RX Family

QE for Display Module Firmware Integration Technology

Introduction

This application note explains the QE for Display module, which uses Firmware Integration Technology (hereinafter, “FIT”). This module is used for GLCDC adjustment by using the standalone version of QE for Display[RX] (hereinafter, “QE for Display”).

This module is not necessary if you are using e^2 studio and the plug-in version of QE for Display[RX].

Because this is module is specifically for GLCDC adjustment, it is recommended that you either uninstall this module after adjustment, or that you provide a project to be used specifically for adjustment.

For details about QE for Display[RX], see our company website.

- QE for Display: Development Assistance Tool for Display Applications | Renesas

Supported Environments

Development environments using the standalone version of QE for Display[RX]

(assuming that Renesas CS+ and IAR EWRX are used)

Target Devices

- RX651 and RX65N groups (ROM capacity: 1.5 MB to 2 MB)
- RX66N group
- RX72N group
- RX72M group

When applying this application note to another microcontroller, modify the relevant details according to the microcontroller’s specifications, and conduct a thorough evaluation.

Target Compilers

- Renesas Electronics C/C++ Compiler Package for RX Family
- GCC for Renesas RX
- IAR C/C++ Compiler for Renesas RX

For details about verifying the operation of each compiler, see 4.3 Confirmed Operation Environment.

Related Documents

- Firmware Integration Technology User’s Manual (R01AN1833)
- Board Support Package Module Using Firmware Integration Technology (R01AN1685)

Contents

1. Overview

1.1 What is the QE for Display FIT Module?

This module can be used to communicate with the standalone version of QE for Display[RX] to adjust timing for a GLCDC or to correct image quality.

1.2 API Functions in the QE for Display FIT Module

Table 1.1 shows the API functions included in this module.

Table 1.1 List of API Functions

Function	Description
R_QE_DISPLAY_Open	This initializes the GLCDC or initializes serial communication with QE Display.
R_QE_DISPLAY_Exec	This executes GLCDC adjustment processing or error processing based on data received from QE Display.
R_QE_DISPLAY_GetVersion	This reads the version of this module.

1.3 Restrictions

The following restrictions apply to this module:

- For the GLCDC FIT module, use Rev1.60 or later. For the SCI FIT module, use Rev5.20 or later.
- If the color depth setting is within 1 to 8 bits, it will be overwritten even if specified in GLCDC FIT. This is because the CLUT table is used not only by this module, but also by the emWin, Aeropoint GUI FIT, and other modules.
- When using this module in combination with the emWin FIT or Aeropoint GUI FIT module, perform adjustments only when the on-screen display is still. Adjustments cannot be performed correctly during double buffering, for example when an animation is playing.

1.4 Software Configuration

Figure 1.1 shows the software configuration of this module.

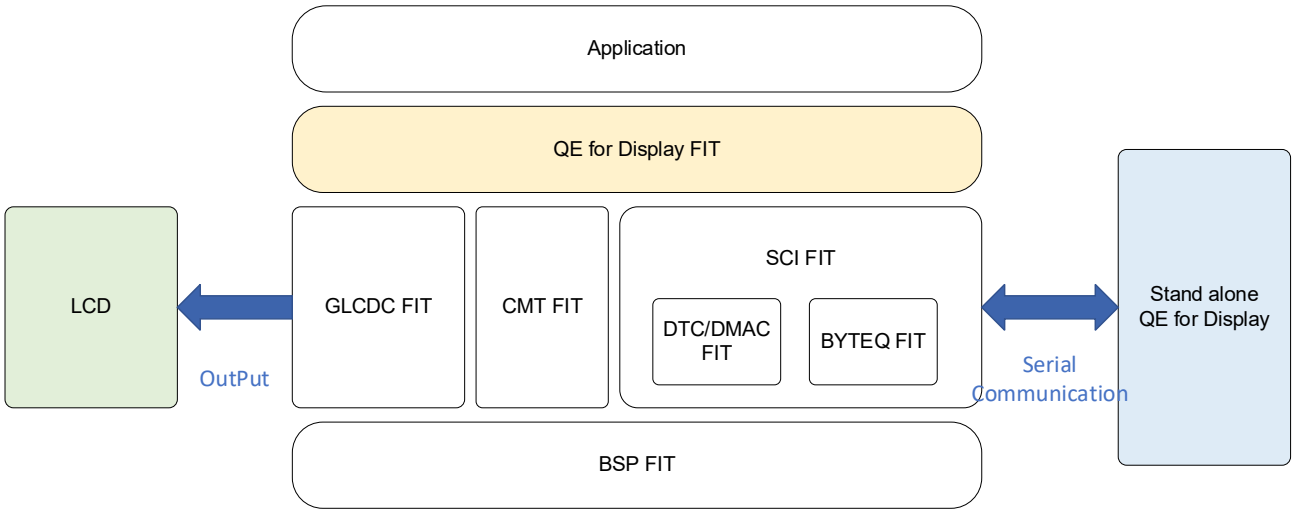


Figure 1.1 Software Configuration

1.5 Basic Operation

This module responds to connection requests and commands from QE for Display via serial communication. Before performing connection from QE for Display, connect the PC and RX devices and execute the program for the RX device.

