

# QuickConnect GitHub Extension

## Introduction

The GitHub Extension for Renesas QuickConnect Studio (QCS) Platform enables developers to seamlessly integrate their QuickConnect projects with GitHub repositories directly inside the QCS workspace. It streamlines collaboration, version control, and CI/CD automation. Users can push, pull, and sync QuickConnect projects with GitHub without leaving the QCS Workspace window.

### Benefits of GitHub Support in QCS

- Seamless collaboration – Developers can push, pull, and sync QuickConnect projects with GitHub without leaving their QCS Development workspace window.
- Time savings – Eliminates manual Git setup and accelerates project onboarding.
- Unified developer experience – Brings QCS project design, coding, and collaboration into a single platform.
- Scalability for teams and education – Distributed teams and students can manage shared QuickConnect projects in real-time without setup complexity.

### Target Audience

- Embedded developers – Enables one-click GitHub integration inside QCS without requiring users to switch between tools.
- Distributed teams – Offers collaborative project management across geographies. Real-time sync and pull request workflows inside QCS reduce friction for distributed teams working on shared QCS projects.
- Educators and students – Offers an easy-to-use GitHub integration that requires no advanced Git skills, promoting quicker project collaboration and submissions.

## Content

<b>1. Getting Started</b> .....	<b>2</b>
1.1 Prerequisites.....	2
1.2 Signing into GitHub.....	2
1.3 Publish to GitHub.....	3
1.4 Commit Changes.....	4
1.5 Clone a GitHub Repository.....	5
1.6 Signing Out of GitHub.....	6
<b>2. Known Limitations</b> .....	<b>6</b>
<b>3. Revision History</b> .....	<b>6</b>

# 1. Getting Started

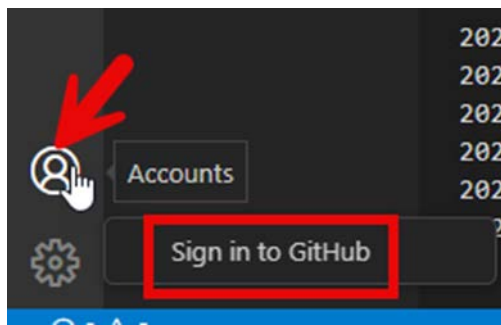
## 1.1 Prerequisites

- Follow the Renesas QCS Platform user manual (*QuickConnect IoT Manual*) from the QCS landing [page](#) to create and build a QCS project.
- GitHub account with repository access
- Stable internet connection

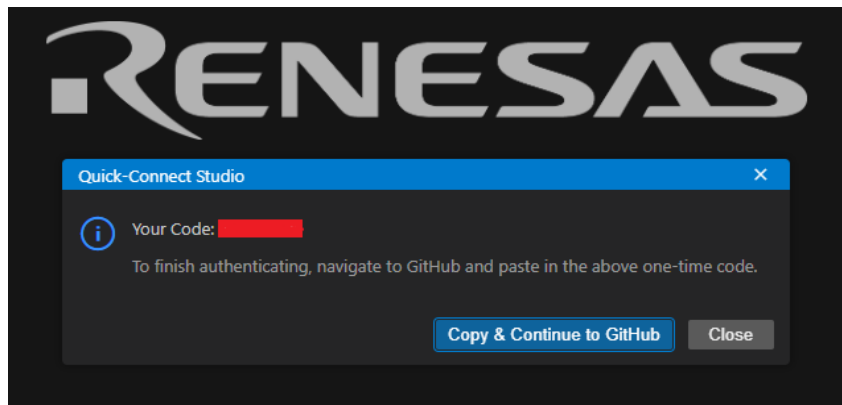
## 1.2 Signing into GitHub

At this stage, follow the instructions in the *QCS User Manual* and become familiar with the QCS platform and its capabilities. If not, it is strongly advised to review the QCS user manual and get familiar with the QCS platform before proceeding further.

1. Launch the QCS workspace in your browser and create a QCS application project.
2. Click on the **Accounts** icon; next, click on the **Sign in to GitHub** pop-up dialog window.



3. A pop-up dialog window appears, displaying an OTA. Copy the OTP and click on the **Copy and Continue to GitHub** icon. A new tab opens in the browser. Follow the instructions, and complete the sign-in process.



