RX Family

R01AN0275EJ0233
Rev.2.33
Jul 31, 2019

USB Host Communications Device Class Driver (HCDC)

Introduction

This application note describes USB Host Communication Device Class Driver (HCDC). This module operates in combination with the USB Basic Host Driver (USB-BASIC-F/W). It is referred to below as the HCDC.

Target Device

RX62N/RX621 Group

RX63N/RX631 Group

RX63T Group

When using this application note with other Renesas MCUs, careful evaluation is recommended after making modifications to comply with the alternate MCU.

Related Documents

1. USB Basic Host and Peripheral Driver Application Note (Document number.R01AN0512)
2. Universal Serial Bus Revision 2.0 specification
   http://www.usb.org/developers/docs/
3. USB Class Definitions for Communications Devices Revision 1.2
4. USB Communications Class Subclass Specification for PSTN Devices Revision 1.2
   http://www.usb.org/developers/docs/
5. RX62N/RX621 Group User's Manual: Hardware (Document number .R01UH0033)
6. RX63N/RX631 Group User's Manual: Hardware (Document number .R01UH0041)
7. RX63T Group User's Manual: Hardware (Document number .R01UH0238)

- Renesas Electronics Website
  http://www.renesas.com/
- USB Devices Page
  http://www.renesas.com/prod/usb/

Contents

1. Overview

The HCDC, when used in combination with the USB-BASIC-F/W, operates as a USB host communications device class driver (HCDC). The HCDC conforms to the PSTN device subclass abstract control model of the USB communication device class specification (CDC) and enables communication with a CDC peripheral device.

This module supports the following functions.

- Checking of connected devices
- Implementation of communication line settings
- Acquisition of the communication line state
- Data transfer to and from a CDC peripheral device
- HCDC can connect maximum 2 CDC devices to 1 USB module by using USB Hub.

1.1 Please be sure to read

Please refer to the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note* when creating an application program using this driver.

This document is located in the "**reference_documents**" folder within this package.

1.2 Note

1. 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.
2. Please be sure to use the documentations (*r01an0275ej0232_usb.pdf, r01an0512ej0232_usb.pdf*) under the "reference_documents" folder when using RX62N/RX621/ RX63T.

1.3 Limitations

HCDC is subject to the following limitations.

1. Only one stage of the USB hub can be used.
2. Suspend and resume are not supported for CDC devices connected to the USB hub and USB hub downstream ports.
3. Suspend is not supported when data transfer is in progress. Confirm that data transfer has completed before executing suspend.
4. Use of compound USB devices with CDC class support is not supported.
5. This driver does not support DMA/DTC transfer.

1.4 Terms and Abbreviations

APL	:	Application program
CDC	:	Communications devices class
CDCC	:	Communications Devices Class － Communications Class Interface
CDCD	:	Communications Devices Class － Data Class Interface
HCD	:	Host control driver of
HCDC	:	Host communication devices class
HDCD	:	Host device class driver (device driver and USB class driver)
HUBCD	:	Hub class sample driver
MGR	:	Peripheral device state manager of HCD
non-OS	:	USB basic firmware for OS less system
RSK	:	Renesas Starter Kits
USB	:	Universal Serial Bus
USB-BASIC-FW	:	USB Basic Host Driver

2. Software Configuration

Table 2-1 lists the modules, and Figure 2.1 shows a block diagram of HCDC.

Table 2-1 Modules

Module	Description
APL	User application program. Created by customer.
HCDC	Requests CDC requests command and the data transfer from APL to HCD .
MGR / HUB	Enumerates the connected devices and starts HCDC. Also performs device state management.
HCD	USB host H/W control driver. (See USB Basic FW.)

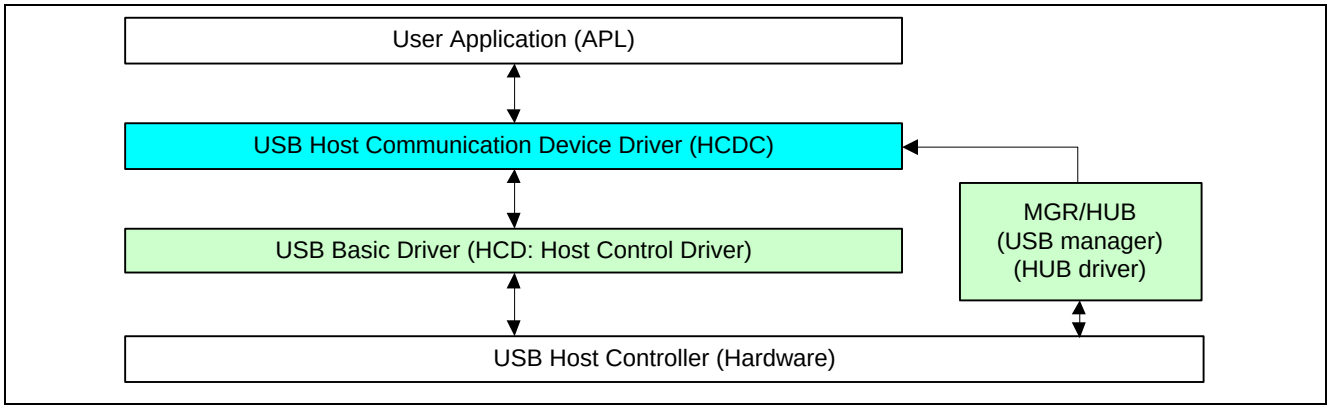


Figure 2.1 Software Block Diagram

3. API Information

This Driver API follows the Renesas API naming standards.

3.1 Hardware Requirements

This driver requires your MCU support the following features:

- USB

3.2 Operating Confirmation Environment

Table 3-1 shows the operating confirmation environment of this driver.