For details about the procedure, see the QE for Display manual.

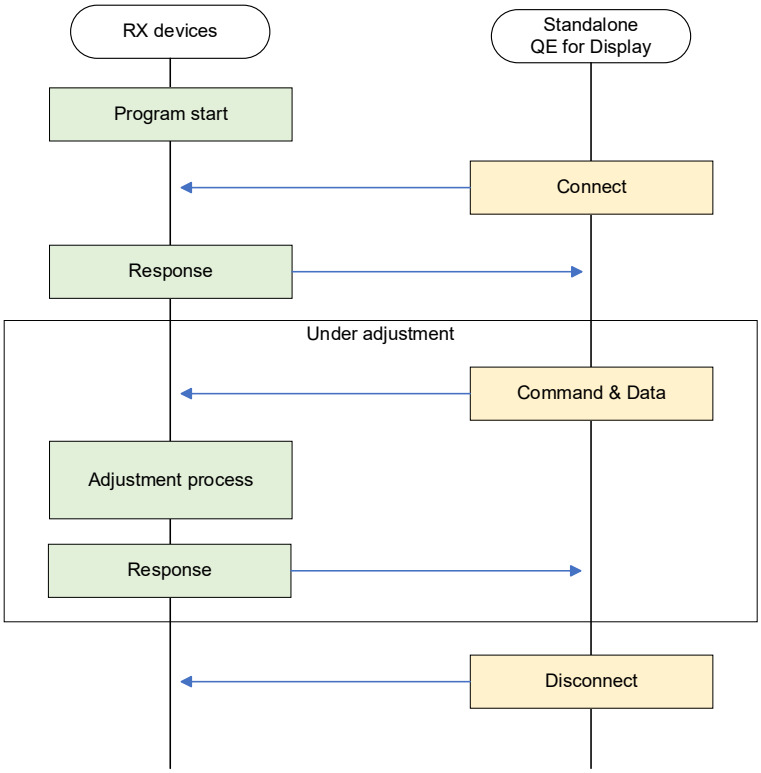


Figure 1.2 Overview of Basic Operation

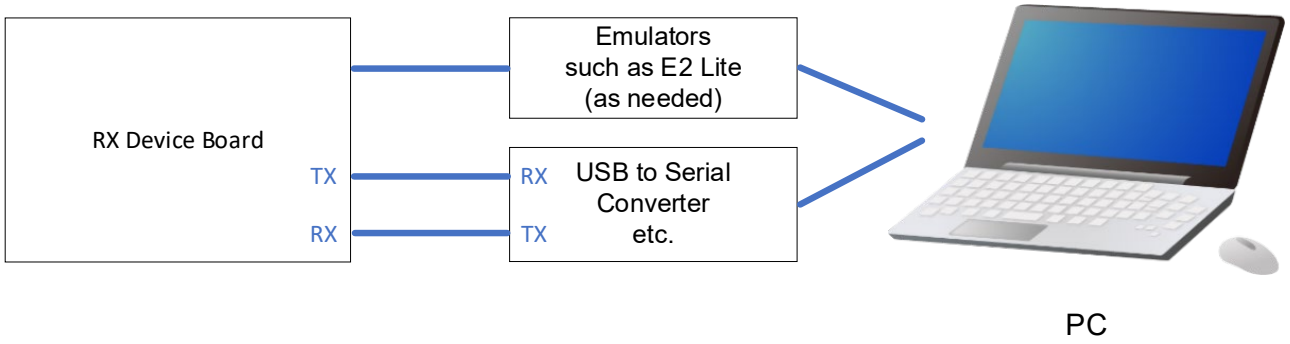


Figure 1.3 Connecting with the PC

Table 1.2 Communication Settings

Setting	Description
Serial-communication method	Asynchronous Mode(UART)
Baud rate (bps)	115200 bps (default) This can be changed in the configuration options. For details, see 2.6 Compile Settings.
Data length	8 bits
Parity	Disabled
Stop bits	1 bit
Flow control	Disabled

2. API Information

This FIT module has been verified to work under the following conditions.

2.1 Hardware Requirements

The MCU you are using must support the following functions:

- Graphic LCD Controller (GLCDC)
- Serial Communication Interface (SCI)
- Compare Match Timer (CMT)

2.2 Software Requirements

This software depends on the following FIT modules:

- Board Support Package (r_bsp) v7.42 or later
- Graphic LCD Controller module (r_glcdc_rx) v1.60 or later
- SCI module (r_sci_rx) v5.20 or later
- CMT module (r_cmt_rx)
- GPIO module (r_gpio_rx)

2.3 Supported Toolchains

This FIT module has been verified to work with the toolchains shown in 4.3 Confirmed Operation Environment.

