

RX Family

R01AN0401EJ0233
Rev.2.33
Jul 31, 2019

USB Peripheral Human Interface Device Class Driver

Introduction

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

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. RX62N/RX621 Group User's Manual: Hardware (Document number .R01UH0033)
3. RX63N/RX631 Group User's Manual: Hardware (Document number .R01UH0041)
4. RX630 Group User's Manual: Hardware (Document number .R01UH0040)
5. RX63T Group User's Manual: Hardware (Document number .R01UH0238)
6. 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	3
2. Software Configuration	4
3. API Information	5
4. USB Peripheral Human Interface Devices Class Driver (PHID)	7
5. API Functions	9
6. Configuration (r_usb_phid_config.h)	10
7. Sample Application	11
8. Setup	18
9. Creating an Application	21
10. Using the e ² studio project with CS+	22

1. Overview

PHID, when used in combination with the USB-BASIC-F/W, operates as a USB peripheral human interface device class driver (PHID). The PHID conforms to the USB Human Interface Device class specifications (referred to here as HID) and implements communication with a HID host.

This module supports the following functions.

- Data transfer to and from a USB host
- Response to HID class requests
- Response to function references from the HID host
- Interrupt OUT transfer

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 (*r01an0401jj0232_usb.pdf*, *r01an0512jj0232_usb.pdf*) under the "reference_documents" folder when using RX62N/RX621/ RX63T/RX630.

1.3 Terms and Abbreviations

API	:	Application Program Interface
APL	:	Application program
HID	:	Human Interface Device class
PCD	:	Peripheral control driver of USB-BASIC-F/W
PDCD	:	Peripheral device class driver (device driver and USB class driver)
PHID	:	Peripheral Human Interface Devices
RSK	:	Renesas Starter Kit
USB	:	Universal Serial Bus
	:	USB Basic Peripheral Driver

