	K3	MB1_AN	ADC0_SE22
D4	GYRO_INT1	GPIO (PTD1)	K4	KW41_UART_RTS	GPIO (PTE26)
D5	UART3_TX	UART3 (PTC17)	K5	KW41_UART_TX	UART4 (PTE25)
D6	KW41_PB18	(GPIO) PTC13	K6	TOUCH_RST	GPIO (PTA2)
D7	RGB_B	FTM3 (PTC9)	K7	K64_SWDIO	SWD (PTA3)
D8	MB_SPI_SCK	SPI0 (PTC5)	K8	BUZZER_PWM	FTM1 (PTA8)
D9	DISP_SCK	SPI2 (PTB21)	K9	I2S_TXD	I2C0 (PTA12)
D10	DISP_SPI_CS	SPI2 (PTB20)	K10	I2S_RX_FS	I2S0 (PTA16)
D11	CAN_RX	CAN0 (PTB19)	K11	I2S_MCLK	I2S0 (PTA17)
D12	CAN_TX	CAN0 (PTB18)	K12	TOUCH_TXEN	GPIO (PTA24)
E1	DISP_EXTCOMIN	FTM3 (PTE6)	L1	GND	ADC1_DP3
E2	SD_D2	SDHC0 (PTE5)	L2	GND	ADC1_DM3
E3	SD_D3	SDHC0 (PTE4)	L3	MB2_AN	ADC0_SE23
E4	SD_CMD	SDHC0 (PTE3)	L4	GND	ADC1_SE23
E5	VCC-MCU	VDD	L5	GND	RTC_WAKEUP_B
E6	VCC-MCU	VDD	L6	VCC-MCU	VBAT
E7	VCC-MCU	VDD	L7	USER_SW1	GPIO / LLWU_P3 (PTA4)
E8	VCC-MCU	VDD	L8	TOUCH_INT	GPIO (PTA9)
E9	UART_DBG_TX	UART0 (PTB17)	L9	MB2_PWM	FTM2 (PTA11)
E10	UART_DBG_RX	UART0 (PTB16)	L10	I2S_RX_BCLK/I2C2_SCL	I2S0 or I2C2 (PTA14)
E11	MB1_RST	GPIO (PTB11)	L11	I2S_RXD	I2S0 (PTA15)
E12	MB3_RST	GPIO (PTB10)	L12	K64_RST	RESET
F1	USER_SW3	GPIO (PTE10)	M1	GND	ADC0_DP3
F2	USER_SW2	GPIO (PTE9)	M2	GND	ADC0_DM3
F3	NC	(PTE8)	M3	-	VREF_OUT
F4	RGB_G	FTM3 (PTE7)	M4	KW41_UART_RX	UART4 (PTE24)
F5	VCC-MCU	VDD	M5	NC	NC
F6	GND	VSS	M6	K64_EXTAL32	EXTAL32
F7	GND	VSS	M7	K64_XTAL32	XTAL32
F8	VCC-MCU	VDD	M8	I2S_TX_BCLK	I2S0 (PTA5)
F9	SPI1_PCS1	SPI1 (PTB9)	M9	MB1_PWM	FTM2 (PTA10)
F10	MB2_INT	GPIO (PTB8)	M10	GND	VSS
F11	MB3_INT	GPIO (PTB7)	M11	K64_XTAL	XTAL0
F12	MB3_AN	ADC1 (PTB6)	M12	K64_EXTAL	EXTAL0

Table 6. MK64FN1M0VMD12 pin allocation in SLN-RPK-NODE

3.3. Wireless Microcontroller

3.3.1. KW41Z features

The SLN-RPK-NODE features the MKW41Z512VHT4 Wireless MCU. This Low-power microcontroller supports Bluetooth Low Energy v4.2, Generic FSK and IEEE 802.15.4 (Thread) standards.

Feature	Description
Performances	• 48 MHz ARM Cortex-M0+ core • Up to four channel DMA for peripheral and memory servicing with reduced CPU loading and faster system throughput • Independent flash banks allowing concurrent code execution and firmware updating with no performance degradation or complex coding routines
Low-power Capabilities	• 9 Low-power modes with power and clock gating for optimal peripheral activity and recovery times • 1 Low-leakage wake-up unit: KW41Z can be woken-up from Low-Leakage Stop (LLS) and Very Low-Leakage Stop (VLLS) modes by K64F PTB0 I/O • Low-power timer for continual system operation in reduced power states
Flash and RAM Memory	• 512 KB Flash • 128 KB RAM
Optimized RF Design	• Low count of external components • On-board 2.4GHz chip antenna • Option to solder a µFL connector to connect an external antenna • Receiver sensitivity is -100dBm typical @1% PER for 802.15.4 applications at the u.FL connector • Receiver sensitivity is -96dBm typical @30.8% PER for BLE applications at the u.FL connector
Power source	• DC-DC converter configured in bypass mode (minimal BOM)
Reference Oscillators	• 32MHz reference oscillator compliant with BLE and IEEE802.15.4 accuracy requirements • 32.768kHz reference oscillator for Low-power modes
OTA Prog	• 16MBytes external serial flash memory shared with K64F for USB/OTAP support
Indicators	• 1 blue LED for BLE and 1 white LED for 802.15.4 wireless status
Interfaces	• 1 UART dedicated to inter-processor communication
Coexistence Interface	• 3 wires interface for WiFi and BLE or IEEE802.15.4 coexistence • Interface signals accessible from the connector J7

Table 7. Features of MKW41Z512VHT4 used in SLN-RPK-NODE

3.3.2. KW41Z RF circuit

The SLN-RPK-NODE implements an onboard chip antenna (E1) and a matching network (C16, L8 and C17) to interface KW41Z radio port (ANT).

An optional SMA connector (J1) can be soldered to connect an external antenna and the casing can be easily modified to provide external access to the antenna connector. To enable that option, the chip antenna should be disconnected to preserve the 50 ohms impedance matching. The simplest way consists to remove capacitor C66 and add C67 instead (10pF).

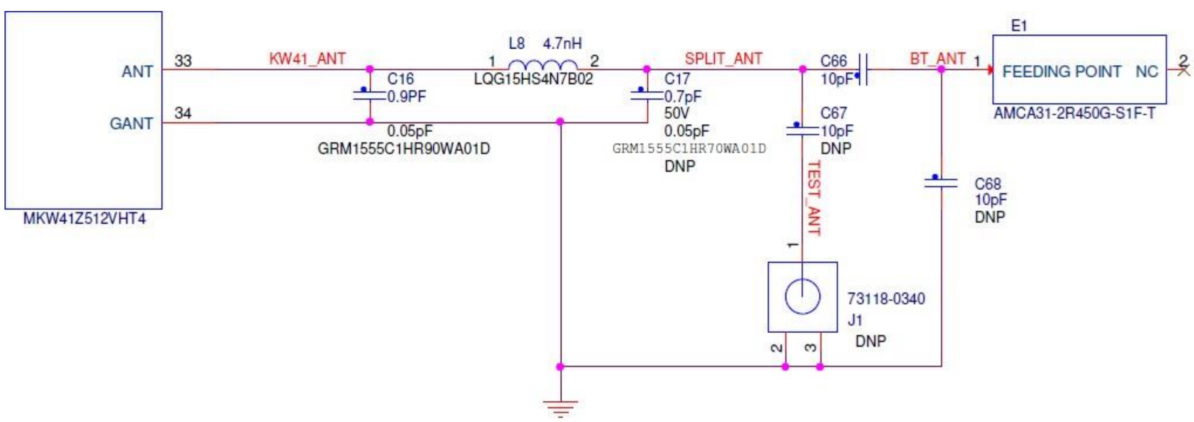


Figure 75. KW41Z RF circuit

3.3.3. KW41Z Coexistence interface

KW41Z supports a coexistence interface with external transceivers that share the same frequency (2.4GHz ISM band). That interface allows a handshake mechanism between the KW41Z and an external radio transceiver (such as WiFi). The purpose of this interface is to coordinate the access to the radio spectrum and optimize the wireless performances of both transceiver.

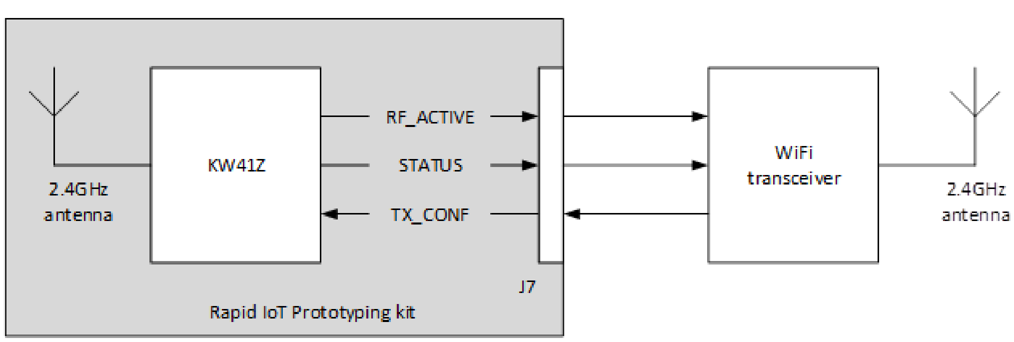


Figure 76. KW41Z CoExistence interface

3.3.4. KW41Z IEEE802.15.4 MAC address

IEEE802.15.4 MAC address is printed on the label sticker accessible in the back of the casing.

Figure 77. KW41Z IEEE802.15.4 MAC Address

3.3.5. KW41Z pin assignment

IC Pin	NET (schematic top)	KW41Z Function (Port)	IC Pin	NET (Schematic Top)	KW41Z Function (Port)
1	KW41_SWDIO	SWD_DIO (PTA0)	33	ANT	KW41_ANT
2	J7_SWDCLK	SWD_CLK (PTA1)	34	GND	GANT
3	RESET_b	RESET (PTA2)	35	VCC_RF	VDD_RF2
4	LED_WHITE	PTA16	36	VCC_RF	VDD_RF1
5	NTAG_FD	PTA17	37	RF_ACTIVE	RF_ACTIVE (PTC1)
6	LED_BLUE	PTA18	38	I2S_RX_BCLK/I2C2_SCL	I2C1_SCL (PTC2)
7	TX_CONF	PTA19	39	I2S_TX_FS/I2C2_SDA	I2C1_SDA (PTC3)
8	GND	PSWITCH	40	KW41_UART_CTS	LPUART0_CTS_b (PTC4)
9	VCC_RF	DCDC_CFG	41	KW41_UART_RTS	LPUART0_RTS_b (PTC5)
10	VCC_RF	VDCDC_IN	42	KW41_UART_RX	UART0_RX (PTC6)
11	-	DCDC_LP	43	KW41_UART_TX	UART0_TX (PTC7)
12	-	DCDC_LN	44	VCC_RF	VDD_1
13	GND	DCDC_GND	45	SPI1_SCK	SPI0_SCK (PTC16)
14	VCC_RF	VDD_1P8OUT	46	SPI1_MOSI	SPI0_SOUT (PTC17)
15	VCC_RF	VDD_1P5OUT_PMCIN	47	SPI1_MISO	SPI0_SIN (PTC18)
16	KW41_WAKE_UP	PTB0 / LLWU_P8	48	SPI1_PCS0	SPI0_PCS0 (PTC19)
17	-	PTB1	49	GND	GND1 (Exposed Pad)
18	-	PTB2	50	GND	GND2 (Exposed Pad)
19	STATUS	PTB3	51	GND	GND3 (Exposed Pad)
20	VCC_RF	VDD_0	52	GND	GND4 (Exposed Pad)
21	KW41_EXTAL32	EXTAL32K	53	GND	GND5 (Exposed Pad)
22	KW_XTAL32	XTAL32K	54	GND	GND6 (Exposed Pad)
23	KW41_PB18	PTB18	55	GND	GND7 (Exposed Pad)
24	-	ADC0_DP0	56	GND	GND8 (Exposed Pad)
25	-	ADC0_DM0	57	GND	GND9 (Exposed Pad)
26	GND	VSSA	58	GND	GND10 (Exposed Pad)
27	VCC_RF	VREFH/VREF_OUT	59	GND	GND11 (Exposed Pad)
28	VCC_RF	VDDA	60	GND	GND12 (Exposed Pad)
29	-	XTAL_OUT	61	GND	GND13 (Exposed Pad)
30	KW41_XTAL	XTAL	62	GND	GND14 (Exposed Pad)
31	KW41_EXTAL	EXTAL	63	GND	GND15 (Exposed Pad)
32	VCC_RF	VDD_RF3	64	GND	GND16 (Exposed Pad)

Table 8. MKW41Z512VHT4 pin allocation in SLN-RPK-NODE

3.4. Board power supply

The architecture of the SLN-RPK-NODE is represented in the following diagram.

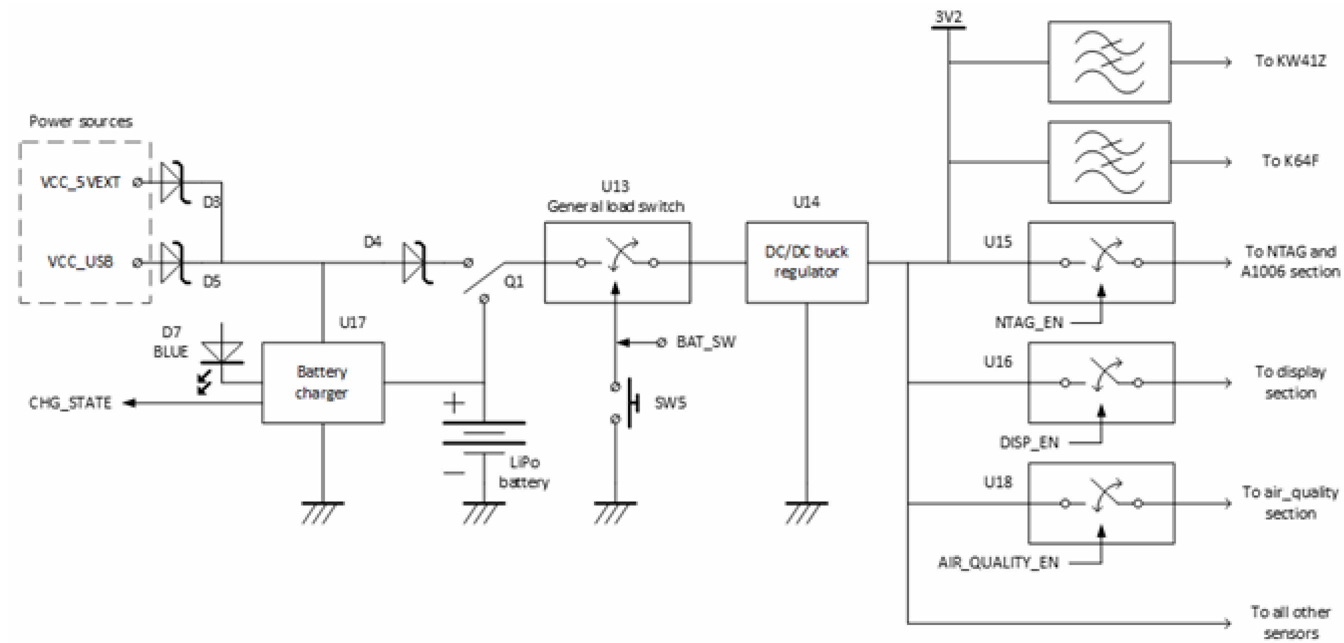


Figure 78. Power supply overview

The Rapid IoT prototyping kit can be powered from 2 distinct sources:

- the Docking station connector (VCC_5VEXT)
- the USB connector (VCC_USB)

Both sources should be 5V typical.

The 5V input is used to power the complete solution and to charge the embedded Lithium Polymer battery. This LiPo battery has a capacity of 240mAh. It is charged using NXP battery charger, MC34671AEP, with a constant current of 150mA.

It is possible to shut off the power to the MCUs, the sensors, the display, etc… pressing SW5/RESET push button or applying 0V to BAT_SW pin on the docking connector.

A DC/DC buck regulator generates a 3.2V to power the two microcontrollers, the sensors, the display… This voltage respects the power specifications for every component embedded in SLN-RPK-NODE.

Dedicated load switches have been added for the NTAG, the display and the air quality sensor. Thanks to those switches, it is possible to isolate those power-hungry parts an therefore preserve the battery.

The charging status is delivered to the K64F, it indicates whether the battery is currently charging (CHG_STATE = 1) or not (CHG_STATE = 0).

Additionally, for debug purposes, an internal BLUE LED (D7), not visible with the casing, indicates if a 5V power-source is present.

Battery voltage is measured thanks to a K64F ADC input. Measurement is enabled / disabled via BAT_SENS_EN control GPIO.

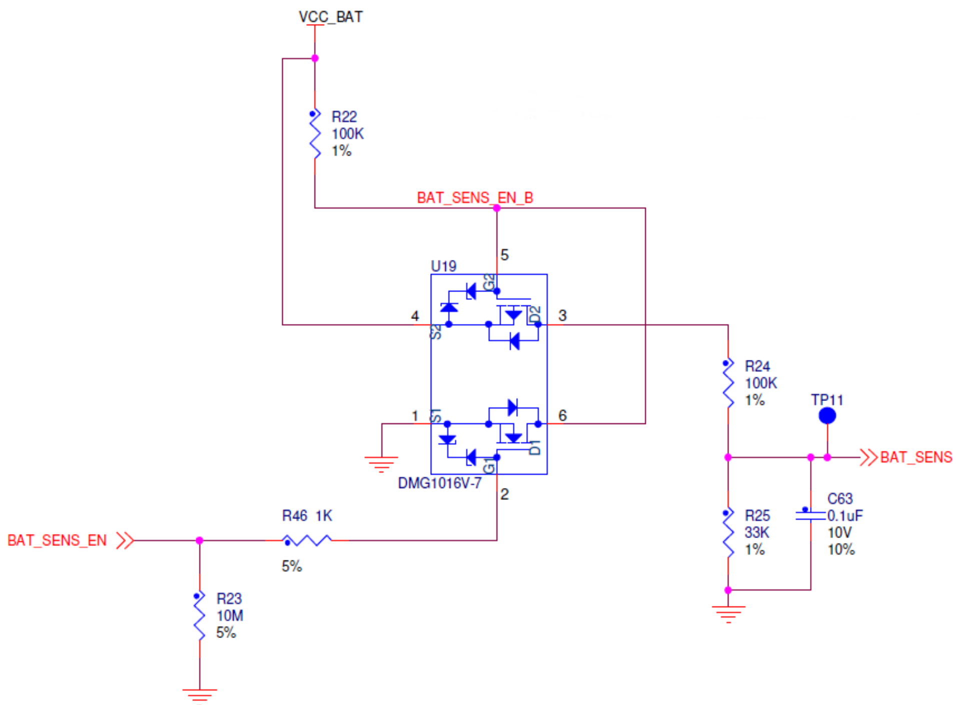


Figure 79. Battery sensing implementation

3.5. Clock sources

3.5.1. K64F clocking

By default, K64F MCU boots from an internal Digitally Controlled Oscillator (DCO). By software, user can select a 12 MHz crystal connected to the main external oscillator pins (EXTAL0/XTAL0).

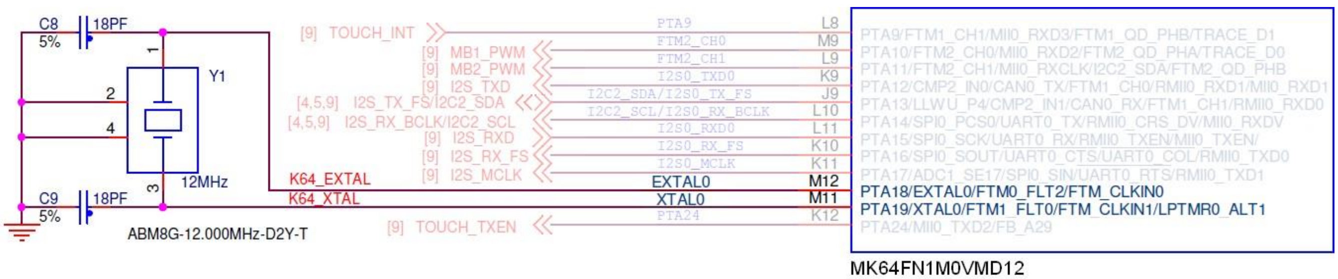


Figure 80. K64F Clock source

K64F LP/32.768 kHz oscillator input is connected to the clock buffer output of the Real Time Clock PCF2123. By default, this function is disabled for the K64F and the PCF2123 output buffer. Please note that the duty cycle of the 32.768 kHz clock provided by the PCF2123 is not a pure 50% and can vary by several percent.

3.5.2. KW41Z clocking

KW41Z Wireless MCU requires two clocks to operate. A 32 MHz oscillator to clock the controller and the radio and a 32.768 kHz oscillator to provide an accurate low-power time base.

3.5.2.1. 32MHz Reference Oscillator

IEEE Standard for 802.15.4 radio requires frequency to be accurate to less than +/- 40 ppm.
The 32 MHz crystal is connected to KW41Z pins (XTAL/EXTAL).
Internal load capacitors provide the crystal load capacitance, which can be adjusted from 0pF to 30pF.

To verify 32MHz oscillator frequency, user can program CLKOUT (PTB0) to provide buffered output clock signal.

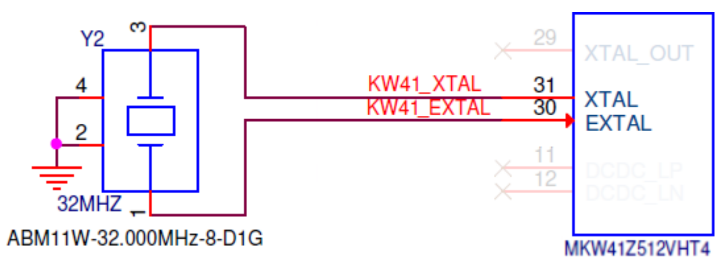


Figure 81. KW41Z Clock source

3.5.2.2. 32.768kHz Crystal oscillator (for accurate low-power time base)

Standard for BLE requires an accurate time-base generated by a 32.768 kHz crystal.
The 32.768kHz crystal is connected to KW41Z pins (XTAL32K/EXTAL32K).
Internal load capacitors provide the crystal load capacitance, which can be adjusted from 0pF to 30pF. Frequency can also be checked from a GPIO.

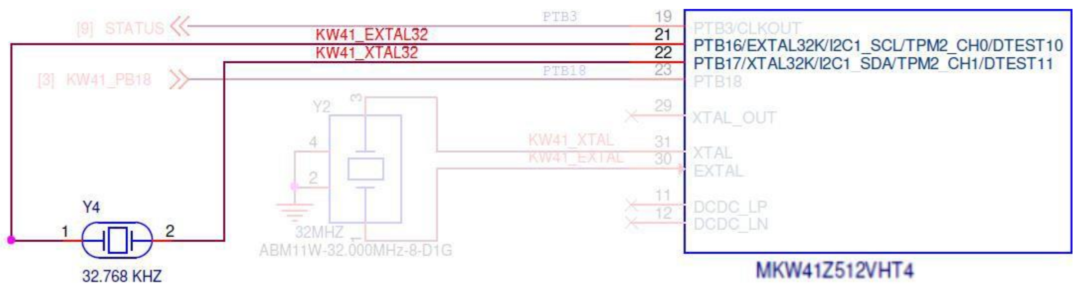


Figure 82. KW41Z Clock source

3.6. Sensors

3.6.1. Accelerometer and Magnetometer

A digital low-power and six-axis accelerometer/magnetometer sensor FXOS8700CQ is connected to the K64F MCU through the serial I2C1 interface and three GPIOs, as described below.

The I2C address for FXOS8700CQ is 0x1E (SA0 and SA1 pulldown).

By default, the reset pin of FXOS8700CQ is pulled-down low via a 100k resistor. By software, K64F MCU can force it to the high level via PTD11 GPIO.

FXOS8700CQ interrupt pin INT1 is connected to K64F low-leakage wake-up unit (LLWU_P6). By software, it allows SLN-RPK-NODE to wake-up from Low-Leakage (LLS) and very low-leakage (VLLS) Stop modes.

FXOS8700CQ Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
INT1	PTC1 (LLWU_P6)	ACCEL_INT1
INT2	PTD13	ACCEL_INT2
RST	PTD11	ACCEL_RST

Table 9. FXOS8700CQ pin allocation in SLN-RPK-NODE

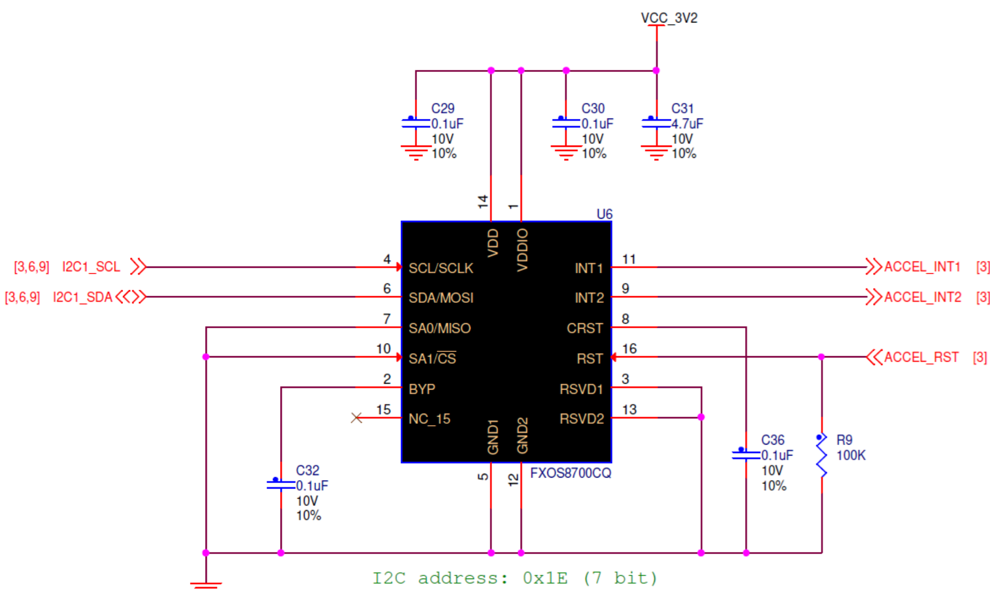


Figure 83. FXOS8700CQ Accelerometer and Magnetometer

3.6.2. Gyroscope

A digital low-power and three-axis gyroscope sensor FXAS21002CQ is connected to the K64F MCU through the serial I2C1 interface and three GPIOs, as described below.

The I2C address for FXAS21002CQ is 0x20 (SA0 pulldown).

FXAS21002CQ can be configured to generate an interrupt, when a defined angular rate threshold is crossed on any one of the selected axes.