2.4 Header Files

All API calls and their supported interface definitions are listed in r_qe_display_rx_if.h.

2.5 Integer Types

This driver uses ANSI C99. These types are defined in stdint.h.

2.6 Configuration Overview

Specify the configuration options for this module in r_qe_display_rx_config.h.

The following table explains the option names and setting values:

Configuration options in r_qe_display_rx_config.h	
QE_DISPLAY_CFG_UART_CH Note: The default value is “1”.	This specifies the SCI channel number to be used for communication with QE for Display. The specifiable range depends on the device you are using.
QE_DISPLAY_CFG_UART_BAUDRATE Note: The default value is “115200”.	This specifies the SCI Baud rate to be used for communication with QE for Display. This must match the value specified in QE for Display. The specifiable range depends on the device you are using.
QE_DISPLAY_CFG_UART_INTERRUPT_PRIORITY_LEVEL Note: The default value is “5”.	This specifies the SCI interrupt priority level to be used for communication with QE for Display. Specifiable range: 0 to 15
QE_DISPLAY_CFG_DEBUG_PRINT_ENABLE Note: The default value is “0”.	This selects whether to send communication and adjustment error messages to the standard output destination (usually the development-environment console currently connected to the debugger). “0”: Error messages are not sent. “1”: Error messages are sent. Because this uses the printf function, specify “0” in environments that cannot use standard output (for example, environments not connected to a debugger). Also, since this may affect communication with QE for Display, normally set this to “0”.

Also, when using this module, define the macro “QE_DISPLAY_CONFIGURATION” from the project settings in the development environment you are using.

If you build a project without this macro defined, this module outputs an error.

When using this module in combination with the emWin FIT or Aeropoint GUI FIT module, also define the following macros:

emWin: “QE_EMWIN_CONFIGURATION”

Aeropoint GUI: “QE_AEROPOINT_CONFIGURATION”

2.7 Code Size

The following table shows this module's ROM size, RAM size, and maximum available stack size. Values for RX72N are listed as representative examples.

The sizes of ROM (code and constants) and RAM (global data) are determined by the configuration options in 2.6 Compile Settings at the time of building.

The values in the following table have been confirmed under the following conditions:

Module revision: r_qe_display_rx rev1.10

Compiler version: Renesas Electronics C/C++ Compiler Package for RX Family V3.06.00
("-lang = c99" option added to the default settings for the integrated development environment)

GCC for Renesas RX 8.3.0.202405
("-std=gnu99" option added to the default settings for the integrated development environment)

IAR C/C++ Compiler for Renesas RX version 5.10.1
(Default settings for the integrated development environment)

Configuration options: Default settings

ROM, RAM, and stack code size

Device	Category	Memory Used		
		Renesas Compiler	GCC	IAR Compiler
RX72N	ROM	7075 bytes	9438 bytes	9953 bytes
	RAM	3284 bytes	3273 bytes	3140 bytes
	Stack	276 bytes	244 bytes	360 bytes

2.8 Return Values

This section shows the return values of API functions. This enumerated data type is included in r_qe_display_rx_if.h along with the prototype declaration of the API function.

```
/* Return values */
typedef enum e_qe_display_err
{
  QE_DISPLAY_SUCCESS = 0,           // Ended successfully
  QE_DISPLAY_ERR_NOT_OPEN,          // Open function not executed
  QE_DISPLAY_ERR_ALREADY_OPENED,    // Open function already executed
  QE_DISPLAY_ERR_LCD_CONFIG,        // LCD configuration error
  QE_DISPLAY_ERR_LCD_OTHER,         // LCD other error
  QE_DISPLAY_ERR_UART_CONFIG,       // Serial-communication configuration error
  QE_DISPLAY_ERR_UART_OTHER,        // Serial-communication other error
} qe_display_err_t;
```

2.9 Adding a FIT Module

This module must be added to each project that uses it. Renesas recommends adding this module by using methods (1) and (3), which use Smart Configurator. However, note that Smart Configurator does not support all RX devices. For unsupported RX devices, use method (2).

(1) Adding the FIT module by using Smart Configurator in CS+
In CS+, use the standalone version of Smart Configurator to automatically add the FIT module to the user project. For details, see the application note “RX Smart Configurator User’s Guide: CS+ (R20AN0470)”.

(2) Adding the FIT module in CS+
In CS+, manually add the FIT module to the user project. For details, see the application note “RX Family Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)”.

(3) Adding the FIT module by using Smart Configurator in IAREW
Use the standalone version of Smart Configurator to automatically add the FIT module to the user project. For details, see the application note “RX Smart Configurator User’s Guide: IAREW (R20AN0535)”.

