Renesas RA Family

GUIX "Hello World" for EK-RA6M3G

Introduction

This application note describes the process of creating a simple two-screen GUI using Azure RTOS GUIX Studio for the EK-RA6M3G kit. This application demonstrates how easily a user can create and configure a new application using the Renesas Flexible Software Package (FSP).

The Renesas Flexible Software Package includes Azure RTOS ThreadX® real-time operating system, the Azure RTOS GUIX library and hardware drivers unified under a single robust software package. This powerful suite of tools provides a comprehensive integrated framework for rapid development of complex embedded applications.

The Hello World application was developed under e^2 studio using the Renesas Flexible Software Package.

Required Resources

Development tools and software

- e^2 studio IDE 2021-04 or greater
- Renesas Flexible Software Package (FSP) v3.1.0 or greater
- Azure RTOS GUIX Studio V6.1.7.0 or greater

Hardware

- Renesas EK-RA6M3G kit (RA6M3 MCU Group)
  (https://www.renesas.com/us/en/products/software-tools/boards-and-kits/eval-kits/ek-ra6m3g.html)

Reference Manuals

- RA Flexible Software Package Documentation Release v3.1.0
- Azure RTOS GUIX and GUIX Studio v6.1.7.0
- Renesas RA6M3 Group User's Manual Rev.1.00
- EK-RA6M3G-v1.0 Schematics

Provided Software Files

- A `Source.zip` folder that has a `touch_ft5x06` folder and 4 `*.c` files inside
- `hal_entry.c`, `system_thread_entry.c`, `touch_thread_entry.c`, `windows_handler.c`

Purpose

This document will guide you through the setup of an Azure RTOS GUIX touch screen interface Hello World application in e^2 studio. This document will show how to configure the drivers and library included with the FSP. These will allow you to set up the GLCD Controller, the touch screen driver, and semaphores to communicate with application tasks. It also shows the steps necessary to create a simple GUI interface using the Azure RTOS GUIX Studio editor. In addition, this app note will also cover project setup along with basic debugging operations. When it is running, the application will respond to touchscreen actions, presenting a basic graphical user interface (GUI).

Intended Audience

The intended audience is users who want to design GUI applications.

Contents

1.   Download and Installing Tools

1.1   Overview

In this section you will copy materials to your PC and install v2021-04/FSP v3.1.0 and Azure RTOS GUIX studio v6.1.7.0.

1.2   Procedural Steps

1. If you already have e2 studio v2021-04 with FSP v3.1.0 or later installed, you can skip this step. Otherwise, you can download it from https://www.renesas.com/us/en/software-tool/flexible-software-package-fsp.

    Detailed installation instructions for the e2 studio and the FSP are available on the Renesas website https://www.renesas.com/fsp. Review the release notes for e^2 studio to ensure that the e2 studio version supports the selected FSP version. The starting version of the installer includes all features of the RA MCUs.

2. You can get Azure RTOS GUIX Studio V6.1.7.0 or greater from this link.

    If it goes well, you will see the window in the next step on the web browser.

    Note: Microsoft Store must be working on your PC to install Azure RTOS GUIX studio.

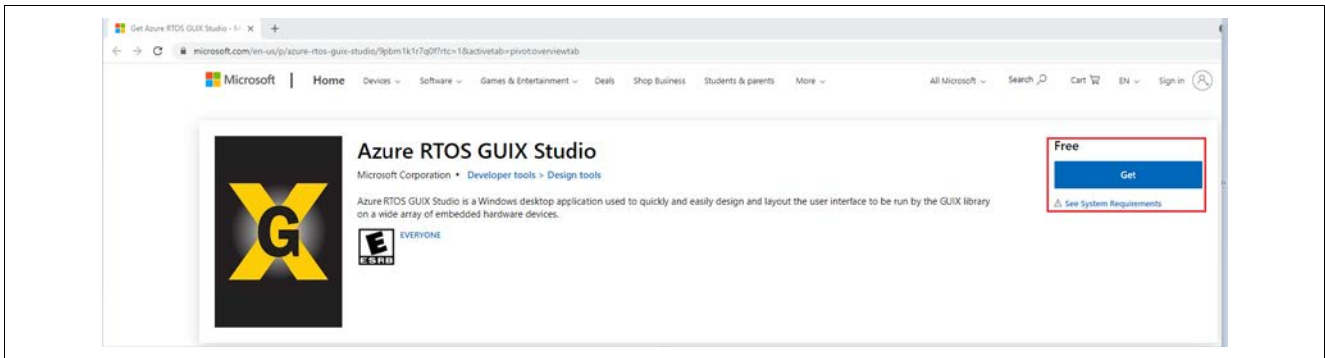3. Click **Get** to start installing Azure RTOS GUIX studio.

Figure 1.   Get Azure RTOS GUIX Studio

4. Click **Open Microsoft store** to continue installing Azure RTOS GUIX studio.

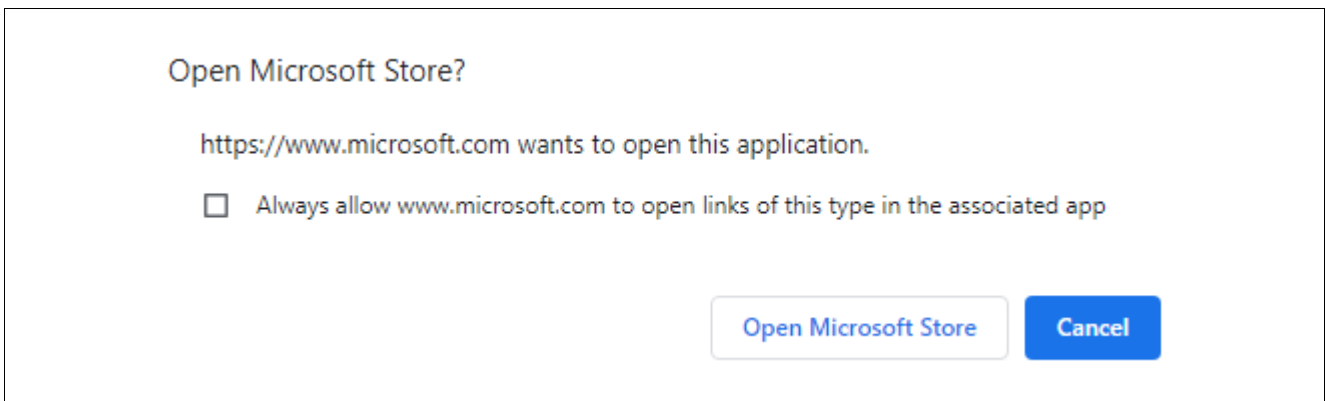**Figure 2.   Open Microsoft Store**

5. Click **Install** to continue. A window shows up to ask for a Microsoft account, which is seen in the next step.
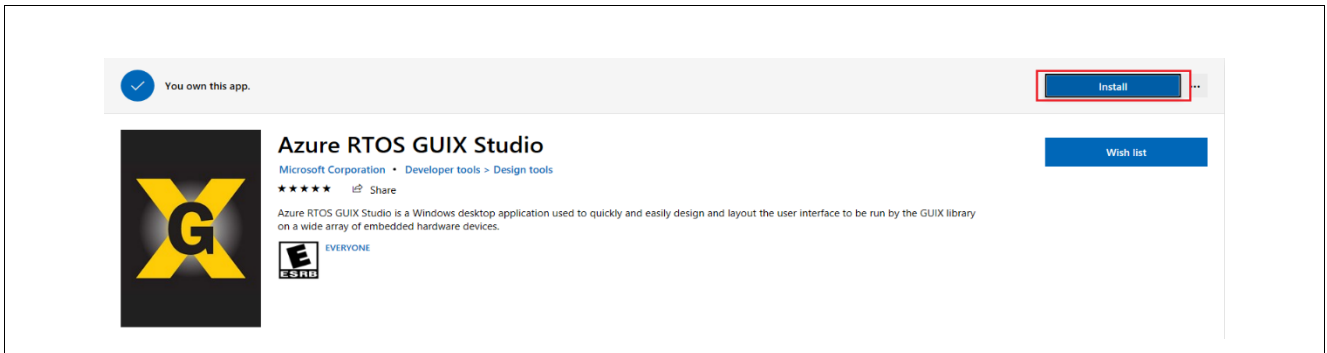

Figure 3.   Install

6. Ignore it by clicking **X** on the top-right to close this pop-up window and continue Azure RTOS GUIX studio installation.
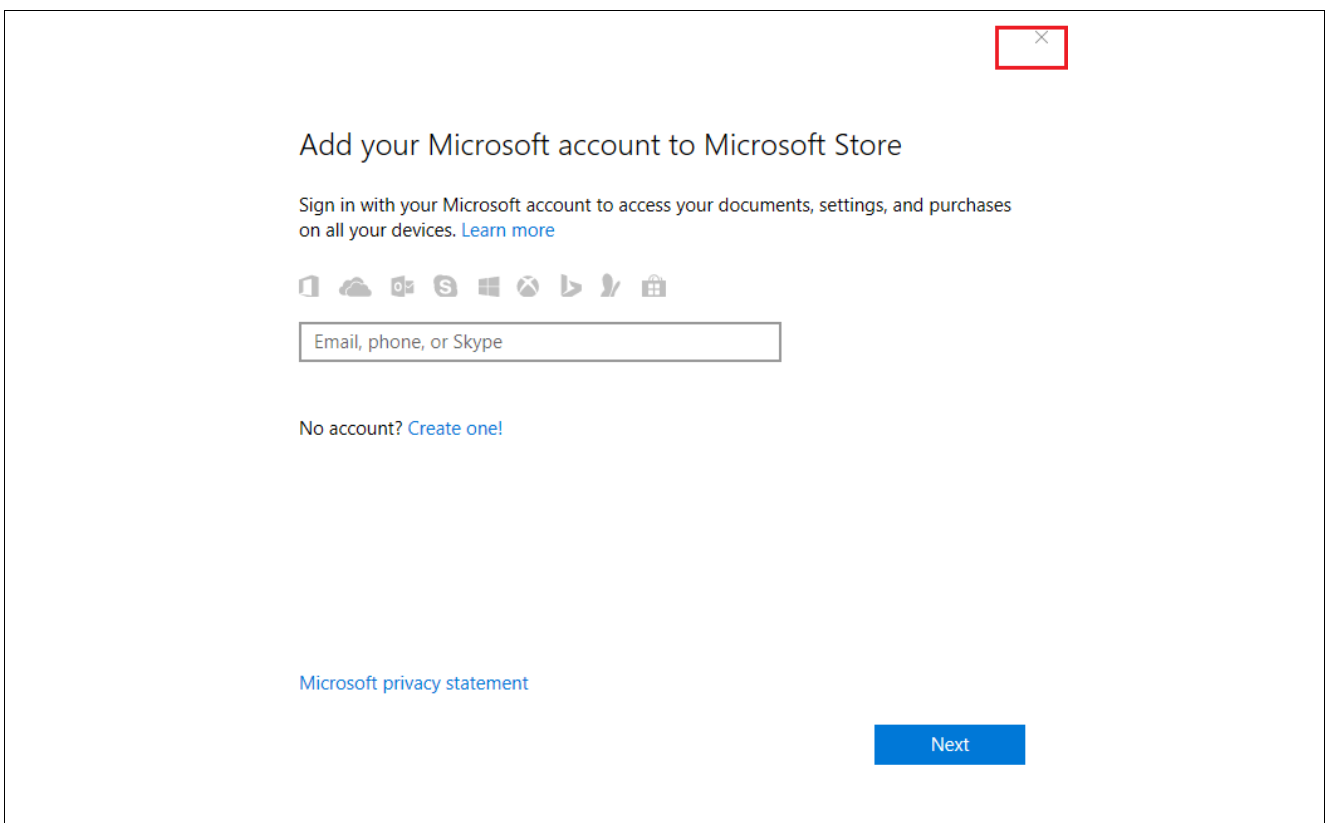

Figure 4.   Close Pop-Up

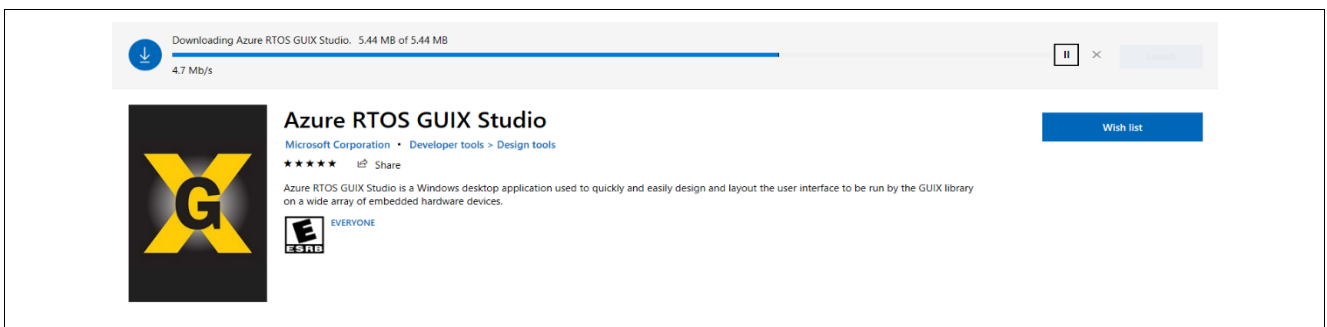7. It starts downloading and installing Azure RTOS GUIX studio.

Figure 5.   Downloading Azure RTOS GUIX Studio

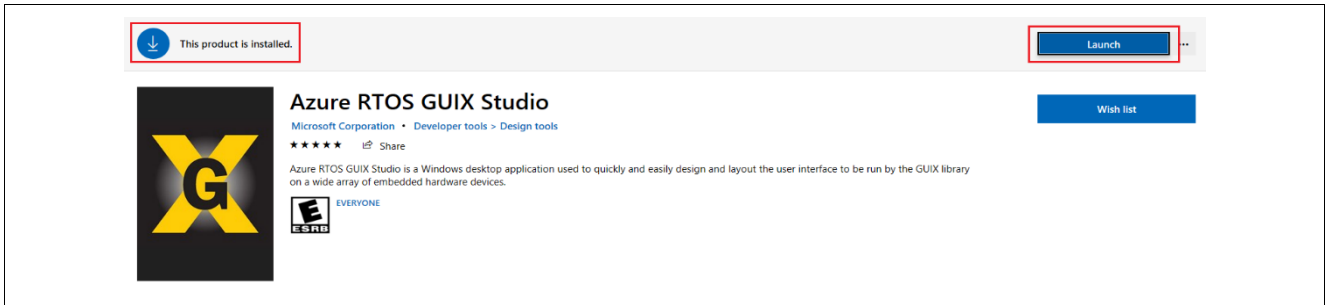8. Click **Launch** to launch Azure RTOS GUI studio.

Figure 6.   Launching Azure RTOS GUIX Studio

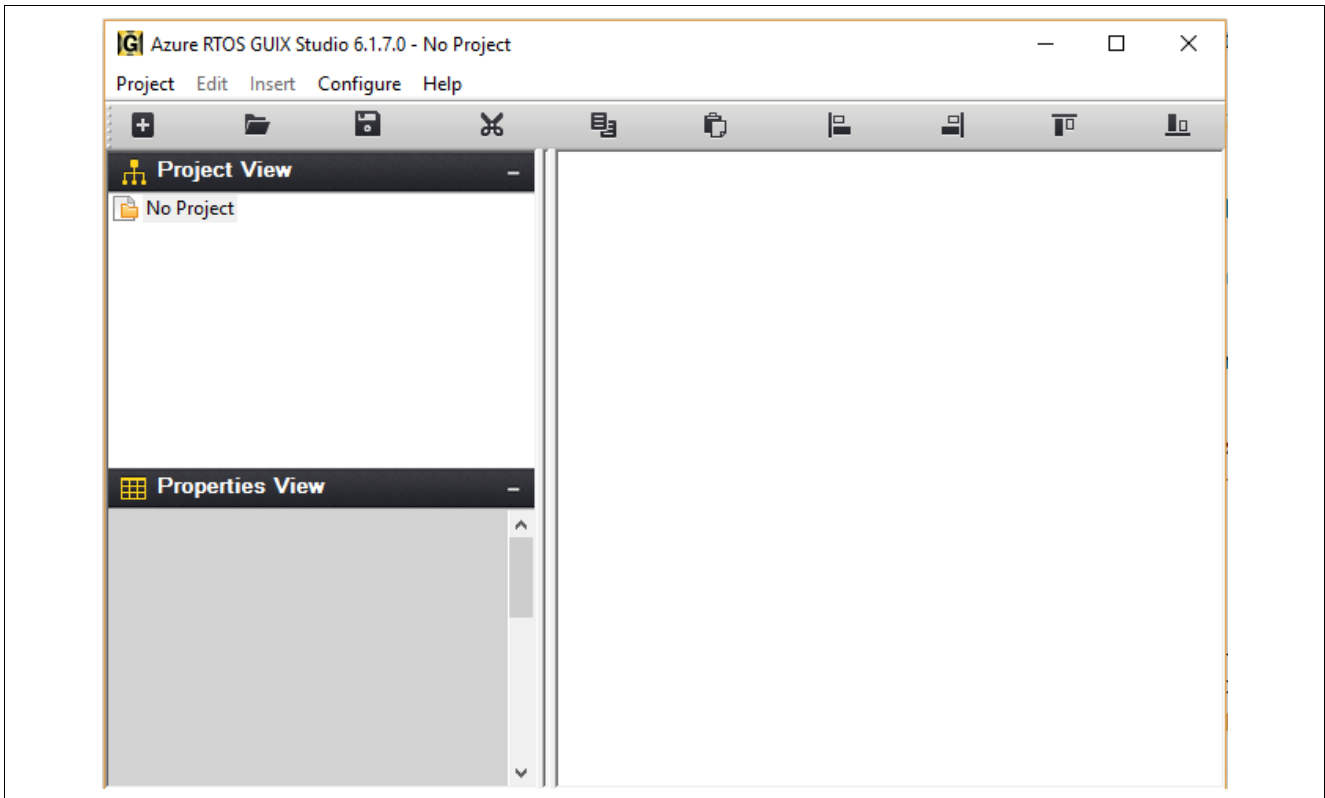9. Azure RTOS GUIX studio 6.1.7.0 launched.

Figure 7.   Azure RTOS GUIX Studio

10. Close Azure RTOS GUIX studio, for now, you will open it again later.

2.   Create Application Note Project and Enabled Backlight

2.1   Overview

In this section, you will create a project to which you will add pre-written source code and integrate it with a pre-created Azure RTOS GUIX studio project.

2.2   Procedural Steps

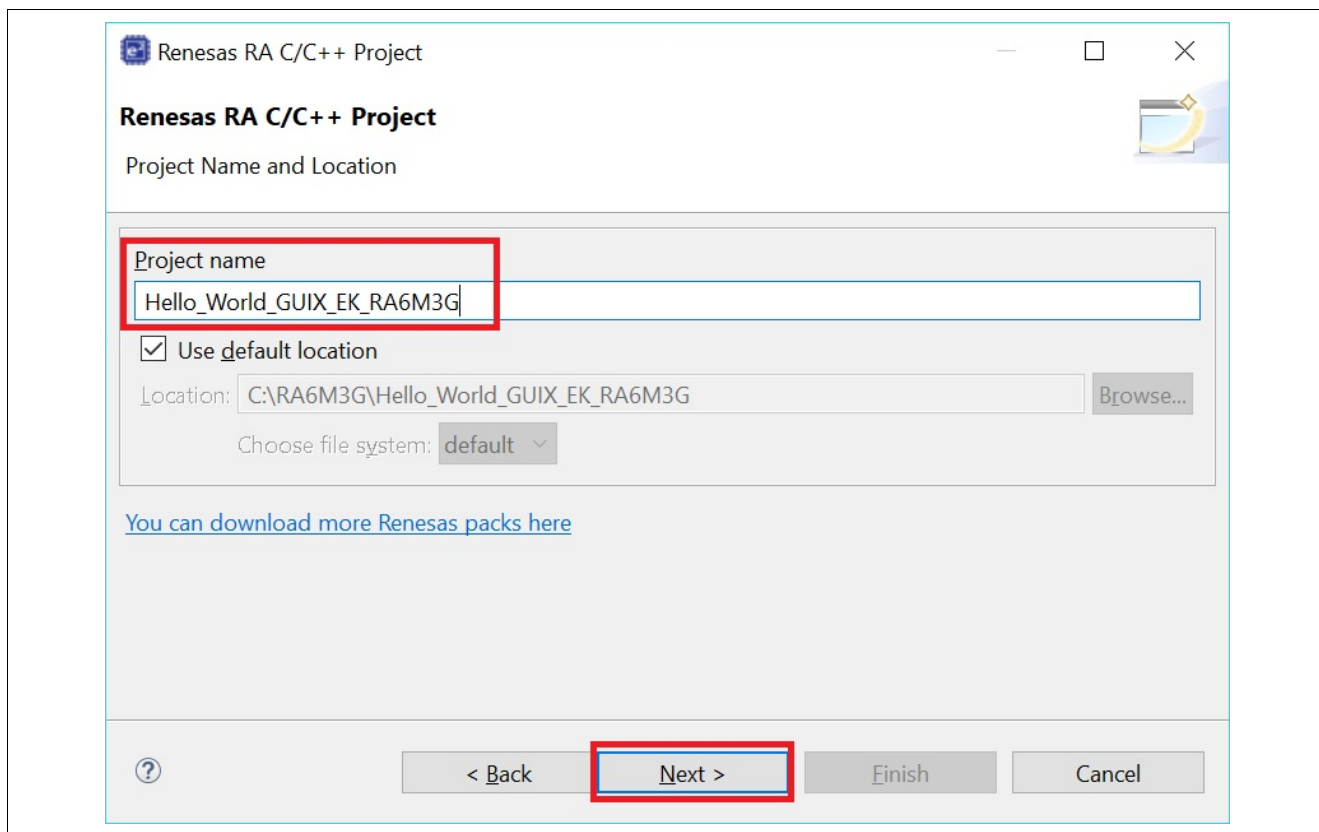1. Create a new **Renesas RA C/C++ project**. Name it **Hello_World_GUIX_EK_RA6M3G**.

Figure 8.   Create New Project
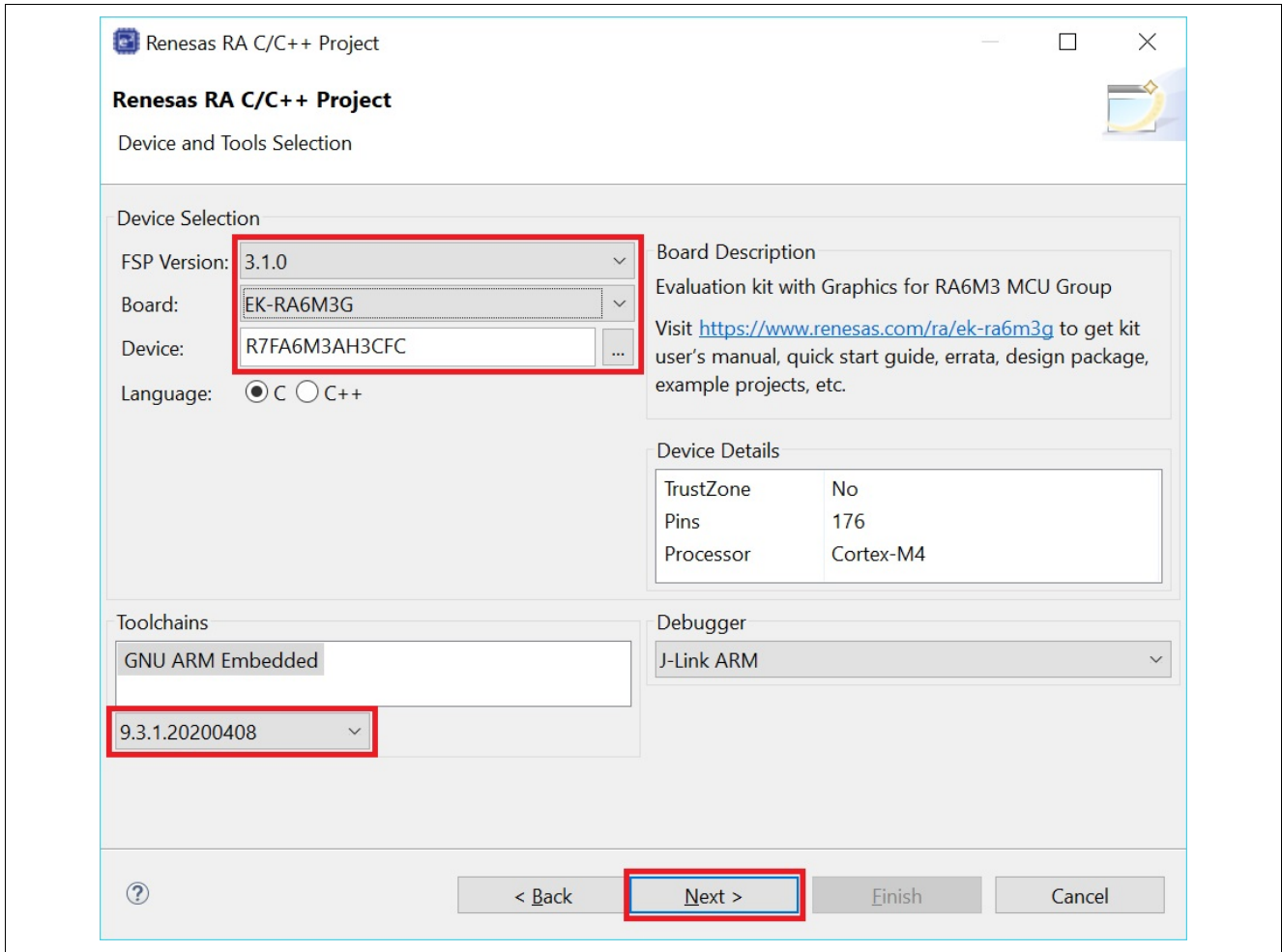
2.   Set board to EK-RA6M3.

