

RAPID IOT PROTOTYPING KIT

JULY 2018



SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

AGENDA

- Product Introduction
- Rapid IoT Studio
- MCUXpresso IDE Integration
- Out-of-box Application
- Create Applications in Atmosphere Studio
- Summary





01.

Product Introduction

Rapid IoT prototyping kit

From IoT idea to proof-of-concept, as easy as 1-2-3

ACCELERATING IOT DESIGN

Web IDE with GUI-based programming
EXTREME EASE OF USE

MCUXpresso project and source code generation
PROTOTYPING TO DEVELOPMENT

SECURE, FROM SENSOR TO CLOUD

Secure Element, Secure Boot, Crypto-accelerators
UNCOMPROMISED SECURITY

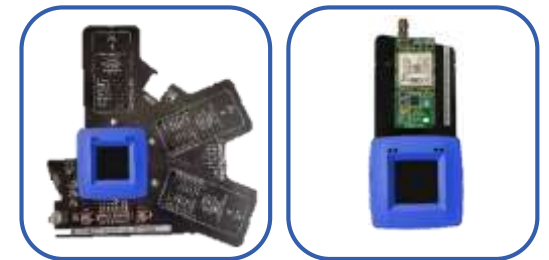
iOS/Android mobile Apps, IoT Cloud, web visualization
SENSOR TO CLOUD

COMPREHENSIVE DESIGN & ENABLEMENT

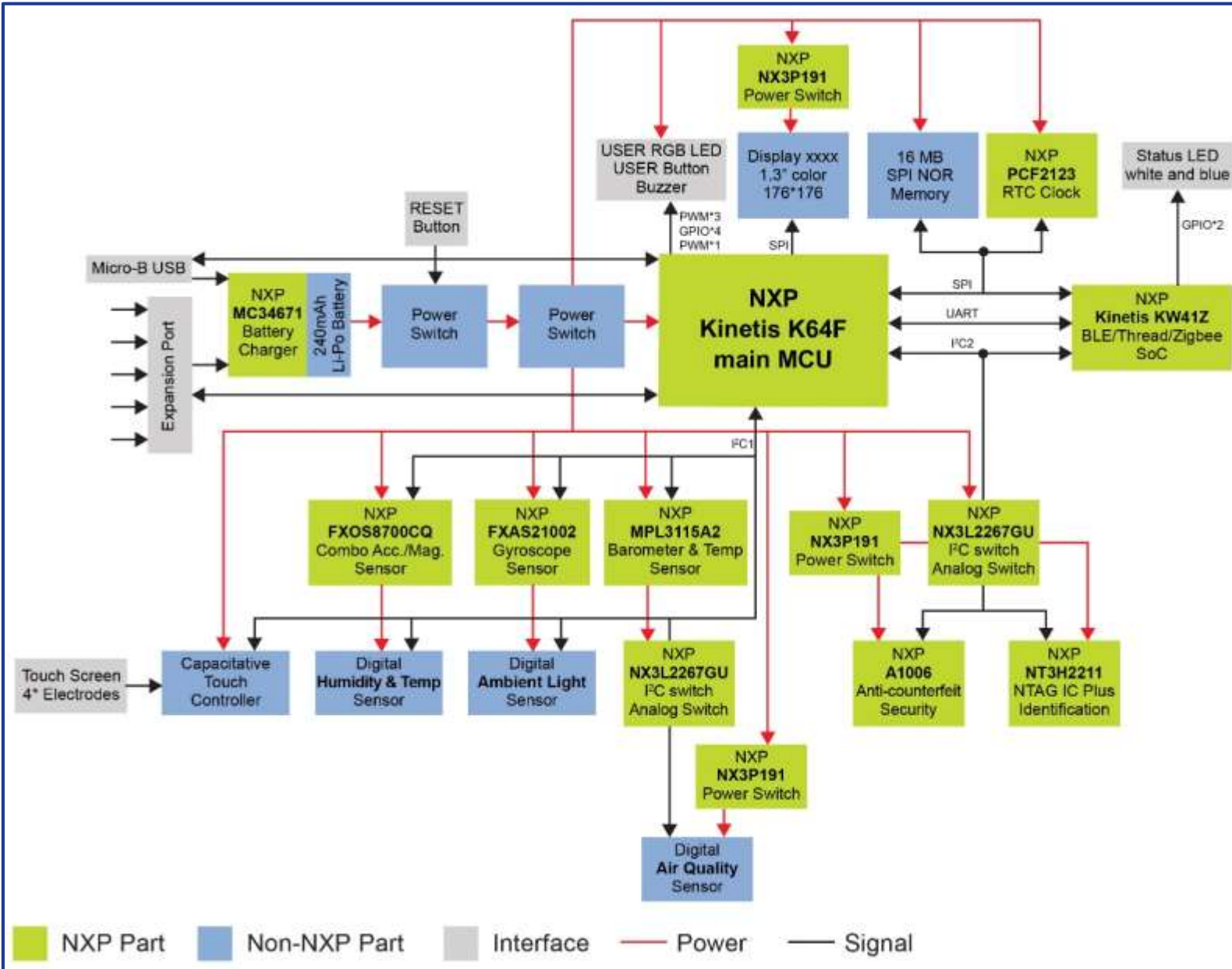
20+ active components
PROCESSING, CONNECTIVITY, SECURITY, SENSING, ANALOG

RTOS, drivers, middleware, example applications
SOFTWARE ENABLEMENT

400+ Click boards™
EXPANDABLE TO MOST IOT END NODE USE CASES



Rapid IoT – Block diagram



Connectivity / IoT cloud options

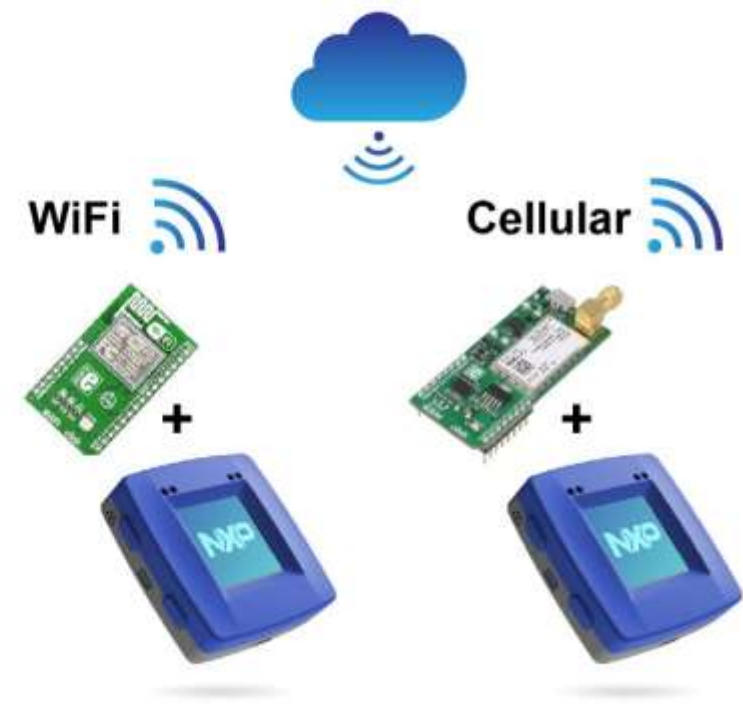
BLE to Android and iOS Device



Thread to NXP's Modular IoT Gateway



Upcoming Options



Target Applications:

Low-power, security, sensing, processing, cloud-connected

Home & Building Automation



Smart Cities



Smart Wearables & Fitness



Industrial



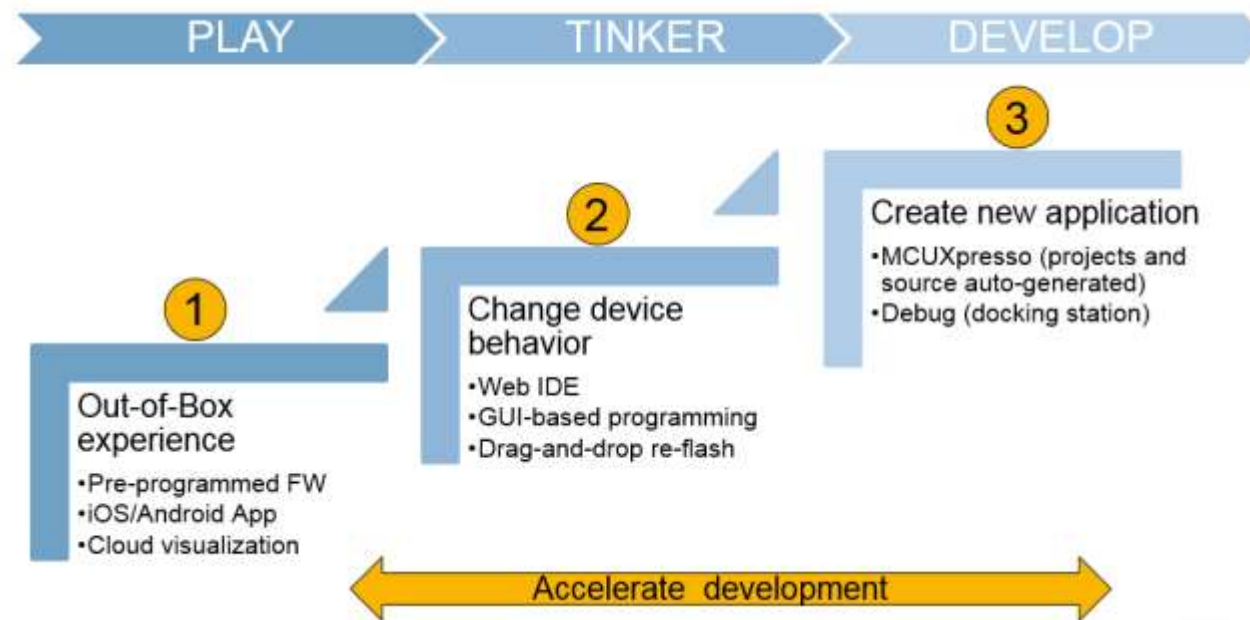
Smart Health



Use cases yet to be discovered



The IoT developer's experience





02.

