APPLICATION NOTE

RX Family

R01AN0273EJ0233
Rev.2.33
Jul 31, 2019

USB Peripheral Communications Device Class Driver (PCDC)

Introduction

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

Target Device

RX62N/RX621 Group

RX63N/RX631 Group

RX630 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. Universal Serial Bus Revision 2.0 specification
2. USB Class Definitions for Communications Devices Revision 1.2
3. USB Communications Class Subclass Specification for PSTN Devices Revision 1.2
   http://www.usb.org/developers/docs/
4. RX62N/RX621 Group User’s Manual: Hardware (Document number .R01UH0033)
5. RX63N/RX631 Group User’s Manual: Hardware (Document number .R01UH0041)
6. RX630 Group User’s Manual: Hardware (Document number .R01UH0040)
7. RX63T Group User’s Manual: Hardware (Document number .R01UH0238)
8. USB Basic Host and Peripheral Driver Application Note (Document number. R01AN0512)

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

USB Devices Page
http://www.renesas.com/prod/usb/

Contents

1. Overview

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

This module supports the following functions.

- Data transfer to and from a USB host
- Response to CDC class requests
- Provision of communication device class notification transmit service

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 (*r01an0273ej0232_usb.pdf*, *r01an0512ej0232_usb.pdf*) under the "reference_documents" folder when using RX62N/RX621/ RX63T/RX630.

1.3 Limitations

PCDC is subject to the following limitations.

1. This driver does not support DMA/DTC transfer.

1.4 Terms and Abbreviations

Terms and abbreviations used in this document are listed below.

ACM	:	Abstract Control Model. This is the USB interface subclass used for virtual COM ports, based in the old V.250 (AT) command standard. See PSTN below.
APL	:	Application program
CDC	:	Communications Devices Class
CDCC	:	Communications Devices Class Communications Interface Class
CDCD	:	Communications Devices Class Data Class Interface
CPD	:	Serial Communication Port Driver
PCD	:	Peripheral control driver of USB-BASIC-F/W
PCDC	:	Communications Devices Class for peripheral
PCDCD	:	Peripheral Communications Devices Class Driver
PSTN	:	Public Switched Telephone Network, contains the ACM (above) standard.
SCI	:	Serial Communication Interface
USB	:	Universal Serial Bus
	:	USB Basic Peripheral Driver for Renesas USB device (non-OS)

2. Software Configuration

Figure 2-1 shows the configuration of the modules related to PCDC.

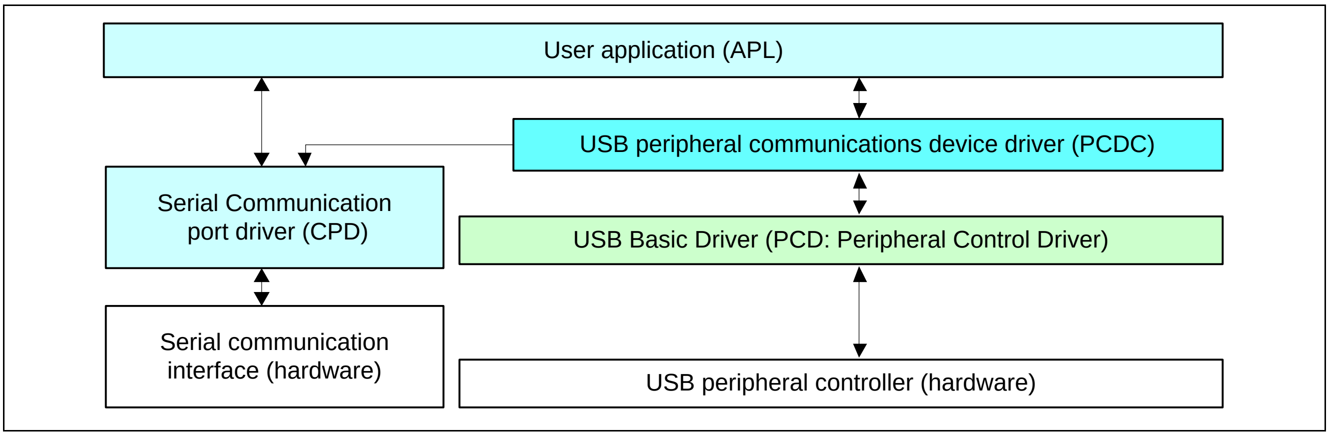

User application (APL)
USB peripheral communications device driver (PCDC)
Serial Communication port driver (CPD)
USB Basic Driver (PCD: Peripheral Control Driver)
Serial communication interface (hardware)
USB peripheral controller (hardware)

Figure 2-1 Source Code Block Diagram

Table 2.1 Modules

Module	Description
APL	User application program.
PCDC	Sends requests from the APL for requests and data communication involving the CDC to the PCD.
PCD	USB peripheral hardware control driver.
CPD	Serial port control driver

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
Host Environment	The operation of this USB Driver module connected to the following OSes has been confirmed. 1. Windows® 7 2. Windows® 8.1 3. Windows® 10

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) USBI1 Interrupt (Vector number: 38) / USBR0 Interrupt (Vector number: 91)

3.4 Header Files

All API calls and their supporting interface definitions are located in r_usb_basic_if.h and r_usb_pcdc_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 **6, Configuration (r_usb_pcdc_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	17.6K bytes (Note 3)	17.1K bytes (Note 4)

RAM size	9.0K bytes	9.0K 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. CDC, PSTN, and ACM (Abstract Control Model)

4.1 Basic Functions

This software conforms to the Abstract Control of the CDC PSTN Subclass. See.0 below.

4.2 Abstract Control Model Overview

The Abstract Control Model subclass of CDC 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. The class requests and class notifications supported are listed below.

4.2.1 Class Requests (Host to Peripheral)

This driver notifies to the application program when receiving the following class request.

For the class request processing, refer to chapter "**USB Class Requests**" in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

Table 4-1 CDC class requests

Request	Code	Description
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.

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.

4.2.2 Data Format of Class Requests

The data format of the class requests supported by the class driver software is described below.

1. SetLineCoding

This is the class request the host transmits to perform the UART line setting.

The SetLineCoding data format is shown below.

Table 4-2 SetLineCoding Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_LINE_CODING (0x20)	0x00	0x00	0x07	Line Coding Structure See Table 4-3, Line Coding Format

Table 4-3 Line Coding 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
6	BdataBits	1	Number	Data bits (5, 6, 7, 8)

The following shows the setting that this S/W supports.

DwDTERate:	1200bps/2400bps/4800bps/9600bps/14400bps/19200bps/38400bps/57600bps/115200bps
BcharFormat:	1 Stop bit/2 Stop bit
BparityType:	None/Odd/Even
BdataBits:	7bit/8bit

2. GetLineCoding

This is the class request the host transmits to request the UART line state.

The GetLineCoding data format is shown below.

Table 4-4 SetLineCoding Format

bmRequestType	bRequest	wValue	WIndex	wLength	Data
0xA1	GET_LINE_CODING (0x21)	0x00	0x00	0x07	Line Coding Structure See Table 4-2, SetLineCoding Format

3. SetControlLineState

This is a class request that the host sends to set up the signal for flow controls of UART.

This software does not support RTS/DTR control.

The SET_CONTROL_LINE_STATE data format is shown below.

Table 4-5 SET_CONTROL_LINE_STATE Format

bmRequestType	bRequest	WValue	wIndex	wLength	Data
0x21	SET_CONTROL_LINE_STATE (0x22)	Control Signal Bitmap See Table 4-6, Control Signal Bitmap	0x00	0x00	None

Table 4-6 Control Signal Bitmap

Bit Position	Description	
D15 to D2	Reserved (reset to 0)	
D1	DCE transmit function control	0 - RTS Off 1 - RTS On
D0	Notification of DTE ready state	0 - DTR Off 1 - DTR On

4.2.3 Class Notifications (Peripheral to Host)

Whether or not a class notification is supported is shown in .

Table 4-7 CDC Class Notifications

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

1. Serial State

The host is notified of the serial state when a change in the UART port state is detected.

This software supports the detection of overrun, parity and framing errors. A state notification is performed when a change from normal state to error is detected. However, notification is not continually transmitted when an error is continually detected.

The SerialState data format is shown below.