FXAS21002 Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
INT1	PTD1	GYRO_INT1
INT2	PTC18	GYRO_INT2
RST	PTC15	GYRO_RST

Table 10. FXAS21002CQ pin allocation in SLN-RPK-NODE

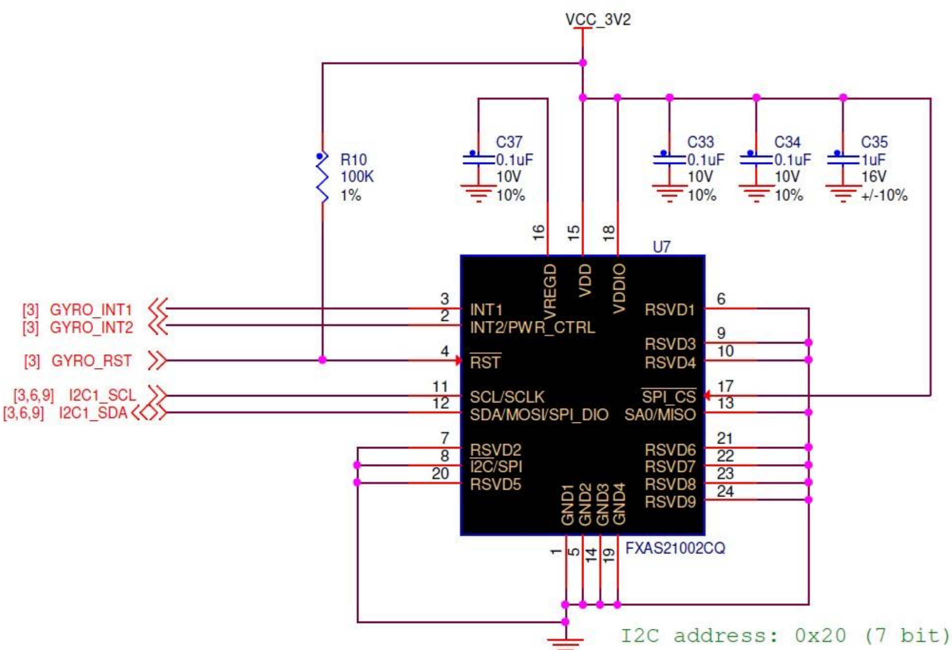


Figure 84. FXAS21002CQ Gyroscope

3.6.3. Pressure and Temperature

A digital barometric/altitude pressure and temperature sensor MPL3115A2 is connected to the K64F MCU through the serial I2C1 interface and two GPIOs, as described below.

The I2C address for MPL3115A2 is 0x60 (SA0 pulldown).

MPL3115A2 features multiple programmable modes such as power saving, interrupt and autonomous data acquisition modes, including programmed acquisition cycle timing and poll-only modes.

MPL3115A2 Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
INT1	PTD12	PRESSURE_INT1
INT2	PTD10	PRESSURE_INT2

Table 11. MPL3115A2 pin allocation in SLN-RPK-NODE

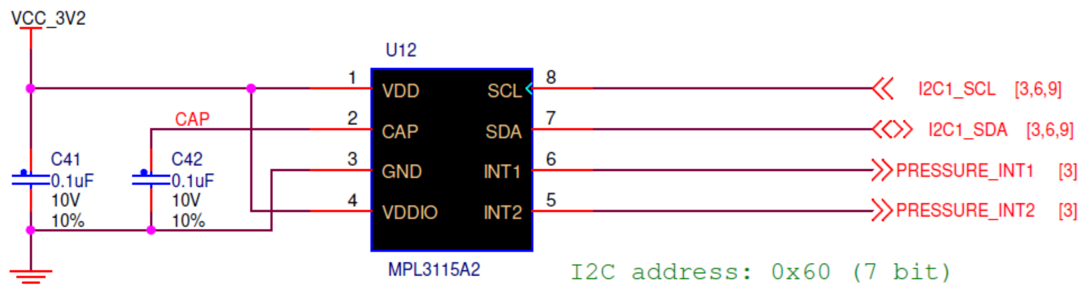


Figure 85. MPL3115A2 Barometric Pressure and Temperature Sensor

NOTE: The Pressure and Temperature sensor (MPL3115) is influenced by its nearby environment. Especially the application code should consider the Rapid IoT self-heating, when measuring the temperature. If both K64F and KW41Z CPUs are running in normal mode at their maximum clock frequency, the temperature measurement is typically offset by 7 to 8°C. On the other hand, an application that is mostly in low power mode and wakes up occasionally to measure the temperature will not be impacted by the device self-heating (offset = 0°C).

3.6.4. Humidity and Temperature

A digital relative humidity and high-accuracy temperature sensor ENS210 is connected to the K64F MCU through the serial I2C1 interface, as described below.

The I2C address for ENS210 is 0x43.

ENS210 features automatic low-power standby (40nA), when not sensing.

ENS210 Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA

Table 12. ENS210 pin allocation in SLN-RPK-NODE

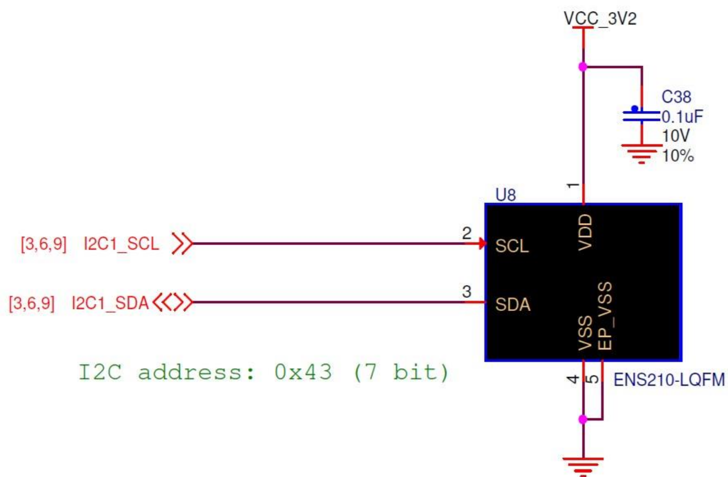


Figure 86. ENS210 Humidity and Temperature Sensor

NOTE: The Humidity and Temperature sensor (ENS210) is influenced by its nearby environment. Especially the application code should consider the Rapid IoT self-heating, when measuring the temperature. If both K64F and KW41Z CPUs are running in normal mode at their maximum clock frequency, the temperature measurement is typically offset by 7 to 8°C. On the other hand, an application that is mostly in low power mode and wakes up once in a while to measure the temperature will not be impacted by the device self-heating (offset = 0°C).

3.6.5. Air Quality

A digital gas sensor for indoor air quality monitoring CCS811 is connected to the K64F MCU through the serial I2C1 interface and six GPIOs, as described below.

The I2C address for CCS811 is 0x5A.

Thanks to power and analog switches user can disable CCS811 power supply and serial I2C interface to reduce system power consumption in low power modes or when application is not sensing air quality.

CCS811 Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
INTN	PTA27	AIR_INTN
WAKEN	PTA1	AIR_RESETN
RESETN	PTA6	AIT_WAKEN

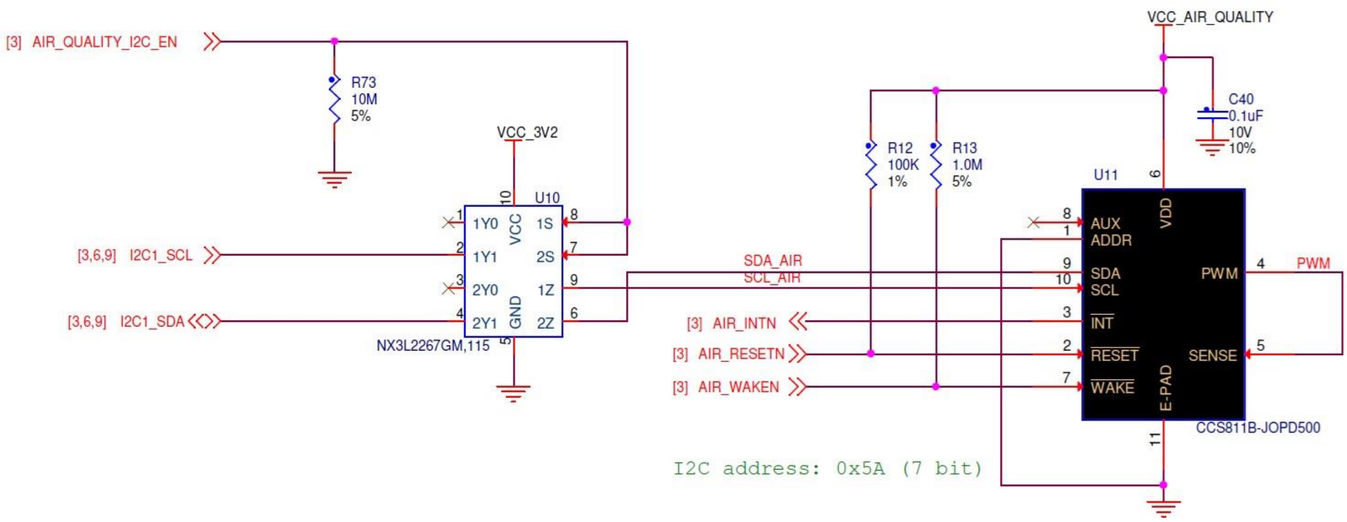
Table 13. CCS811 pin allocation in SLN-RPK-NODE

CCS811 Power supply	K64F Signal	Schematic Net Name
Power	PTA7	AIR_QUALITY_EN

Table 14. CCS811 supply control in SLN-RPK-NODE

SPDT analog switch (U10)	K64F	Schematic net
I2C	PTD8	AIR_QUALITY_I2C_EN

Table 15. CCS811 I2C switch in SLN-RPK-NODE



	SDA_AIR	SCL_AIR
AIR_QUALITY_I2C_EN = 1	I2C1_SDA	I2C1_SCL
AIR_QUALITY_I2C_EN = 0	High Z	High Z

	VCC_AIR_QUALITY
AIR_QUALITY_EN = 1	3.2V
AIR_QUALITY_EN = 0	0V

Figure 87. CCS811 Air Quality Sensor

3.6.6. Ambient Light

A digital ambient light sensor TSL25721FN with programmable interrupt is connected to the K64F MCU through the serial I2C1 interface and one GPIO, as described below.

The I2C address for TSL25721FN is 0x39.

TSL25721FN provides ambient light sensing that approximates human eye response to light intensity. In addition, the operating range is extended to 60,000 lux in sunlight when the low-gain mode is used.

TSL25721FN	K64F	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
INTN	PTC0	AMB_INT

Table 16. TSL25721FN pin allocation in SLN-RPK-NODE

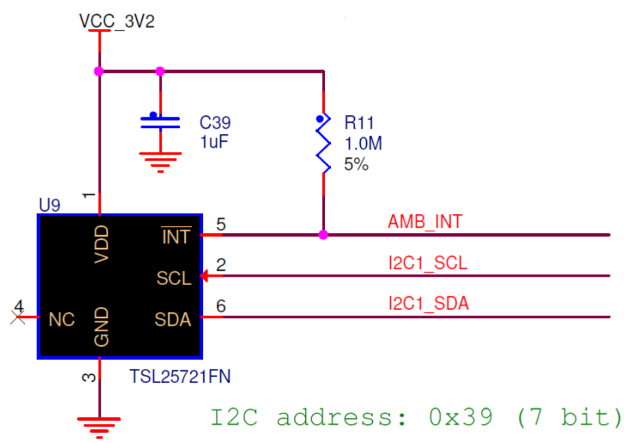


Figure 88. TSL25721FN Ambient Light Sensor

3.7. Color LCD Display

A 1.28inch color display LPM013M126C with Memory-in-Pixel technology and 176x176 pixels resolution is connected to the K64F MCU through the serial SPI2 interface and three GPIOs.

Thanks to its reflectance background this screen is very efficient under outdoor/strong ambient light with an ultra-low power consumption, typically 2µW, with a static image and backlight disabled.

For indoor/dark environment, a backlight LED can be activated and dimmed via a PWM output (DISP_BLIGHT) from the K64F MCU.

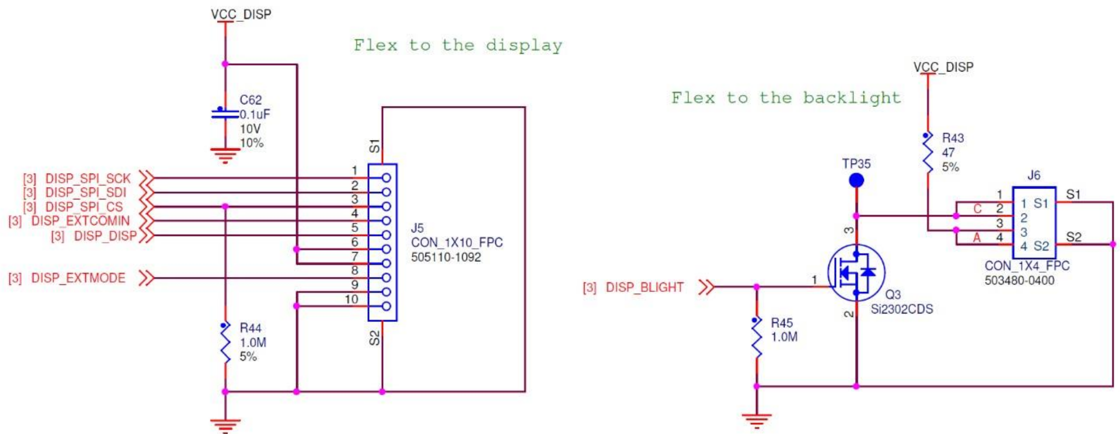


Figure 89. Connectors to the LCD display

NOTE: To avoid DC bias occurring, while displaying still image for a long time, Memory-In-Pixel technology requires to invert the voltage VCOM continuously. This signal is driven by a FlexTimer from the K64F MCU (DISP_EXTCOMIN).

LPMC013M126C Display pin	K64F Signal	Schematic Net Name
SLCK	SPI2_SCK (PTB21)	DISP_SPI_SCK
SI	SPI2_SOUT (PTB22)	DISP_SPI_SDI
SCS	SPI2_PCS0 (PTB20)	DISP_SPI_CS
EXTCOMIN	FTM3_CH1 (PTE6)	DISP_EXTCOMIN
DISP	PTD15	DISP_DISP
VDDA	-	VCC_DISP
VDD	-	VCC_DISP
EXTMODE	PTE11	DISP_EXTMODE
VSS	-	GND
VSSA	-	GND

Table 17. LPM013M126C pin allocation in SLN-RPK-NODE

Power	K64F	Schematic Net Name
VCC_DISP	PTD9	DISP_EN

Table 18. LPM013M126C supply control in SLN-RPK-NODE

LPMC013M126C Backlight pin	K64F	Schematic Net Name
CATHODE	FTM3_CH7 (PTE12) via transistor Q3	DISP_BLIGHT

Table 19. LPM013M126C backlight control in SLN-RPK-NODE

3.8. Security for authentication

A security chip A1006 is connected to the K64F and the KW41Z MCUs through the serial I2C interface to provide anti-counterfeit protection.

A1006 security solution is based on industry standard asymmetric cryptographic challenge-response protocols, using NIST approved elliptic curves, Elliptic Curve Diffie-Hellman challenge response (ECDH), and customizable X.509 certificates signed using the Elliptic Curve Digital Signature Algorithm (ECDSA). Advanced anti-tampering countermeasures are incorporated into the A1006 to prevent various attacks and minimize the scalability of any attempts to clone the A1006.

I2C lines are shared with the NTAG I2C plus (NT3H2211). I2C address for A1006 is 0x50 (7-bit).

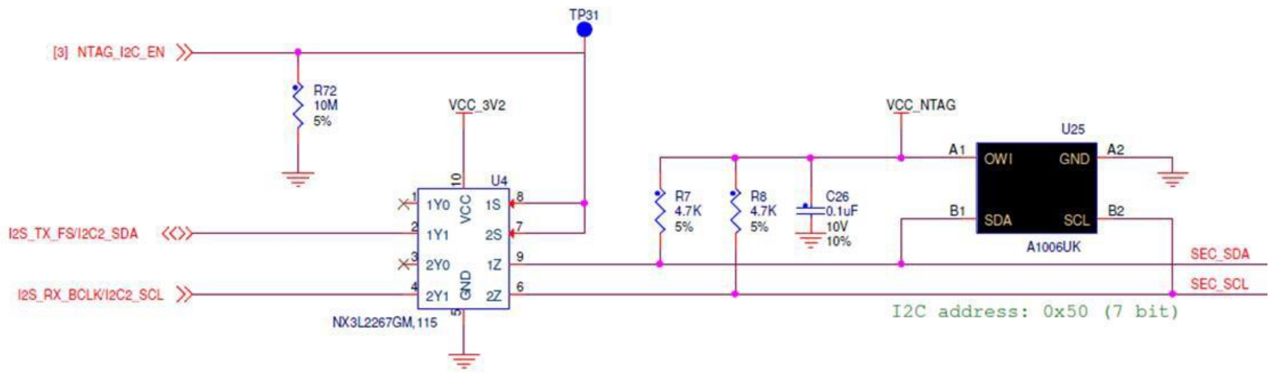


Figure 90. A1006 Security Chip

Each Rapid IoT comes with a pre-programmed user certificate and a unique identifier.
The authentication unique ID is printed on the back of the device (8 bytes).

Figure 91. A1006 authentication unique ID

The following configurations can be managed by the K64F

A1006 Power supply	K64F	Schematic net
VCC_NTAG	PTC12	NTAG_EN

Table 20. A1006 supply control in SLN-RPK-NODE

SPDT analog switch (U4)	K64F	Schematic net
1S & 2S	PTB5	NTAG_I2C_EN

Table 21. A1006 I2C switch in SLN-RPK-NODE

A1006 interface	K64F	Schematic net
SDA	I2C2_SDA (PTA13) via SPDT switch	SEC_SDA
SCL	I2C2_SCL (PTA14) via SPDT switch	SEC_SCL

Table 22. A1006 pin allocation in SLN-RPK-NODE

3.9. NFC TAG for identification

A NFC Tag NT3H2211 together with a NFC ferrite antenna is connected to the K64F and the KW41Z MCUs through the serial I2C interface. This combination enables NFC "tap-and-go" connectivity typically for commissioning a new device in a home-automation system.

I2C lines are shared with the security chip (A1006). I2C address for NT3H2211 is 0x55 (7-bit).

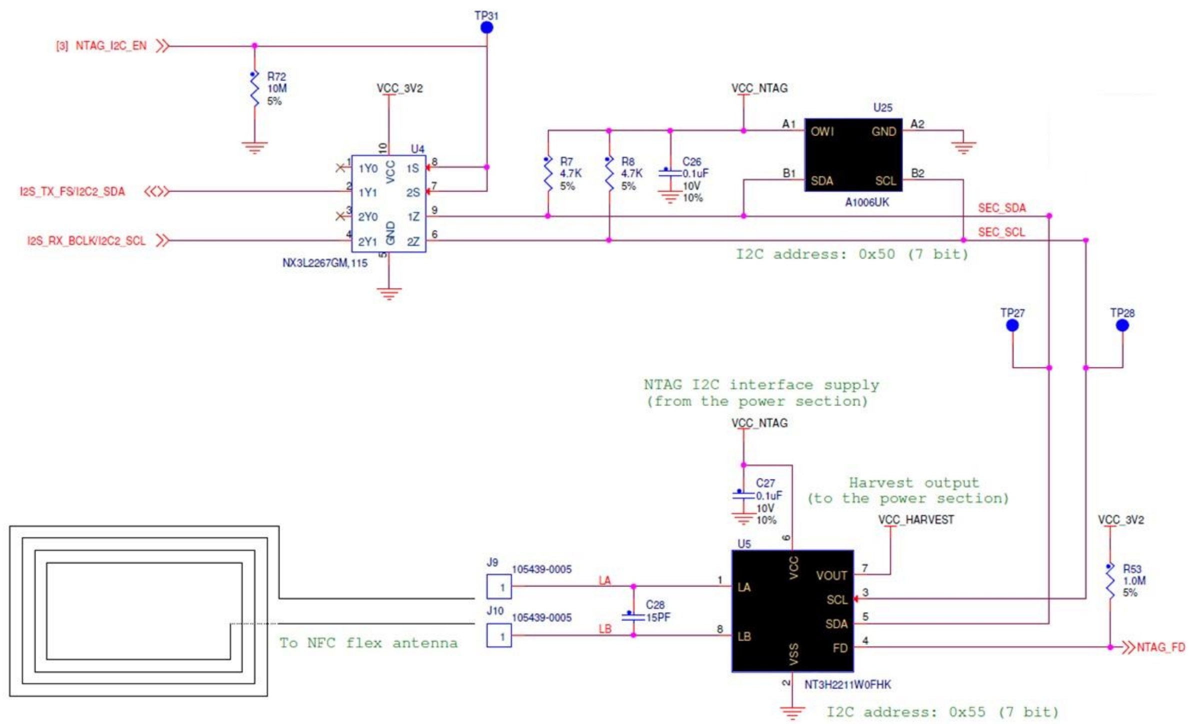


Figure 92. NT3H2111 NFC Tag

The NFC ferrite antenna is stuck inside the Rapid IoT casing, the antenna location is highlighted with NFC logo (see Rapid IoT rear view)

Figure 93. Rear view, NFC antenna location

3.10. Serial Flash memory

A 128Mbit serial NOR flash memory MT25QL128ABA1EW7 is connected to the K64F and the KW41Z MCUs through the serial SPI interface. This external memory provides storage for Over The Air Programming (OTAP), data logging and factory images storage (K64F application and KW41Z wireless firmware).

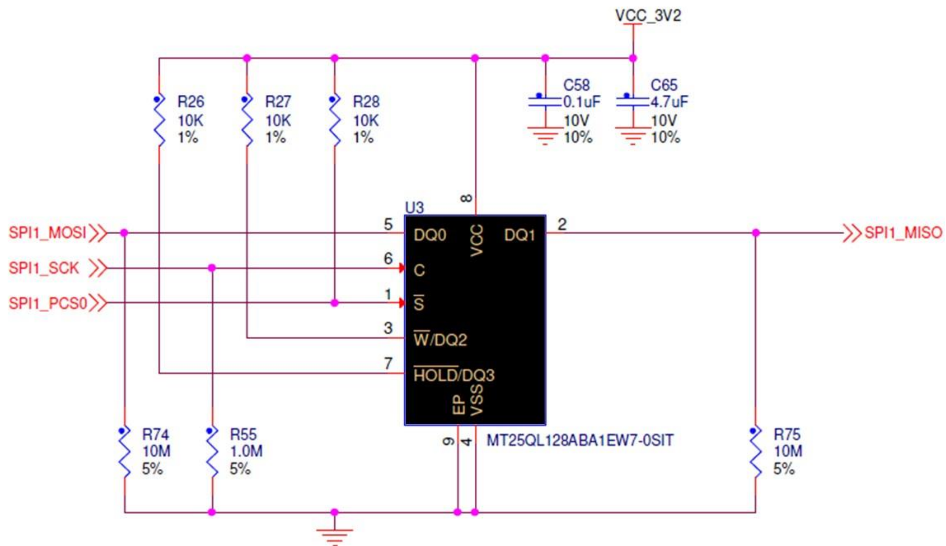


Figure 94. MT25QL128ABA1EW7 Serial NOR Flash

MT25QL128ABA pin	K64F Signal	KW41Z Signal	Schematic Net Name
DQ0	SPI1_SOUT	SPI0_SOUT	SPI1_MOSI
C	SPI1_SCK	SPI0_SCK	SPI1_SCK
S	SPI1_PCS0	SPI0_PCS0	SPI1_PCS0
W/DQ2	-	-	Pull-up to VCC
HOLD/DQ3	-	-	Pull-up to VCC
DQ1	SPI1_SIN	SPI0_SIN	SPI1_MISO

Table 23. MT25QL128ABA1EW7 pin allocation in SLN-RPK-NODE

NOTE: Both MCUs access the serial flash via a common SPI bus. To prevent collisions in case K64F and KW41Z want to access simultaneously the memory a contention mechanism is also implemented. The lines KW41_UART_RTS ans KW41_UART_CTS are typically used as GPIOs in Rapid IoT demo application.

3.11. LED's

3.11.1. RGB LED

A RGB LED SML-LX0404SIUPGUSB is connected to K64F MCU through three GPIOs configured as Pulse Width Modulation outputs to support both color and brightness control.

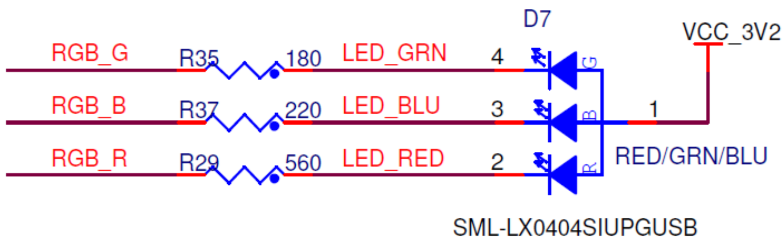


Figure 95. SML-LX0404SIUPGUSB RGB LED

LED Color	K64F Signal	Schematic Net Name
RED	FTM3_CH4 (PTC8)	RGB_R
GREEN	FTM3_CH2 (PTE7)	RGB_G
BLUE	FTM3_CH5 (PTC9)	RGB_B

Table 24. SML-LX0404SIUPGUSB pin allocation in SLN-RPK-NODE

3.11.2. Connectivity application LEDs

Two LEDs, one blue and one white, are connected to the KW41Z MCU through two GPIOs to provide some status for the wireless connections.

Blue LED is dedicated to Bluetooth, while white LED is reserved for 802.15.4 radio.

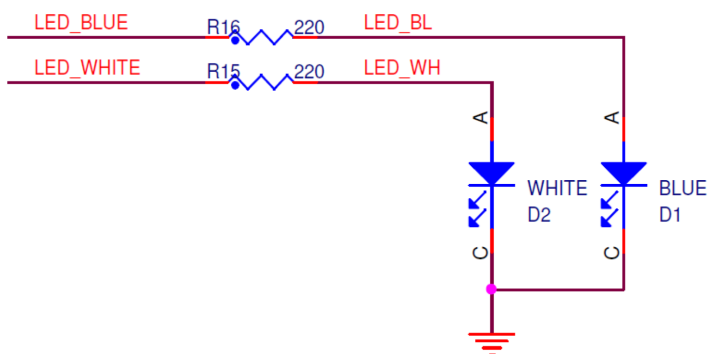


Figure 96. Connectivity Status LEDs

BLE/Thread Connection status	Blue/White LED status
BLE/Thread Off	OFF
BLE/Thread On, not connected	Blinking twice every 10 second
BLE/Thread On, joining	Blinking
BLE/Thread On, connected	Blinking once every 10 second

Table 25. Blue/White LED Status

3.12. Buzzer

A buzzer CMT-4023S-SMT is connected to the K64F MCU through a GPIO (PTA8/FTM0_CH1) and a P-Channel MOSFET.

GPIO is configured as a Pulse Width Modulation output to support control of sound frequency and level. Default output frequency is 4kHz.

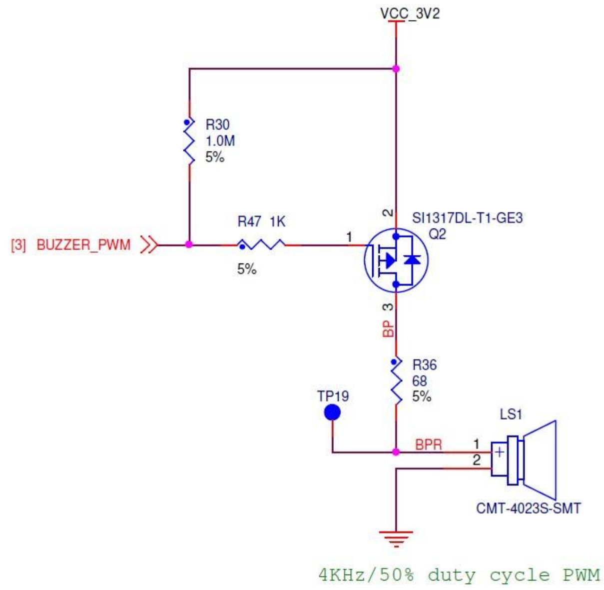


Figure 97. CMT-4023S-SMT Buzzer

3.13. Reset Button

K64F and KW41Z can be reset simultaneously by pressing the button SW5/RESET available at the back of the SLN-RPK-NODE. A needle should be inserted in the dedicated hole of the casing.

Figure 98. Rear view, Reset button location

3.14. User Buttons

Four push buttons are connected to the K64F MCU through four GPIOs.