Table 3-1 Operation Confirmation Environment

Item	Contents
C compiler	Renesas Electronics C/C++ compiler for RX Family V.3.01.00
	Compile Option：-lang = c99
Endian	Little Endian, Big Endian
Using Board	Renesas Starter Kits for RX63N

3.3 Usage of Interrupt Vector

Table 3-2 shows the interrupt vector which this driver uses.

Table 3-2 List of Usage Interrupt Vectors

Device	Contents
RX63N/RX631	USBI0 Interrupt (Vector number: 35) / USBR0 Interrupt (Vector number: 90)

3.4 Header Files

All API calls and their supporting interface definitions are located in r_usb_basic_if.h and r_usb_hcdc_if.h.

3.5 Integer Types

This project uses ANSI C99 “Exact width integer types” in order to make the code clearer and more portable. These types are defined in *stdint.h*.

3.6 Compile Setting

For compile settings, refer to chapter **8, Configuration (r_usb_hcdc_config.h)** in this document and chapter "Configuration" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

3.7 ROM / RAM Size

The follows show ROM/RAM size of this driver.

	Checks arguments	Does not check arguments
ROM size	33.3K bytes (Note 3)	32.7K bytes (Note 4)
RAM size	16.6K bytes	16.6K bytes

Note:

1. ROM/RAM size for USB Basic Driver is included in the above size.
2. The default option is specified in the compiler optimization option.
3. The ROM size of “Checks arguments” is the value when *USB_CFG_ENABLE* is specified to *USB_CFG_PARAM_CHECKING* definition in *r_usb_basic_config.h* file.
4. The ROM size of “Does not check arguments” is the value when *USB_CFG_DISABLE* is specified to *USB_CFG_PARAM_CHECKING* definition in *r_usb_basic_config.h* file.

3.8 Argument

For the structure used in the argument of API function, refer to chapter "**Structures**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

4. Target Peripheral List（TPL）

For the structure used in the argument of API function, refer to chapter " **How to Set the Target Peripheral List (TPL)**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

5. Communication Device Class (CDC), PSTN and ACM

This software conforms to the Abstract Control Model (ACM) subclass of the Communication Device Class specification, as specified in detail in the PSTN Subclass document listed in “Related Documents”.

The Abstract Control Model subclass is a technology that bridges the gap between USB devices and earlier modems (employing RS-232C connections), enabling use of application programs designed for older modems.

5.1 Basic Functions

The main functions of HCDC are as follows.

1. Verify connected devices
2. Make communication line settings
3. Acquire the communication line state
4. Transfer data to and from the CDC peripheral device

5.2 Abstract Control Model Class Requests - Host to Device

This driver supports the following class requests.

For the class request processing, refer to chapter "**USB Class Requests**" in the document (Document number: R01AN2025) for *USB Basic Host and Peripheral Driver Application Note*.

Table 5-1 CDC Class Requests

Request	Code	Description
SendEncapsulatedCommand	0x00	Transmits an AT command as defined by the protocol used by the device (normally 0 for USB).
GetEncapsulatedResponse	0x01	Requests a response to a command transmitted by SendEncapsulatedCommand.
SetCommFeature	0x02	Enables or disables features such as device-specific 2-byte code and country setting.
GetCommFeature	0x03	Acquires the enabled/disabled state of features such as device-specific 2-byte code and country setting.
ClearCommFeature	0x04	Restores the default enabled/disabled settings of features such as device-specific 2-byte code and country setting.
SetLineCoding	0x20	Makes communication line settings (communication speed, data length, parity bit, and stop bit length).
GetLineCoding	0x21	Acquires the communication line setting state.
SetControlLineState	0x22	Makes communication line control signal (RTS, DTR) settings.
SendBreak	0x23	Transmits a break signal.

For details concerning the Abstract Control Model requests, refer to Table 11, “Requests - Abstract Control Model” in “USB Communications Class Subclass Specification for PSTN Devices”, Revision 1.2.

The following describes the class request data formats supported by this class driver software.

5.2.1 SendEncapsulatedCommand

The SendEncapsulatedCommand data format is shown in Table 5-2.

Table 5-2 SendEncapsulatedCommand Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SEND_ENCAPSULATED _COMMAND(0x00)	0x0000	0x0000	Data length	Control protocol command

Note: Items such as AT commands for modem control are set as Data, and wLength is set to match the length of the data.

5.2.2 GetEncapsulatedResponse

The GetEncapsulatedResponse data format is shown Table 5-3.

Table 5-3 GetEncapsulatedResponse Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	GET_ENCAPSULATED_ RESPONSE (0x01)	0x0000	0x0000	Data length	The data depends on the protocol.

Note: The response data to SendEncapsulatedCommand is set as Data, and wLength is set to match the length of the data.

5.2.3 SetCommFeature

The SetCommFeature data format is shown Table 5-4.

Table 5-4 SetCommFeature Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_COMM_FEATURE (0x02)	Feature Selector Note	0x0000	Data length	Status Either the country code or the Abstract Control Model idle setting/multiplexing setting for Feature Selector.

Note: Shown in Table 4.6 Feature selector Settings.

5.2.4 GetCommFeature Data Format

The GetCommFeature data format is shown below.

Table 5-5 GetCommFeature Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	GET_COMM_FEATURE (0x03)	Feature Selector Note	0x0000	Data length	Status Either the country code or the Abstract Control Model idle setting/multiplexing setting for Feature Selector.

Note: Shown in Table 4.6 Feature selector Settings.

A Feature selector setup is shown in Table 5-6. The Status format at the time of ABSTRACT_STATE is shown in Table 5-7.

Table 5-6 Feature Selector Settings

Feature Selector	Code	Targets	Length of Data	Description
RESERVED	0x00	None	None	Reserved
ABSTRACT_STATE	0x01	Interface	2	Selects the setting for Abstract Control Model idle state and signal multiplexing.
COUNTRY_SETTING	0x02	Interface	2	Selects the country code in hexadecimal format, as defined by ISO 3166.