Table 4-8 SerialState Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	SERIAL_STATE (0x20)	0x00	0x00	0x02	UART State bitmap See Table 4-9, UART state bitmap format

Table 4-9 UART state bitmap format

Bits	Field	Description	Supported
D15～D7		Reserved	-
D6	b_over_run	Overrun error detected	Yes
D5	b_parity	Parity error detected	Yes
D4	b_framing	Framing error detected	Yes
D3	b_ring_signal	INCOMING signal (ring signal) detected	No
D2	b_break	Break signal detected	No
D1	btx_arrier	Data Set Ready: Line connected and ready for communication	No
D0	brx_carrier	Data Carrier Detect: Carrier detected on line	No

4.3 PC Virtual COM-port Usage

The CDC device can be used as a virtual COM port when operating in Windows OS.

Use a PC running Windows OS, and connect an RSK board. After USB enumeration, the CDC class requests *GetLineCoding* and *SetControlLineState* are executed by the target, and the CDC device is registered in Windows Device Manager as a virtual COM device.

Registering the CDC device as a virtual COM-port in Windows Device Manager enables data communication with the CDC device via a terminal app such as "HyperTerminal" which comes standard with Windows OS. When changing settings of the serial port in the Windows terminal application, the UART setting is propagated to the firmware via the class *request SetLineCoding*.

Data input (or file transmission) from the terminal app window is transmitted to the RSK board using endpoint 2 (EP2); data from the RSK board side is transmitted to the PC using EP1.

When the last packet of data received is the maximum packet size, and the terminal determines that there is continuous data, the received data may not be displayed in the terminal. If the received data is smaller than the maximum packet size, the data received up to that point is displayed in the terminal.

The received data is outputted on the terminal when the data less than Maximum packet size is received.

5. 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*.

6. Configuration (r_usb_pcdc_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 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 for the definitions of *USB_CFG_PCDC_BULK_IN* and *USB_CFG_PCDC_BULK_OUT*.

```
#define     USB_CFG_PCDC_BULK_IN          Pipe number (USB_PIPE1 to USB_PIPE5)
#define     USB_CFG_PCDC_BULK_OUT         Pipe number (USB_PIPE1 to USB_PIPE5)
```

(2). Interrupt IN transfer

Set the pipe number (PIPE6 to PIPE9) to use for Interrupt IN transfer.

```
#define     USB_CFG_PCDC_INT_IN           Pipe number (USB_PIPE6 to USB_PIPE9)
```

7. Sample Application

7.1 Application Specifications

The main functions of the APL are as follows:

1) Echo(Loopback) mode (Note1)

   Transmits data received from the USB host back to the USB host.

2) USB-serial converter mode (Note1) (Note2)

   Transmits data received from the USB host back to a COM port and data received from the COM port to the USB host. Note that if a COM port error (parity error, framing error, or overrun error) occurs, the APL sends a “SerialState” class notification to the USB host.

3) Low-power functionality

   This functionality transitions the MCU to low-power mode according to the status of the USB.

   a) The APL transitions the MCU to sleep mode when the USB is suspended.
   b) When the USB is detached (disconnected), the APL transitions the MCU to software standby mode.

（Note1） The selection of Echo(Loopback) mode or USB-serial converter mode is made by setting in the file “r_usb_pcdc_apl_config.h.”

（Note2） The following jumper settings must be made on the RSK board to use USB-serial converter mode:

Table 7-1 USB-Serial Converter Mode Jumper Settings

RSK+RX63N		
Port	Signal	Jumper Setting
P20	TXD0	J12.1-2
P21	RXD0	J13.1-2

7.2 Application Processing

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

7.2.1 Initial Setting

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

7.2.2 Main Loop (Echo mode)

In Echo mode, loop-back processing in which data sent by the USB host is received and then transmitted unmodified back to the USB host takes place 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 enumeration with the USB host completes, *USB_STS_CONFIGURED* is set as the return value. When the APL confirms *USB_STS_CONFIGURED*, it calls the *R_USB_Read* function to make a data receive request for data sent by the USB host.
2. When enumeration with the USB host completes, the USB host sends an CDC class request to the CDC device. After it receives the CDC class request, the CDC device calls the *R_USB_GetEvent* function and the return value is set to *USB_STS_REQUST*. When the APL confirms *USB_STS_REQUEST*, it analyzes the received class request and performs processing corresponding to it.
3. When the *R_USB_GetEvent* function is called after reception of data from the USB host has completed, *USB_STS_READ_COMPLETE* is set as the return value. When the APL confirms *USB_STS_READ_COMPLETE*, it calls the *R_USB_Write* function to make a data transmit request to transmit the received data to the USB host.
4. When the *R_USB_GetEvent* function is called after transmission of data to the USB host 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 USB host.
5. The processing in steps 3 and 4, above, is repeated.
6. When it confirms reception of a suspend signal from the USB host or DETACH, the APL performs processing to transition the CDC device (RSK) to low-power mode. For information on low-power mode, refer to 7.3, MCU Low power consumption processing. Note that confirmation of reception of a suspend signal or DETACH is performed by referencing the return value (*USB_STS_SUSPEND* or *USB_STS_DETACH*) of the *R_USB_GetEvent* function.

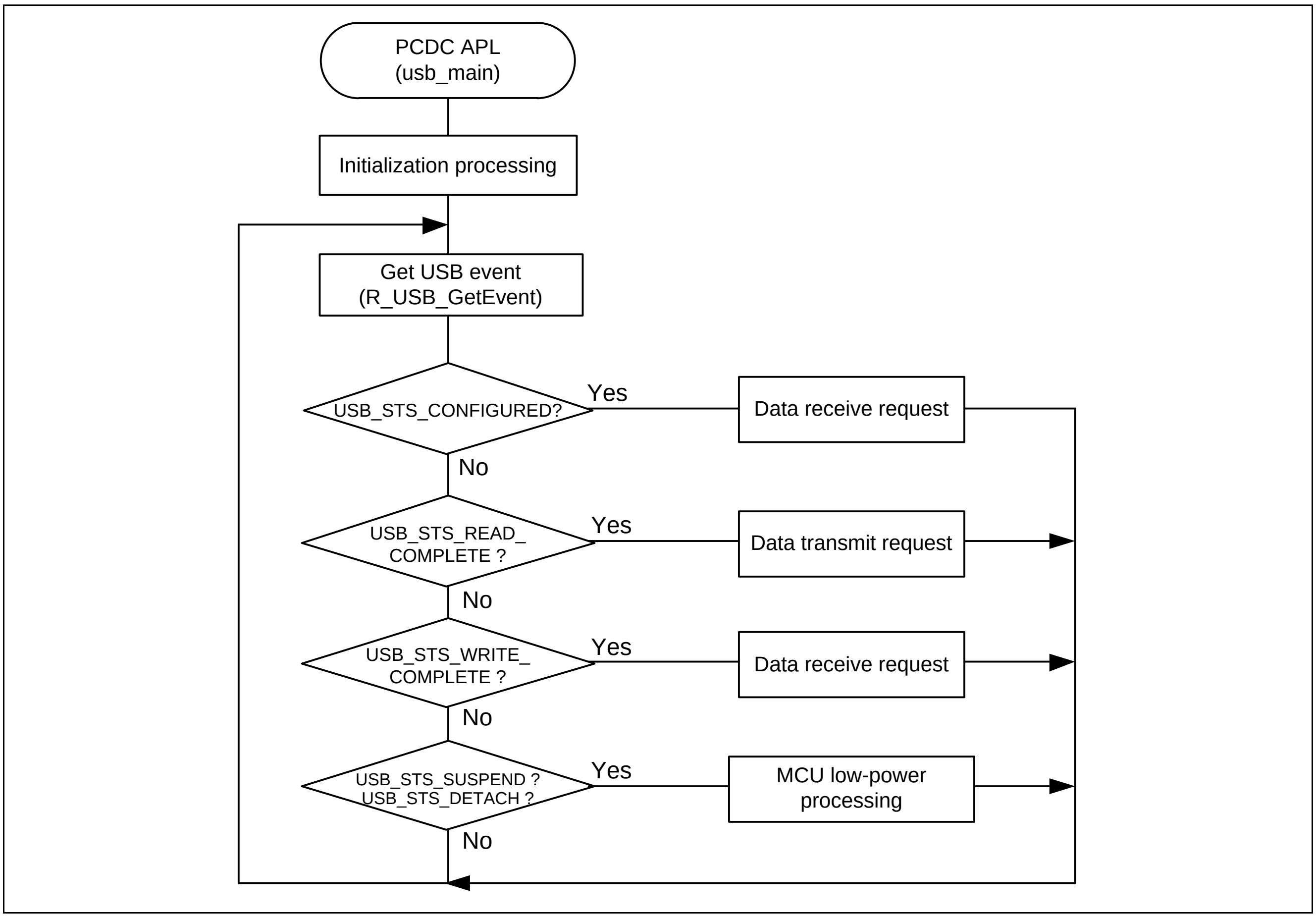


Figure 7-1 Main Loop processing (Echo mode)

7.2.3 Main Loop (USB-serial converter mode)

The processing performed in USB serial convert-back mode is described below.

a. Reception of data from the USB host, and transmission of the received data to the COM port.

b. Transmission to the USB host of data received from the COM port.