Figure 9.   Set Board to EK-RA6M3G

3. Select **Azure RTOS ThreadX (v6.1.8+FSP.3.1.0)**.
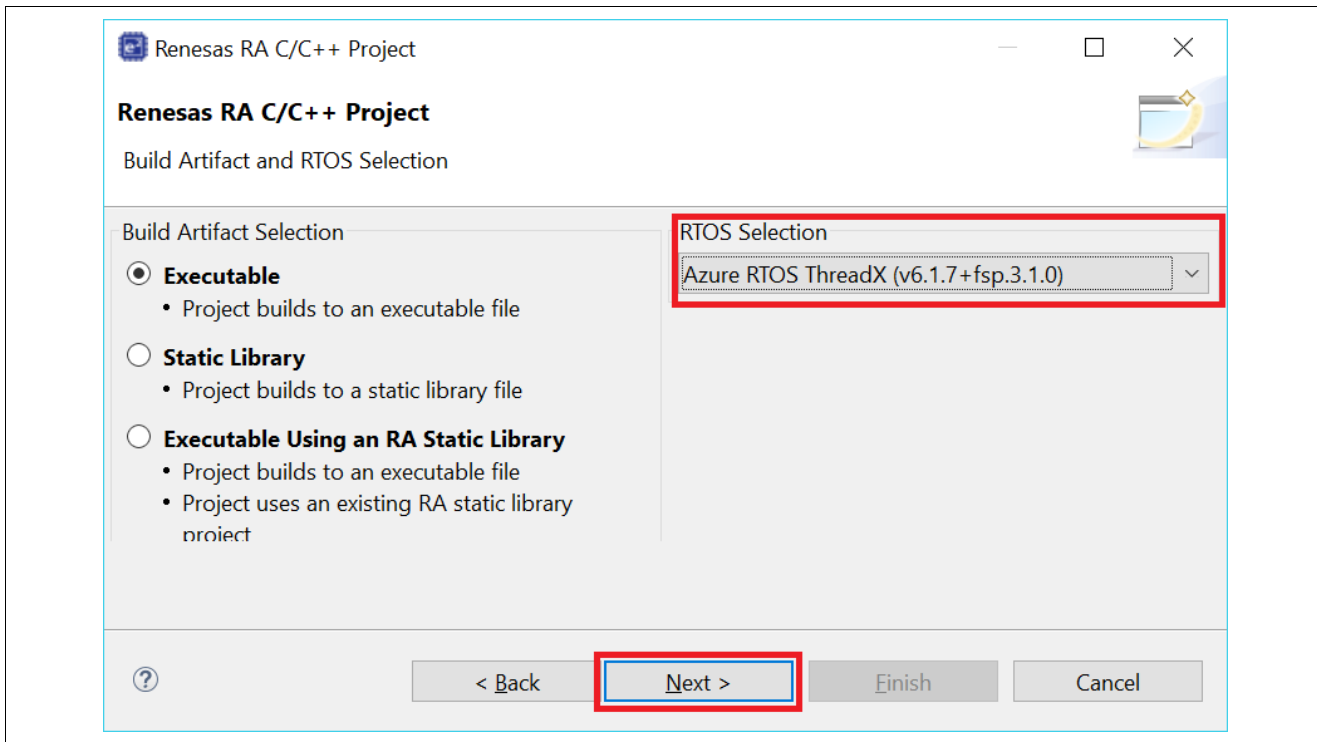

Figure 10. Select Azure RTOS ThreadX

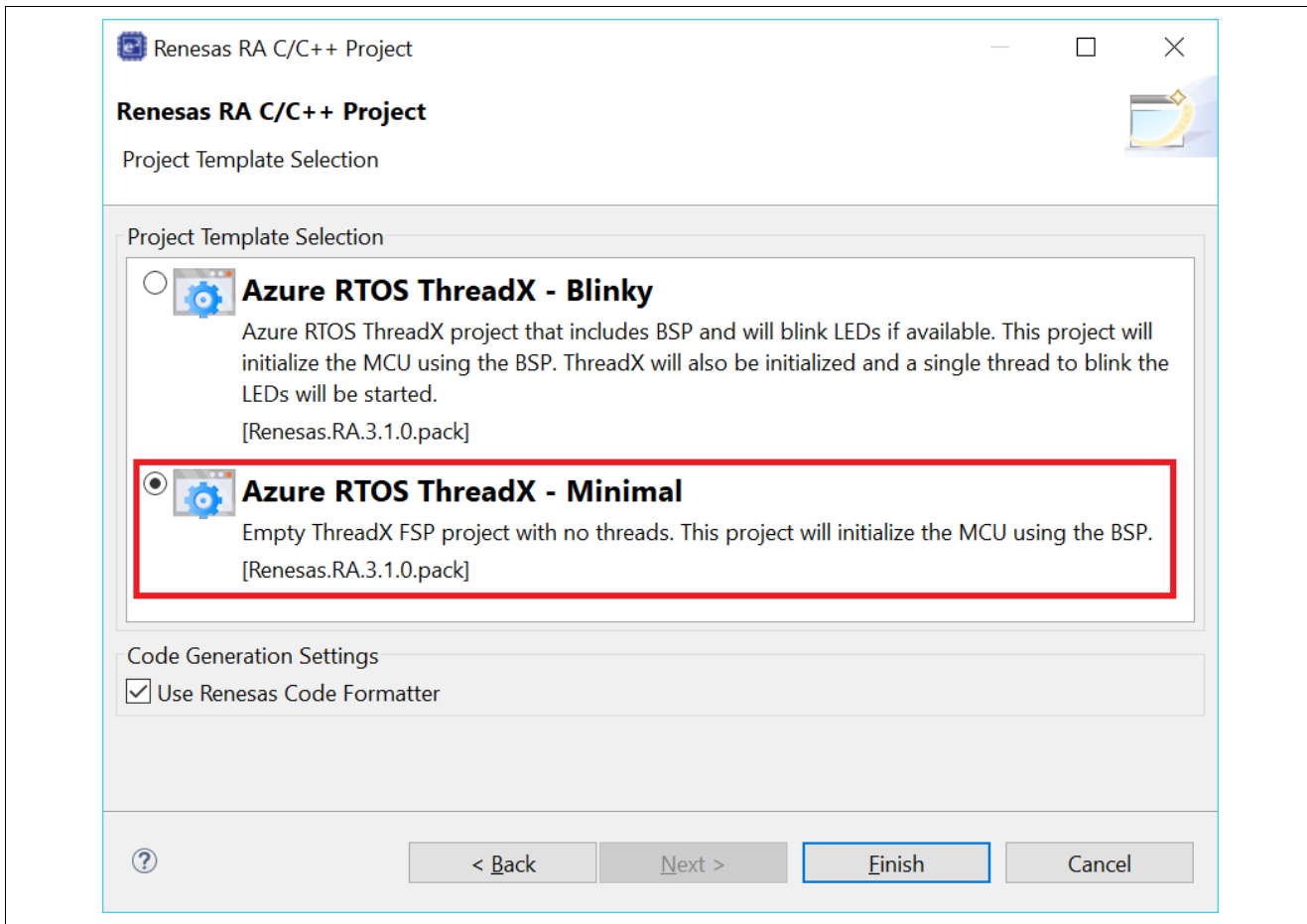4. Use Azure RTOS ThreadX-Minimal template.

Figure 11. Select Minimal template

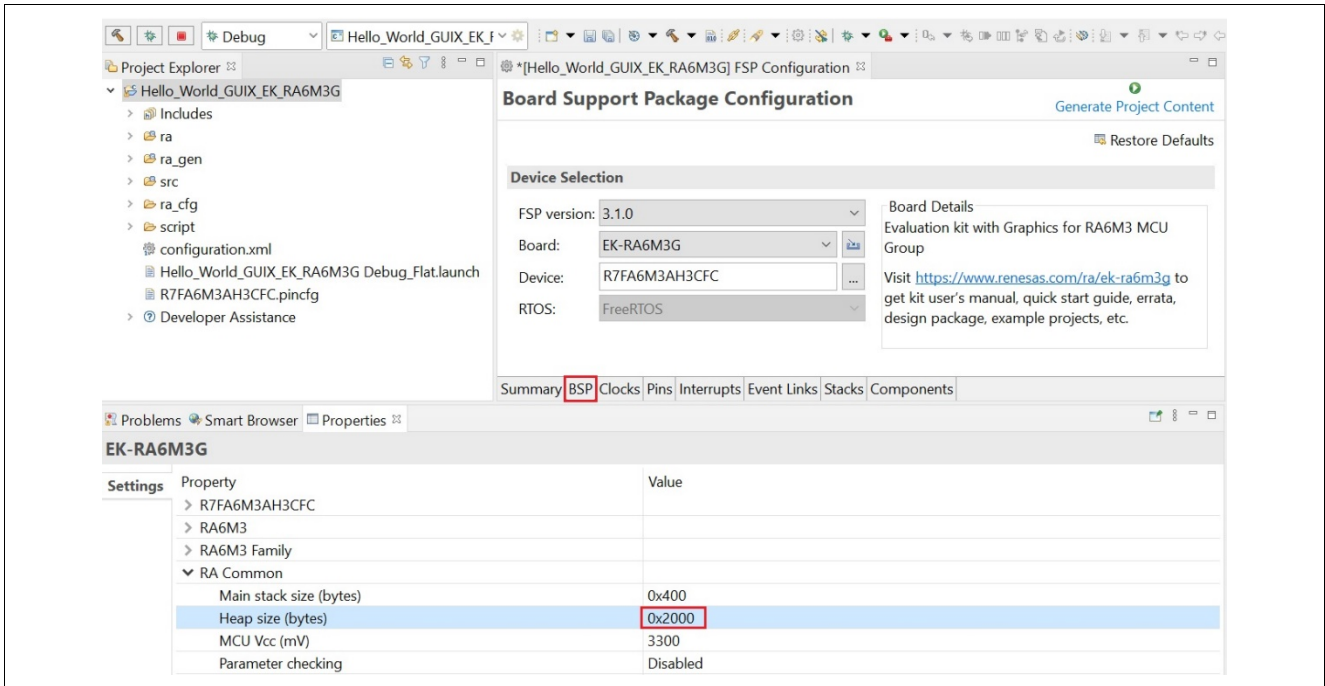5. Open the project configuration and go to the **BSP** tab. Change **Heap size (bytes)** to **0x2000**.

Figure 12.   Change Heap Size

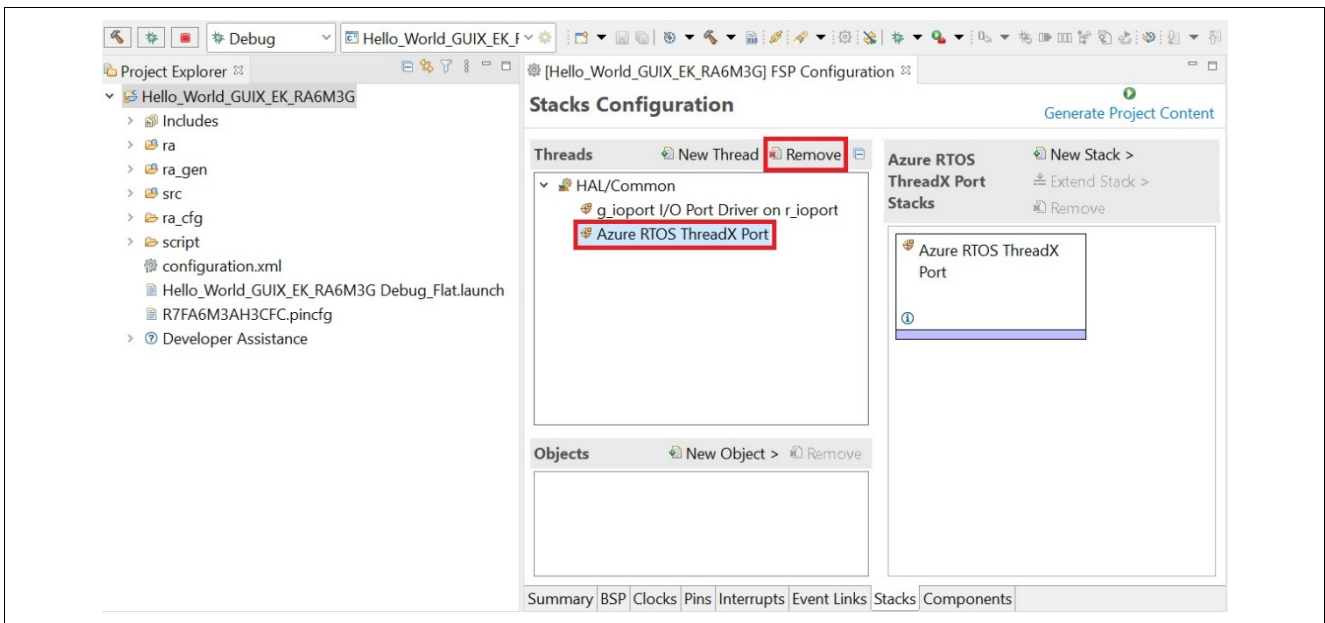6. You can remove **Azure RTOS ThreadX Port** if you want.

Figure 13.   Remove Azure RTOS ThreadX Port

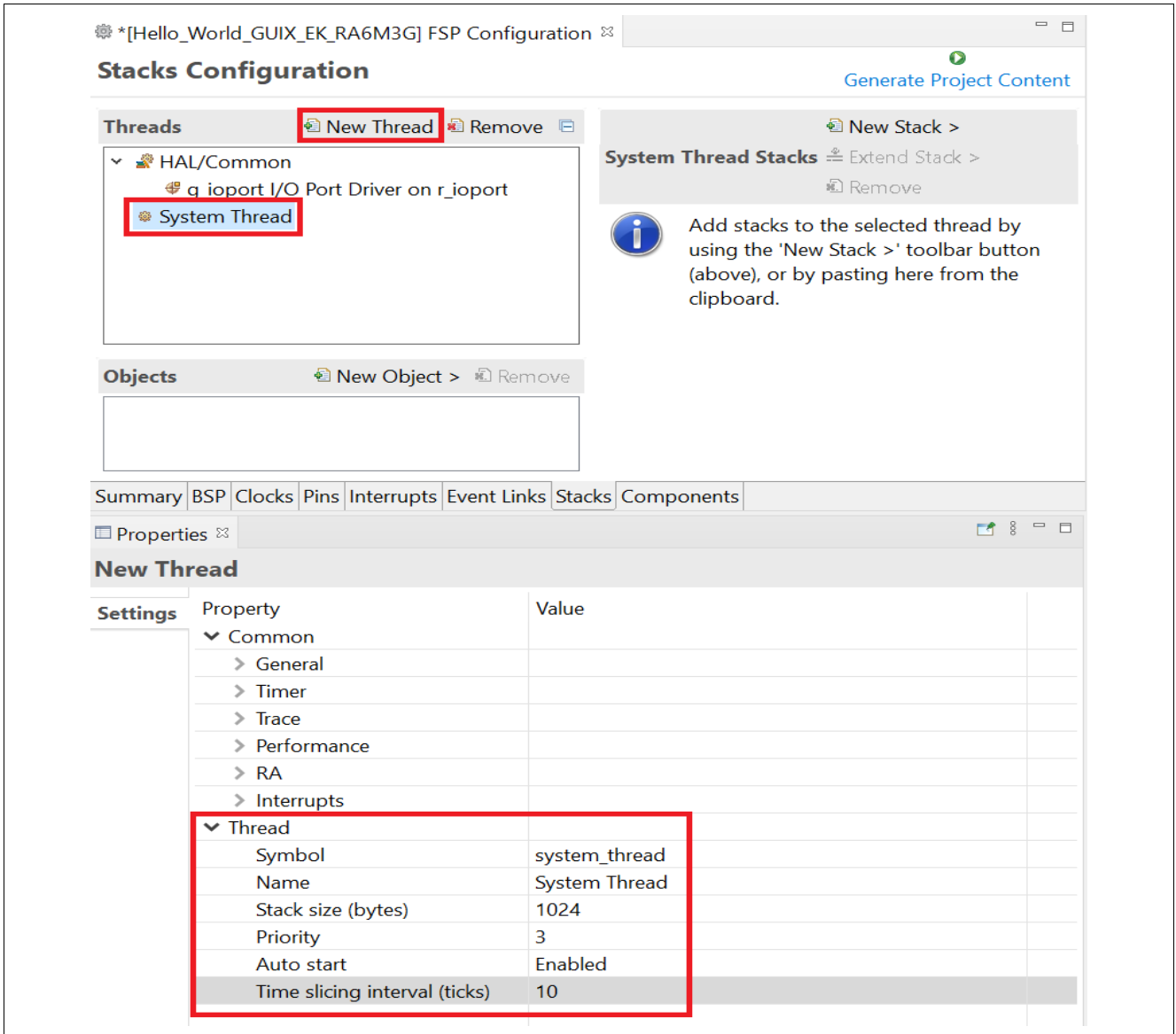7. Add a **new thread** and name it as **System Thread** with below settings.

Figure 14.   Add System Thread

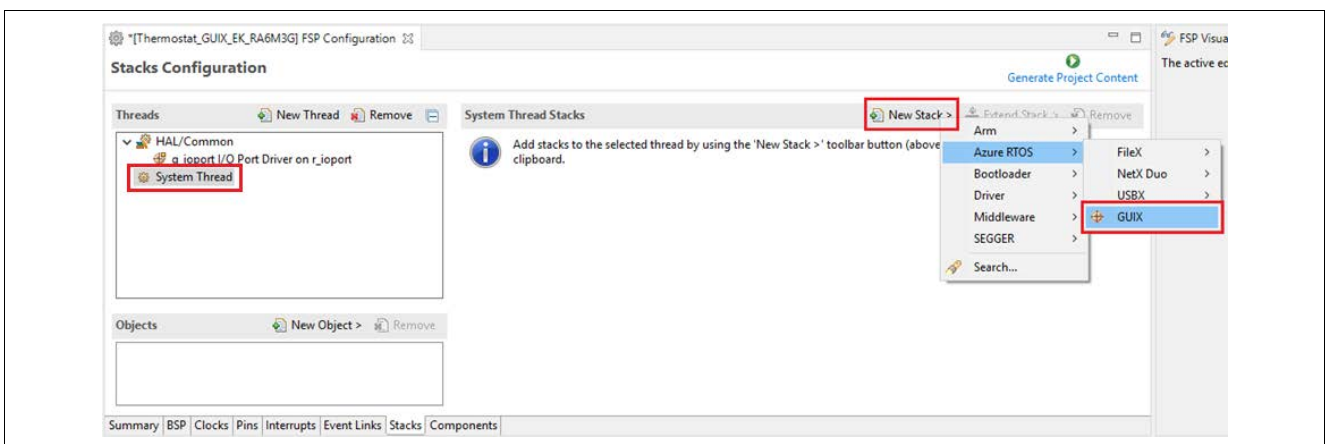8. Add Azure RTOS GUIX to system thread.

Figure 15.   Add Azure RTOS GUIX

9. Add **GUIX** framework for GLCDC configuration.

Figure 16. Add GUIX Framework for GLCDC Configuration

10. The **Display Rotation** option is also available for user but in this application, we will not use this option. Skip this setting and keep original setting **None**.

Figure 17.   Scren Orientation

11. Property settings for the **g_display0 Display Driver on r_glcdc**.

Figure 18.   Setting Properties for g_display0 Display Drive on r_glcdc

12. Configure the GLCD's **Graphics Layer 1 Properties** as shown.

Figure 19.   Graphics Layer 1 Properties

13. Configure the **Output** properties as shown.

Figure 20.   Output Properties

14. Configure the Hsync, Vsync, Data Enable pins, and clock as shown.

Figure 21. TCON Properties

15. In the **Pin Configuration**, first set **Mode** to **Disabled** for the pin P603 to disable LCD panel back light.

Figure 22. Disable LCD Back light

16. In the **Stacks** tab, select **System Thread** and add in a **New Stack > Driver > Timers > Timer Driver on r_gpt**.
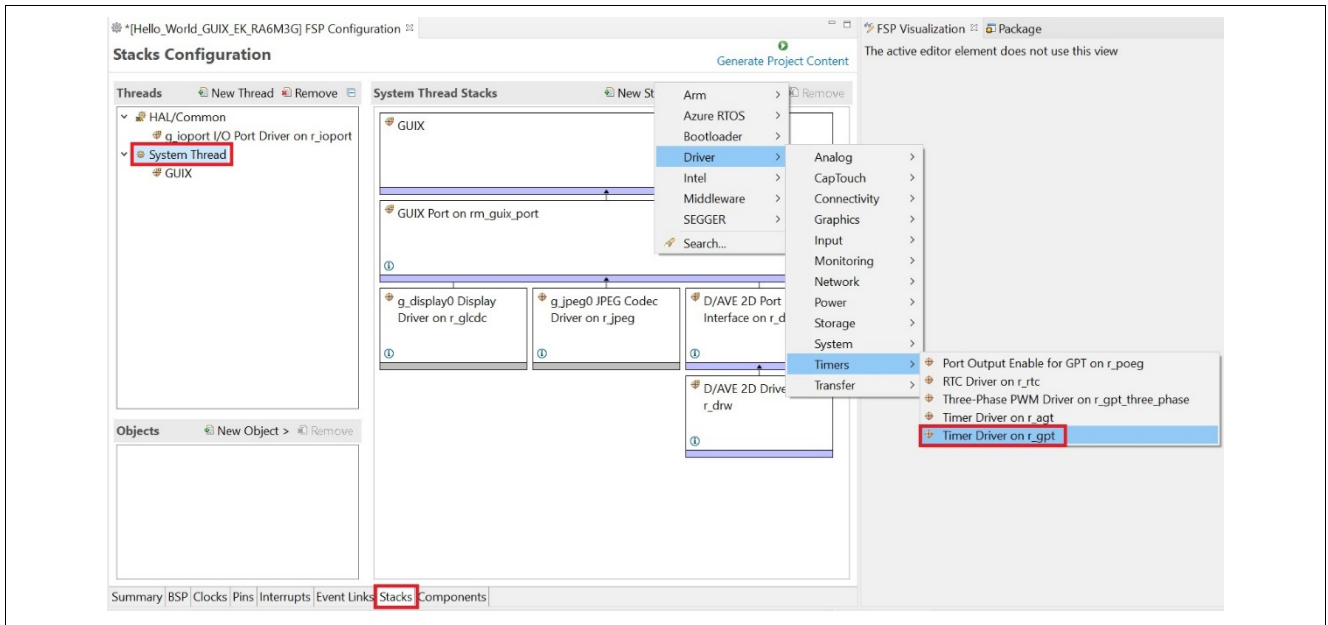
Figure 23. Add Timer Driver on r_gpt

17. Now, set and configure the properties of **Timer Driver on r_gpt**. Select **GTIOCA Output Enable**. Select **GTIOCA** and hit the arrow **=>** for selecting and setting properties of the pin P603 (follow the images below).
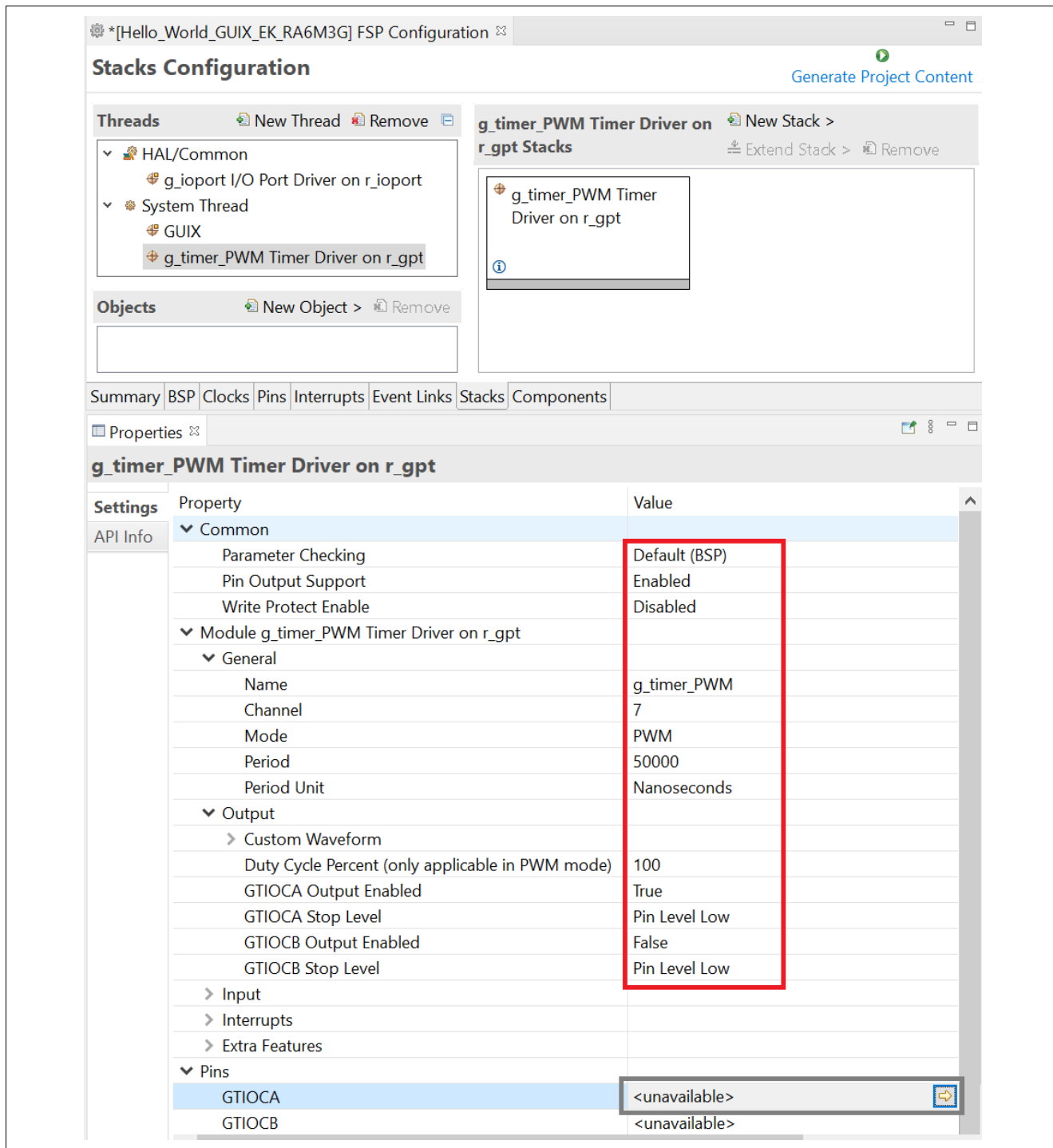
Figure 24. Properties of Timer Driver on r_gpt

18. Set and configure the pin P603 for the LCD_BLEN signal of the LCD panel.
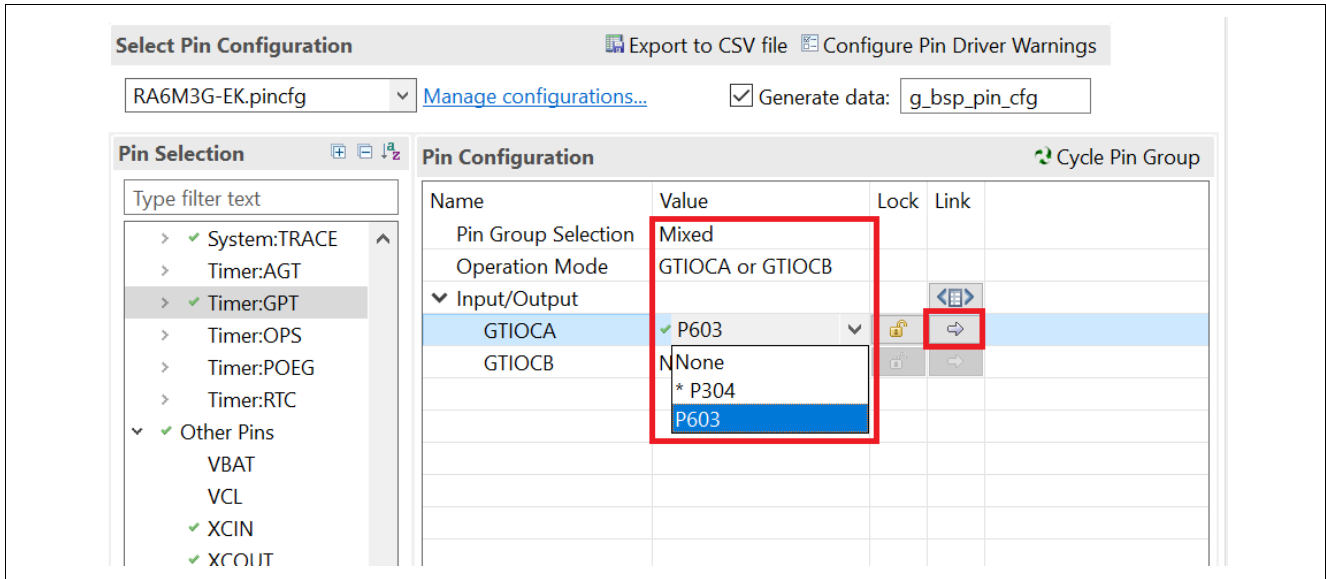
Figure 25.   P603 Properties