Buttons are labeled SW1, SW2, SW3 and SW4 and placed two each side to the SLN-RPK-NODE to interact with the user.

SW1 button is connected to K64F Low-Leakage Wake-up Unit (LLWU_P3). Properly configured, pushing SW1 could wake-up SLN-RPK-NODE from Low-Leakage Stop (LLS) and Very Low-Leakage Stop (VLLS) modes.

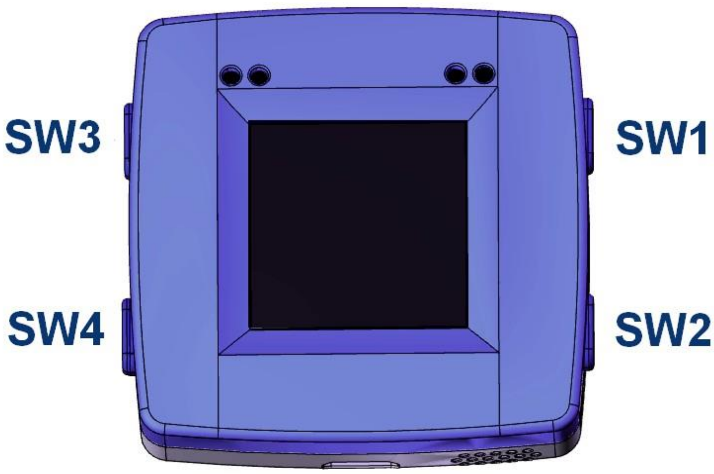


Figure 99. User push button location

User Button	K64F Signal	Schematic Net Name
SW1	PTA4 (LLWU_P3)	USER_SW1
SW2	PTE9	USER_SW2
SW3	PTE10	USER_SW3
SW4	PTE28	USER_SW4

Table 26. User Buttons pin allocation in SLN-RPK-NODE

3.15. Capacitive Touch

A low-power, four channels, capacitive proximity controller SX9500 is connected to the K64F MCU through the serial I2C1 interface and three GPIOs.

The I2C address for SX9500 is 0x28

SX9500 operates either as a proximity or button sensor. It includes sophisticated on-chip auto-calibration circuitry to regularly perform sensitivity adjustments, maintaining peak performance over a wide variation of temperature, humidity and noise environments

Upon a proximity detection, the NIRQ output asserts, enabling the user to either determine the relative Proximity distance, or simply obtain an indication of detection

A dedicated transmit enable (TXEN) pin is available to synchronize capacitive measurements for applications that require synchronous detection.

The four electrodes are located around the display, one on each side as described below.

Figure 100. Touch electrodes location

SX9500 Signal	K64F Signal	Schematic Net Name
SCL	I2C1_SCL (PTC10)	I2C1_SCL
SDA	I2C1_SDA (PTC11)	I2C1_SDA
TXEN	PTA24	TOUCH_TXEN
NIRQ	PTA9	TOUCH_INT
NRST	PTA2	TOUCH_RST

Table 27. SX9500 pin allocation in SLN-RPK-NODE

3.16. Communication interfaces

K64F MCU provides a USB2.0 interface compatible with USB 2.0 full-speed mode that supports USB CDC, HID, MSD and Device profiles.

The micro USB type-B connector is also connected to the battery charger (see chapter 3.4 Board power supply).

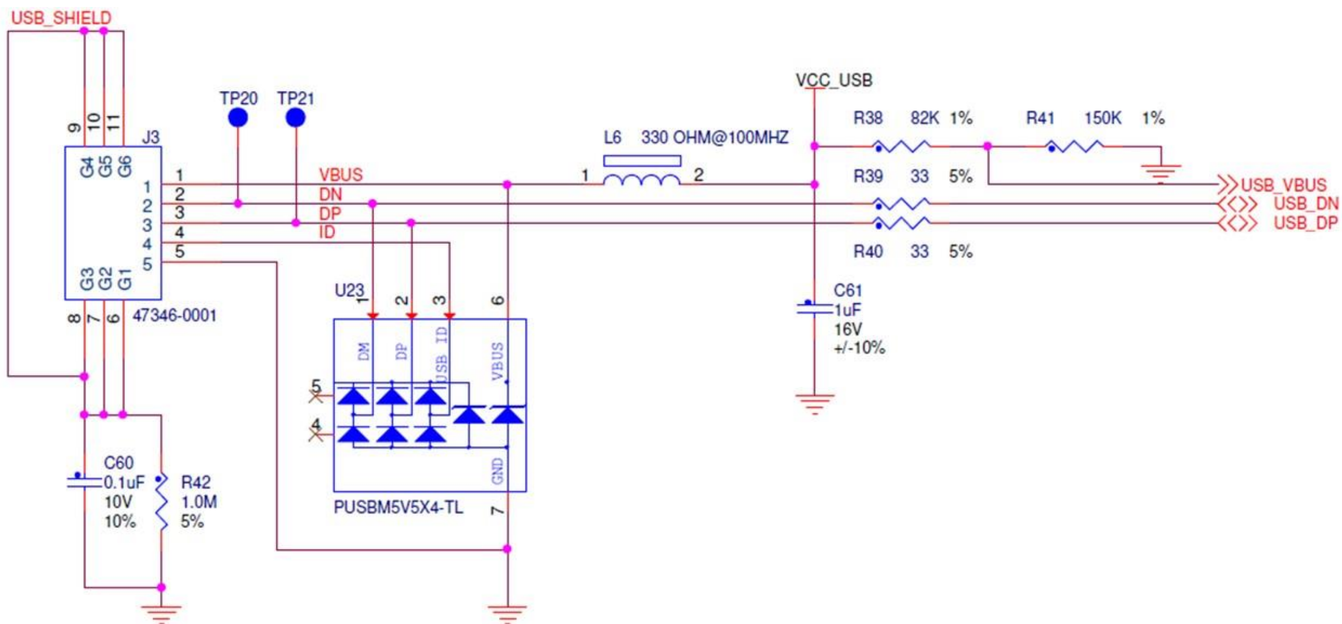


Figure 101. K64F USB interface

NOTE: Thanks to the preloaded bootloader, both K64F and KW41Z firmware can be updated through the USB interface.

3.17. Extension connectors

SLN-RPK-NODE specifications can be extended thanks to two board-to-board connectors.

50-pins extension connector is compatible with Hexiwear Docking-station to flash, serial-monitor or debug both SLN-RPK-NODE MCUs with OpenSDA circuitry or connect up to three MikroE Click™ Add-on boards.

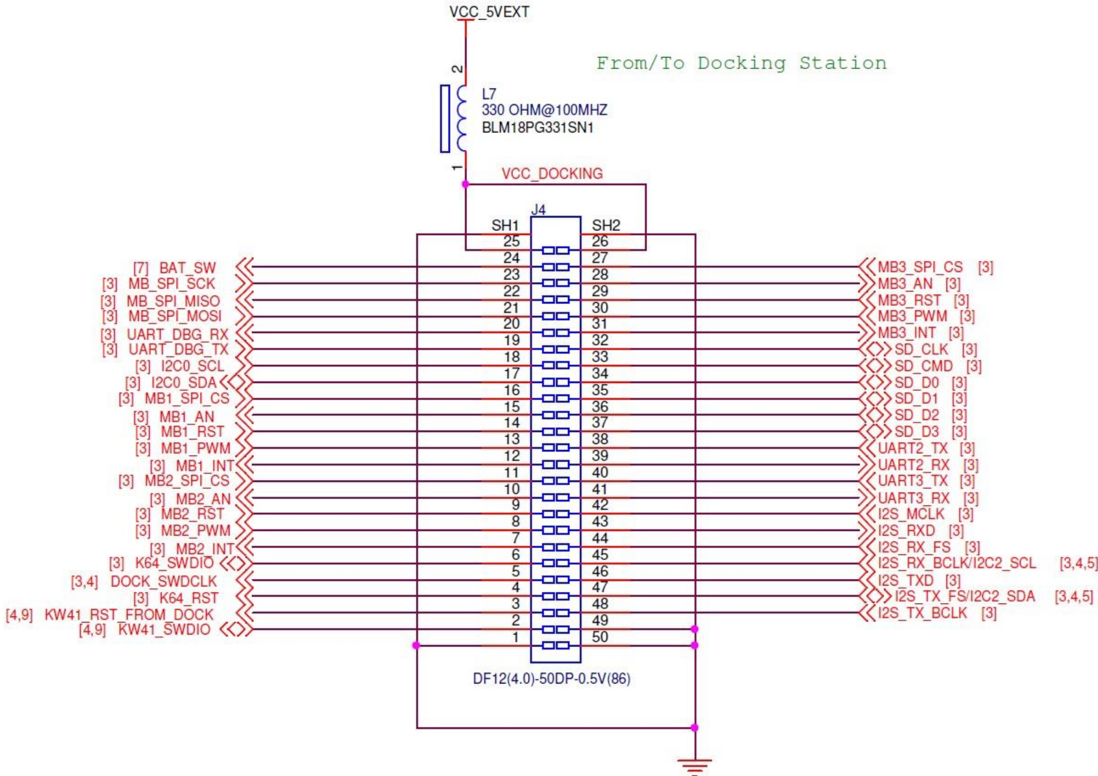


Figure 102. 50-pins extension connector

20-pins extension connector enables SLN-IOT-RPK upgrade with WiFi/Cellular and CAN connectivity. This connector will be compatible with a future Docking-station-plus (available end-of-2018)

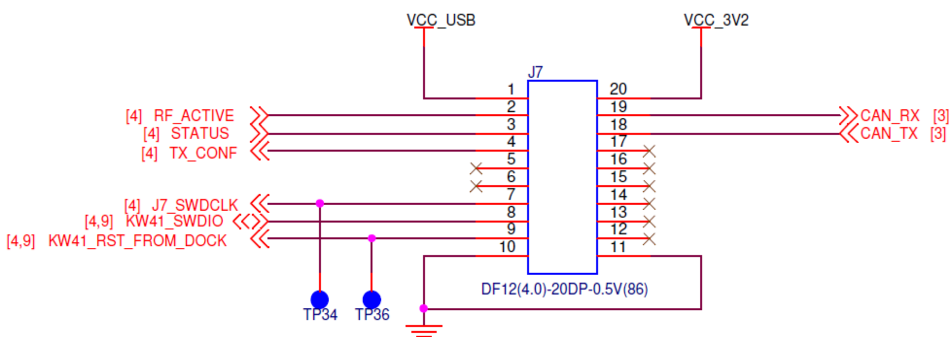


Figure 103. 20-pins extension connector

20-pins extension connector also allow simultaneous debug of K64F and KW41Z MCUs through SWD interface. R48 must be unsoldered from main PCB to split SWD clock signal from K64F and KW41Z.

- K64F debug signals will be accessible through 50-pins extension connector to use with the embedded probe of the Hexiwear Docking-station, or any other external debug probe.
- KW41Z debug signals will be accessible through 20-pins extension connector to use with any external debug probe.

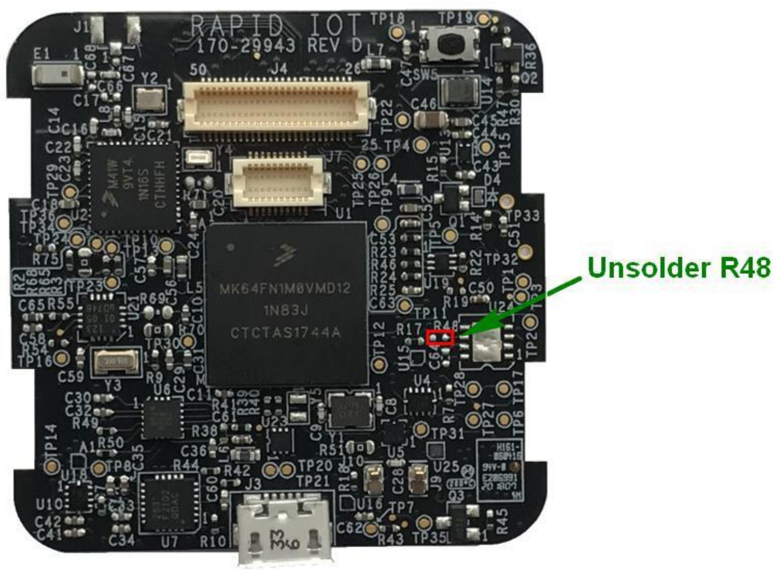


Figure 104. SLN-RPK-NODE modification to support simultaneous K64F/KW41Z debug

3.18. Mikroelektronika Docking-station

SLN-RPK-NODE is fully compatible with Mikroelektronika Hexiwear Docking-station (MIKROE- 2094).

Mikroelektronika Hexiwear Docking-station features:

- Micro USB type-B connector to give access to OpenSDA circuitry and provide power to both Docking-station and SLN-RPK-NODE boards
- OpenSDA probe to program, serial-monitor and debug via SWD both SLN-RPK-NODE MCUs
- Three mikroBUS™ sockets
- ON/OFF power-switch
- Two reset push buttons, one for K6x and another one for KW4x
- MicroSD slot
- JTAG connector for external programmers
- I2S interface (10 pins header)
- 50-pin connector to plug SLN-RPK-NODE hardware

MikroElektronika Click™ boards can be used to expand Rapid IoT functionality nearly infinitely.

Figure 105. MikroElektronika Docking-station

The following table provides the equivalence between Mikroelektronika Docking-station labels and K64F signals from the SLN-RPK-NODE.

	MikroE docking station label	Rapid IoT K64F function
Click™ header 1	AN / PTB2	ADC0_SE22
	RST / PTB11	PTB11
	CS / PTC4	SPI0_PCS0 / PTC4
	SCK / PTC5	SPI0_SCK / PTC5
	MISO / PTC7	SPI0_SIN / PTC7
	MOSI / PTC6	SPI0_SOUT / PTC6
	PWM / PTA10	FTM2_CH0 / PTA10
	INT / PTB13	PTB4
	RX / PTD2	UART2_RX / PTD2
	TX / PTD3	UART2_TX / PTD3
	SCL / PTD8	I2C0_SCL / PTB2
	SDA / PTD9	I2C0_SDA / PTB3
Click™ header 2	AN / PTB3	ADC0_SE23
	RST / PTB19	PTA26
	CS / PTC3	SPI0_PCS1 / PTC3
	SCK / PTC5	SPI0_SCK / PTC5
	MISO / PTC7	SPI0_SIN / PTC7
	MOSI / PTC6	SPI0_SOUT / PTC6
	PWM / PTA11	FTM2_CH1 / PTA11
	INT / PTB8	PTB8
	RX / PTC16	UART3_RX / PTC16
	TX / PTC17	UART3_TX / PTC17
	SCL / PTD8	I2C0_SCL / PTB2
	SDA / PTD9	I2C0_SDA / PTB3
Click™ header 3	AN / PTB6	ADC1_SE12 / PTB6
	RST / PTB10	PTB10
	CS / PTC2	SPI0_PCS2 / PTC2
	SCK / PTC5	SPI0_SCK / PTC5
	MISO / PTC7	SPI0_SIN / PTC7
	MOSI / PTC6	SPI0_SOUT / PTC6
	PWM / PTA4	FTM1_CH1 / PTB1
	INT / PTB7	PTB7
	RX / PTC16	UART3_RX / PTC16
	TX / PTC17	UART3_TX / PTC17
	SCL / PTD8	I2C0_SCL / PTB2
	SDA / PTD9	I2C0_SDA / PTB3

Table 28. Signals access from the Docking-station for K64F embedded in SLN-RPK-NODE

IMPORTANT: The Docking station shipped by MikroElektronika is preprogrammed in factory with an OpenSDA DAP-LINK application dedicated for the MikroElektronika Hexiwear platform. User should reprogram (once) the OpenSDA application of their Docking station before programming or debugging their Rapid IoT board.

To download for Docking station DAP-LINK for Rapid IoT go online at: www.nxp.com/rapid-iot

Select from the PRODUCTS tab Rapid IoT prototyping kit.

Go to SOFTWARE & TOOLS tab and select Docking Station DAP-LINK for SLN-RPK-NODE.

You will be asked to sign-in/up with a free NXP user-account.

When the download of the file completed, copy the Docking Station DAP-LINK firmware in your MCUXpresso Workspace on your computer HDD.

To reprogram the OpenSDA Application from the MikroElektronika Docking-station follow those steps (Rapid IoT board should not be mounted to the Docking station):

- Set the Power switch of the Docking station (red below) to OFF.
- Set the jumper switches of the Docking station to 11001000.
- Connect your Docking station to the USB port of your computer.
- Keep MK64 RESET button of the Docking station pressed, while setting the Power switch of the Docking station to ON, wait 2 seconds then release the MK64 RESET button.

Your computer will detect a new Mass Storage drive named MAINTENANCE and automatically install the appropriate drivers.

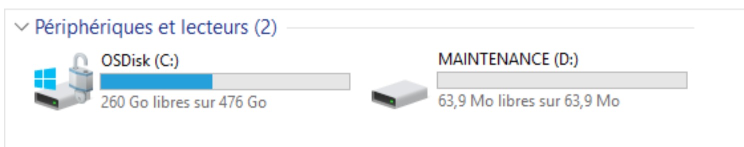


Figure 106. MAINTENANCE MSD drive

- From your computer file explorer, drag-n-drop or copy-paste into the MAINTENANCE drive the binary file k20dx_rapid_iot_if_crc_legacy_0x8000.bin.

Wait for the download to complete.

- Set the Power switch of the Docking station (red below) to OFF.
- Set the Power switch of the Docking station (red below) to ON.

Your computer should detect new Mass Storage drive named IOT-DAPLINK and automatically install the appropriate drivers.

Figure 107. IOT-DAPLINK MSD drive

NOTE: You can download those 3rd parties OpenSDA debug-only Applications at the address below:

- J-Link OpenSDA Generic Firmware v2.1 at: https://www.segger.com/downloads/jlink/
- Pemicro DEBUG_OpenSDA_for_MBED_Bootloader_by_Pemicro_v108_v2.1.bin at: http://www.pemicro.com/opensda/

IMPORTANT: Before connecting your Rapid IoT to the MikroElektronika Docking- station or changing the Jumper Switches make sure to set the Power switch of the Docking station (red below) to OFF.

Plug your Rapid IoT to the Docking station aligning carefully the board to board connectors (yellow below).

Figure 108. Docking station connector

The OpenSDA circuitry embedded in the Mikroelektronika Hexiwear Docking-station (MIKROE-2094) can program, serial-monitor and debug alternately the K64F and the KW41Z target MCUs featured in the Rapid IoT.

Set the jumper switches of the Docking station to 11001000 to program or debug the Application/K64F controller.

Figure 109. Docking Station Jumper Switch configuration to debug K64F

Set the jumper switches of the Docking station to 00111000 to program or debug the Wireless/KW41Z controller.

Figure 110. Docking Station Jumper Switch configuration to debug KW41Z

3.19. Mikroelektronika Click^{TM} add-on boards

Thanks to MikroElektronika Docking-station and Battery-pack, SLN-RPK-NODE is hardware compatible with every MikroElektronika Click add-on boards, which portfolio includes over 300 products, to extend its sensing, control or connectivity capabilities.

SLN-RPK-NODE drivers and application examples for selected Click add-on boards are available through the Rapid IoT Studio Web/GUI programming-tool and the offline SDK package for NXP MCUXpresso IDE.

At launch, SLN-RPK-NODE will fully support the 4x4 RGB Click through Rapid IoT Studio and MCUXpresso tool suites to emulate some lighting applications.

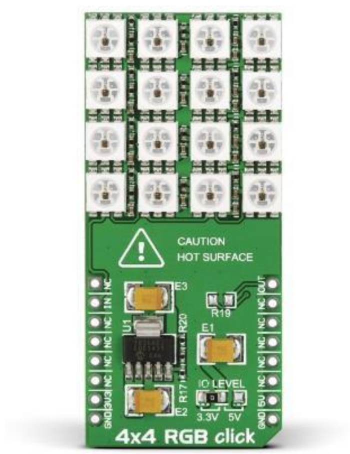


Figure 111. 4X4 RGB Click^{TM}

Our team will periodically release new SLN-RPK-NODE drivers and application examples for more ClickTM add-on boards through those two platforms

3.20. NXP Modular IoT Gateway

Out-Of-The-Box SLN-RPK-NODE connects through Thread radio with the i.MX6UL/ULL Gateway (included in the NXP SLN-IOT-GPI kit) to monitor and control its sensors and actuators from the Cloud. According to Thread standards, one single gateway can monitor and control up to 200 Rapid IoT end-devices.

Figure 112. i.MX6UL/ULL Modular IoT Gateway

IMPORTANT: The firmware of the gateway included in the NXP SLN-IOT-GPI kit might have to be updated first (once) to fully support our new SLN-RPK- NODE device. Follow the detailed instructions below including new gateway configuration.

Search, download and install the modular gateway Android or iOS phone App from the Google or Apple App store.

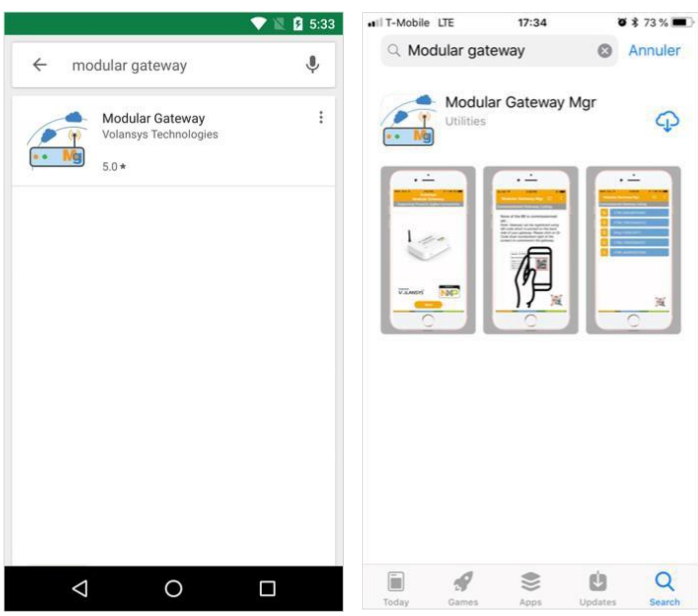


Figure 113. Modular Gateway Android/iOS App Download and Install

Then launch the modular gateway phone App from your phone/tablet and press Next.

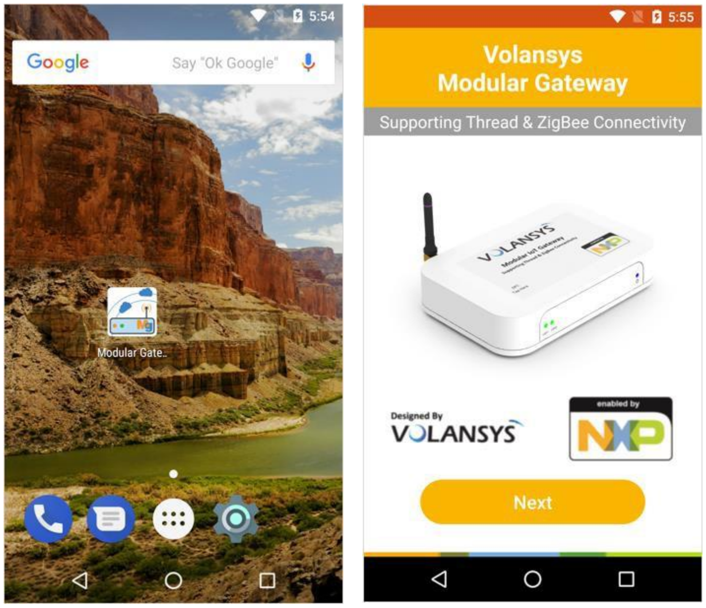


Figure 114. Launch modular gateway phone App

To the question Allow Modular Gateway Mgr to make and manage phone calls choose Allow.

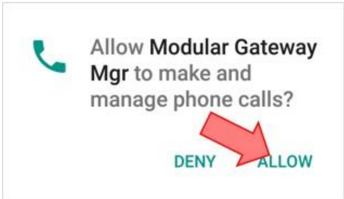


Figure 115. Authorize Phone calls

Register your gateway in the phone App with the Serial Number printed on its back, check the box Remember me and press Submit.

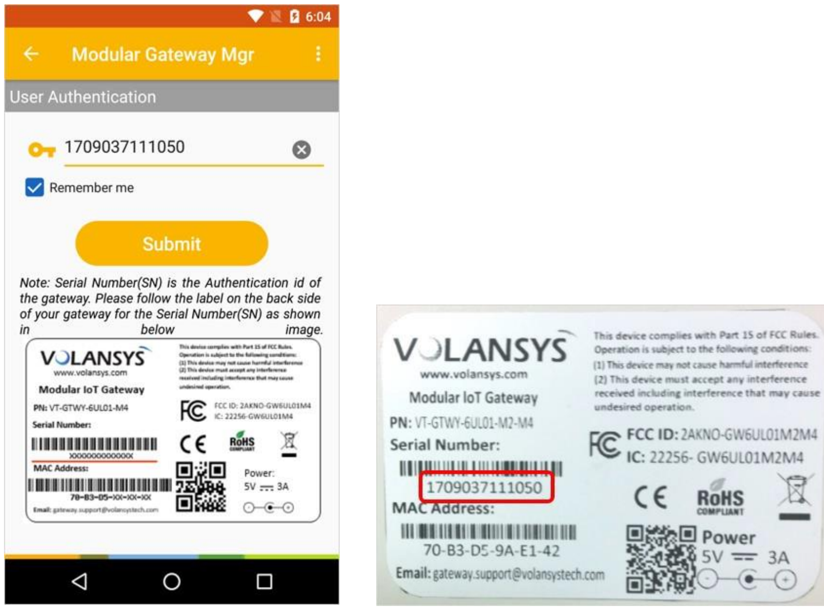


Figure 116. Register gateway in the phone App

If you are registering your gateway for the first time, you will be required to fill out some small details to register the NXP Modular Gateway. An email will be sent to you with a verification code that needs to be added into the OTP field (One-Time-Password).

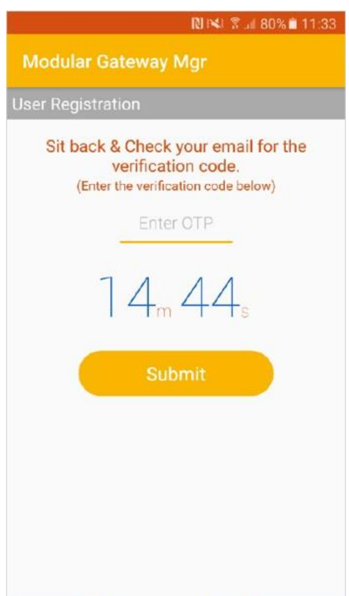


Figure 117. Verification code screen

On successful authentication, user is redirected to the list of registered gateways. If no gateway is already commissioned the following screen will invite the user to scan the QR code printed in the back of the gateway by selecting the icon in the bottom right corner.

Figure 118. New Gateway Commissioning

To the question Allow Modular Gateway Mgr to take pictures and record video select Allow.

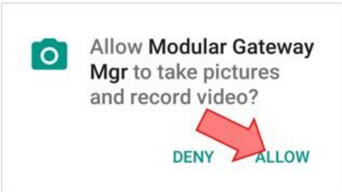


Figure 119. Authorize camera access

Then scan the QR code printed in the back of the gateway to finalize its registration.

Figure 120. Gateway QR code scan

Now we are ready to connect the Gateway to the Cloud over Ethernet or WiFi.

To connect the Gateway to the Cloud over Ethernet:

Plug one end of the Ethernet cable to the router and the other end to the Gateway Ethernet port.

Figure 121. Connect Gateway to internet over Ethernet

Plug the power-adapter to the outlet and to the Gateway D/C power input port.