Rapid IoT Studio

Rapid IoT Studio

A Radically Simple Way to Create IoT Solutions

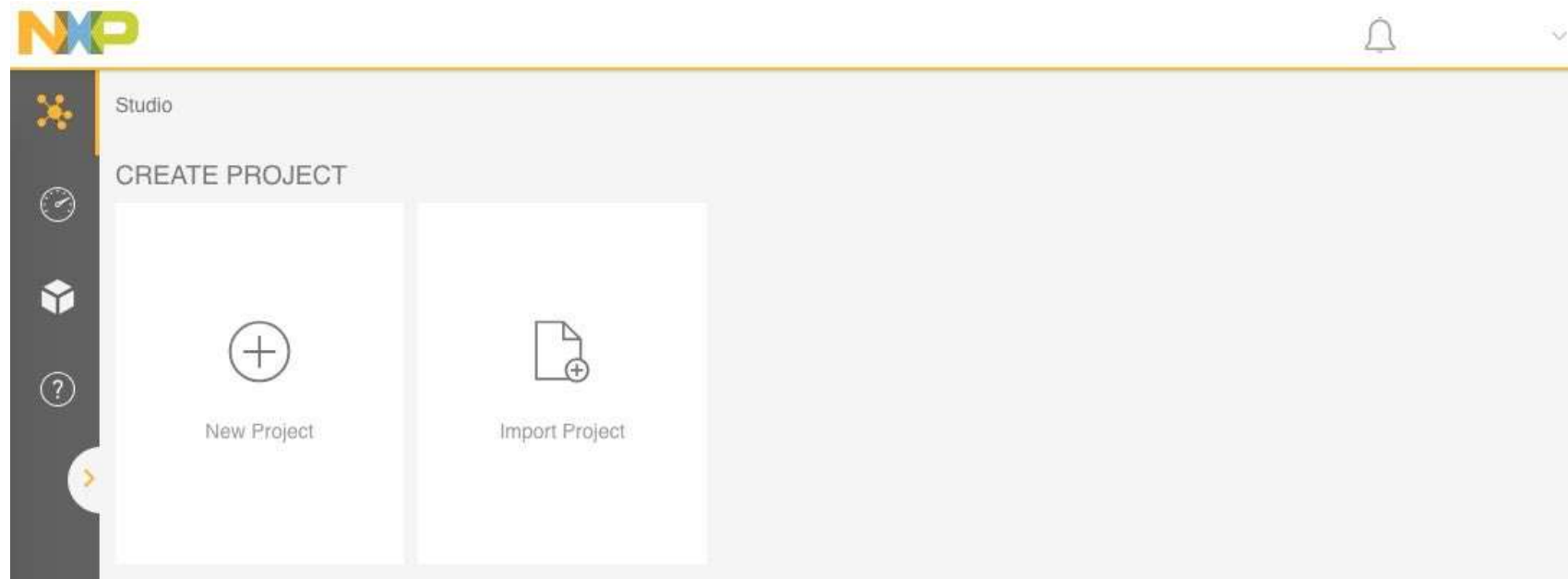


- **A Web-based Integrated Development Environment (IDE)**
 - Visual, drag-and-drop user interface
- **Simultaneous creation of embedded, mobile app and web app code**
- **Robust Library with numerous hardware and software IoT building blocks**
 - Masks complexity of multiple SDKs, APIs, Drivers
 - Using Elements and Connectors
- **Reduces need for expensive, specialized programming skills**
 - Ideal for Small-to-Mid Sized IoT Solution Developers

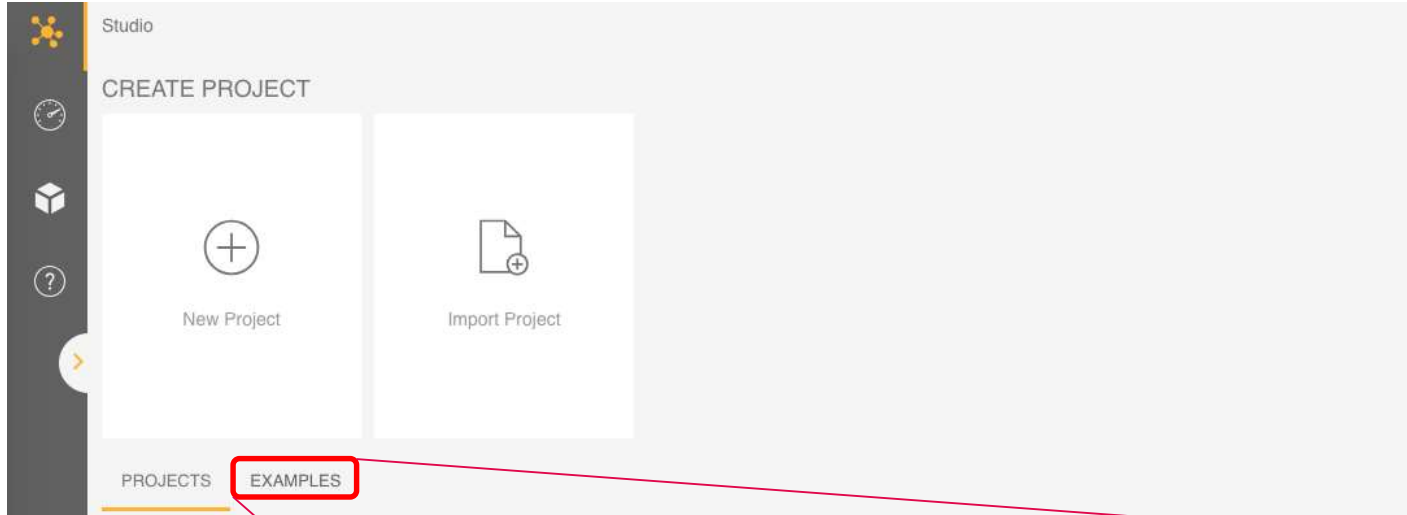


Rapid IoT Studio IDE

- Requires a personal account = authentication
- The landing page shows the project manager
 - create a new project,
 - open an existing project,
 - or import an external project.



Adding Example Projects



EXAMPLE PROJECTS				Search	10	
NAME		TYPE	DESCRIPTION	DOWNLOAD		
Rapid IoT Weather Station		NXP Rapid IoT	New Project			
Rapid IoT Kit Out Of Box Demo		NXP Rapid IoT	New Project			
Rapid IoT Blinking an LED		NXP Rapid IoT	New Project			
Control LED with app over BLE		NXP Rapid IoT	New Project			

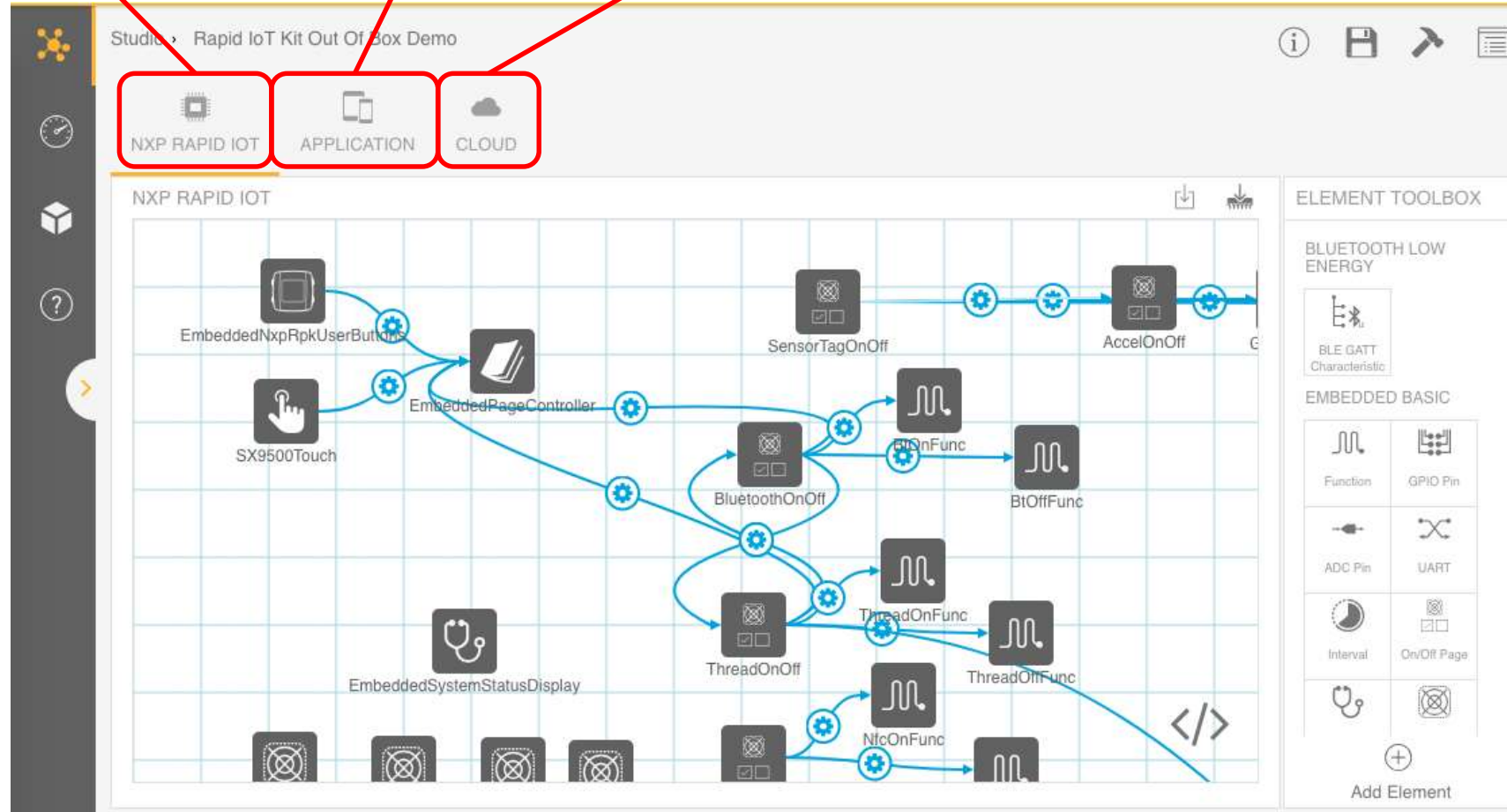
Studio IDE Views

Embedded App

Mobile App

Cloud App

1 current project
=
3 views



Embedded View

Canvas (work area)

Output Controls

Project Controls

The screenshot displays the Embedded View interface for a project titled "Studio > Rapid IoT Kit Out Of Box Demo". The interface is divided into several key areas:

- Project Controls:** Located at the top right, containing icons for information, save, undo, and redo.
- Element Toolbox:** A vertical panel on the right side, categorized into "BLUETOOTH LOW ENERGY" (with "BLE GATT Characteristic") and "EMBEDDED BASIC" (with "Function", "GPIO Pin", "ADC Pin", "UART", "Interval", "On/Off Page", and "Add Element").
- Canvas (work area):** The central workspace showing a flowchart with components like "EmbeddedNxpRpkUserButton", "SX9500Touch", "EmbeddedPageController", "BluetoothOnOff", "ThreadOnOff", "ThreadOnFunc", "ThreadOffFunc", "NfcOnFunc", "SensorTagOnOff", "AccelOnOff", "BtOffFunc", and "EmbeddedSystemStatusDisplay".
- Output Controls:** A small panel at the top right of the canvas containing download and upload icons.
- Source View Toggle:** A red-bordered icon with the code symbols "</>" located at the bottom right of the canvas area.
- Navigation:** A vertical sidebar on the left with icons for home, search, and navigation.
- Top Bar:** Contains the project name and three tabs: "NXP RAPID IOT" (highlighted with a red box), "APPLICATION", and "CLOUD".

