R01AN2293EJ0120
Rev.1.20
Jun 1, 2020

RX Family

Sample Program using USB Host Communication Device Class Driver(HCDC) for USB Mini Firmware to communicate via USB with CDC device Firmware Integration Technology

Introduction

This document describes the following sample firmware: USB Host Communication Devices Class Driver using Firmware Integration Technology. The sample firmware is referred to below as the HCDC.

When developing an actual software, be sure to use the “USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology Application Note” (Document number: R01AN2166) together with the user’s manual for each MCU (Hardware). In addition, also refer to the " USB Host Communication Device Class Driver (HCDC) for USB Mini Firmware using Firmware Integration Technology Application Note” (Document number:R01AN2167), if necessary. “USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology Application Note” (Document number: R01AN2166) is located in the "reference_documents" folder within the package.

Target Device

RX111 Group
RX113 Group
RX231 Group
RX23W Group

The operation of this program has been confirmed using the Renesas Starter Kits (RSK) or the Renesas Solution Starter Kit (RSSK).

Contents

1. Introduction

1.1 Functions

The HCDC conforms to the Abstract Control Model of the USB communication device class (CDC) specification and implements communication with CDC devices.

The HCDC provides the following functionality:

- Performs communication class data transfer when a CDC device is connected.

1.2 FIT Module Configuration

The HCDC comprises the following FIT modules and a sample application:

Table 1-1 FIT Module Configuration

FIT Module	Folder Name
RX Family Board Support Package Module Using Firmware Integration Technology	r_bsp
RX Family USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology	r_usb_basic_mini
RX Family USB Host Communication Devices Class Driver (CDC) for USB Mini Firmware using Firmware Integration Technology	r_usb_hcdc_mini
RX Family DTC Module Using Firmware Integration Technology	r_dtc_rx
RX Family DMA Controller DMACA Control Module Firmware Integration Technology	r_dmaca_rx

Refer to the related documentation for details of each FIT module. Note that the latest versions of the FIT modules used by the sample firmware are available for download from the following website:

Renesas Electronics website: http://www.renesas.com/

1.3 Note

This driver is not guaranteed to provide USB communication operation. The customer should verify operation when utilizing it in a system and confirm the ability to connect to a variety of different types of devices.

1.4 Operating Confirmation Environment

The operating confirmation environment for the HCDC is described below:

Table 1-2 Operation Confirmation Environment

Item	Contents
C compiler	Renesas Electronics C/C++ compiler for RX Family V.3.02.00 (The option "-lang=C99" is added to the default setting of IDE)
	GCC for Renesas RX 8.3.0.201904 (The option "-std=gnu99" is added to the default setting of IDE)
	IAR C/C++ Compiler for Renesas RX version 4.14.1
Real-Time OS	FreeRTOS V.10.0.0 RI600V4 V.1.06
Endian	Little Endian, Big Endian
USB Driver Revision Number	Rev.1.20
Using Board	Renesas Starter Kit for RX111 Renesas Starter Kit for RX113 Renesas Starter Kit for RX231 Renesas Solution Starter Kit for RX23W

2. Software Configuration

2.1 Module Configuration

The HCDC transfers data to and from a device via the HCD and reports the results to the APL. Also, it reports data transfer requests from the APL to devices via HCDC and HCD.
Figure 2-1 shows the module configuration of the HCDC, and Table 2-1 lists the functions of the modules.

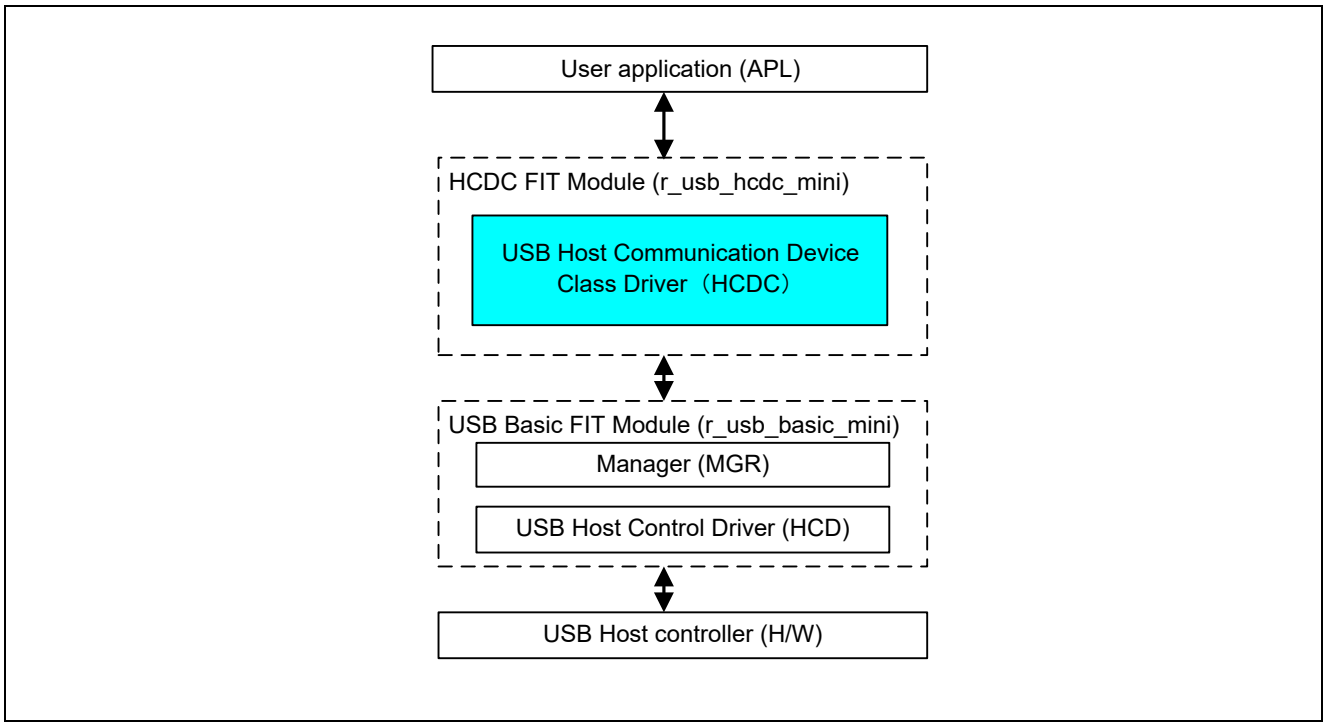


Figure 2-1 Module Configuration

Table 2-1 Functions of Modules

Module Name	Function
APL	Sample application program
HCDC (r_usb_hcdc_mini)	CDC Class Driver ・ Sends CDC-related requests from the APL and data transfer requests to the HCD.
HCD (r_usb_basic_mini)	USB Host Control Driver

3. Setup

3.1 Hardware

3.1.1 Example Operating Environment

Figure 3-1 shows an example operating environment for the HCDC. Refer to the associated instruction manuals for details on setting up the evaluation board and using the emulator, etc.