19. After that you can hit the **=>** sign to see the pin P603 settings.
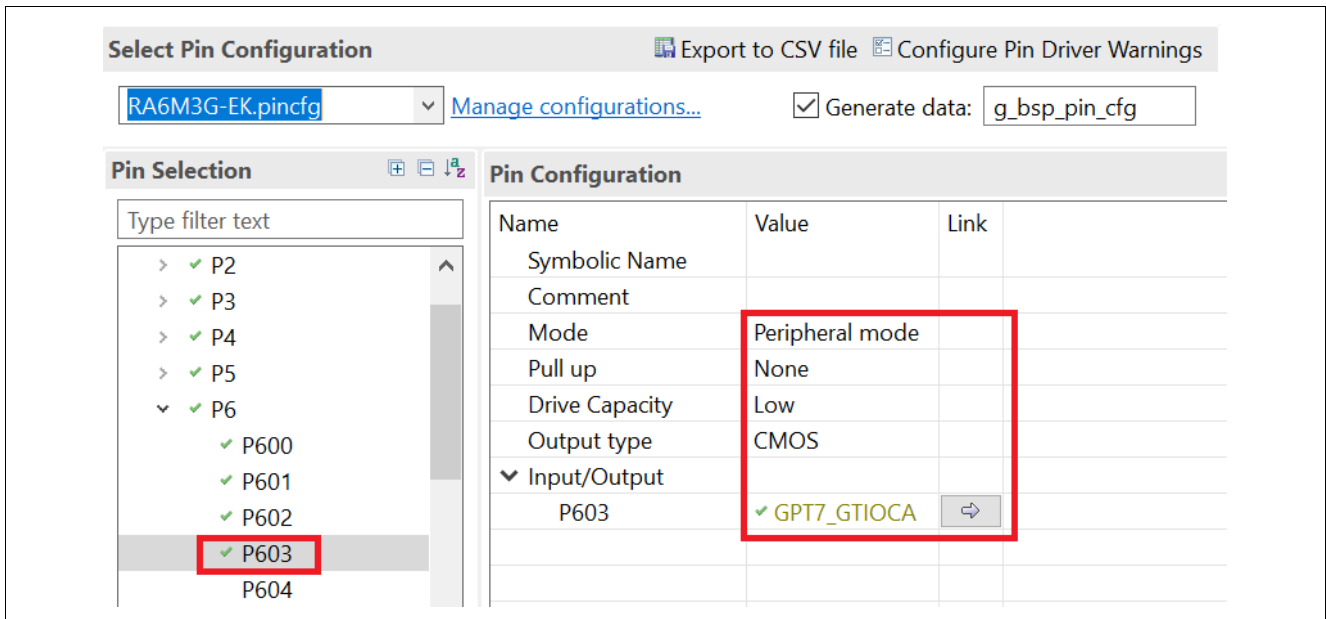
Figure 26.   P603 Settings

20. Figure 27 is a schematic of the LCD_BLEN signal.
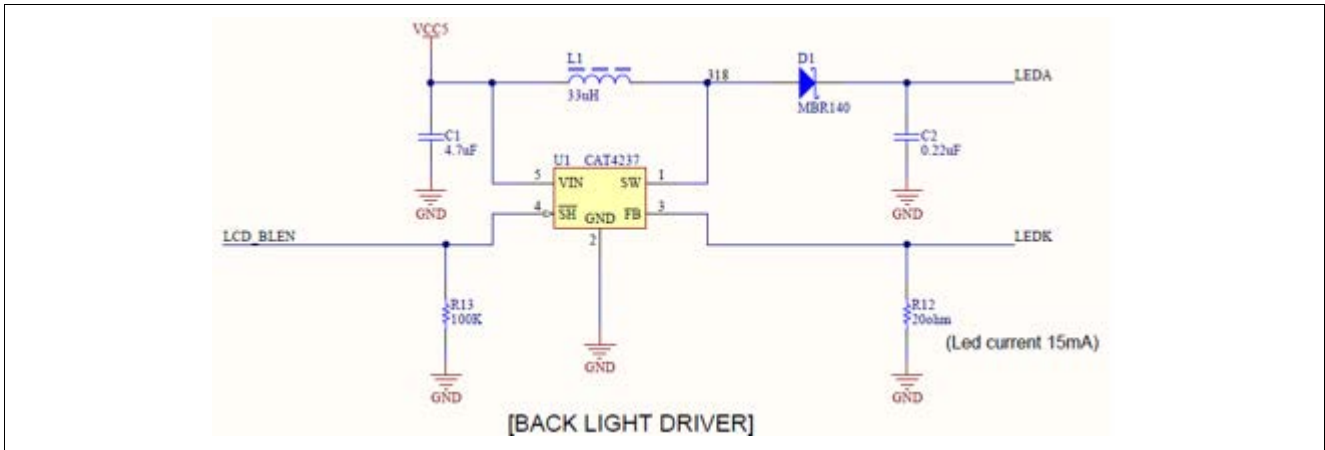
Figure 27.   LCD_BLEN Schematic

21. Refer to "system_thread_entry.c" file in "**Source.zip**" for more information. This function below controls the PWM output.

```c
/************************************************************************
 * @brief This function is setting up GPT/PWM timer
 * @param[in] None
 * @retval    fsp_err_t
 ************************************************************************/
static fsp_err_t gpt_timer_PWM_setup(void)
{
    fsp_err_t err = FSP_SUCCESS;
    /* Open GPT */
    err = R_GPT_Open(&g_timer_PWM_ctrl, &g_timer_PWM_cfg);
    if(FSP_SUCCESS != err)
    {
        return err;
    }
    /* Enable GPT Timer */
    err = R_GPT_Enable(&g_timer_PWM_ctrl);
    /* Handle error */
    if (FSP_SUCCESS != err)
    {
        return err;
    }
    /* Start GPT timer */
    err = R_GPT_Start(&g_timer_PWM_ctrl);;
    if(FSP_SUCCESS != err)
    {
        return err;
    }

    return err;
}
```

3. Add and Configure "Touch Function Driver"

1. Add a **New Thread** and name it **Touch Thread** with the setting below.
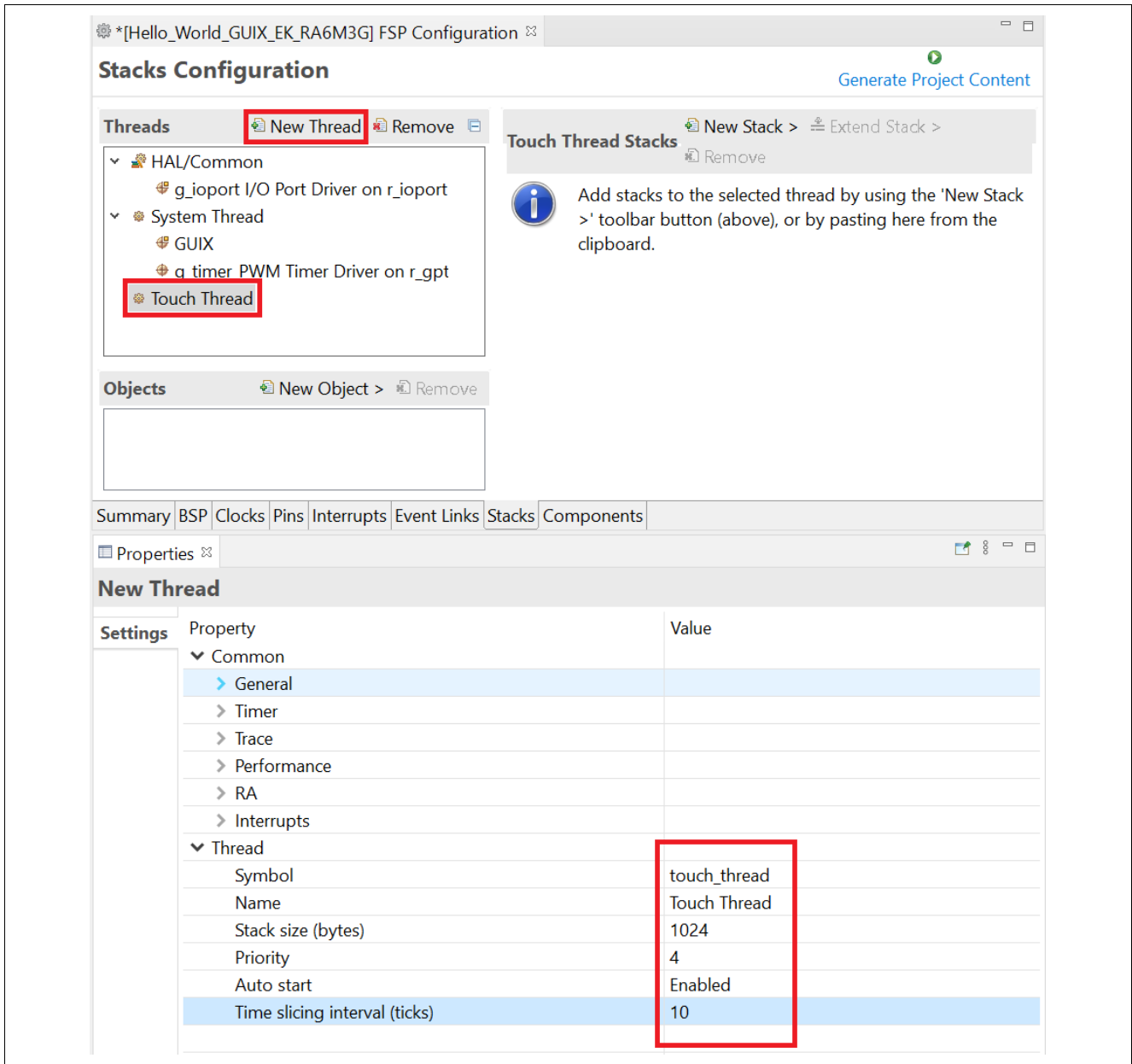
Figure 28. Create Touch Thread

2. Select **Touch Thread** and add new stack **External IRQ Driver on r_icu**. **New Stack > Driver > Input > External IRQ Driver on r_icu**.
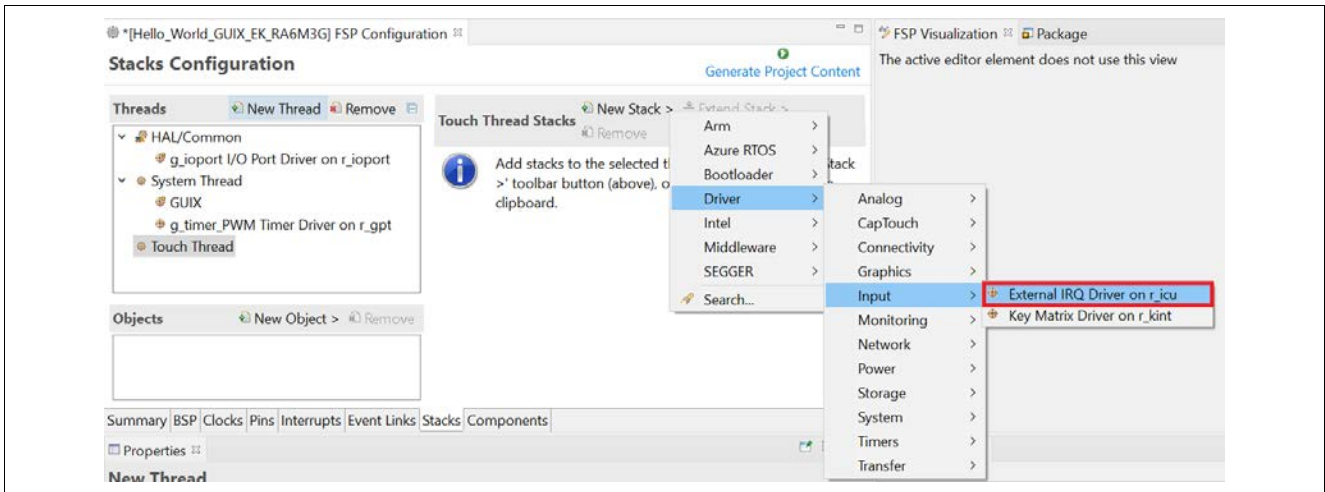
Figure 29.   Add External IRQ Driver on r_icu

3. User should disable the pin P206 before we move to the next step. Click the **Pins** tab and follow the image below.
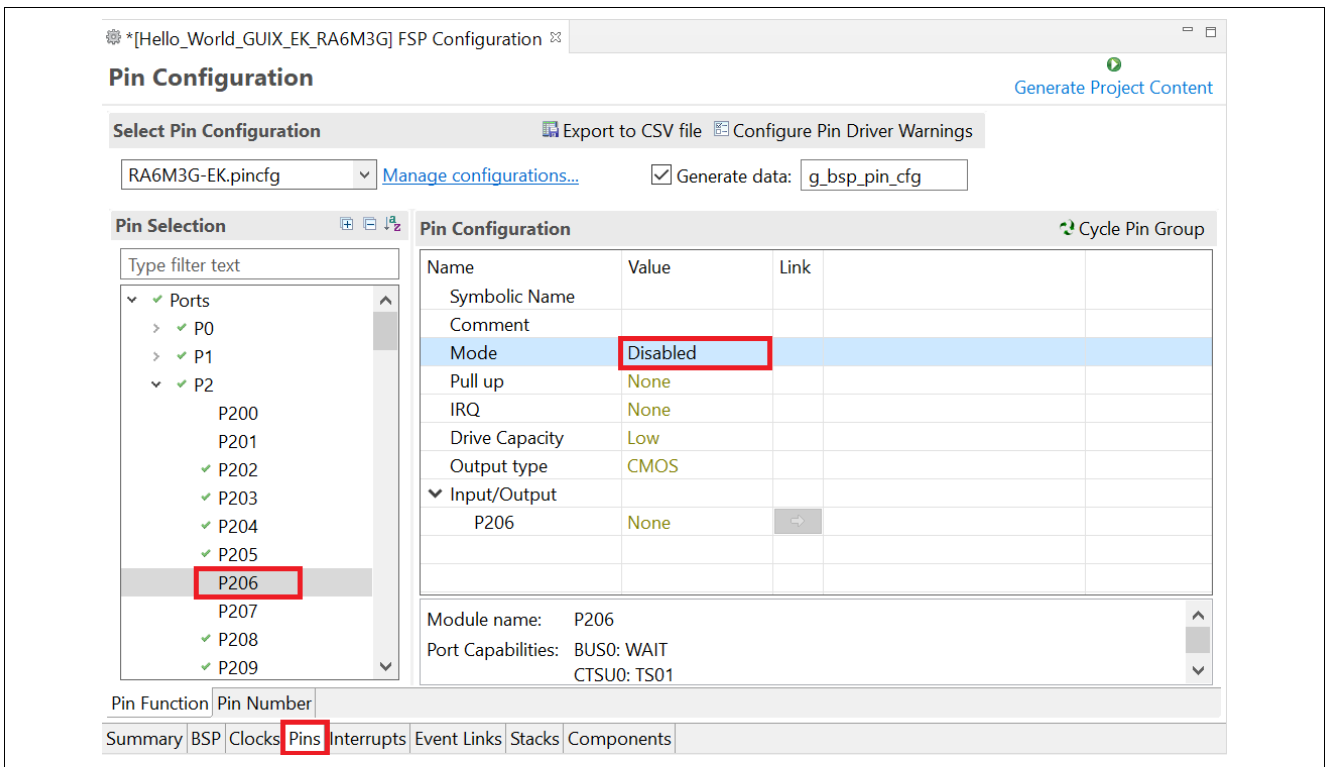
Figure 30.   Disable Pin P206

4.   Setting the properties for the **External IRQ Driver on r_icu**.

Figure 31.   Set Properties for External IRQ Driver on r_icu

5. The External Touch IRQ Driver on r_icu will use pin P206. Be sure this pin configuration is set as shown below.
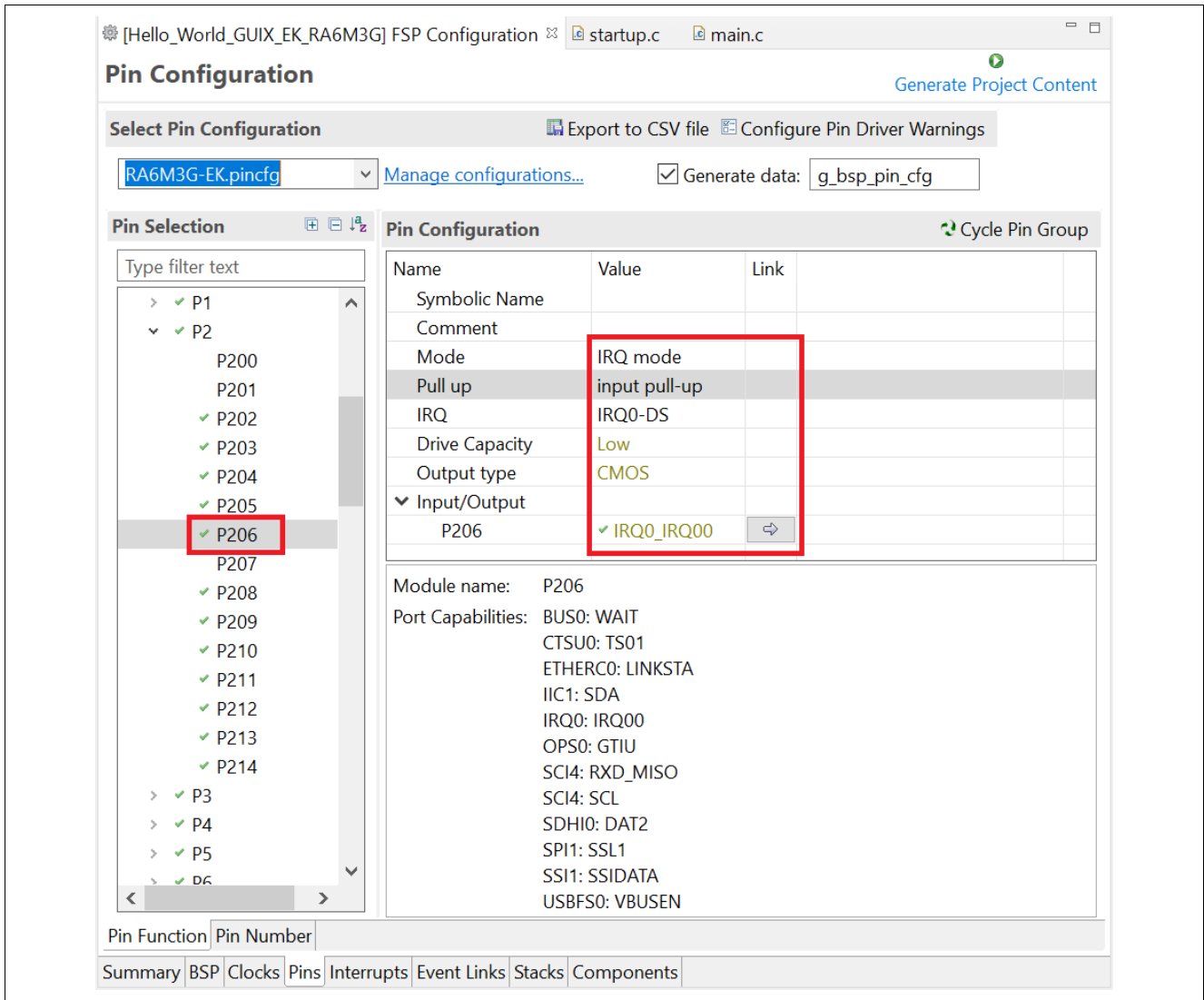
Figure 32. Configure Pin P206

6. Select **Touch Thread** and add a new stack. **New Stack > Driver > Connectivity > I2C Master Driver on r_iic_master**.
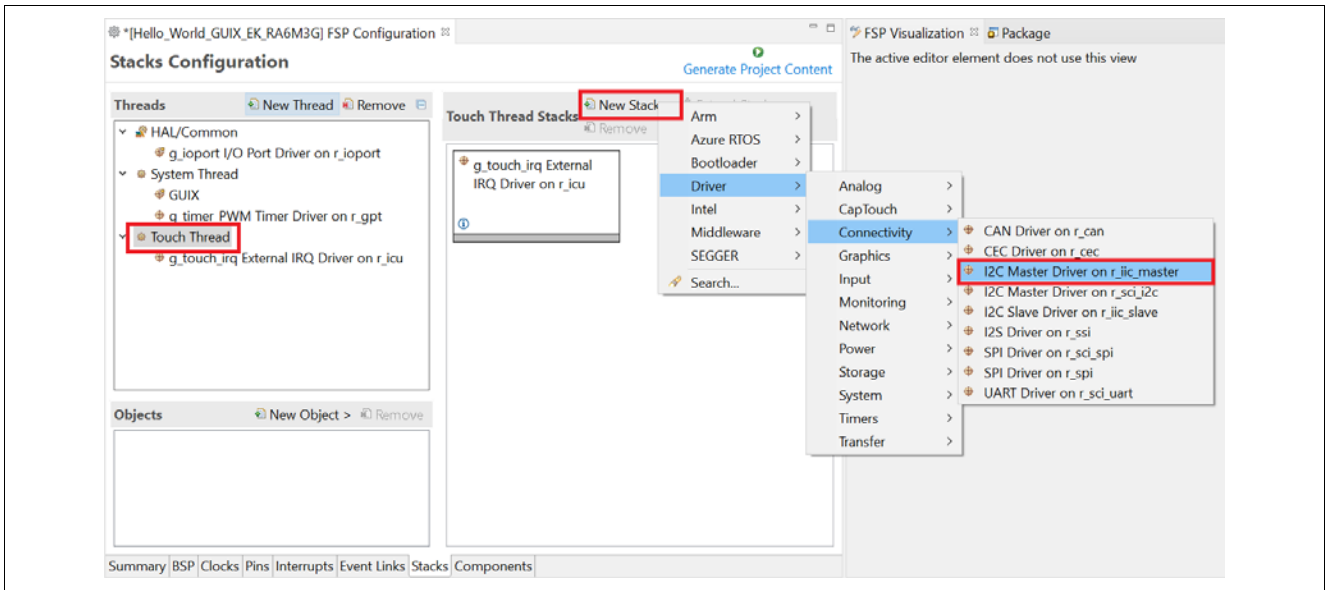
Figure 33. Add I2C Master Driver on r_iic_master

7. Setting the properties for **g_i2c_touch I2C Master Driver on r_iic_master**.
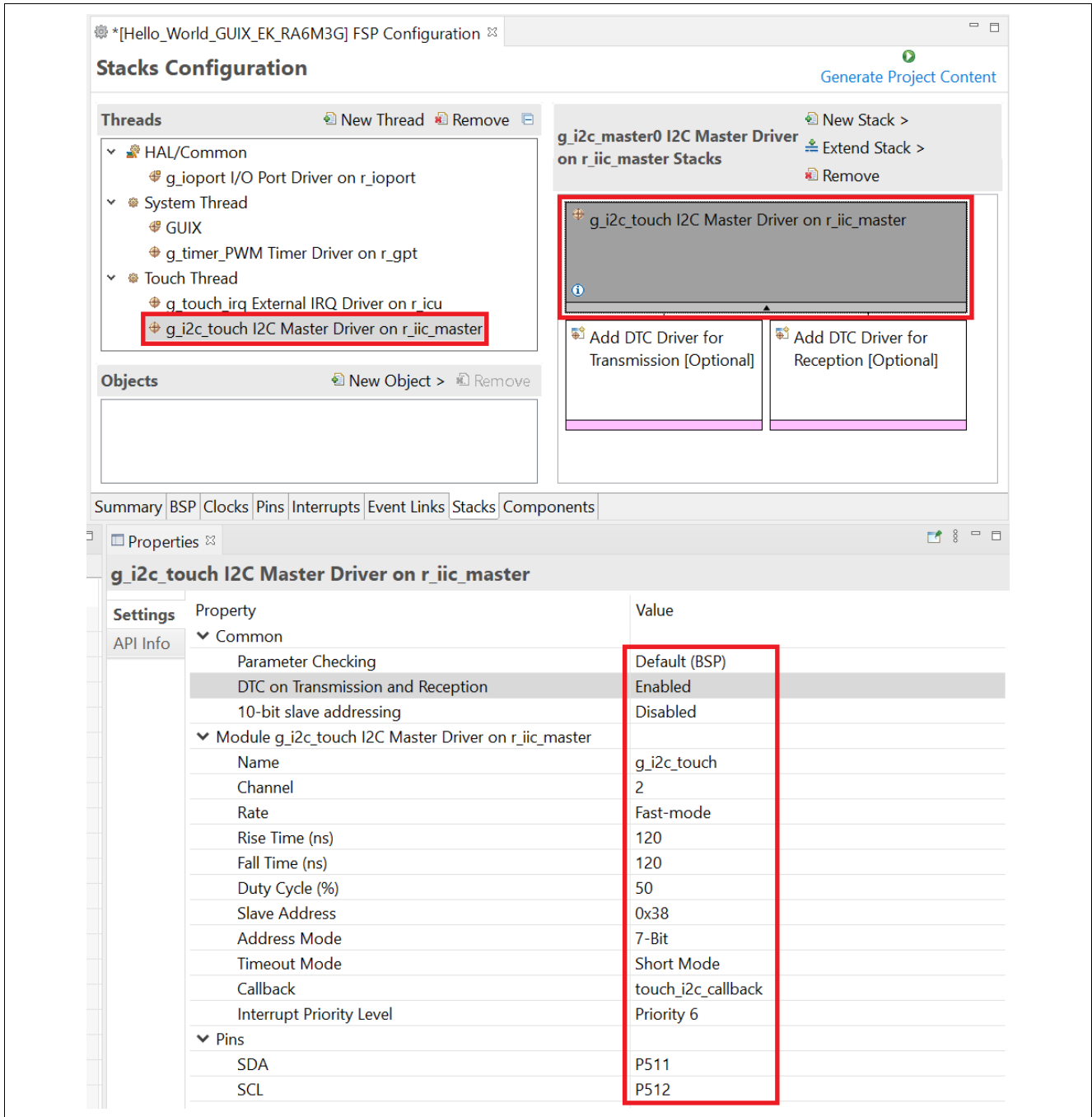
Figure 34. Set Properties for g_i2c_touch I2C Master Driver on r_iic_master