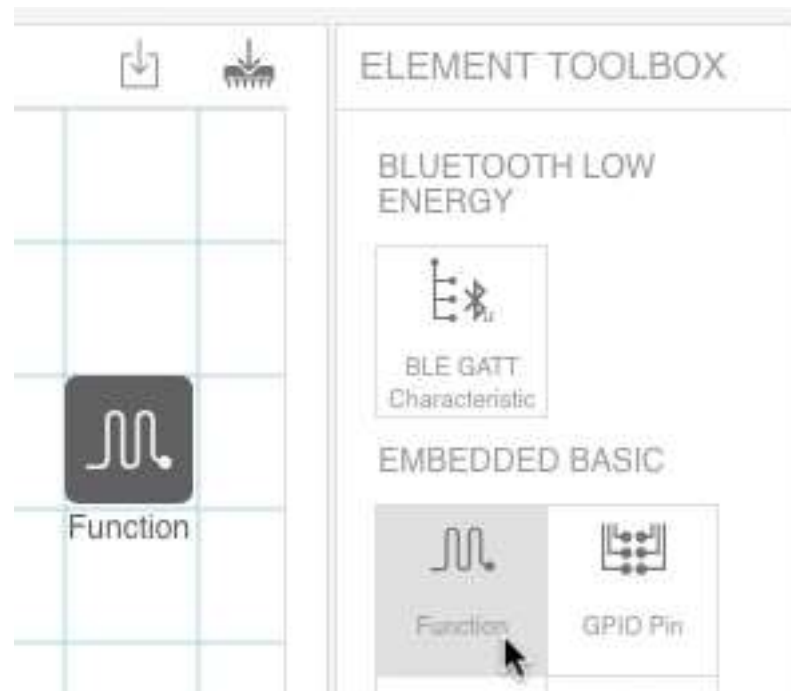
Element Toolbox

Source View Toggle



Elements

- Elements are added by clicking on one of the squares in the Element Toolbox. Once you click, the element will show up in your work area.



Elements (contd.)

- Adding an element automatically adds source code.
- Elements have unique properties and triggers.

The screenshot displays the NXP RAPID IOT IDE interface. On the left is a code editor with the following C code:

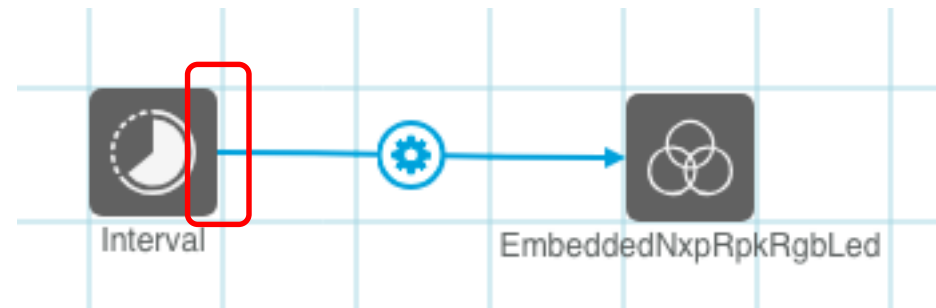
```
1 #include "callbacks.h"
2
3
4
5 void ATMO_Setup() {
6
7
8 }
9
10
11 ATMO_Status_t Function_trigger(ATMO_Value_t *in, ATMO_Value_t *out) {
12     return ATMO_Status_Success;
13 }
14
15
16
17
```

In the center workspace, a 'Function' element (represented by a sine wave icon) is placed on a grid. A red box highlights the code editor icon (</>) at the bottom right of the workspace.

On the right, the 'Function Element' properties panel is visible. It includes sections for 'PROPERTIES', 'NAME', 'Function', 'CODE', 'VARIABLES', 'TRIGGERS', and 'Triggered'. A red box highlights the question mark icon (?) at the top right of the panel, labeled 'Online help'. Another red box highlights the trash can icon at the bottom right of the panel, labeled 'Delete selected element'.

Connecting Elements

- Elements can be joined by hovering your mouse near an element. When the sideways arrow appears, click and drag the arrow to the desired connection point.
- The element connector can be configured similar to an element and has its own source code.



Adding and Editing Source Code

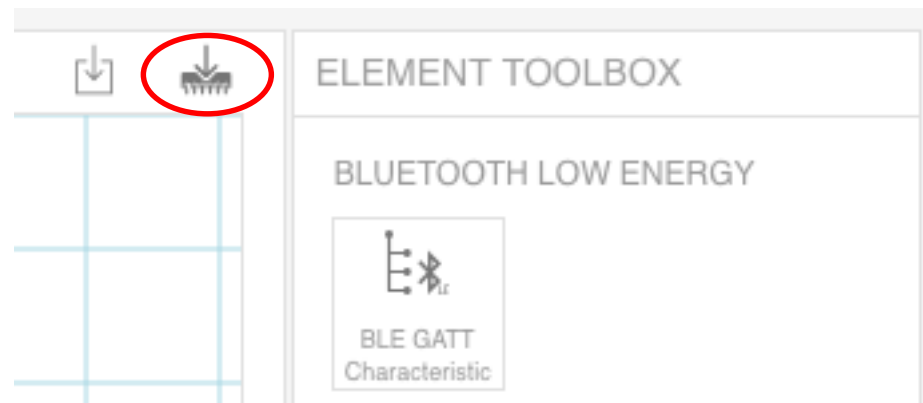
- Source code is automatically added when you add a new element to your application.
- You can add additional source or modify what was generated for you by opening the source code view.

The screenshot displays the NXP RAPID IOT development environment. On the left, a code editor shows the source code for a project named "NXP RAPID IOT". The code includes a header file "callbacks.h" and defines several callback functions for ATMO setup and various button presses. On the right, a block diagram shows the hardware and software components. The diagram includes an "EmbeddedNxpRpkUserButtons" block, an "SX9500Touch" block, and an "EmbeddedPageController" block. The "EmbeddedPageController" block is connected to the "EmbeddedNxpRpkUserButtons" and "SX9500Touch" blocks. Below the "EmbeddedPageController" block, there is an "EmbeddedSystemStatusDisplay" block. At the bottom of the diagram, there are four blocks labeled "Apps", "Settings", "Info", and "Sensor". On the far right, an "ELEMENT TOOLBOX" is visible, containing various components for "BLUETOOTH LOW ENERGY" (BLE GATT Characteristic), "EMBEDDED BASIC" (Function, GPIO Pin, ADC Pin, UART, Interval, On/Off Page, System Status Page, Labelled Icon, Text Page, Icon Text Lines, Display Page Controller, RGB LED, User Buttons, Buzzer, Volansys Thread), and "IMPORTED" (Add Element).

```
1 #include "callbacks.h"
2
3
4
5 void ATMO_Setup() {
6
7
8 }
9
10
11 ATMO_Status_t NFCOnOff_trigger(ATMO_Value_t *in, ATMO_Value_t *out) {
12     return ATMO_Status_Success;
13 }
14
15
16 ATMO_Status_t NFCOnOff_onDisplayed(ATMO_Value_t *in, ATMO_Value_t *out) {
17     return ATMO_Status_Success;
18 }
19
20 }
21
22
23 ATMO_Status_t NFCOnOff_topRightButtonPressed(ATMO_Value_t *in, ATMO_Value_t *out) {
24     return ATMO_Status_Success;
25 }
26
27 }
28
29
30 ATMO_Status_t NFCOnOff_bottomRightButtonPressed(ATMO_Value_t *in, ATMO_Value_t *out) {
31     return ATMO_Status_Success;
32 }
```

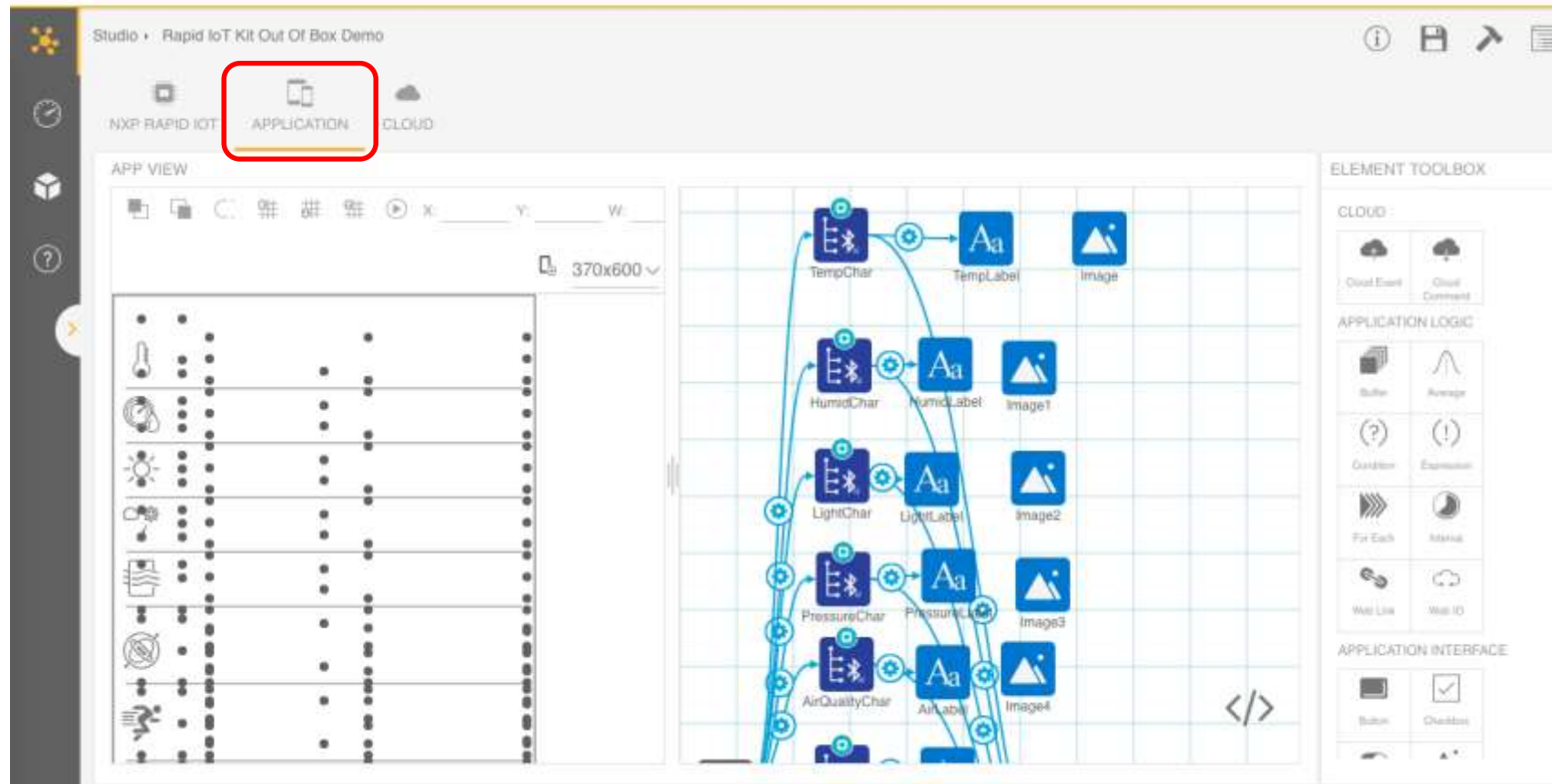
Generating Binary Output

- Studio cannot debug your Rapid IoT board directly.
- It builds the application using the remote IDE and creates a binary file that you can drag and drop onto the Rapid IoT's USB MSD bootloader.
 - The bootloader automatically detects the target MCU and loads the binary accordingly.
- **Click on the “arrow and chip” icon** in the upper right corner to download project binary/firmware (once compiled)



