QuickConnect Studio Platform

Introduction

Renesas QuickConnect Studio (QCS) is a cloud-based embedded system design and prototyping platform that enables developers, educators, and engineers to create, configure, compile, and debug embedded applications entirely within a web browser.

It provides a visual, drag-and-drop interface where users can assemble microcontroller boards, sensors, connectivity modules, and power components into complete systems, and automatically generate the corresponding embedded software stack for rapid prototyping.

QCS provides a browser-based environment that integrates the following:

- Hardware kits (Renesas RA, RZ families and partner ecosystems)
- Middleware and driver stacks (RTOS, connectivity, sensors, security, AI/ML)
- Remote board-farm for instant hardware validation and debugging
- Automated project generation with export to Renesas e² studio and toolchains (GCC, LLVM)

Key Features

- Application-centric Design – Start with use-case driven application templates and expand with modular building blocks.
- Board Farm Integration – Remotely access real hardware with SEGGER tunnel/J-Link for flashing, debugging, and validation.
- Low-Code Environment – Drag-and-drop system design blocks while providing real-time hardware compatibility checks.
- Project Export – Seamless export to e² studio for advanced customization.
- Collaboration and Education – Integrated support for GitHub and Renesas University Program, enabling rapid onboarding for students and researchers.

Contents

1. Getting Started

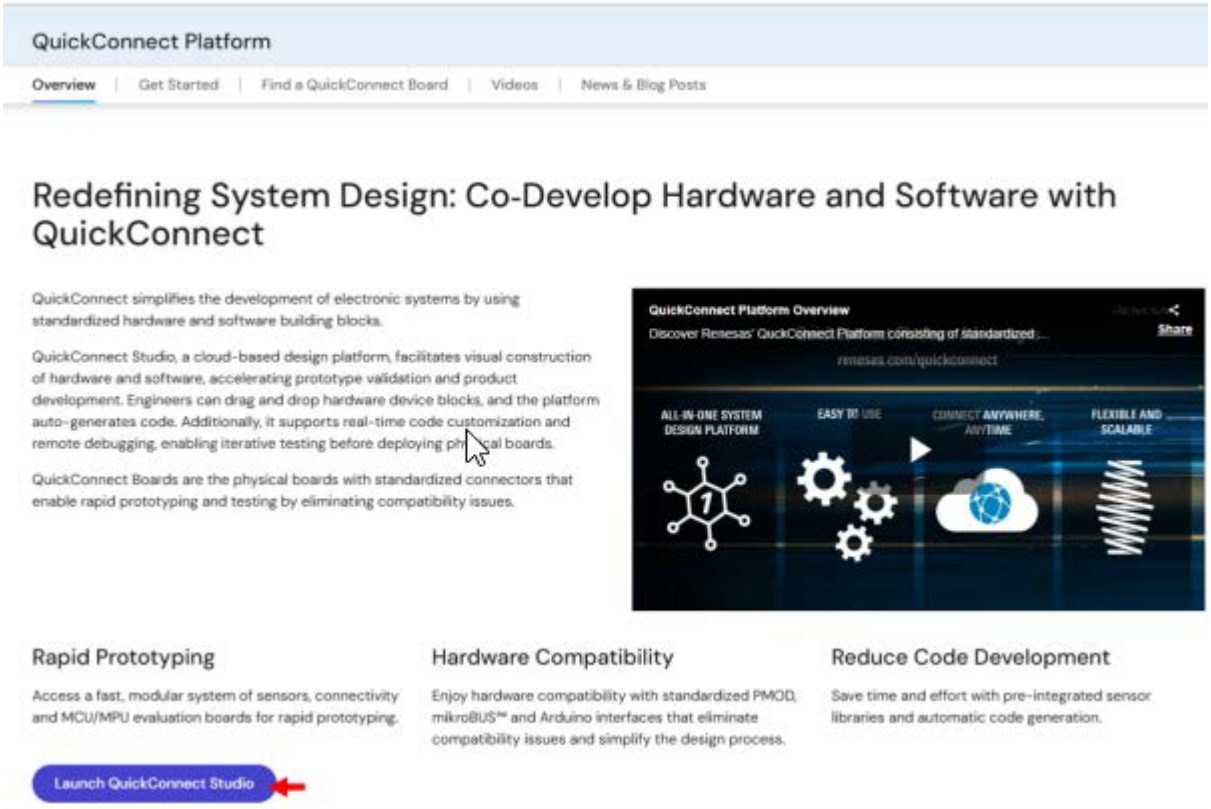
1.1 Prerequisites

The following prerequisites are the default requirements. Depending on the application use cases, the prerequisites can change. Refer to the individual application project `README.md` generated in the QCS application project folder for application-specific requirements.