8. Under **g_i2c_touch I2C Master Driver on r_iic_master** add in **Add DTC Driver for Transmission**. Also, set properties for **Add DTC Driver for Transmission**.
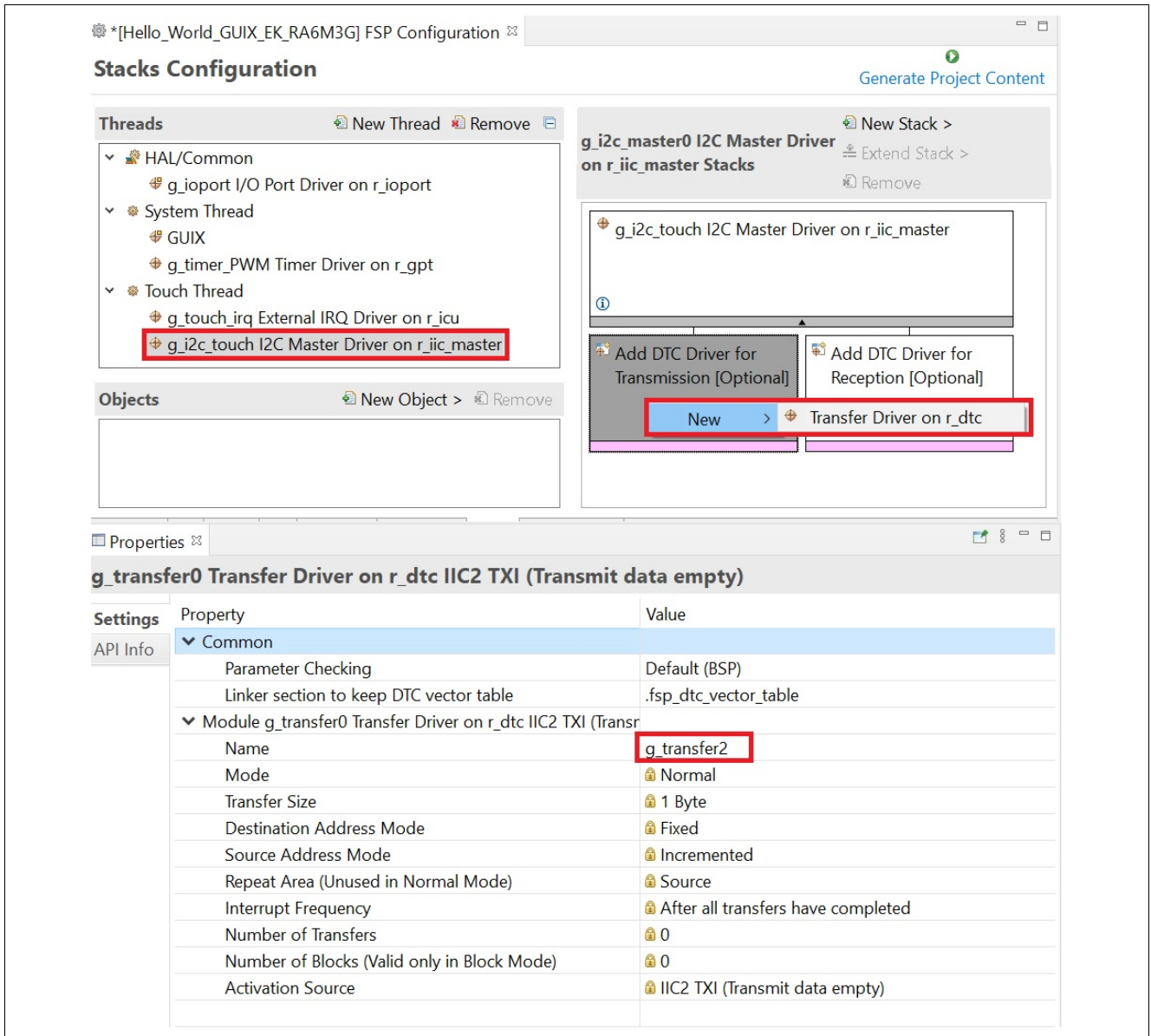
Figure 35.   Add DTC Driver for Transmission

9.   Add in **Add DTC Driver for Reception**. Then set properties for **Add DTC Driver for Reception**.
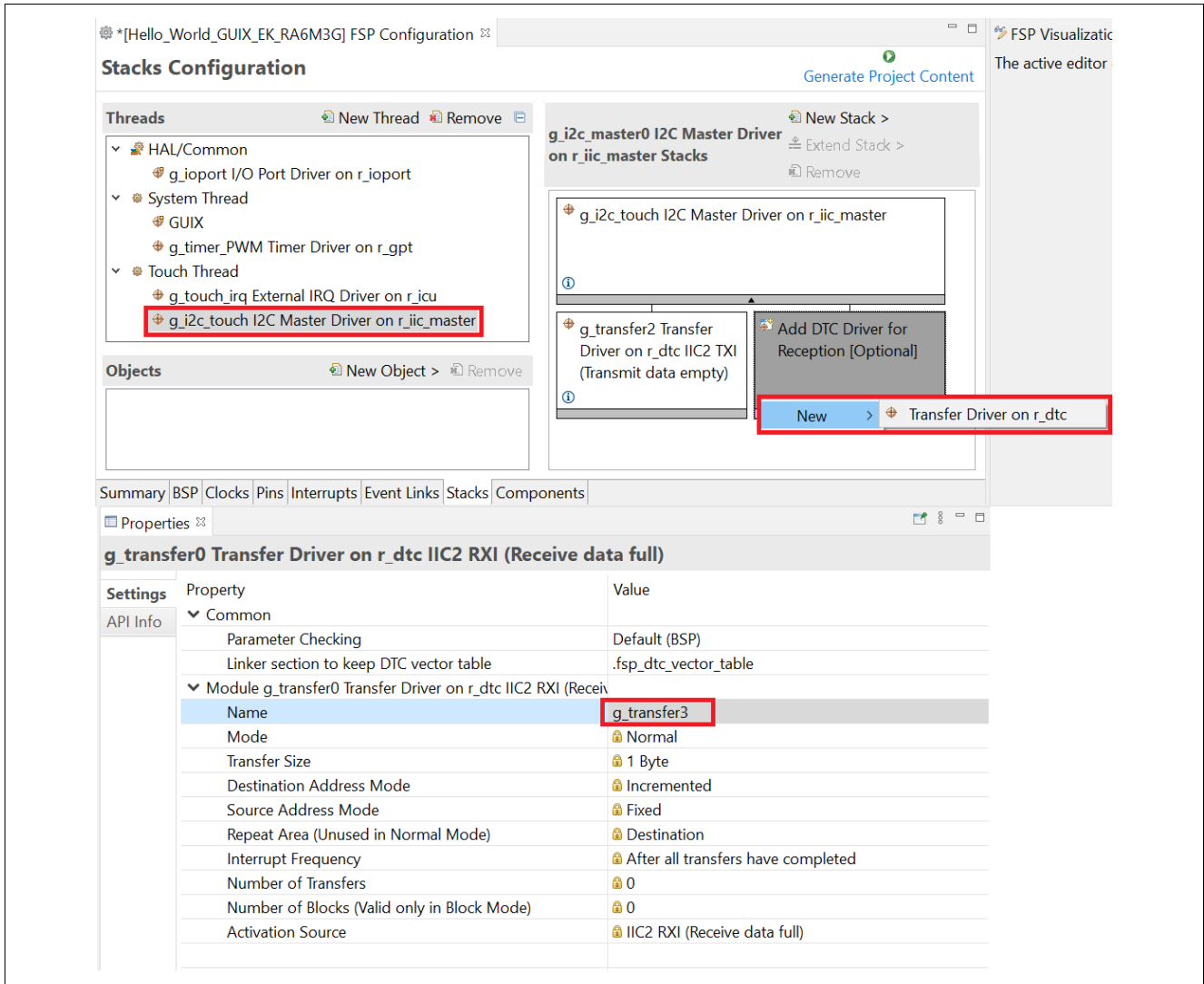
Figure 36.   Add DTC Driver for Reception

10. After adding transmission and reception, the configuration looks like below.
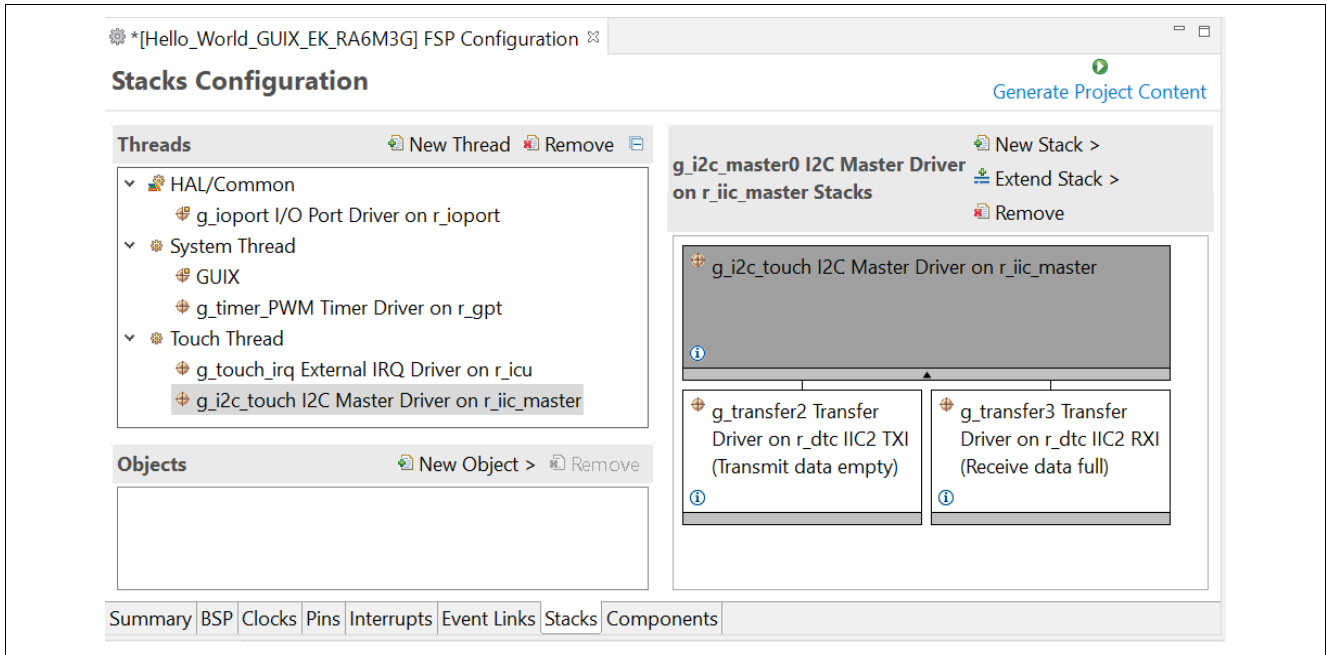
Figure 37.   Configuration for Transmission and Reception

11. From the **Objects** window, click on **New Object** and select **Semaphore**. Then set **g_touch_semaphore Semaphore** properties.
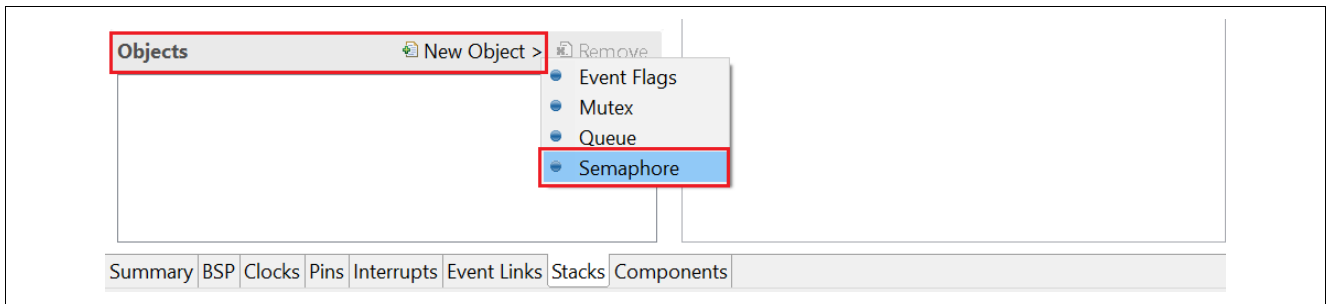
Figure 38.   Add Semaphore
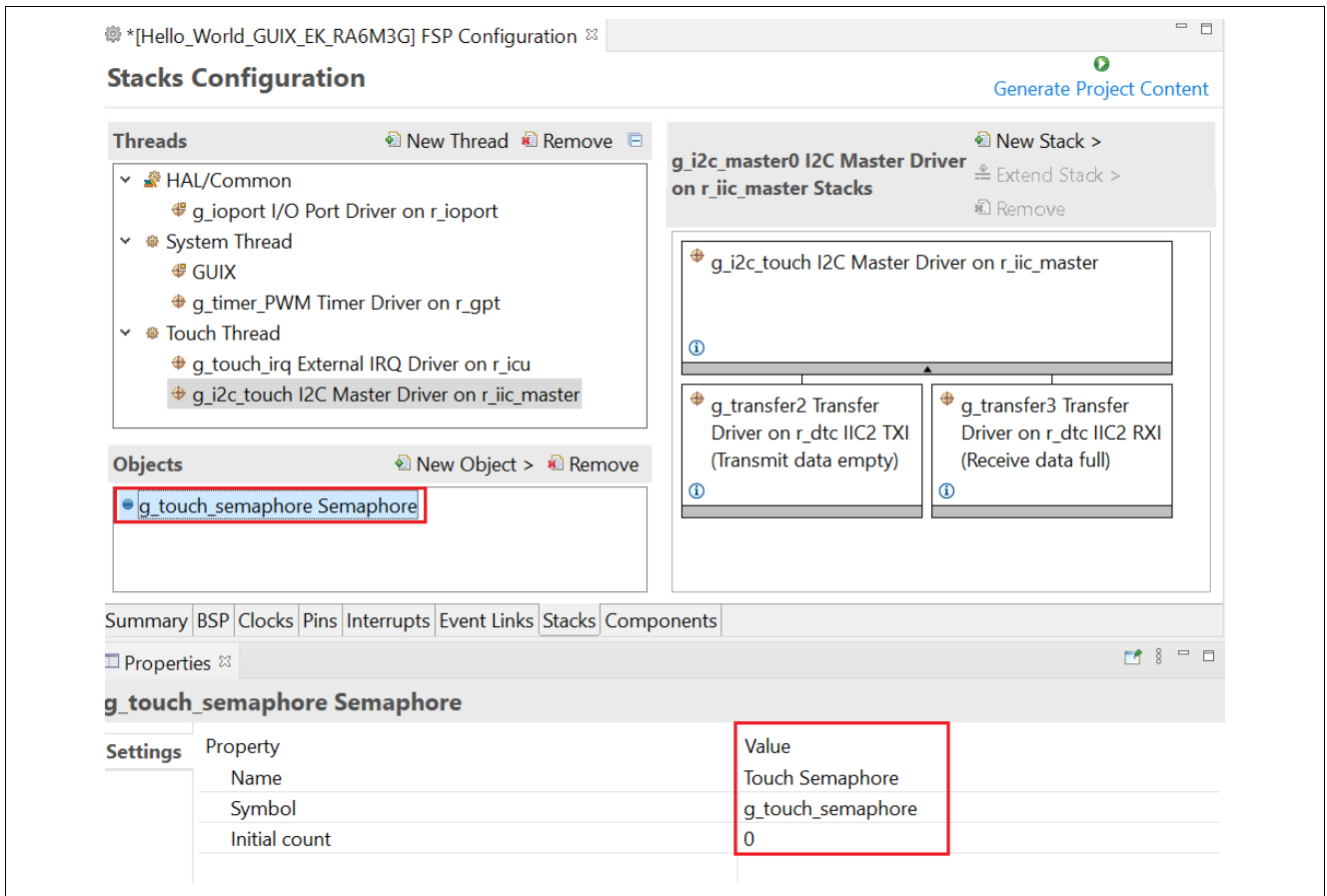
Figure 39.   Set g_touch_semaphore Semaphore Properties

12. Again, from the **Objects** window. Click on **New Object** and select **Semaphore**. Then set **g_i2c_semaphore Semaphore** properties.
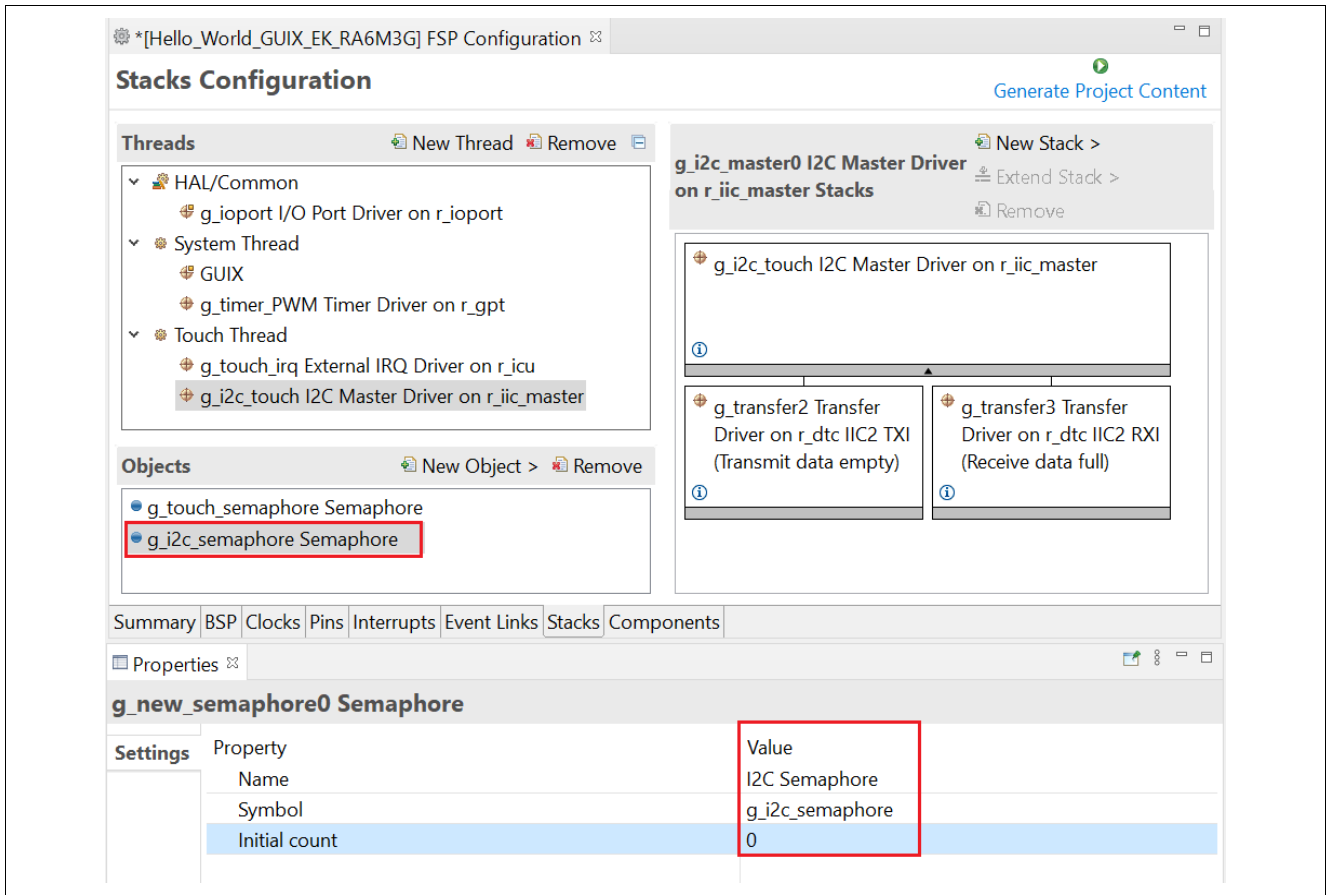
Figure 40.   Add Semaphore
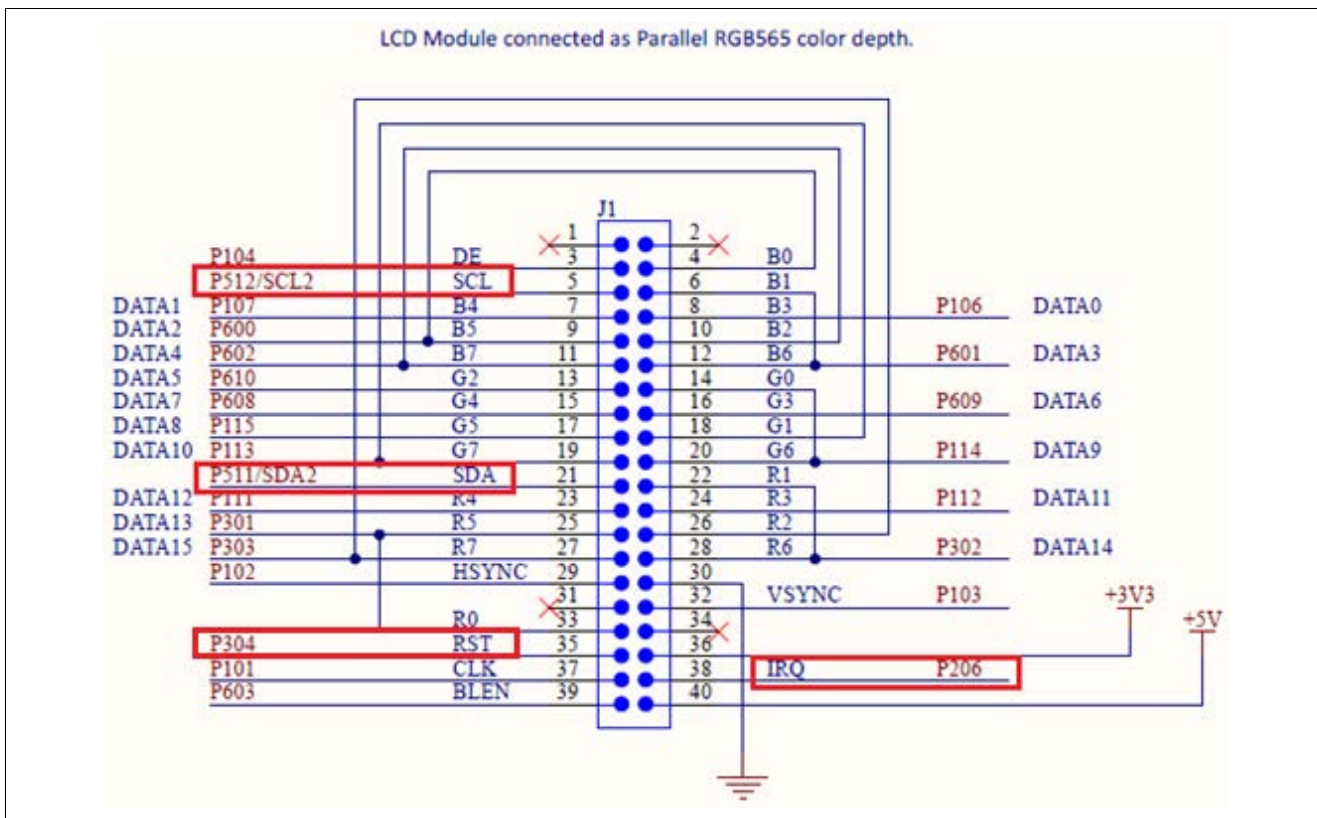
Figure 41. Set g_i2c_semaphore Semaphore Properties

The pins marked in red above are used for the touch panel controller on the LCD board:
— IRQ0 interrupt (P206) is used to trigger touch events.
— I2C channel 2 (P512, P511) is used to read and write data to the touch controller.
— Touch driver folder `touch_ft5x06` for touch function is inside the provided `Source` folder.
— P304 is used to reset the LCD's touch controller.

13. Since the IRQ0 (P206) needs a pull-up to function properly with the LCD board, don't change the setting that was done by FSP, as shown below.

Figure 42.   P206 Pin Configurtion

14. Note: Refer to the `touch_thread_entry.c` file in `Source.zip` for more information. The following code initializes the touch controller and process touch events.

```c
/* Touch Thread entry function */
void touch_thread_entry(void)
{
    GX_EVENT gxe = {0};
    UINT status  = 0;
    /*
     * Initialize Touch controller:
     *  FT5X06 driver
     *  IRQ interrupt
     */
    ft5x06_init(&g_i2c_touch, &g_i2c_semaphore, BSP_IO_PORT_03_PIN_04);
    /* Enable touch IRQ */
    R_ICU_ExternalIrqOpen(g_touch_irq.p_ctrl, g_touch_irq.p_cfg);
    R_ICU_ExternalIrqEnable(g_touch_irq.p_ctrl);

    while (1)
    {
        /* Wait for IRQ from touch controller. */
        status = tx_semaphore_get(&g_touch_semaphore, TX_WAIT_FOREVER);
        if(TX_SUCCESS != status)
        {
            APP_ERR_TRAP(FSP_ERR_ASSERTION);
        }

/* Get touch data from the FT5X06 */
ft5x06_payload_get(&touch_data);

if(0U < touch_data.num_points)
{
    /* Send only the TOUCH event of the first finger to the GUIX Model View Controller. */
    if( TOUCH_EVENT_DOWN == touch_data.point[0].event)
    {
        gxe.gx_event_payload.gx_event_pointdata.gx_point_x       = touch_data.point[0].x;
        gxe.gx_event_payload.gx_event_pointdata.gx_point_y       = touch_data.point[0].y;
        gxe.gx_event_type = GX_EVENT_PEN_DOWN;
        gx_system_event_send(&gxe);

    }
    else if (TOUCH_EVENT_UP == touch_data.point[0].event && GX_EVENT_PEN_DOWN == gxe.gx_event_type)
    {
        gxe.gx_event_payload.gx_event_pointdata.gx_point_x       = touch_data.point[0].x;
        gxe.gx_event_payload.gx_event_pointdata.gx_point_y       = touch_data.point[0].y;
        gxe.gx_event_type = GX_EVENT_PEN_UP;
        gx_system_event_send(&gxe);

    }
```

15. In RA Configurator, click Generate Project Content to generate project content.

4. Create Folders in the Hello_World GUIX_EK_RA6M3G Project for Azure RTOS GUIX Studio Project

1. Under folder `src` create a new folder and name it `guix_gen`. Follow the image below, then click **Finish**.
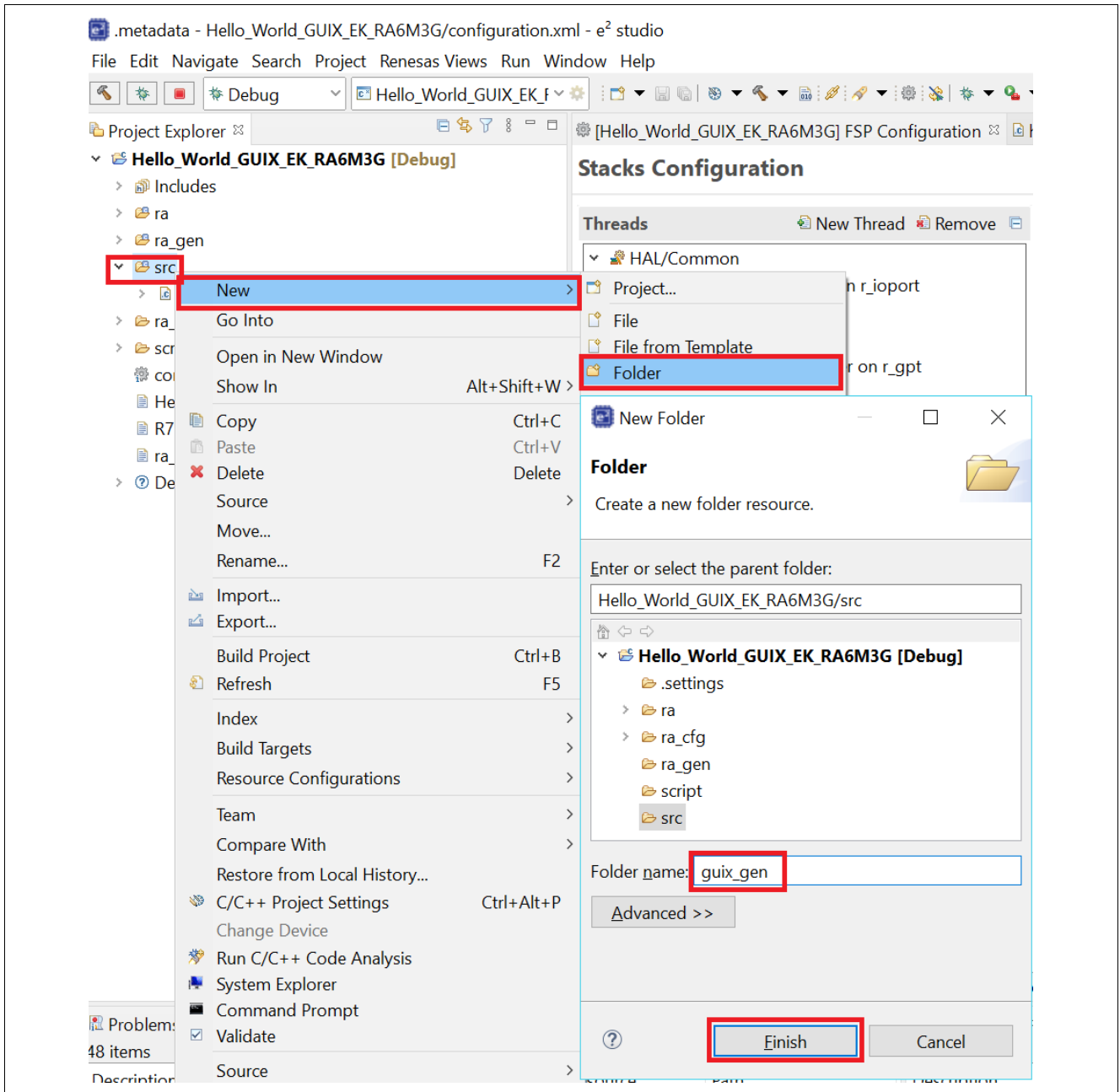
Figure 43. Create guix_gen Folder