2. Software Configuration

Figure 2-1 shows the configuration of the modules related to PHID

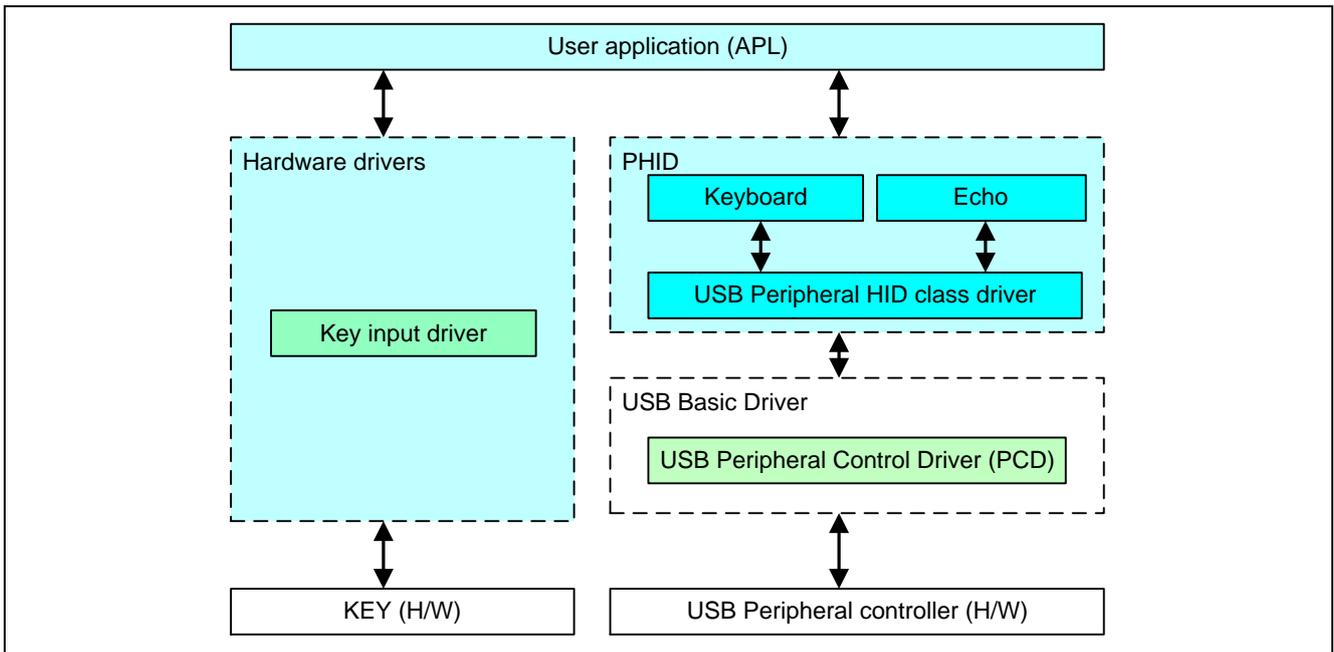


Figure 2-1 Source Code Block Diagram

Table 2.1 Modules

Module	Description
PHID	User switch operation on the RSK board is converted into HID reports. The transfer result is notified to APL by the callback function. In addition, communicate the output report of HID host to APL.
PCD	Peripheral Control Driver (USB Basic 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. <ol style="list-style-type: none"> 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_phid_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_phid_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	17.5K bytes (Note 3)	17.0K 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. USB Peripheral Human Interface Devices Class Driver (PHID)

4.1 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 HID class requests

Request	Code	Description
Get_Report	0x01	Receives a report from the HID host
Set_Report	0x09	Sends a report to the HID host
Get_Idle	0x02	Receives a duration (time) from the HID host
Set_Idle	0x0A	Sends a duration (time) to the HID host
Get_Protocol	0x03	Reads a protocol from the HID host
Set_Protocol	0x0B	Sends a protocol to the HID host
Get_Descriptor Descriptor Type : Class Class Descriptor Type : Report	0x06 (Standard)	Transmits a report descriptor
Get_Descriptor Descriptor Type : Class Class Descriptor Type : HID	0x06 (Standard)	Transmits an HID descriptor

4.2 Class Request Data Format

1. GetReport

Table 4-1 GetReport Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_REPORT (0x01)	ReportType & ReportID	Interface	ReportLength	Report

2. SetReport

Table 4-2 SetReport Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_REPORT (0x09)	ReportType & ReportID	Interface	ReportLength	Report

3. GetIdle

Table 4-3 GetIdle Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_IDLE (0x02)	0(Zero) & ReportID	Interface	1(one)	Idle rate

4. SetIdle

Table 4-4 SetIdle Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_IDLE (0x0A)	Duration & ReportID	Interface	0(zero)	Not applicable

5. GetProtocol

Table 4-5 GetProtocol Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0xA1	GET_PROTOCOL (0x03)	0(zero)	Interface	0(zero)	0 (Boot Protocol) / 1 (Report Protocol)

6. SetProtocol

Table 4-6 SetProtocol Format

bmRequestType	bRequest	wValue	wIndex	wLength	Data
0x21	SET_PROTOCOL (0x0B)	0 (Boot Protocol) / 1 (Report Protocol)	Interface	0(zero)	Not applicable

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_phid_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 (PIPE6 to PIPE9) to use for Interrupt IN/OUT transfer. Do not set the same pipe number for the definitions of `USB_CFG_PHID_INT_IN` and `USB_CFG_PHID_INT_OUT`.

```
#define    USB_CFG_PHID_INT_IN        Pipe number (USB_PIPE6 to USB_PIPE9)
#define    USB_CFG_PHID_INT_OUT      Pipe number (USB_PIPE6 to USB_PIPE9)
```

Note:

For a system that does not support the OUT transfer, set `USB_NULL` as the definition of `USB_CFG_PHID_INT_OUT`.

7. Sample Application

7.1 Application Specifications

The main functionalities of the PHID sample application (APL) are described below.

1. Keyboard Mode: Keyboard Functionality

When the RSK connects to the USB host, the USB host recognizes the RSK as a keyboard. The RSK operates as a keyboard, transmitting keyboard data to the USB host using interrupt IN transfer.

2. Echo Mode: USB Loopback Functionality (Interrupt IN/OUT Data Transfer)

The RSK connects to the USB host and performs interrupt IN/OUT data transfer. This functionality performs processing to transmit the data received from the USB host back to the USB host unaltered.

3. Low-Power-Consumption Functionality

This functionality transitions the microcontroller to a low-power mode according to the USB state.

- a) In the USB suspend state, the microcontroller transitions to sleep mode.
- b) In the USB detached state, the microcontroller transitions to software standby mode.