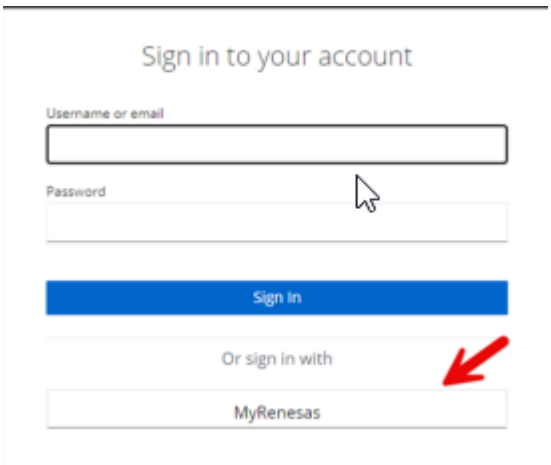
- Windows PC – A computer running the Windows operating system. Renesas recommends using a system running Windows 10 or later versions, equipped with at least one available USB port.
- Web Browser – Ensure a web browser is installed on a Windows PC. For optimal performance, use Google Chrome.
- Internet Connection – A stable internet connection is required to access online resources and download the required files.
- MyRenesas Registration – An active MyRenesas account must be obtained to access QuickConnect. If you do not have one, register on the MyRenesas page.

1.2 Launch QuickConnect Studio Workspace

1. Launch the QuickConnect Studio platform in a PC browser window.
   a. To launch a QuickConnect Studio user workspace, visit the QuickConnect Studio landing page at www.renesas.com/quickconnect.
   b. Click on the **Launch QuickConnect Studio** button to launch a unique workspace in a browser window.



2. At the following screen, click on the **MyRenesas** button to log in using MyRenesas login credentials.



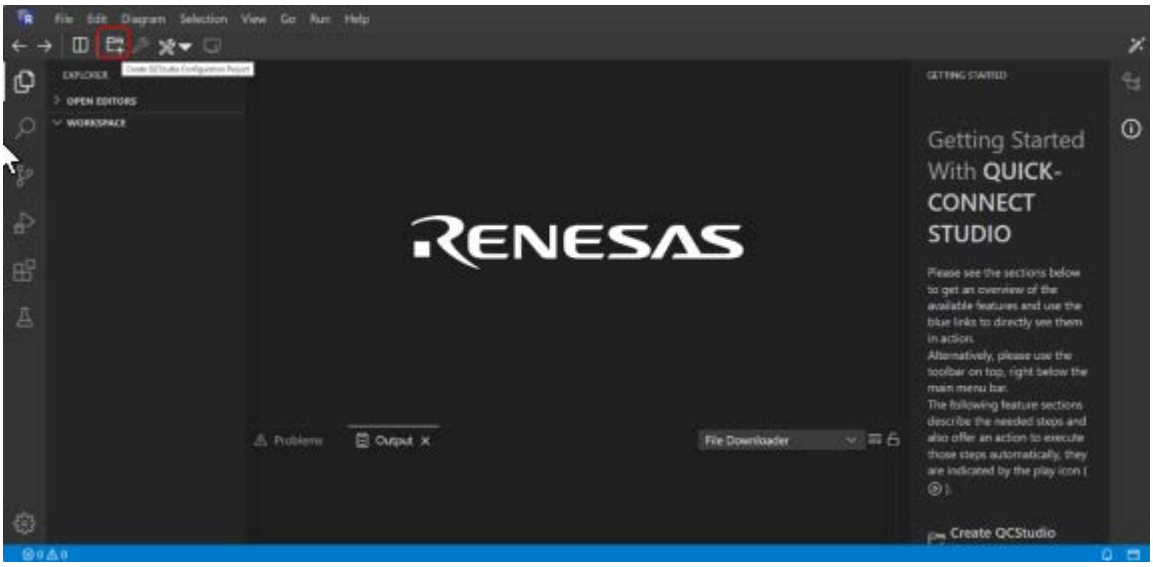
Note: If the user has logged into the MyRenesas account with the same browser window and the session is active, the login window is skipped, and the user workspace is automatically created.