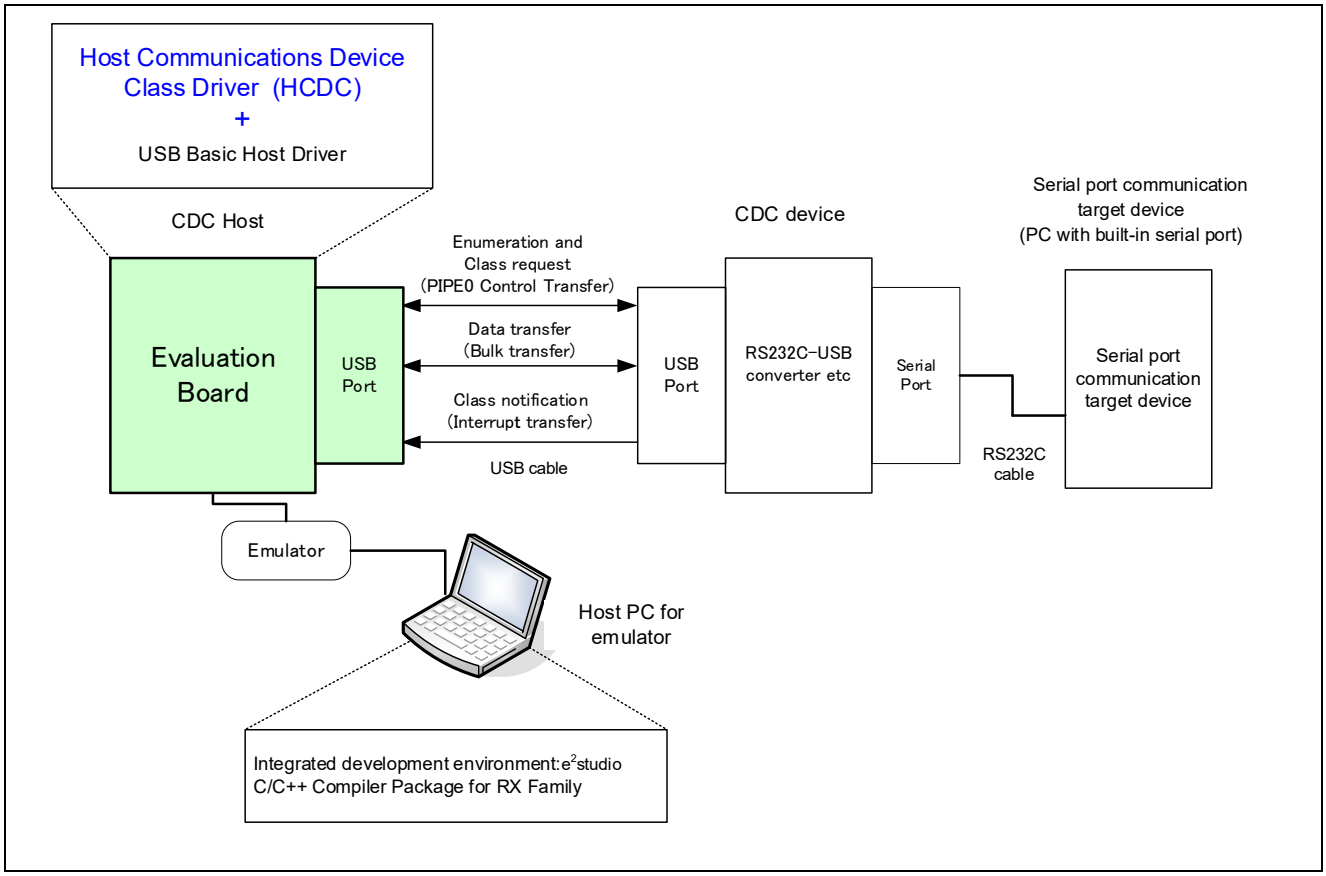


Figure 3-1 Example Operating Environment

Table 3-1 shows the evaluation board on which operation has been confirmed.

Table 3-1 Evaluation Board on which HCDC Operation Has Been Verified

MCU	Evalulation Board
RX111	RSKRX111
RX113	RSKRX113
RX231	RSKRX231
RX23W	RSSKRX23W

3.1.2 RSK / RSSK Setting

It is necessary to set RSK/RSSK to operate in the host mode. Please refer to the following.

Table 3-2 **RSK / RSSK Setting**

RSK / RSSK	Jumper Setting
RSKRX111	J12: Shorted Pin1-2
RSKRX113	J12: Shorted Pin1-2
RSKRX231	J15: Shorted Pin1-2
RSSKRX23W	J5: Shorted Pin2-3

Note:

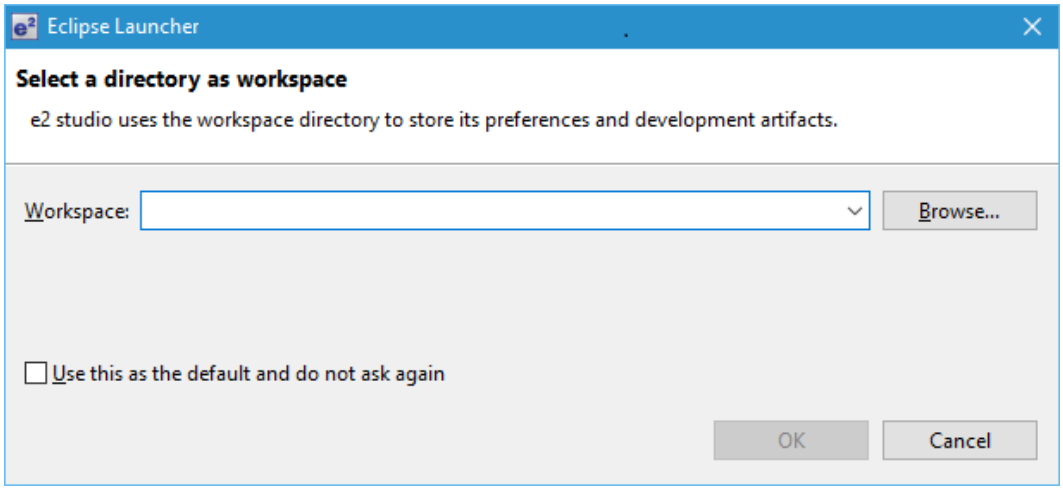
For the detail of RSK/RSSK setting, refer to the user's manual of RSK/RSSK.

3.2 Software

1) Setup e^2 studio

a) Start e^2 studio

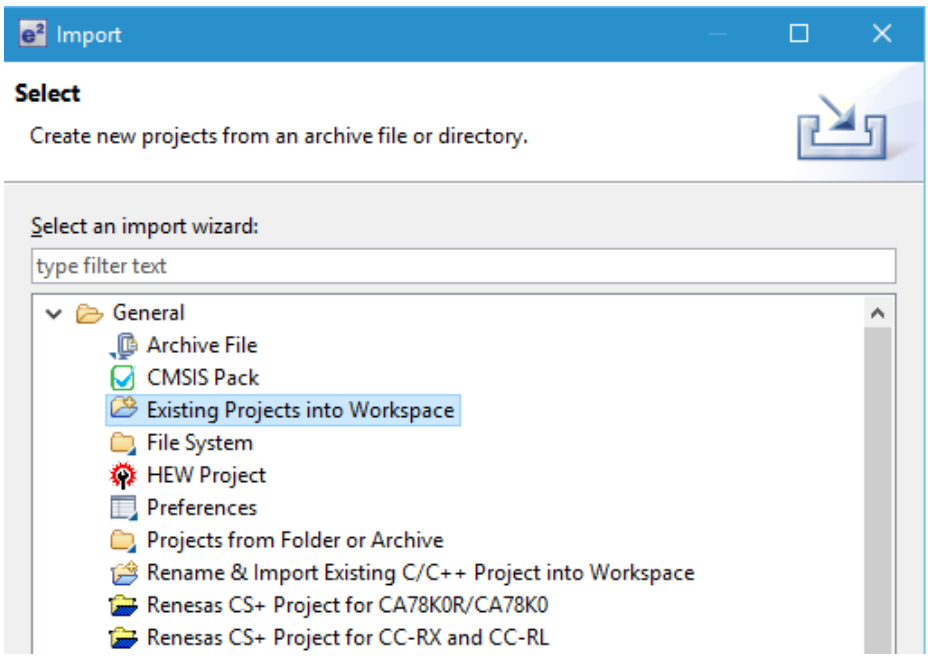
b) If you start up e^2 studio at first, the following dialog is displayed. Specify the folder to store the project in this dialog.