[Note]

1. Make the selection of Keyboard mode or Echo mode by setting in `r_usb_phid_apl_config.h`.
2. Make the selection(using or unusing) of the low power consumption functionality in `r_usb_phid_apl_config.h`.
3. Echo mode enables communication with USB hosts supporting USB loopback functionality. Keyboard mode enable USB communication with PCs (USB hosts) supporting OSes such as Windows 7, Windows 8.1 and Windows 10.

7.2 Application Processing

The APL consists of two parts: processing of initial settings and the main loop. These are described in outline below.

7.2.1 Initial Setting

The initial settings include microcontroller pin settings, USB driver settings, and USB controller initial settings.

7.2.2 Main Loop (Keyboard mode)

In keyboard mode, processing is performed to send information about the switches(buttons) on the RSK board to the USB host. When the RSK (HID device) is connected to the USB host (PC), the RSK is recognized as a keyboard, and the main loop consists of a main routine that sends the switch(button) information to the USB host as key input data. For details of the switch(button) information (key input data), refer to **7.4.1, Switch specification**.

1. When enumeration with the USB host completes, the USB host sends an HID class request to the HID device. After it receives the HID class request, the HID device calls the *R_USB_GetEvent* function and the return value is set to *USB_STS_REQUEST*. When the APL confirms *USB_STS_REQUEST*, it analyzes the received class request and performs processing corresponding to it.
2. When the *R_USB_GetEvent* function is called after processing of the class request mentioned in 1, above, completes, the return value is set to *USB_STS_REQUEST_COMPLETE*. The APL performs processing to make request information settings, etc.
3. The APL confirms whether a switch(button) on the RSK was depressed or not. If a switch(button) is depressed, the APL references the status management variable to confirm whether or not data transmission is in progress. If data transmission is not in progress, it calls the *R_USB_Write* function to send information on the depressed switch(button) as key information.
4. When the *R_USB_GetEvent* function is called after transmission of HID data as mentioned in 3, above, completes, the return value is set to *USB_STS_WRITE_COMPLETE*. When the APL confirms *USB_STS_WRITE_COMPLETE*, it calls the *R_USB_Write* function to request transmission of zero data (8 bytes). (In keyboard mode it is necessary to send zero data to inform the USB host that the key input has been released.
5. When the *R_USB_GetEvent* function is called after transmission of zero data as mentioned in 4, above, completes, the return value is set to *USB_STS_WRITE_COMPLETE*. When the APL confirms *USB_STS_WRITE_COMPLETE*, it checks whether or not transmission of zero data has completed, and if it is zero data, the APL makes a setting (NO_WRITING) to the status management variable.
6. If a suspend signal is received from the USB host or a detach event is confirmed while the processing in steps 1, to 5, above is repeating, the APL performs processing to transition the HID device (RSK) to low-power mode. For information on low-power mode, refer to **7.2.4, MCU Low power consumption processing**. Note that confirmation of reception of a suspend signal or a detach event involves reading the return value (*USB_STS_SUSPEND* or *USB_STS_DETACH*) of the *R_USB_GetEvent* function. An outline of the processing of the APL is shown below.