1.3 Create a QCS Application

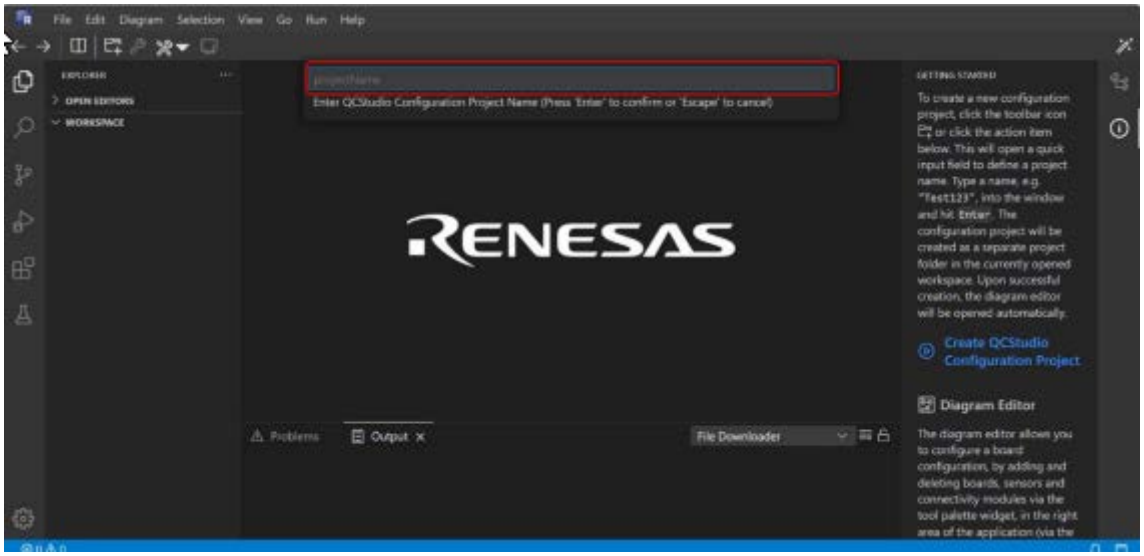
At this stage, it is assumed that the user reviewed the prerequisites and launched a unique user workspace.

In this section, a blinky project is created for the BGK-RA6E2 kit, which acts as a getting started guide to showcase the QCS platform capabilities to prototype an idea into a working solution in minutes.

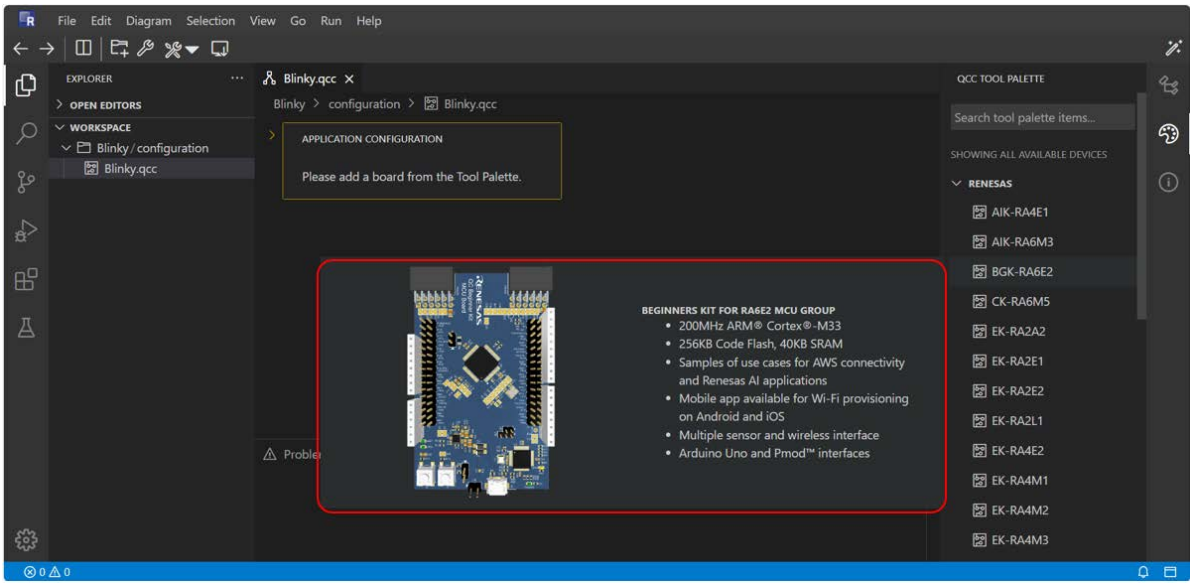
1. Click on the **Create QCStudio Configuration Project** icon as highlighted in the following image.



