RX Family

R01AN2030EJ0142
Rev.1.42
Sep 29, 2023

USB Peripheral Communications Device Class Driver (PCDC) Firmware Integration Technology

Introduction

This application note describes USB Peripheral Communication Device Class Driver(PCDC), which utilizes Firmware Integration Technology (FIT). This module operates in combination with the USB Basic Host and Peripheral Driver (USB-BASIC-FW FIT module). It is referred to below as the USB PCDC FIT module.

Target Device

RX65N/RX651 Group
RX64M Group
RX71M Group
RX66T Group
RX72T Group
RX72M Group
RX66N Group
RX72N Group
RX671 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. RX64M Group User's Manual: Hardware (Document number. R01UH0377)
5. RX71M Group User’s Manual: Hardware (Document number. R01UH0493)
6. RX65N/RX651 Group User's Manual: Hardware (Document number. R01UH0590)
7. RX65N/RX651-2M Group User's Manual: Hardware (Document number. R01UH0659)
8. RX66T User's Manual: Hardware (Document number. R01UH0749)
9. RX72T User's Manual: Hardware (Document number. R01UH0803)
10. RX72M User's Manual: Hardware (Document number. R01UH0804)
11. RX66N User's Manual: Hardware (Document number. R01UH0825)
12. RX72N User's Manual: Hardware (Document number. R01UH0824)
13. RX671 User's Manual: Hardware (Document number. R01UH0899)
14. USB Basic Host and Peripheral Driver using Firmware Integration Technology Application Note (Document number. R01AN2025)

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

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

Contents

1. Overview 3
2. Software Configuration 4
3. API Information 5
4. CDC, PSTN, and ACM (Abstract Control Model) 9
5. API Functions 13
6. Configuration (r_usb_pcdc_config.h) 14
7. CDC Driver Installation 15
8. Creating an Application 18

1. Overview

The USB PCDC FIT module, when used in combination with the USB-BASIC-FW FIT module, 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: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology 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

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

1.3 Limitation

This driver does not support the composite device (PCDC+PCDC) in Azure RTOS version.

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
H/W	:	Renesas USB device
IDE	:	Integrated Development Environment
Non-OS	:	USB Driver for OS-less
PCD	:	Peripheral Control Driver for USB-BASIC-FW
PCDC	:	Peripheral Communications Devices Class
PCDCD	:	Peripheral Communications Devices Class Driver
PSTN	:	Public Switched Telephone Network, contains the ACM (above) standard.
RTOS	:	USB Driver for the real-time OS
SCI	:	Serial Communication Interface
USB-BASIC-FW	:	USB Basic Host and Peripheral Driver

1.5 USB PCDC FIT Module

User needs to integrate this module to the project using r_usb_basic. User can control USB H/W by using this module API after integrating to the project.