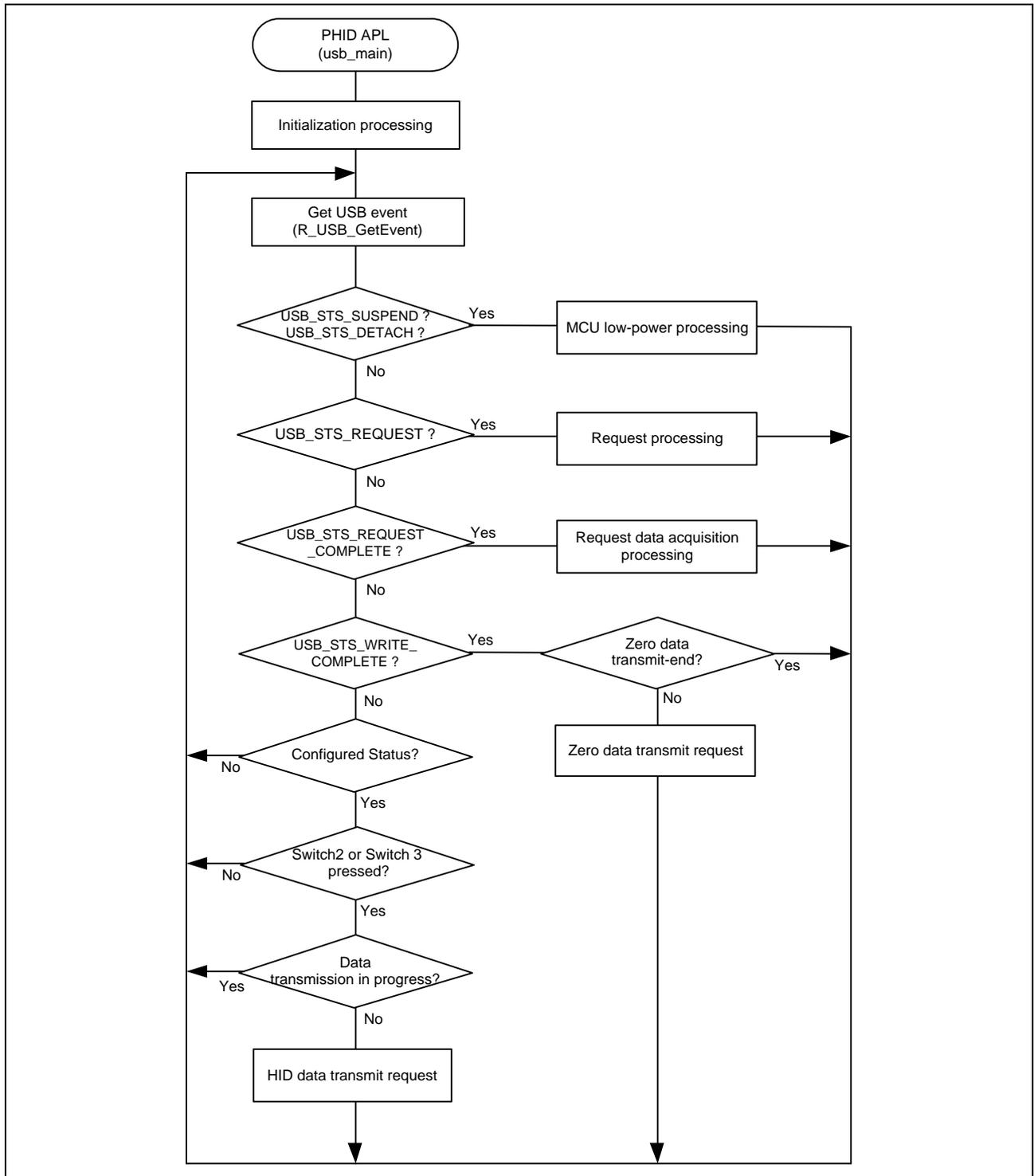


Figure 7-1 Main Loop Processing (Keyboard mode)

7.2.3 Main loop (Echo mode)

The echo mode main loop performs loop-back processing in which data received from the USB host is transmitted unaltered back to the USB host 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 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 transmit the received data to the USB host.
3. 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.
4. The processing in steps 2 and 3, above, is repeated.
5. 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 **7.2.4, 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.

An outline of the processing of the APL is shown below.