2. Under folder `src` create a new folder and name it `guix_studio`. Then follow the image below and click **Finish**.
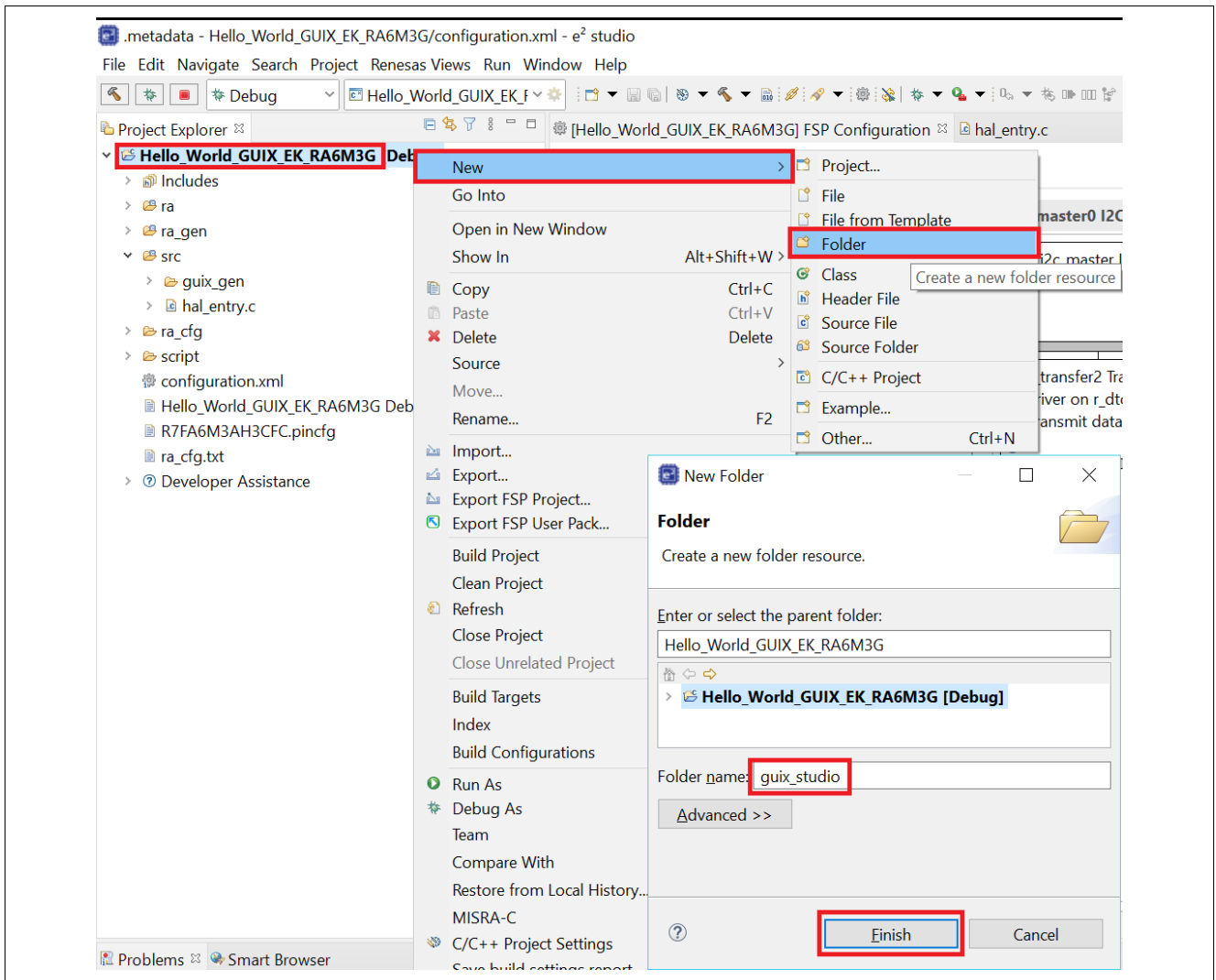
Figure 44.   Create guix_studio Folder

3. Go to the Hello_World_GUIX_EK_RA6M3G project. Select the `guix_studio` folder, right-click and select **Resource configuration**, then select **Exclude from Build…** since it contains the Azure GUIX Studio project, which won't be built by FSP.
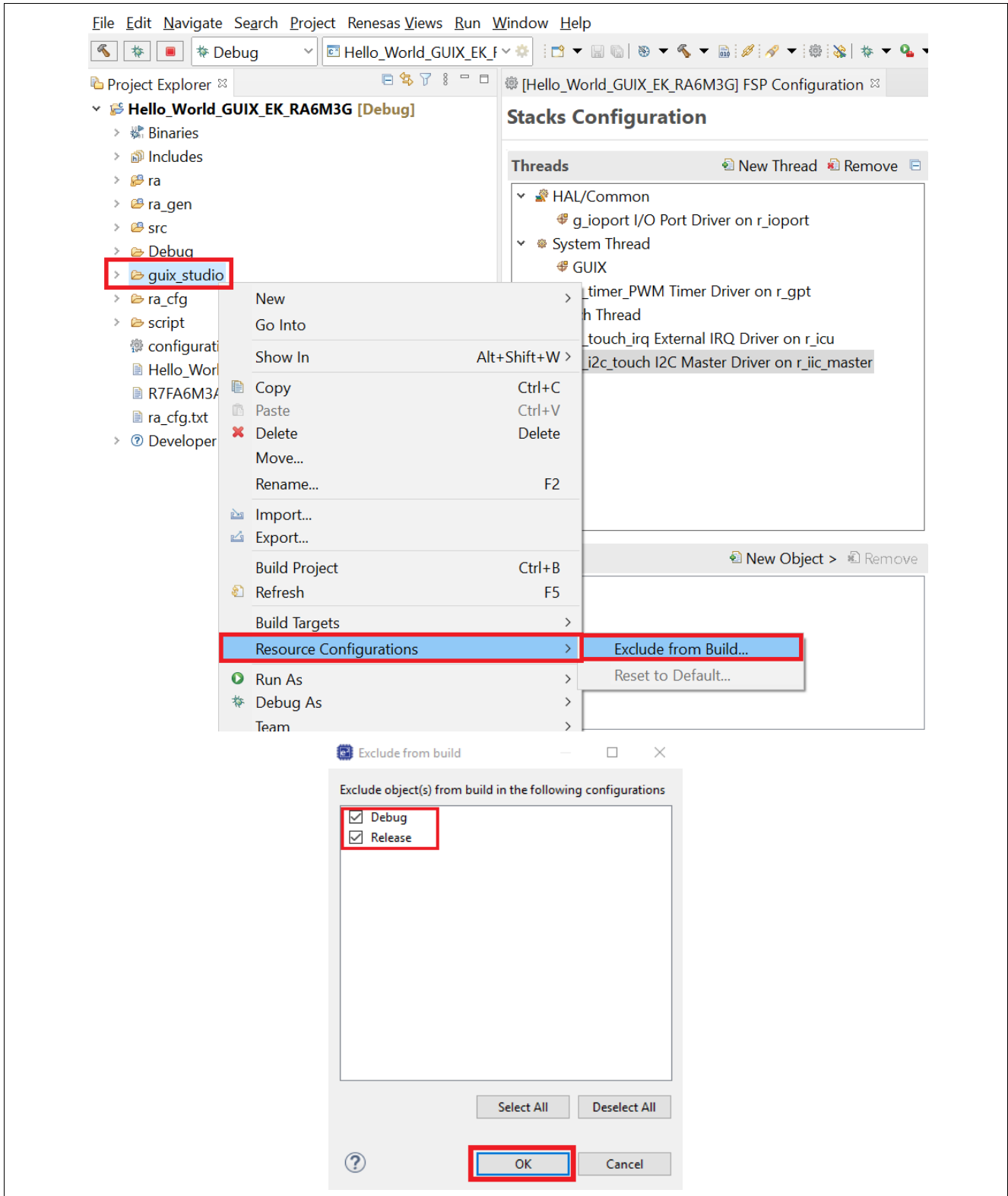
Figure 45. Exclude guix_studio from Build

4. Under folder `guix_studio` that was created earlier, create a new folder and name it `GNU`. Follow the image below, then click **Finish**.
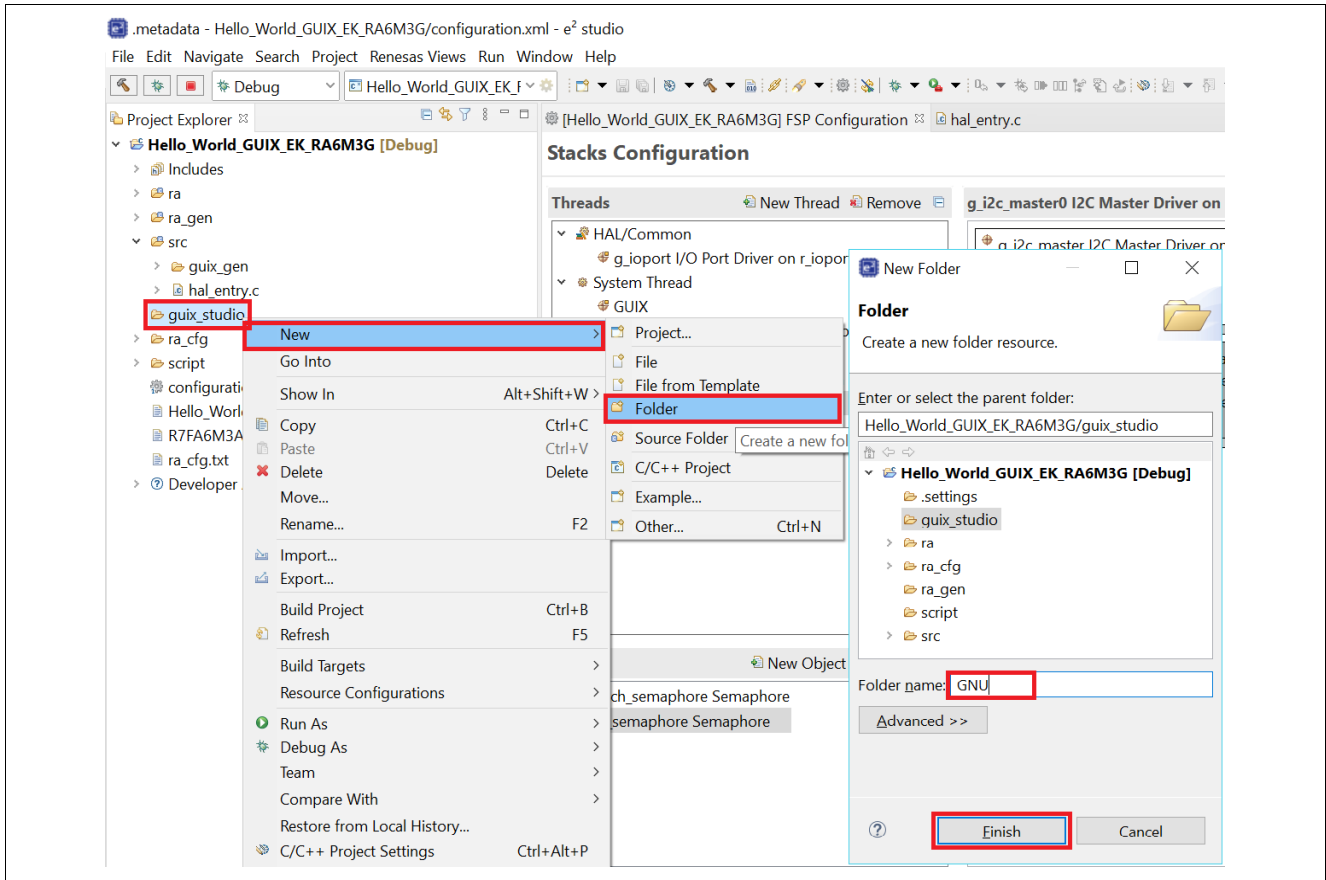
Figure 46. Create GNU Folder

5. After the sub-folder GNU is created, the folder structure should look like the image below.
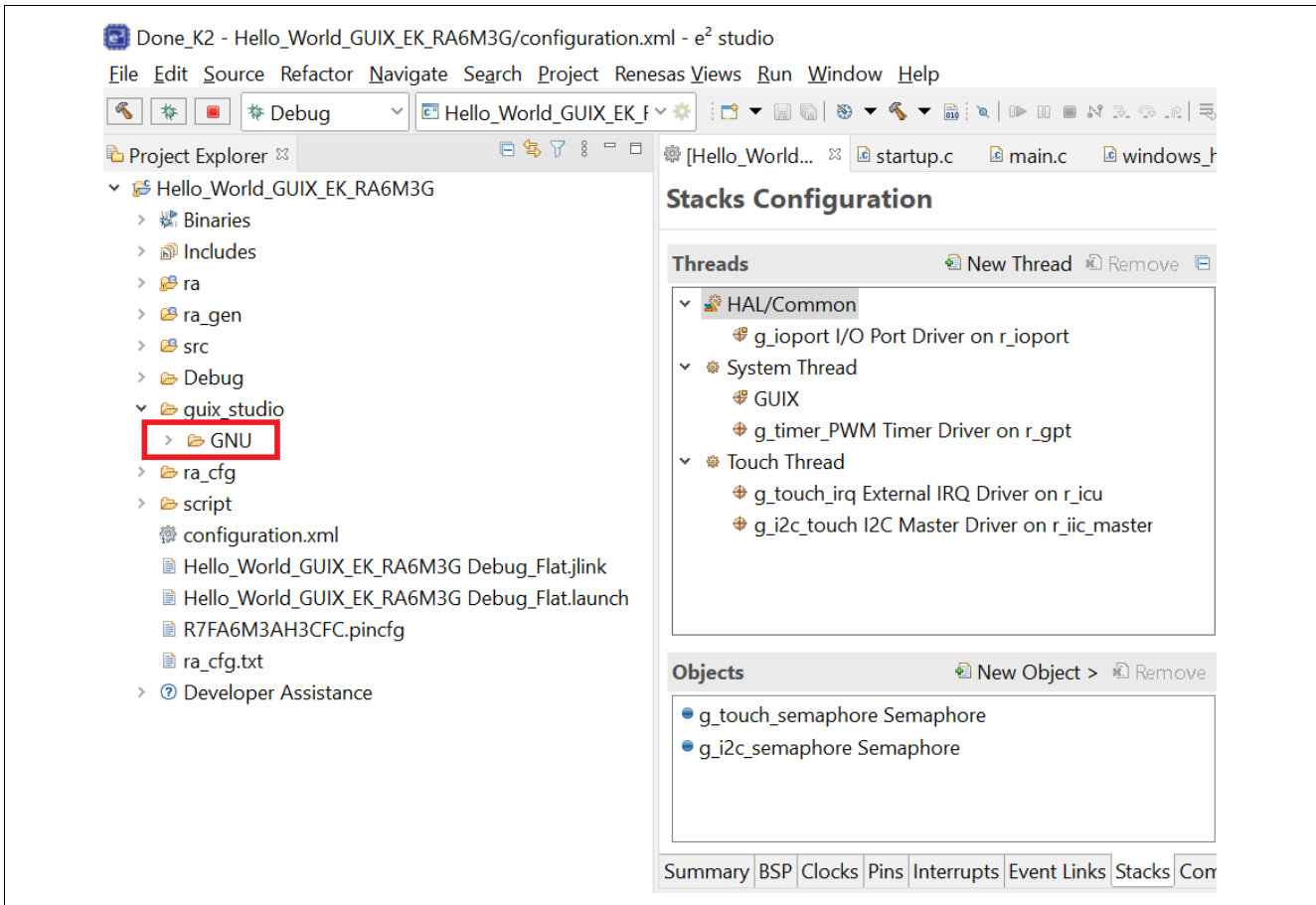
Figure 47. After GNU Folder is Created

6. Unzip and open the provided folder `Source.zip`. Copy the four `*.c` files and one folder `touch_ft5x06` and paste into the folder `src` of your project **Hello_World_GUIX_EK_RA6M3G**.
   — `Touch_ft5x06` folder
   — `hal_entry.c`
   — `system_thread_entry.c`
   — `touch_thread_entry.c`
   — `windows_handler.c`
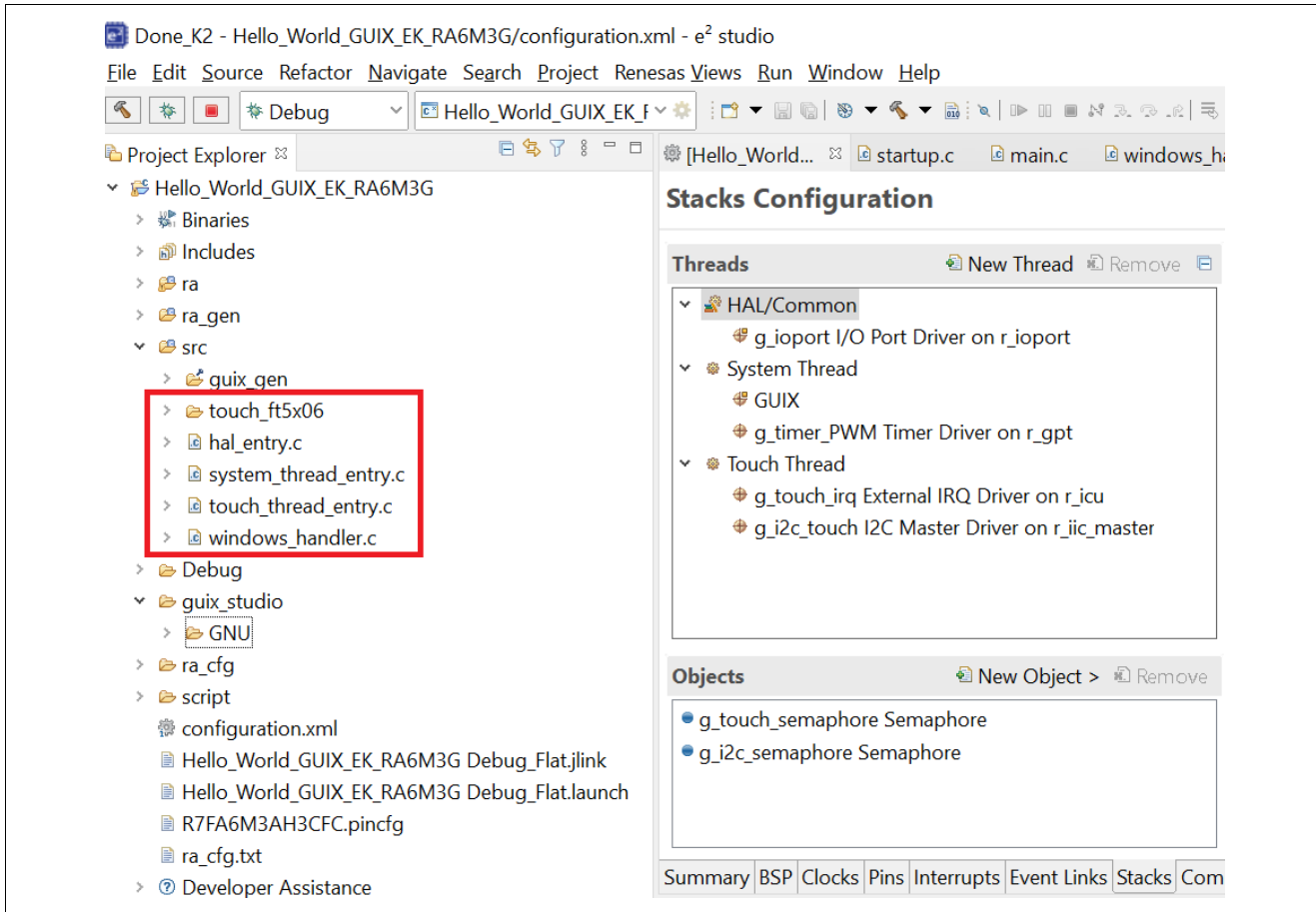
Figure 48.   Files Copied from Source.zip

5.   Using Azure RTOS GUIX Studio create GUI Windows

1. Open Azure RTOS GUIX Studio v6.1.7 or greater.
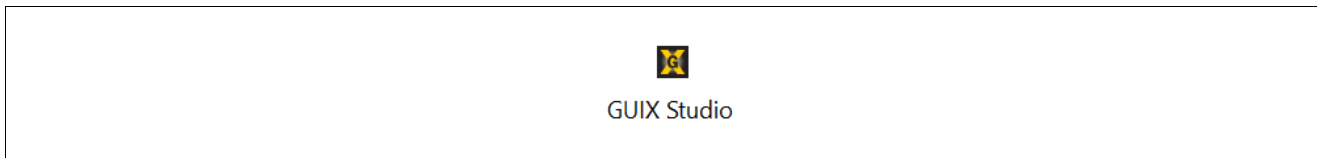
Figure 49.   GUIX Studio Icon

2. Create a **New Project** and name it **Hello World**.
   a. Select **Project**. From the drop-down list, select **New Project**.
   b. Project Name: **Hello_World**.
   c. Project Path: Browse to the location of the folder you created in the
      Hello_World_GUIX_EK_RA6M3G > **guix_studio\GNU** as shown in the image below.
   d. Hit **OK** button and then the **Save** button to confirm your selections.
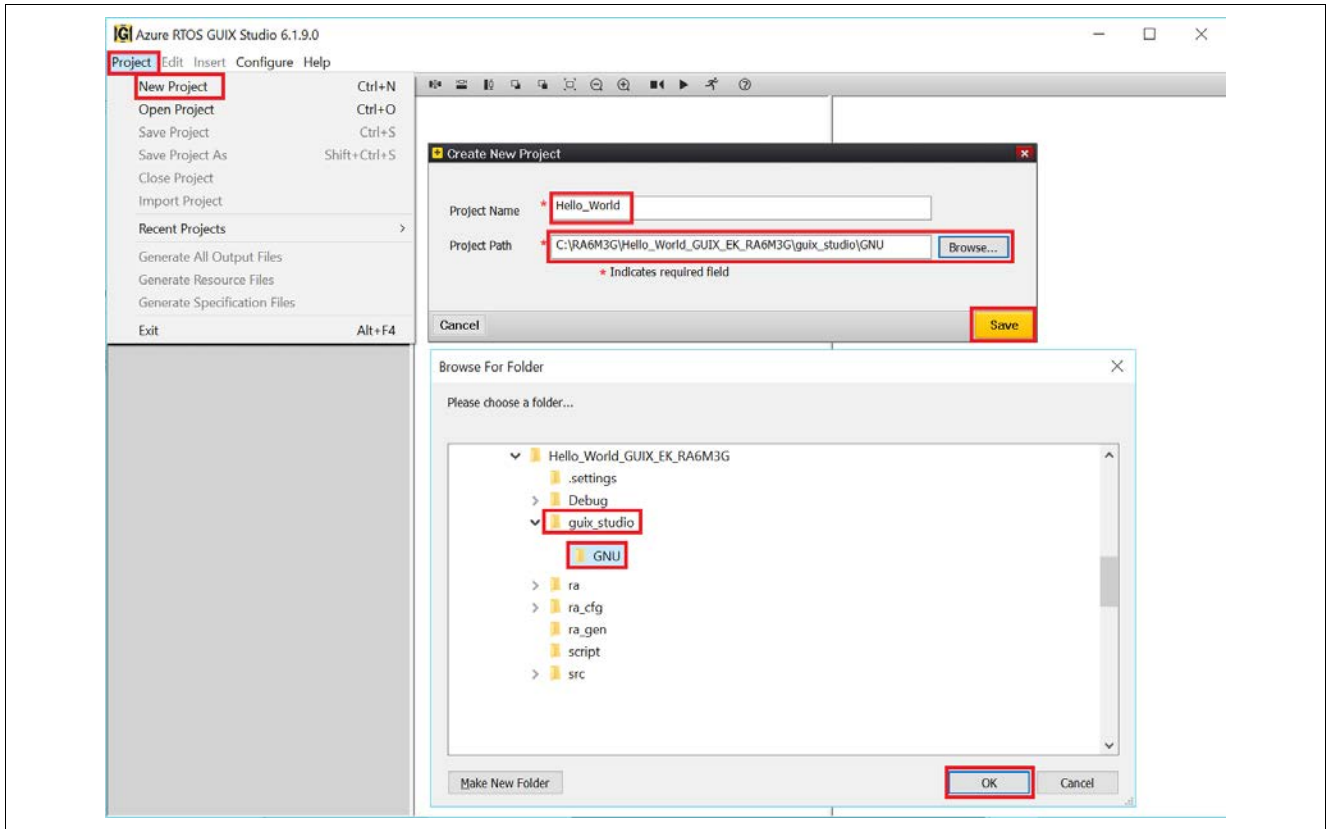
Figure 50.   Create Hello World Project