2.10 “for”, “while”, and “do while” Statements

FIT modules use “for”, “while”, and “do while” statements (loops) for processing such as waiting for data to be reflected to the register. These loops are annotated with comments using key words such as “WAIT_LOOP”. Therefore, if a user has integrated failsafe processing into a loop, “WAIT_LOOP” can be used to search for the processing in question.

The following are examples of such commented code:

```
Example “while” statement:
/* WAIT_LOOP */
while(0 == SYSTEM.OSCOVFSR.BIT.PLOVF)
{
    /* The delay period needed is to make sure that the PLL has stabilized. */
}

Example “for” statement:
/* Initialize reference counters to 0. */
/* WAIT_LOOP */
for (i = 0; i < BSP_REG_PROTECT_TOTAL_ITEMS; i++)
{
    g_protect_counters[i] = 0;
}

Example “do while” statement:
/* Reset completion waiting */
do
{
    reg = phy_read(ether_channel, PHY_REG_CONTROL);
    count++;
} while ((reg & PHY_CONTROL_RESET) && (count < ETHER_CFG_PHY_DELAY_RESET)); /* WAIT_LOOP */
```

3. API Functions

3.1 R_QE_DISPLAY_Open ()

This function initializes the GLCDC or initializes serial communication with QE Display. This function must be executed before using other API functions.

Format

```
qe_display_err_t R_QE_DISPLAY_Open (void)
```

Parameters

None

Return Values

QE_DISPLAY_SUCCESS	*/* Ended successfully */*
QE_DISPLAY_ERR_ALREADY_OPENED	*/* Open function already executed */*
QE_DISPLAY_ERR_LCD_CONFIG	*/* LCD configuration error */*
QE_DISPLAY_ERR_UART_CONFIG	*/* Serial-communication configuration error */*

Properties

r_qe_display_rx_if.h contains the prototype declaration.

Description

Executing this function initializes the GLCDC or initializes serial communication with QE Display.

Example

See 4.1 Sample Code.

Special Notes:

To perform adjustment based on an image designed in the emWin FIT or Aeropoint GUI FIT module, first execute the corresponding initialization processing before executing this function.

When DMAC or DTC is used in the SCI FIT module, the initialization function for DMAC or the DTC FIT module (the R_DMAC_Init or R_DTC_Open function) is not executed in this module. Therefore, first execute the corresponding initialization function (R_DMAC_Init or R_DTC_Open function) before executing this function.

3.2 R_QE_DISPLAY_Exec ()

This function executes GLCDC adjustment processing or error processing based on data received from QE Display.

Format

```
qe_display_err_t R_QE_DISPLAY_Exec (void)
```

Parameters

None

Return Values

QE_DISPLAY_SUCCESS	*/* Ended successfully */*
QE_DISPLAY_ERR_NOT_OPEN	*/* Open function not executed */*
QE_DISPLAY_ERR_LCD_OTHER	*/* LCD other error */*
QE_DISPLAY_ERR_UART_OTHER	*/* Serial communication other error */*

Properties

r_qe_display_rx_if.h contains the prototype declaration.

Description

This executes GLCDC adjustment processing. Ensure that this function is executed repeatedly in the main routine.

Example

See 4.1 Sample Code.

Special Notes:

None.

3.3 R_QE_DISPLAY_GetVersion ()

This function returns the API version.

Format

```
uint32_t R_QE_DISPLAY_GetVersion (void)
```

Parameters

None

Return Values

Version number

Properties

r_qe_display_rx_if.h contains the prototype declaration.

Description

This function returns the currently installed version of this FIT module. The version number is encoded as follows: The first two bytes indicate the major version number, and the last two bytes indicate the minor version number. For example, for version 4.25, the return value is “0x00040019”.

Example

```
/* When acquiring the version of the FIT module */
volatile uint32_t   version;

version = R_QE_DISPLAY_GetVersion();
```

Special Notes:

None.

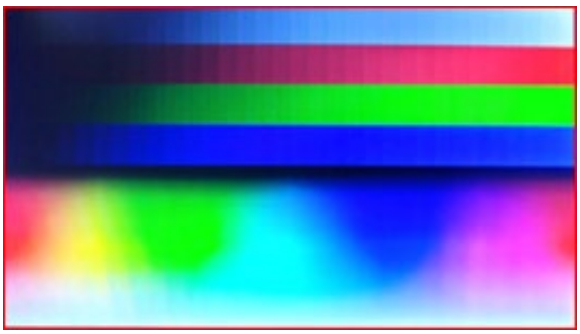
4. Appendix

4.1 Sample Code

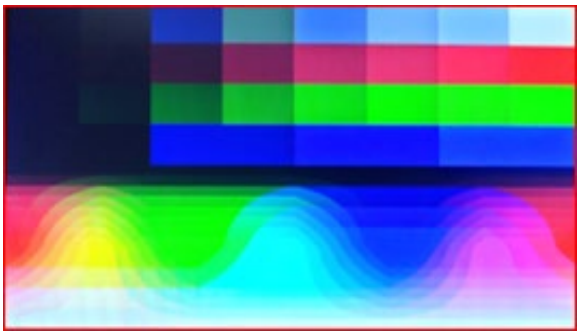
The following shows sample code for this module. When using only the GLCDC FIT module, see 4.1.1. When using the GLCDC FIT module in combination with the emWin FIT module, see 4.1.2. When using the GLCDC FIT module in combination with the Aeropoint GUI FIT module, see 4.1.3.