Figure 122. Power-on Gateway

During Gateway startup the LED located in the front panel will change as follow:

- The right power LED will turn BLUE confirming that power is supplied to the Gateway
- LED2 will turn solid Green to confirm Gateway is powered on and NFC/BLE provisioning are disabled
- LED1 will sequentially turn Orange during Cloud connection then solid Green to confirm that Gateway is connected to the Cloud. LED1 turning RED indicates no IP address is assigned or no internet connectivity. For Ethernet connected Gateway, User must confirm the LED1 as solid green to further go ahead

Figure 123. Gateway connected to Cloud over Ethernet

To connect the Gateway to the Cloud over WiFi:

Plug out the Ethernet cable from the Gateway Ethernet port.
Plug the power-adapter to the outlet and to the Gateway D/C power input port.

Figure 124. Power-on Gateway

During Gateway startup the LEDs located in the front panel will change as follow:

- The right power LED will turn BLUE confirming that power is supplied to the Gateway
- LED2 will turn GREEN confirming Gateway is powered-on and NFC/BLE provisioning is disabled
- LED1 will turn RED indicating no IP address was assigned or Gateway doesn't have valid Wi-Fi credential and no internet connectivity is available.

Figure 125. Gateway not connected to the Cloud

We will now provide the Wi-Fi credentials to the Gateway through NFC/BLE commissioning.

Select your Gateway from the list of the Commissioned Gateway in the phone App.

Figure 126. Commissioned Gateway Selection

Open the Gateway Options by selecting the icon located in the top right corner.

Figure 127. Gateway Options

From the menu select Gateway Properties.

Figure 128. Gateway Properties

The Gateway Property menu provides useful information such like your Gateway Name and the version of the Firmware (when connected to the Cloud).

Choose **WiFi** for the Network Interface.

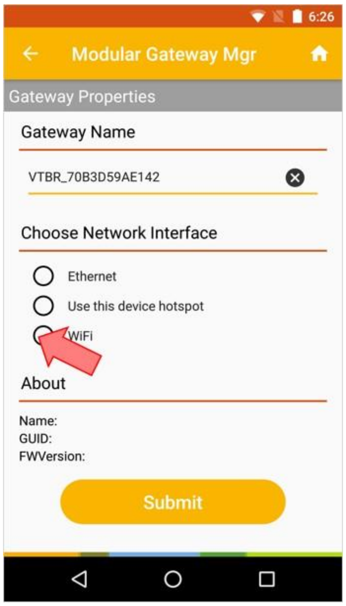


Figure 129. WiFi Connectivity

To the question Allow Modular Gateway Mgr to access this device's location select Allow.

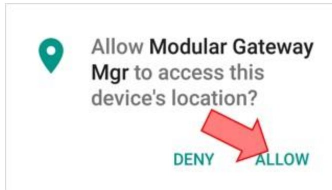


Figure 130. Authorize device's location access

Select from the list your WiFi network and enter its Password.

Figure 131. WiFi SSID and Password

Scroll down to choose the commissioning interface.
Select NFC if you have an Android phone NFC-enabled or BLE otherwise and press Submit.

Following instructions will consider an NFC-enabled equipment.

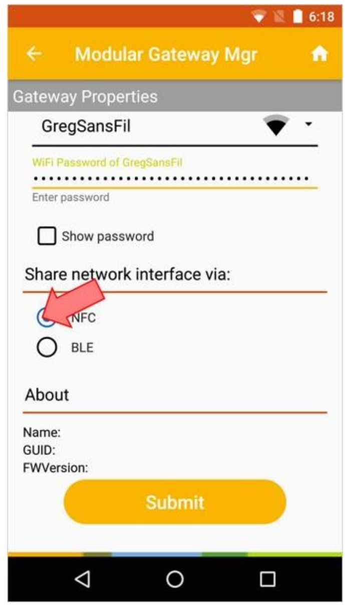


Figure 132. Transfer WiFi details to the Gateway over NFC

The phone App will display the instructions to initiate the commissioning on the Gateway.

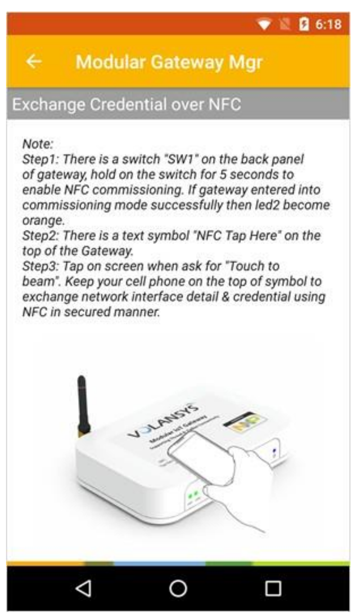


Figure 133. WiFi Provisioning over NFC

At the back of the Gateway, keep the SW1 button pressed for 5 seconds.

Figure 134. ***Initiate Gateway Commissioning***

LED2 will turn ORANGE confirming Gateway NFC/BLE provisioning is enabled.

Figure 135. ***LED for Gateway Commissioning***

Position your phone on the top of the Gateway near the inscription NFC Tap Here.
Your phone should emit a sound to confirm that the NFC reader of the Gateway has been detected.
Touch the phone screen to start the WiFi credential transfer over NFC.

Figure 136. Phone positioning for NFC Commissioning

A message will be displayed in the phone App during the NFC transfer.

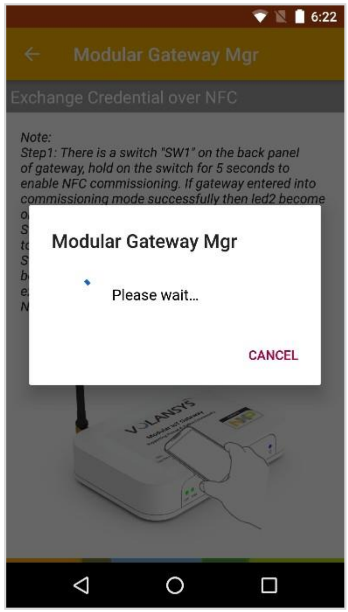


Figure 137. ***NFC Commissioning in progress***

When the NFC commissioning is completed, the phone App should display a confirmation message. Select OK to close the message.

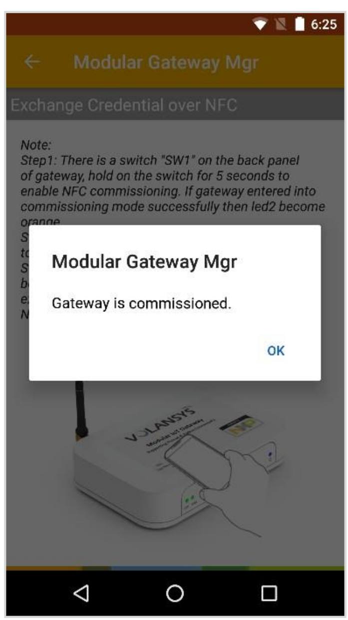


Figure 138. NFC Commissioning successful

LED1 will turn solid GREEN confirming that the Gateway is connected to the Cloud over WiFi. You can now remove your phone from the top of the Gateway.

Figure 139. ***Gateway connected to the Cloud over WiFi***

Congratulation, your Gateway is now successfully connected to the Cloud.

Now let’s update the firmware of the Gateway.

Select your Gateway from the list of the Commissioned Gateway in the phone App.

Figure 140. Commissioned Gateway Selection

Open the Gateway Options by selecting the icon located in the top right corner.

Figure 141. Gateway Options

From the menu select Gateway Firmware Update.

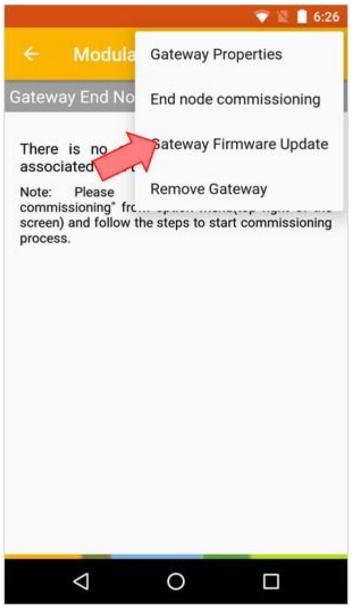


Figure 142. Gateway Firmware Update

Now that the Gateway is connected to the Cloud, the version of the Firmware programmed in the Gateway is available, in this case 1.8.1.
The available Firmware available through Over-The-Air Update are listed below.
Select the first option full_x.x.x_golden.swu (ota_demo_test is for test purpose).

NOTE: RPK-NODE is supported by the Gateway firmware 1.8.5 or higher

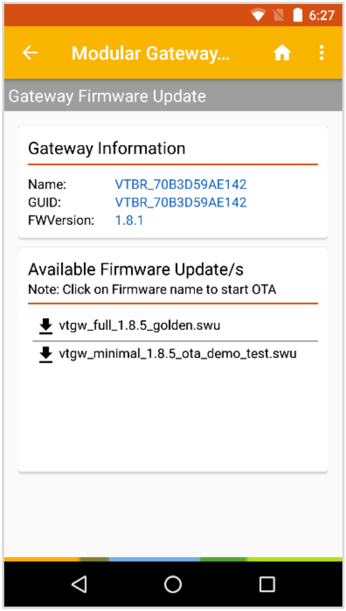


Figure 143. Gateway Firmware Information and Selection

Start the Firmware Update by pressing UPDATE in the confirmation window.

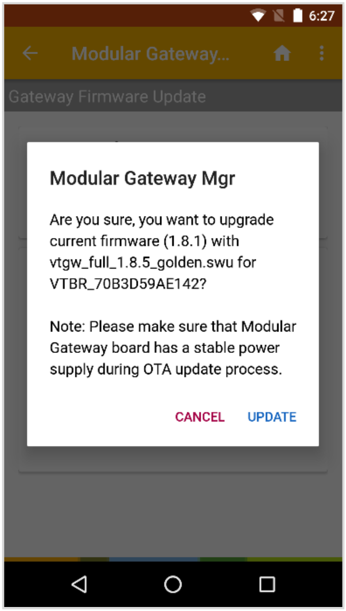


Figure 144. ***Gateway Firmware Update Confirmation***

A progression bar will appear at the bottom of the screen to indicate the progress of the update.

The Gateway firmware is large and takes some time to download and program in the Gateway memory

IMPORTANT: Don't power-off your gateway or turn-off the internet connection during the update.

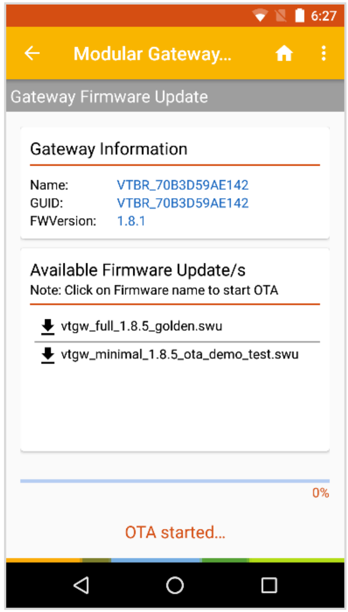


Figure 145. Gateway Firmware Information and Selection

Your Gateway will reboot after reprogramming and the phone App will display a confirmation message that the Update ended up successfully.

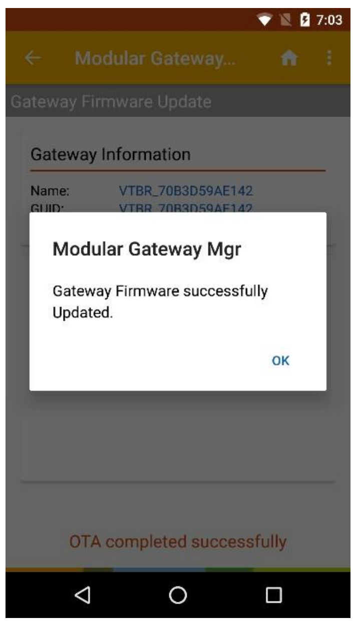


Figure 146. ***Update successful***

If you return to the Gateway Properties menu in the phone App, your Gateway Firmware Information should now indicate the Firmware version selected for the update, in this case 1.8.5

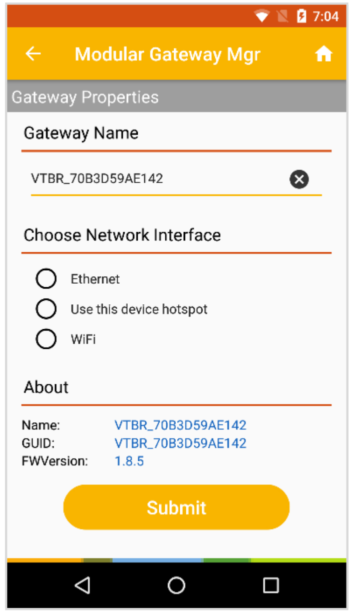


Figure 147. Update successful

Congratulation, your Gateway is now successfully updated with the latest Firmware version

Now let's connect our SLN-RPK-NODE to the Gateway over Thread

The application programmed in factory in the SLN-RPK-NODE supports both Bluetooth and Thread connectivity alternately. By default, Bluetooth radio is enabled so we must activate Thread from the application settings to connect SLN-RPK-NODE to the Gateway.

From the Welcome screen touch the Electrode T1 to open the Application screen.
Then touch the Electrode T2 to open the Settings screen.
Next touch the Electrode T1 to access the Bluetooth menu.
Finally touch the Electrode T2 to reach the Thread menu.

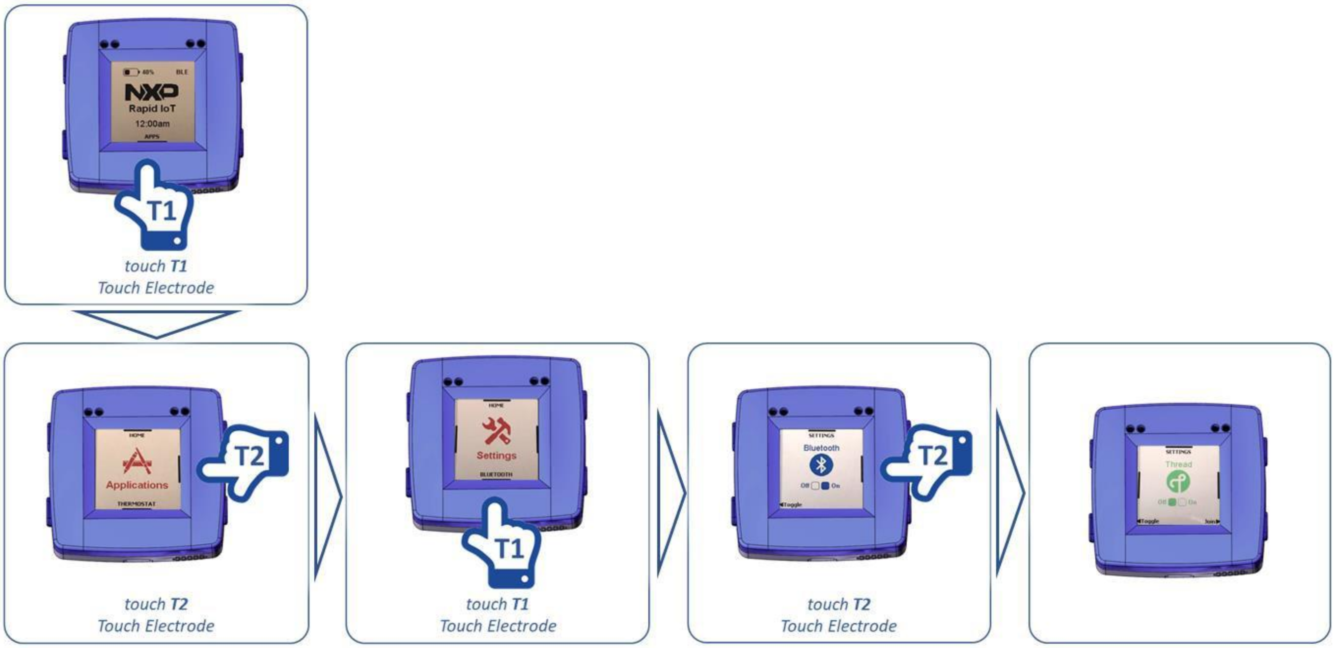


Figure 148. Access Thread on SLN-RPK-NODE

Now press the push button SW4 to activate Thread radio (and automatically disable Bluetooth radio).

After few seconds a message will confirm the selection and indicates that SLN-RPK-NODE must restarts to start Thread radio

Press the push button SW2 to restart the board

Figure 149. Access Thread on SLN-RPK-NODE

Open the Gateway Options by selecting the icon located in the top right corner.

Figure 150. Gateway Options

From the menu select End node commissioning.

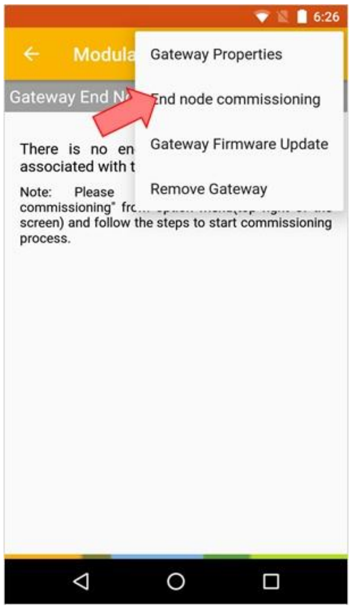


Figure 151. End node commissioning

NFC Commissioning will be available in the next release.
For now we will share the SLN-RPK-NODE MAC address manually with the Gateway.

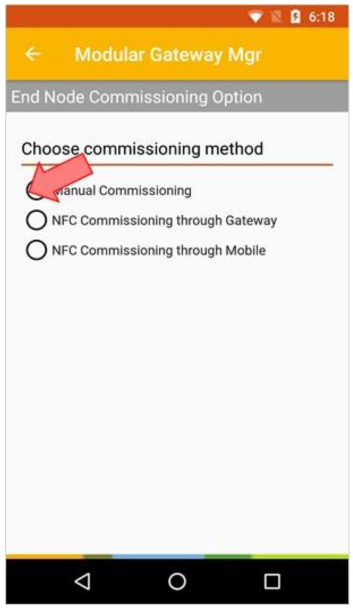


Figure 152. Manual Commissioning

Select Thread and enter the SLN-RPK MAC address printed at the back of the SLN-RPK-NODE casing and press Submit**.**

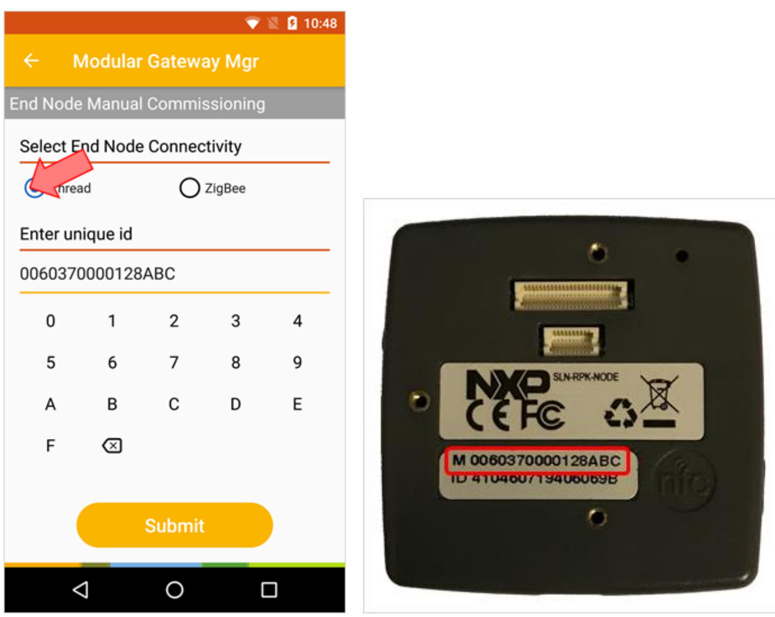


Figure 153. SLN-RPK-NODE MAC Address

The phone App will display shortly a confirmation message that the SLN-RPK-NODE is successfully commissioned for Thread and open a new window with the instruction to connect each type of End Node supported by the Gateway (including the Rapid IoT board).

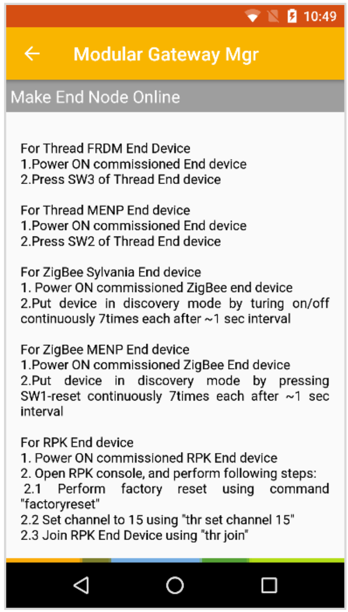


Figure 154. Connect End Node

From the SLN-RKP-NODE Thread menu press the push button SW2 to join the Thread network. The White LED will start blinking quickly to confirm that SLN-RPK-NODE is trying to connect.

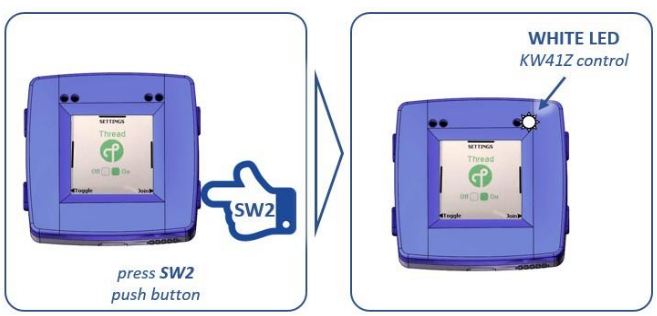


Figure 155. Join Network

After few seconds the phone App will display a message to confirm that SLN-RPK-NODE joined successfully the Thread network and ask if we want to commission/connect another device.
Choose NO to display the list of the connected End nodes.

Figure 156. Connection successful

Select your Rapid IoT board from the list of the connected device thanks to its unique MAC address.

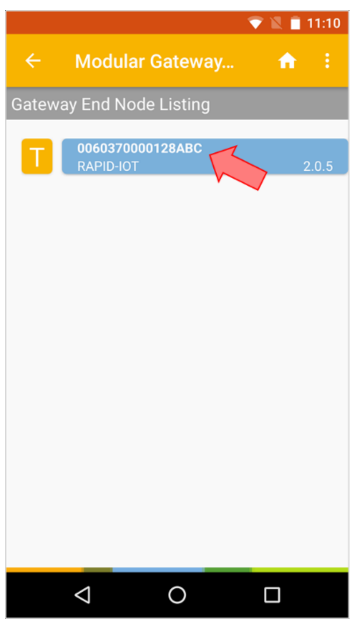


Figure 157. SLN-RPK-NODE Selection

The phone App will display a new interface to monitor and control SLN-RPK-NODE sensors and actuators from the Cloud through Thread connectivity.

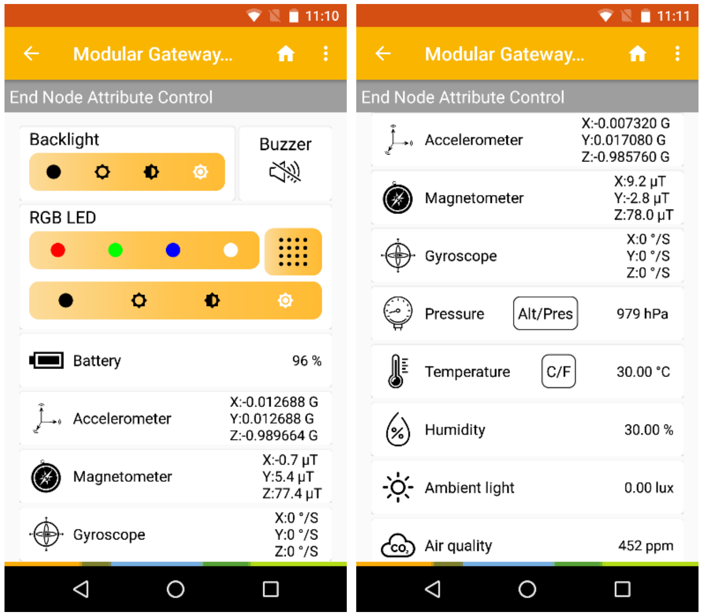


Figure 158. SLN-RPK-NODE Monitor and Control Gateway Interface

Congratulation, your SLN-RPK-NODE is now successfully connected to the Cloud through Thread!!

4. Rapid IoT Advanced features

4.1. Embedded Bootloader

SLN-RPK-NODE MCUs are both pre-programmed in factory with an innovative Bootloader to easily update their firmware from a computer over USB or from a smartphone over BLE, or restore from the embedded serial Flash memory their default application.

At every startup the bootloader for the Application and the Wireless controllers will check the integrity of the application loaded in their internal flash memory, as well as the push button status.

If no push button is pressed during reset, the K64F and KW41Z bootloaders will execute the following LED sequences before running the preloaded K64F and KW41Z applications.

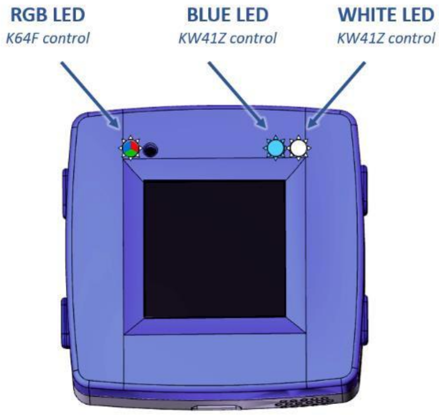


Figure 159. Boot LEDs

RGB LED	BLUE LED	WHITE LED
GREEN	OFF	OFF
GREEN	BLINK	BLINK
BLUE/RED	BLINK	BLINK
BLUE	OFF	OFF

Table 29. Boot LED Sequences

During reset, the push button combination below will launch the following (re)programming mode:

- Keep SW1 pressed during startup/reset to launch Application/K64F MCU firmware recovery.
- Keep SW2 pressed during startup/reset to launch Wireless/KW41Z MCU firmware recovery.
- Keep SW3 pressed during startup/reset to launch the USB MSD Programming mode.

Figure 160. Bootloader push buttons

4.1.1. Factory application recovery for the Application/K64F MCU

Rapid IoT embeds a serial Flash memory pre-programmed with the factory application for the Application/K64F MCU. If you accidentally erased the application programmed in the internal flash of the Application/K64F MCU, follow those steps to recover the default factory application:

- Check your Rapid IoT battery is full or provide power through USB connector.
- Keep SW1 pressed, while pushing shortly SW5/Reset button.
- Wait 1-2s for RGB LED to turn green then release SW1 button.

Figure 161. Instructions for K64F application recovery

RGB LED will blink green during the reprogramming of K64F internal flash with the factory application and automatically reset, when it is ready.

Don't press any button or reset the board until reprogramming completed successfully or the application will be corrupted and you will have to restart the programming from the beginning.

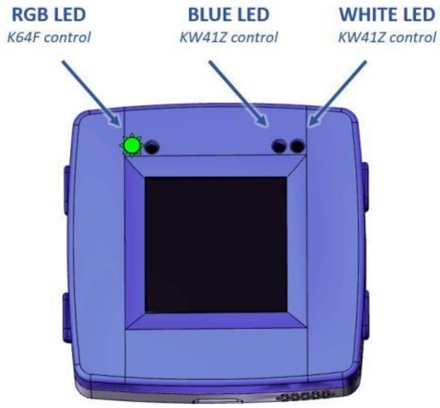


Figure 162. K64F Recovery LED

RGB LED	BLUE LED	WHITE LED
GREEN BLINK	OFF	OFF

Table 30. K64F Recovery LED Sequence

4.1.2. Factory application recovery for the Wireless/KW41Z MCU

Rapid IoT embeds a serial Flash memory pre-programmed with the factory application for the Wireless/KW41Z MCU. If you accidentally erased the application programmed in the internal flash of the Wireless/KW41Z MCU, follow those steps to recover the default factory application:

- Check your Rapid IoT battery is full or provide power through USB connector.
- Keep SW2 pressed, while pushing shortly SW5/Reset button.
- Wait 1-2s for RGB LED to blink Green then release SW2 button.