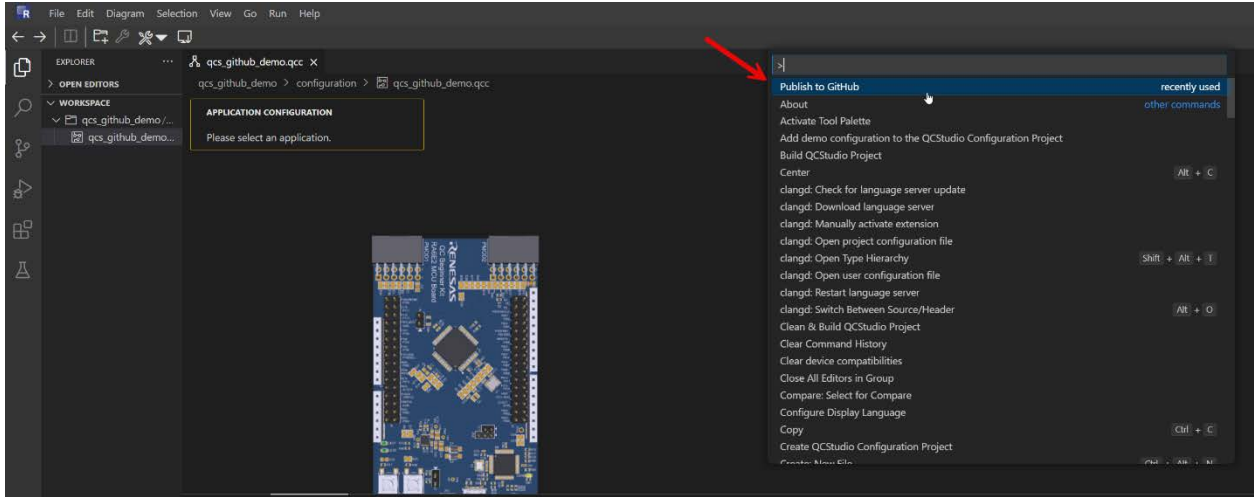
4. After a successful GitHub sign-in, a pop-up window appears on the QCS workspace in the bottom right. The GitHub sign-in process is now complete.

### 1.3 Publish to GitHub

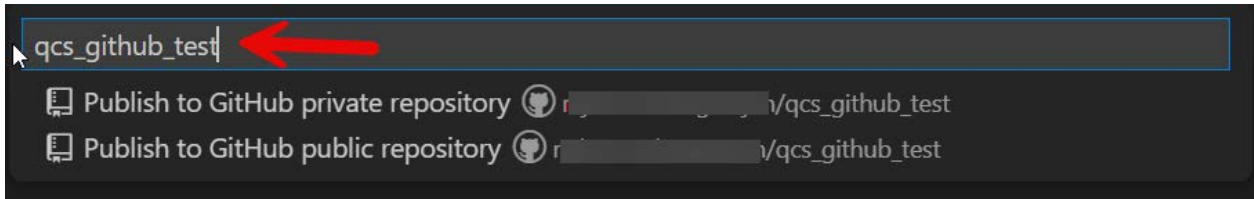
The Publish to GitHub feature allows for publishing a local QCS project to a GitHub repository directly from the QCS workspace window.

Before continuing, complete the steps mentioned in the Prerequisites and Signing into GitHub sections.

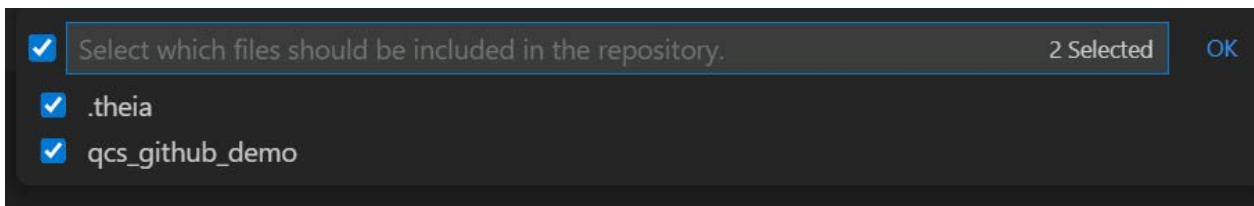
1. The QCS project is ready to be published. Press **Ctrl + Shift + P** on the keyboard (for a Windows PC) to open the Command Palette; next, type and select **Publish to GitHub**.



2. Enter the workspace for publishing the QCS project and select whether the repository to be published is public or private.



3. A pop-up window shows the list of projects that should be selected for publishing. Select the required files to be published and press the **OK** button.



- a. After a successful publish, the following message appears on the bottom right of the QCS User Workspace.