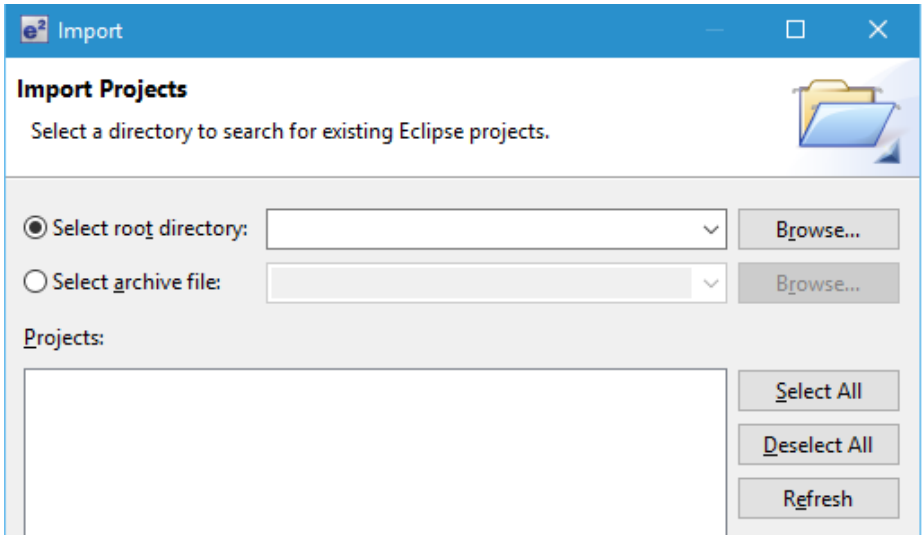
2) Import the project to the workspace

a) Select [File] > [Import]

b) Select [General] => [Existing Projects into Workspace]



c) Select the root directory of the project, that is, the folder containing the ".cproject" file.



d) Click "Finish".

You have now imported the project into the workspace. Note that you can import other projects into the same workspace.

3) Generate the binary target program by clicking the "Build" button.

4) Connect the target board to the debug tool and download the executable. The target is run by clicking the "Run" button.

4. Sample Application

4.1 Application Specifications

The main functions of the APL are as follows:

1. Sends receive (Bulk In transfer) requests to the CDC device and receives data.
2. Transfers received data to the CDC device by means of Bulk Out transfers (loopback).
3. The communication speed and other settings are made by transmitting the class request *SET_LINE_CODING* to the CDC device. This class request can be used to set the communication speed, number of data bits, number of stop bits, and the parity bit.

4.1.1 Data Transfer Image

Figure 4-1 shows the data transfer image.

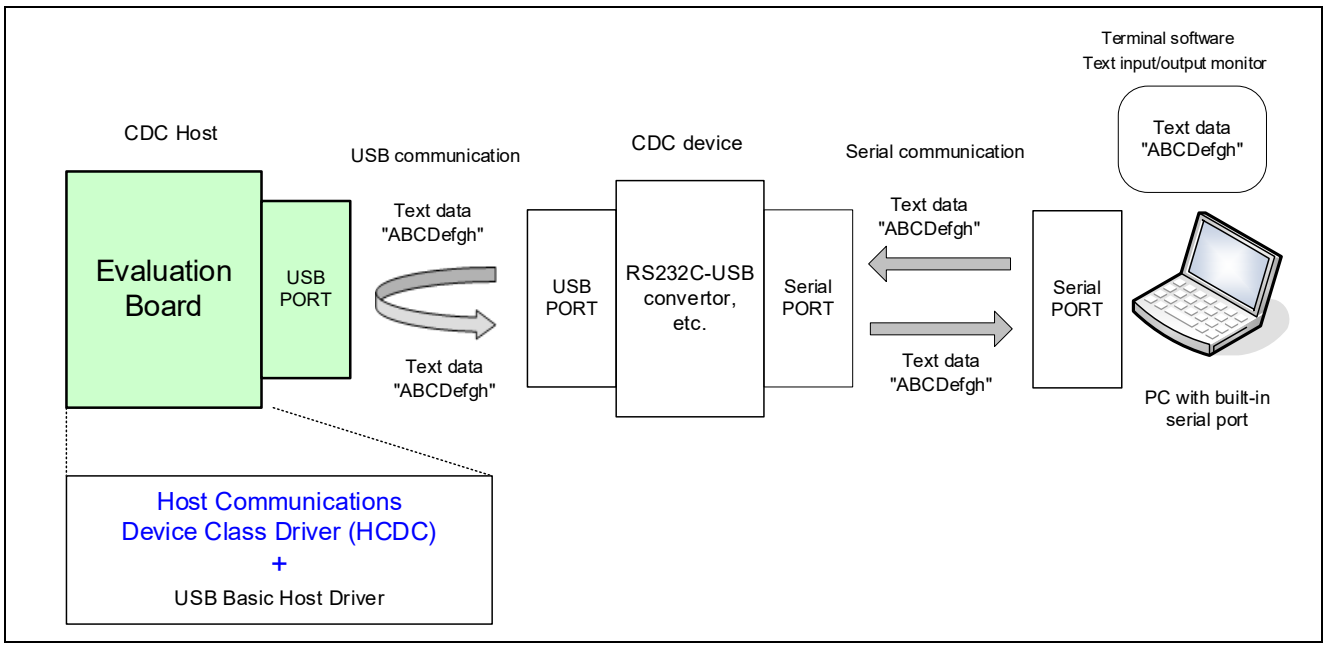


Figure 4-1 Data Transfer (Loopback) Image

4.2 Application Processing

The application comprises two parts: initial settings and main loop. An overview of the processing in these two parts is provided below.

4.2.1 Initial setting

Initial settings consist of MCU pin settings, USB driver settings, and initial settings to the USB controller.

4.2.2 Main Loop (Non-OS)

The main loop performs loop-back processing in which data received from the CDC device is transmitted unaltered back to the CDC device as part of the main routine. An overview of the processing of the main loop is presented below.