Figure 163. Instructions for KW41Z application recovery

White LED will blink during the reprogramming of KW41Z internal flash with the factory application and automatically reset, when it is ready.

Don’t press any button or reset the board until reprogramming completed successfully or the application will be corrupted and you will have to restart the programming from the beginning.

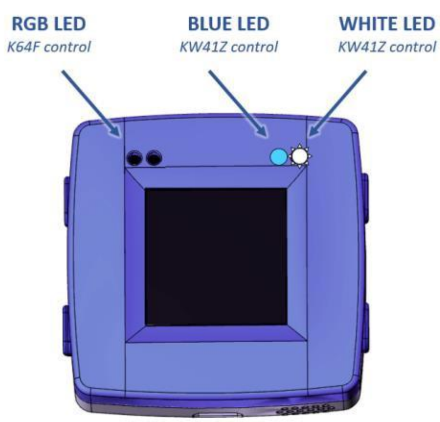


Figure 164. KW41Z Recovery LED

RGB LED	BLUE LED	WHITE LED
GREEN	ON	BLINK
BLUE/RED	ON	BLINK
OFF	ON	BLINK
OFF	BLINK SYNCHRO 4 TIMES	

Table 31. KW41Z Recovery LED Sequence

4.1.3. USB Mass-Storage Device Programming for both K64F/KW41Z MCUs

SLN-RPK-NODE Application and Wireless MCU are pre-programmed in factory with a Bootloader to easily update their application through USB. If you want to reprogram K64F or KW41Z internal flash with a new application, simply follow those steps:

- Connect one end of the provided USB cable to the computer and the other end to the micro USB type-B connector of the SLN-RPK-NODE.
- Keep SW3 button pressed while pushing shortly SW5/Reset button.
- Wait 1-2s for RGB LED to blink Green then release SW3 button.

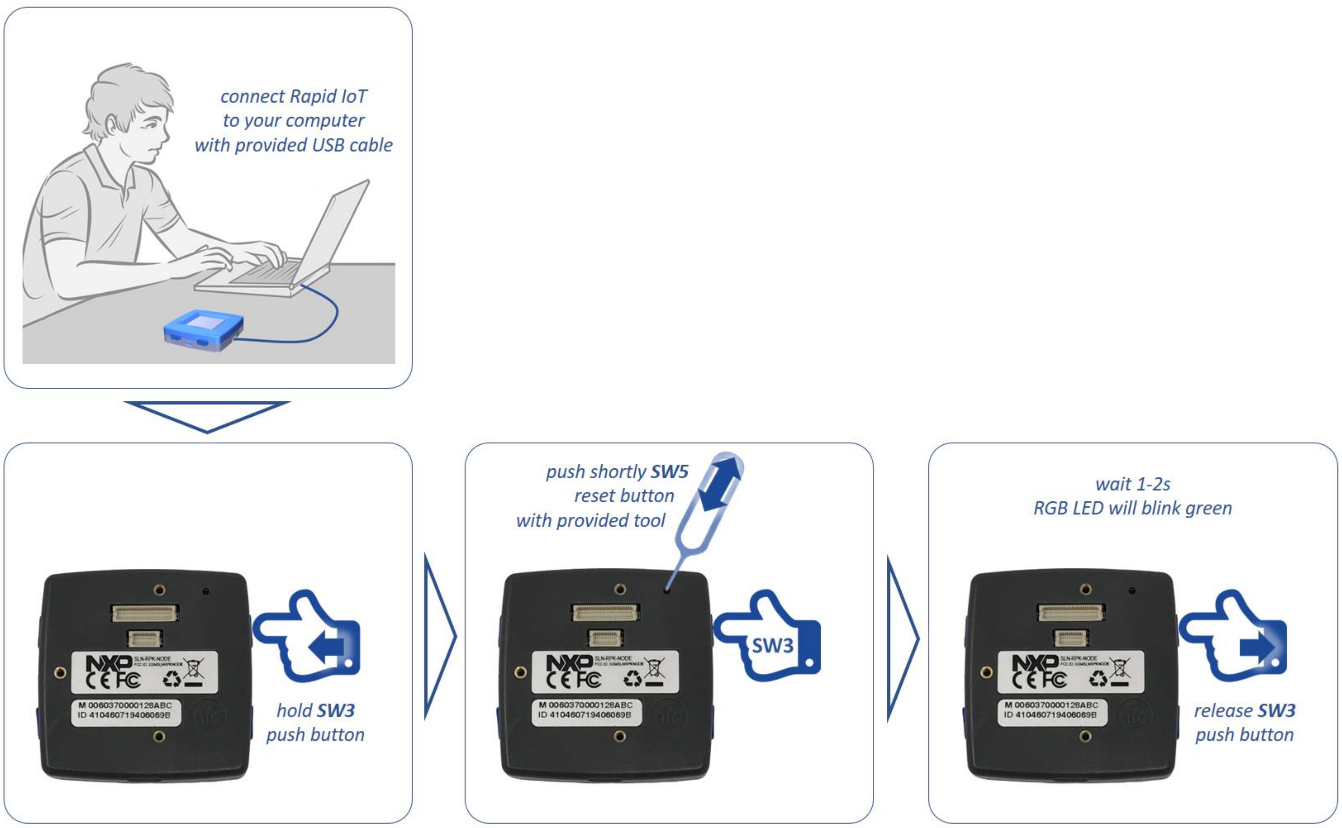


Figure 165. Instructions for USB Mass Storage Device Programming

RGB LED will blink green and your computer will detect a new Mass Storage drive and automatically install the appropriate drivers.

- From your computer file explorer, drag-n-drop or copy-paste into the Mass Storage drive a precompiled application (*.bin file) for the K64F or the KW41Z MCU.

Figure 166. Instructions for pushing a new application through USB

Bootloader will automatically identify the MCU target to reprogram thanks to the binary file signature.

RGB LED will blink purple during download, then blink blue during programming of the serial Flash memory and automatically reset.

If you pushed a new binary for the Application/K64F MCU, RGB LED will blink green during the reprogramming of K64F internal flash with the new application and automatically reset, when ready.

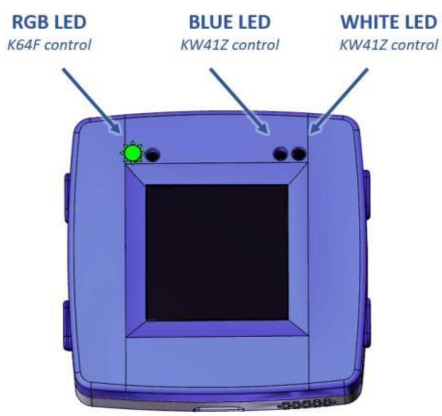


Figure 167. K64F Recovery LED

RGB LED	BLUE LED	WHITE LED
PURPLE BLINK	OFF	OFF
BLUE BLINK	OFF	OFF
GREEN BLINK	OFF	OFF

Table 32. K64F USB Programming LED Sequence

If you pushed a new binary for the Wireless/KW41Z MCU, White LED will blink during the reprogramming of KW41Z internal flash with the new application and automatically reset, when ready.

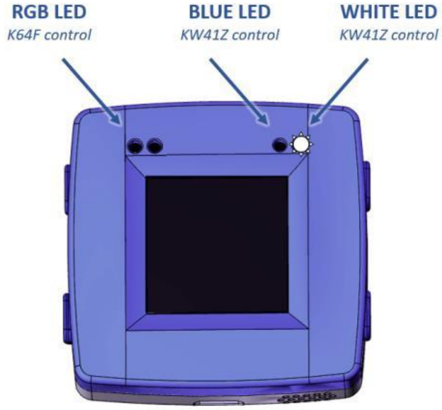


Figure 168. KW41Z Recovery LED

RGB LED	BLUE LED	WHITE LED
PURPLE BLINK	OFF	OFF
BLUE BLINK	OFF	OFF
BLUE/RED	BLINK	ON
OFF	OFF	BLINK

Table 33. KW41Z USB Programming LED Sequence

Don’t press any button, reset the board or remove the USB cable until reprogramming completed successfully or the application will be corrupted and you will have to restart the programming from the beginning. Depending from the size of the file, reprogramming may take up to one minute.

4.1.4. (Re)Programming of the Factory Applications in the Serial Flash

The K64F bootloader also supports the (re)programming of both K64F/KW41Z Recovery applications stored in the serial Flash memory through USB. Simply follow those steps to (re)program/replace those firmware in serial Flash memory:

- Connect one end of the provided USB cable to the computer and the other end to the micro USB type-B connector of the SLN-RPK-NODE.
- Keep SW1, SW2 and SW3 buttons pressed while pushing shortly SW5/Reset button.
- Wait 1-2s for RGB LED to blink Yellow then release SW1, SW2 and SW3 buttons.

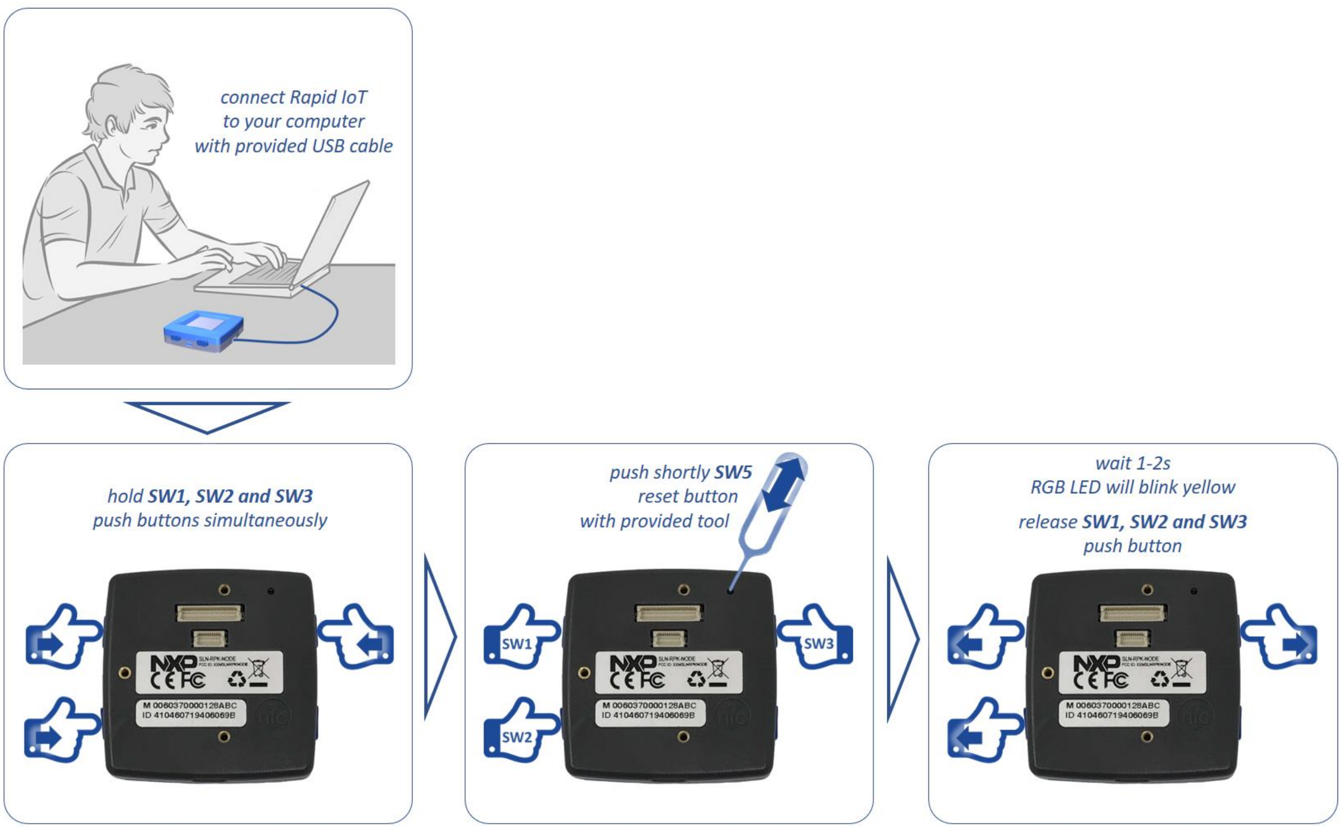


Figure 169. Instructions for reprogramming the Recovery Applications through USB

RGB LED will blink yellow and your computer will detect a new Mass Storage drive and automatically install the appropriate drivers.

- From your computer file explorer, drag-n-drop or copy-paste into the Mass Storage drive a precompiled application (*.bin file) for the K64F or the KW41Z MCU.

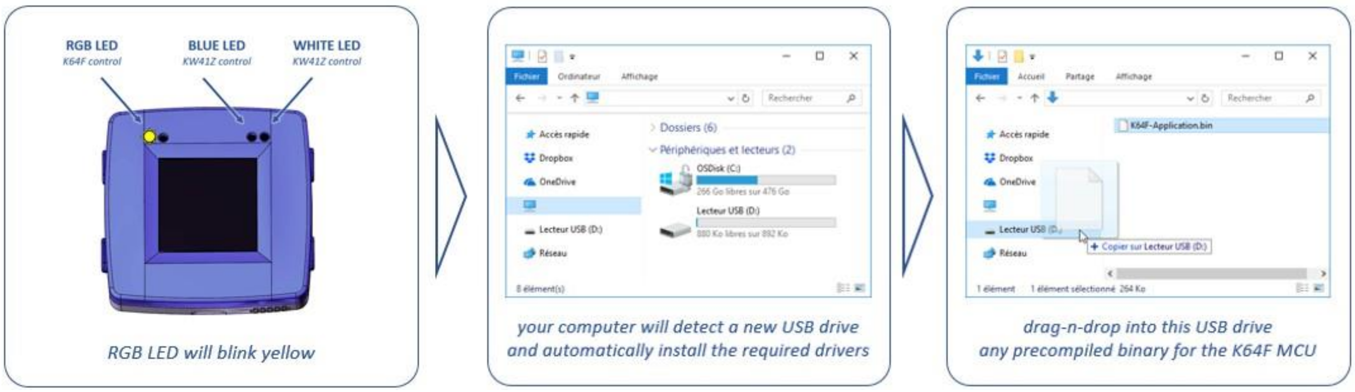


Figure 170. Instructions for pushing a new recovery application through USB

Bootloader will automatically identify thanks to the binary file signature the MCU target and program the recovery firmware at the right location of the serial Flash memory.

RGB LED will blink purple during download, then blink blue during programming of the serial Flash memory and automatically reset.

SLN-RPK-NODE will now launch the new application just programmed in the internal memory of the Application/K64F and the Wireless/KW41Z MCUs.

4.1.5. (Re)Programming of K64F and KW41Z Bootloaders

The Rapid IoT bootloaders are running from the internal Flash memory of the K64F and the KW41Z MCUs. They can be (re)programmed with the Docking station following one of the two methods described below.

4.1.5.1. Using Segger J-Link commander and precompiled binary files

Segger proposes a free software flashing tool controled by command lines compatible with the OpenSDA circuitry embedded in the Docking station thanks to a specific firmware, which emulates a JLink-lite probe. Follow those steps to flash the binaries of the Rapid IoT bootloaders with Segger tools:

- Download the latest version of the J-Link Software and Documentation Pack (min v6.32) and the J-Link OpenSDA Generic Firmware v2.1 at: https://www.segger.com/downloads/jlink/
- Install the J-Link Software package on your computer

The latest version of the bootloader binaries kboot_k64f.bin and kboot_kw41z.bin are available in the Rapid IoT SDK package.

- To download for free Rapid IoT SDK package go online at: www.nxp.com/rapid-iot
- Select from the PRODUCTS tab Rapid IoT prototyping kit.
- Go to SOFTWARE & TOOLS tab and select the latest version of the Rapid IoT SDK package.
- You will be asked to sign-in/up with a free NXP user-account.

- Copy those two binary files in the folder C:\Program Files (x86)\SEGGER\JLink_V632\
- Set the Power switch of the Docking station (red below) to OFF.
- Plug the USB cable to your computer USB port and to the Docking Station micro-USB connector.
- Plug your Rapid IoT to the Docking station aligning carefully the board to board connectors (yellow below).

Figure 171. Docking station connector

- Set the jumper switches of the Docking station to **11001000** to debug the Application/K64F MCU

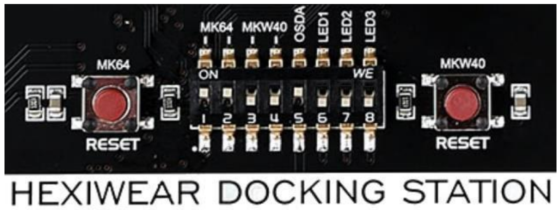


Figure 172. Docking Station Jumper Switch configuration to debug K64F

- Keep MK64 RESET button of the Docking station pressed, while setting the Power switch of the Docking station to ON.

Your computer will detect a new Mass Storage drive named MAINTENANCE and automatically install the appropriate drivers

- From your computer file explorer, drag-n-drop or copy-paste into the MAINTENANCE drive the binary file OpenSDA_V2_1.bin.

Wait for the download to complete. Your computer should detect new peripherals and automatically install the appropriate drivers.

- Launch JLink Commander
- Type the command "connect"
- Enter the device "MK64FN1M0XXX12"
- Type "**S**" and press ENTER
- Select the default speed by pressing ENTER
- Type "erase" and press ENTER to erase the K64F memory
- Type "loadbin kboot_k64f.bin 0"and press ENTER to program the bootloader into K64F memory

Wait for the download to complete

- Close the program by typing **"**exit" and pressing ENTER

Congratulation you successfully programmed the bootloader in the K64F internal memory!!

Now let's configure the Docking station to flash the bootloader in the KW41Z internal memory

- Set the Power switch of the Docking station to OFF
- Set the jumper switches of the Docking station to 00111000 to debug the Application/KW41Z MCU

Figure 173. Docking Station Jumper Switch configuration to debug KW41Z

- Set the Power switch of the Docking station to ON
- Launch JLink Commander
- Type the command “connect”
- Enter the device “MKW41Z512XXX4”
- Type “S” and press ENTER
- Select the default speed by pressing ENTER
- Type “erase” and press ENTER to erase the KW41Z memory
- Type “loadbin kboot_kw41z.bin 0” and press ENTER to program the bootloader into KW41Z

Wait for the download to complete

- Close the program by typing “exit” and pressing ENTER

Congratulation you successfully programmed the bootloader in the KW41Z internal memory!!

4.1.5.2. Using MCUXpresso and source-code projects

The SDK package for the Rapid IoT embeds the drivers for every components onboards as well as the project source-code for both K64F and KW41Z Bootloader and factory Applications for the MCUXpresso IDE. Follow those steps to reprogram the Rapid IoT bootloaders with MCUXpresso tools:

- Download for free NXP MCUXpresso IDE online at: www.nxp.com/MCUXpresso
- Select from the PRODUCTS tab MCUXpresso IDE.
- Go to DOWNLOADS tab and select the LATEST VERSION of the tool.
- You will be asked to sign-in/up with a free NXP user-account.
- Install MCUXpresso IDE on your computer

- Download for free Rapid IoT SDK package go online at: www.nxp.com/rapid-iot
- Select from the PRODUCTS tab Rapid IoT prototyping kit.
- Go to SOFTWARE & TOOLS tab and select the latest version of the Rapid IoT SDK package.
- You will be asked to sign-in/up with a free NXP user-account.

- Launch MCUXpresso IDE and select C:\MCUXpresso.Workspace\ as workspace location.

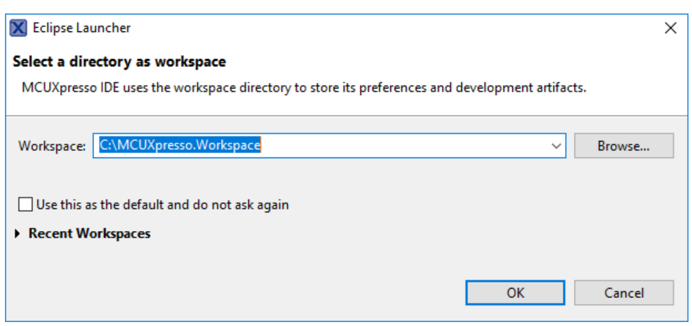


Figure 174. Define MCUXpresso default Workspace Location

- Extract the Rapid IoT SDK Package in the location C:\MCUXpresso.Workspace\
- From MCUXpresso Quickstart Panel, select Import project(s) from file system…

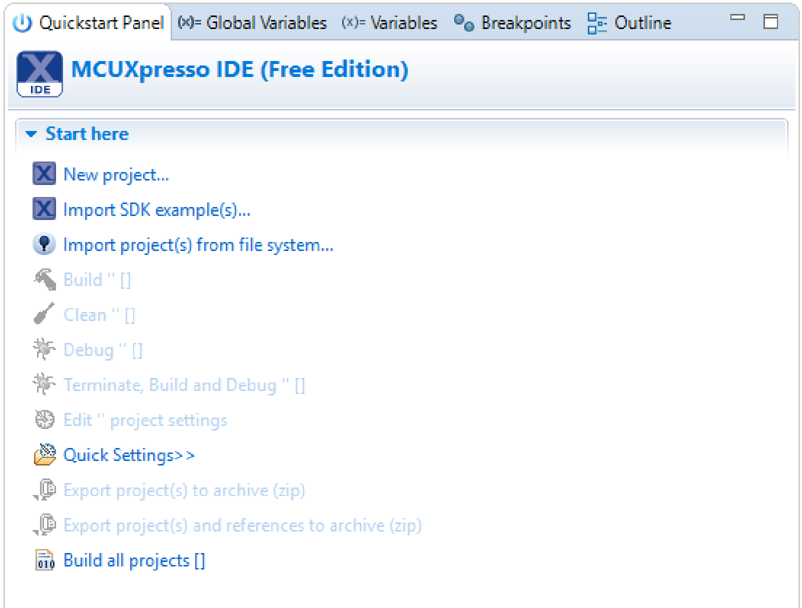


Figure 175. Import Project in MCUXpresso

- Under Project directory (unpacked) press Browse…

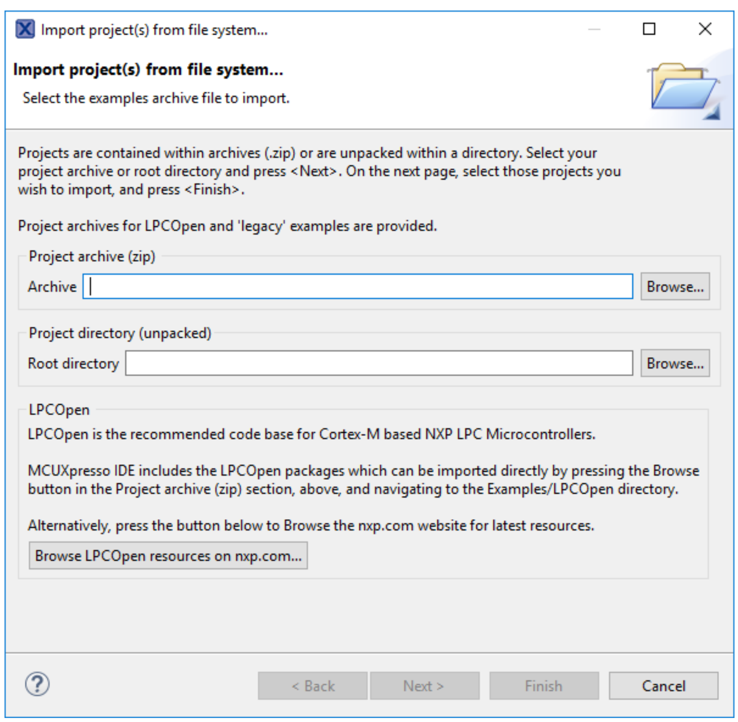


Figure 176. Browse project folder

- Select the folder \SDK_2.3.0_RapidIoT_MK64F12\boards\rpk_k64f\rapid_iot_apps\kboot and press OK.

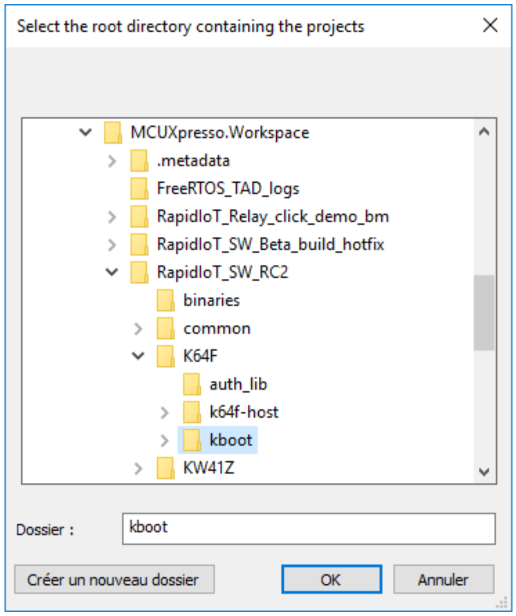


Figure 177. Select K64F Bootloader Project Folder

- In the new window, press Next.

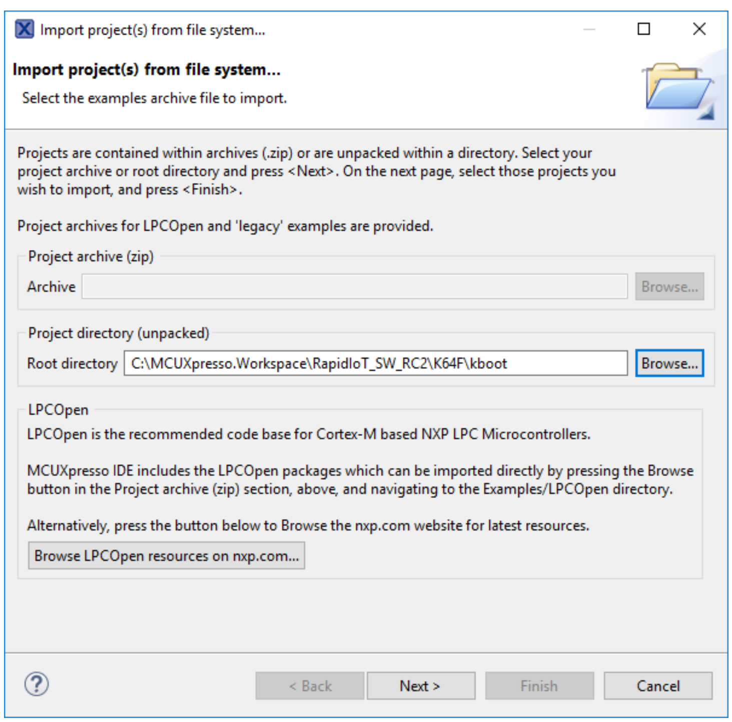


Figure 178. Search Projects from Folder

- Check the project kboot_k64f and make sure to uncheck the option Copy projects into workspace, then press Finish.