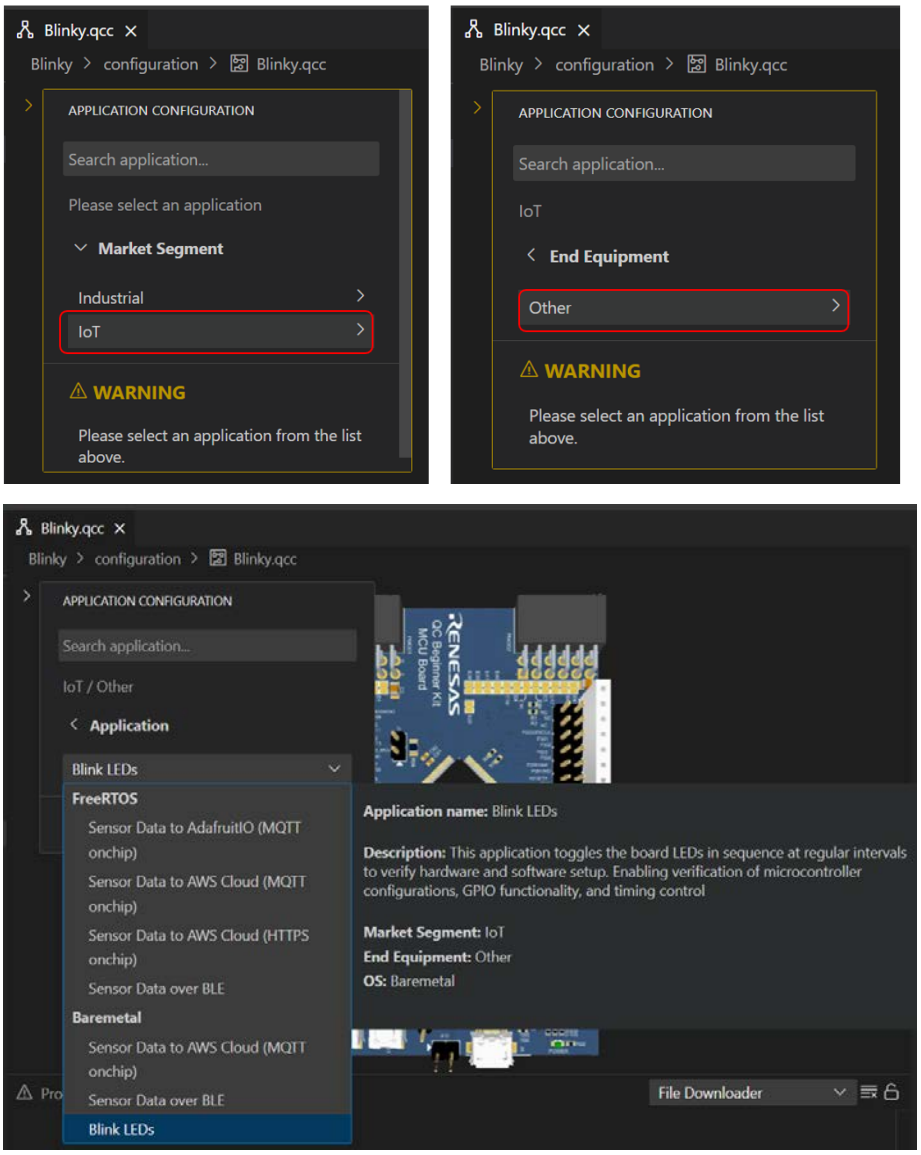
2. Enter the project name in the field that appears and press the **Enter** key.



3. Drag and drop the MCU kit (BGK-RA6E2) from the QCC tool palette. Hover the cursor over the MCU kit at the QCC Tool palette to view its key features.



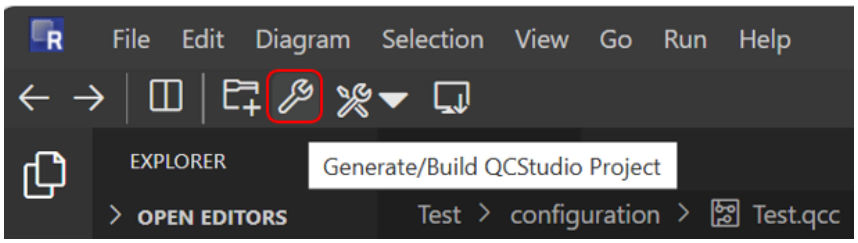
4. In the Application configuration Window, select **IoT** > **Other** > **Blink LEDs**.