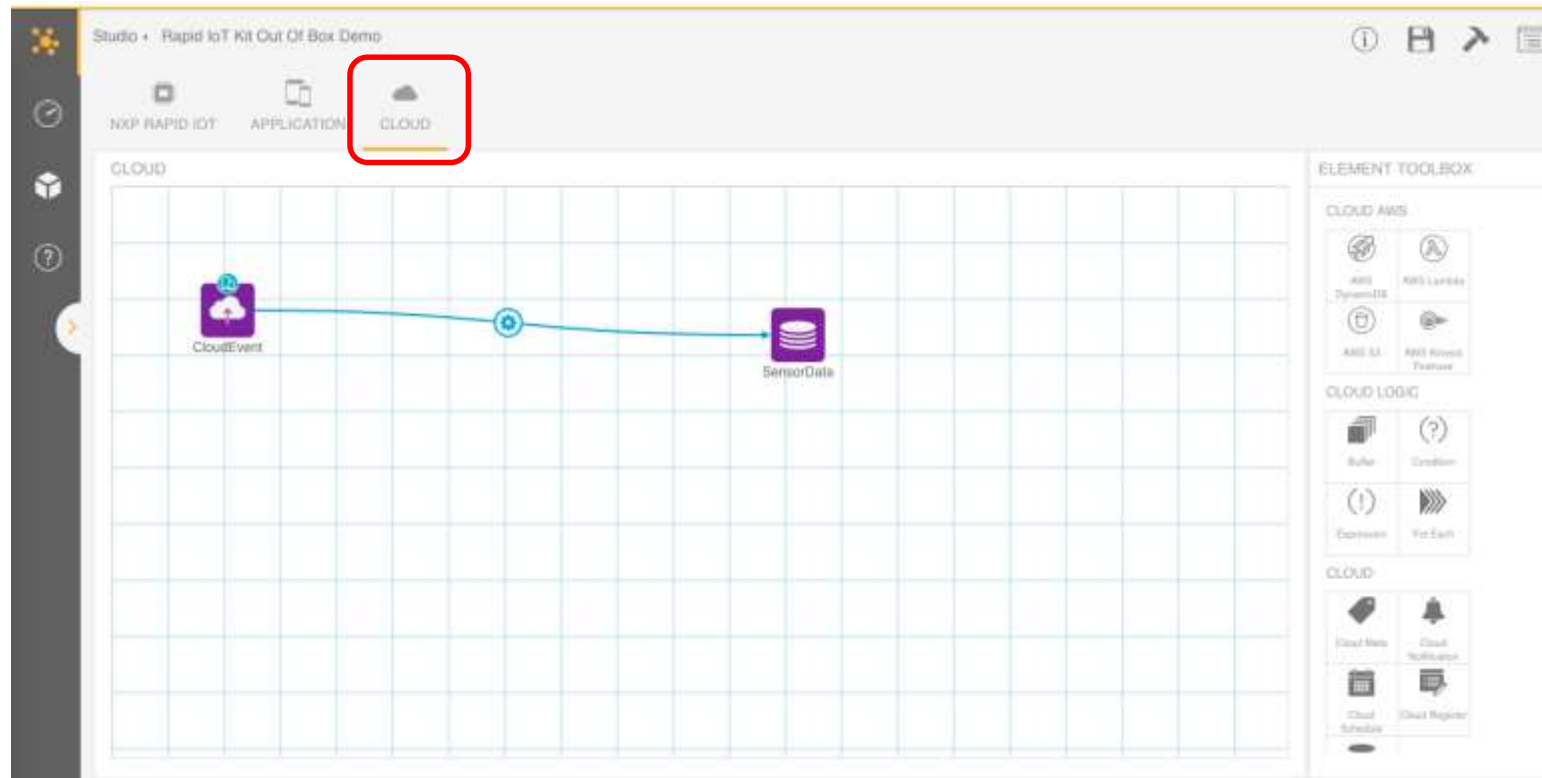
Application (Mobile) View

- Studio provides the ability to create cross-platform (Android and iOS) mobile apps for your IoT application.
- Uses elements just like the embedded application view.



Cloud View

- Studio also provides the ability to create cloud apps for your project.
- Uses elements just like the other views
- Manages Cloud Events and Cloud Commands (after being added in the App View)



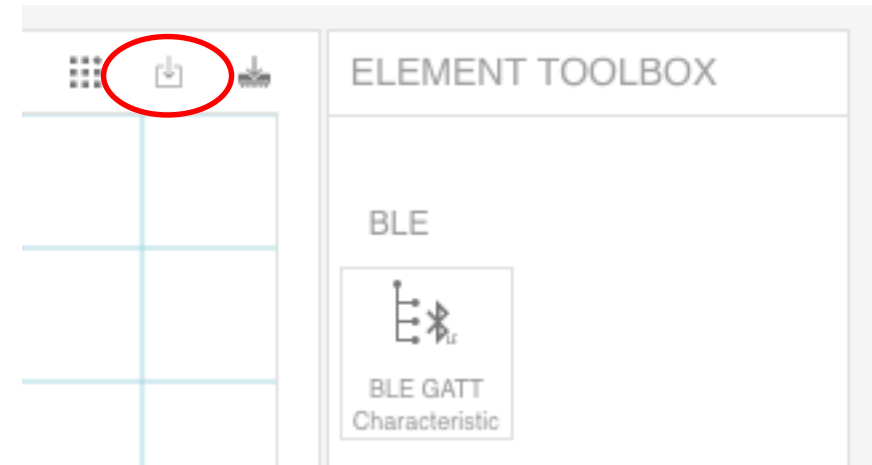


03.

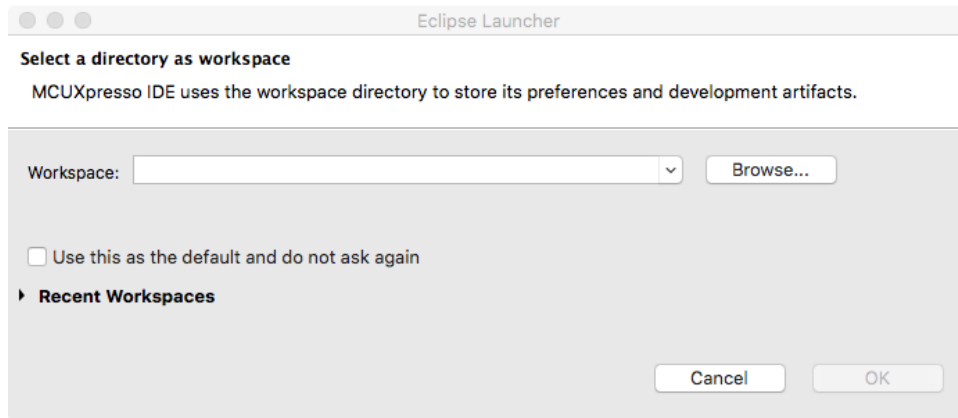
Exporting to MCUXpresso IDE

Exporting MCUXpresso Project from Studio IDE

- Generates source package that is importable into MCUXpresso IDE
- Based on the Solution Software package
- Uses Atmosphere's ATMO API
- A zip file named "<Project_Name> source.zip" is created and contains all source files – there is no linking to SDKs, etc.