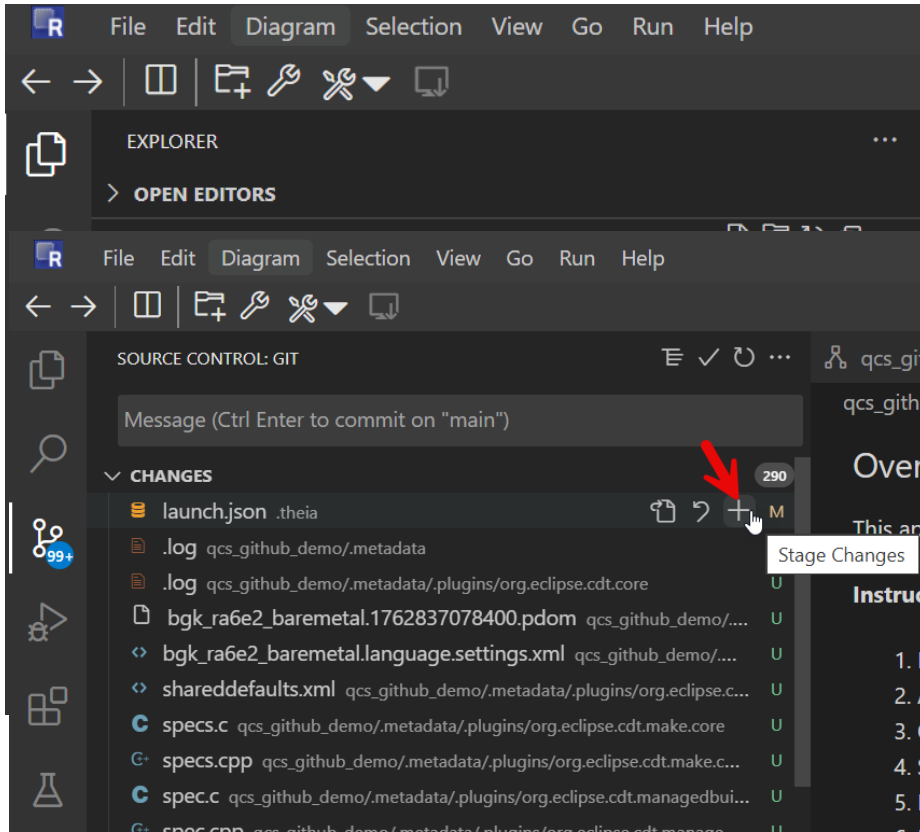


## 1.4 Commit Changes

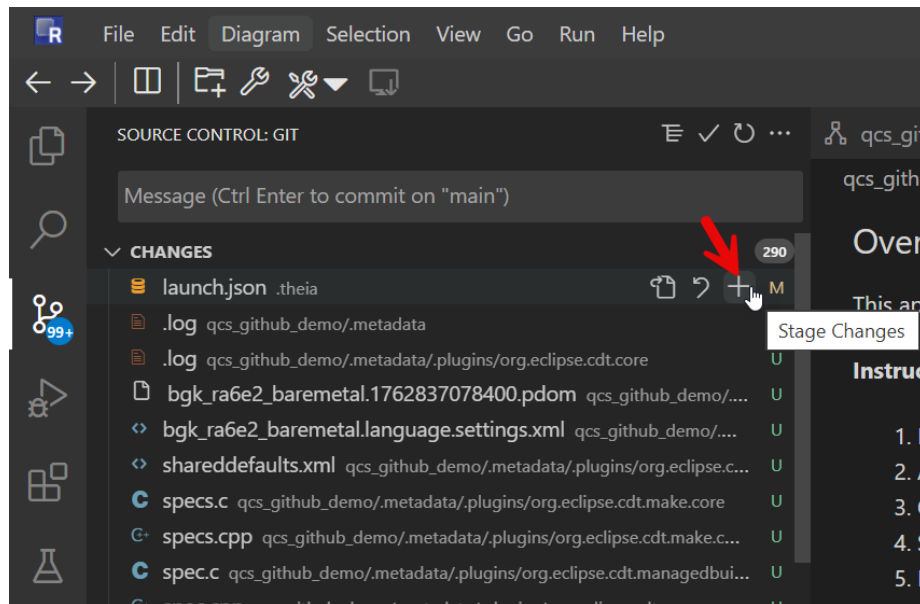
The Commit Changes feature allows for recording snapshots of the project state in the Git repository and creating a persistent, organized history of work.

Before continuing, complete the procedures from the previous sections.