An overview of the processing of the main loop is presented below.

1. When the *R_USB_GetEvent* function is called after enumeration with the USB host completes, *USB_STS_CONFIGURED* is set as the return value. When the APL confirms *USB_STS_CONFIGURED*, it calls the *R_USB_Read* function to make a data receive request for data sent by the USB host.

2. When enumeration with the USB host completes, the USB host sends an CDC class request to the CDC device. After it receives the CDC class request, the CDC device calls the *R_USB_GetEvent* function and the return value is set to *USB_STS_REQUST*. When the APL confirms *USB_STS_REQUEST*, it analyzes the received class request and performs processing corresponding to it.

3. When the *R_USB_GetEvent* function is called after processing of the class request mentioned in 2, above, completes, the return value is set to *USB_STS_REQUST_COMPLETE*. The APL performs processing to make request information settings, etc.

4. When the *R_USB_GetEvent* function is called after reception of Bulk data by the USB host has completed, *USB_STS_READ_COMPLETE* is set as the return value. When the APL confirms *USB_STS_READ_COMPLETE*, it sets the SCI transmit request flag in order to send the received bulk data to the COM port. This flag is referenced in step 6, below.

5. When the *R_USB_GetEvent* function is called after sending of USB data (refer to step 6, below) to the USB host has completed, *USB_STS_WRITE_COMPLETE* is set as the return value. When the APL confirms *USB_STS_WRITE_COMPLETE*, it references member type in the *usb_ctrl_t* structure to identify the device class on which data transmission has completed. It then sets the completion flag corresponding to the device class on which data transmission has completed. This flag is referenced in step 6, below.

6. The following transmission processing is performed after completing the above processing. Note that before the transmission processing described below, the flags set in steps 4 and 5 above are referenced to determine if transmission is possible.

   (1). SCI transmission processing to send bulk data received from the USB host to the COM port

   (2). If a SCI error (parity error, framing error, overrun error, etc.) has been detected, class notification (serial state) transmit request processing to notify the USB host

   (3). Data transmit request processing to send the data received from the COM port to the USB host

7. If reception of a suspend signal from the USB host or DETACH is confirmed while steps 4 to 6 to, above, are repeating, the APL performs processing to transition the CDC device (RSK) to low-power mode. For information on low-power mode, refer to 7.3, MCU Low power consumption processing. Note that confirmation of reception of a suspend signal or DETACH is performed by referencing the return value (*USB_STS_SUSPEND* or *USB_STS_DETACH*) of the *R_USB_GetEvent* function.

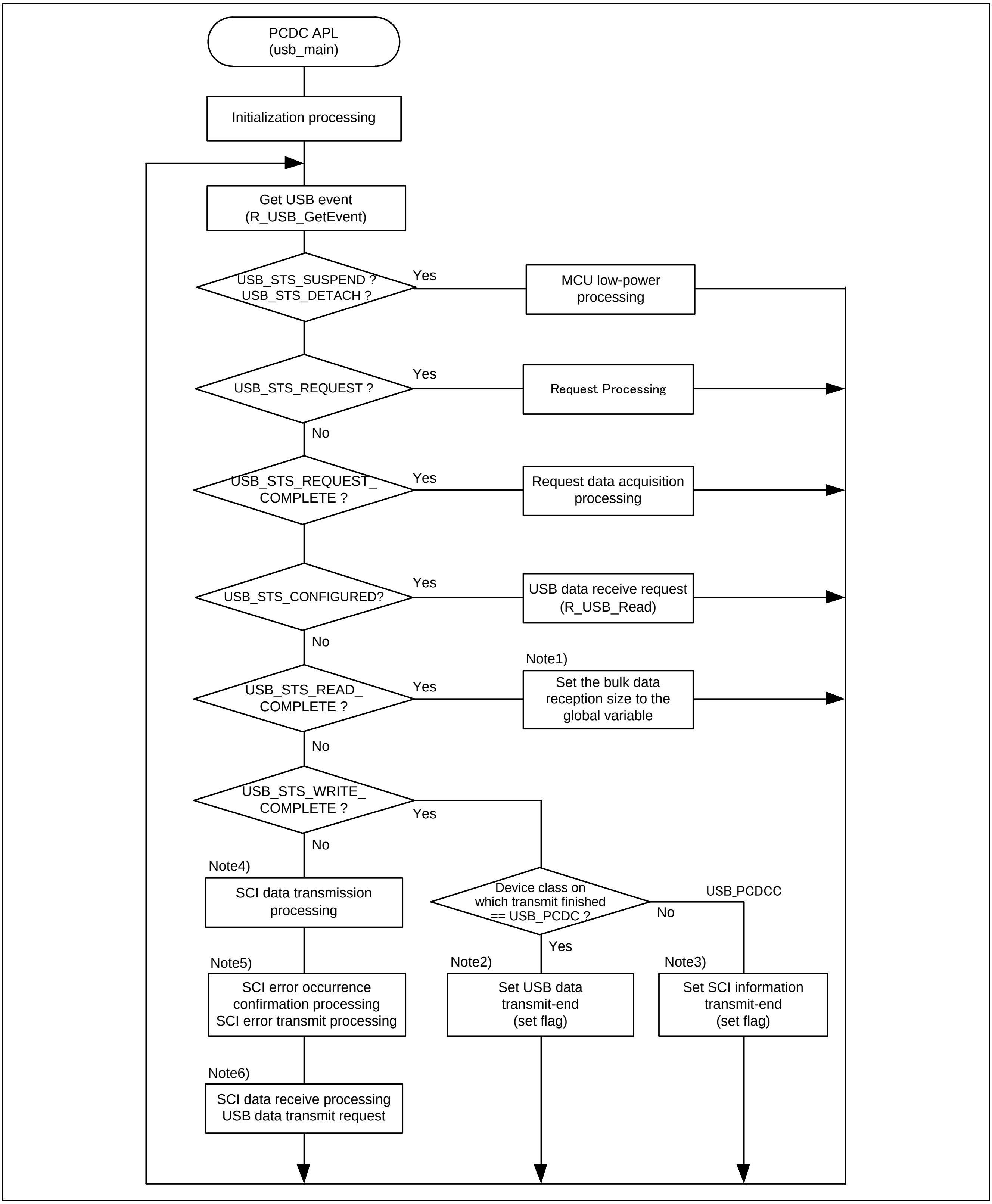

PCDC APL
(usb_main)
Initialization processing
Get USB event
(R_USB_GetEvent)
USB_STS_SUSPEND ?
USB_STS_DETACH ?
Yes
MCU low-power processing
USB_STS_REQUEST ?
Yes
Request Processing
No
USB_STS_REQUEST_ COMPLETE ?
Yes
Request data acquisition processing
USB_STS_CONFIGURED?
Yes
USB data receive request
(R_USB_Read)
No
Note1)
USB_STS_READ_ COMPLETE ?
Yes
Set the bulk data reception size to the global variable
No
USB_STS_WRITE_ COMPLETE ?
Yes
No
Note4)
SCI data transmission processing
Device class on which transmit finished == USB_PCDC ?
USB_PCDCC
No
Yes
Note5)
SCI error occurrence confirmation processing
SCI error transmit processing
Note2)
Set USB data transmit-end (set flag)
Note3)
Set SCI information transmit-end (set flag)
Note6)
SCI data receive processing
USB data transmit request

Figure 7-2 Main Loop processing (USB-serial converter mode)

(Note1) Sets the SCI transmit request flag. This flag is referenced in the processing described in (Note4).