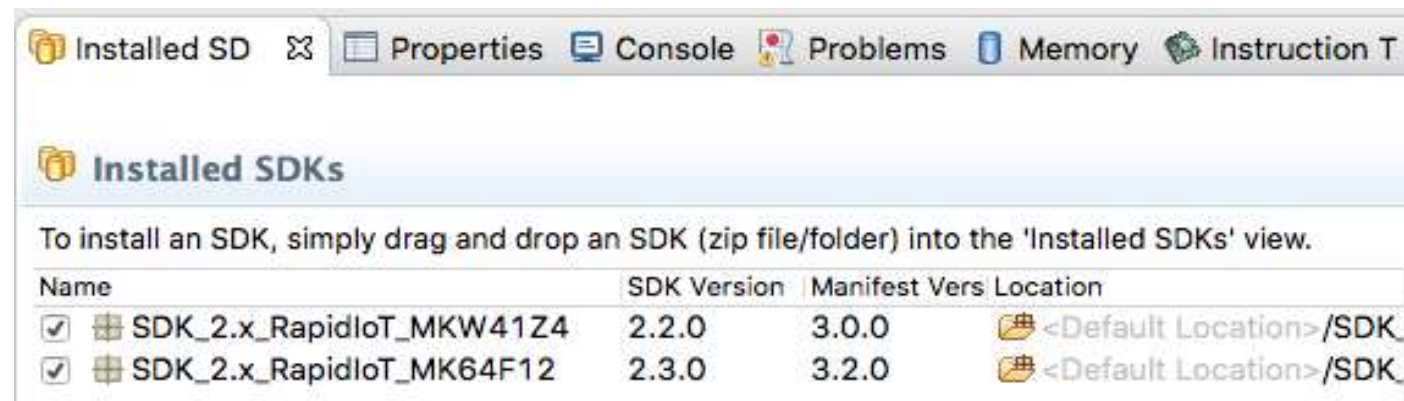


Setting up MCUXpresso for the Rapid IoT

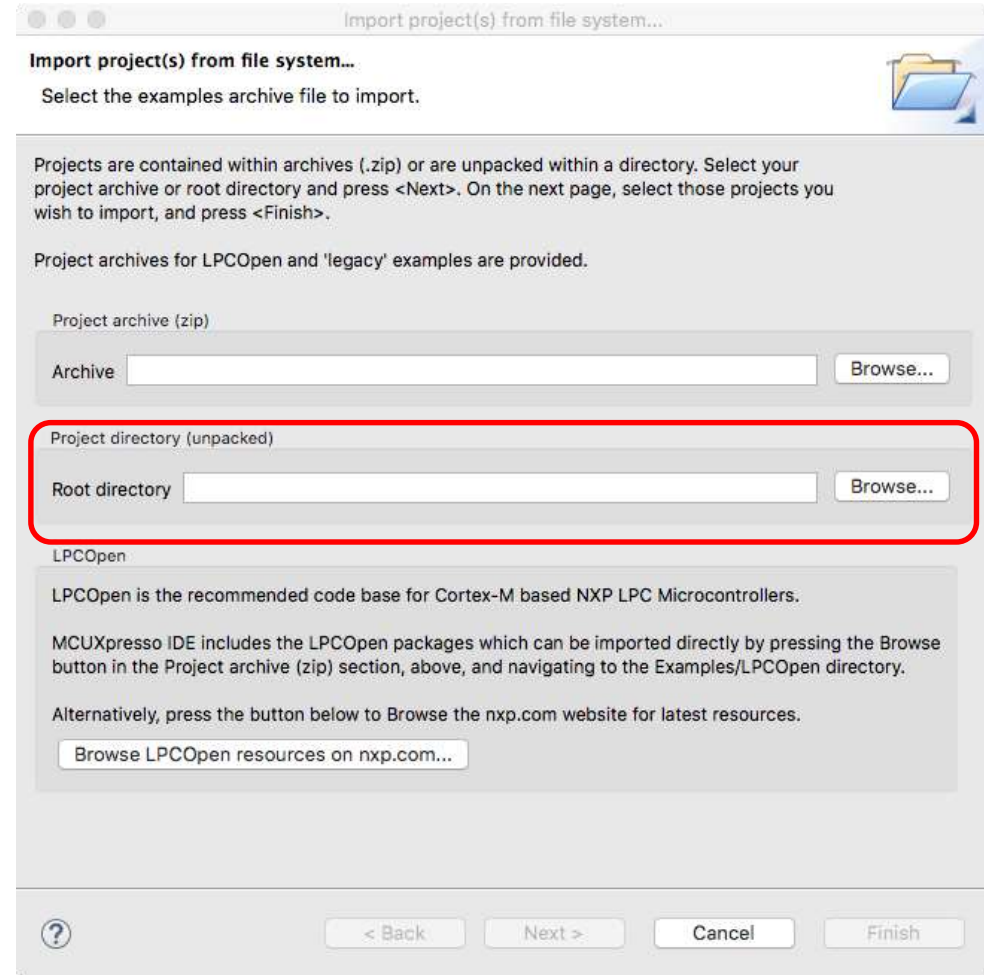
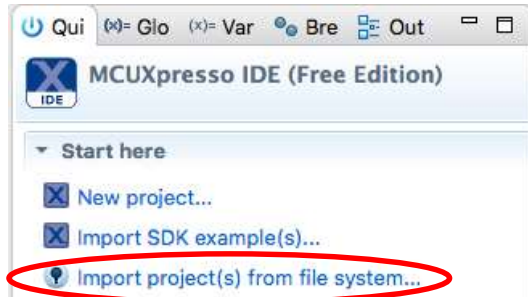


- 1) Create a workspace in MCUXpresso (10.2 as of today). You can name it anything you want.

- 2) Drag-and-drop the K64F and KW41Z SDKs from the Rapid IoT Solution software package into MCUXpresso IDE. Note: these are not included in the Atmosphere package, but are required to install device support for K64F and KW41Z into MCUXpresso IDE.

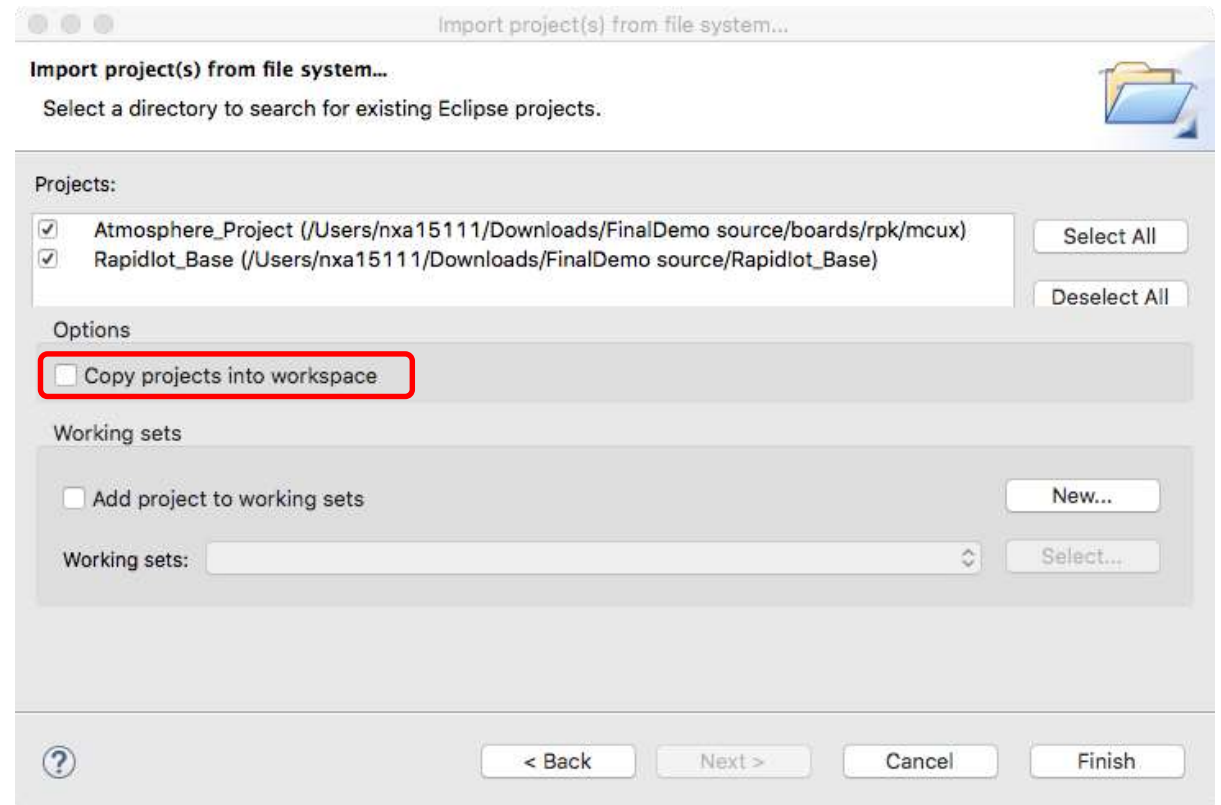


Importing Into MCUXpresso IDE



Importing Into MCUXpresso IDE (contd.)

- Navigate to the location where you downloaded and extracted the Atmosphere source package and select the folder. Click 'Next'.
- On the next screen, select both projects. Make sure to uncheck the “Copy projects into workspace” option. Click 'Finish'.



Building the Projects

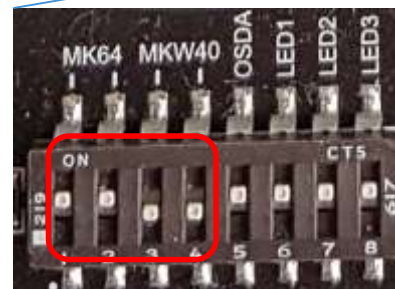
- 1) First, build the “Rapidlot_Base” library. To do this, select the project in the workspace and then click on the hammer icon.



- 2) Once the base library builds, repeat the same process as above but for the “Atmosphere_Project” project.

Debugging the Board Using the Docking Station

- The Rapid IoT plugs into the Hexiwear docking station using the larger of the two high density connectors on the back side.
- Plug in a micro-USB cable to the debug port of the docking station.
- Use the power switch to power the Rapid IoT on or off.
- The dip switches must be set as shown below to correctly target either the K64F (desired for this exercise) or the KW41Z.



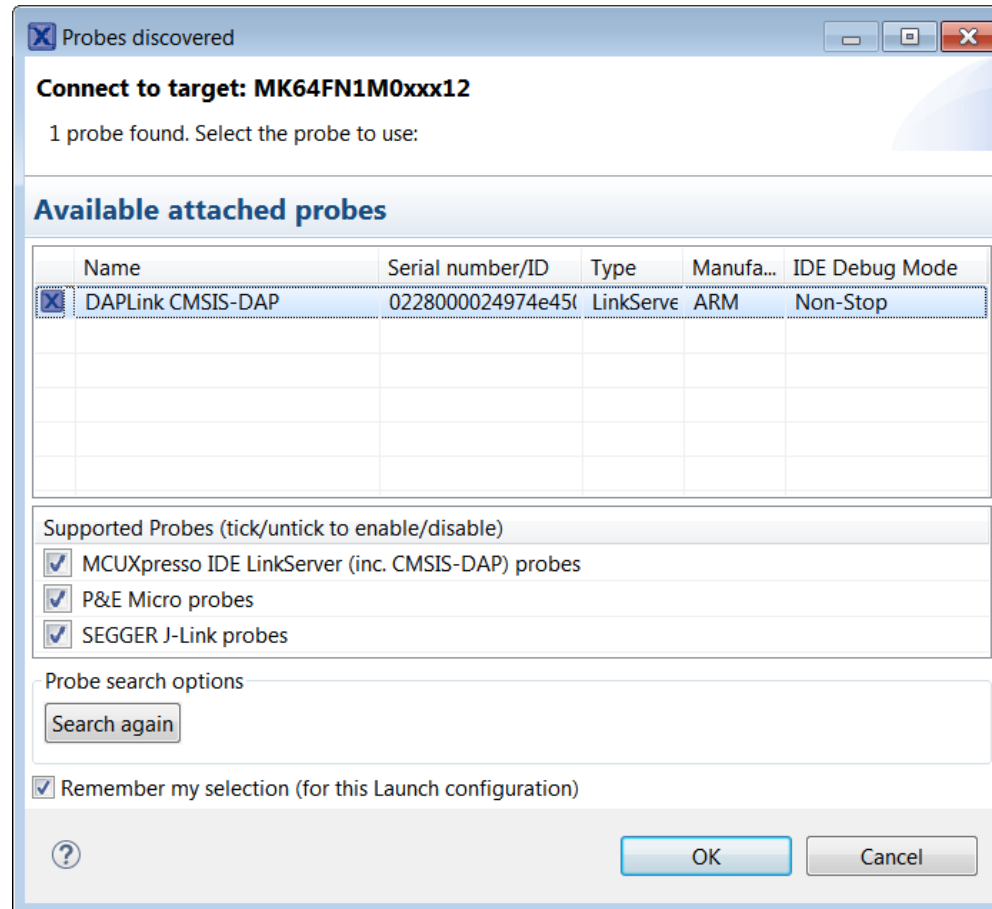
Debugging the Board Using the Docking Station (contd.)

- With the “Atmosphere_Project” project selected in MCUXpresso IDE, click on the “Debug ‘Atmosphere_Project’” button in the Quickstart pane.
- Selecting this will launch the debugger. MCUXpresso IDE will automatically launch the debug perspective if the docking station is plugged in via USB.