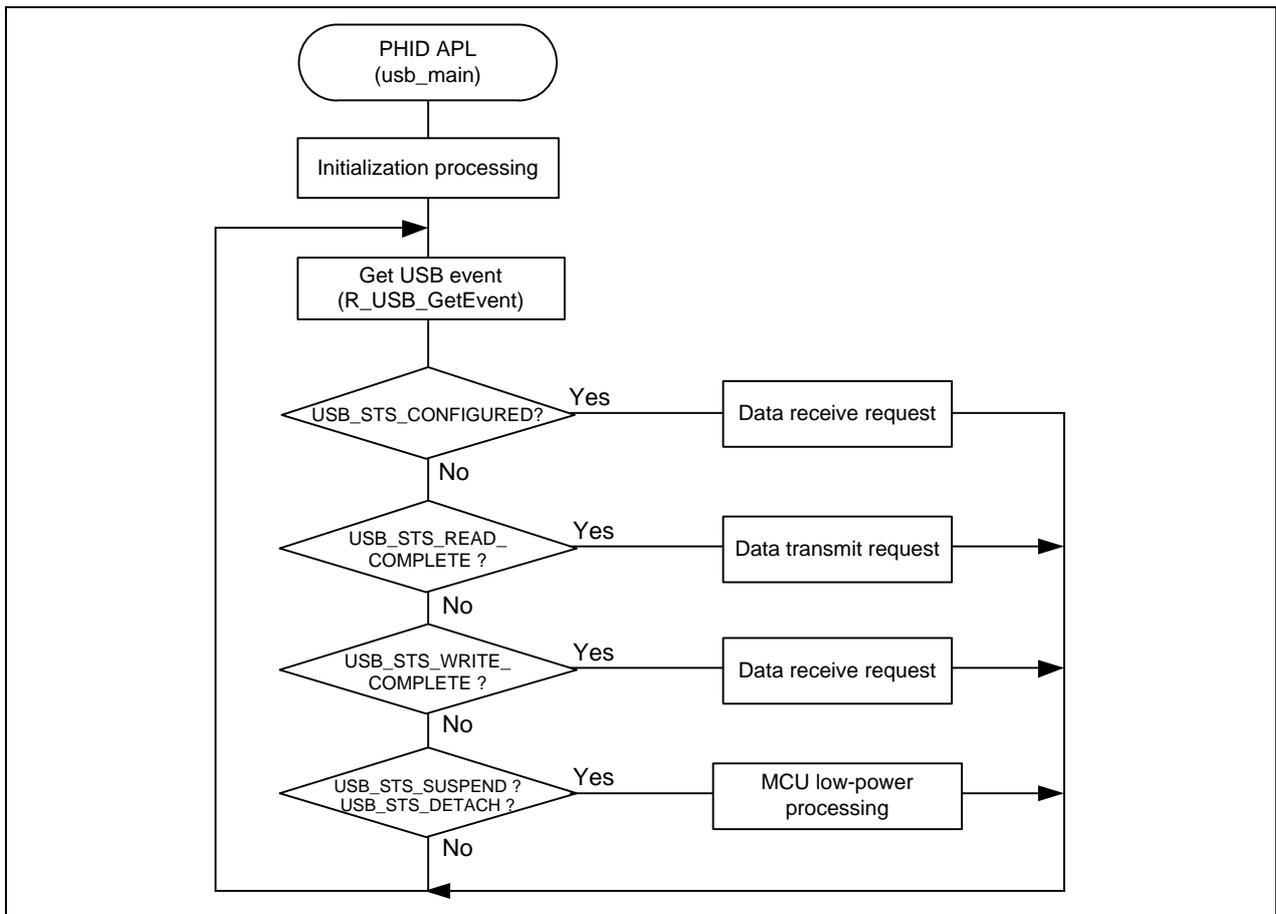


Figure 7-2 Main Loop Processing (Echo mode)

7.2.4 MCU Low power consumption processing

MCU low-power processing occurs when the conditions in Table 7-1 are met, causing a transition to low-power mode. Note that this processing is enabled by setting the USE_LPW definition in the r_usb_phid_apl_config.h file to USE_LPW.

Table 7-1 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 HID 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 HID device (RSK) attaches to the USB host.
2. When a suspend signal sent by the USB host is received while the HID 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.