1. When the *R_USB_GetEvent* function is called after the CDC device attaches to the RSK/RSSK and enumeration completes, *USB_STS_CONFIGURED* is set as the return value. When the APL confirms *USB_STS_CONFIGURED*, it sends class request *SET_LINECODING* to the CDC device.
2. When it confirms that the class request processing has completed, the APL calls the *R_USB_Read* function to make a data receive request for data sent from the CDC device. Note that in addition to the data receive request a receive request is also sent for a class notification from the CDC device.
3. When the *R_USB_GetEvent* function is called after reception of data from the CDC device has completed, *USB_STS_READ_COMPLETE* is set as the return value. The received data is stored in external variable *g_data*. The receive data size can be confirmed by means of the size member of the usb_ctrl_t structure. The APL determines that a null packet has been received if the value of the size member is 0 (zero) and performs another data receive request. If the value of the size member is other than 0 (zero), the APL determines that data has been received from the CDC device. It then makes a transmit request to send the received data to the CDC device.
4. When the *R_USB_GetEvent* function is called after transmission of data to the CDC device completes, *USB_STS_WRITE_COMPLETE* is set as the return value. When the APL confirms *USB_STS_WRITE_COMPLETE*, it calls the R_USB_Read function to make a data receive request for data sent by the CDC device.
5. The processing in steps 3 and 4, above, is repeated.

An overview of the processing performed by the APL is shown below:

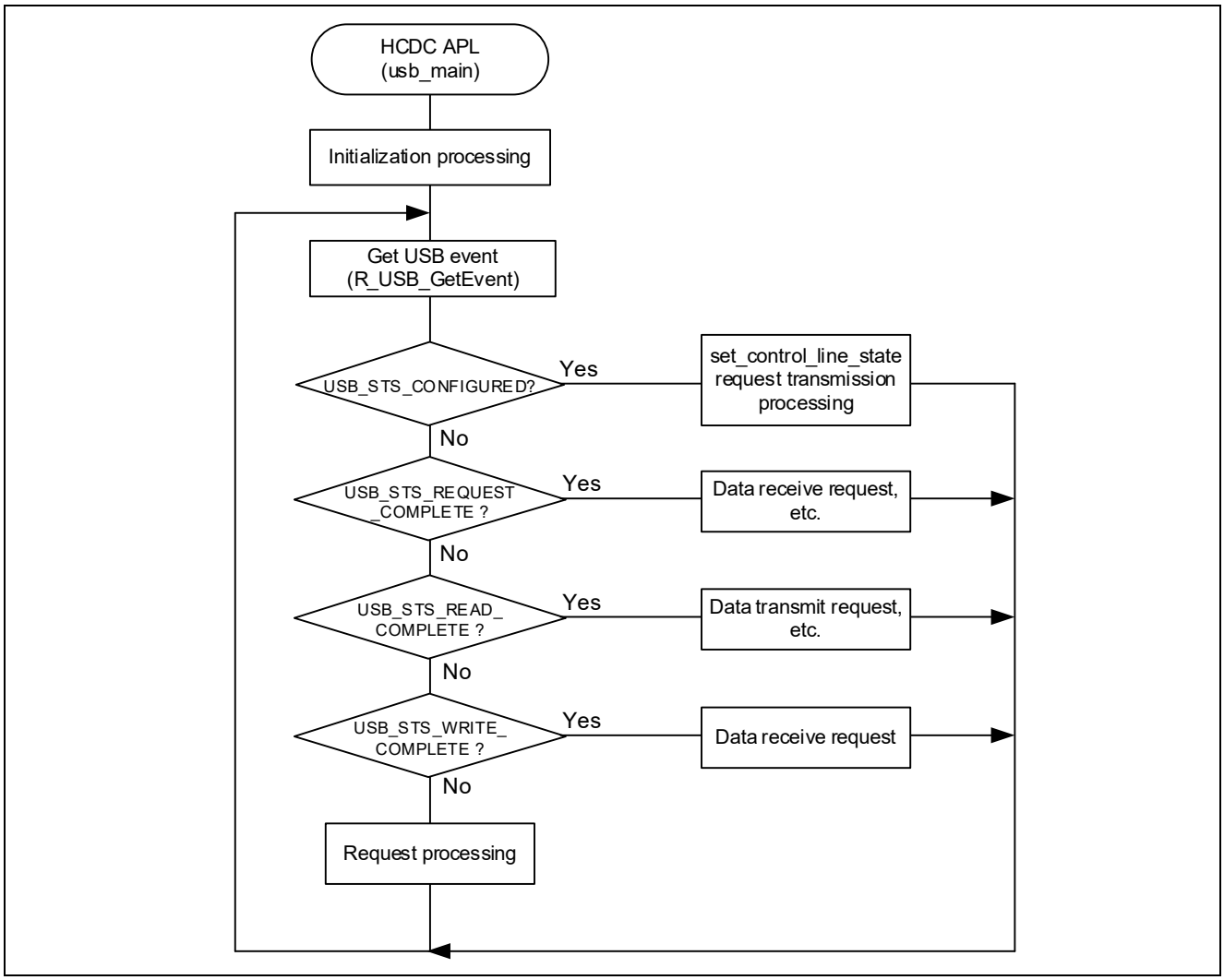


Figure 4-2 Main Loop processing (for Non-OS)

4.2.3 Main Loop (for RTOS)

The loop performs loop-back processing in which data received from the CDC device is transmitted unaltered back to the CDC device as part of the main routine. An overview of the processing performed by the loop is shown below.

1. When a USB-related event has completed, the USB driver calls the callback function (*usb_apl_callback*). In the callback function (*usb_apl_callback*), the application task (APL) is notified of the USB completion event using the real-time OS functionality.
2. In APL, information regarding the USB completion event was notified from the callback function is retrieved using the real-time OS functionality.
3. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_CONFIGURED*, APL sends the class request (*SET_LINECODING*) to the CDC device.
4. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_REQUEST_COMPLETE*, APL performs a data reception request to receive data transmitted from the CDC device by calling the *R_USB_Read* function and also performs a class notification reception request from CDC device.
5. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_READ_COMPLETE*, APL performs a data transmission request to send the reception data by calling the *R_USB_Write* function. The reception data is stored in the gloval variable (*g_data*). The reception data size

is set in the member (*size*) of the *usb_ctrl_t* structure. If this member (*size*) is zero, the USB driver judges that the *NULL* packet is received and performs a data reception request to the CDC device again.

6. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_WRITE_COMPLETE*, APL performs a data reception request to receive the data sent from CDC device.
7. The avove processing is repeated.