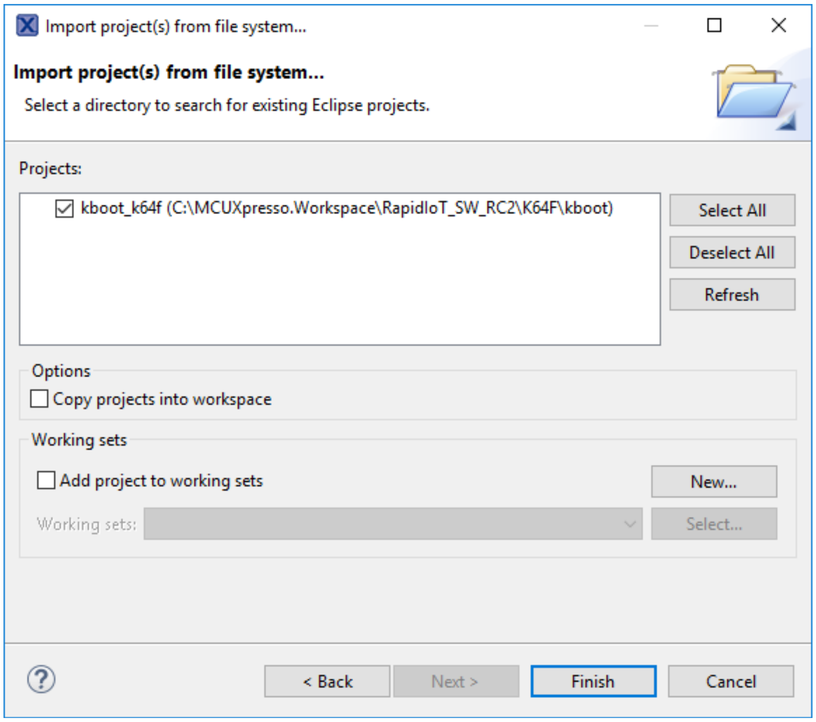


Figure 179. Finalize Project Import in MCUXpresso

- From MCUXpresso Project Explorer, select and expand the project kboot_k64F.

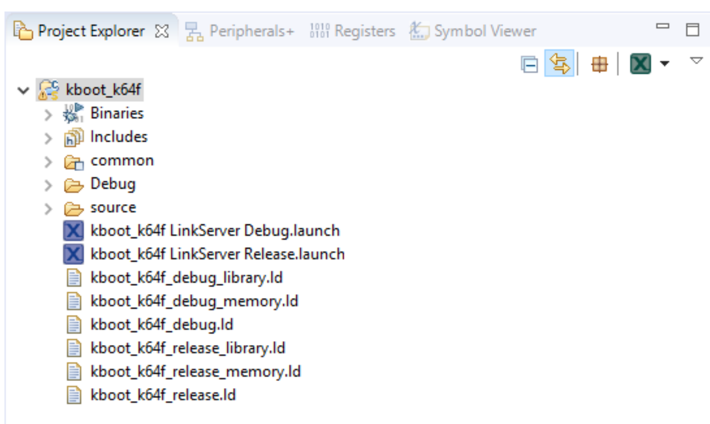


Figure 180. Activate Project in MCUXpresso

- From MCUXpresso Quickstart Panel, select Build ‘kboot_k64f’ [debug].

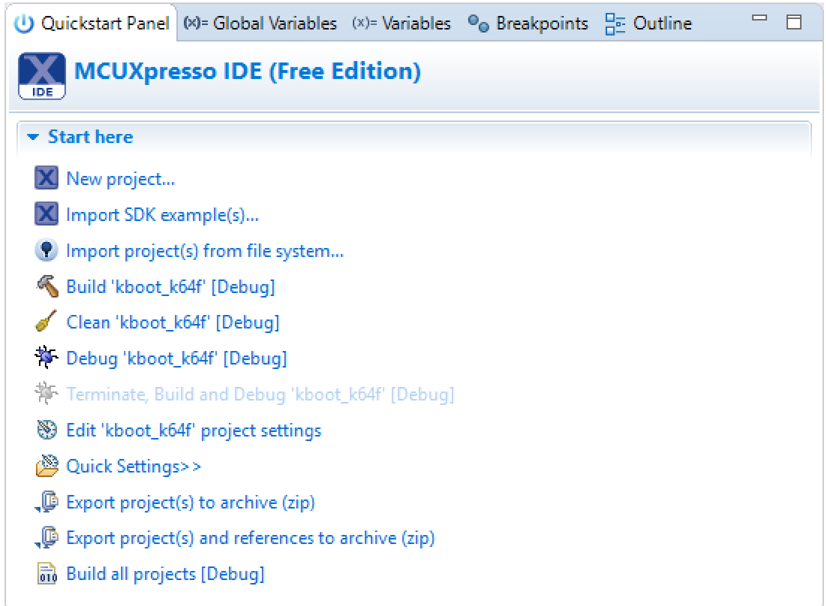


Figure 181. Build kboot_k64f project

Wait for the compilation to finish successfully. It may take few minutes the first time be patient.

Make sure that your Docking station is programmed with the Rapid IoT DAP-LINK debug interface. Instruction to reprogram your Docking station are described section 3.18.

- Set the Power switch of the Docking station (red below) to OFF.
- Plug your Rapid IoT to the Docking station aligning carefully the board to board connectors (yellow below).

Figure 182. Docking station connector

- Set the jumper switches of the Docking station to 11001000 to debug the Application/K64F MCU

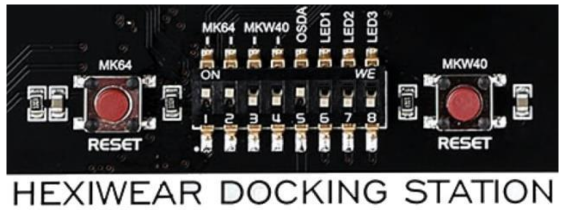


Figure 183. Docking Station Jumper Switch configuration to debug K64F

- Set the Power switch of the Docking station to ON
- From MCUXpresso Quickstart Panel, select Debug ‘kboot_k64f’ [debug]

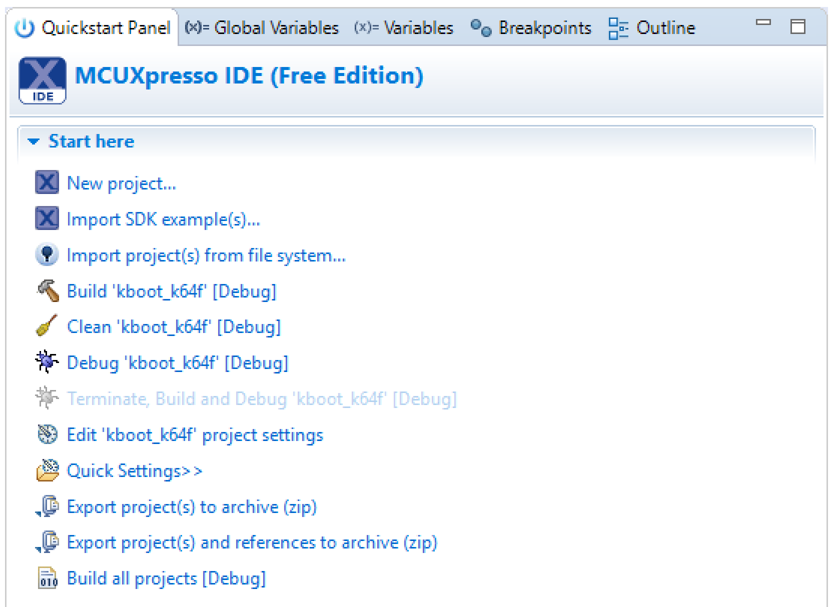


Figure 184. Debug kboot_k64f project

- In the message window select the probe CMSIS-DAP and press Debug.

Wait for the application to be programmed in the K64F internal flash and the debug session to start.

- From the debug view press the stop button to halt the debug session.

- Repeat the previous step to import KW41Z bootloader project located at \SDK_2.2_RapidIoT_MKW41Z4\boards\rpk_kw41z\rapid_iot_apps\kboot and debug it.

- Set the Power switch of the Docking station to OFF**.**
- Set the jumper switches of the Docking station to 00111000 to debug the Wireless/KW41Z MCU.

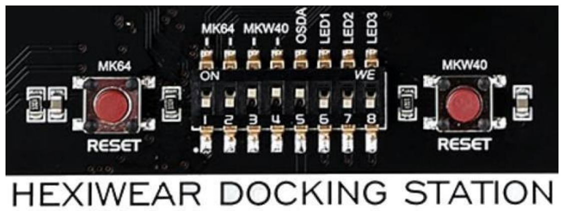


Figure 185. Docking Station Jumper Switch configuration to debug KW41Z

- Set the Power switch of the Docking station to ON.
- From MCUXpresso Quickstart Panel, select Debug ‘kboot_kw41z’ [debug].
- In the message window select the probe CMSIS-DAP and press Debug.

Wait for the application to be programmed in the KW41Z internal flash and the debug session to start.

- From the debug view press the stop button to halt the debug session.

Congratulation, you have successfully reprogrammed both K64F and KW41Z bootloaders!!

4.2. Rapid IoT Studio WEB/GUI-Programming tool

Rapid IoT Studio is a robust IoT platform for creating, deploying, and managing connected products. Rapid IoT Studio enables users to bring their IoT solutions to market quickly, affordably, and securely through a complete end-to-end experience.

The graphic below represents the typical architecture of connected products managed using Rapid IoT Studio.

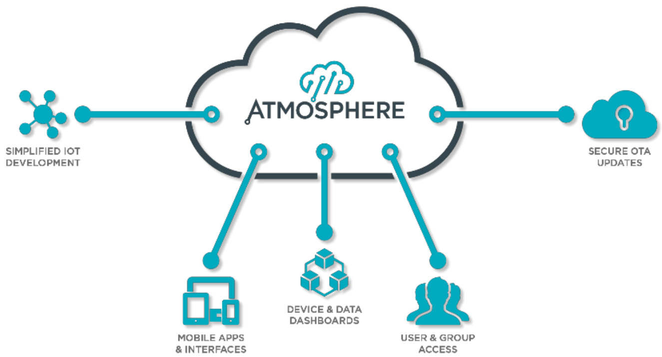


Figure 186. Rapid IoT Studio Tool Suite

All aspects of your connected product needs can be managed with Rapid IoT Studio no matter where you are in the process, from concept and prototype through deployment and management.

Unlike platforms that focus purely on developing embedded hardware or software, or managing devices, users, and data, Rapid IoT Studio accomplishes each of them simultaneously, providing the means to create and manage IoT solutions from start to finish. Rapid IoT Studio uses common connectivity protocols such as Wi-Fi and Bluetooth Low Energy to connect to your devices where it then receives device data, allowing for powerful aggregation, analysis, and reporting.

4.2.1. Introduction

Rapid IoT Studio is composed of the following suite of applications:

The Rapid IoT Studio Platform is cloud-based software that offers a complete IoT development and management experience. Through its easy-to-use development environment, configurable dashboard editor, and device management console, Rapid IoT Studio provides you a single secure platform for every area of your IoT solution.

The Rapid IoT Studio app is a mobile app available in the Apple App Store and Google Play Store. The app is an extension of the Rapid IoT Studio Platform, and enables you to use Rapid IoT Studio as you would from within a desktop browser, including registering and managing devices, viewing device data through dashboards, and more.

The Rapid IoT Studio Agent is a local application that runs in your computer's system tray, and acts as an intermediary between the computer's connectivity protocol drivers and the browser running Rapid IoT Studio. It is used for programming hardware, and enables you to program an end device from within the Rapid IoT Studio Platform.

4.2.2. Create a new Project

Visit the dedicated Rapid IoT Studio portal at: http://rapid-iot-studio.nxp.com

You will need a free NXP account to access Rapid IoT Studio.

Press LOGIN NXP SSO to log-in with your free NXP account
(section 2.2 Get Started with Rapid IoT Studio to get instructions to register for a free NXP account).
You will be automatically redirected to the NXP Sign In page.
Enter you Email Address and Password and press Sign In.

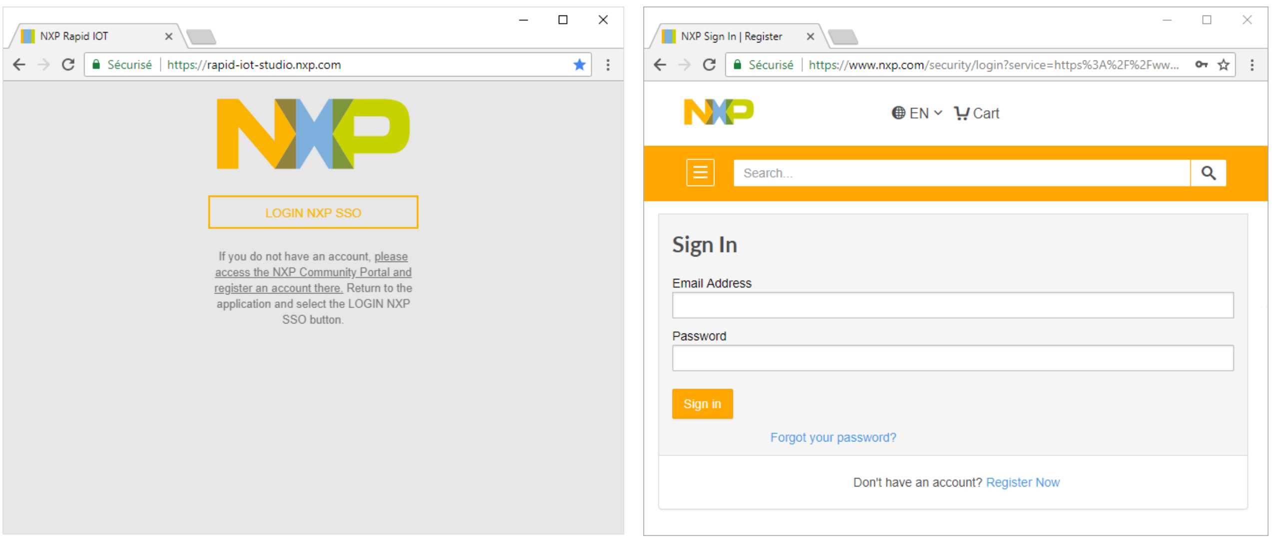


Figure 187. Sign-in/up with Rapid IoT Studio portal

The Dashboard view will open automatically.
Open the left panel by selecting the icon located in the left edge.

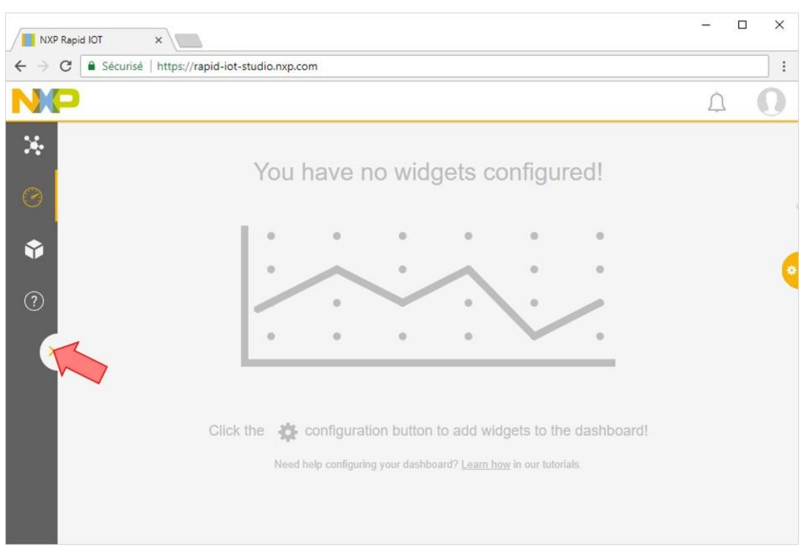


Figure 188. Dashboard view

Select the option Studio to launch the online programming interface.

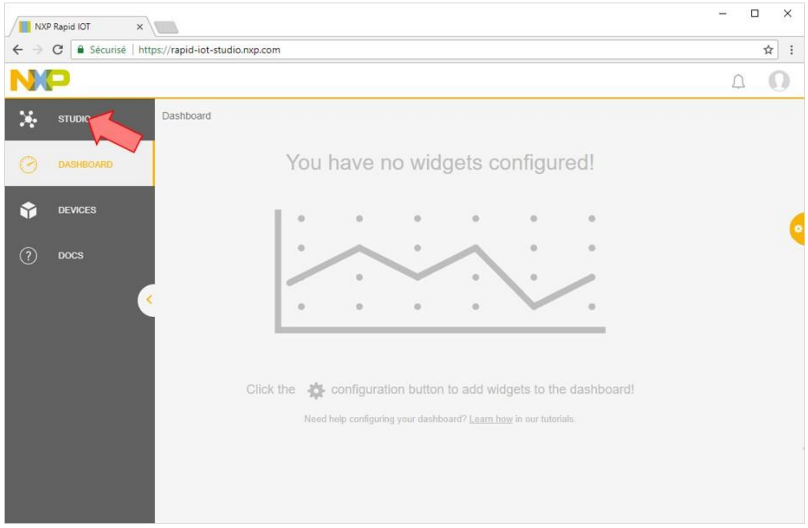


Figure 189. Studio Shortcut

IMPORTANT: Applications build/programmed with Rapid IOT Studio are private applications. User will be able to connect their board over BLE only using the same user-account to login to the phone App.

From the Studio view under CREATE PROJECT select New Project to create a new project.

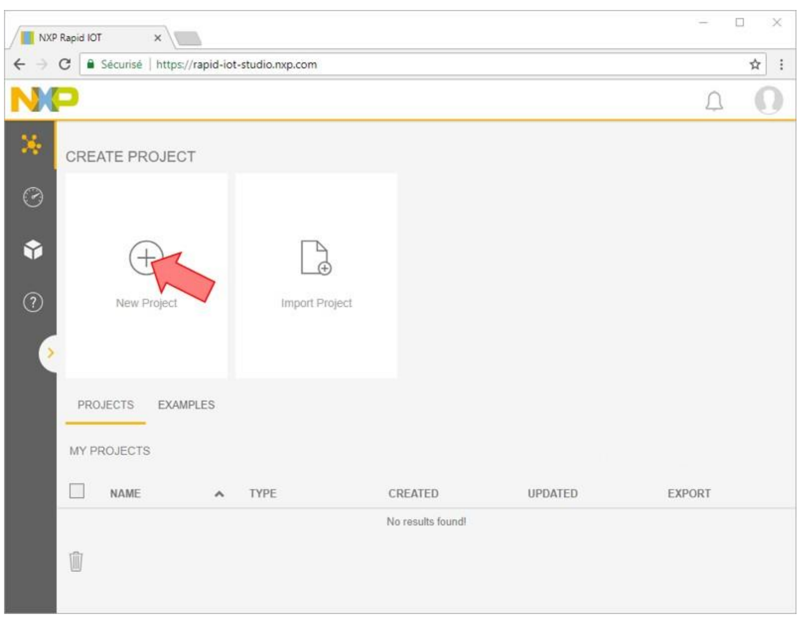


Figure 190. Create New Project with Project manager

From the NEW PROJECT window enter your new project name (ex. BLE Light Sensor) and press the button CONFIRM.

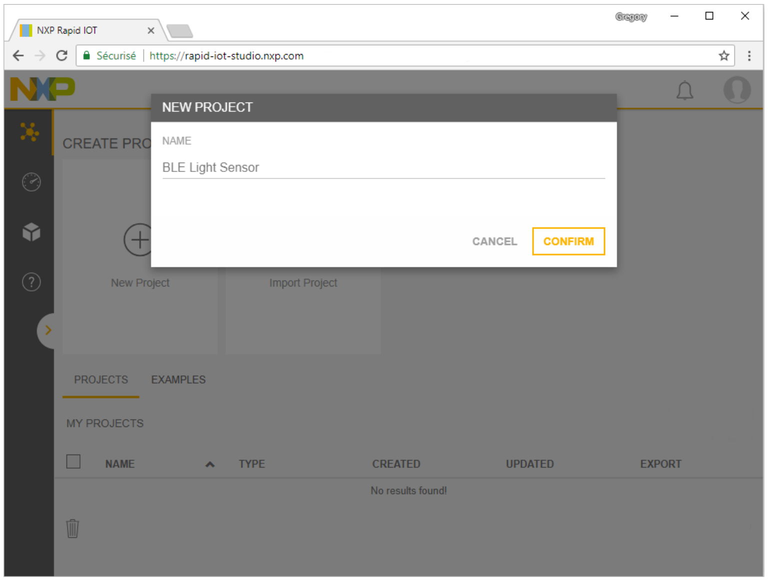


Figure 191. New Project Name

A new blank NXP RAPID IOT workspace will appear.

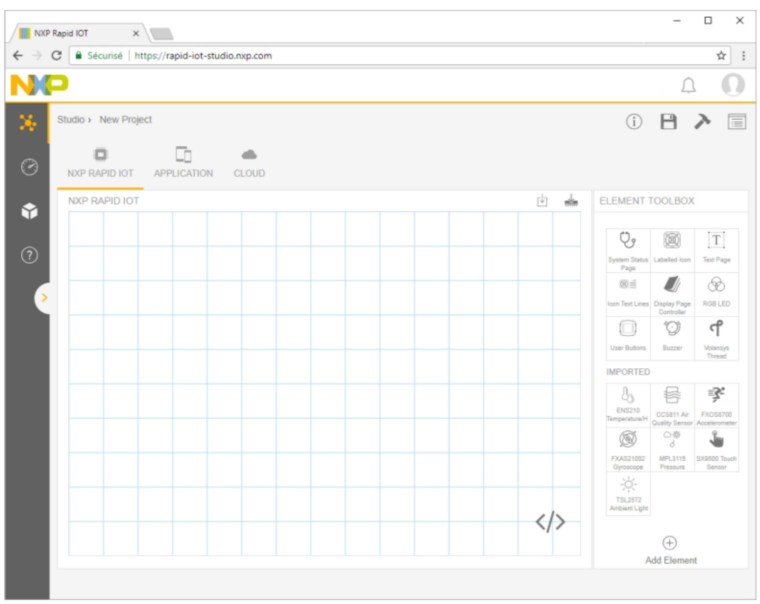


Figure 192. New Workspace

4.2.3. Embedded View

Embedded View is a workspace in Rapid IoT Studio for developing an IoT project's embedded firmware. While in Embedded View, the Element Toolbox displays elements that execute on the project's end device. Embedded elements operate as a bridge between the embedded firmware and device interacting with it.

Embedded View's toolbar includes access to the code editor and the programmer.

Code editor is an advanced text editor for writing and modifying the embedded code of a project.

Click the Code Editor button to display the embedded code editor beside Embedded View's canvas. While the code editor is open, any embedded element selected in the canvas has its corresponding code highlighted.

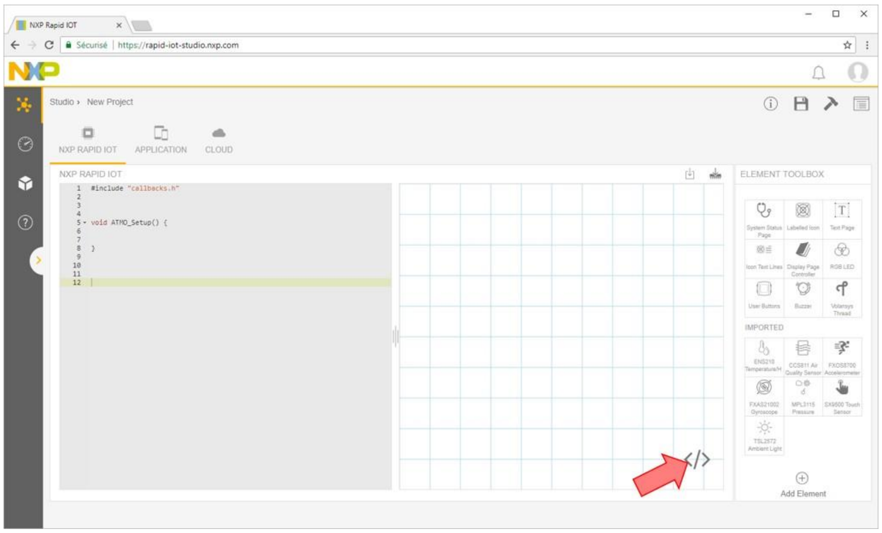


Figure 193. Code Editor Button and View

In addition to being added to the canvas, embedded elements may have their source code customized when added to the canvas and when connected to other elements. While most elements do not require their code to be customized and are designed to operate as-is, the option is available to expand their functionality through custom code. The function element for example is a blank slate element, which relies entirely on customizing code to your liking.

The default headers in the code editor include an area to define variables and custom functions, an interval function, a connected function, a disconnected function, and a setup function.

Function	Description
Interval	Perform actions on a set interval.
Connected	Perform actions when a connection is established to a target device.
Disconnected	Perform actions when a connected device is disconnected.
Setup	Define anything that requires initialization upon being loaded.

Table 34. Function Description

Elements are blocks of code that perform a variety of programmable functions within Studio. Located within the Element Toolbox's elements panel, elements enable the functionality, visualization, and cloud connectivity for an Rapid IoT Studio project.

Elements are added to a view canvas from the Element Toolbox to visually construct your project, and may also be clicked within the toolbox and added directly to the canvas.

For this project, we want to use Bluetooth to send ambient light sensor data to a mobile application. To accomplish this, we only need four elements: a time interval, display text, the light sensor and a BLE GATT characteristic.

To add the elements, select individually the following items in the Element Toolbox located in the right panel:

Figure 194. Add Elements to Project

Selecting an element in the canvas displays its properties in the Element Toolbox's properties panel. Each element has a set of default properties and functionality included within them. Properties vary based on the element, and may be changed based on their needs.

To remove an element, select it in the canvas, then remove it pressing Delete or using the Trash Can button in the element's properties.

When all the Elements have been added to your canvas, your workspace should look like as below.

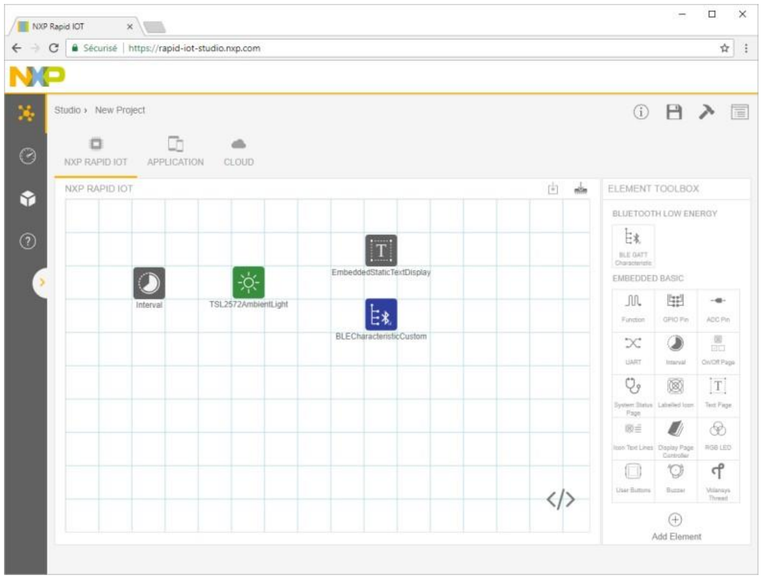


Figure 195. Element added to New Workspace

Elements on the canvas can connect to one another to create a chain of functionality between them by using a "connector." Connectors represent the flow of information from a "source" (beginning) element to a "target" (ending) element.

To create a connector, click and hold the edge of the source element, then drag the cursor to the target element and release it.

Now connect the Elements like represented below.