2. Software Configuration

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

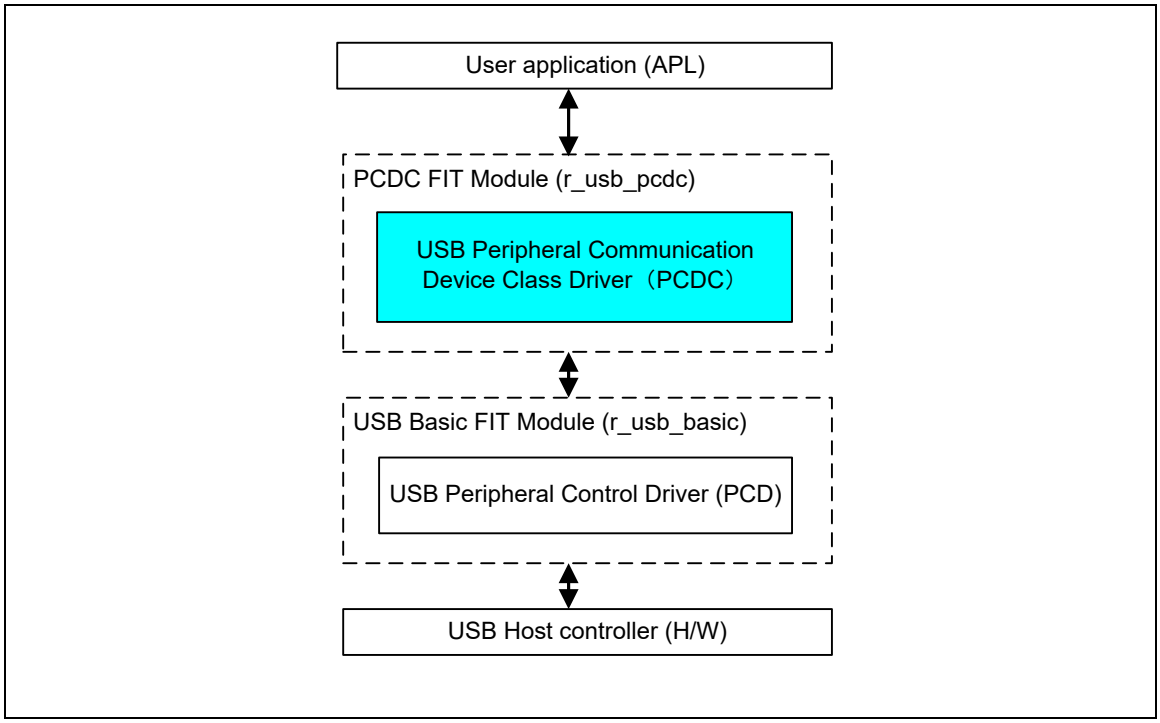


Figure 2-1 Software Module Structure

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. (Basic USB FW.)
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 Software Requirements

This driver is dependent upon the following packages:

- r_bsp
- r_usb_basic

3.3 Operating Confirmation Environment

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

Table 3-1 Operating Confirmation Environment

Item	Contents
C compiler	Renesas Electronics C/C++ compiler for RX Family V.3.03.00 (The option "-lang=C99" is added to the default setting of IDE)
	GCC for Renesas RX 4.08.04.201902 (The option "-std=gnu99" is added to the default setting of IDE)
	IAR C/C++ Compiler for Renesas RX version 4.10.01
Real-Time OS	FreeRTOS V.10.0.0 RI600V4 Azure RTOS (USBX) 6.1.11
Endian	Little Endian, Big Endian
USB Driver Revision Number	Rev.1.42
Using Board	Renesas Starter Kits for RX64M Renesas Starter Kits for RX71M Renesas Starter Kits for RX65N, Renesas Starter Kits for RX65N-2MB Renesas Starter Kits for RX72T Renesas Starter Kits for RX72M Renesas Starter Kits for RX72N Renesas Starter Kits for RX671
Host Environment	The operation of this USB Driver module connected to the following OSes has been confirmed. 1. Windows® 8.1 2. Windows® 10

3.4 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
RX64M RX71M	USBI0 Interrupt (Vector number: 189, Interrupt source number : 62, Software Configurable Interrupt B) USB D0FIFO0 Interrupt (Vector number: 34) / USB D1FIFO0 Interrupt (Vector number: 35) USBR0 Interrupt (Vector number:90)
	USBAR Interrupt (Vector number: 94) USB D0FIFO2 Interrupt (Vector number: 32) / USB D1FIFO2 Interrupt (Vector number: 33)
RX65N RX651 RX72M RX66N RX72N	USBI0 Interrupt (Vector number: 185, Interrupt source number : 62, Software Configurable Interrupt B) USB D0FIFO0 Interrupt (Vector number: 34) / USB D1FIFO0 Interrupt (Vector number: 35) USBR0 Interrupt (Vector number:90)
RX66T RX72T	USBI0 Interrupt (Vector number: 174) / USBR0 Interrupt (Vector number: 90) USB D0FIFO0 Interrupt (Vector number: 34) / USB D1FIFO0 Interrupt (Vector number: 35)
RX671	USBI0 Interrupt (Vector number: 185, Interrupt source number : 62, Software Configurable Interrupt B) USB D0FIFO0 Interrupt (Vector number: 34) / USB D1FIFO0 Interrupt (Vector number: 35) USBR0 Interrupt (Vector number:90)
	USBI1 Interrupt (Vector number: 182, Interrupt source number : 63, Software Configurable Interrupt B) USB D0FIFO1 Interrupt (Vector number: 36) / USB D1FIFO1 Interrupt (Vector number: 37)

3.5 Header Files

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

3.6 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*.