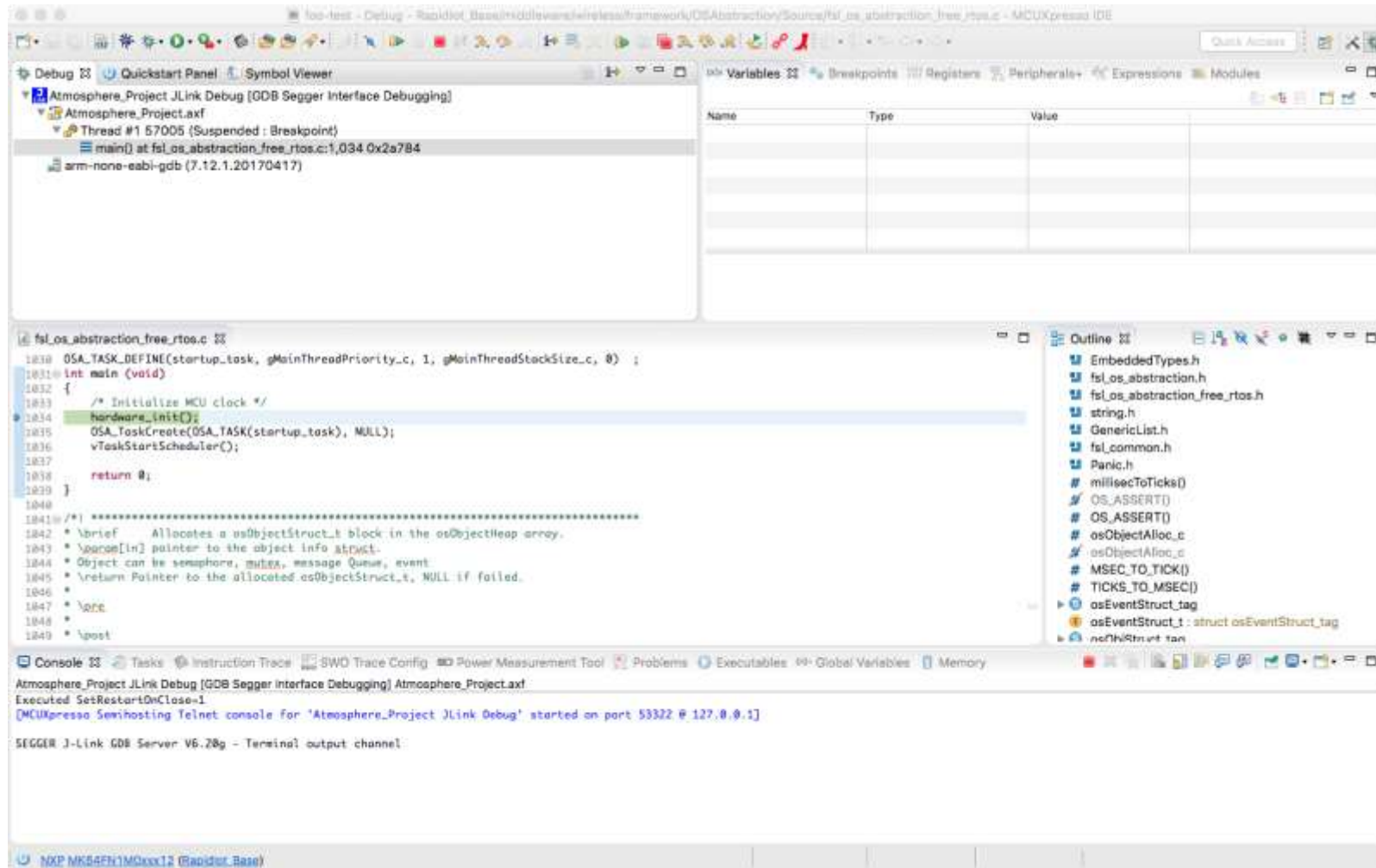
Debugging the Board Using the Docking Station (contd.)

- In some cases MCUXpresso IDE will show a dialog window with the available debug probes. Select the **DAPLink** option and click on “OK”.



Debugging the Board Using the Docking Station (contd.)

- Once the debugger launches, MCUXpresso IDE will look like the image below.





04.

Solution Software Overview

Get the SLN-RPK-NODE Software Package

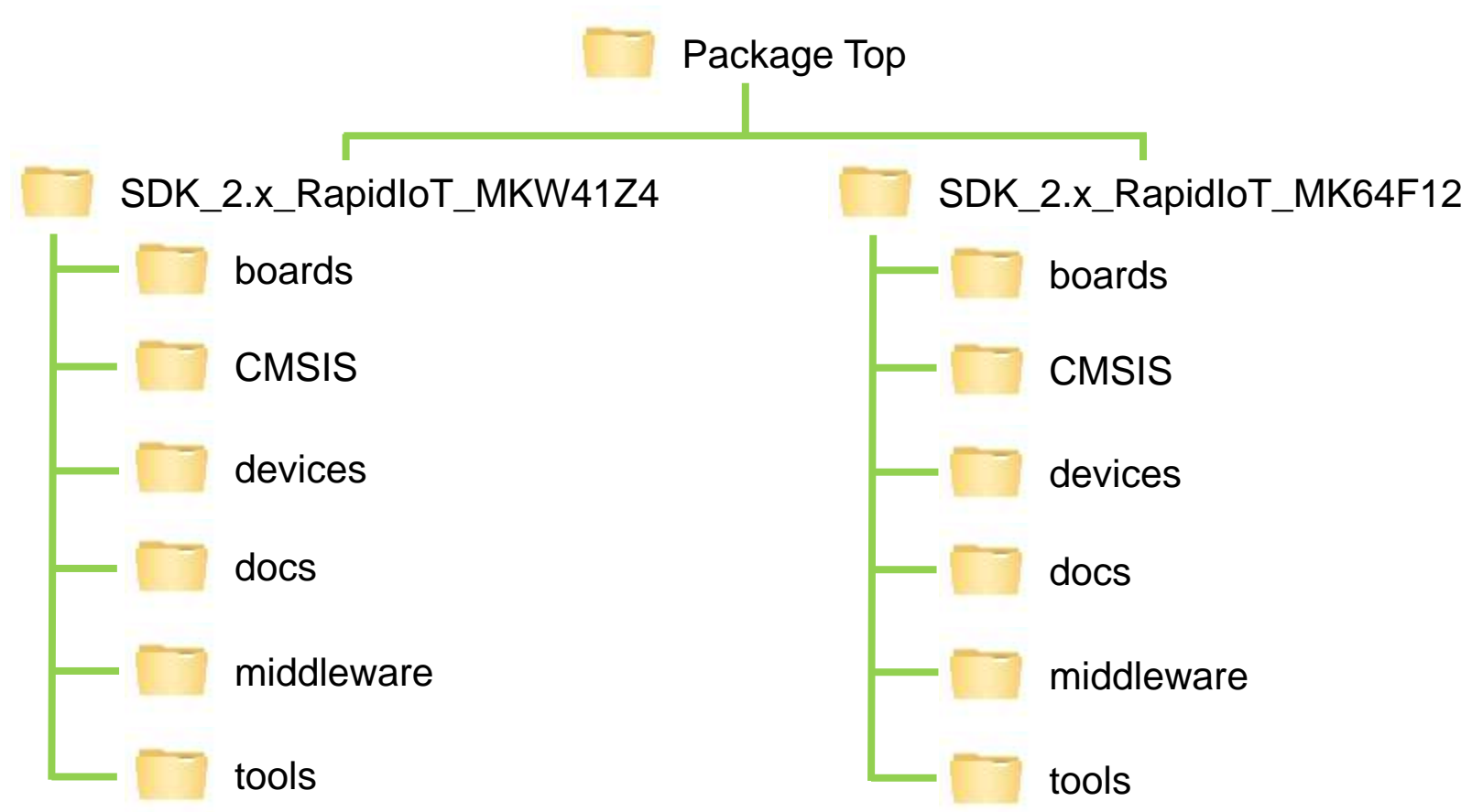
- At launch, software is available for download from the Rapid IoT website at www.nxp.com/rapid-iot



Based on MCUXpresso SDK

- Familiar source code layout and folder structure
- Importable into MCUXpresso IDE
 - Rapid IoT SDKs import into the tool, so you don't need to download any additional packages
 - Post launch, full MCUXpresso IDE integration will be supported
 - Post launch, available via MCUXpresso SDK builder
- Uses MCUXpresso/Kinetis Connectivity Framework to support BLE and Thread
- Interface between K64F and KW41Z is FSCI – host controlled black-box implementation for connectivity

Directory Structure





05.

Creating Projects with Rapid IoT Studio



Demo – Weather Station

- Go to <http://rapid-iot-studio.nxp.com>
- Enter your credentials (or register first)
- Go to Studio main page
- Open the Weather Station example
- Look at Embedded/Application/Cloud views
- Build the project
- Download the binary
- Flash the binary



Demo – Weather Station

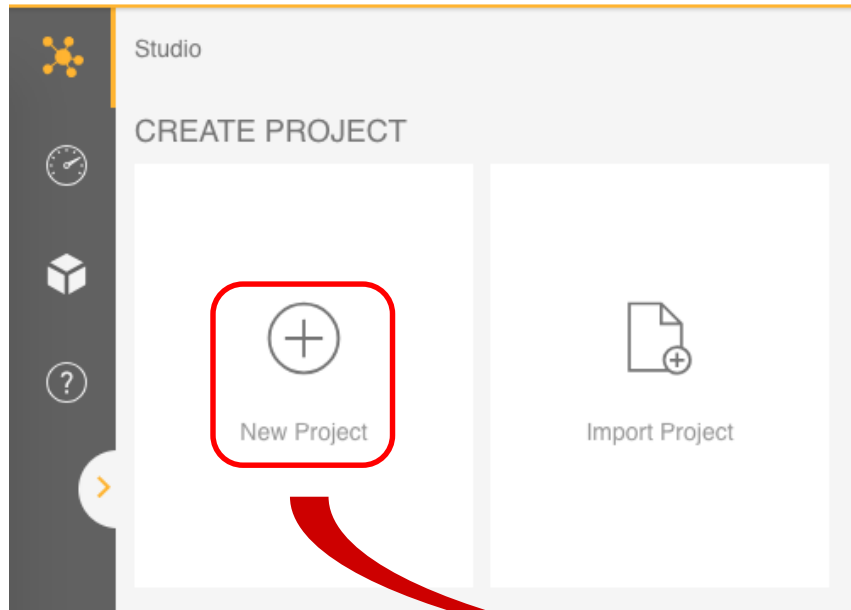
- Install the smartphone application (Android or iOS)
- Log in with your credentials
- Provision your device (BLE)
- Monitor the sensor data (real time)
- Monitor the sensor data (Cloud storage)

Project 1 – LED Toggling

- Assign buttons on the Rapid IoT to toggle various colors of the RGB LED
- Button 1 – All off
- Button 2 – Toggle Red
- Button 3 – Toggle Blue
- Button 4 – Toggle Green





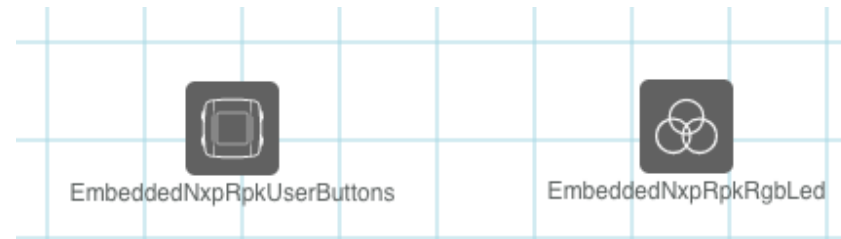
Create the Project



The 'NEW PROJECT' dialog box is shown. It has a title bar 'NEW PROJECT'. Below the title bar, there is a 'NAME' label followed by a text input field containing the text 'LED Buttons'. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'CONFIRM'. The 'CONFIRM' button is highlighted with a yellow border.

Add Elements

- 1) We only need two elements to create this application: user buttons and the RGB LED. In the Element Toolbox, click on the  and  to add the elements to your work area.