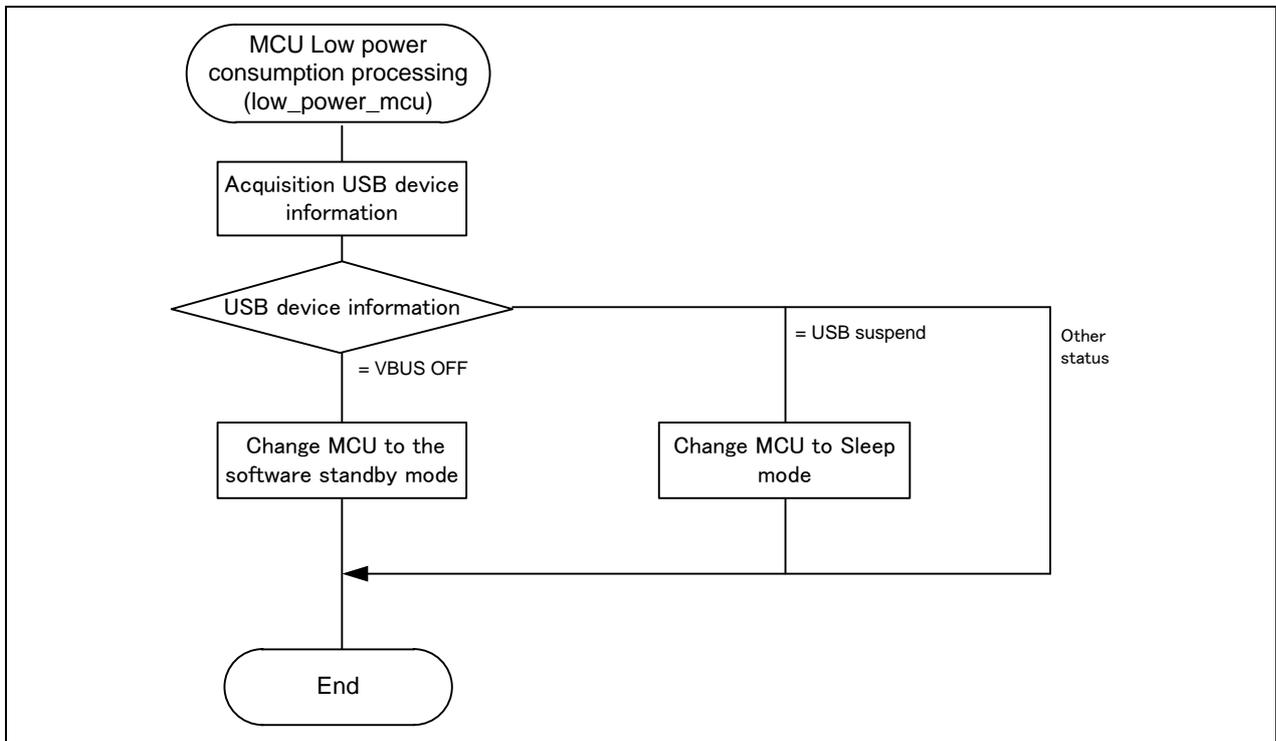


Figure 7-3 Flowchart of MCU Low Power Consumption Processing

7.3 Configuration File for the application program (r_usb_phid_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. OPERATION_MODE Definition

Specify one of the following settings for the OPERATION_MODE definition.

```
#define OPERATION_MODE HID_KEYBOARD // Keyboard mode
#define OPERATION_MODE HID_ECHO // Echo mode
```