(Note2) Sets the USB data transmit-end flag. This flag is referenced in the processing described in (Note6).

(Note3) Sets the SCI information (serial state) transmit-end flag. This flag is referenced in the processing described in (Note5).

(Note4) References the flag mentioned in (Note1) and performs SCI data transmit processing. If SCI data transmission completes normally, the SCI transmit request flag is cleared. If SCI data transmission fails, the SCI transmit request flag it not cleared so that SCI data transmit processing will be retried.

(Note5) Performs processing to check the SCI status for the occurrence of an error. If an error has occurred, the *R_USB_Write* function is used to send error information to the USB host.

(Note6) References the flag mentioned in (Note2) to confirm whether USB transmission is possible. If transmission is possible, the *USB_Write* function is used to send the received SCI data to the USB host. Note that confirmation of reception of data from the COM port is not performed until transmission of data to the USB host completes.

7.3 MCU Low power consumption processing

MCU low-power processing occurs when the conditions in Table 7-2 are met, causing a transition to low-power mode. To enable this processing, specify *APL_ENABLE* to *USB_SUPPORT_LPW* definition in the *r_usb_pcdc_apl_config.h file*.

Table 7-2 Conditions for Transition to Low-Power Mode

Transition Condition		Transition Status
VBUS	USB State	
OFF	—	Software standby mode
ON	Suspend Configured	Sleep mode
ON	Other than Suspend Configured	Normal mode (program running)

(1). When the CDC device (RSK) detaches from the USB host (VBUS OFF), the APL performs processing to transition the MCU to software standby mode. Recovery from software standby mode occurs when the CDC device (RSK) attaches to the USB host.

(2). When a suspend signal sent by the USB host is received while the CDC device (RSK) is connected to the USB host, the APL performs processing to transition the MCU to sleep mode. Note that recovery from sleep mode occurs when a resume signal is received from the USB host.

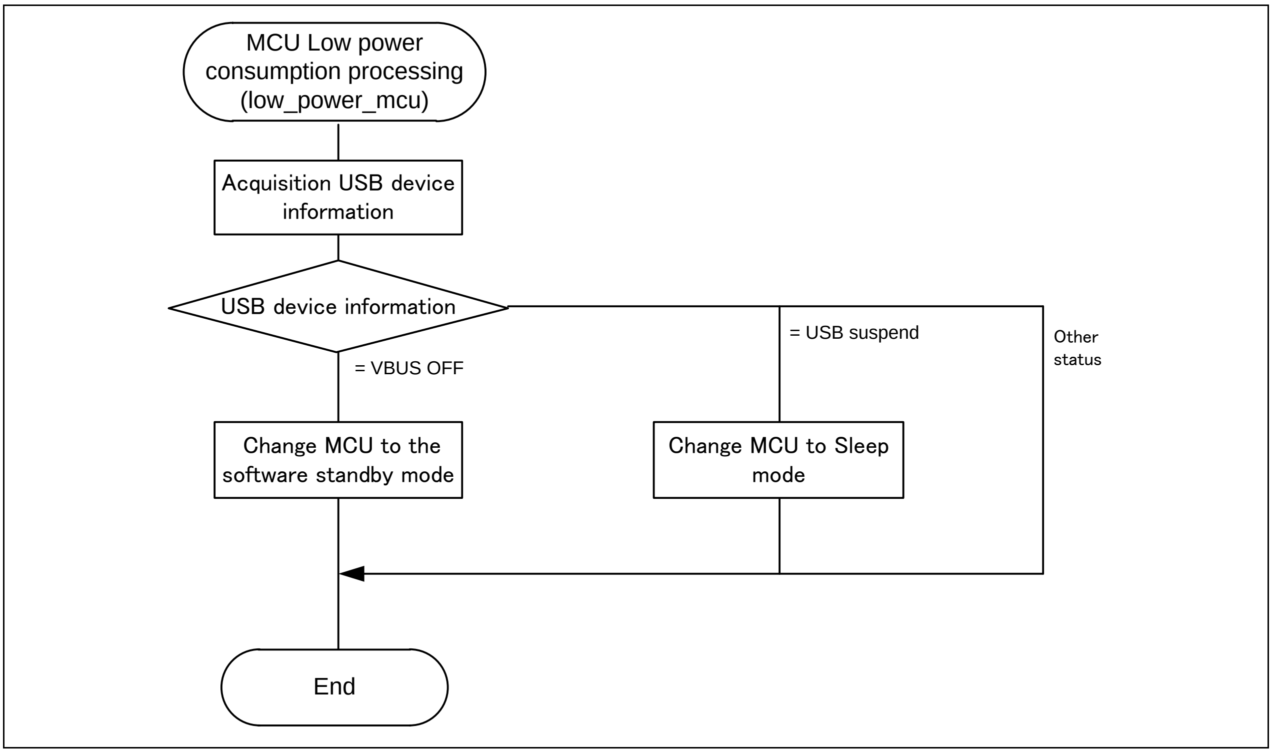

MCU Low power consumption processing (low_power_mcu)
Acquisition USB device information
USB device information
= VBUS OFF
= USB suspend
Other status
Change MCU to the software standby mode
Change MCU to Sleep mode
End

Figure 7-3 Flowchart of MCU Low Power Consumption Processsing

7.4 Configuration File for the application program (r_usb_pcdc_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. Specify one of the following settings for the USE_USBIP definition.

```
#define     USE_USBIP           USE_USBIP0    // Specify USB_IP0.
#define     USE_USBIP           USE_USBIP1    // Specify USB_IP1.
```

[Note]

Specify *USE_USBIP0* when using RX63T or RX630.

2. USB_SUPPORT_SPEED Definition

Specify the USB operation speed (Hi-speed or Full-speed) which CDC device (RSK) supports.

```
#define     USB_SUPPORT_SPEED   FULL_SPEED   // Full-Speed setting
```

3. OPERATION_MODE Definition

Specify one of the following settings for the *OPERATION_MODE* definition.

```
#define     OPERATION_MODE    USB_ECHO     // Echo mode
#define     OPERATION_MODE    USB_UART     // USB-serial conversion mode
```

4. Low-Power Function Definition

Specify whether or not the low-power function will be used. If the low-power function will be used, specify *USB_APL_ENABLE* to *USB_SUPPORT_LPW* definition.

```
#define     USE_SUPPORT_LPW    USB_APL_DISABLE   // No use the low-power function
#define     USB_SUPPORT_LPW    USB_APL_ENABLE    // Use the low-power function
```

5.

5. 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).

7.5 Descriptor

The PCDC's descriptor information is contained in r_usb_pcdc_descriptor.c. Also, please be sure to use your vendor ID.

8. CDC Driver Installation

If USB Host is PC, CDC driver must be installed in the PC. When you connect RSK which this PCDC sample program is written in to your PC, the wizard shown in Figure 10-1 will appear on your screen and prompt the CDC driver installation

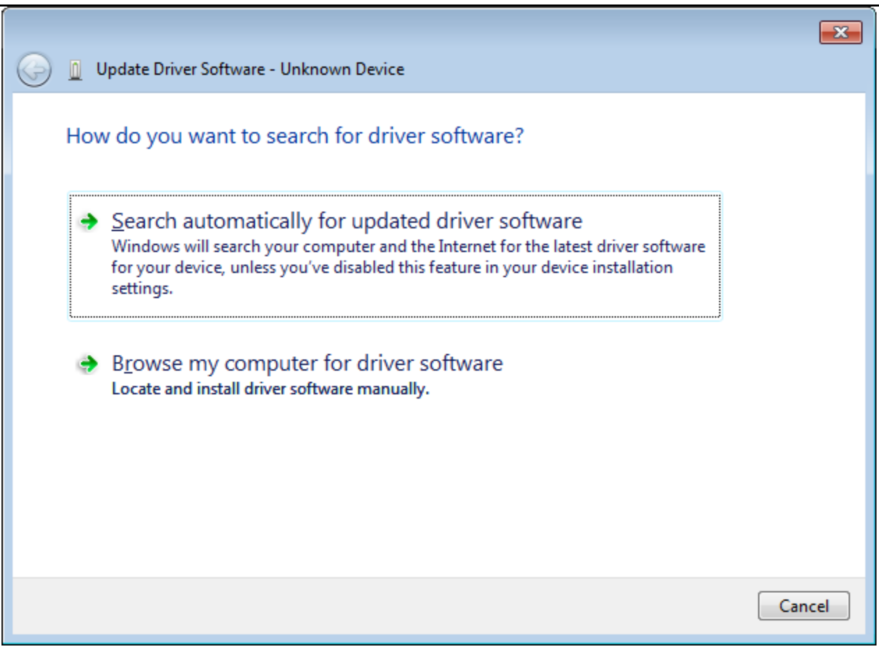
(1). Select **Update Driver Software** from the device manager.