An overview of the processing performed by the APL is shown below:

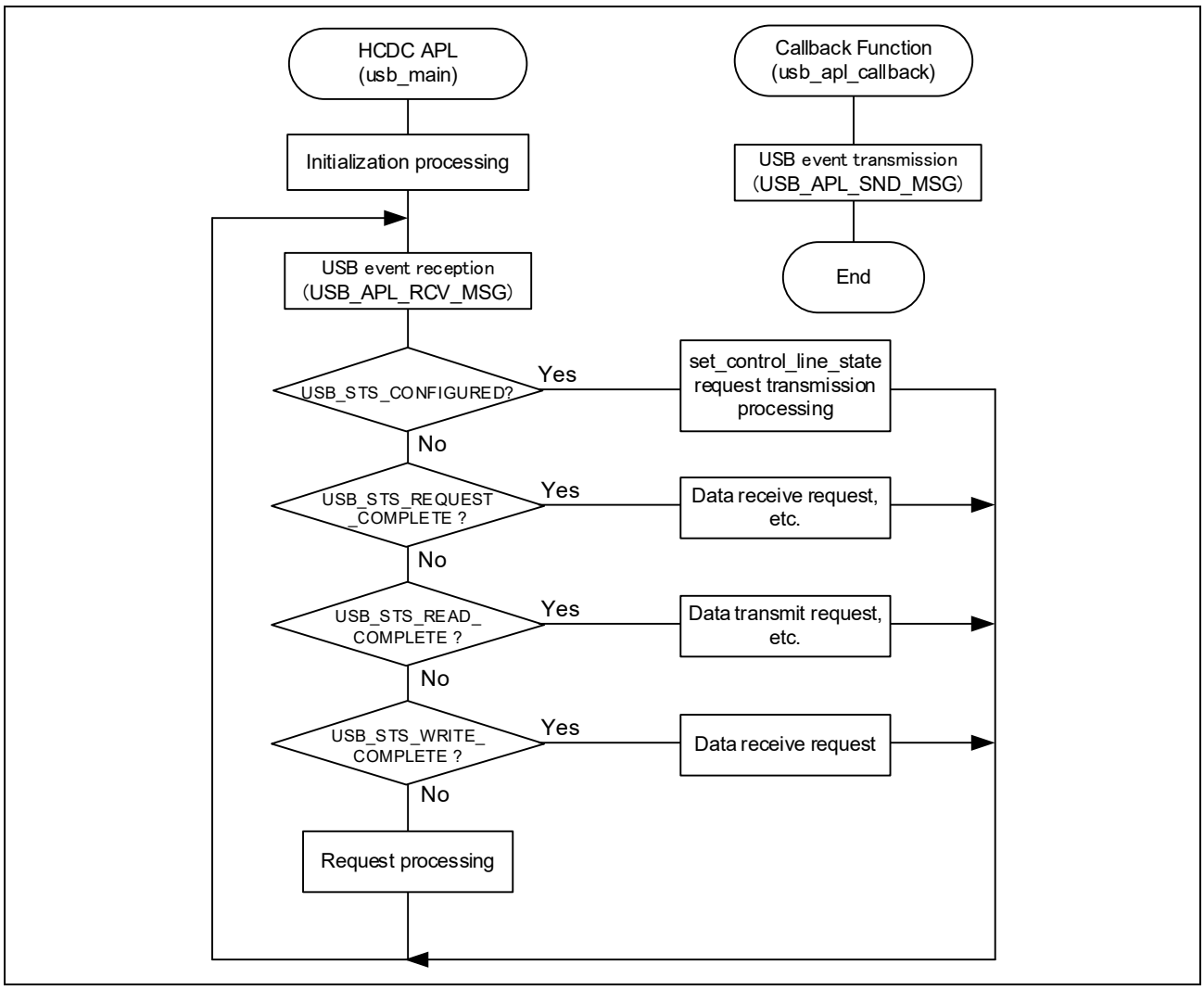


Figure 4-3 Main Loop processing (for RTOS)

4.3 Configuration File for the application program (r_usb_hcdc_apl_config.h)

Make settings for the definitions listed below.

1. COM_SPEED Definition

Please specify the baud rate value. This baud rate value is set to the CDC device by the class request (*SET_LINE_CODING*). Specify a setting of *BPS_9600 / BPS_14400 / BPS_19200 / BPS_38400 / BPS_57600 / BPS_115200* to *COM_SPEED* defintion.

```
#define     COM_SPEED           BPS_57600               // Baud rate value
```

2. COM_PARITY_BIT Definition

Please specify the parity bit. This parity bit is set to the CDC device by the class request (*SET_LINE_CODING*). Specify a setting of *PARITY_EVEN / PARITY_ODD / PARITY_NONE* to *COM_PARITY_BIT* definition.

```
#define     COM_PARITY_BIT      PARITY_NONE             // Parity bit
```

3. COM_STOP_BIT Definition

Please specify the stop bit. This stop bit is set to the CDC device by the class request (*SET_LINE_CODING*). Specify a setting of *STOP_BIT1*(1 bit) / *STOP_BIT15*(1.5 bit) / *STOP_BIT2*(2 bit) to *COM_STOP_BIT* definition.

```
#define     COM_STOP_BIT        STOP_BIT1               // Stop bit
```

4. COM_DATA_BIT Definition

Please specify the data bit. This data bit is set to the CDC device by the class request (*SET_LINE_CODING*). Specify a setting of *DATA_BIT7*(7 bit) / *DATA_BIT8*(8 bit) to *COM_DATA_BIT* definition.

```
#define     COM_DATA_BIT        DATA_BIT8               // Data bit
```

5. USB_SUPPORT_RTOS Definition

Please specify *USB_APL_ENABLE* to *USB_SUPPORT_RTOS* definition when using the real-time OS.

```
#define     USB_SUPPORT_RTOS    USB_APL_DISABLE     // No use the real-time OS
#define     USB_SUPPORT_RTOS    USB_APL_ENABLE      // Use the real-time OS
```

6. Note

The above configuration settings apply to the application program. USB driver configuration settings are required in addition to the above settings. For information on USB driver configuration settings, refer to the application note *USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology* (Document number. R01AN2166).

5. Class Driver Overview

5.1 Class Request

Table 5-1 shows the class requests supported by HCDC.

Table 5-1 Supported Basic Requests and CDC Class Requests

Request	Code	Description	Suppoted
SendEncapsulatedCommand	0x00	Sends protocol-defined commands such as AT.	YES
GetEncapsulatedResponse	0x01	Requests response to command sent in SendEncapsulatedCommand.	YES
SetCommFeature	0x02	Sets enable/disable for device-specific 2-byte code and country code.	YES
GetCommFeature	0x03	Sets enable/disable for device-specific 2-byte code and country code.	YES
ClearCommFeature	0x04	Returns enable/disable status for device-specific 2-byte code and country code to default settings.	YES
SetLineCoding	0x20	Sets transmission line coding (transmission speed, data length, parity bit, stop bit length)	YES
GetLineCoding	0x21	Obtains transmission line coding status.	YES
SetControlLineState	0x22	Sets control signals RTS and DTR for transmission line.	YES
SendBreak	0x23	Sends break signal.	YES

5.2 Class Notification（Notification from CDC device to USB Host）

Table 5-2 shows class notifications supported by the HCDC.