If you are using only the GLCDC FIT module, the following demo screens appear after the sample code is executed.

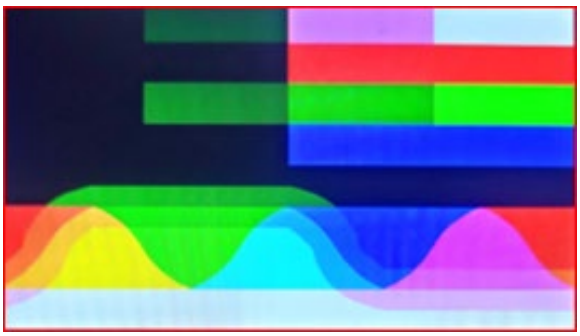
ARGB8888, RGB888, RGB565, ARGB1555, ARGB444



CLUT8



CLUT4



CLUT1

Figure 4.1 Demo Screens

4.1.1 Sample code when using only the GLCDC FIT module

```
#include "r_smc_entry.h"
#include "r_qe_display_rx_if.h"
#include "r_glcdc_rx_if.h"


/* Initialize a first time interrupt flag
 *   Unintended specified line notification from graphic 2 and
 *   graphic 1, 2 underflow is detected only
 *   for first time after release GLCDC software reset.
 *   This variable is a flag to skip the first time interrupt processing.
 *   Refer to Graphic LCD Controller (GLCDC) section of
 *   User's Manual: Hardware for details. */
bool first_interrupt_flag = false;


void main(void)
{
    qe_display_err_t ret;

   /* When using DMAC/DTC with uart, initialize the DMAC/DTC
    * before using the R_QE_Display_Open function.*/
   //R_DMACA_Init(); or R_DTC_Open();

    ret = R_QE_DISPLAY_Open();
    if(ret != QE_DISPLAY_SUCCESS)
    {
       while(1)
       {
          /* Check GLCDC and SCI setting */
       }
    }

   while(1)
   {
      R_QE_DISPLAY_Exec();
    }
}


/* If you use the GLCDC callback function,                          */
/* please enable the settings required for interrupt generation.    */
/* Ex. VPOS interrupt                                               */
/* - LCD_CH0_DETECT_VPOS            (true)                          */
/* - LCD_CH0_INTERRUPT_VPOS_ENABLE  (true)                          */
/* - LCD_CH0_CALLBACK_ENABLE        (true)                          */
/* - LCD_CH0_PCALLBACK           (my_glcdc_callback)                */

void my_glcdc_callback(void * pdata)
{
   if (false == first_interrupt_flag)
   {
      first_interrupt_flag = true;
      /* do nothing */
   }
   else
   {
      glcdc_callback_args_t * pdecode;
```

```
        pdecode = (glcdc_callback_args_t *)pdata;
        ...
    }
}
```

4.1.2 Sample code when using the emWin FIT module

```
#include "r_smc_entry.h"
#include "r_qe_display_rx_if.h"

/* 1. API programming */
#include “GUI.h”

void main(void)
{
   qe_display_err_t ret;

   /* When using DMAC/DTC with uart, initialize the DMAC/DTC
    * before using the R_QE_Display_Open function.
    * When using DMAC, initialization is not required
    * if it is initialized within the emWin FIT module. */
   //R_DMACA_Init(); or R_DTC_Open();

   /* initialize emwin */
   GUI_Init();

    /* ~~ emWin drawing user code ~~ */

   ret = R_QE_DISPLAY_Open();
   if(ret != QE_DISPLAY_SUCCESS)
   {
      while(1)
      {
         /* Check GLCDC and SCI setting */
      }
   }

   while(1)
   {
      GUI_Exec();
      R_QE_DISPLAY_Exec();
    }
}


/* 2. Use AppWizard */
#include "Generated/Resource.h"

void main(void)
{
    qe_display_err_t ret;

   /* When using DMAC/DTC with uart, initialize the DMAC/DTC
    * before using the R_QE_Display_Open function.
    * When using DMAC, initialization is not required
    * if it is initialized within the emWin FIT module. */
   //R_DMACA_Init(); or R_DTC_Open();

    APPW_X_Setup();
    APPW_Init(APPW_PROJECT_PATH);
    APPW_CreatePersistentScreens();
    APPW_CreateRoot(APPW_INITIAL_SCREEN, WM_HBKWIN);

   ret = R_QE_DISPLAY_Open();
```

```
    if(ret != QE_DISPLAY_SUCCESS)
    {
       while(1)
       {
          /* Check GLCDC and SCI setting */
       }
    }

     while (1) {
         while (GUI_Exec1()) {
             APPW_Exec();
         }
         APPW_Exec();
         GUI_Delay(5);
         R_QE_DISPLAY_Exec();
     }
}
```