3.7 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: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology Application Note.*

3.8 ROM / RAM Size

The follows show ROM/RAM size of this driver.

1. CC-RX (Optimization Level: Default)

(1). Non-OS

	Checks arguments	Does not check arguments
ROM size	22.0K bytes (Note 3)	21.6K bytes (Note 4)
RAM size	5.5K bytes	5.5K bytes

(2). RTOS

a. FreeRTOS

	Checks arguments	Does not check arguments

ROM size	35.4K bytes (Note 3)	34.0K bytes (Note 4)
RAM size	21.0K bytes	21.0K bytes

b. RI600V4

	Checks arguments	Does not check arguments
ROM size	38.6K bytes (Note 3)	38.2K bytes (Note 4)
RAM size	11.4K bytes	11.4K bytes

c. Azure RTOS

ROM size	51.2K bytes
RAM size	17.4K bytes

2. GCC (Optimization Level: -O2)

(1). Non-OS

	Checks arguments	Does not check arguments
ROM size	27.1K bytes (Note 3)	26.6K bytes (Note 4)
RAM size	5.3K bytes	5.3K bytes

(2). Azure RTOS

ROM size	59.7K bytes
RAM size	16.7K bytes

3. IAR (Optimization Level: Medium

(1). Non-OS

	Checks arguments	Does not check arguments
ROM size	21.4K bytes (Note 3)	20.9K bytes (Note 4)
RAM size	4.1K bytes	4.1K bytes

(2). Azure RTOS

ROM size	41.0K bytes
RAM size	11.4K bytes

Note:

1. ROM/RAM size for BSP and USB Basic Driver is included in the above size.
2. The above is the size when specifying RX V2 core 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.
5. The result of RTOS includes the ROM/RAM size of the real-time OS.
6. The result of Azure RTOS includes the ROM/RAM size of USBX.

3.9 Argument

For the structure used in the argument of API function, refer to chapter "**Structures**" in the document (Document number: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology Application Note*.

3.10 “for”, “while” and “do while” arguments

In FIT module, when using “for”, “while” and “do while” statements (loop processing) in register reflection waiting processing, etc., write comments with “WAIT_LOOP” as a keyword for these loop processing. Also, write in the FIT documentation that “WAIT_LOOP” is written as a comment in these loop processes.

3.11 Adding the FIT Module to Your Project

This module must be added to each project in which it is used. Renesas recommends the method using the Smart Configurator described in (1) or (3) below. However, the Smart Configurator only supports some RX devices. Please use the methods of (2) or (4) for RX devices that are not supported by the Smart Configurator.

(1) Adding the FIT module to your project using “Smart Configurator” on e^2 studio

By using the Smart Configurator in e^2 studio, the FIT module is automatically added to your project. Refer to “Renesas e^2 studio Smart Configurator User Guide (R20AN0451)” for details.

(2) Adding the FIT module to your project using the FIT Configurator in e^2 studio

By using the FIT Configurator in e^2 studio, the FIT module is automatically added to your project. Refer to “Adding Firmware Integration Technology Modules to Projects (R01AN1723)” for details.

(3) Adding the FIT module to your project using the Smart Configurator in CS+

By using the Smart Configurator Standalone version in CS+, the FIT module is automatically added to your project. Refer to “Renesas e^2 studio Smart Configurator User Guide (R20AN0451)” for details.

(4) Adding the FIT module to your project on CS+

In CS+, please manually add the FIT module to your project. Refer to “Adding Firmware Integration Technology Modules to CS+ Projects (R01AN1826)” for details.

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.4.2 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: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology 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.
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.

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.

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: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology 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: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology Application Note*.

1. USB Pipe Setting for Bulk IN/OUT transfer

Set the pipe number (PIPE1 to PIPE5) to use for Bulk IN/OUT transfer.