Table 5-2 CDC Class Specific Notification

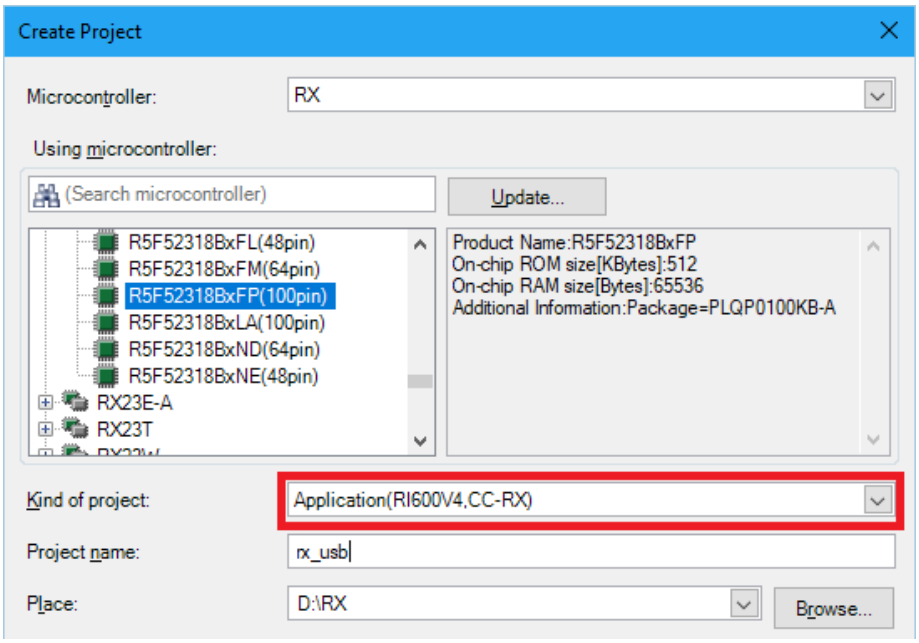
Notification	Code	Description	Supported
NETWORK_CONNECTION	0x00	Notifies network connection state.	NO
RESPONSE_AVAILABLE	0x01	Response to GET_ENCAPSLATED_RESPONSE.	NO
SERIAL_STATE	0x20	Notifies serial line status.	YES

6. Using RI600V4 project with CS+

The RI600V4 project in the package does not support CS+. The user needs to create a project for CS+ according to the following procedure when using RI600V4 project on CS+.

6.1 New Project Creation

Select "Application(RI600V4, CC-RX) for the Kind of project.

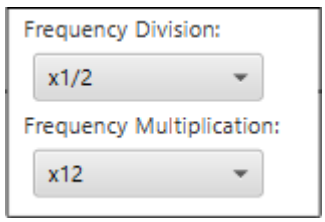


6.2 Launch Smart Configurator

1. Clock Setting (Select "Clocks" tab)

Set the related clock so that "48MHz" is set to UCLK (USB clock).

The following is a setting example when using the oscillator(8MHz).



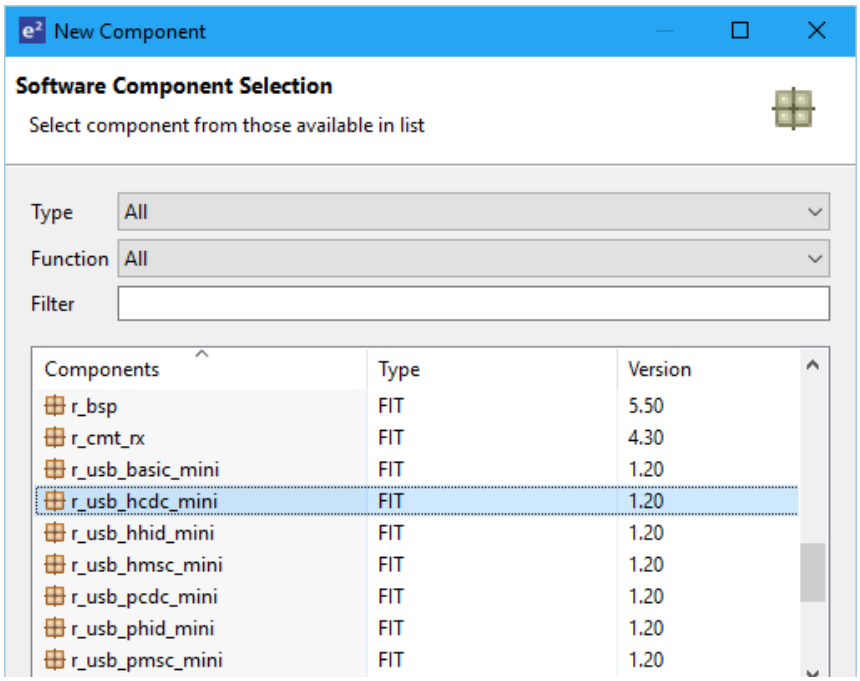
2. Component Setting (Select "Components" tab)

(1). Import the USB FIT module

Select the *r_usb_hcdc_mini* module and press the "Finish" button. The *r_usb_basic_mini* module is imported at the same time.

Note:

Select the *r_dtc_rx* / *r_dmaca_rx* module when using the DTC/DMA.

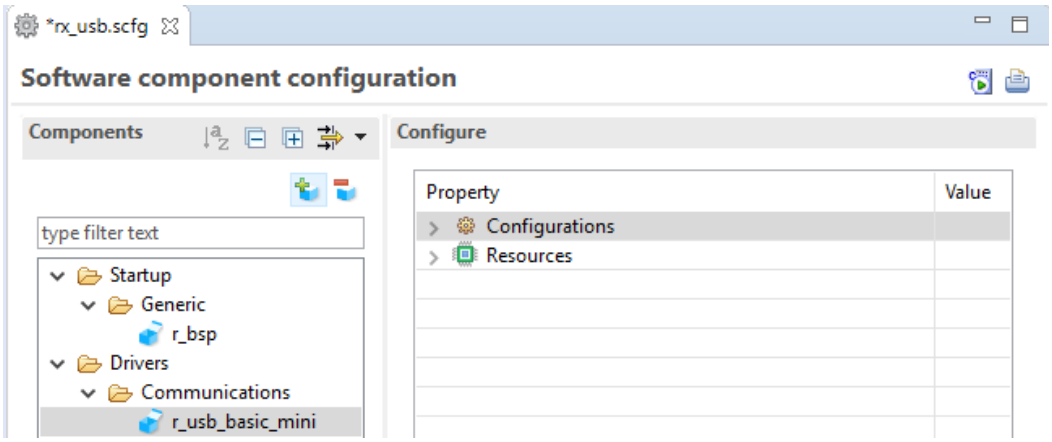


(2). Configuration Setting

a. r_bsp

Change the heap size when using DTC transfer. For the setting value, refer to the documentation for DTC FIT module.

b. r_usb_basic_mini



(a). Configurations

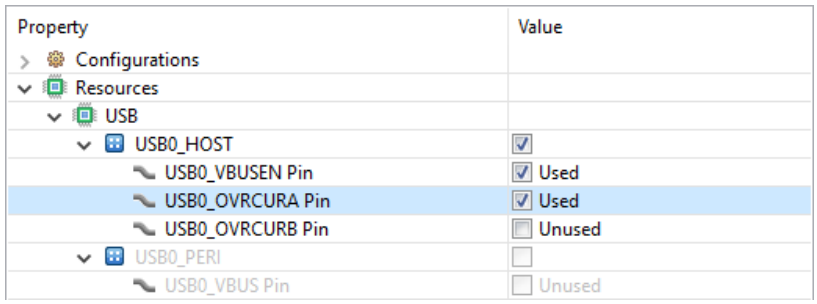
Set each item according to the user system.
For the detail of each item, refer to chapter "Configuration" in *USB Basic Mini Host and Peripheral Driver Firmware Integration Technology* application note (Document number: R01AN2166).