4.1.3 Sample code when using the Aeropoint GUI FIT module

```
#include <stdio.h>
#include "r_smc_entry.h"
#include "cri_aero_config_ais.h"
#include "cri_aero_player.h"
#include "cri_aero_error.h"
#include "rx72n_env_test_GUI.h"
#include "r_qe_display_rx_if.h"

void main(void)
{
  qe_display_err_t ret;

  CriBool flag = CRI_TRUE;

    /* When using DMAC/DTC with uart, initialize the DMAC/DTC
     * before using the R_QE_Display_Open function.
     * When using DMAC, initialization is not required
     * if it is initialized within the Aeropoint GUI FIT module. */
    //R_DMACA_Init(); or R_DTC_Open();

  /* Initialize */
#if CRI_AERO_CONFIG_AIS_USE_SD || CRI_AERO_CONFIG_AIS_USE_FLASH
  /* When using slides in SD or external FLASH. */
  flag = CriAeroPlayer_Initialize(NULL);
#else
  /* When using slides in Memory. */
  flag = CriAeroPlayer_Initialize( &rx72n_env_test_GUI_gData );
#endif

  if(flag == CRI_FALSE)
  {
    CriUint32 errorNo = CriAeroError_GetLastError();
    printf("CriAeroPlayer_Initialize failed. errorNo:%d", errorNo);
    for(;;)
    {
    }
  }

#if 1
  /* Switch from title to slide 1. */
  /* You can also switch slides by LAN or UART command. */
  CriAeroPlayer_ReadSlide( 1 );
#endif

  ret = R_QE_DISPLAY_Open();
  if(QE_DISPLAY_SUCCESS != ret)
  {
      while(1)
      {
         /* Check GLCDC and SCI setting */
      }
   }

  /* Main loop */
  while (flag)
  {
    /* Main */
```

```
    flag = CriAeroPlayer_Main();
    R_QE_DISPLAY_Exec();
  }

  /* Finalize */
  CriAeroPlayer_Finalize();

  return;
}
```

4.2 Configuring the Development Environment (CS+, EWRX)

4.2.1 Configuring macro definitions

When using this module, you must define the macro “QE_DISPLAY_CONFIGURATION” for each development environment. This section shows how to define the macro “QE_DISPLAY_CONFIGURATION”.

When using this module in combination with the emWin FIT module, also define the macro “QE_EMWIN_CONFIGURATION”. When using this module in combination with the Aeropoint GUI FIT module, also define the macro “QE_AEROPOINT_CONFIGURATION”.