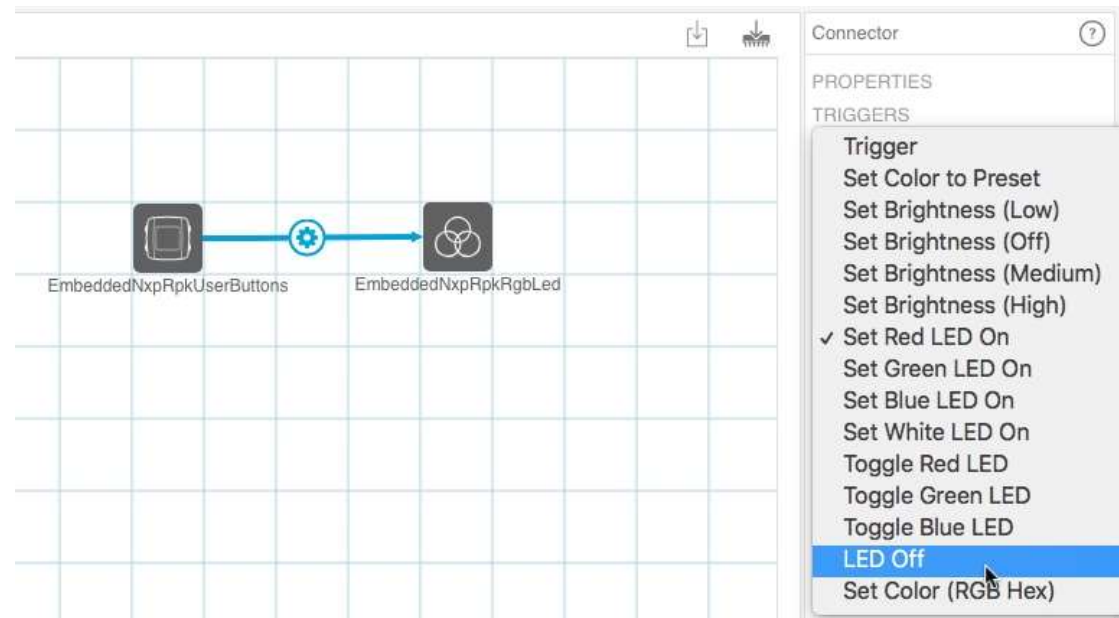


- 2) Next, connect the two elements.



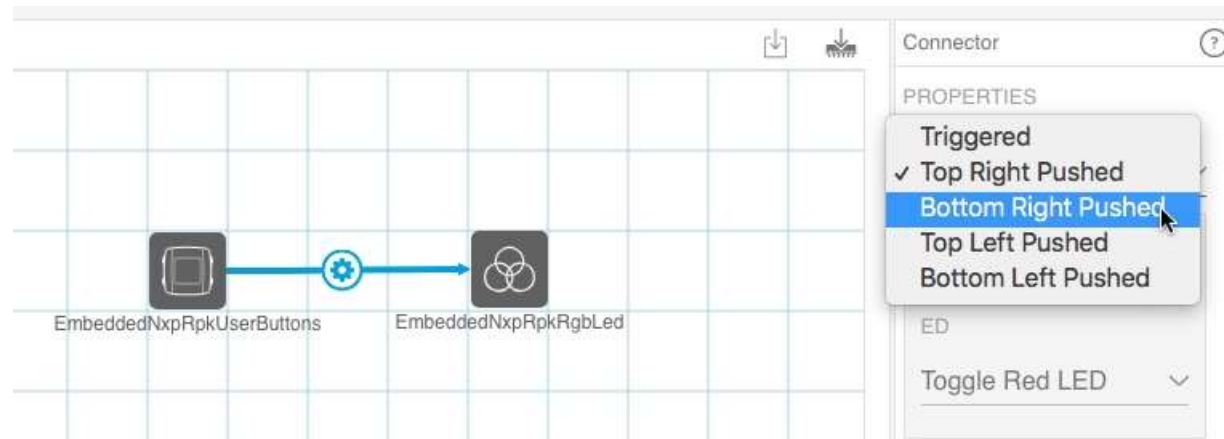
Configure the Connector


- 3) Select/click on the connector.
- 4) The connector automatically adds an event for **Top Right** by default. We want to change the functionality from **Toggle Red LED**, which is the default, to **LED Off**. In the connector properties on the right side of the screen, select **LED Off** in the trigger action drop-down box.





Configure the Connector (contd.)

- 5) To configure the other buttons, select the action in the "Triggers" drop-down. Lets go in order, so select **Bottom Right Pushed** first.



- 6) Click on the **Add Event** button  to create the event for the **Bottom Right** trigger. In the trigger action drop-down, select **Toggle Red LED**.
- 7) Repeat these steps for Top Left and Bottom Left. For **Top Left** choose **Toggle Blue LED**, and for **Bottom Left** choose **Toggle Green LED**.

That's It! Now we Build

- 8) That's it! The next step is to build the project. Click on the **Compile** button  to build.
- 9) Once the build completes, we generate the output binary by clicking on the **Program Firmware** button  .
- 10) Atmosphere IDE will then link the project and generate the binary output. Your browser will automatically trigger the download when complete. The output will be in a file named “<Project Name> firmware.bin”, so in this case, **LED Buttons firmware.bin**.

Entering Bootloader Mode

11) Remove Rapid IoT from the docking station and plug a USB cable into the bottom of the Rapid IoT

12) Press and hold the top left and bottom right buttons until the green LED is blinking



13) The left-most LED will blink green when in bootloader mode



Programming Binary Using Bootloader

- 14) Drag and drop the binary file onto the USB drive that shows up in your PC's file browser. Note the drive letter may be different on your PC. On Mac or Linux machines, the USB drive may be named "RPK".

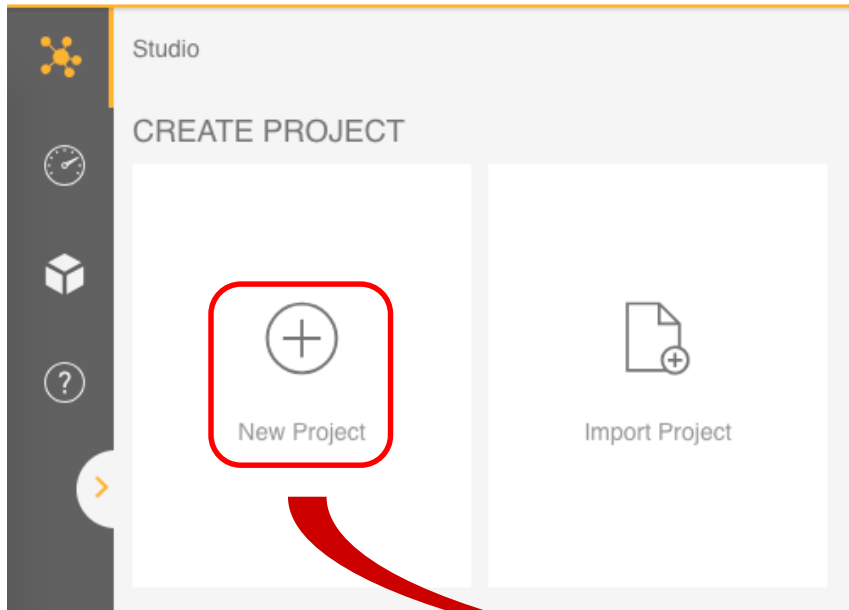




Project 2 – BLE Light Sensor

- Use BLE to send ambient light data to mobile application
- Show same data on the Rapid IoT's screen

Create the Project



The 'NEW PROJECT' dialog is shown with a dark header. Below the header, there is a 'NAME' label and a text input field containing 'BLE Light Sensor'. At the bottom right of the dialog, there are two buttons: 'CANCEL' and 'CONFIRM'. The 'CONFIRM' button is highlighted with a red border.

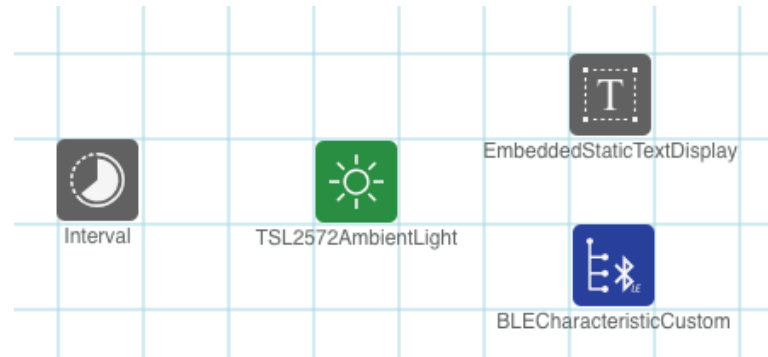
Add Elements

- 1) For this project, we want to use BLE to send ambient light 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, use the following items in the Element Toolbox:

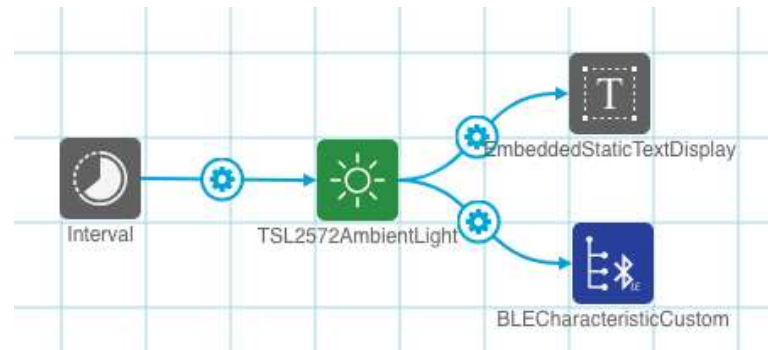


Add Elements (contd.)

2) When all of the elements are added, you should see something like this:



3) Next, connect the elements like you see below:



Configure the BLE Characteristic

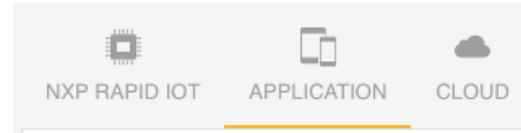
- 4) When adding a characteristic, Studio automatically adds UUIDs and sets some default parameters. We need to adjust them for our specific need. We need to make the characteristic **read-only** and change the data types to **Unsigned Integer**.