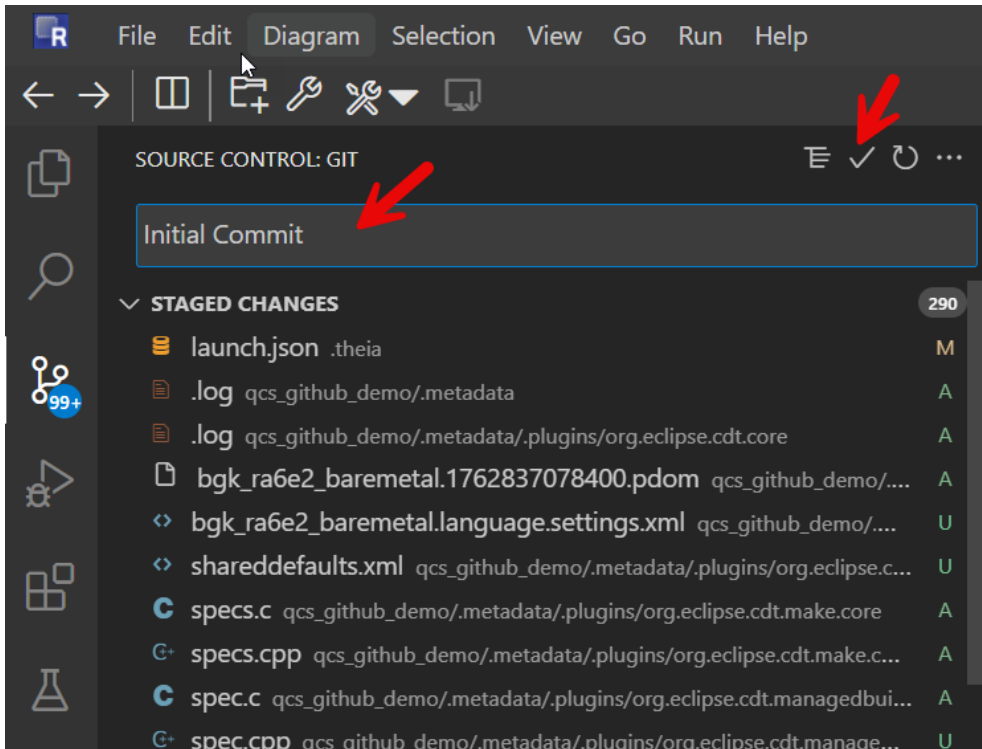
1. Go to the **Source Control Panel** (the icon on the left sidebar) as shown in the following image.



2. Stage your changes by clicking on the + icon that aligns with the file to be staged.



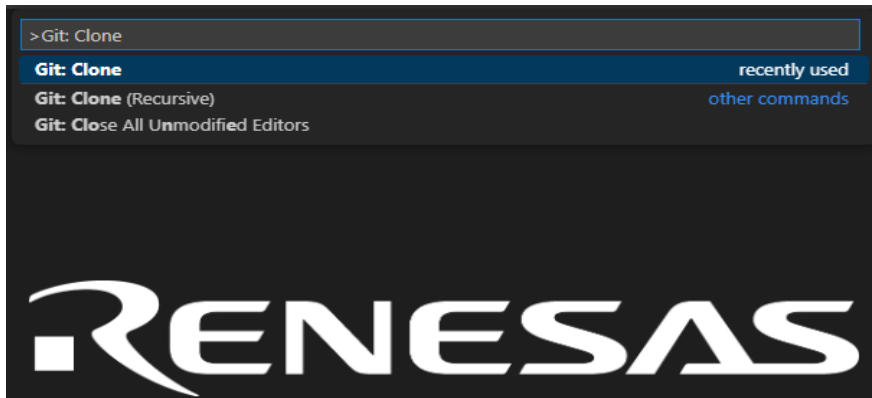
3. Add a commit message and click the ✓ Commit icon as in the following image.



## 1.5 Clone a GitHub Repository

The git clone feature allows users to copy (clone) a repository from a remote server to their local machine.

1. Press **Ctrl + Shift + P** on the keyboard (for a Windows PC) to open the Command Palette; next, type **Git: Clone**.

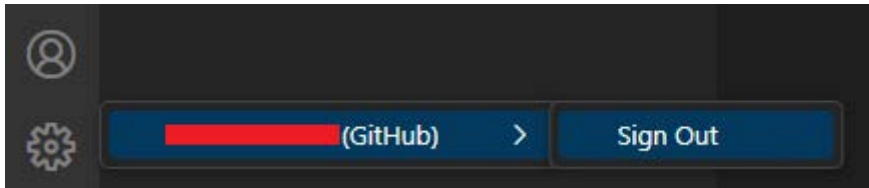


2. After selecting **Git: Clone**, click the option **Clone from GitHub** when it appears.
3. Next, a window displays a list of repositories associated with the current GitHub account.
4. Choose a repository and clone it by selecting a local folder.

## 1.6 Signing Out of GitHub

Complete the following steps to successfully sign out of GitHub.

1. Click on the **Account** menu; next, click on the **User** icon or **Account Name**. Select **Sign Out** from the continuing menu.



2. Confirm the signing out: a prompt appears confirming the selection to sign out. Click **Yes** to proceed.

## 2. Known Limitations

The following are known limitations with the QuickConnect GitHub Extension.

- Does not support a GitHub Enterprise Account.
- Users cannot commit to a repository inside a GitHub Organization.

## 3. Revision History

Revision	Date	Description
1.00	Feb 5, 2026	Initial release.

## 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](http://www.renesas.com)

### Trademarks

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

### 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/](http://www.renesas.com/contact-us/).