Table 5-7 Status Format when ABSTRACT_STATE Selected

Bit Position	Description
D15 to D2	Reserved
D1	Data multiplexing setting 1: Multiplexing of call management commands is enabled for the Data class. 0: Multiplexing is disabled.
D0	Idle setting 1: No endpoints of the target interface accept data from the host, and data is not supplied to the host. 0: Endpoints continue to accept data and it is supplied to the host.

5.2.5 ClearCommFeature

The ClearCommFeature data format is shown in Table 5-8.

Table 5-8 ClearCommFeature Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	CLEAR_COMM_FEATURE (0x04)	Feature Selector Note	0x0000	0x0000	None

Note: Shown in Table 4.6 Feature selector Settings.

5.2.6 SetLineCoding

The SetLineCoding data format is shown in Table 5-9.

Table 5-9 SetLineCoding Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_LINE_CODING (0x20)	0x0000	0x0000	0x0000	Line Coding Structure See Table 5-10, Line Coding Structure Format

Line Coding Structure Format is shown in Table 5-10.

Table 5-10 Line Coding Structure Format

Offset	Field	Size	Value	Description
0	dwDTERate	4	Number	Data terminal speed (bps)
4	bCharFormat	1	Number	Stop bits 0 - 1 stop bit 1 - 1.5 stop bits 2 - 2 stop bits
5	bParityType	1	Number	Parity 0 - None 1 - Odd 2 - Even 3 - Mask 4 - Space
6	bDataBits	1	Number	Data bits (5, 6, 7, 8)

5.2.7 GetLineCoding

The GetLineCoding data format is shown in Table 5-11.

Table 5-11 GetLineCoding Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_LINE_CODING (0x21)	0x0000	0x0000	0x0007	Line Coding Structure See Table 5-10, Line Coding Structure Format

5.2.8 SetControlLineState

The SetControlLineState data format is shown below.

Table 5-12 SetControlLineState Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_CONTROL_LINE_STATE (0x22)	Control Signal Bitmap See Table 5-13, Control Signal Bitmap Format	0x0000	0x0000	None

Table 5-13 Control Signal Bitmap

Bit Position	Description
D15 to D2	Reserved
D1	DCE transmit function control 0 - RTS OFF 1 - RTS ON
D0	Notification of DTE ready state 0 - DTR OFF 1 - DTR ON

5.2.9 SendBreak

The SendBreak data format is shown below.

Table 5-14 SendBreak Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SEND_BREAK (0x23)	Break signal output duration	0x0000	0x0000	None

5.3 ACM Notifications from Device to Host

The class notifications supported and not supported by the software are shown in Table 5-15.

Table 5-15 CDC Class Notifications

Notification	Code	Description	Supported
NETWORK_CONNECTION	0x00	Notification of network connection state	No
RESPONSE_AVAILABLE	0x01	Response to GET_ENCAPSLATED_RESPONSE	Yes
SERIAL_STATE	0x20	Notification of serial line state	Yes

5.3.1 SerialState

The SerialState data format is shown below.

Table 5-16 SerialState Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	SERIAL_STATE (0x20)	0x0000	0x0000	0x0000	UART State bitmap See Table 5-17, UART State bitmap Format

UART State bitmap format is shown in Table 5-17.

Table 5-17 UART State bitmap Format

Bits	Field	Description
D15 to D7		Reserved
D6	bOverRun	Overrun error detected
D5	bParity	Parity error detected
D4	bFraming	Framing error detected
D3	bRingSignal	INCOMING signal (ring signal) detected
D2	bBreak	Break signal detected
D1	bTxCarrier	Data Set Ready: Line connected and ready for communication
D0	bRxCarrier	Data Carrier Detect: Carrier detected on line

5.3.2 ResponseAvailable

The ResponseAvailable data format is shown below.

Table 5-18 ResponseAvailable Data Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	RESPONSE_AVAILABLE (0x01)	0x0000	0x0000	0x0000	None

6. USB Host Communication Device Class Driver (HCDC)

6.1 Basic Functions

This software conforms to the Abstract Control Model subclass of the communication device class specification.

The main functions of HCDC are to:

1. Send class requests to the CDC peripheral
2. Transfer data to and from the CDC peripheral
3. Receive communication error information from the CDC peripheral

6.2 Structure / Union

The following structure or union is defined in *r_usb_hcdc_if.h*.

6.2.1 HCDC Request Structure

Table 6-1 describes the "UART settings" parameter structure used for the CDC requests *SetLineCoding* and *GetLineCoding*.

Table 6-1 usb_hcdc_linecoding_t Structure

Type	Member	Description	Remarks
uint32_t	dwdte_rate	Line speed	Unit: bps
uint8_t	bchar_ormat	Stop bits setting	
uint8_t	bparity_type	Parity setting	
uint8_t	bdata_bits	Data bit length	

Table 6-2 describes the "UART settings" parameter structure used for the CDC requests *SetControlLineState*.

Table 6-2 usb_hcdc_controllinestate_t Structure

Type	Member	Description	Remarks
uint16_t (D1)	brts:1	Carrier control for half duplex modems 0 - Deactivate carrier, 1 - Activate carrier	
uint16_t (D0)	bdtr:1	Indicates to DCE if DTE is present or not 0 - Not Present, 1 - Present	

Table 6-3 describes the "AT command" parameter structure used for the CDC requests *SendEncapsulatedCommand* and *GetEncapsulatedResponse*.

Table 6-3 usb_hcdc_encapsulated_t Structure

Type	Member	Description	Remarks
uint8_t	*p_data	Area where AT command data is stored	
uint16_t	wlength	Size of AT command data	Unit: byte

Table 6-4 describes the "Break signal" parameter structure used for the CDC requests *SendBreak*.