(2). Select “**Browse my computer for driver software**”.

Note:

(1). It is not necessary the following installation work for CDC driver when using Window® 10.

(2). The catalog file with the digital signature is required when using Windows® 8.1. The customer needs to create this catalog file.


Update Driver Software - Unknown Device

How do you want to search for driver software?

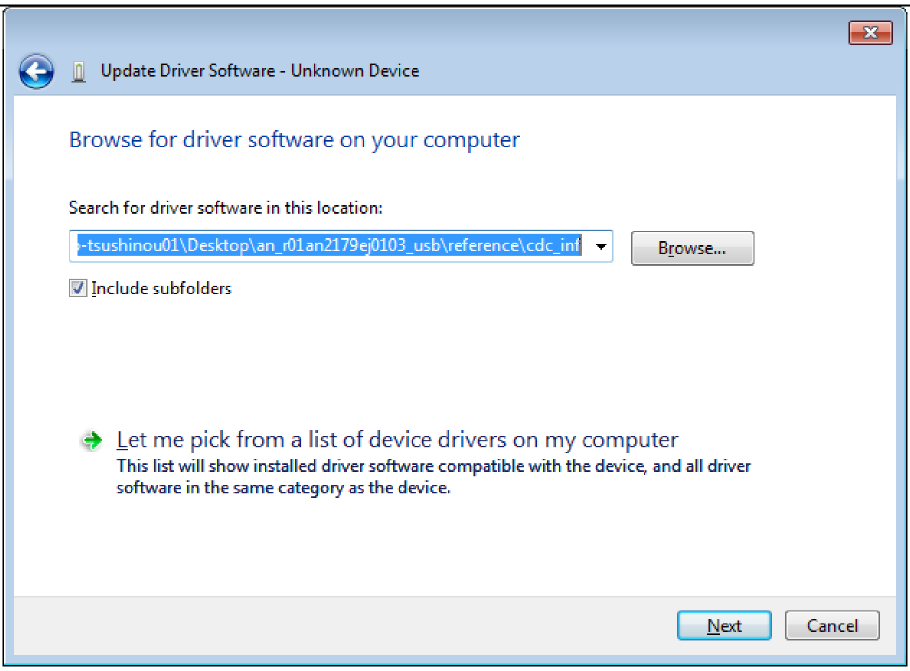
→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software
Locate and install driver software manually.

Figure 8-1 New Hardware Search Wizard

(3). Select **“Browse for driver software on your computer"**

Click **Browse**, specify the folder in which the *CDC_Demo.inf* is stored, then click **“Next”**



Update Driver Software - Unknown Device

Browse for driver software on your computer

Search for driver software in this location:

-tsushinou01\Desktop\an_r01an2179ej0103_usb\reference\cdc_inf

☑ Include subfolders

Let me pick from a list of device drivers on my computer

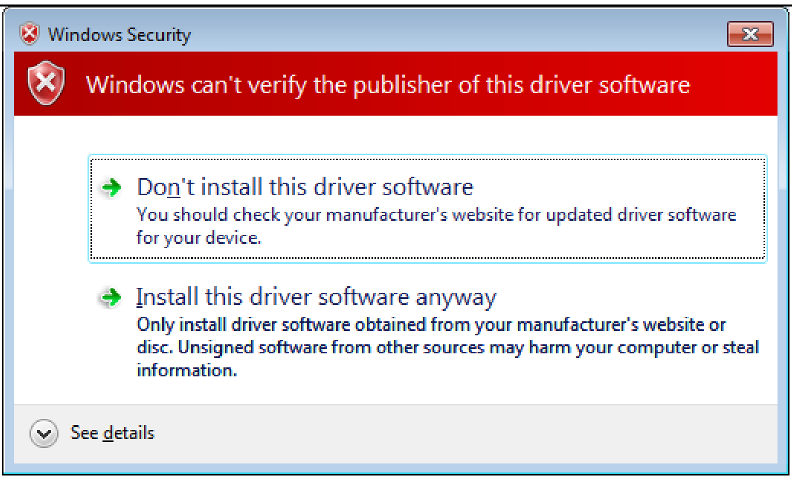
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Figure 8-2 Select Driver Location

Note:

The *CDC_Demo.inf* file is stored in "r_usb_pcdc\utilities" folder in the package.

(4). If the following installation confirmation screen appears, click **“Browse for driver software on your computer”**



Windows Security

Windows can't verify the publisher of this driver software

Don't install this driver software

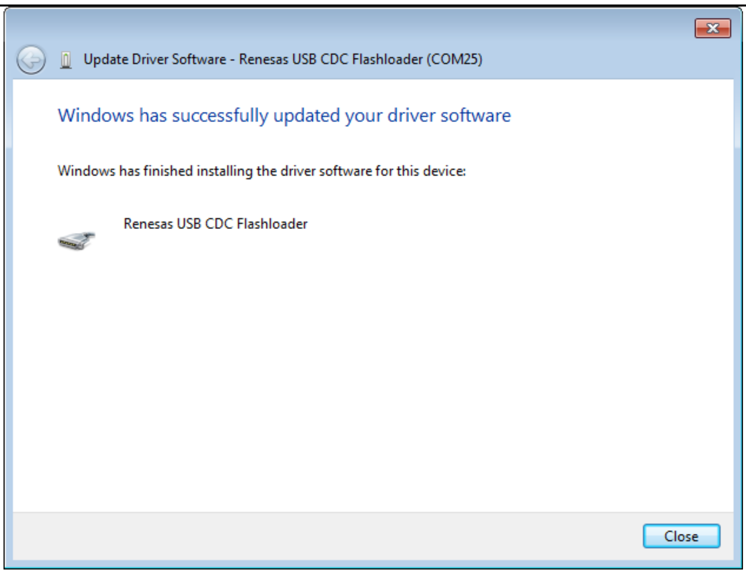
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway

Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Figure 8-3 Installation Confirmation Screen

(5). When the following window appears, the CDC driver has been successfully installed. Click **"Close."**



Update Driver Software - Renesas USB CDC Flashloader (COM25)

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Renesas USB CDC Flashloader

Figure 8-4 Installation Complete

9. Setup

9.1 Hardware

9.1.1 Example Operating Environment

Figure 9-1 shows an example operating environment for the PCDC. Refer to the associated instruction manuals for details on setting up the evaluation board and using the emulator, etc.