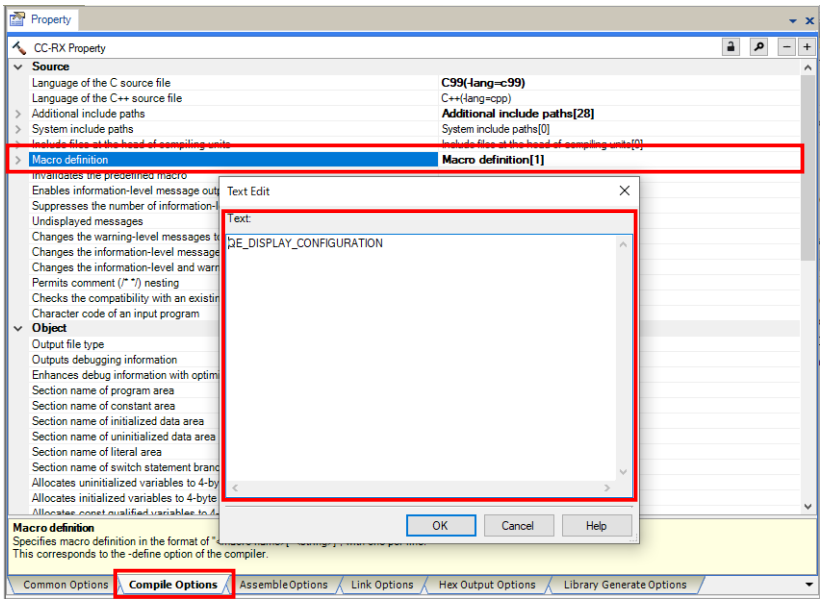


Figure 4.2 Configuration in CS+

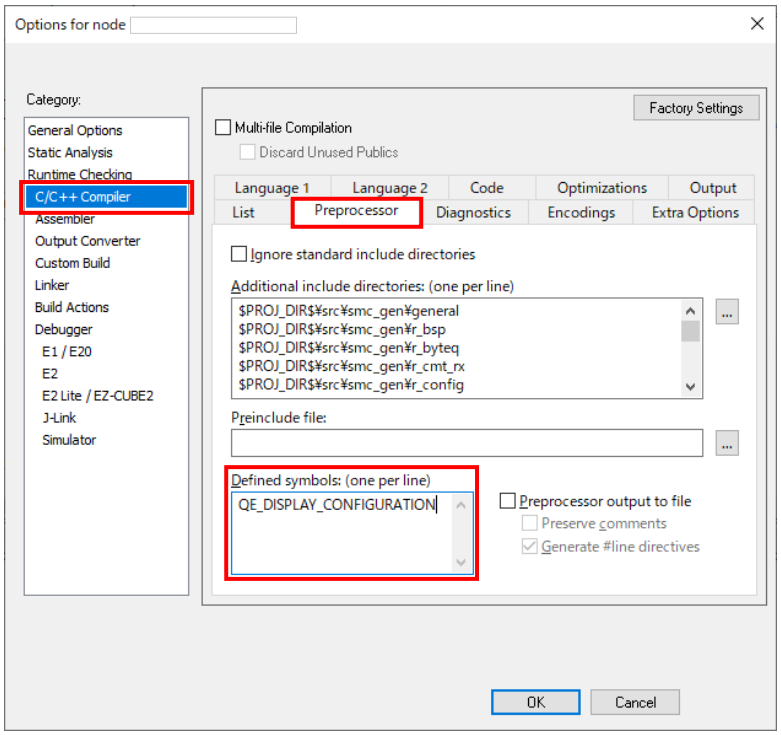
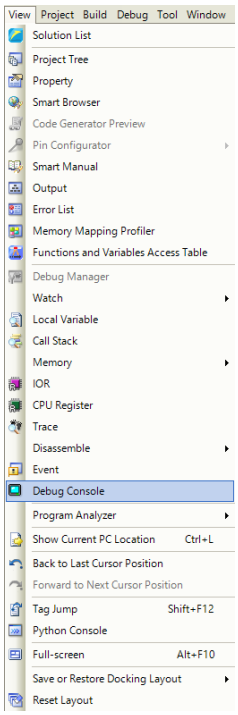


Figure 4.3 Configuration in EWRX

4.2.2 Displaying the debug console

This module sends an error message to the standard output destination when an adjustment or communication error occurs. The following shows how to display the console screen in CS+ and EWRX.



From the menu bar
View
➡ Debug Console
(Valid while debug connected)

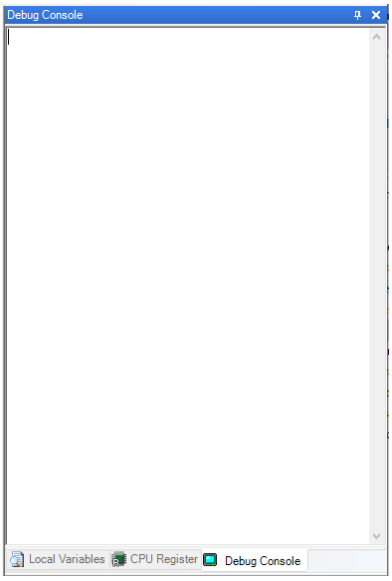
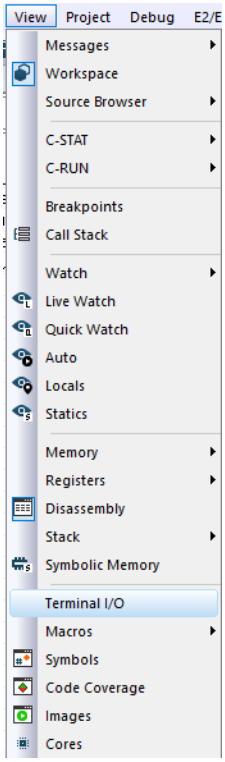


Figure 4.4 Configuration in CS+



From the menu bar
View
➡ Terminal I/O
(Valid while debug connected)

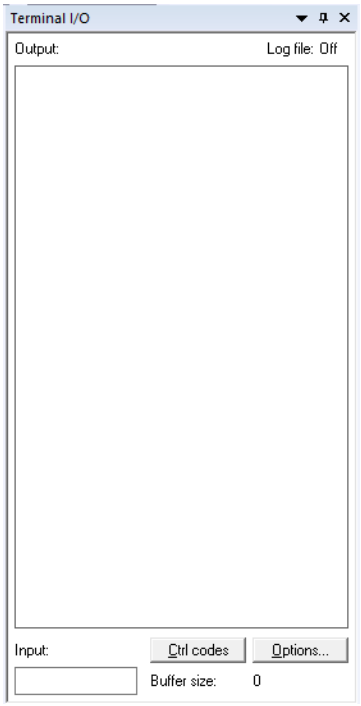


Figure 4.5 Configuration in EWRX

4.3 Confirmed Operation Environment

The following shows the confirmed operation environment for this FIT module.