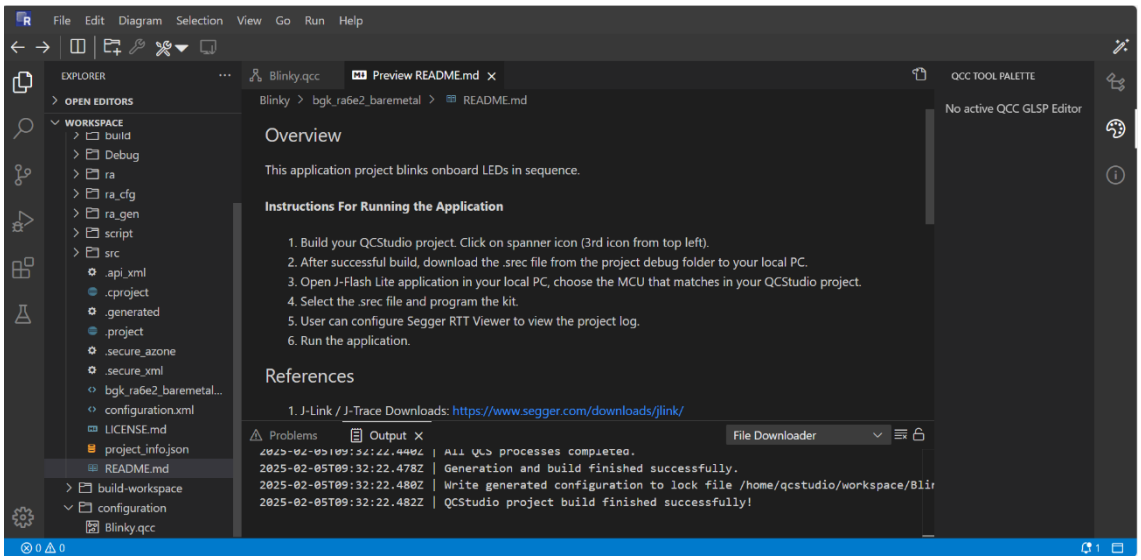
1.4 Build QCS Application

At this stage, it is assumed that the user created the QCS application following the steps shown in the Create QCS Application section.

1. Click on the QCS Application `.qcc` file which has the following path:
   `<qcs_project_name>/configuration/qcs_project_name.qcc`
2. Click on the **Build Project** icon. The build progress is displayed in the output log.
   During the build process, the required software package is autogenerated including drivers, middleware, and application codes.



3. After the project is built, a notification appears stating, **QCStudio project build finished successfully**. The application-specific `README.md` file gets loaded for further instructions.



1.5 Download and Debug QCS Application

At this stage, it is assumed that the user created the QCS application and built the application project following the instructions from the previous sections.

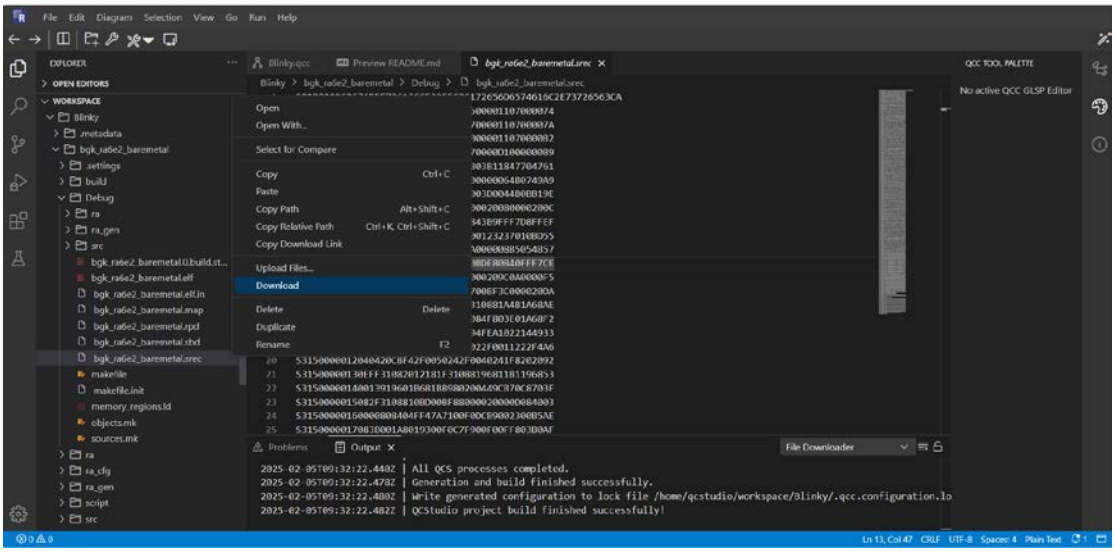
There are three ways to validate the QCS application project output.