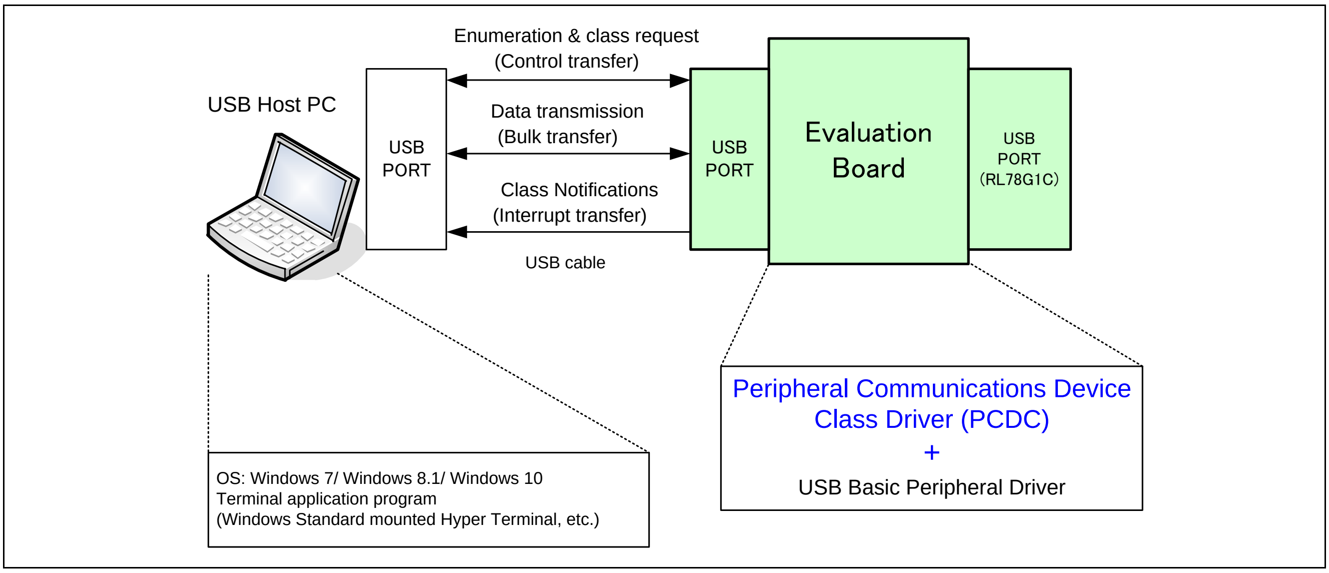

Enumeration & class request
(Control transfer)
USB Host PC
USB PORT
Data transmission
(Bulk transfer)
Class Notifications
(Interrupt transfer)
USB cable
USB PORT
Evaluation Board
USB PORT (RL78G1C)
Peripheral Communications Device Class Driver (PCDC)
+
USB Basic Peripheral Driver
OS: Windows 7/ Windows 8.1/ Windows 10
Terminal application program
(Windows Standard mounted Hyper Terminal, etc.)

Figure 9-1 Example Operating Environment

9.1.2 RSK Setting

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

Table 9-1 **RSK Setting**

RSK	Jumper Setting
RSK+RX63N	J3: Shorted Pin1-2 J4: Shorted Pin1-2 J18:Shorted Pin2-3

Note:

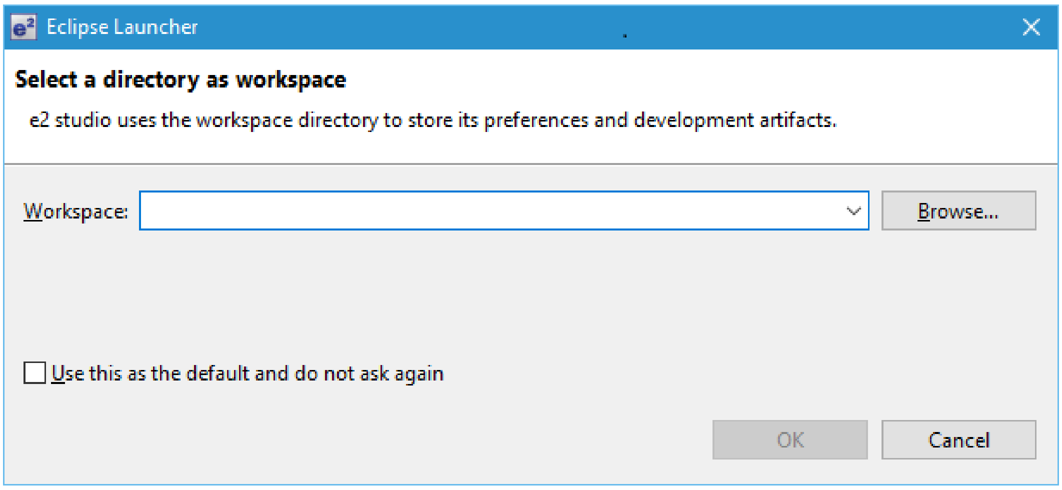
For the detail of RSK setting, refer to the user's manual of RSK.

9.2 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.



Eclipse Launcher

Select a directory as workspace

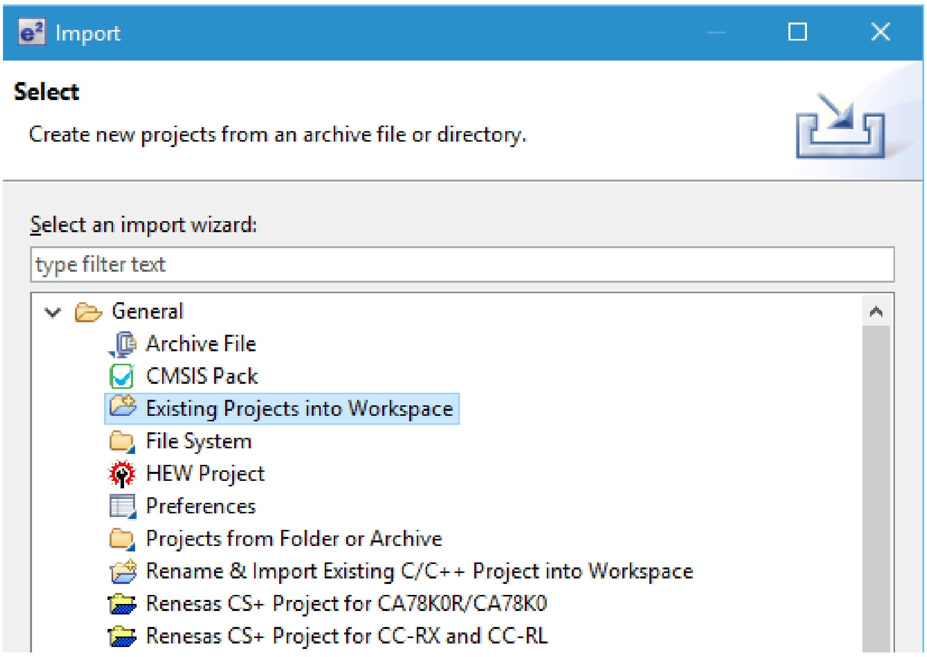
e2 studio uses the workspace directory to store its preferences and development artifacts.

Workspace:

2) Import the project to the workspace

a) Select [File] > [Import]

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



Import

Select

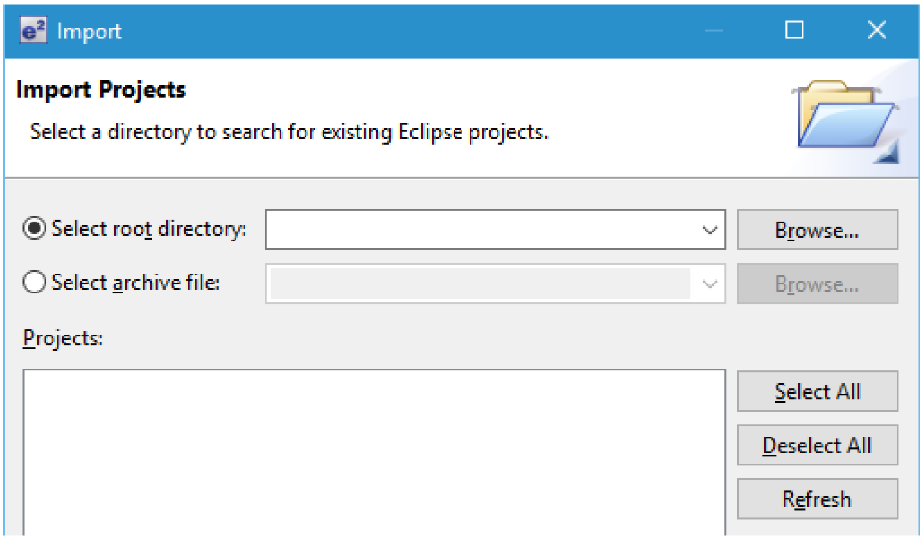
Create new projects from an archive file or directory.

Select an import wizard:

type filter text

- General
  - Archive File
  - CMSIS Pack
  - Existing Projects into Workspace
  - File System
  - HEW Project
  - Preferences
  - Projects from Folder or Archive
  - Rename & Import Existing C/C++ Project into Workspace
  - Renesas CS+ Project for CA78K0R/CA78K0
  - Renesas CS+ Project for CC-RX and CC-RL

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



Import

Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive 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.