Table 4.1 Confirmed Operation Environment (Rev.1.00)

Item	Description
Integrated development environment	Renesas Electronics e^2 studio 2024-04 IAR Embedded Workbench for Renesas RX 5.10.1
C compiler	Renesas Electronics C/C++ Compiler for RX Family V3.06.00 Compile options: Default settings of the integrated development environment with the following option added: -lang = c99
	GCC for Renesas RX 8.03.00.202405 Compile options: Default settings of the integrated development environment with the following option added: -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 5.10.01 Compile options: Default settings of the integrated development environment
Endianness	Big endian or little endian
Module revision	Rev.1.00
Board used	Renesas Envision Kit RPBRX65N (product No.: RTK5RX65N2C00000BR) Renesas Envision Kit RPBRX72N (product No.: RTK5RX72N0C00000BJ) Renesas Starter Kit+ for RX72N (product No.: RTK5572NNHS10000BE)

Table 4.2 Confirmed Operation Environment (Rev.1.10)

Item	Description
Integrated development environment	Renesas Electronics e^2 studio 2024-07 IAR Embedded Workbench for Renesas RX 5.10.1
C compiler	Renesas Electronics C/C++ Compiler for RX Family V3.06.00 Compile options: Default settings of the integrated development environment with the following option added: -lang = c99
	GCC for Renesas RX 8.03.00.202405 Compile options: Default settings of the integrated development environment with the following option added: -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 5.10.01 Compile options: Default settings of the integrated development environment
Endianness	Big endian or little endian
Module revision	Rev.1.10
Board used	Renesas Envision Kit RPBRX65N (product No.: RTK5RX65N2C00000BR) Renesas Envision Kit RPBRX72N (product No.: RTK5RX72N0C00000BJ) Renesas Starter Kit+ for RX72N (product No.: RTK5572NNHS10000BE)

Table 4.3 Confirmed Operation Environment (Rev.1.11)

Item	Description
Integrated development environment	Renesas Electronics e^2 studio 2025-01 IAR Embedded Workbench for Renesas RX 5.10.1
C compiler	Renesas Electronics C/C++ Compiler for RX Family V3.07.00 Compile options: Default settings of the integrated development environment with the following option added: -lang = c99
	GCC for Renesas RX 8.03.00.202411 Compile options: Default settings of the integrated development environment with the following option added: -std=gnu99
	IAR C/C++ Compiler for Renesas RX version 5.10.01 Compile options: Default settings of the integrated development environment
Endianness	Big endian or little endian
Module revision	Rev.1.11
Board used	-

4.4 Troubleshooting

(1) Q: I added this FIT module to my project, but executing the build results in one of the following error messages: “Error!! Please declare QE_DISPLAY_CONFIGURATION definition to the compiler.”, “Could not open source file ‘r_image_config.h’”, or “Could not open source file ‘r_lcd_timing.h’”

A: Confirm that a file is output from the standalone version of QE for Display, and that the macro “QE_DISPLAY_CONFIGURATION” is defined in the project.

(2) Q: I cannot start communication with QE for Display successfully, and an error occurs.

A: Check for problems in the SCI FIT module settings, terminals used, wiring, and so on. Also check whether communication is possible at a slower Baud rate. The Baud rate can be specified in this module’s configuration options.

(3) Q: When performing adjustment, none of my adjustments are reflected in the LCD.

A: Adjustments might not appear on the LCD display depending on your individual settings. First, try adjusting image quality, such as the luminance and contrast, and check whether the adjustments are reflected in the LCD. If QE for Display appears unresponsive, or if nothing appears to be reflected in the LCD, disconnect from QE for Display, reset the device, and try performing adjustment again. Additionally, if the configuration option “QE_DISPLAY_CFG_DEBUG_PRINT_ENABLE” is set to “1”, an error message is output to the standard output destination (when debugging in CS+, the debug console). Read the output message.

5. Reference Documents

User's Manual: Hardware
(Obtain the latest version from the Renesas Electronics website.)

Renesas Technical Update and Technical News
(Obtain the latest version from the Renesas Electronics website.)

User's Manual: Development environment
RX Family CC-RX Compiler User's Manual (R20UT3248)
(Obtain the latest version from the Renesas Electronics website.)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Jul. 19. 24	—	First edition issued
1.10	Sep. 20. 24	8 23	2.7 Code Size Modified 4.3 Confirmed Operation Environment Added table 4.2
		program	1.Fixed an issue where a communication error occurred when changing the receive buffer size (SCI_CFG_CHx_RX_BUFSIZ) of the SCI FIT module from the default value of “80”. 2.Fixed an issue where "Bit-endian of Output Data" and "Pixel Order of Output Data" were not set correctly in the TCON/LCD Setting tab of QE for Display.
1.11	Mar. 20. 25	24	4.3 Confirmed Operation Environment Added table 4.3
		program	Changed the disclaimer in program sources

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/.