(b). Resources

Check the following check box.

i. *USBx_VBUSEN* pin

ii. *USBx_OVRCURA* pin or *USBx_OVRCURB* pin

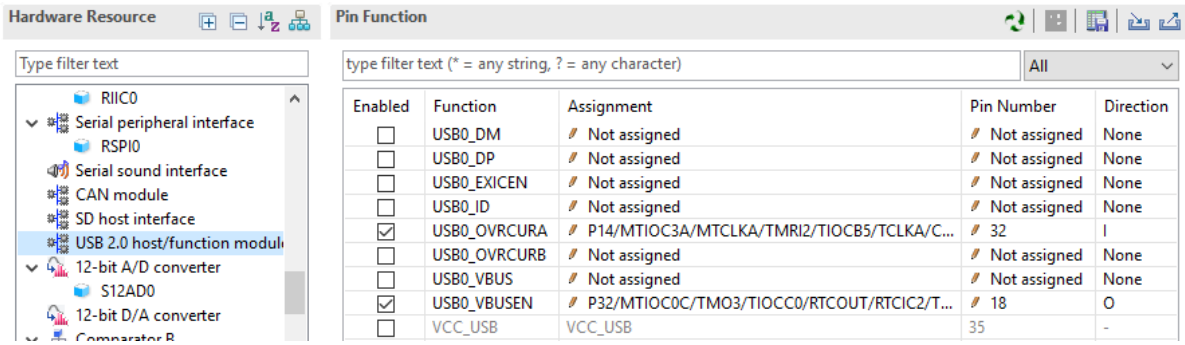


c. r_usb_hcdc_mini

Refer to chapter "Configuration" in *USB Host Communications Devices Class Driver (HCDC) for USB Mini Firmware Firmware Integration Technology* application note (Document number: R01AN2167).

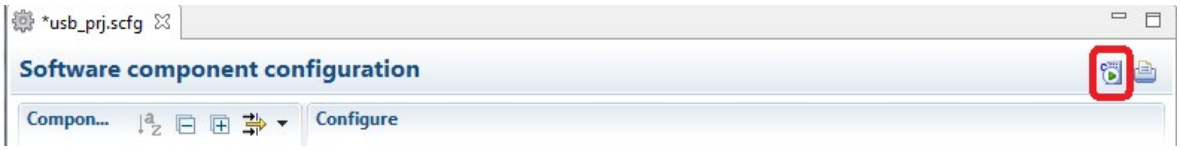
3. Pin Setting (Select "Pins" tab)

Select the port for USB pin match the user system.



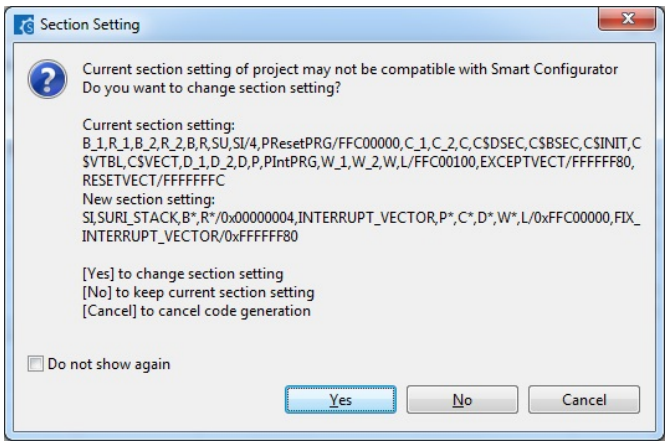
4. Generate Code

The Smart Configurator genrates source codes for USB FIT module and USB pin setting in "*<ProjectDir>¥src¥smc_gen*" folder by by clicking on the [(Generate Code)] button.



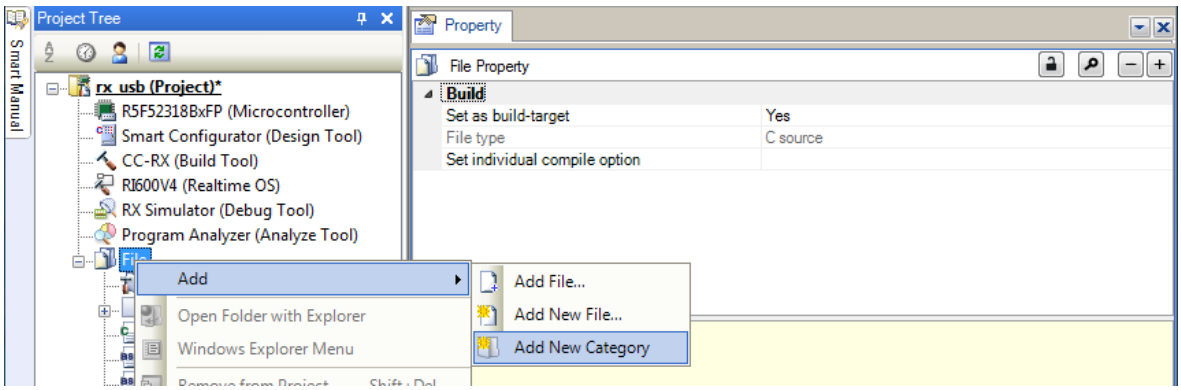
Note:

Select "Yes" if the following dialog box is displayed.



6.3 Add the application program and the configuration file

1. Copy the *demo_src* folder in this package to the *"<ProjectDir>¥src"* folder.
2. Copy the RI600V4 configuration file (.cfg file) to *"<ProjectDir>"* folder.
3. Select "File" in the "Project Tree" and click the right button. Select [Add] → [Add New Category] and create the category to store the application program. Then select [Add File] and register the application program and the configuration file which are copied at the above 2.



Note:

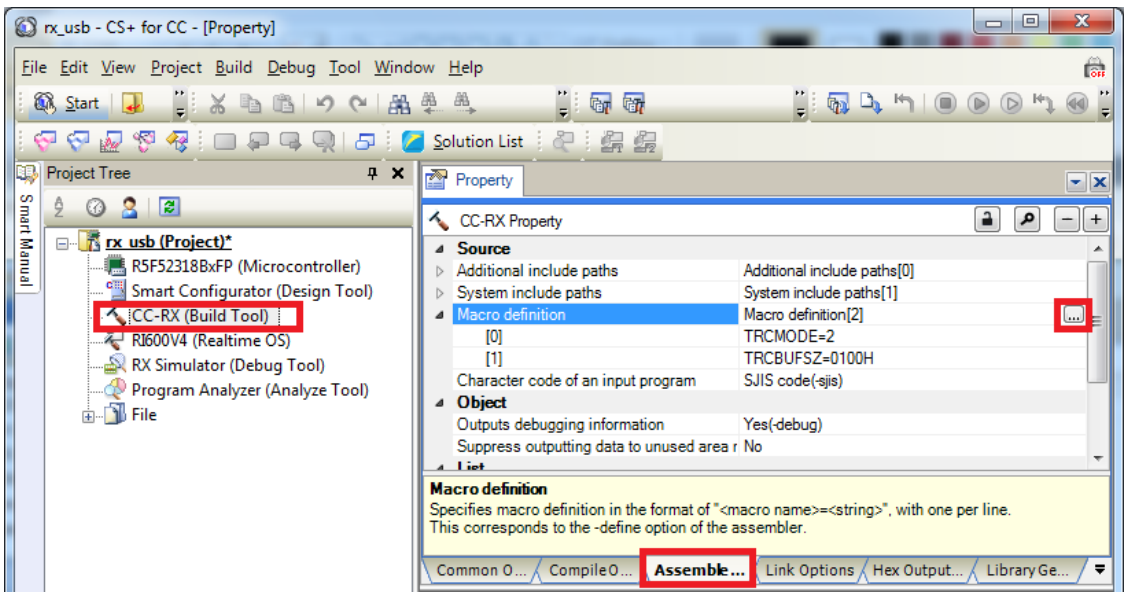
Remove the "task.c" file and "sample.cfg" created in "<*ProjectDir*>" folder by CS+.