10. CDC Driver Installation

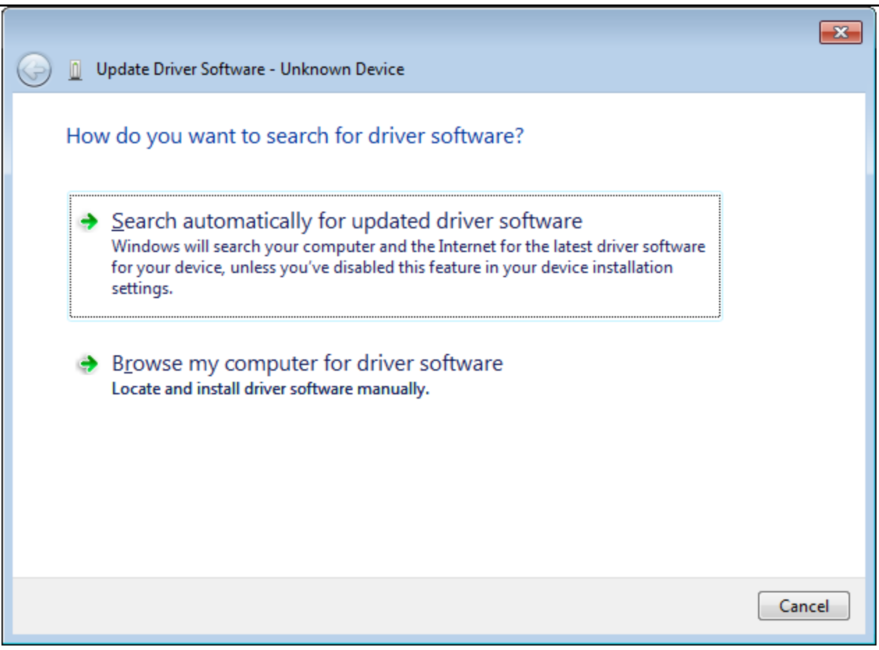
If USB Host is PC, CDC driver must be installed in the PC. When you connect RSK which this PCDC sample program is written in to your PC, the wizard shown in Figure 10-1 will appear on your screen and prompt the CDC driver installation

(6). Select **Update Driver Software** from the device manager.

(7). Select "**Browse my computer for driver software**".

Note:

The following works are unnecessary when using Windows 10 PC.



Update Driver Software - Unknown Device

How do you want to search for driver software?

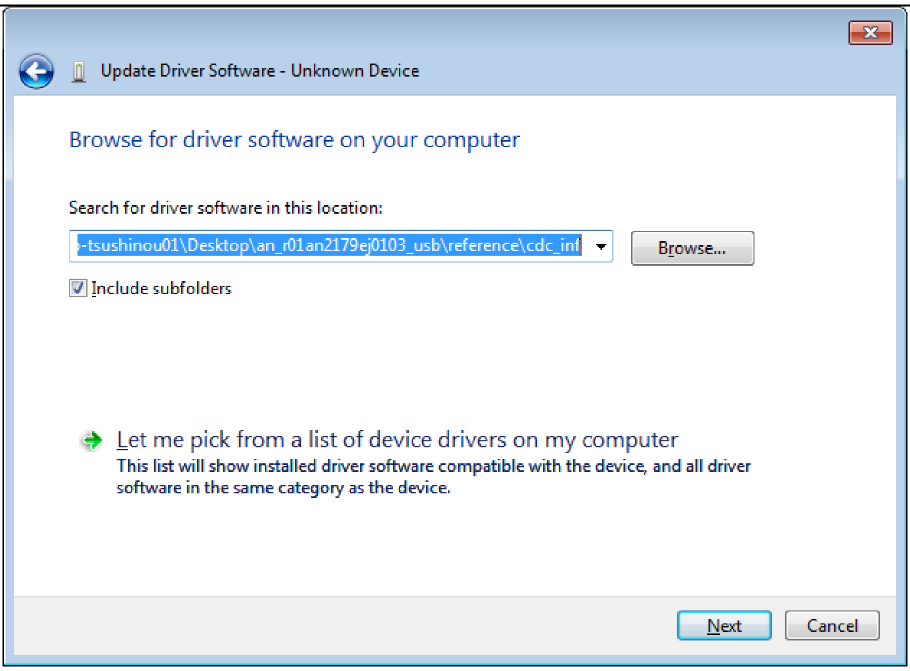
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

Figure 10-1 New Hardware Search Wizard

(8). Select **“Browse for driver software on your computer"**

Click **Browse**, specify the folder in which the *CDC_Demo.inf* is stored, then click **“Next”**



Update Driver Software - Unknown Device

Browse for driver software on your computer

Search for driver software in this location:

-tsushinou01\Desktop\an_r01an2179ej0103_usb\reference\cdc_inf

☑ Include subfolders

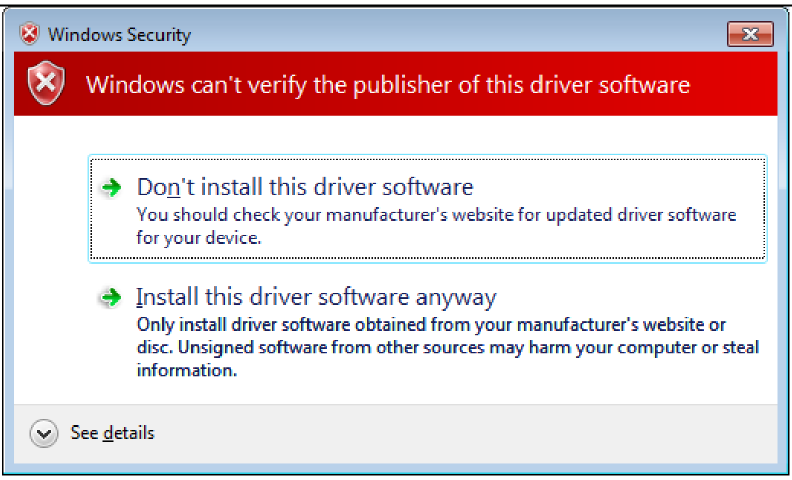
Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Figure 10-2 Select Driver Location

Note:

The *CDC_Demo.inf* file is stored in "r_usb_pcdc\utilities" folder in the package.

(9). If the following installation confirmation screen appears, click **“Browse for driver software on your computer”**



Windows Security

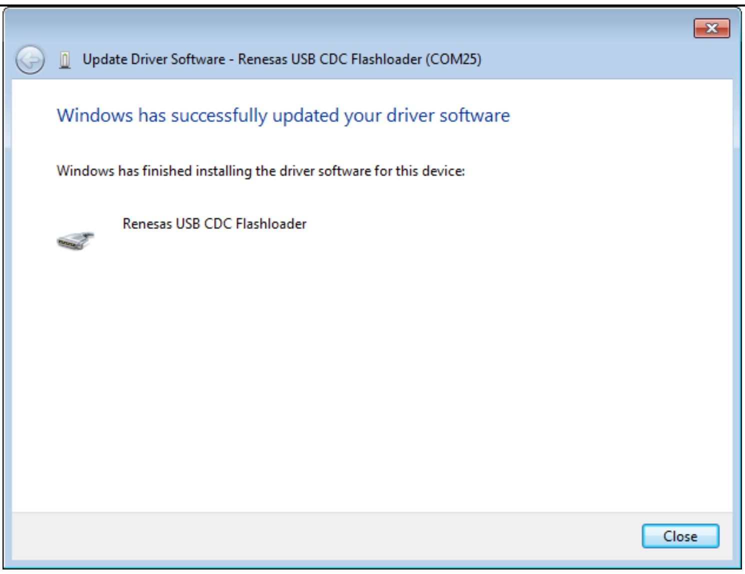
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

Figure 10-3 Installation Confirmation Screen

(10). When the following window appears, the CDC driver has been successfully installed. Click **“Close.”**



Update Driver Software - Renesas USB CDC Flashloader (COM25)

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Renesas USB CDC Flashloader

Figure 10-4 Installation Complete

※ An error may occur when installing the driver in the Windows 8.1 environment. In this case the installation confirmation screen will not appear.