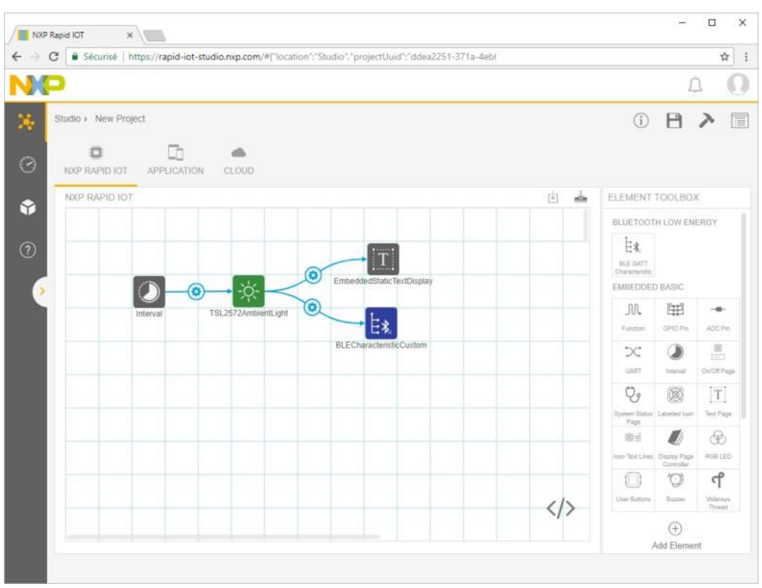


Figure 196. Element added to New Workspace

When two elements become connected, it creates an "event" between them. Events define the actions that occur between the source and target element.

Events can be modified either in the source element's properties, or by selecting the connector's Configuration button between the two elements. Events consist of the following:

- The trigger that causes the event
- The target element
- The target element's ability
- Any ability arguments

When adding the Bluetooth Element to our Workspace, Studio automatically configured the UUIDs and sets some default settings, but we still need to adjust some parameters for our specific need.

From the Workspace, select the Bluetooth Element to display its Properties in the right panel. Disable the options Write and Notify since we will output the Light Sensor data via Bluetooth. Change Data Type from String to Unsigned Integer to display properly the Ambient Light value.

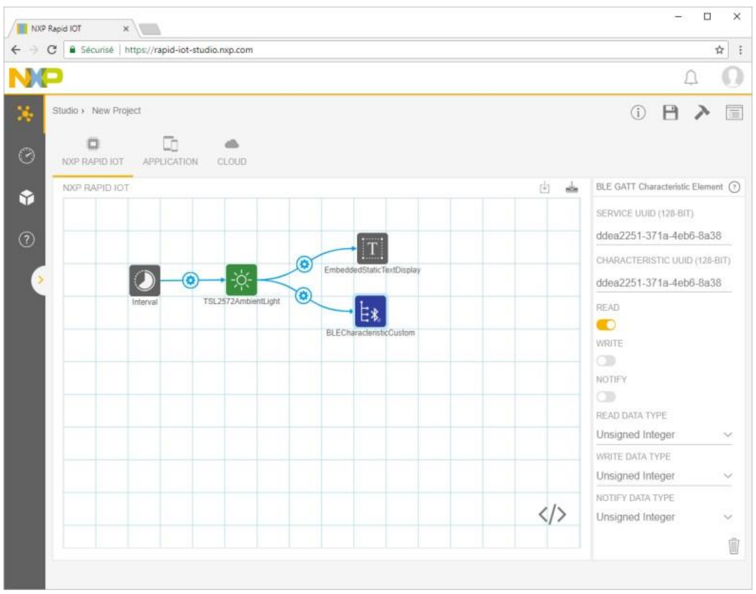


Figure 197. Bluetooth Element Properties

Coupled Elements, like Bluetooth Element, bridge a project's functionality across one view to another.

When a coupled element is added to the canvas, it creates a corresponding copy of it on an associated view (likewise, deleting a coupled element will remove its copy). This allows the element to extend its functionality to that view. Clicking on a coupled element's badge moves you to that view with that element selected and its properties displayed.

4.2.4. Application View

Application View is a workspace in Rapid IoT Studio for developing a phone App interface that interacts with the NXP Rapid IoT hardware. Application View also acts as a gateway between Embedded View and Cloud View, and is used to have data sent from the embedded firmware up to Atmosphere Cloud.

Open the Application view from the upper Project toolbar.

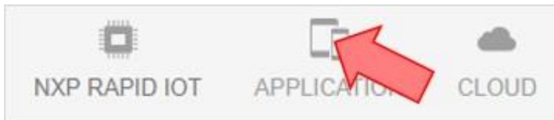


Figure 198. Application View Selector

You'll notice that a Bluetooth Element has been automatically added to the Application Workspace. It is coupled to the Bluetooth Element from the Embedded application.

Now we need to configure the mobile application. First, we need an interval to determine how often to read data from the Rapid IoT. Afterward, we need to display properly the data on the phone screen.

While in Application View, the Element Toolbox displays elements that execute on the application, which includes visual elements that represent interface components and functional elements that run in the background.

Select individually the following items in the Element Toolbox to add them to your Application Workspace.

Figure 199. Application Elements

Select the Element Label in the Workspace.
Change its PROPERTIES NAME to SensorData and remove TEXT information.

Select the Element Label1 in the Workspace.
Change its PROPERTIES NAME to Title and TEXT to Ambient Light (lux).

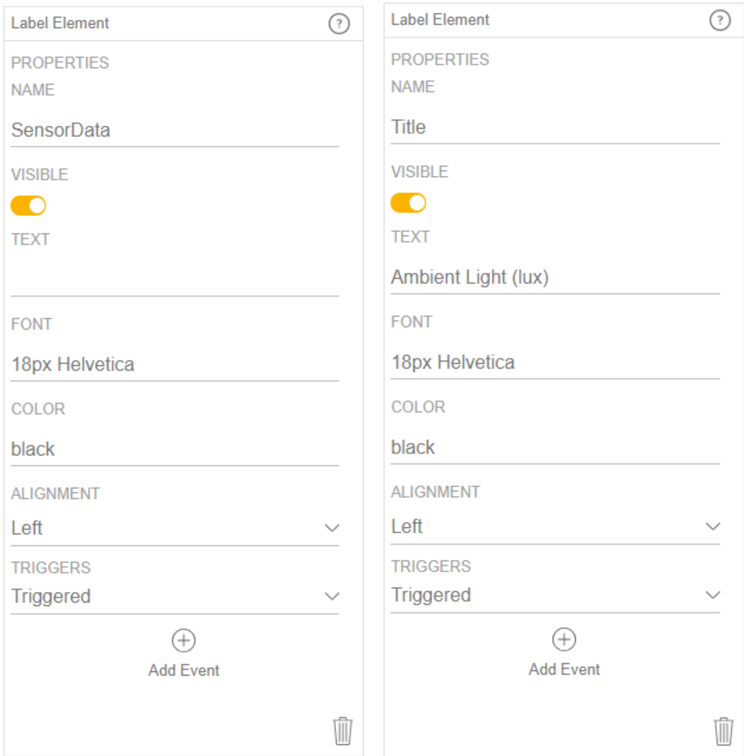


Figure 200. Label Element Properties for Application

The APP VIEW window located in the left side provides an overview of the phone App interface.
Place the Title and the SensorData at the preferred location on the phone screen.

Figure 201. Label Element Positioning in the app View

Now, add connectors as shown below. The title label doesn't need to be connected to anything since it just displays static text.

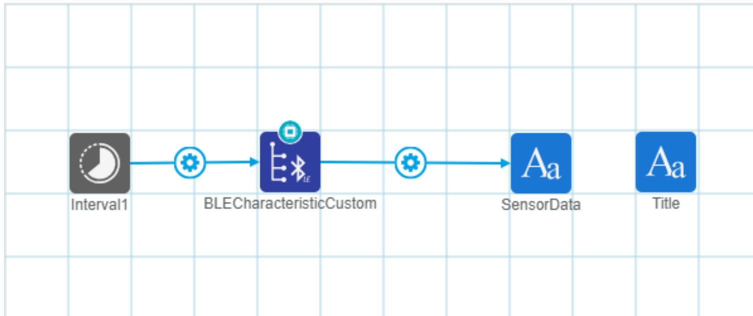


Figure 202. Application Element Connections

There is no additional configuration needed for the Application.

Now, we need to select layouts that our application will support. To do this, select the little phone icon next to the resolution drop-down.

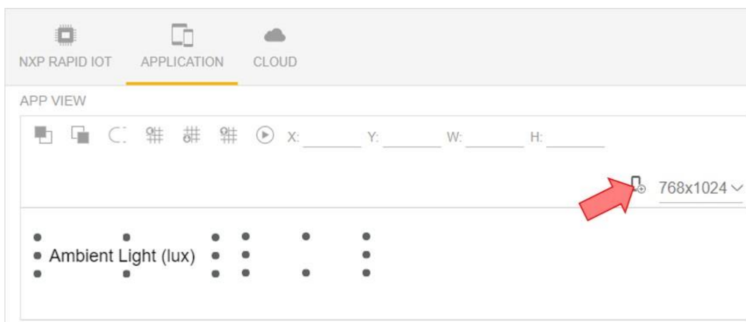


Figure 203. Phone/Tablet Selector

In the window that pops up, select any devices you want to support (your device or the closest match).

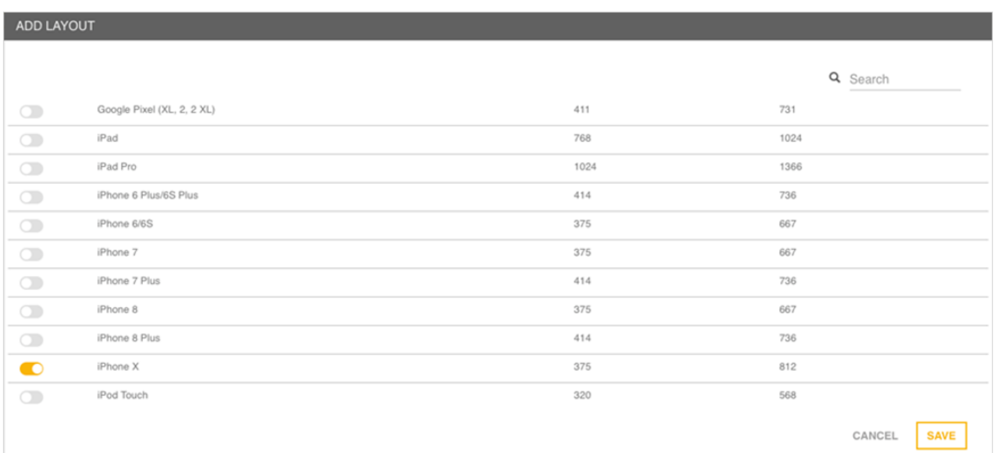


Figure 204. Phone/Tablet Resolution

Before launching any Build or Programming, please make sure to save first your project.
Select the Save button from the toolbar located in the top right corner to save your modifications.

Figure 205. Save Project

In both the NXP Rapid IoT (Embedded) and Application (Mobile) views there is a Compile button that must be selected.

Figure 206. Compile Embedded and Application views

Once both applications are built, click on the Program Firmware button in the NXP Rapid IoT view to download the binary for the Rapid IoT.

Figure 207. Program Firmware

Rapid IoT Studio IDE will then link the project and generate the binary output. The file BLE Light Sensor firmware.bin will be automatically downloaded by your web browser.

SLN-RPK-NODE Application and Wireless MCUs are pre-programmed in factory with a Bootloader to easily update their application through the onboard USB connector. To reprogram K64F internal flash with the new Rapid IoT Blinking an LED application, simply follow those steps:

- Connect one end of the provided USB cable to the computer and the other end to the micro USB type-B connector of the SLN-RPK-NODE.
- Keep SW3 button pressed while pushing shortly SW5/Reset button.
- Wait 1-2s for RGB LED to blink Green then release SW3 button.

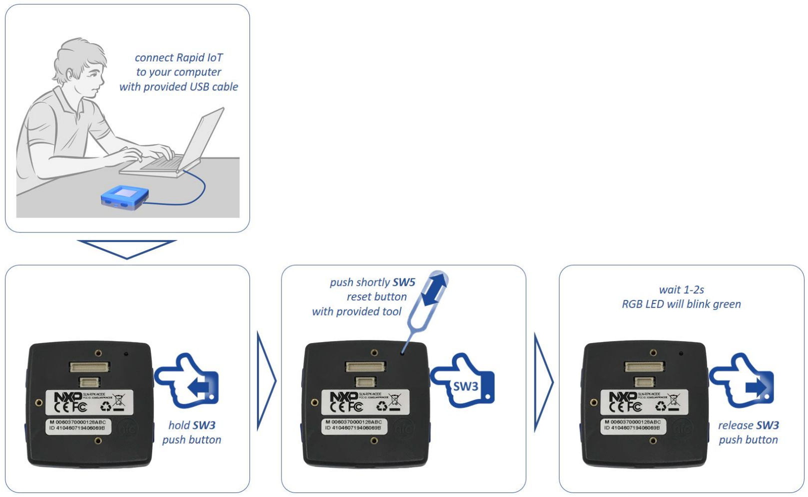


Figure 208. Instructions for USB Mass Storage Device Programming

RGB LED will blink green and your computer will detect a new Mass Storage drive and automatically install the appropriate drivers

- From your computer file explorer, drag-n-drop into the Mass Storage drive the file BLE Light Sensor firmware.bin generated by Rapid IoT Studio available from your download folder.

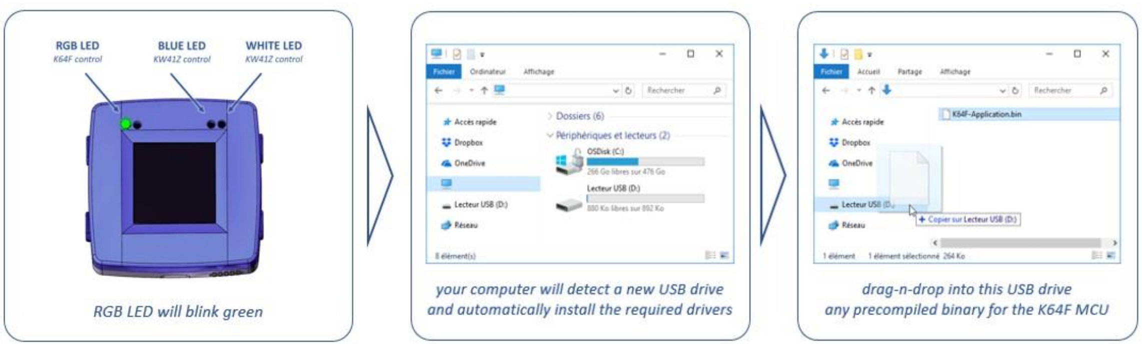


Figure 209. Instructions for pushing a new application through USB

Bootloader will automatically identify the MCU target to reprogram thanks to the binary file signature. RGB LED will blink purple during download and blink blue during serial flash programming.
RGB LED will blink green during K64F internal flash (re)programming with the new application (read from Serial Flash memory) and automatically reset, when ready.

Figure 210. USB Programming LED

RGB LED	BLUE LED	WHITE LED
PURPLE BLINK	OFF	OFF
BLUE BLINK	OFF	OFF
GREEN BLINK	OFF	OFF

Table 35. K64F USB Programming LED Sequence

Launch the Rapid IoT phone App from your phone/tablet.

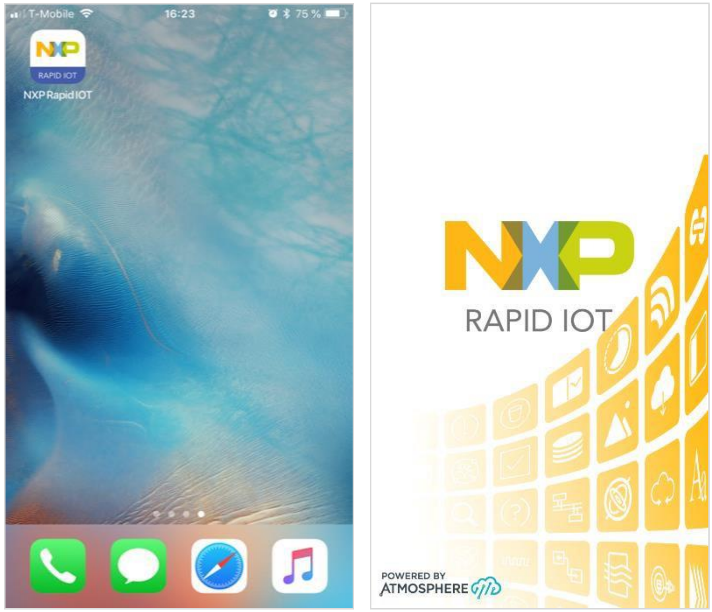


Figure 211. ***Launch Rapid IoT phone App***

Press the button LOGIN NXP SSO.
Enter the Email Address and Password of your free NXP account and press Sign In.

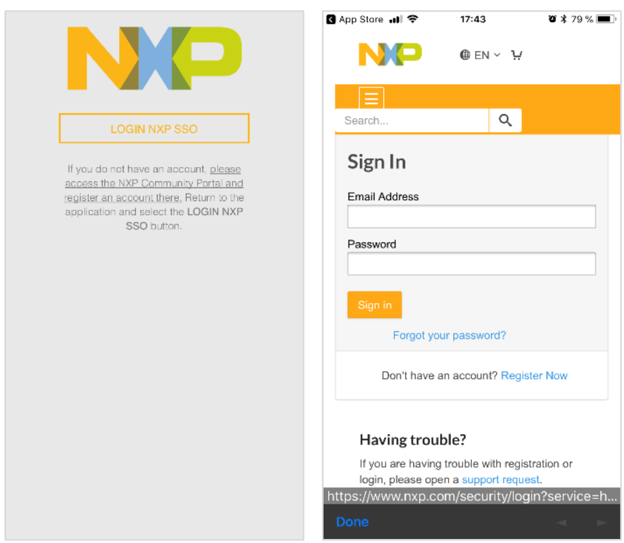


Figure 212. Sign-in/up with Rapid IoT phone App

The phone App will automatically launch the Dashboard view.

Open the left panel by selecting the icon located in the top left corner.
Select the Device view to pair your equipment with your Rapid IoT board.

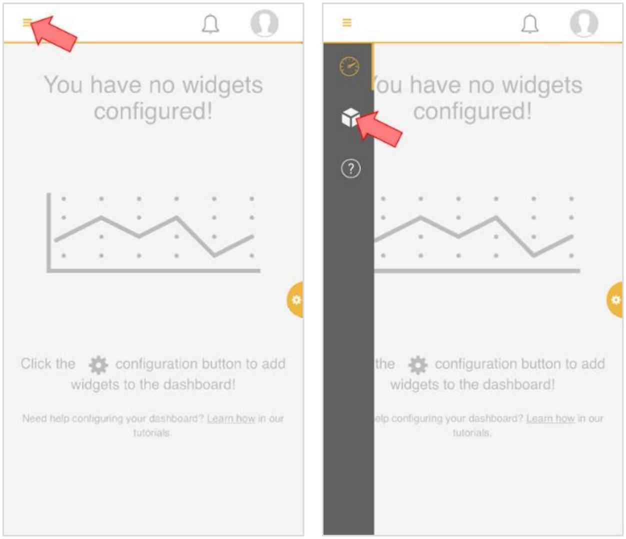


Figure 213. Open the Device View

Select the icon located in the top right corner to scan the Bluetooth devices in the range.

Figure 214. ***Connect New Device***

The phone App indicates the list of the Bluetooth devices detected with the type of Application programmed in their memory as well as the last four Hexadecimal numbers of their MAC address.

Select your Rapid IoT board with TYPE BLE Light Sensor and press PROVISION.

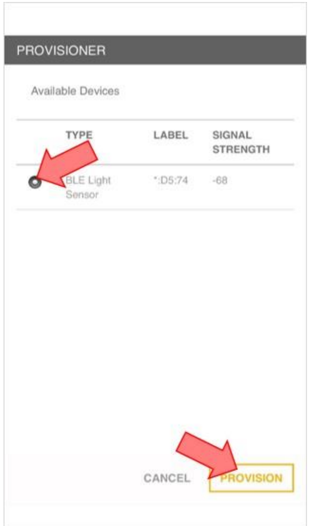


Figure 215. Select and Provision your Rapid IoT board

The phone App will automatically launch the phone App interface for the BLE Light Sensor Demo. It takes approx. 5s for the first sensor value to appear on the phone App.

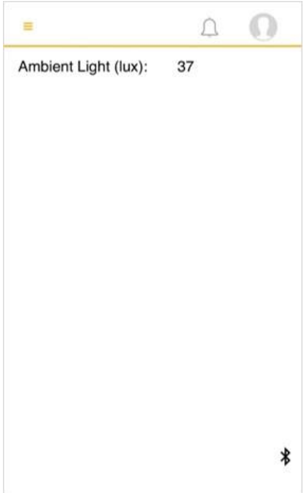


Figure 216. ***Phone App View***

Congratulation you now successfully created, built and programmed your first Rapid IoT Project featuring Embedded and Mobile services!!

4.3. NXP MCUXpresso OFFLINE/DEBUG tool

4.3.1. MCUXpresso SDK Architecture

The Rapid IoT SDK is a composite SDK. It features two SDK target specific SDKs, for each target MCU in the Rapid IoT device.

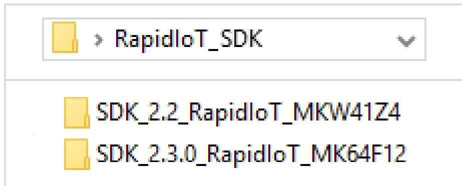


Figure 217. Rapid IoT target specific SDKs

Each target specific SDK features a similar structure to traditional MCUXpresso SDKs. Under the ‘rapid_iot_apps’ directory you will find the factory installed applications on your Rapid IoT device.

Figure 218. KW41Z Target SDK Rapid IoT Apps

Figure 219. K64F Target SDK Rapid IoT Apps

The included K64F application is from the Rapid IoT Studio IDE, repackaged to this SDK format. The application’s code base has been split between ‘app_src’ (shown in Figure 220) and ‘middleware/atmosphere’ (shown in Figure 221). This provides a clean directory structure for users to copy and modify as needed.

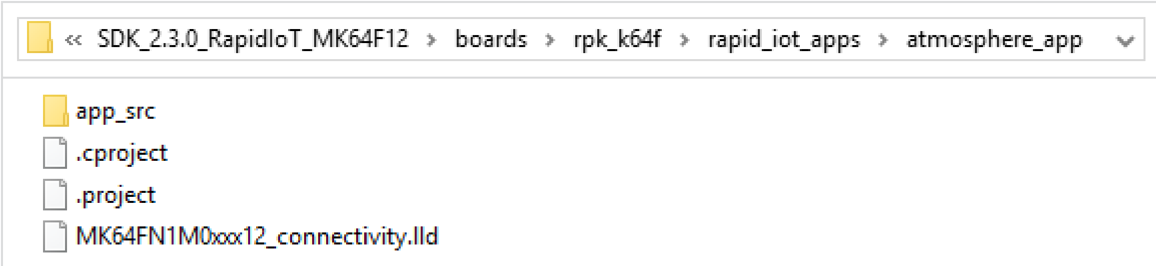


Figure 220. Rapid IoT Out of Box Application

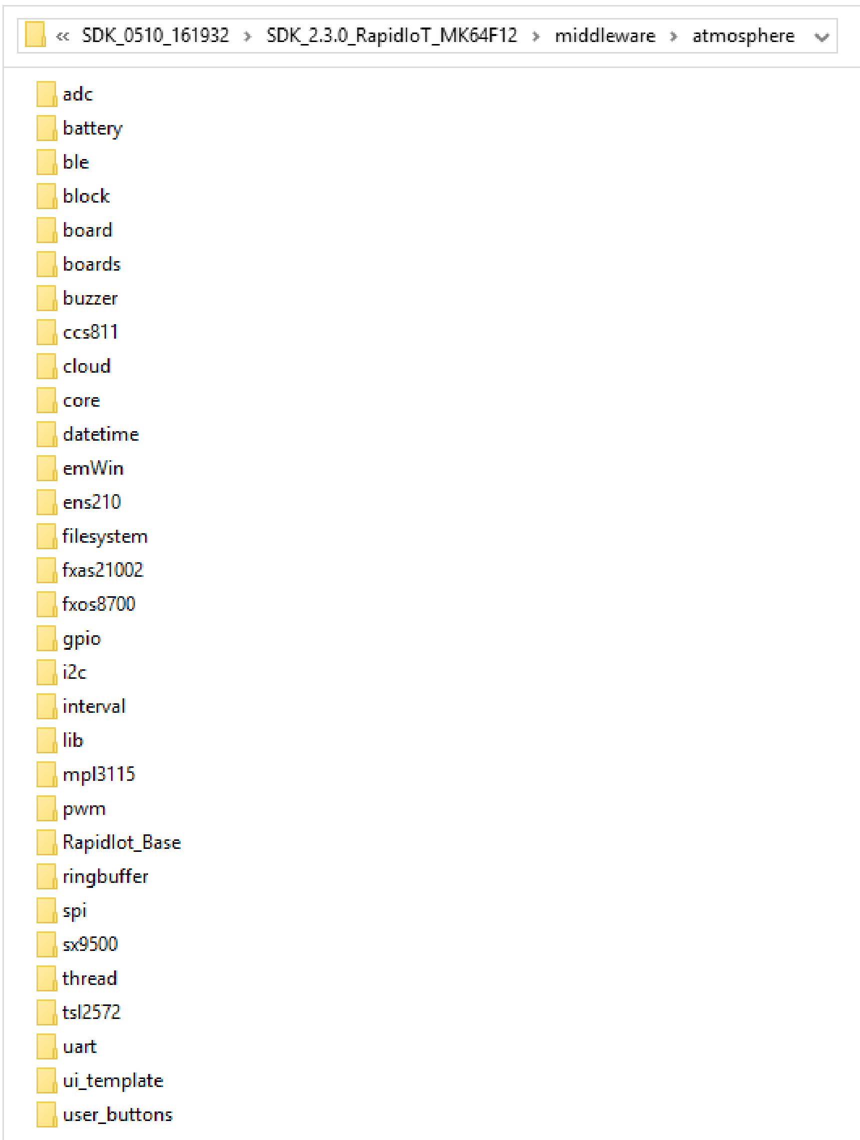


Figure 221. Rapid IoT Studio middleware

The Atmosphere source is divided further into two groups. The first is shown above in Figure 221. The second is a static library that is illustrated in Figure 222. This static library features lower level code for application updating, connectivity, RTOS, etc...

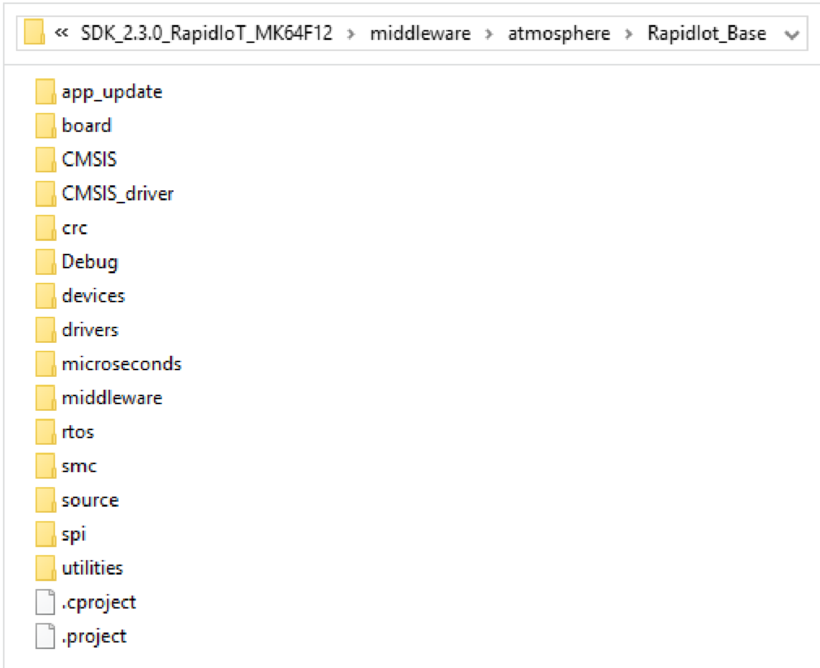


Figure 222. Atmosphere base library

For non-Atmosphere based projects (kboot(s) & rpk_kw41z_hybrid), there is a rapid_iot specific middleware directory. Figure 223 illustrates the structure of this middleware folder.