Table 6-4 usb_hcdc_breakduation_t Structure

Type	Member	Description	Remarks
uint16_t	wtime_ms	Duration of Break	Unit: ms

6.2.2 CommFeature Function Selection Union

Table 6-5 and Table 6-6 describe the “Feature Selector” parameter structure used for the CDC requests *SetCommFeature* and *GetCommFeature*, and Table 6-7 describes the parameter union.

Table 6-5 usb_hcdc_abstractstate_t Structure

Type	Member	Description	Remarks
uint16_t	rsv1:14	Reserved	
uint16_t	bdms:1	Data Multiplexed State	
iomt16_t	bis:1	Idle Setting	

Table 6-6 usb_hcdc_countrysetting_t Structure

Type	Member	Description	Remarks
uint16_t	country_code	Country code in hexadecimal format as defined in [ISO3166],	

Table 6-7 usb_hcdc_commfeature_t Union

Type	Member	Description	Remarks
usb_hcdc_abstractstate_t	abstract_state	Parameter when selecting Abstract Control Model	
usb_hcdc_countrysetting_t	country_setting	Parameter when selecting Country Setting	

6.2.3 CDC Notification Format

Table 6-8 and Table 6-9 describe the data format of the CDC notification.

Table 6-8 Response_Available notification format

Type	Member	Description	Remarks
uint8_t	bmRequestType	0xA1	
uint8_t	bRequest	RESPONSE_AVAILABLE(0x01)	
uint16_t	wValue	0x0000	
uint16_t	wIndex	Interface	
uint16_t	wLength	0x0000	
uint8_t	Data	none	

Table 6-9 Serial_State notification format

Type	Member	Description	Remarks
uint8_t	bmRequestType	0xA1	
uint8_t	bRequest	SERIAL_STATE(0x20)	
uint16_t	wValue	0x0000	
uint16_t	wIndex	Interface	
uint16_t	wLength	0x0002	
uint16_t	Data	UART State bitmap	Refer to Table 6-10

The host is notified of the "*SerialState*" when a change in the UART port state is detected. Table 6-10 describes the structure of the UART State bitmap.

Table 6-10 usb_hcdc_serialstate_t Structure

Type	Member	Description	Remarks
uint16_t (D15-D7)	rsv1:9	Reserved	
uint16_t (D6)	bover_run:1	Overrun error detected	
uint16_t (D5)	bparity:1	Parity error detected	
uint16_t (D4)	bframing:1	Framing error detected	
uint16_t (D3)	bring_signal:1	Incoming signal (Ring signal) detected	
uint16_t (D2)	bbreak:1	Break signal detected	
uint16_t (D1)	btx_carrier:1	Line connected and ready for communication	Data Set Ready
uint16_t (D0)	brx_carrier:1	Carrier detected on line	Data Carrier Detect

7. API Functions

For API used in the application program, refer to chapter "**API Functions**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

8. Configuration (r_usb_hcdc_config.h)

Please set the following according to your system.

Note:

Be sure to set *r_usb_basic_config.h* file as well. For *r_usb_basic_config.h* file, refer to chapter "**Configuration**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

1. Setting connection of multiple CDC devices

To simultaneously connect multiple CDC devices and perform USB communication, set *USB_CFG_ENABLE* as the definition below. If multiple CDC devices are not connected simultaneously, then set *USB_CFG_DISABLE*.

```
#define   USB_CFG_HCDC_MULTI   USB_CFG_ENABLE    // Multiple connection supported
#define   USB_CFG_HCDC_MULTI   USB_CFG_DISABLE   // Multiple connection not supported
```

2. Setting CDC class

Specify the device class ID of the CDC device to be connected.

```
#define   USB_CFG_HCDC_IFCLS   USB_CFG_CDC       // CDC class supported device
#define   USB_CFG_HCDC_IFCLS   USB_CFG_VEN       // Vendor class device
```

Note:

With regard to the USB serial conversion device in the marketplace, the device class ID may be the Vendor class. Check the CDC device specifications before use. If the device class is Vendor class, then set *USB_CFG_VEN*.

3. Setting pipe to be used

Set the pipe number to use for data transfer.

(1). Bulk IN/OUT transfer

Set the pipe number (PIPE1 to PIPE5) to use for Bulk IN/OUT transfer. Do not set the same pipe number.

```
#define   USB_CFG_HCDC_BULK_IN     Pipe number (USB_PIPE1 to USB_PIPE5)
#define   USB_CFG_HCDC_BULK_OUT    Pipe number (USB_PIPE1 to USB_PIPE5)
#define   USB_CFG_HCDC_BULK_IN2    Pipe number (USB_PIPE1 to USB_PIPE5)
#define   USB_CFG_HCDC_BULK_OUT2   Pipe number (USB_PIPE1 to USB_PIPE5)
```

(2). Interrupt IN transfer

Set the pipe number (PIPE6 to PIPE9) to use for Interrupt IN transfer. Do not set the same pipe number. If the USB Hub is being used, then PIPE9 cannot be set as the following definitions.

```
#define   USB_CFG_HCDC_INT_IN      Pipe number (USB_PIPE6 to USB_PIPE9)
#define   USB_CFG_HCDC_INT_IN2     Pipe number (USB_PIPE6 to USB_PIPE9)
```

Note:

a. Only if *USB_CFG_ENABLE* is set for the definition of *USB_CFG_HCDC_MULTI* in 1 above, set the pipe number for the definitions of *USB_CFG_HCDC_BULK_IN2*, *USB_CFG_HCDC_BULK_OUT2*, and *USB_CFG_HCDC_INT_IN2*.

b. If USB_CFG_DISABLE is set for the definition of *USB_CFG_HCDC_MULTI*, set *USB_NULL* for the definitions of *USB_CFG_HCDC_BULK_IN2*, *USB_CFG_HCDC_BULK_OUT2*, and *USB_CFG_HCDC_INT_IN2*.