11. Creating an Application

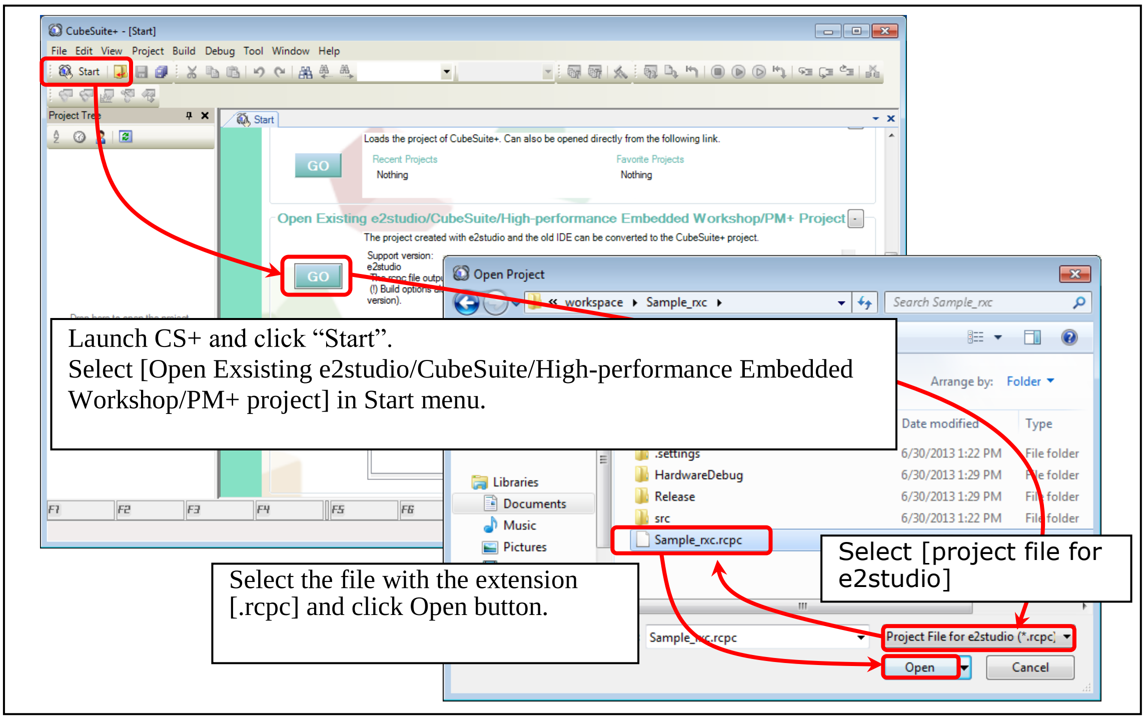
Refer to the chapter “**Creating an Application Program**” in the document (Document number: R01AN0512) for *USB Basic Host and Peripheral Driver Application Note*.

12. Using the e^2 studio project with CS+

The PCDC contains a project only for e^2 studio. When you use the PCDC 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.



Loads the project of CubeSuite+. Can also be opened directly from the following link.

Open Existing e2studio/CubeSuite/High-performance Embedded Workshop/PM+ Project

The project created with e2studio and the old IDE can be converted to the CubeSuite+ project.

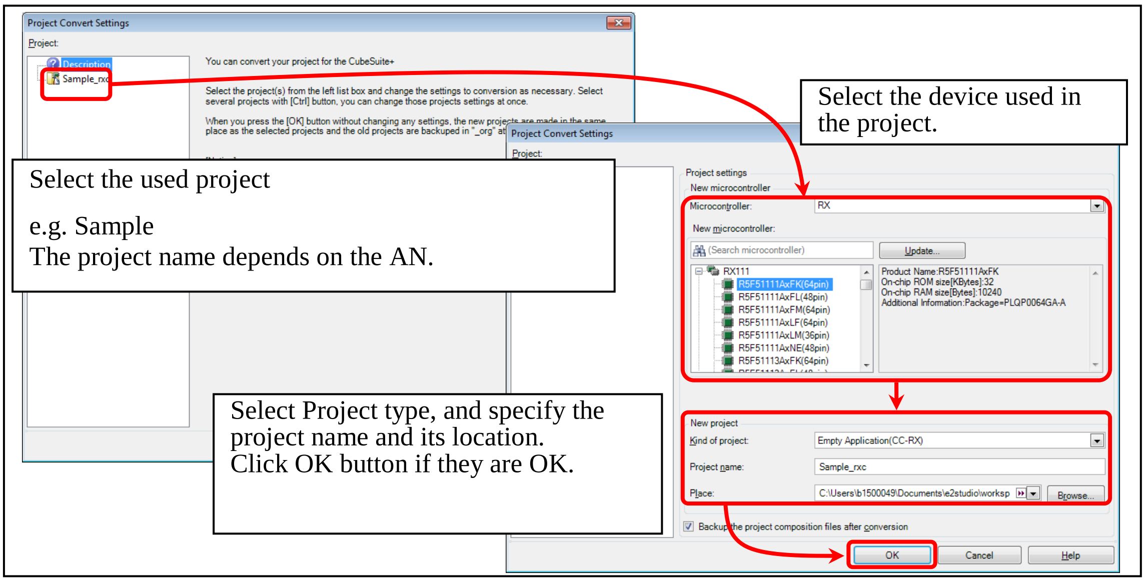
.settings	6/30/2013 1:22 PM	File folder
HardwareDebug	6/30/2013 1:29 PM	File folder
Release	6/30/2013 1:29 PM	File folder
src	6/30/2013 1:22 PM	File folder
Sample_rxc.rcpc		

File name: Sample_rxc.rcpc — Project File for e2studio (*.rcpc)

Launch CS+ and click “Start”.
Select [Open Exsisting e2studio/CubeSuite/High-performance Embedded Workshop/PM+ project] in Start menu.

Select the file with the extension [.rcpc] and click Open button.

Select [project file for e2studio]



You can convert your project for the CubeSuite+

Select the project(s) from the left list box and change the settings to conversion as necessary. Select several projects with [Ctrl] button, you can change those projects settings at once.

When you press the [OK] button without changing any settings, the new projects are made in the same place as the selected projects and the old projects are backuped in "_org" at

Microcontroller: RX

New microcontroller: (Search microcontroller)

RX111
- R5F51111AxFK(64pin)
- R5F51111AxFL(48pin)
- R5F51111AxFM(64pin)
- R5F51111AxLF(64pin)
- R5F51111AxLM(36pin)
- R5F51111AxNE(48pin)
- R5F51113AxFK(64pin)

Product Name:R5F51111AxFK
On-chip ROM size[KBytes]:32
On-chip RAM size[Bytes]:10240
Additional Information:Package=PLQP0064GA-A

New project
Kind of project: Empty Application(CC-RX)
Project name: Sample_rxc
Place: C:\Users\b1500049\Documents\e2studio\worksp

☑ Backup the project composition files after conversion

Select the device used in the project.

Select the used project

e.g. Sample
The project name depends on the AN.

Select Project type, and specify the project name and its location.
Click OK button if they are OK.

Figure 12-1 Using the e^2 studio project with CS+

Website and Support

Renesas Electronics Website

http://www.renesas.com/

Inquiries

http://www.renesas.com/inquiry

All trademarks and registered trademarks are the property of their respective owners.

Revision Record

Rev.	Date	Description Page	Summary
1.00	Mar.09.11	—	First edition issued
1.10	Jun.10.11	Cover, Other	Add RX630 Group and R8A66597 to Target Device (The content according to the Target device addition is changed.)
		4	Deleted 2.2 Serial port driver file in the file list
		30	5. Peripheral CDC Sample Application (APL) Modified APL specification - unused SW1 - SW description is changed to the function switch description.
		52	6. Communication Port Driver (CPD) Modified function name and constant definition "R_usb_ComPort" "usb_ComPort" -> "usbc_cpu_Sci" "USBC_PCDC_" -> "USB_SCI"
2.00	Mar.23.12	—	First edition issued for V.2.00
2.01	Feb.01.13	—	Description error and the reference error is fixed.
2.10	Apr.01.13	—	First Release for V.2.10 Add Target Device RX63T and R8A66593. Add the information on RX63T and R8A66593.
2.20	Sep.30.15	—	Change the application program. Change the folder structure. RX63N, RX631, R8A66597 and R8A66593 are deleted from Target Device.
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

© 2019 Renesas Electronics Corporation. All rights reserved.

Colophon 8.0