- Flashing using the SEGGER J-Flash Lite Utility
- QCS Direct Debugging
- QCS Remote Debugging

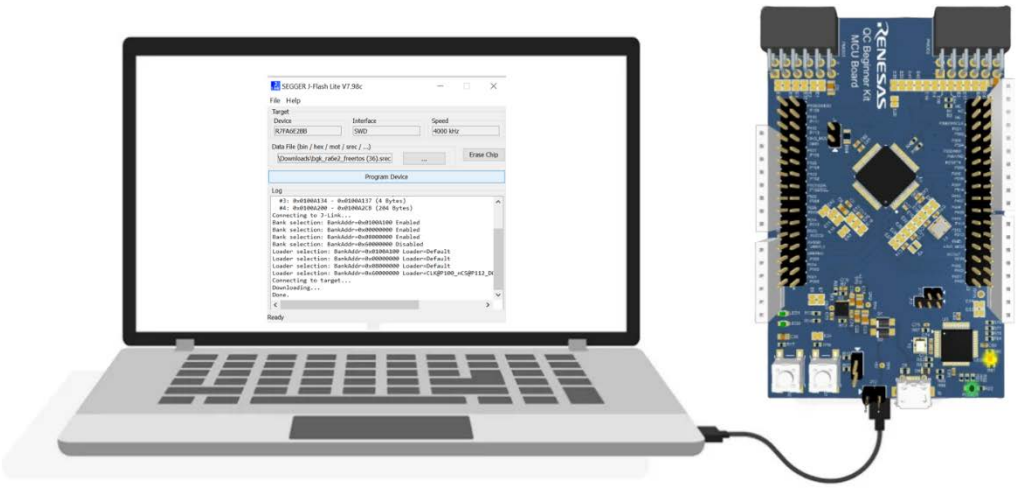
1.5.1 Flashing Using the SEGGER J-Flash Lite Utility

In this scenario, the user should have access to the MCU kit (BGK-RA6E2).

1. Download the QCS Application project `.srec` file, which can be found under the Debug folder inside the QCS Application project folder.

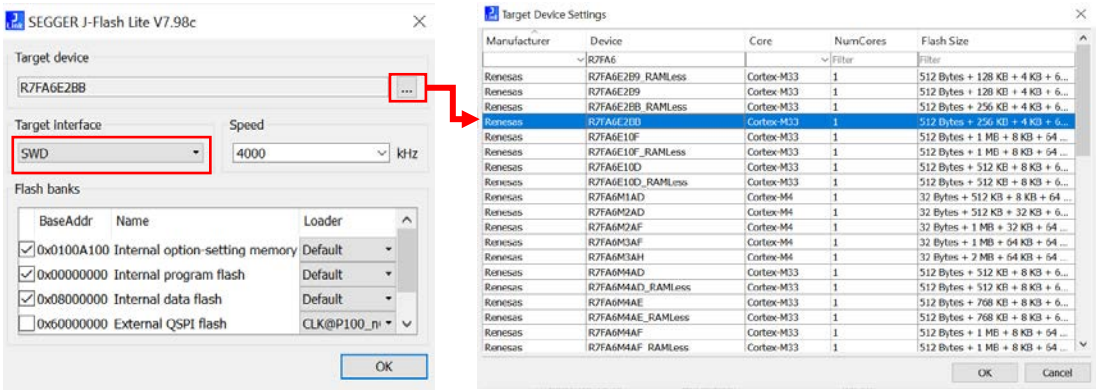


2. Connect the BGK-RA6E2 to a laptop using a USB cable.

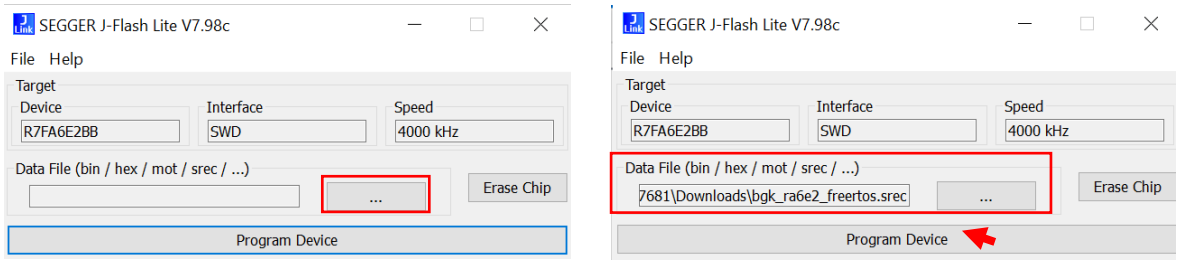


Note: Download the SEGGER J-Link software to the local PC if it is not already installed. According to the device compatibility, choose the installer under J-Link Software and Documentation Pack. Renesas recommends installing the 64-bit installer.