9. Sample Application

9.1 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.

9.1.1 Data Transfer Image

Figure 9-1 shows the data transfer image.

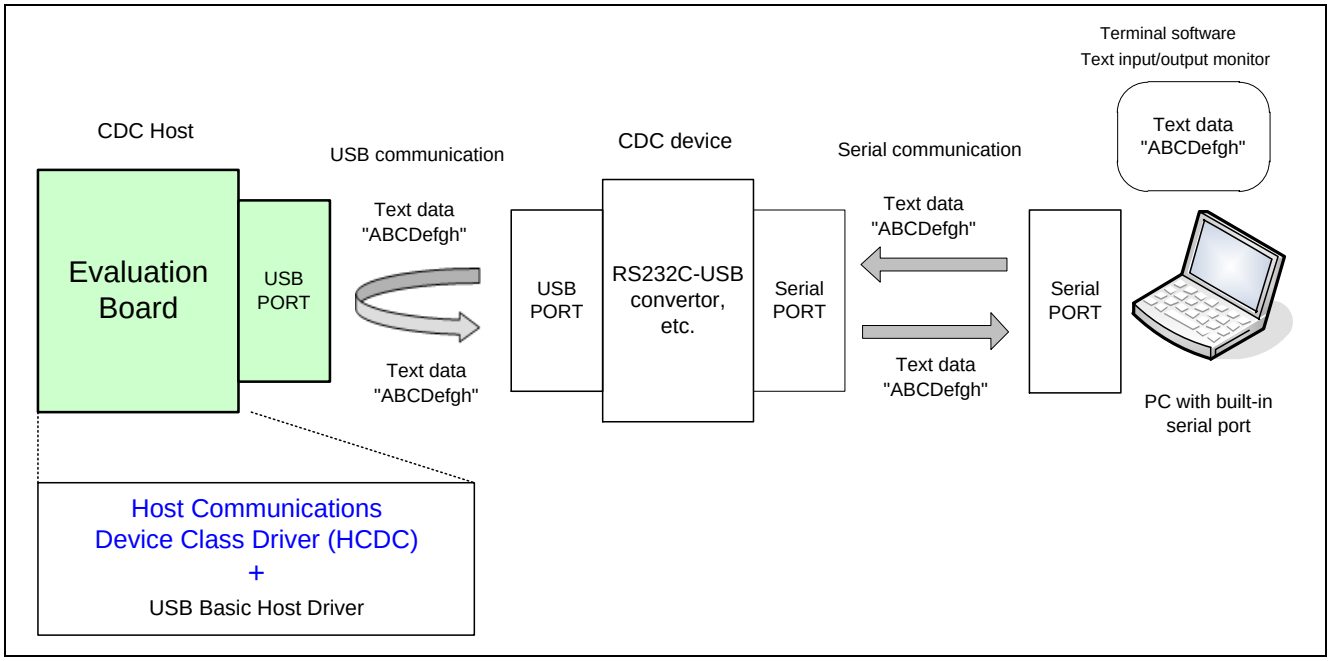


Figure 9-1 Data Transfer (Loopback) Image

9.2 Application Processing

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

9.2.1 Initial setting

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

9.2.2 Main Loop

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 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_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:

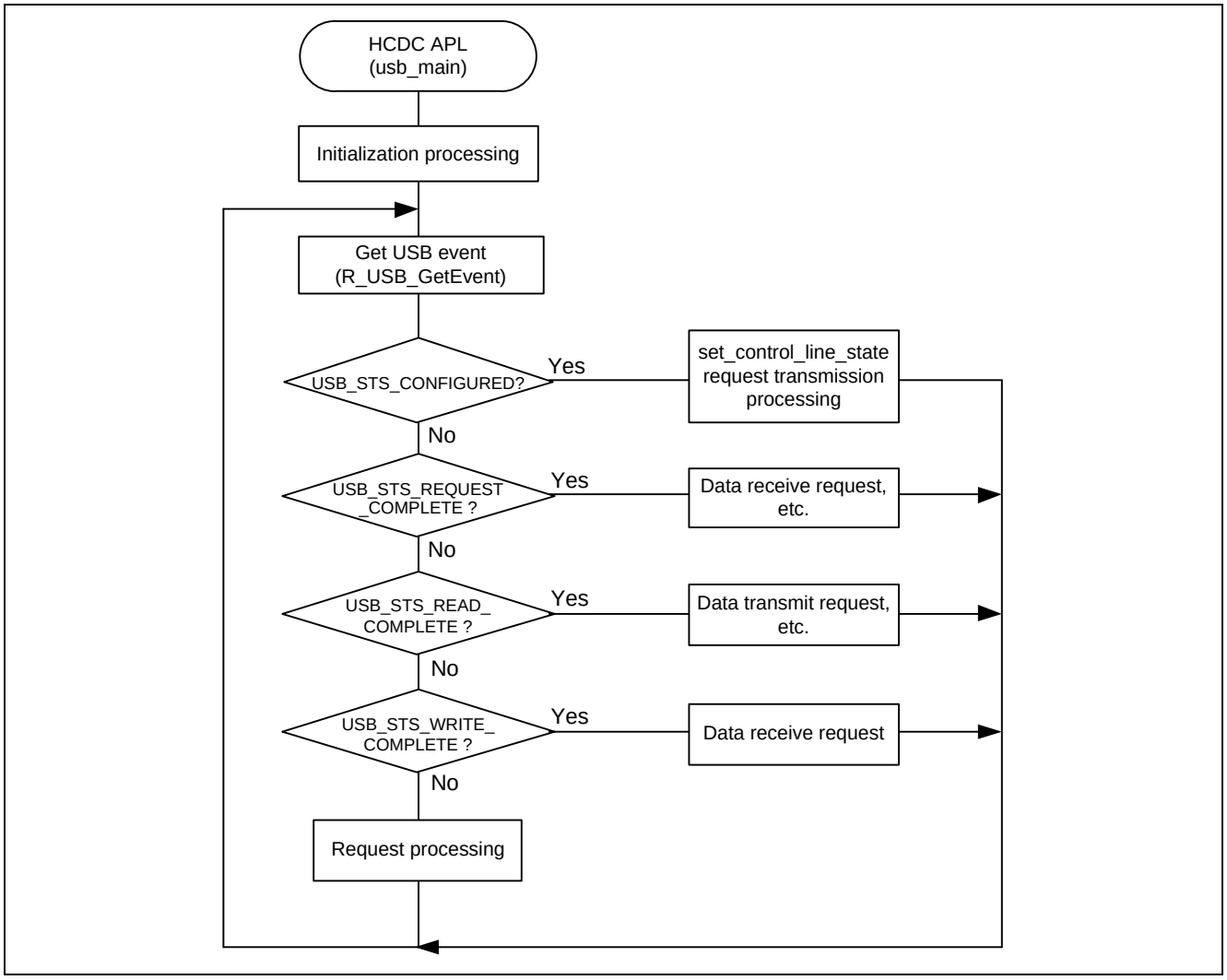


Figure 9-2 Main Loop processing

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

Make settings for the definitions listed below.

1. USE_USBIP Definition

Specify the module number of the USB module you are using.

```
#define  USE_USBIP  USE_USBIP0                 // Specify USB_IP0.
#define  USE_USBIP  USE_USBIP1                 // Specify USB_IP1.
#define  USE_USBIP  (USE_USBIP1|USE_USBIP0)    // Specify USB_IP1 and USB_IP0
```

2. 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 the CDC device.

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

3. 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
```

4. 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
```