3. A new **Configure Project** window will pop up and you need to set all the options as shown in Figure 51 and Figure 52. Finally, click **Save**.
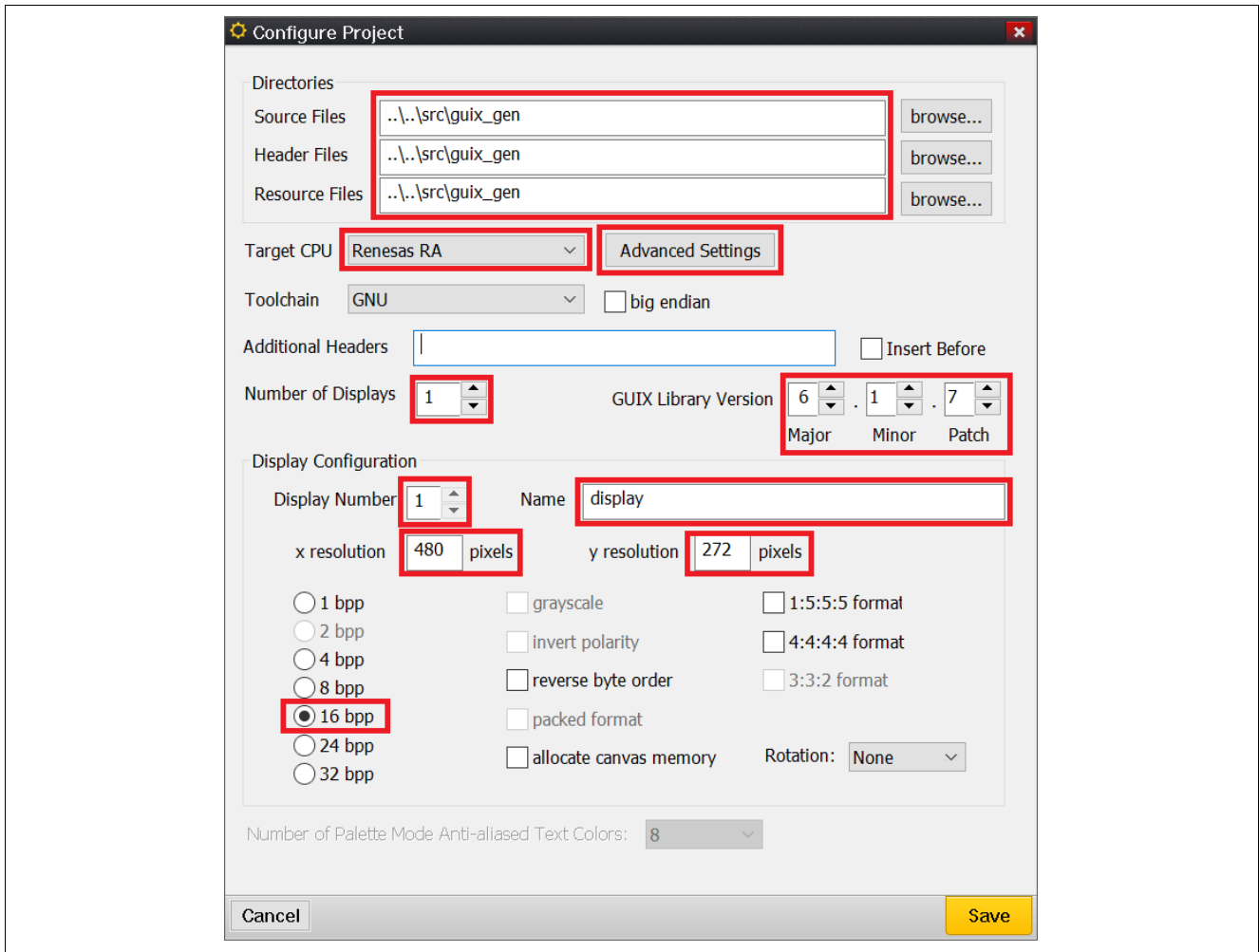
Figure 51.   Configure Project

Advanced Settings:
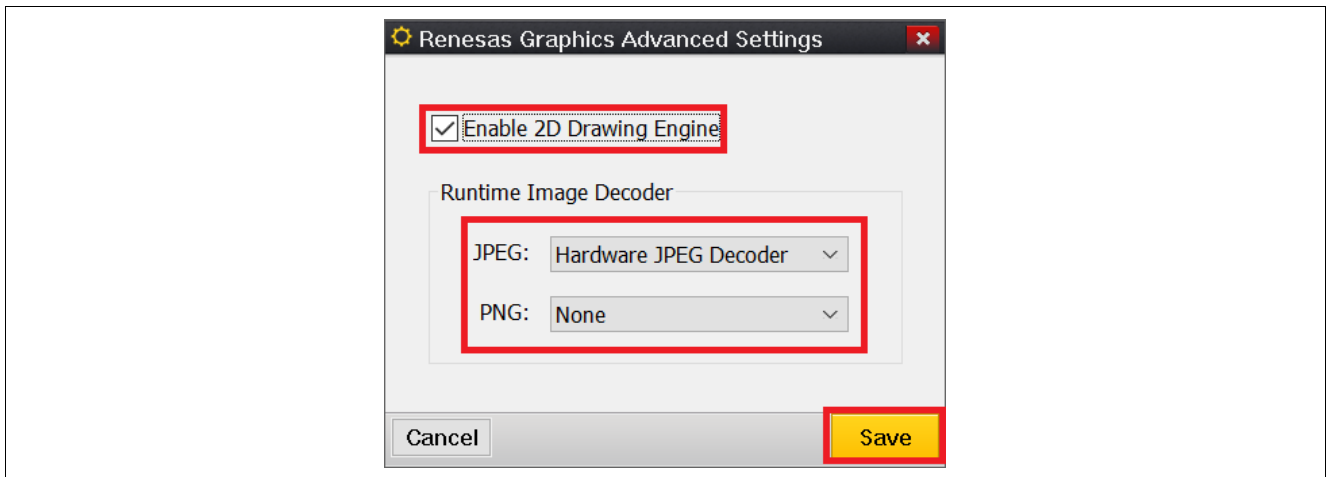
Figure 52.   Advanced Settings

5.4 Setting **Properties View** of **Window1**
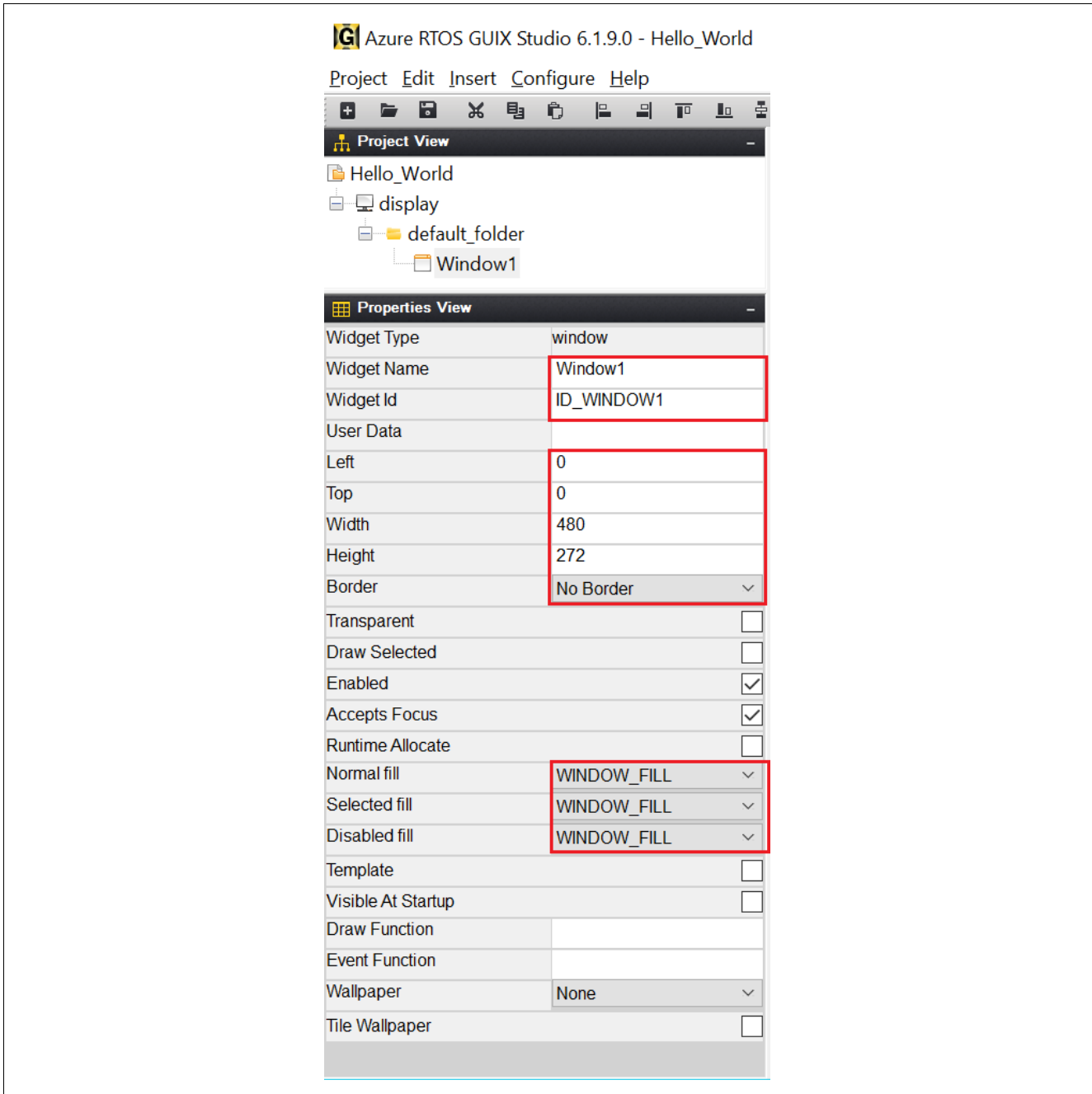
Figure 53.   Property View of Window1

4.   To add String ID, click on **Strings**. Follow the images below.

Figure 54.   Strings

From the **Strings** dropdown, click **+ Add New String**.
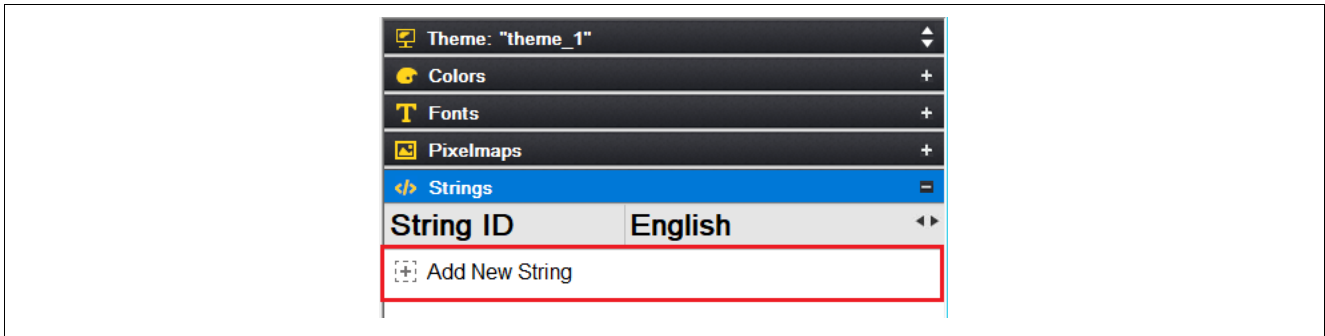
Figure 55.   Add New String

A new **String Table Editor** window will pop up. Click the **Add String** button.
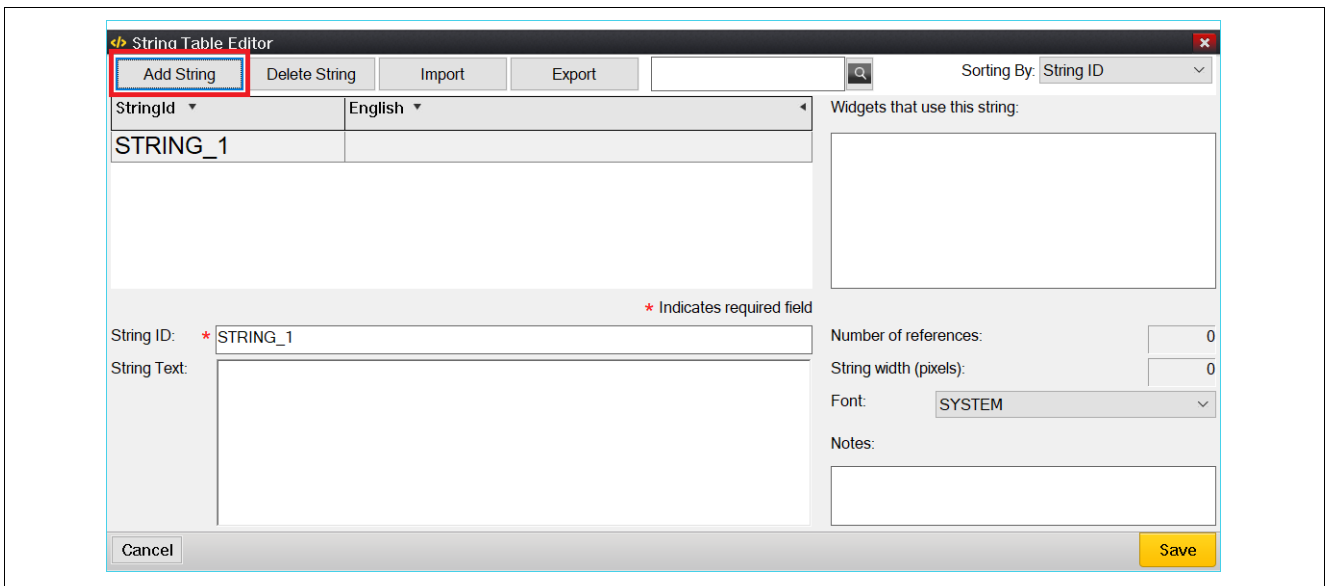
Figure 56.   String Table Editor

5. Edit **String ID** and **String Text**.
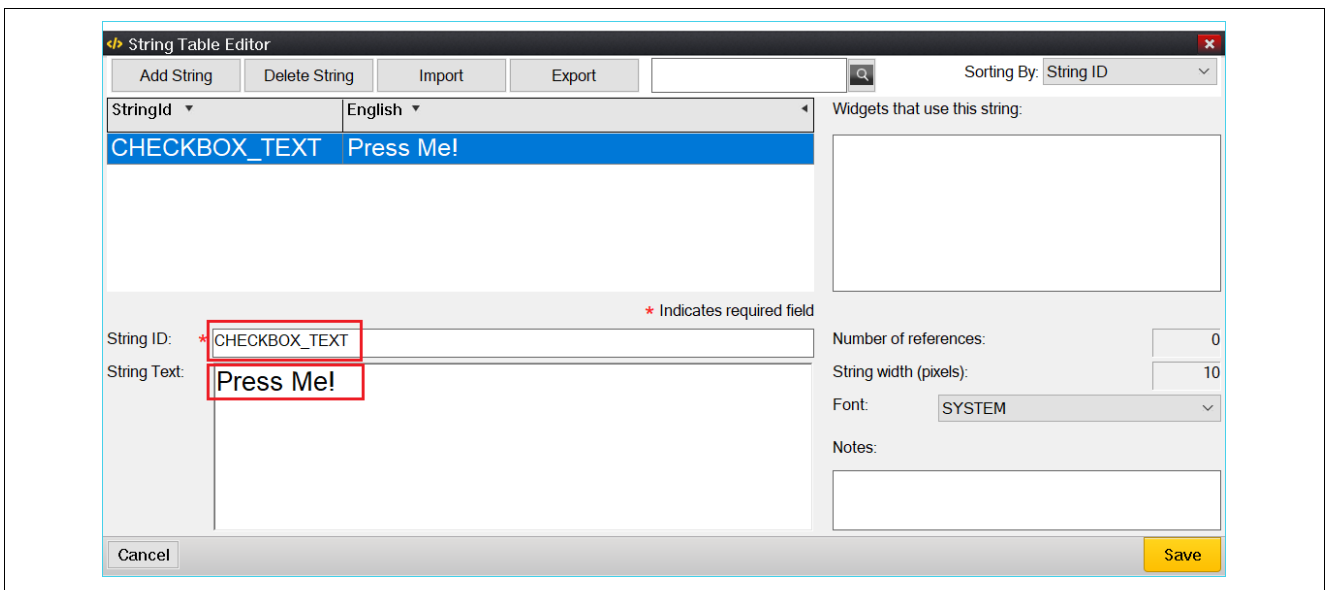
Figure 57.   Edit String ID and String Text

6. Continue to click **Adding String**, then edit **String ID** and **String Text** until the table appears like Figure 58. Then click the **Save** button
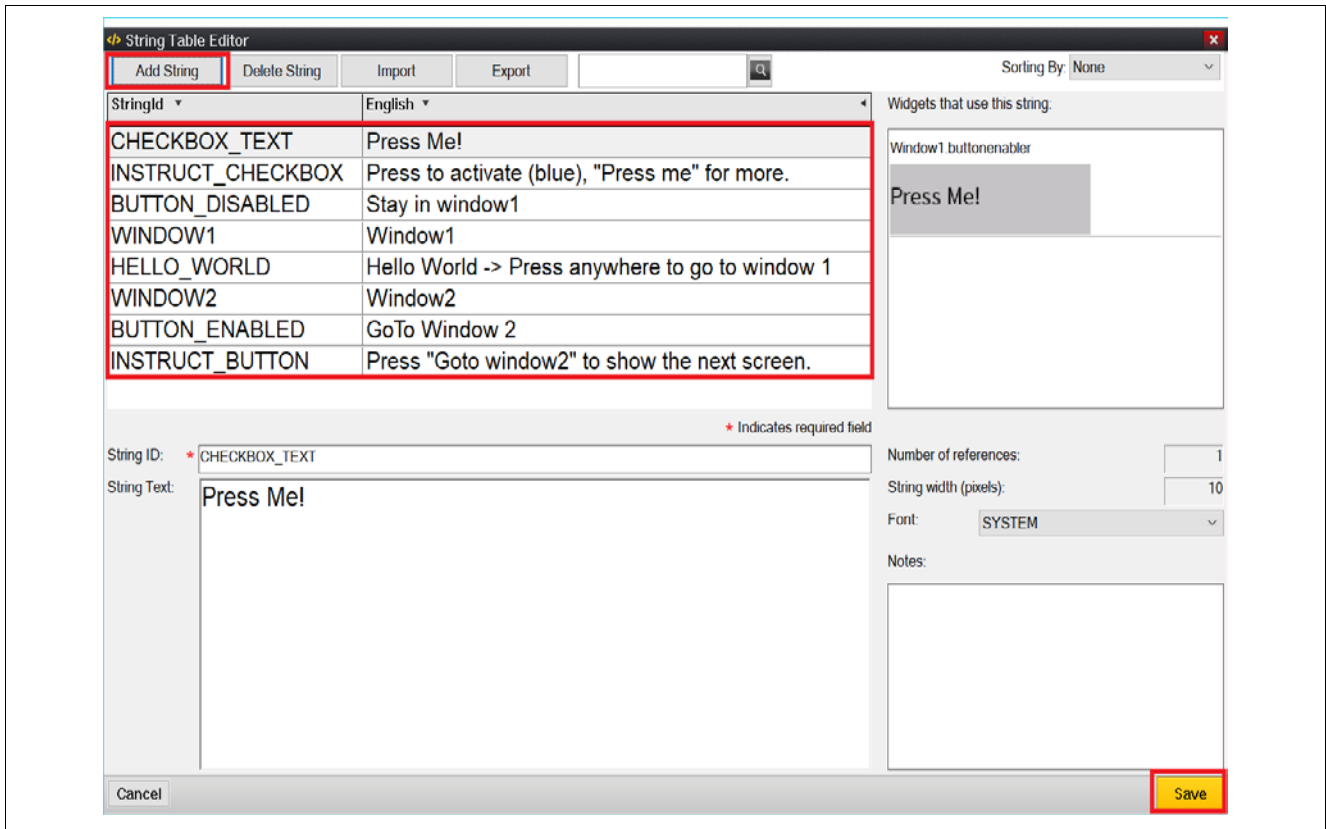
Figure 58. All Strings

5.8 Right-click on **Window1** to insert a **Text Button** and follow Figure 59.
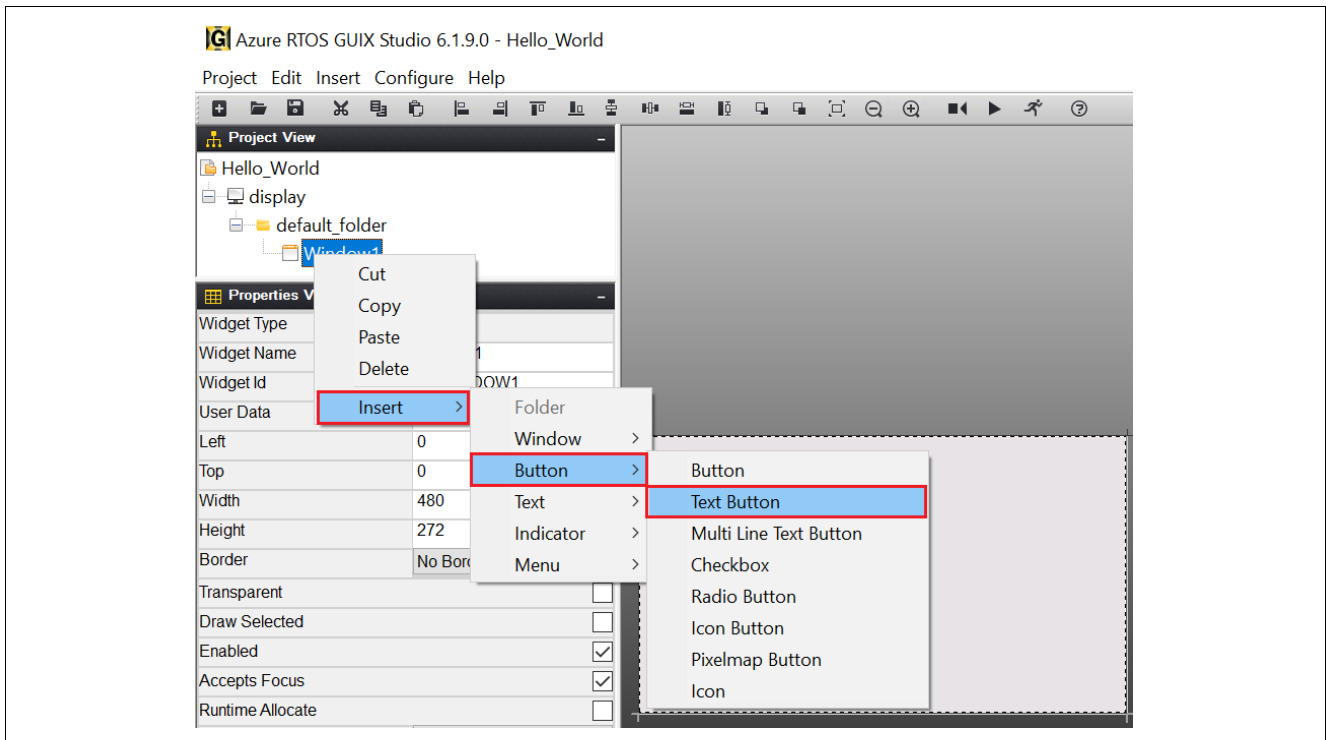
Figure 59. Insert Text Button

7. Set **Properties View** of "**ext_button**.
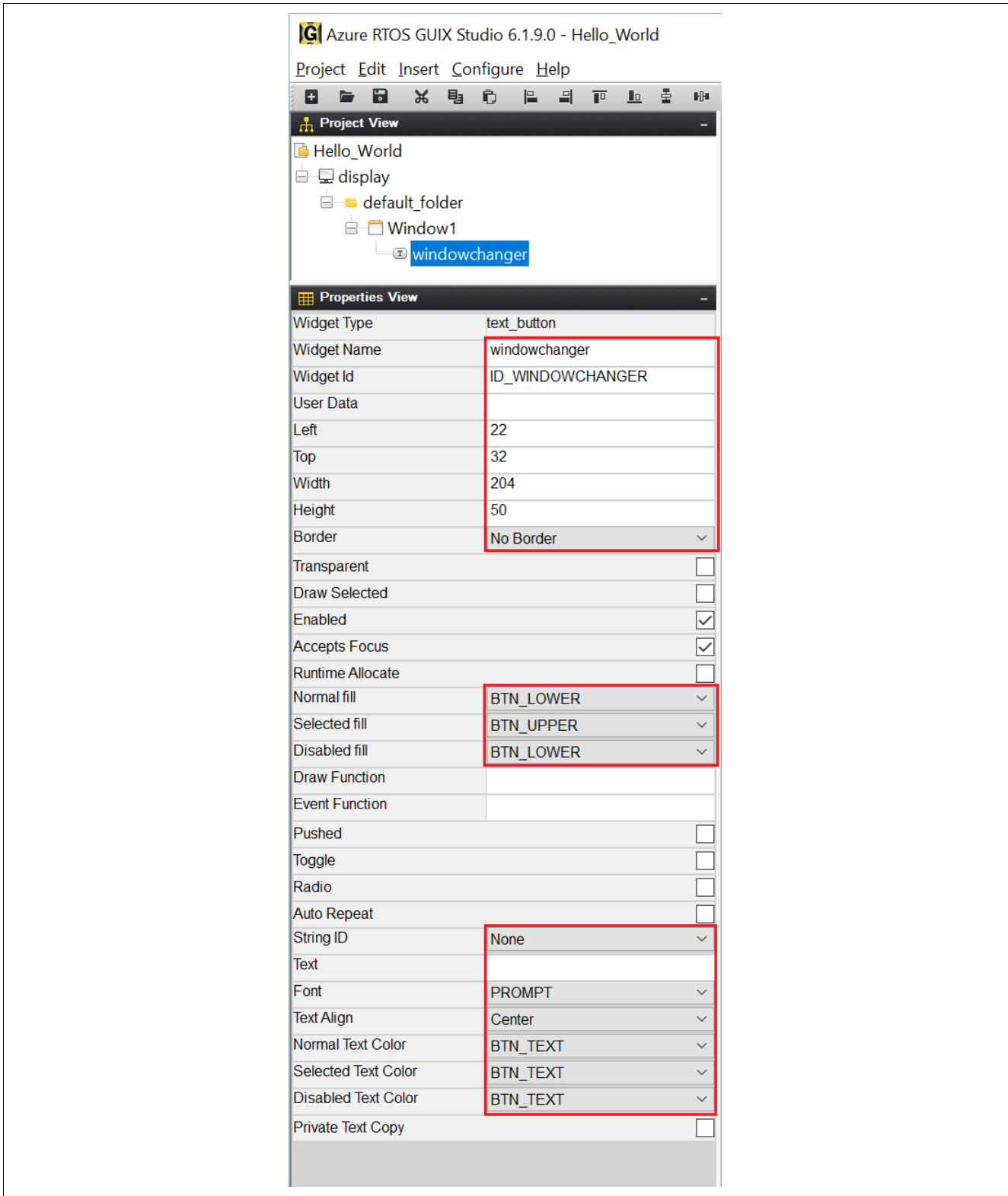
Figure 60. Properties View of text_button

8.   Right-click **windowchanger** to insert a **Prompt** and follow Figure 61.
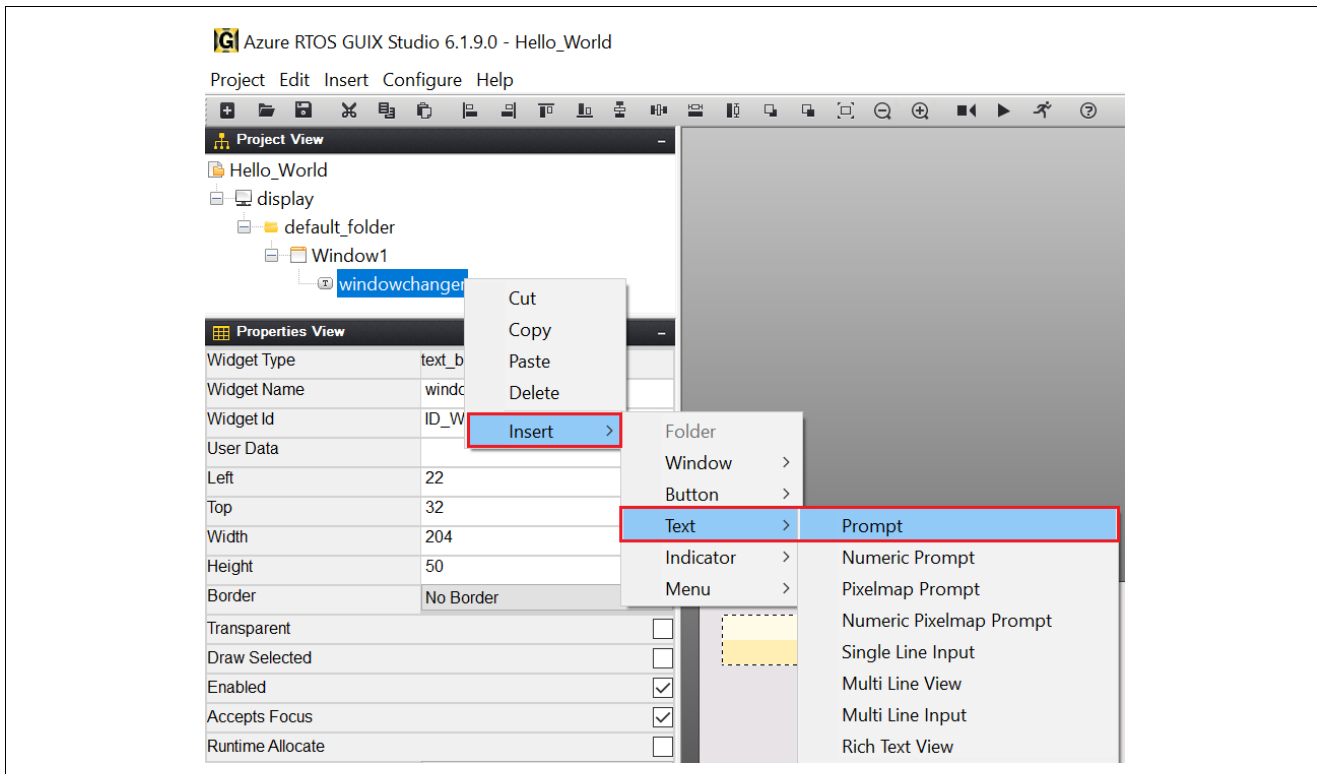
Figure 61.   Insert Prompt