The screenshot displays the Studio IDE interface. On the left, a block diagram shows the following components and connections: an 'Interval' component (moon icon) is connected to a 'TSL2572AmbientLight' component (sun icon). The 'TSL2572AmbientLight' component is connected to two other components: 'EmbeddedStaticTextDisplay' (text icon) and 'BLECharacteristicCustom' (Bluetooth icon). On the right, the 'BLE GATT Characteristic' configuration panel is visible. It shows the following settings:

- INSTANCE: 0
- SERVICE UUID (128-BIT): 9243f6fc-7f9e-4921-a848
- CHARACTERISTIC UUID (128-BIT): 9243f6fc-7f9e-4921-a848
- READ:
- WRITE:
- NOTIFY:
- READ DATA TYPE: Unsigned Integer
- WRITE DATA TYPE: Unsigned Integer
- NOTIFY DATA TYPE: Unsigned Integer

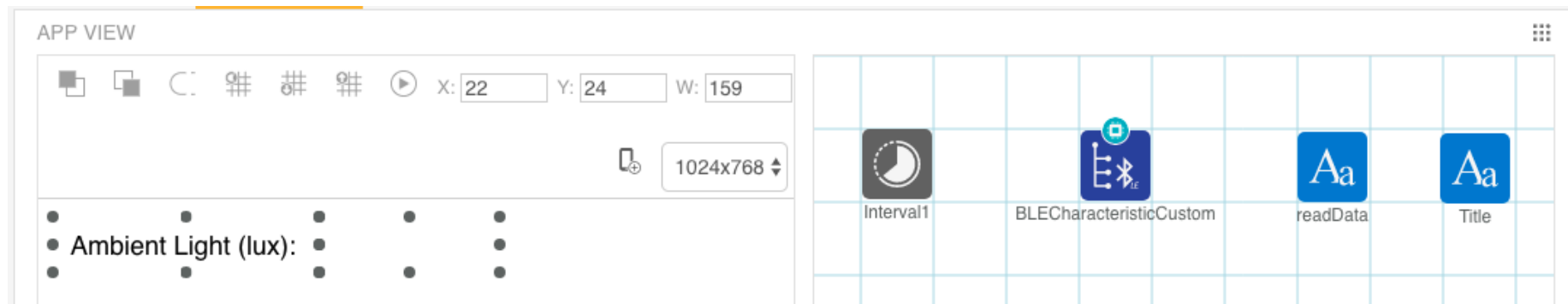
Create the Mobile Phone Application

5) Select the **Application** view



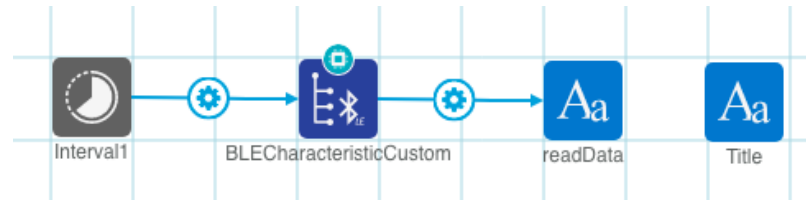
6) You'll notice that there is already a BLE characteristic that corresponds to the one you added in your embedded application. Now we need to configure the mobile application. First, we need an interval to determine how often to read data from the Rapid IoT. After that, we need to display the data somewhere.

The application configuration below will accomplish the task. Use the interval and label elements in the Element Toolbox to add the additional elements.



Create the Mobile Phone Application (contd.)

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



- 8) That's it for the connectors! There is no additional configuration needed.
- 9) 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. 📱

Create the Mobile Phone Application (contd.)

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



ADD LAYOUT

Search

<input type="checkbox"/>	Google Pixel (XL, 2, 2 XL)	411	731
<input type="checkbox"/>	iPad	768	1024
<input type="checkbox"/>	iPad Pro	1024	1366
<input type="checkbox"/>	iPhone 6 Plus/6S Plus	414	736
<input type="checkbox"/>	iPhone 6/6S	375	667
<input type="checkbox"/>	iPhone 7	375	667
<input type="checkbox"/>	iPhone 7 Plus	414	736
<input type="checkbox"/>	iPhone 8	375	667
<input type="checkbox"/>	iPhone 8 Plus	414	736
<input checked="" type="checkbox"/>	iPhone X	375	812
<input type="checkbox"/>	iPod Touch	320	568

CANCEL SAVE

Build the Applications

- 11) To build the applications (Embedded and Mobile), find the **Compile** button on the top right of the window and click on it. 
- 12) Then press the **Program Firmware** button in the Embedded view to download the binary for the Rapid IoT. 
- 13) Atmosphere IDE will then link the project and generate the binary output. Your browser will automatically trigger the download when complete. The output will be in a file named “<Project Name> firmware.zip”, so in this case, **BLE Light Sensor firmware.zip**. Extracting the file yields the Atmosphere_Project.bin file.

Entering Bootloader Mode

14) Remove Rapid IoT from the docking station and plug a USB cable into the bottom of the Rapid IoT

15) Press and hold the top left and bottom right buttons until the green LED is blinking



16) The left-most LED will blink green when in bootloader mode



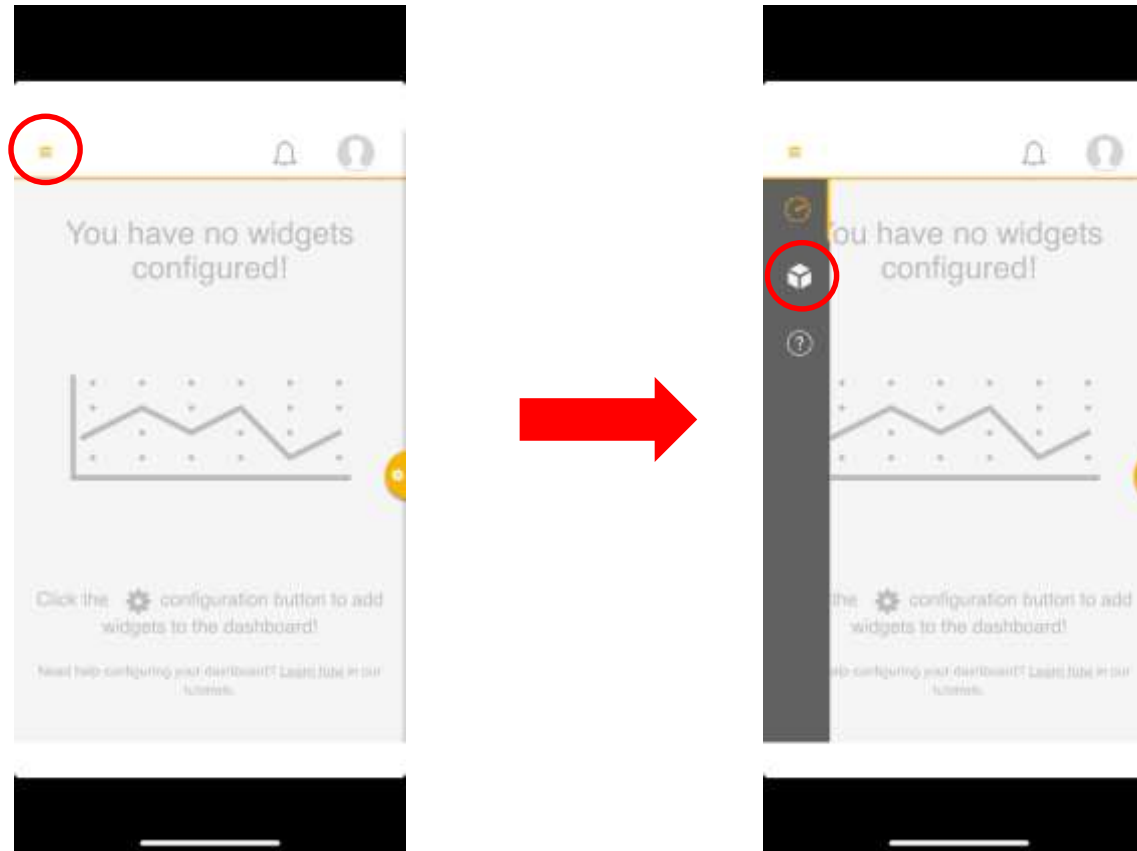
Programming Binary Using Bootloader

- 17) Drag and drop the binary file onto the USB drive that shows up in your PC's file browser. Note the drive letter may be different on your PC. On Mac or Linux machines, the USB drive may be named "RPK".



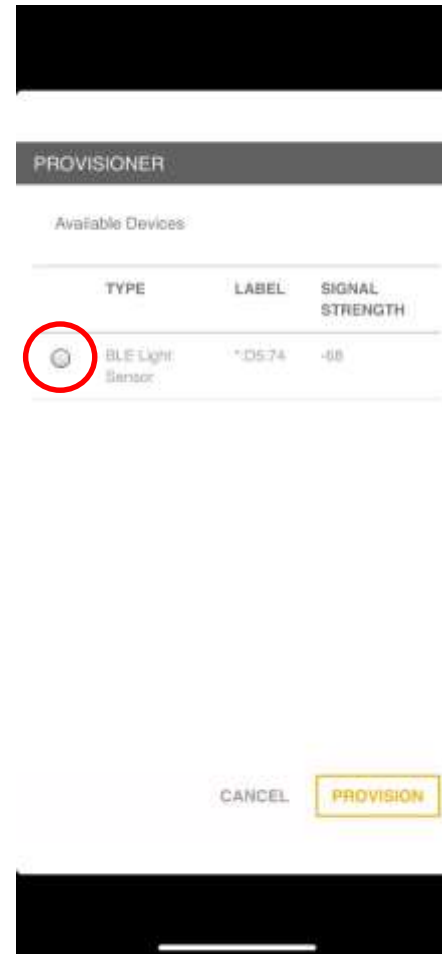
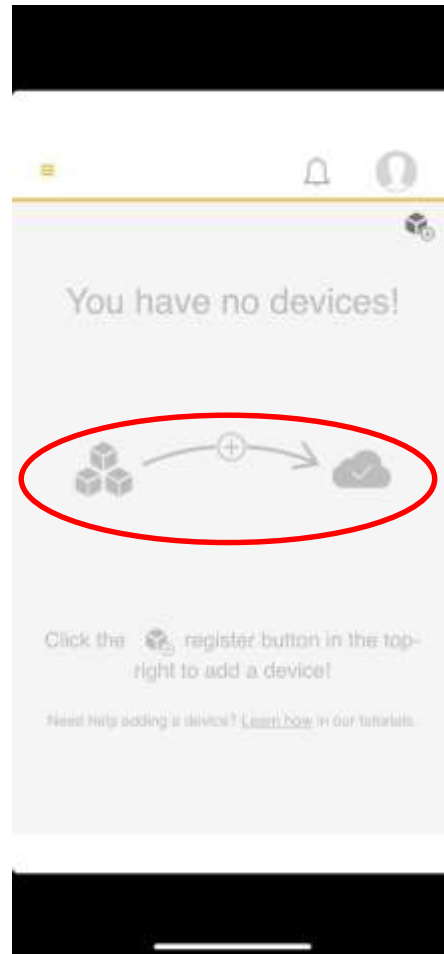
Using the Rapid IoT Mobile Application

- Download the NXP Rapid IoT app from your respective app store. Your NXP credentials are used to log in to the app.
- The first step is provisioning your application within the mobile app.



Using the Rapid IoT Mobile Application (contd.)

- Click on the button to add/provision a new device.



Using the Rapid IoT Mobile Application (contd.)

- The app will automatically start





06.

Summary

Summary



Comprehensive, pre-engineered IoT node

- 20+ devices: Processing, connectivity, security, sensing
- Expandable with 400+ Click boards™



Extreme ease of use

- Software enablement: RTOS, drivers, middleware
- Web IDE with GUI based programming



Secure from sensor to cloud

- NXP's proven security
- iOS/ Android mobile apps and IoT Cloud connect





SECURE CONNECTIONS
FOR A SMARTER WORLD