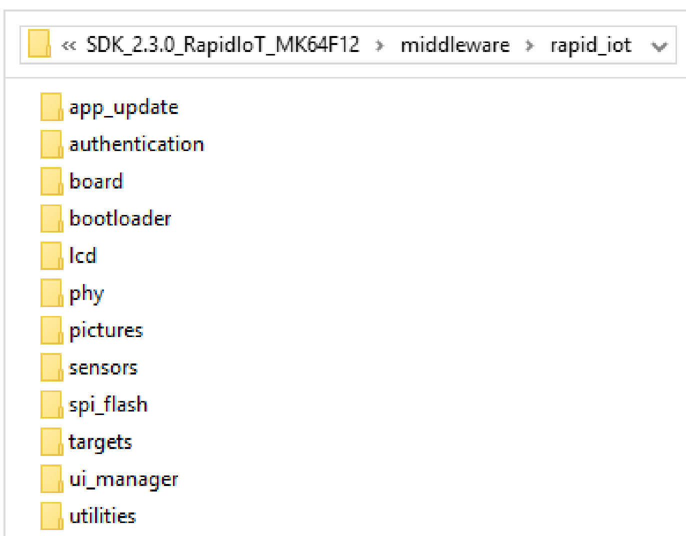


Figure 223. Rapid IoT non-Atmosphere middleware

4.3.2. Create a new Project

We will now create a new MCUXpresso project for Rapid IoT using the Hello project as a default template to get all the components embedded in the SLN-RPK-NODE board properly initialized.

If you already followed the instruction to Get started with MCUXpresso, you can skip the following steps to download and import Rapid IoT SDKs into MCUXpresso IDE.

4.3.2.1. Download Rapid IoT SDKs

Download first the Rapid IoT SDK package online at: www.nxp.com/rapid-iot

Select from the PRODUCTS tab Rapid IoT prototyping kit.

Go to SOFTWARE & TOOLS tab and select the latest version of the Rapid IoT SDK package.

You will be asked to sign-in/up with a free NXP user-account.

When the download of the compressed file completes, extract it in your MCUXpresso Workspace on your computer HDD.

4.3.2.2. Import Rapid IoT SDKs

The Rapid IoT SDK is a composite SDK. It features two SDK target specific SDKs, for each target MCU in the Rapid IoT device.

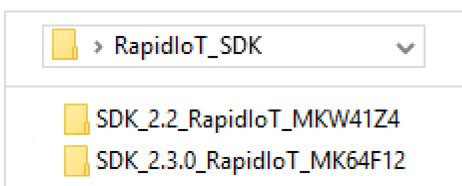


Figure 224. Rapid IoT target specific SDKs

Before building the Rapid IoT SDK example projects, the target SDKs need to be imported into MCUXpresso IDE by drag and dropping each target SDK into the “Installed SDKs” window from MCUXpresso IDE.

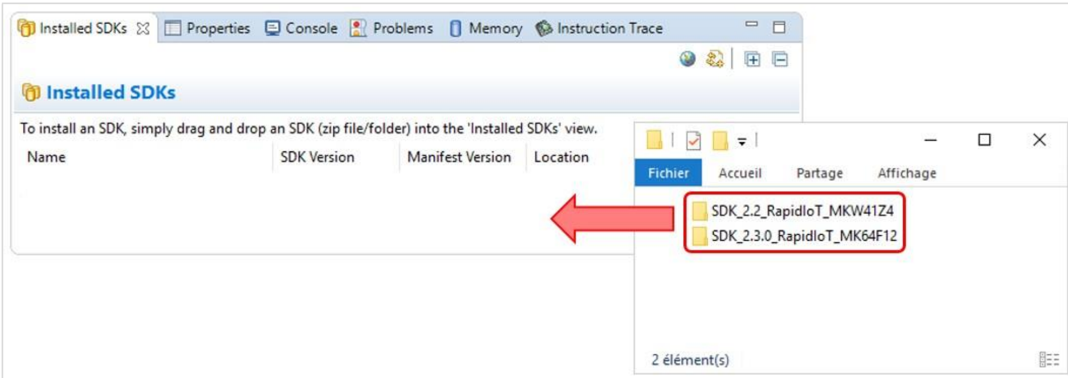


Figure 225. Rapid IoT target SDK install for MCUXpresso IDE

For each package a confirmation window will pop-up, select OK to validate.

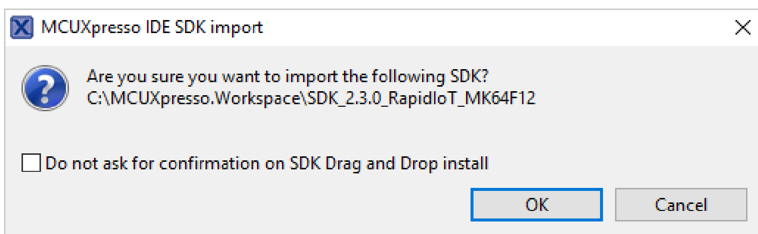


Figure 226. MCUXpresso SDK import confirmation window

4.3.2.3. Import the Hello World project for Rapid IoT

The Hello World project for Rapid IoT can be automatically imported and cloned into MCUXpresso IDE like a classical SDK Example.

To import the Hello World project for Rapid IoT projects into MCUXpresso IDE, follow those steps:

From the Quickstart Panel, select the option Import SDK Example(s)…

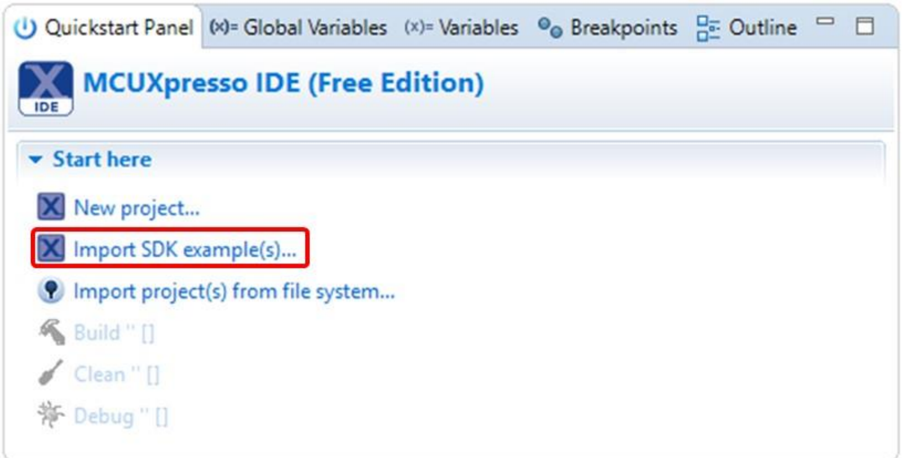


Figure 227. MCUXpresso Quickstart Project Import SDK Example(s)

From the Board and/or Device selection page select the rapid iot k64f then press Next.

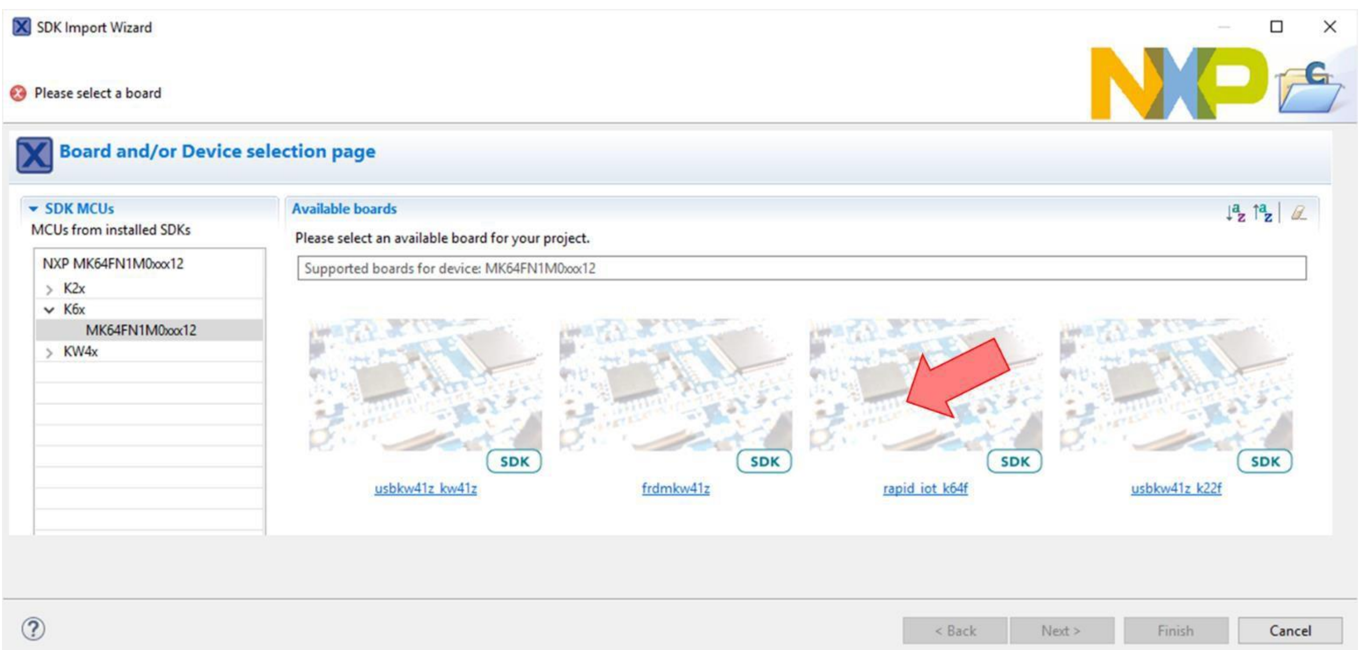


Figure 228. Board and/or Device selection

From the project selection page expand the rapid_iot_apps and the hello_world fields then select the hello_world example and press Next.

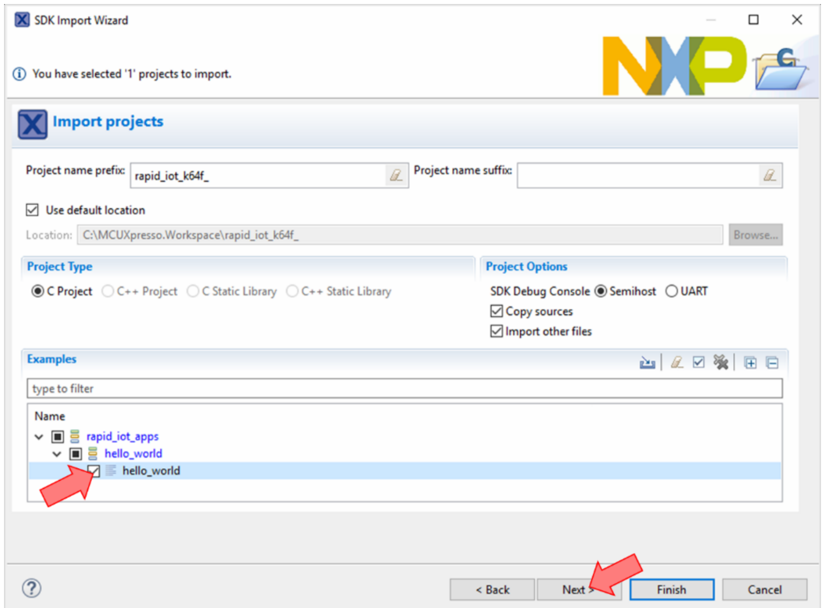


Figure 229. Project Example selection

From the project Advanced Settings page, uncheck the two options Redirect SDK "PRINTF" to C library "printf" and Include semihost Hardfault handler and check the option Redlib: Use floating point version of printf, then press Finish.

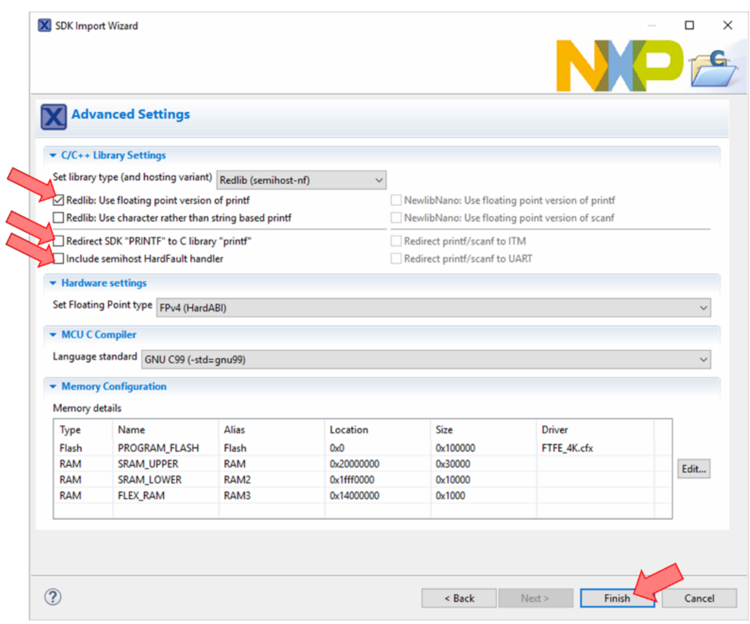


Figure 230. Project advanced Settings

The project Hello World for Rapid IoT is now accessible from your Project Workspace with every files required to create any kind of Rapid IoT application.

From the source folder, double-click on the file hello_world.c

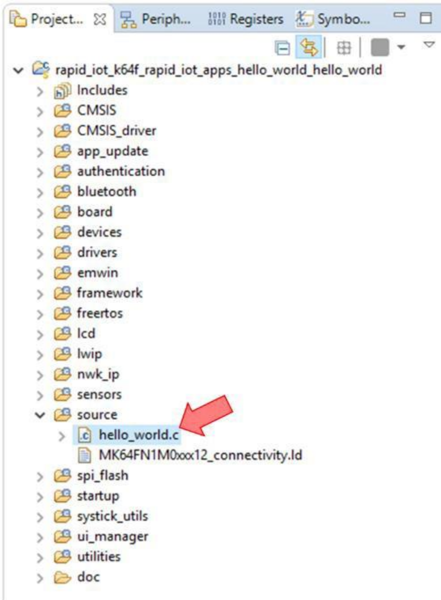


Figure 231. Hello World project for Rapid IoT overview

4.3.2.4. Create a new project for Rapid IoT

The Hello World project is a default template to start any new Rapid IoT application.

Let's modify the hello_world.c code to control the RGB LED.

The RGB LED, like every Rapid IoT components (including Application/K64F MCU peripherals), has been already configured and initialized.

Right after the intialization of the LED, under the code below:

```
/* Init Led module */
LED_Init();
```

Add the following code to turn the LED blue, then white, then red during 1s each and turn it off.

```
RGB_Led_Set_State(RGB_LED_BRIGHT_HIGH, RGB_LED_COLOR_BLUE);
App_WaitMsec(1000);
RGB_Led_Set_State(RGB_LED_BRIGHT_HIGH, RGB_LED_COLOR_WHITE);
App_WaitMsec(1000);
RGB_Led_Set_State(RGB_LED_BRIGHT_HIGH, RGB_LED_COLOR_RED);
App_WaitMsec(1000);
RGB_Led_Set_State(RGB_LED_BRIGHT_OFF, RGB_LED_COLOR_WHITE);
```

Now let's modify the hello_world.c code to display a simple message on the Rapid IoT screen.

The display has been already configured and initialized in the Hello World project.

Right after the intialization of the LED, under the code below:

```
/* Turn on LCD display */
Init_Display();
```

Add the following code to display Rapid IoT Hello World in blue on white background.

```
/* Display a Message */
GUI_SetFont(&GUI_Font8x18);
GUI_SetBkColor(GUI_WHITE);
GUI_Clear();
GUI_SetColor(GUI_BLUE);
GUI_DispString("\n     Rapid IoT\n\n");
GUI_DispString("     Hello World!\n\n");
```

Don't forget to save your modification by pressing Save All from the toolbar or Ctrl + Shift + S.

4.3.2.5. Build the Hello World project for Rapid IoT

From the Quickstart Panel, select Build ‘rapid_iot_k64f_rapid_iot_apps_hello_world’ [Debug].

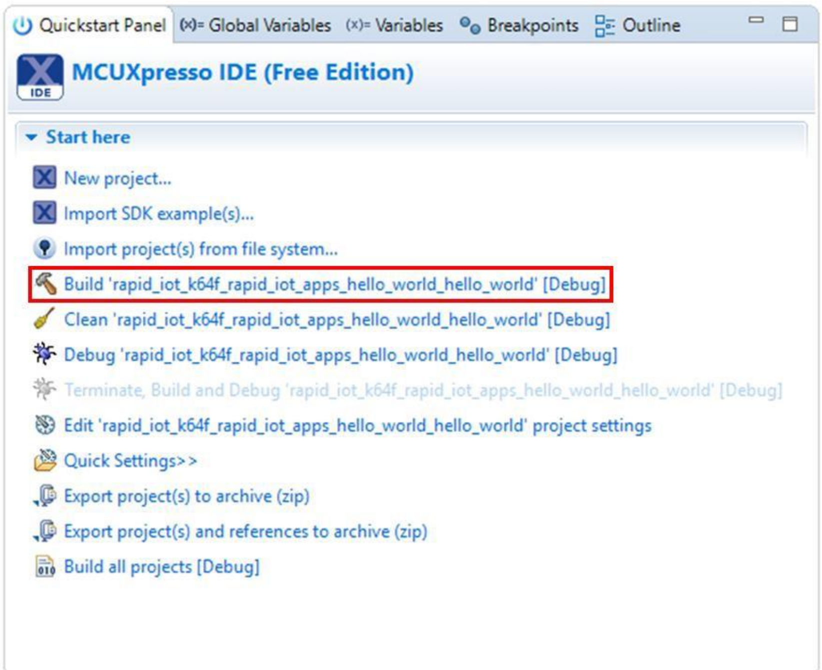


Figure 232. Project Build

Wait for the Console window to confirm that the compilation ended up successfully.

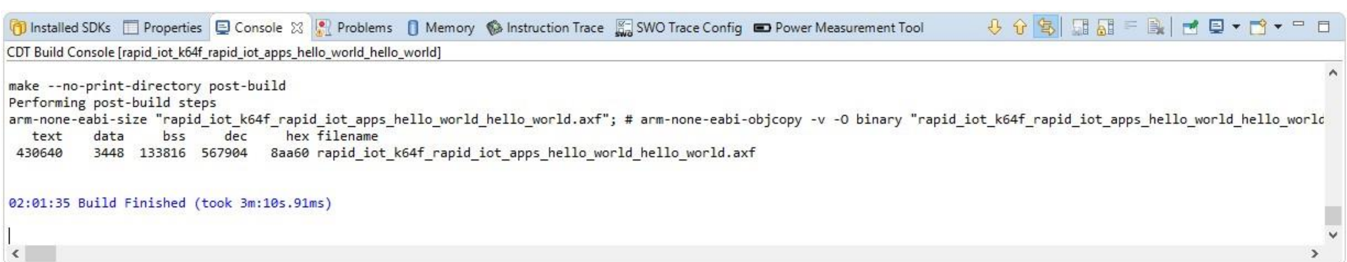


Figure 233. Console Window

4.3.2.6. Debug a Rapid IoT Project

Before launching the debug session make sure that your Docking station has been properly updated.

If you already followed the instruction to Get started with MCUXpresso, you can skip the following step to update the Docking station firmware.

IMPORTANT: The Docking station shipped by MikroElektronika is preprogrammed in factory with an OpenSDA DAP-LINK application dedicated for the MikroElektronika Hexiwear platform. User should reprogram (once) the OpenSDA application of their Docking station before programming or debugging their Rapid IoT board.

To download for Docking station DAP-LINK for Rapid IoT go online at: www.nxp.com/rapid-iot
Select from the PRODUCTS tab Rapid IoT prototyping kit.
Go to SOFTWARE & TOOLS tab and select Docking Station DAP-LINK for SLN-RPK-NODE.
You will be asked to sign-in/up with a free NXP user-account.
When the download of the file completed, copy the Docking Station DAP-LINK firmware in your MCUXpresso Workspace on your computer HDD.

To reprogram the OpenSDA Application from the MikroElektronika Docking-station follow those steps (Rapid IoT board should not be mounted to the Docking station):

- Set the Power switch of the Docking station (red below) to OFF.
- Set the jumper switches of the Docking station to 11001000.
- Connect your Docking station to the USB port of your computer.
- Keep MK64 RESET button of the Docking station pressed, while setting the Power switch of the Docking station to ON, wait 2 seconds then release the MK64 RESET button.

Your computer will detect a new Mass Storage drive named MAINTENANCE and automatically install the appropriate drivers.

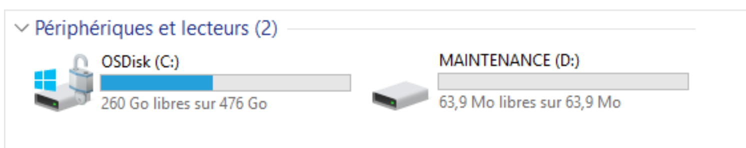


Figure 64. MAINTENANCE MSD drive

- From your computer file explorer, drag-n-drop or copy-paste into the MAINTENANCE drive the binary file k20dx_rapid_iot_if_crc_legacy_0x8000.bin.

Wait for the download to complete.

- Set the Power switch of the Docking station (red below) to OFF.
- Set the Power switch of the Docking station (red below) to ON.

Your computer should detect new Mass Storage drive named IOT-DAPLINK and automatically install the appropriate drivers.

Figure 65. IOT-DAPLINK MSD drive

Congratulation your MikroElektronika Docking-station is now ready to program, serial-monitor and debug safely both MCU targets of your Rapid IoT board!!

IMPORTANT: Before connecting Rapid IoT to the MikroElektronika Docking-station or changing the Jumper Switches make sure to set the Power switch of the Docking station (red below) to OFF.

Plug your Rapid IoT to the Docking station aligning carefully the board to board connectors (yellow below).

Figure 108. Docking station connector

The OpenSDA circuitry embedded in the Mikroelektronika Hexiwear Docking-station (MIKROE-2094) can program, serial-monitor and debug alternately the K64F and the KW41Z target MCUs featured in the Rapid IoT.

Set the jumper switches of the Docking station to 11001000 to program or debug the Application/K64F controller.

Figure 109. Docking Station Jumper Switch configuration to debug K64F

Set the jumper switches of the Docking station to 00111000 to program or debug the Wireless/KW41Z controller.

Figure 110. Docking Station Jumper Switch configuration to debug KW41Z

From the Quickstart Panel, select Debug 'rapid_iot_k64f_rapid_iot_apps_hello_world' [Debug].

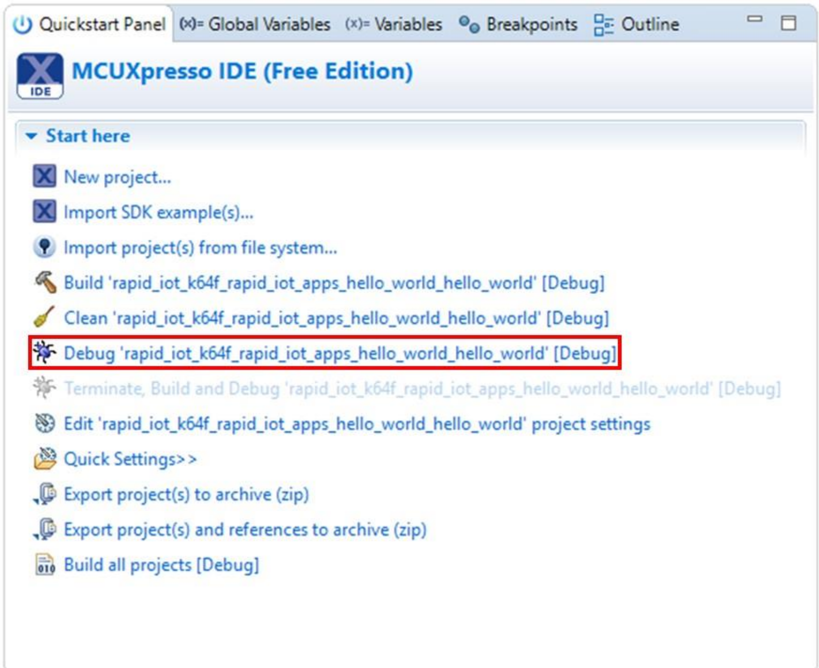


Figure 239. Project Debug

A configuration window should popup to ask you to select your debug Probe.

If your Docking station has been properly programmed with the DAP-LINK application for the Rapid IoT (see Docking station programming in the Hardware section), a CMSIS-DAP probe should be detected.

Otherwise check the box MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes and press Search again.

When your probe is properly detected, simply select it from the list of the Available attached probes, check the box Remember my selection (for this Launch configuration) to avoid the same window to popup at the next debug and press OK.

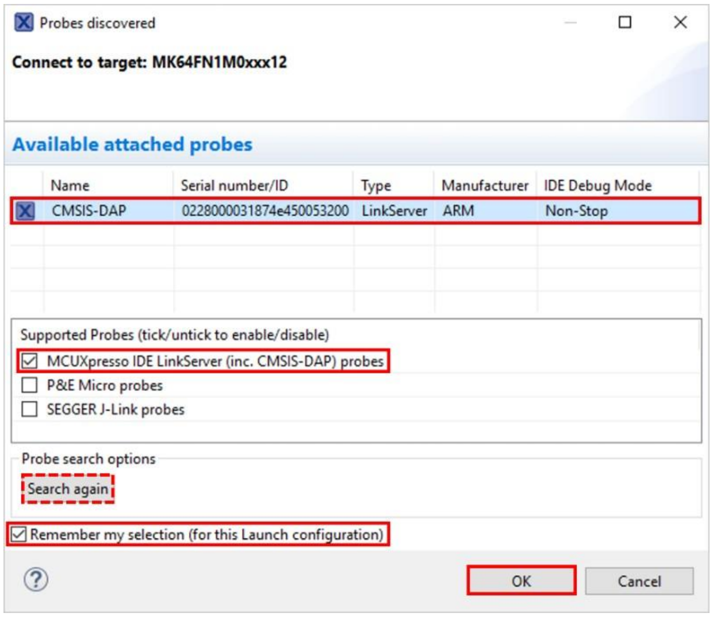


Figure 240. Probe Selection Window

Wait for MCUXpresso IDE to switch to the Debug view and press the Resume button from the toolbar.

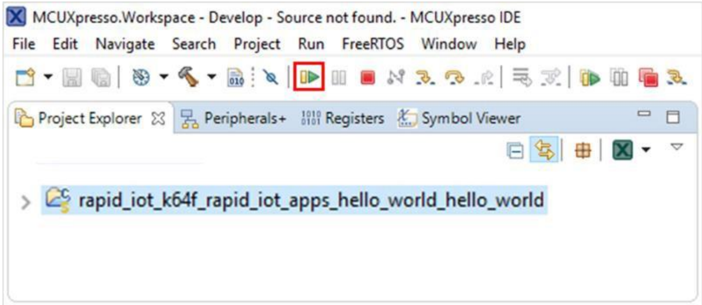


Figure 241. Debug View

Congratulation, you successfully debugged your Rapid IoT with MCUXpresso IDE!!

Figure 242. Rapid IoT Hello World

How to Reach Us:

Home Page:
www.nxp.com

Web Support:
www.nxp.com/support

Document Number: SLN-RPK-NODE-UG
Rev. 0
08/2018