9. Set **Properties View** of **prompt**.
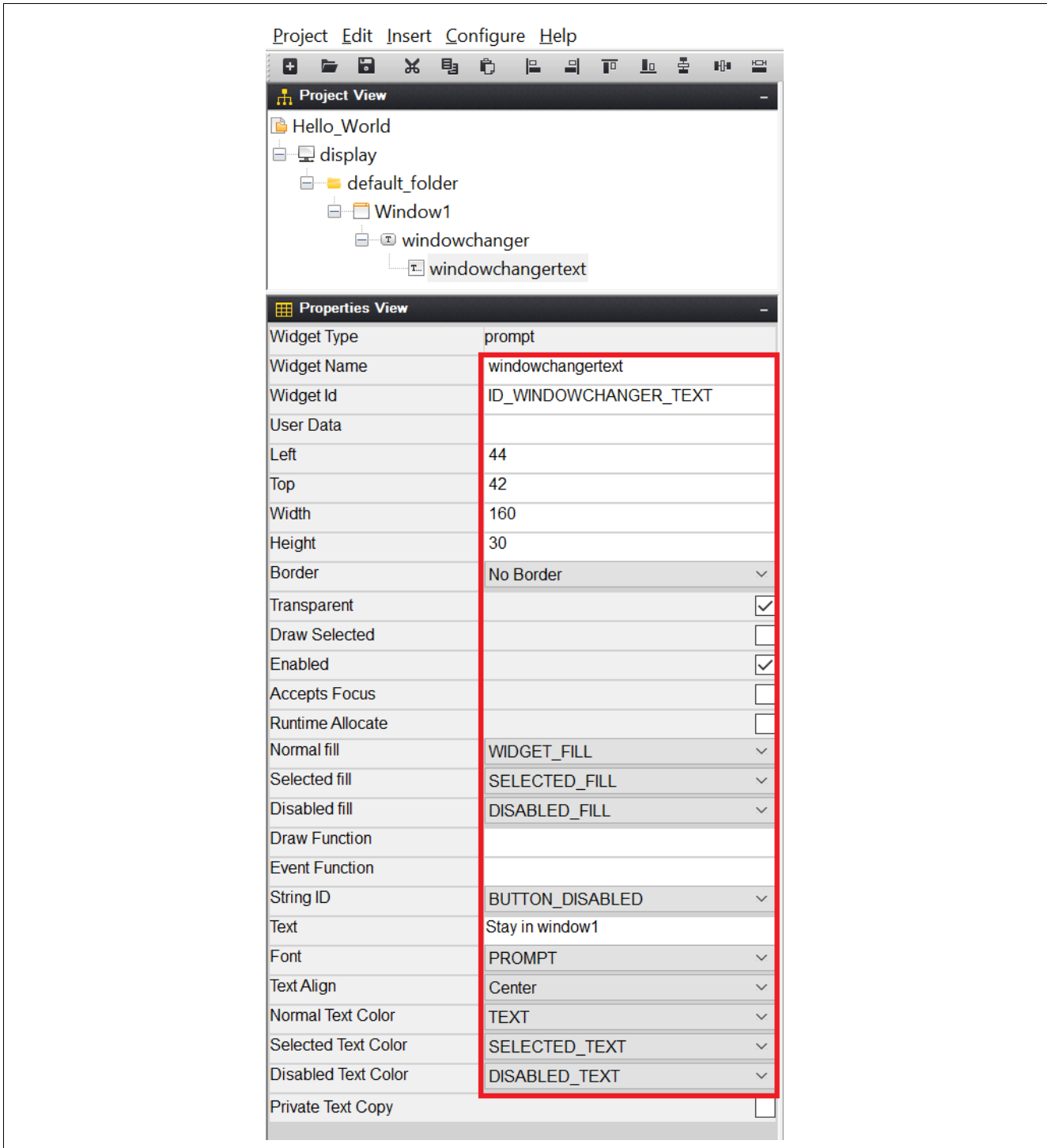
Figure 62. Properties View of prompt

10. Insert new **Button**. Right-click on **windowchangertext** and follow Figure 63.
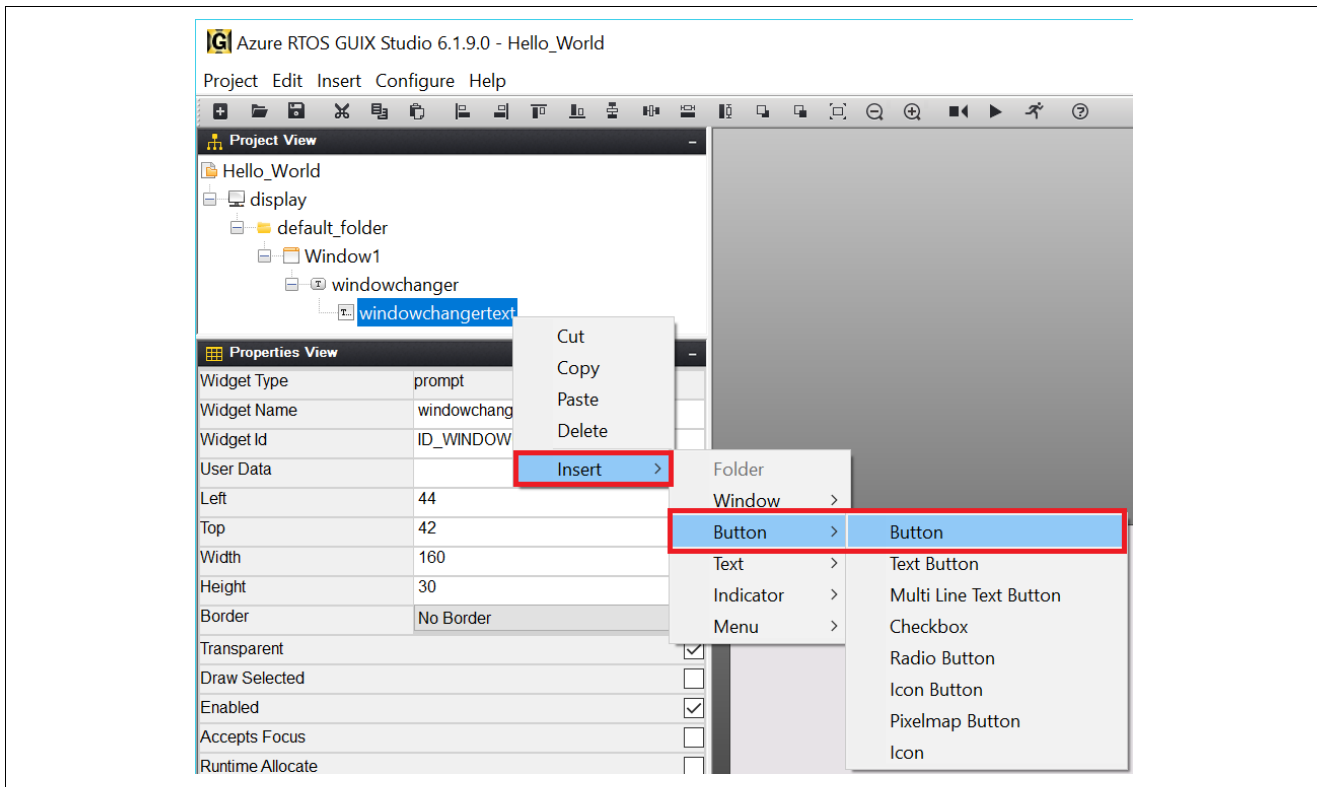
Figure 63. Insert Button

11. Set the **Properties View** of **button**.
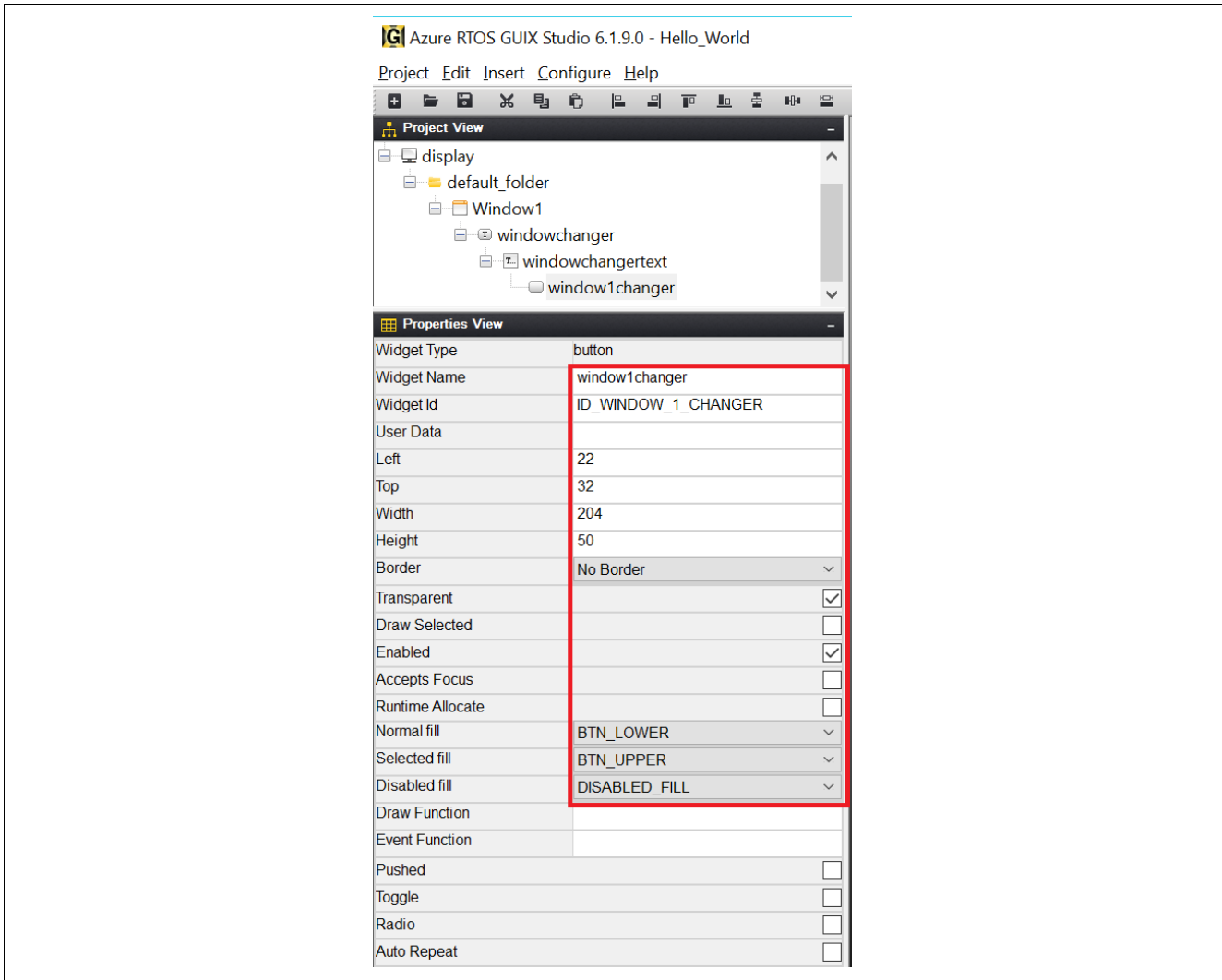
Figure 64. Properties View of button

12. Insert **Checkbox**. Right-click on **Window1** and follow Figure 65.
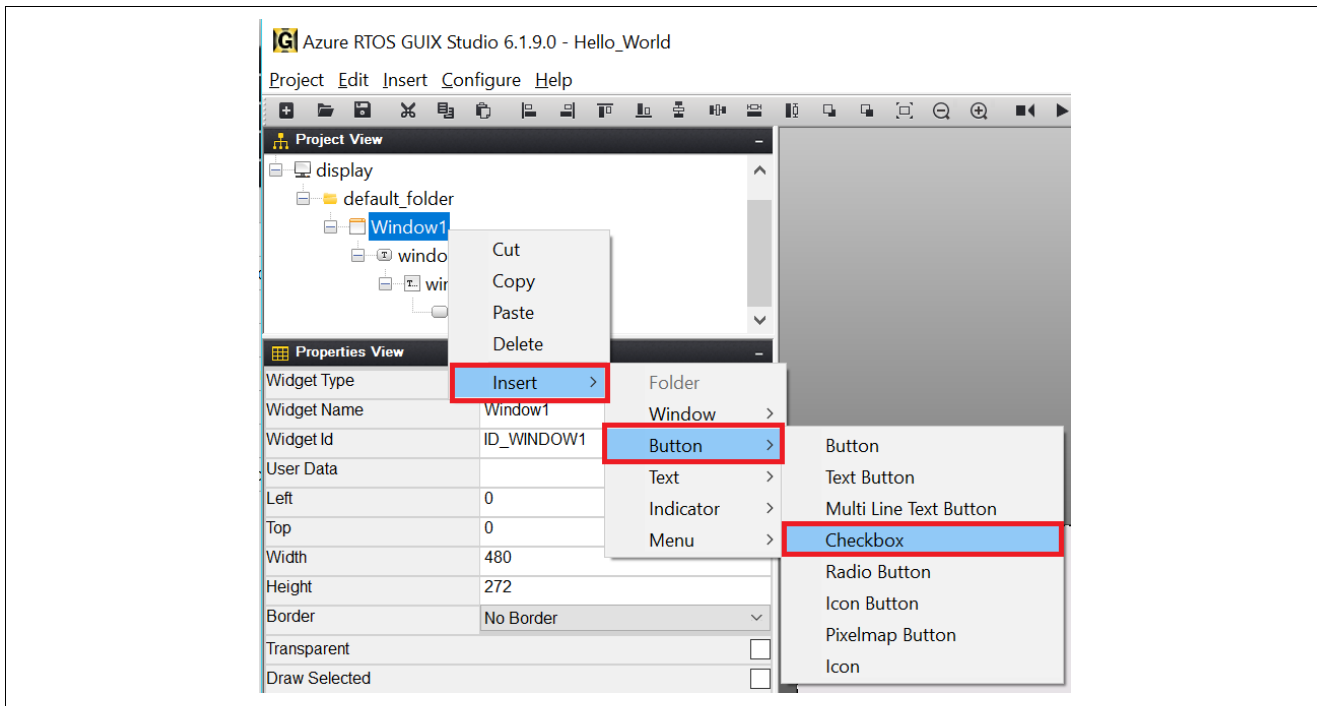
Figure 65.   Insert Checkbox

13. Set the **Properties View** of **button**.
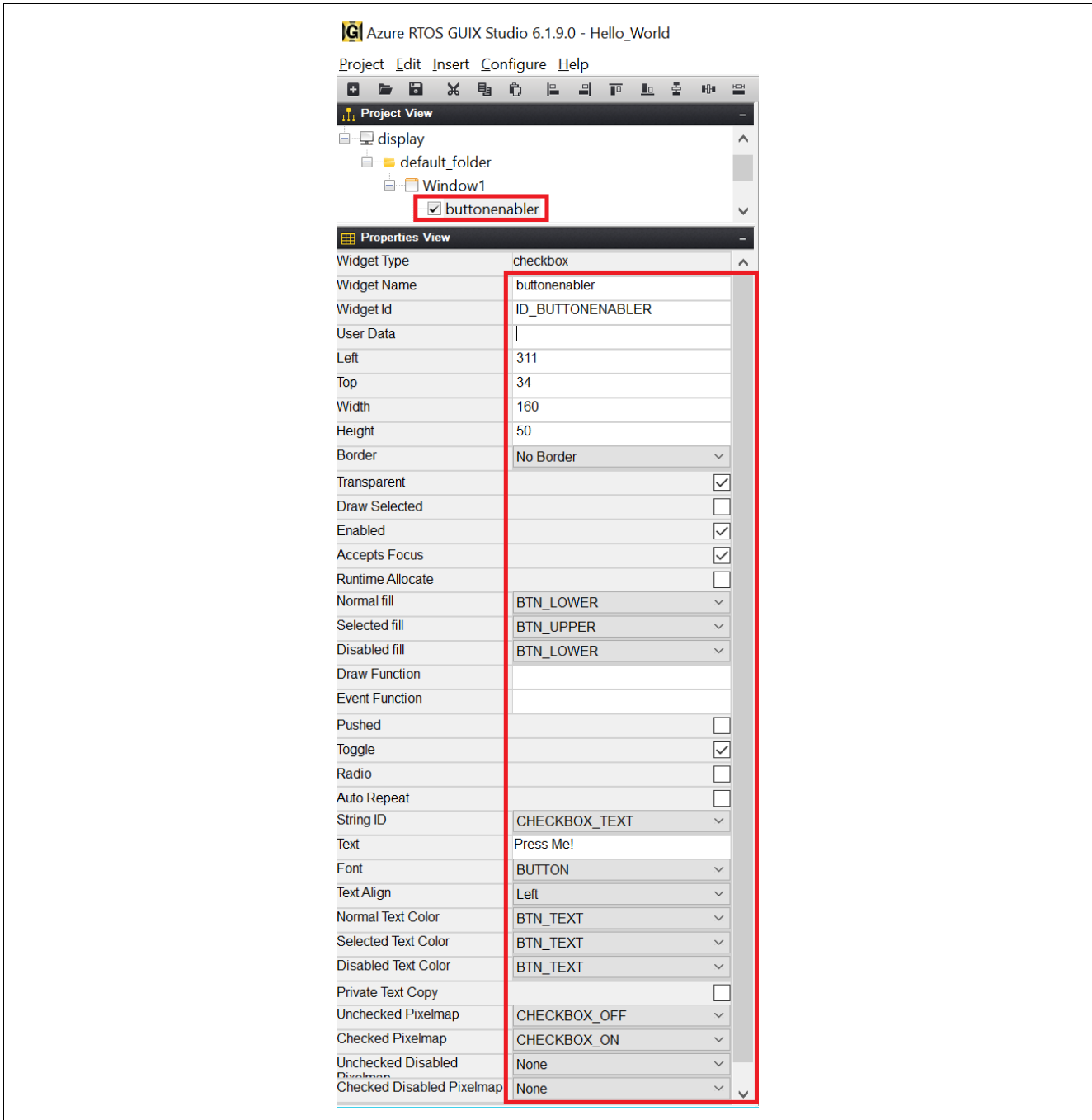
Figure 66. Properties View of button

14. Insert **Prompt**. Right click on **Window1** and follow Figure 67. Insert two times to get two prompts. **Prompt** and **Prompt1** will show up once you finish.
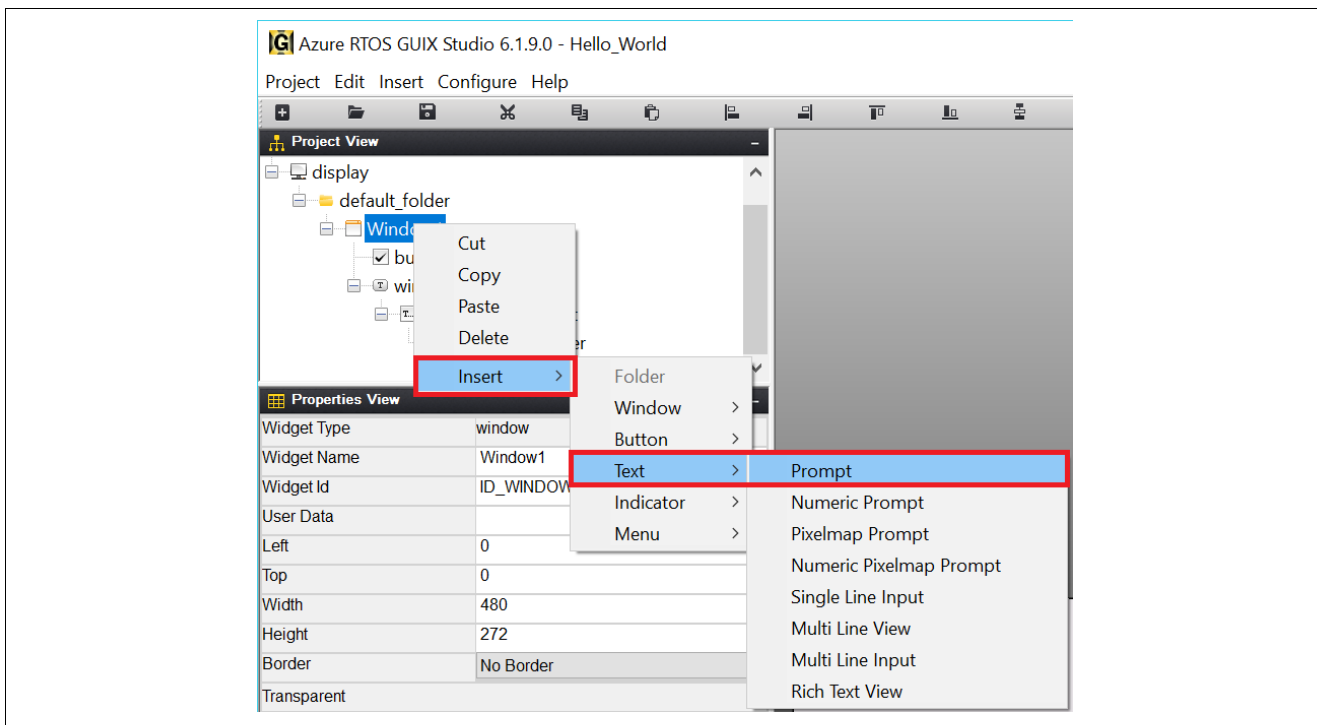
Figure 67. Insert Prompts

15. Set the **Properties View** of **Prompt**.
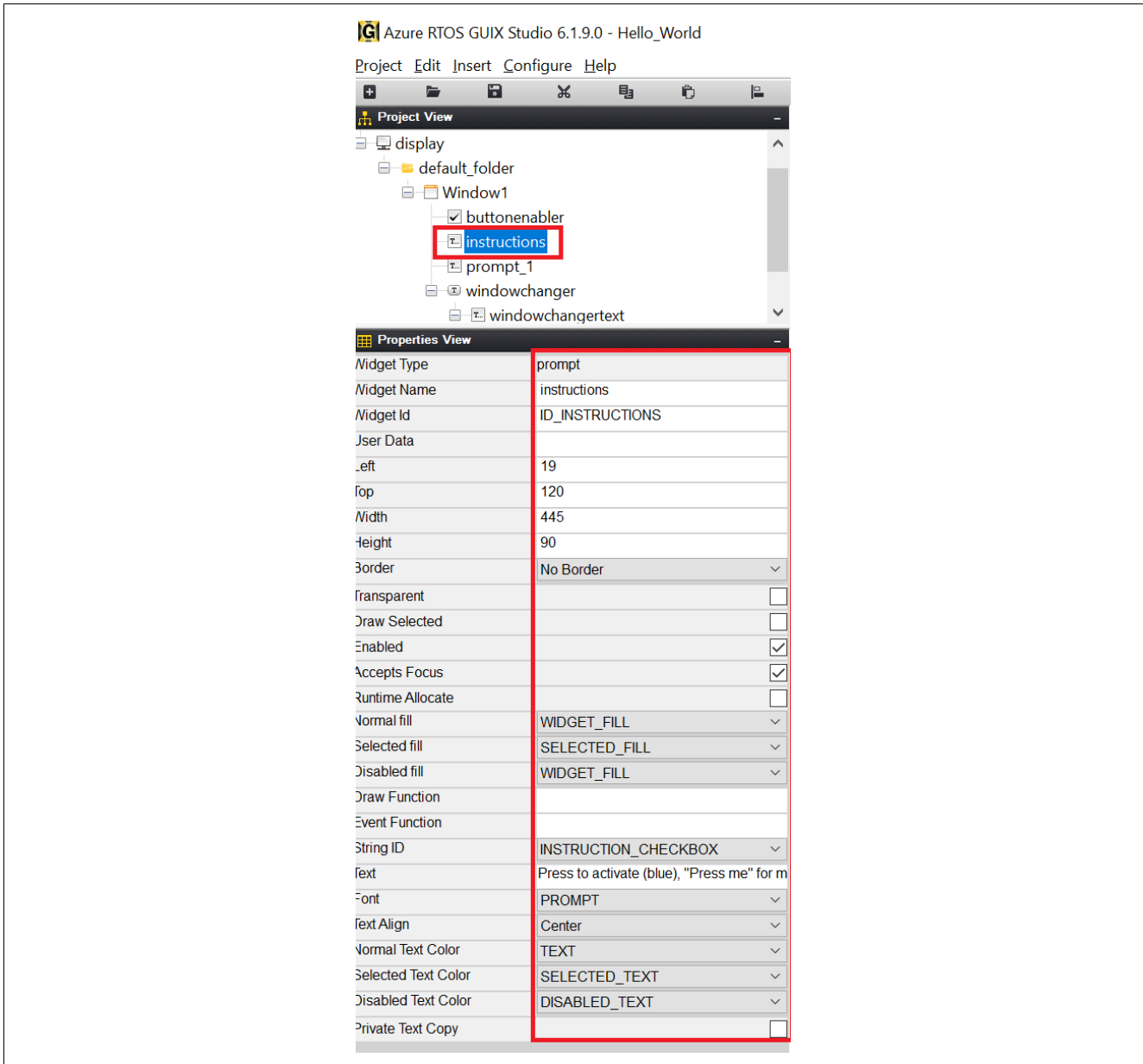
Figure 68. Properties View of Prompt