5. 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
```

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 *USB Basic Host and Peripheral Driver Application Note* (Document number. R01AN0512EJ).

9.4 Connecting Multiple CDC Devices

Refer to the following sample programs for reference when developing application programs that connect with multiple CDC devices using a USB hub, etc.

r_usb_hcdc_apl_multi.c

10. Setup

10.1 Hardware

10.1.1 Example Operating Environment

Figure 10-1 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.

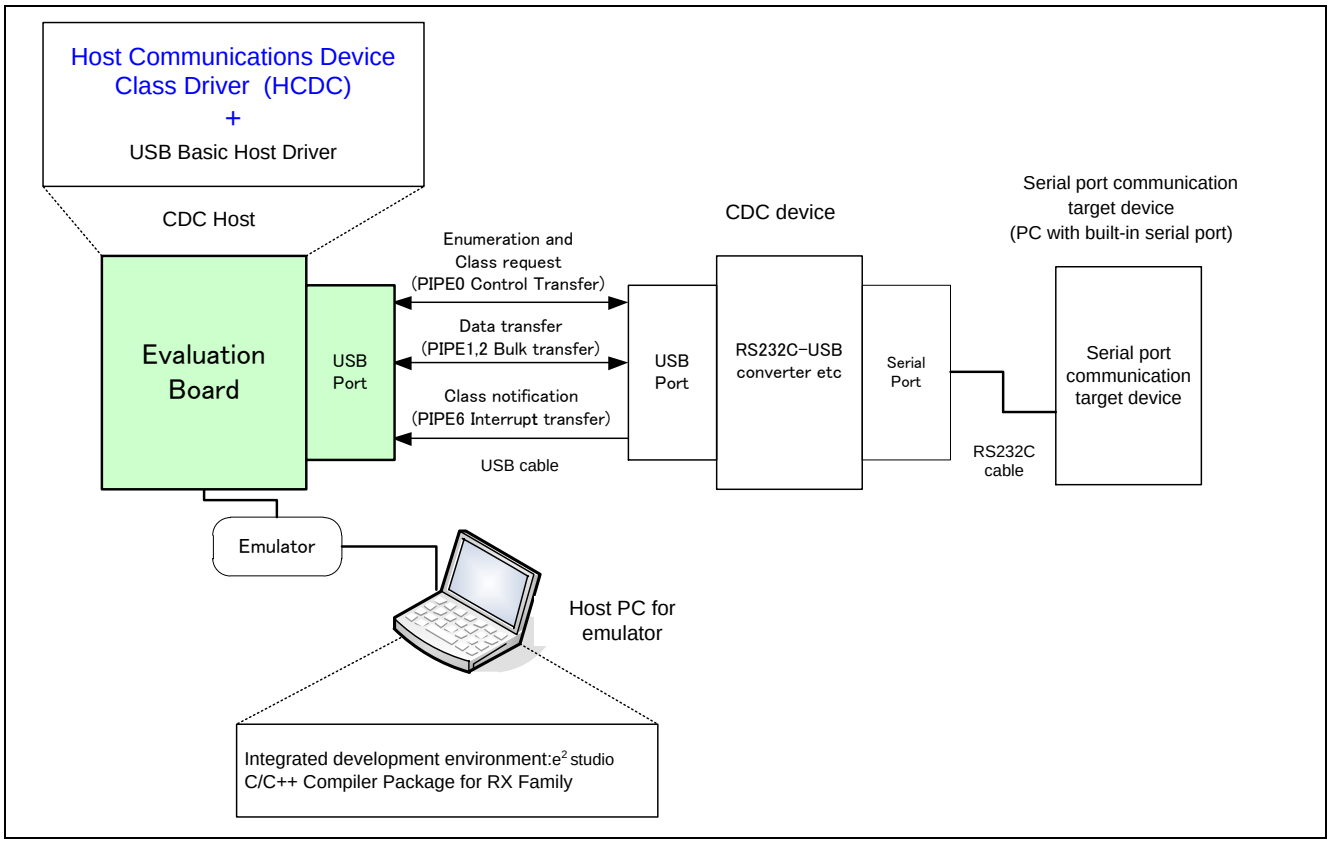


Figure 10-1 Example Operating Environment

10.1.2 RSK Setting

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

Table 10-1 RSK Setting

RSK	Jumper Setting
RSK+RX63N	J3: Shorted Pin 2-3 J4: Shorted Pin 2-3 J18:Shorted Pin1-2

Note:

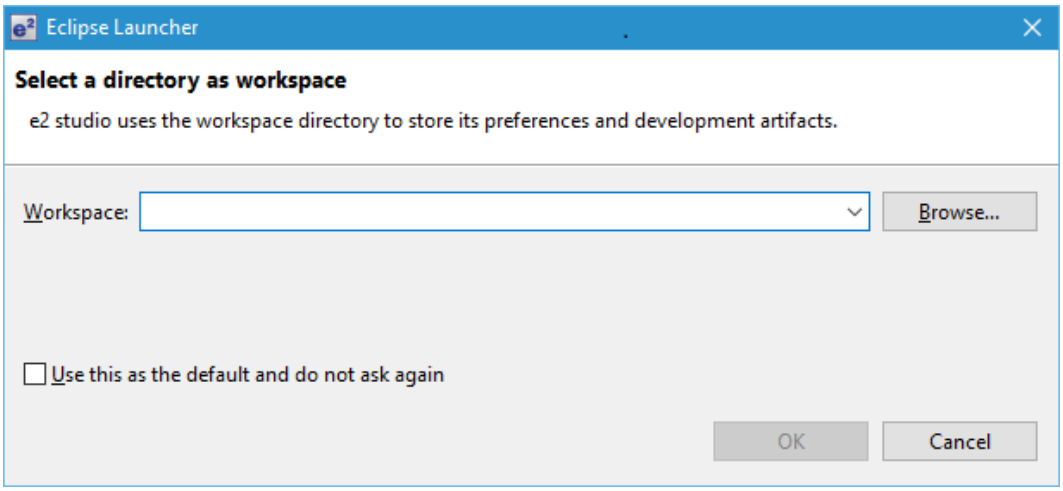
For the detail of RSK setting, refer to the user's manual of RSK.

10.2 Software

1) Setup e² studio

a) Start e² studio

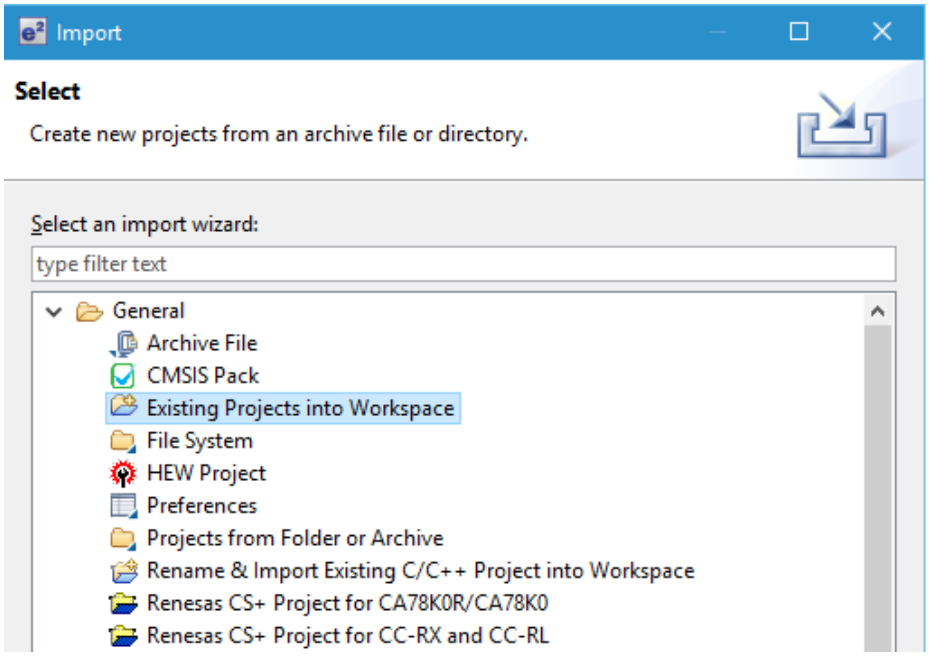
b) If you start up e² studio at first, the following dialog is displayed. Specify the folder to store the project in this dialog.



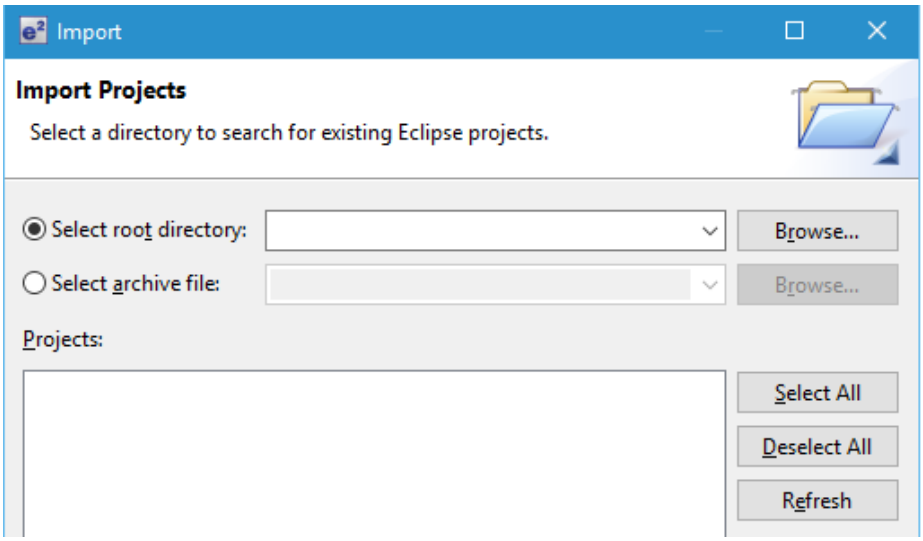
2) Import the project to the workspace

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.

11. Creating an Application

Refer to the chapter "**Creating an Application Program**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

12. 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]

Uncheck the checkbox Backup the project composition files after conversion in Project Convert Settings window.