1. Open SEGGER J-Flash Lite.
   a. Navigate to the **Program Files** on the PC.
   b. Open the **SEGGER – Jlink** folder.
   c. Launch `JFlashLite.exe`.
2. Select the Target Device.
   a. In the J-Flash Lite window, click on the **(…)** button next to the **Target Device** field.
   b. A new window appears; the user can select the manufacturer and device.
   c. For this project, the **RA6E2 MCU** is used, search for the part number **R7FA6E2BB**.
   d. Select the target device and click **OK**.
   e. Ensure the target interface is set to **SWD**.
   f. Click **OK**.

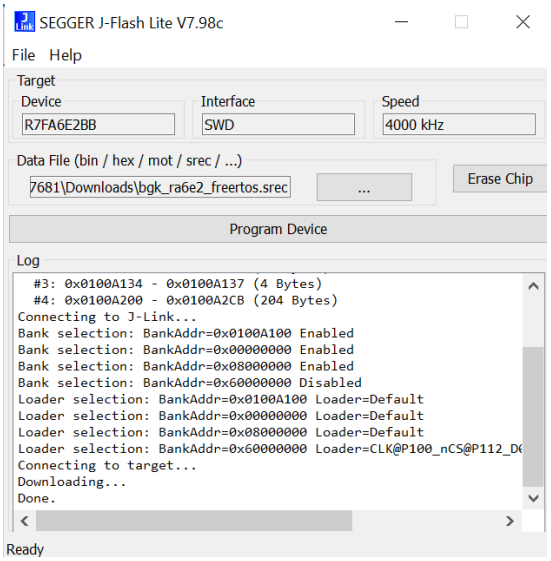


3. Import the `.srec` file.
   a. In the main J-Flash Lite window, locate the `Data File (bin / Hex / mot / srec / …)` section.
   b. Click on the **(…)** button to import the `.srec` file.
   c. Select the `.srec` file that was downloaded by following the steps in the Quick Start procedure.



4. Program the Device.
   a. Click on the **Program Device** button.
   b. A prompt can appear asking if an update to the latest firmware version is required. Select **No**.
   c. The code now flashes to the MCU.

d. After the process is complete, the log section of the screen displays **Done**.



e. Observe the output, LED 1 and LED 2 blinking in a pattern.

1.5.2 QCS Direct Debugging

In this method, the user should have access to the MCU kit (BGK-RA6E2).

Refer to the QCS Direct Debugging user guide from the QuickConnect landing page (www.renesas.com/quickconnect) on how to download and debug the QCS Application onto an MCU Kit (BGK-RA6E2).

1.5.3 QCS Remote Debugging

In this method, the user does not have access to the MCU kit but wishes to validate the QCS application output; they can do so by connecting to the Renesas managed board farm deployed globally.

Refer to the QCS Remote Debugging user guide from the QuickConnect landing page (www.renesas.com/quickconnect) on how to download and debug the QCS Application onto an MCU Kit (BGK-RA6E2).

1.6 Customize the QCS Application

Users can customize the application source files of their QCS application projects found under `/src` directory inside their project.

In the sample Blinky project created, the user can control which LED should be enabled by completing the following code customization steps.

1. To enable/disable a specific LED, comment or uncomment the following macros in the `main_application.h` file. Lines marked with a **+** symbol indicate the additions made. Users need to delete the + symbol if they copy and paste the below code snippet.