16. Set the **Properties View** of **Prompt1**.
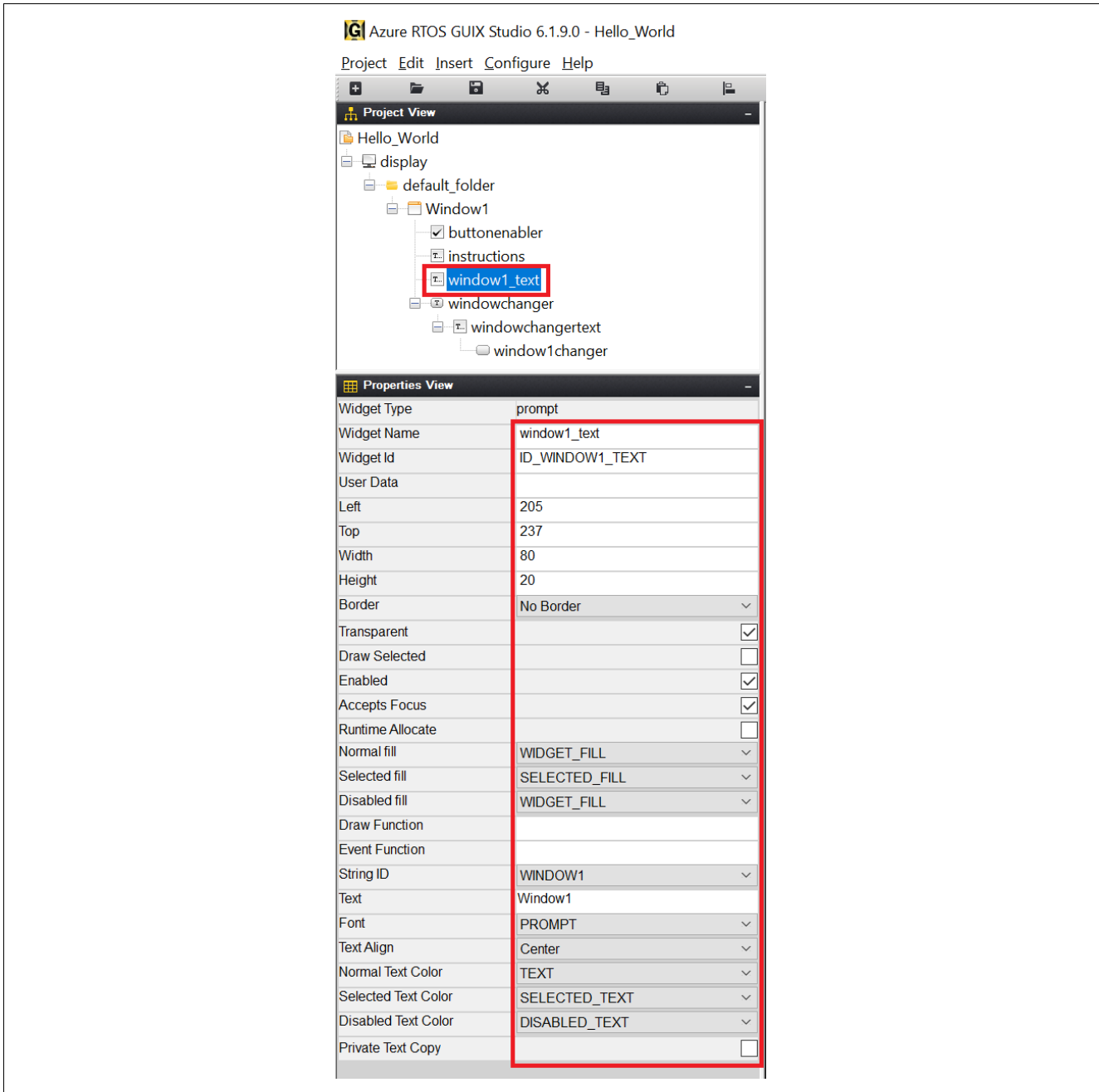
Figure 69.   Properties View for Prompt1

17. After you finished creating **Window1**, it should be like the image below.
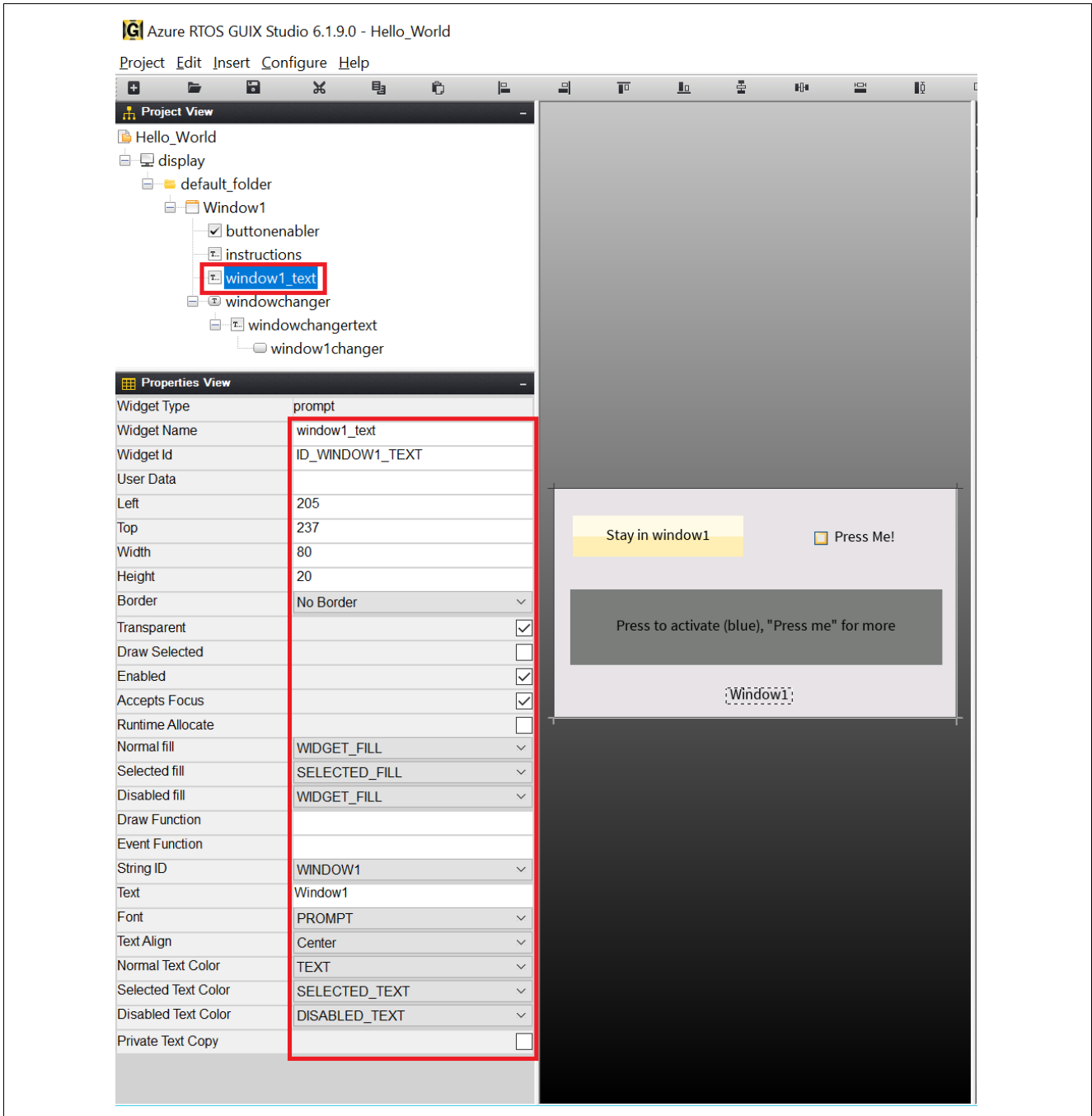
Figure 70. Window1 Created

18. Follow all these concepts to create **Window2**. After you are done, then you should have **Window2** as shown in Figure 71.
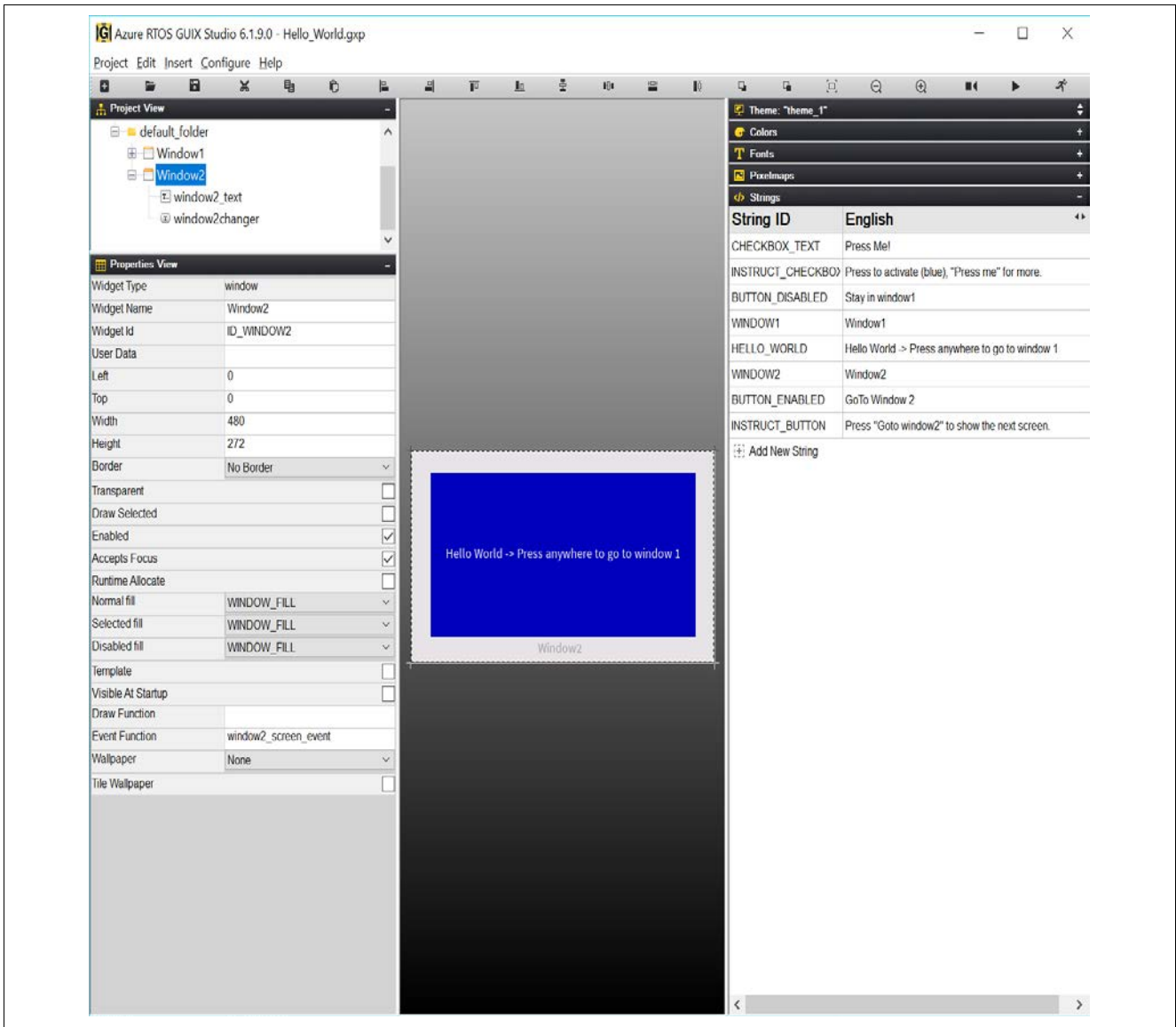
Figure 71.   Window2

19. Click on drop-down list **Pixelmaps**, double-click on **CHECKBOX_OF**" and a new window will pop up. Uncheck **Compress Output** then click **Save**. Do the same for **CHECKBOX_ON**.
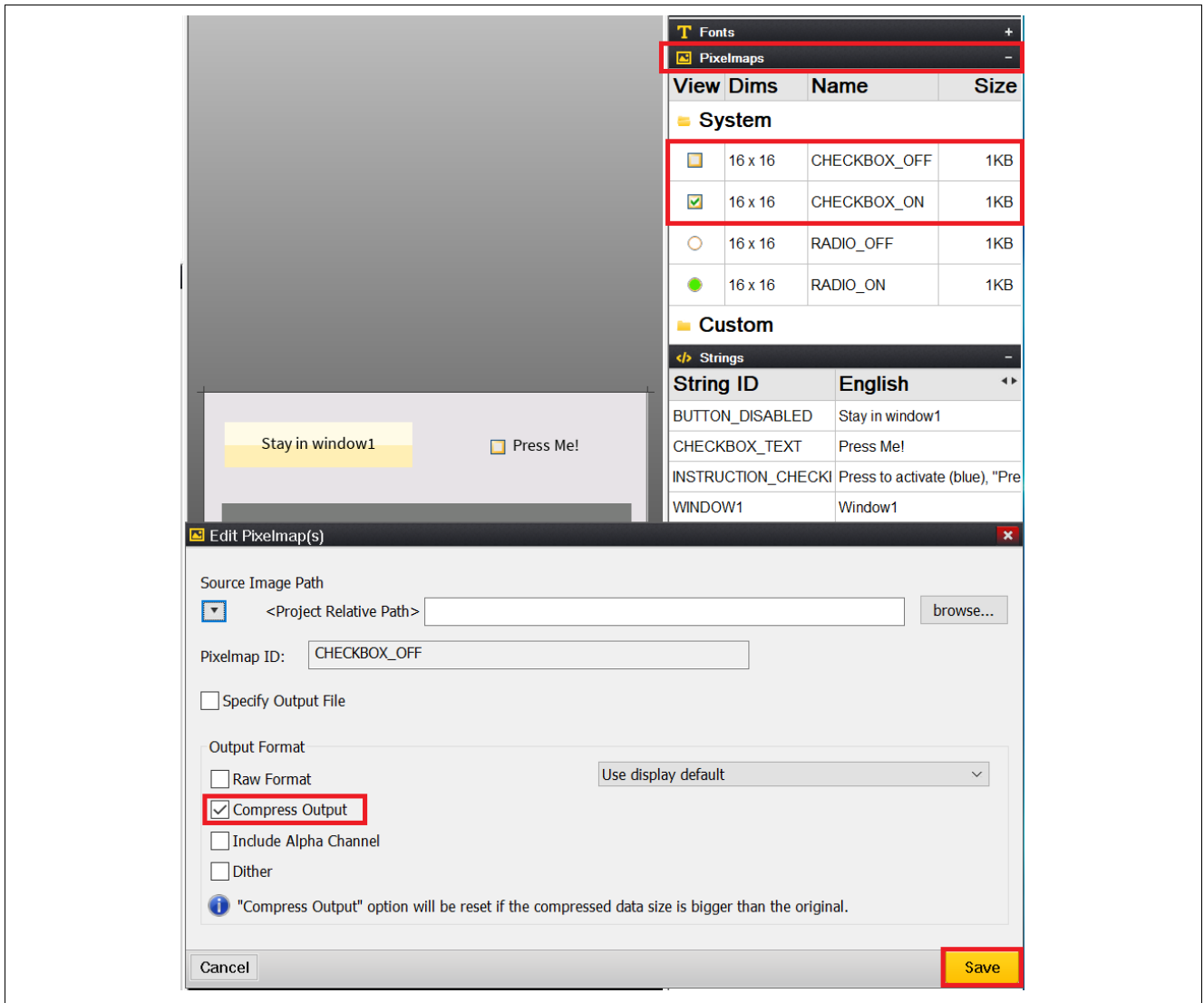
Figure 72.   Set Up Pixelmap

20. Now you can click on the **Project** drop down list, **Save Project**, and **Generate All Output Files**. You completed the process of creating and exporting GUIX Hello World into the project.
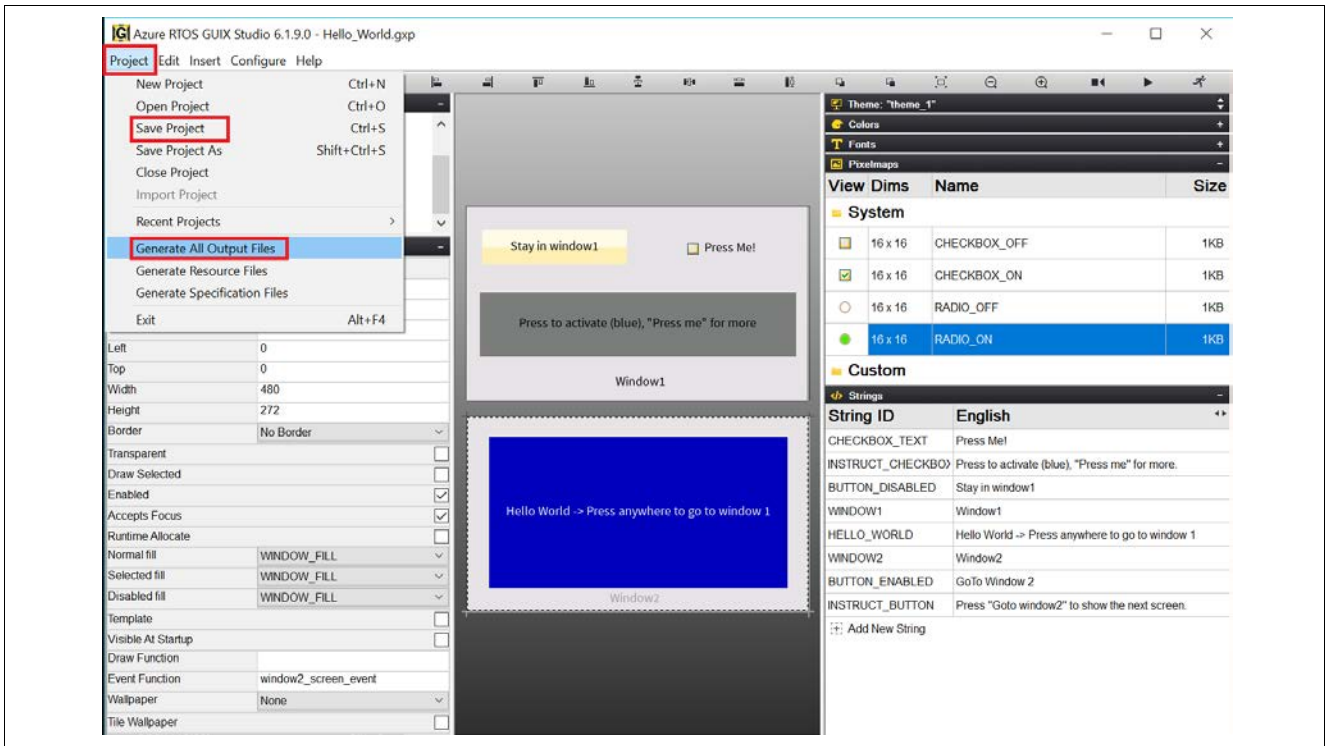
Figure 73. Save and Generate Project

21. Make sure the project is active and click to build the project. It may take a long time to finish building an Azure RTOS/GUIX project on your PC.
Project Hello_World_GUIX_EK_RA6M3G should build with no errors or warnings.
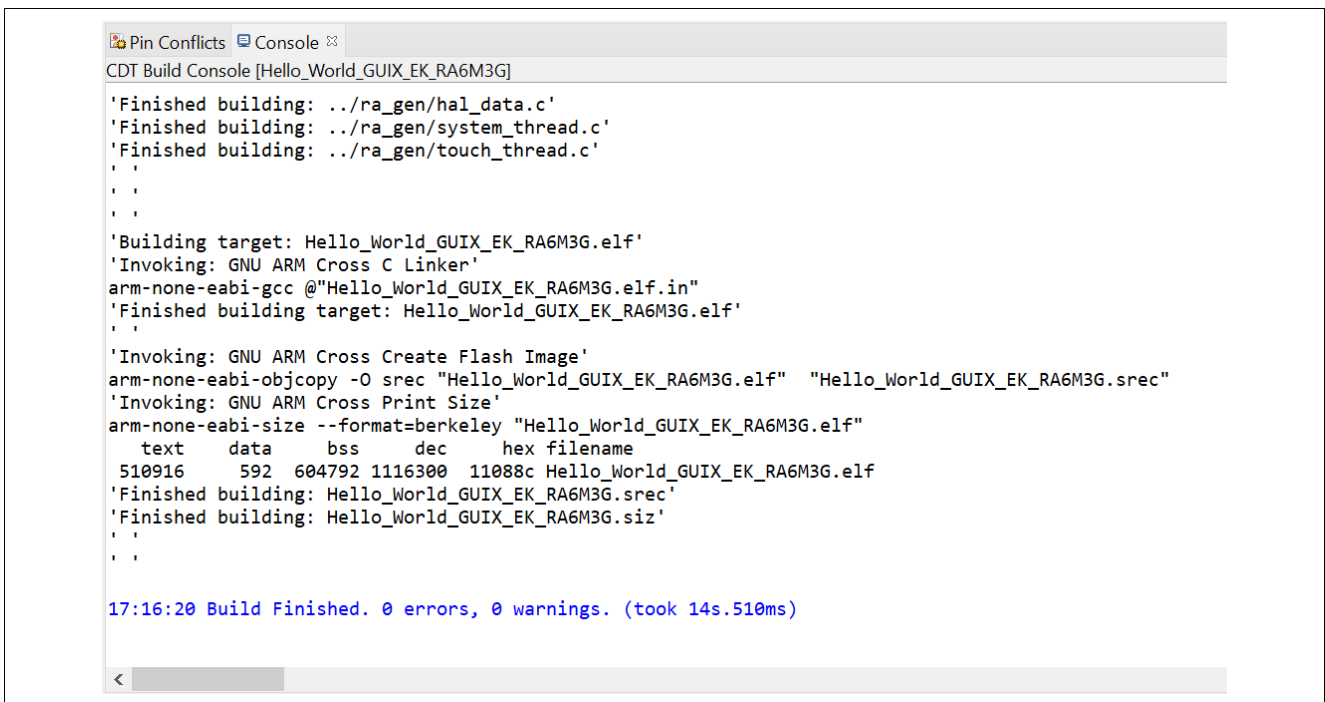
Figure 74. Project Builds Without Errors

22. Using the Micro USB cable, connect to J10 on EK-RA6M3G board and the other end to your PC. Download and run the Hello_World_GUIX_EK_RA6M3G project.

23. You will see the Window1 display on LCD panel.
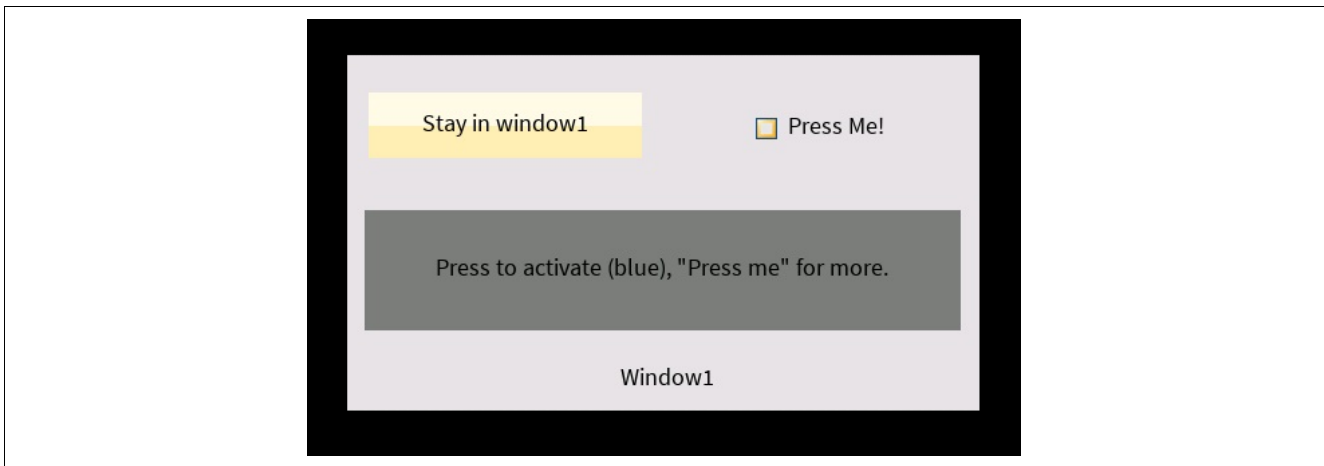
Figure 75. Window1 Display

6.   Overview of Full Function Project

6.1   Overview

In this section, you will import and run the complete Hello_World_GUI_EK_RA6M3G project. You can enable or disable check box function. The text on the button, which is "Stay in window1" or "Go to Window 2", will be updated. Once you press the button, the screen will change from window1 to window2 . Follow the text message on the screen, you can change from window2 back to window1.

6.2   Procedural Steps

1. You can try the completed project **Hello_World_GUI_EK_RA6M3G** for the full function application. Use the **Rename & Import Existing C/C++ Project into Workspace** feature of the **Import** menu to do so since you already had a project with the same in the workspace.
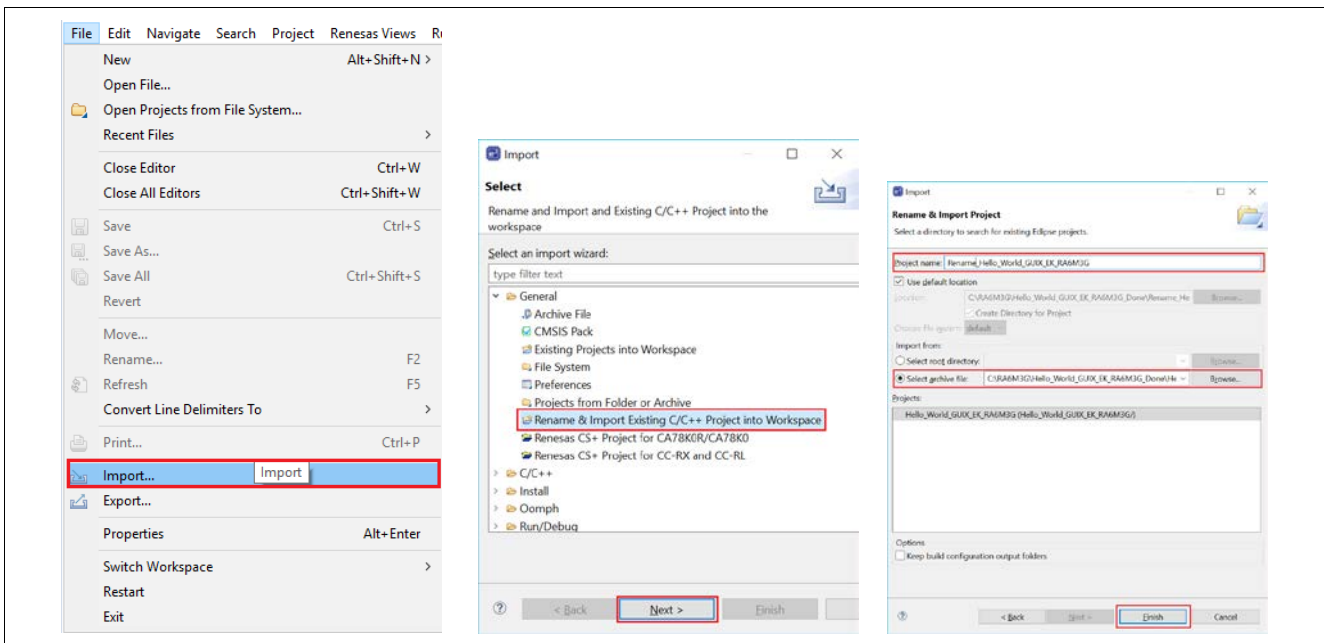
Figure 76.   Import Existing Project

Website and Support

Visit the following vanity URLs to learn about key elements of the RA family, download components and related documentation, and get support.

RA Product Information	www.renesas.com/ra
RA Product Support Forum	www.renesas.com/ra/forum
RA Flexible Software Package	www.renesas.com/FSP
Renesas Support	www.renesas.com/support

Revision History

| Rev. | Date | Description | |
		Page	Summary
1.00	Nov.12.21	—	First release document

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.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.

2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.

3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.

4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.

5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.

6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

    "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

    "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

    Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.

7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.

8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.

9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.

10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.

11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.

12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.

13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

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/.