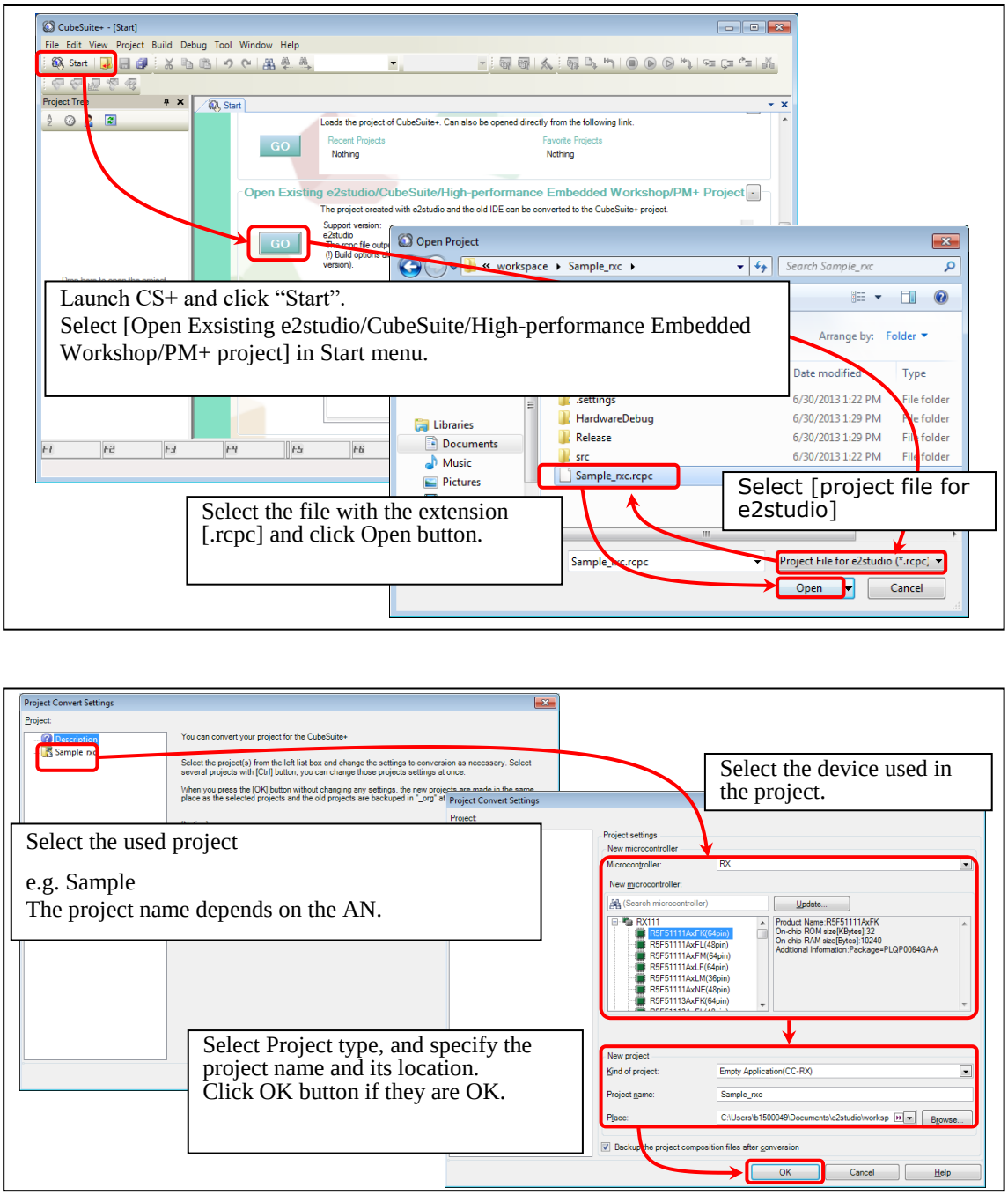


Figure 12-1 Using the e² studio project with CS+

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	Mar.22.11	—	First edition issued
1.10	July.11.07	—	Add Target Device RX630, R8A66597 Add the information on RX630 and R8A66597(Hi-Speed USB)
		30	5 Host CDC Sample Application Program (APL) ・Change the baud rate settings when pressing SW2 ・Set baud rate when pressing SW3
2.00	Mar.28.12	—	First edition issued for V.2.00
2.01	Feb.1.13	—	Add "How To Read This Document" chapter and the careless mistake is fixed.
2.10	Apr.1.13	—	First Release for V.2.10 Add Target Device RX63T.Add the information on RX63T
2.20	Sep.30.15	—	Change the application program. Change the folder structure. RX63N, RX631 and R8A66597 are deleted from Target Device. The multiple connecting of CDC device is supported.
2.30	Sep 30, 2016	—	1. Supporting DMA transfer. 2. Supporting USB Host and Peripheral Interface Driver application note(Document No.R01AN3293EJ)
2.31	Sep 30, 2017	—	1. DMA/DTC transfer has been changed to unsupported. 2. The contents of USB Host and Peripheral Interface Driver application note (Document number: R01AN3293EJ) is moved to this document and USB Host and Peripheral Interface Driver application note is deleted.
2.32	Mar 31, 2018	—	The revision of USB Basic driver has been updated.
2.33	Jul 31, 2019	—	RX63N and RX631 are added in Target Device.

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

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, Unit #06-02 Hyflux Innovation Centre, 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 8.0