6.4 Remote Macro Definition

Remove these macros since the following macros is defined in the new created project.

Select [CC-RX(Build Tool)] → [Assemble Options] tab, remove the following macros.

1. TRCMODE = 2
2. TRCBUFSZ = 0100H



6.5 Build Execution

Excecute the build and generate the binary target program.

7. Using the e² studio project with CS+

The HCDC contains a project only for e² studio. When you use the HCDC with CS+, import the project to CS+ by following procedures.

[Note]

1. Uncheck the checkbox Backup the project composition files after conversion in Project Convert Settings window.
2. The following method is not supported when using RI600V4. Refer to chapter **6, Using RI600V4 project with CS+**.

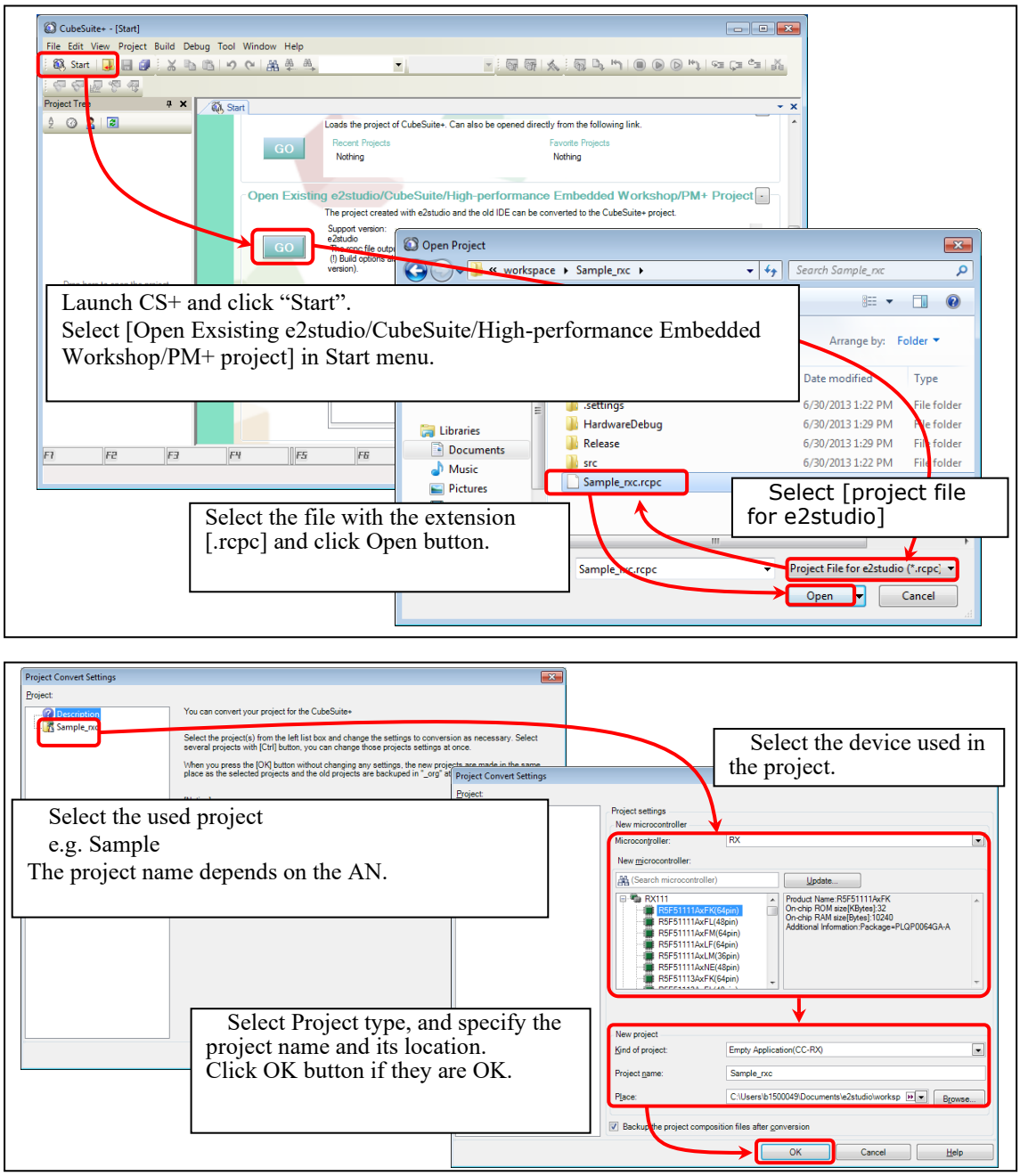


Figure 7-1 Using the e² studio project with CS+

Website and Support

Renesas Electronics Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry/

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec 1, 2014	—	First Edition Issued.
1.01	Jun 1, 2015	—	RX231 is added in Target Device.
1.02	Dec 28, 2015	—	Upgrading of this USB driver by upgrading of "USB Basic Mini Firmware (R01AN2166)".
1.10	Nov 30, 2018	—	1. The following chapter has beed added. (1). 3.1.2 RSK/RSSK Setting 2. The following chapter has beed changed. (1). 4. Sample Application
1.12	Jun 30, 2019	—	RX23W is added in Target Device.
1.20	Jun 1, 2020	—	Supported the real-time OS.

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. 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

   The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Clock Signals

   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

5. Differences between Products

   Before changing from one product to another, i.e. to a product with a different part number, confirm that the change will not lead to problems.

   — The characteristics of Microprocessing unit or Microcontroller unit products in the same group but having a different part number may differ in terms of the 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 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

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

Renesas Electronics America Inc. Milpitas Campus
1001 Murphy Ranch Road, Milpitas, CA 95035, U.S.A.
Tel: +1-408-432-8888, Fax: +1-408-434-5351

Renesas Electronics America Inc. San Jose Campus
6024 Silver Creek Valley Road, San Jose, CA 95138, USA
Tel: +1-408-284-8200, Fax: +1-408-284-2775

Renesas Electronics Canada Limited
9251 Yonge Street, Suite 8309 Richmond Hill, Ontario Canada L4C 9T3
Tel: +1-905-237-2004

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 101-T01, Floor 1, Building 7, Yard No. 7, 8th Street, Shangdi, Haidian District, Beijing 100085, China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai 200333, China
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1611, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, #06-02 Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit No 3A-1 Level 3A Tower 8 UOA Business Park, No 1 Jalan Pengaturcara U1/51A, Seksyen U1, 40150 Shah Alam, Selangor, Malaysia
Tel: +60-3-5022-1288, Fax: +60-3-5022-1290

Renesas Electronics India Pvt. Ltd.
No.777C, 100 Feet Road, HAL 2nd Stage, Indiranagar, Bangalore 560 038, India
Tel: +91-80-67208700

Renesas Electronics Korea Co., Ltd.
17F, KAMCO Yangjae Tower, 262, Gangnam-daero, Gangnam-gu, Seoul, 06265 Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5338

Colophon 9.0