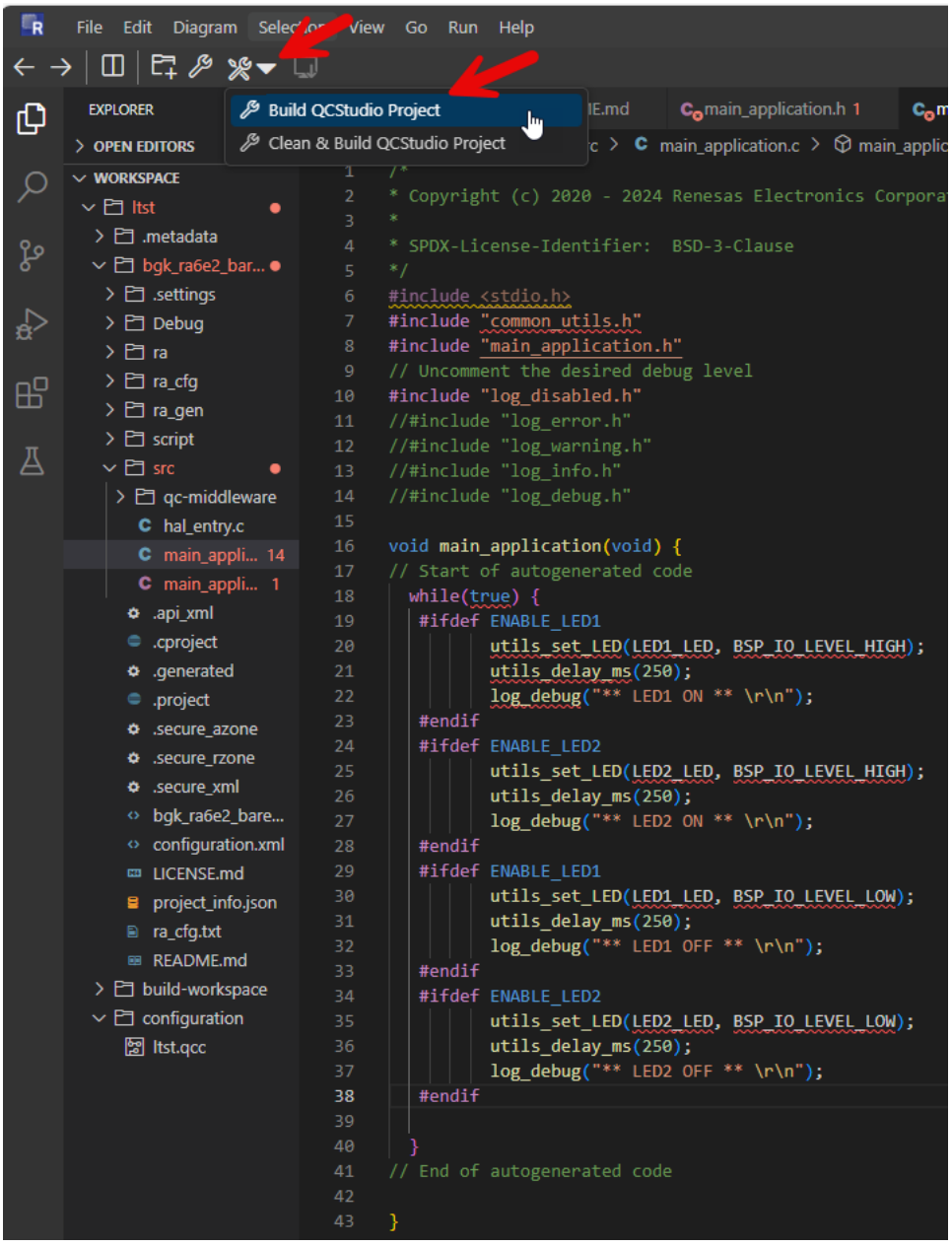
```
#ifndef __MAIN_A#PPLICATION_H
#define __MAIN_APPLICATION_H

#include "common_utils.h"
+ #define ENABLE_LED1   // enable or disable the code for LED1 using this macro

+ #define ENABLE_LED2     // enable or disable the code for LED2 using this macro
/* Function declaration */
void main_application(void);

#endif /* __MAIN_APPLICATION_H */
```

2. After defining the macros, the user should update the `main_application.c` file with the following lines.

   Note: Users must delete the + symbol when they copy and paste the following code snippet.

```
void main_application(void) {

// Start of autogenerated code
  while(true)
  {

+   #ifdef ENABLE_LED1
          utils_set_LED(LED1_LED, BSP_IO_LEVEL_HIGH);
          utils_delay_ms(250);
+         log_debug("** LED1 ON ** \r\n");
+   #endif
+   #ifdef ENABLE_LED2
          utils_set_LED(LED2_LED, BSP_IO_LEVEL_HIGH);
          utils_delay_ms(250);
+         log_debug("** LED2 ON ** \r\n");
+   #endif
+   #ifdef ENABLE_LED1
          utils_set_LED(LED1_LED, BSP_IO_LEVEL_LOW);
          utils_delay_ms(250);
+         log_debug("** LED1 OFF ** \r\n");
+   #endif
+   #ifdef ENABLE_LED2
          utils_set_LED(LED2_LED, BSP_IO_LEVEL_LOW);
          utils_delay_ms(250);
+         log_debug("** LED2 OFF ** \r\n");
+   #endif

  }
// End of autogenerated code

}
```

3. Click on the dropdown menu and click the **Build QCStudio Project** option to build the application project with the new updates.



4. Now the application project is built with the updated code customization, and it is ready for testing. Next, follow the steps in the section Download and Debug QCS Application.

2. Revision History

Revision	Date	Description
1.00	Jan 15, 2026	Initial release.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES (“RENESAS”) PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.