#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)
#define	USB_CFG_PCDC_BULK_IN2	Pipe number (USB_PIPE1 to USB_PIPE5)
#define	USB_CFG_PCDC_BULK_OUT2	Pipe number (USB_PIPE1 to USB_PIPE5)

Note:

a. Do not set the same pipe number to the above each macro.

b. If the user system does not support, it is not necessary to specify the USB pipe number to *USB_CFG_PCDC_BULK_IN2* and *USB_CFG_PCDC_BULK_OUT2* macros.

2. USB Pipe Setting for 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)
#define	USB_CFG_PCDC_INT_IN2	Pipe number (USB_PIPE6 to USB_PIPE9)

Note:

a. Do not set the same pipe number to the above each macro.

b. If the user system does not support, it is not necessary to specify the USB pipe number to *USB_CFG_PCDC_INT_IN2* macro.

7. CDC Driver Installation

If USB Host is PC(Windows®), a CDC driver must be installed in the PC. The wizard shown in Figure 7-1 will appear on your screen and prompt the CDC driver installation when you connect RSK.

(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.

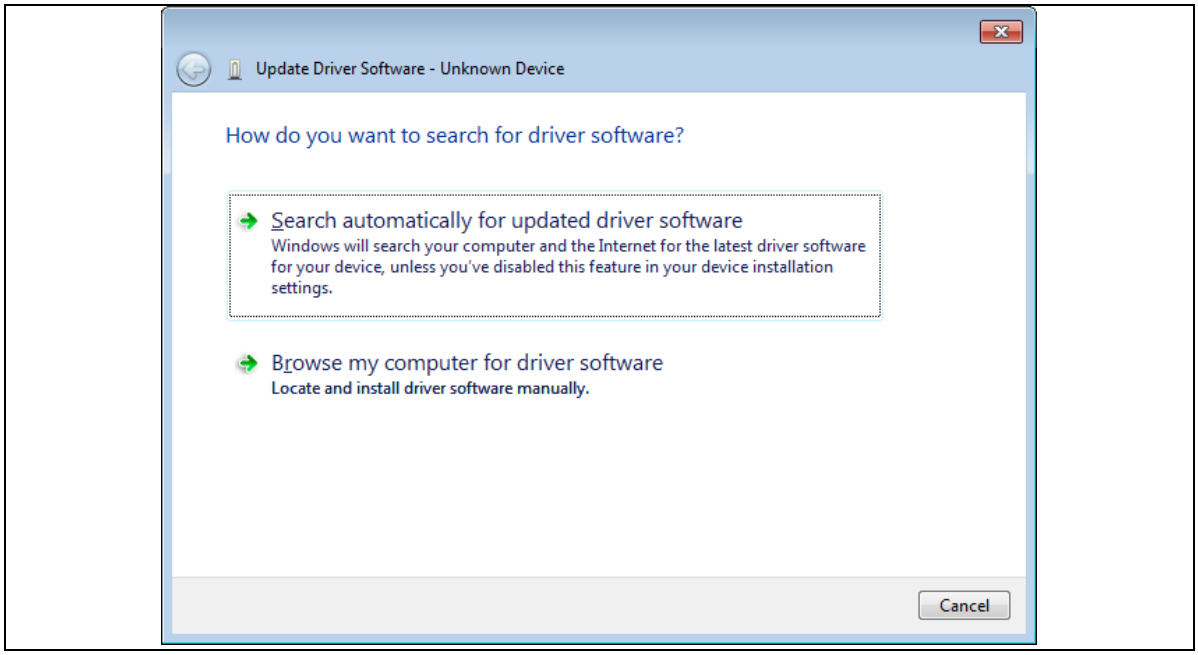


Figure 7-1 New Hardware Search Wizard

(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”**

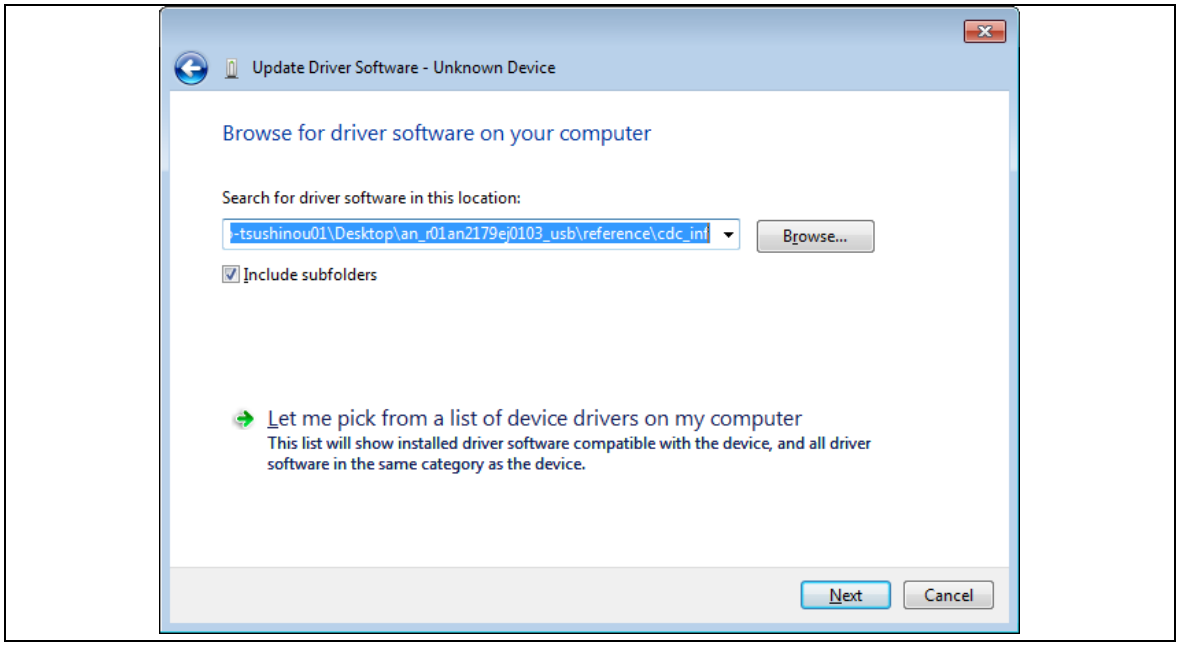


Figure 7-2 Select Driver Location

Note:

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

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

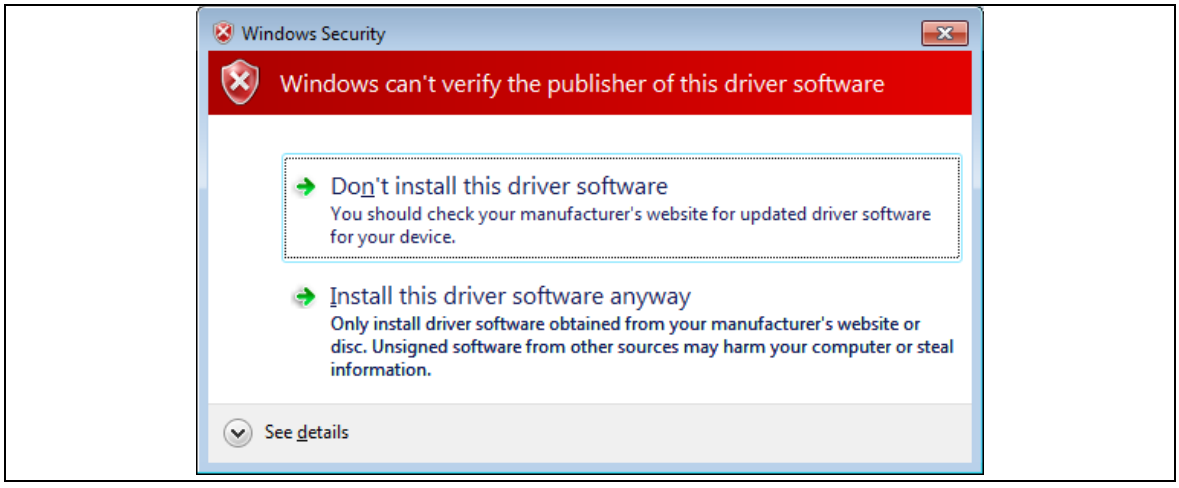


Figure 7-3 Installation Confirmation Screen

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

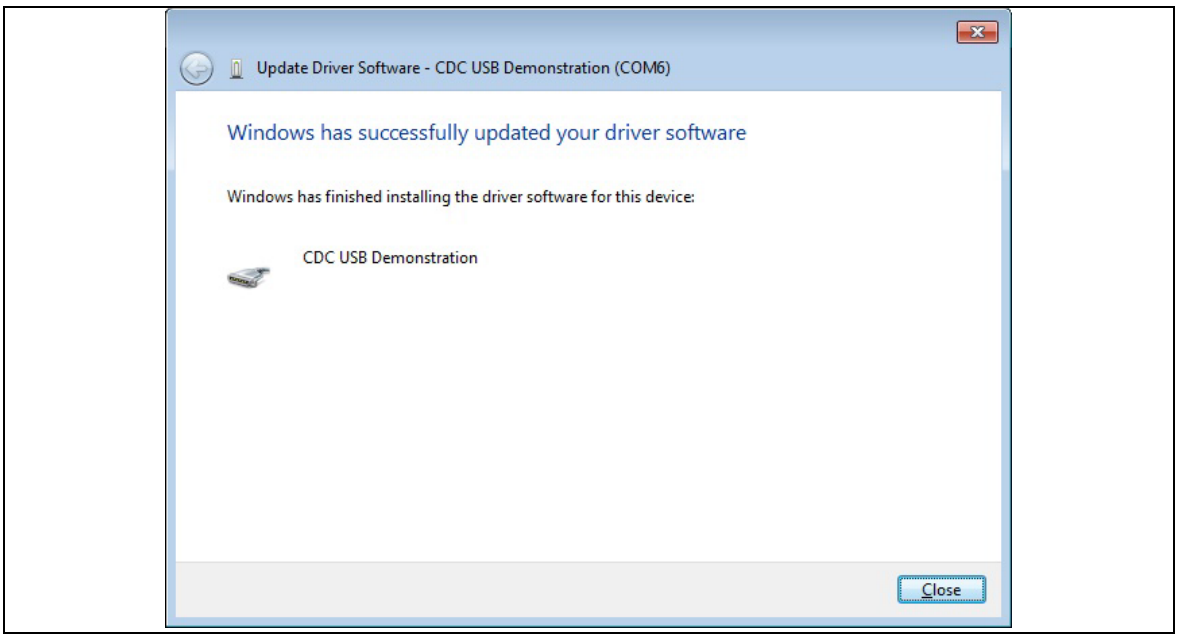


Figure 7-4 Installation Complete

8. Creating an Application

Refer to the chapter "**Creating an Application Program**" in the document (Document number: R01AN2025) for *USB Basic Host and Peripheral Driver using Firmware Integration Technology Application Note*.

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	Aug 1, 2014	—	First edition issued
1.10	Dec 26, 2014	—	RX71M is supported newly.
1.11	Sep 30, 2015	—	RX63N and RX631 are added in Target Device.
1.20	Sep 30, 2016	—	1. RX65N and RX651 are added in Target Device. 2. Supporting DMA transfer. 3. Supporting USB Host and Peripheral Interface Driver application note(Document No.R01AN3293EJ)
1.21	Mar 31, 2017	—	1. Supported Technical Update (Document number. TN-RX*-A172A/E) 2. The chapter *API Functions* is moved to the document (Document number: R01AN2025) of *USB Basic Host and Peripheral Driver Firmware Integration Technology*.
1.22	Sep 30, 2017	—	Supporting RX65N/RX651-2M
1.23	Mar 31, 2018	—	Supporting the Smart Configurator.
1.24	Dec 28, 2018	—	Supporting RTOS.
1.25	Apr 16, 2019	—	Added RX66T/RX72T in Target Device.
1.26	May 31, 2019	—	1. Support GCC compiler and IAR compiler. 2. Remove RX63N from Target Device.
1.27	Jul 31, 2019	—	RX72M is added in Target Device.
1.30	Mar 1, 2020	—	1. Supported the real time OS (uITRON:RI600V4). 2. Added RX72N/RX66N in Target Device.
1.31	Mar 1, 2021	—	Added RX671 in Target Device.
1.40	Jun 30, 2022	—	Support Azure RTOS (USBX PCDC).
1.42	Sep 29, 2023	—	Added chapter 3.10

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.

© 2023 Renesas Electronics Corporation. All rights reserved.