3. 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 USE_SUPPORT_LPW USB_APL_ENABLE // Use the low-power function
```

4. 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).

7.4 Keyboard operation

In keyboard mode the switches(buttons) on the RSK board are used to make the RSK operate as an HID device. The switch(button) input information is used as keyboard key data.

7.4.1 Switch specification

The specifications of the switches used in keyboard mode are listed below. The switch input information is used as the key data of the keyboard.

Switch Number	Operation
Switch2(SW2)	One of the key codes for characters "a" to "z" or "Enter" is reported to the host each time SW is pressed.
Switch3(SW3)	One of the key codes for "1" to "9" and "0" or "Enter" is notified to the host each time SW is pressed.

7.4.2 Data Format

The table below shows the transmit report format used for sending data to and from USB Host. These data formats are set in conjunction with the HID report descriptor contents notified to USB Host.

Table 7-2 Data Formats Used when Notifying the Host

offset	Keyboard Mode (8Bytes)
0	Modifier keys
1	Reserved
2	Keycode 1
3	Keycode 2
4	Keycode 3
5	Keycode 4
6	Keycode 5
7	Keycode 6

7.5 Descriptor

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

8. Setup

8.1 Hardware

8.1.1 Example Operating Environment

Figure 8-1 shows an example operating environment for the PHID. Refer to the associated instruction manuals for details on setting up the evaluation board and using the emulator, etc.

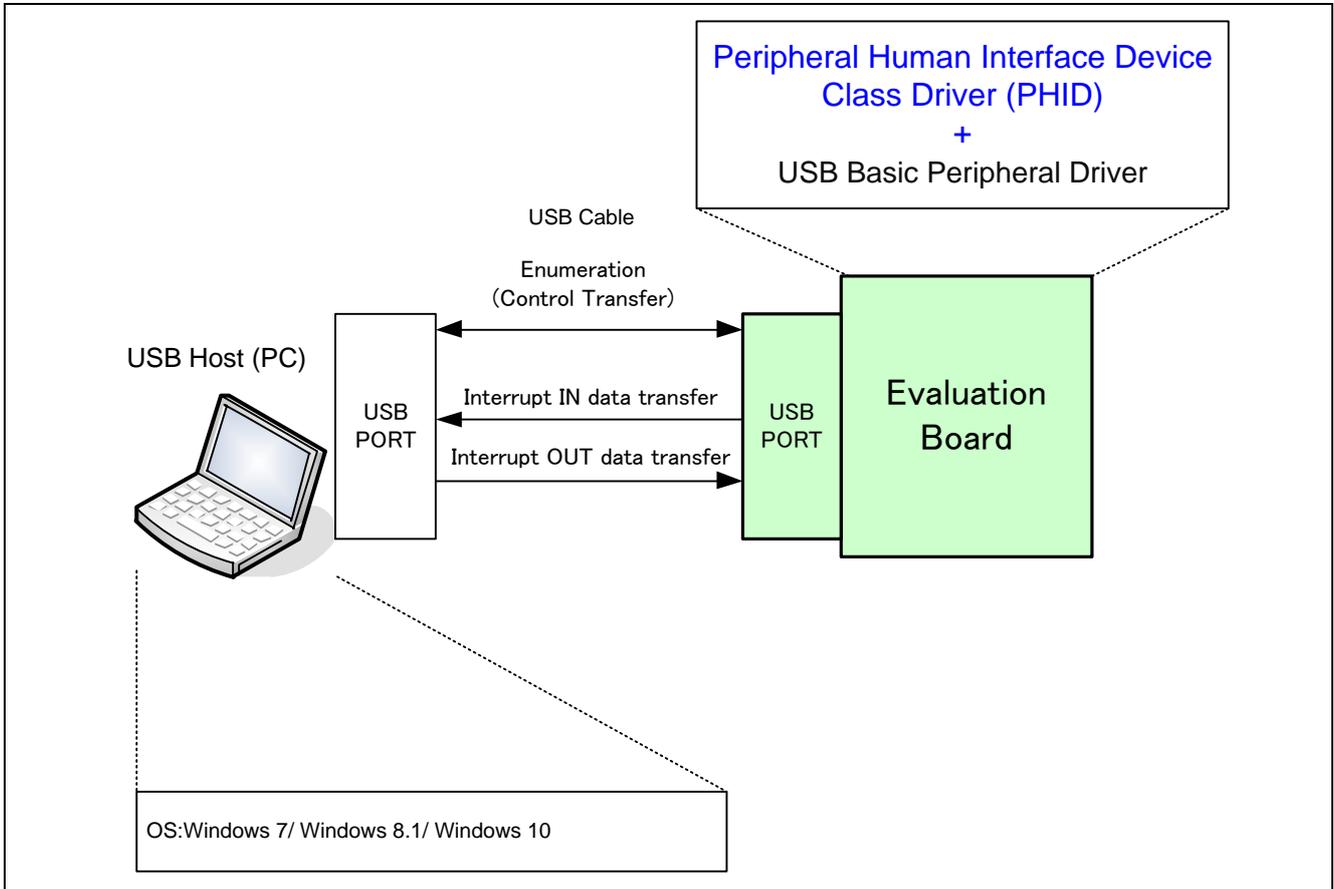


Figure 8-1 Example Operating Environment

8.1.2 RSK Setting

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

Table 8-1 **RSK Setting**

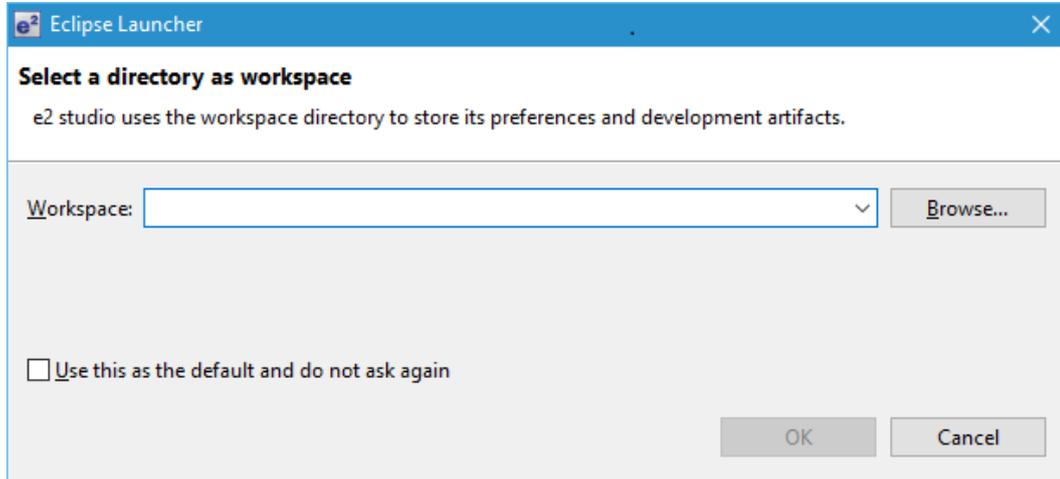
RSK	Jumper Setting
RSK+RX63N	J3: Shorted Pin1-2 J4: Shorted Pin1-2 J18: Shorted Pin2-3

Note:

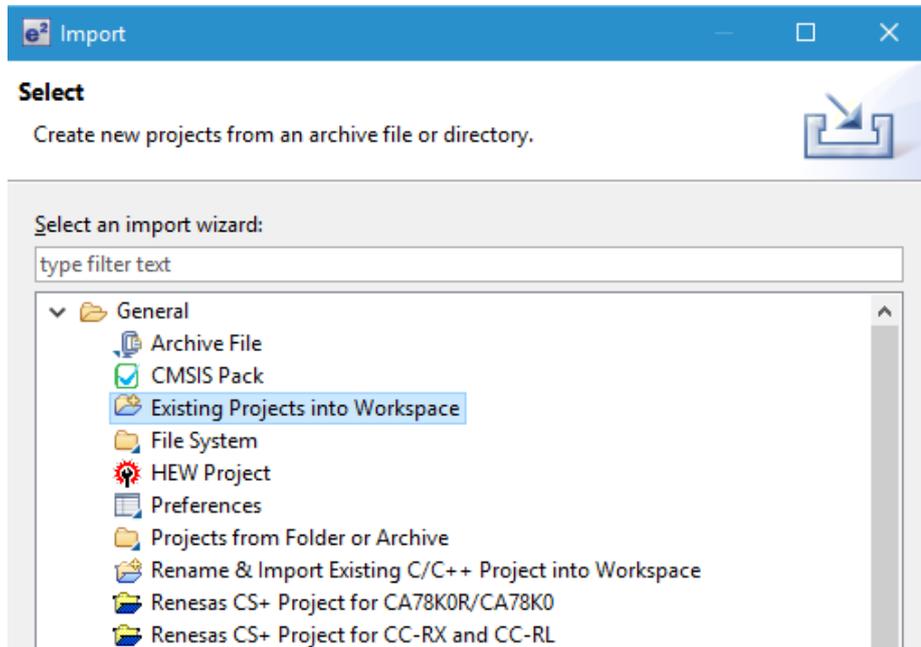
For the detail of RSK setting, refer to the user's manual of RSK.

8.2 Software

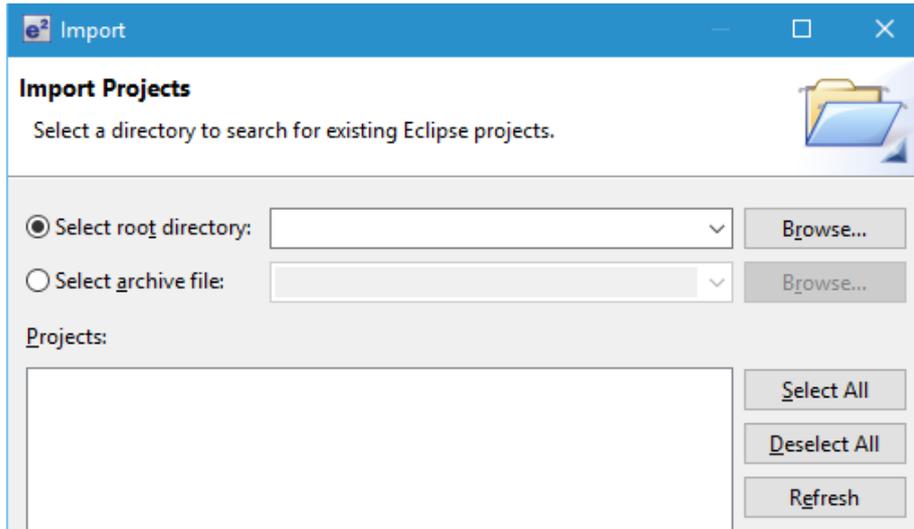
- 1) Setup e² studio
 - a) Start e² studio
 - b) If you start up e² studio at first, the following dialog is displayed. Specify the folder to store the project in this dialog.



- 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.

9. 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*.

10. Using the e² studio project with CS+

The PHID contains a project only for e² studio. When you use the PHID 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.

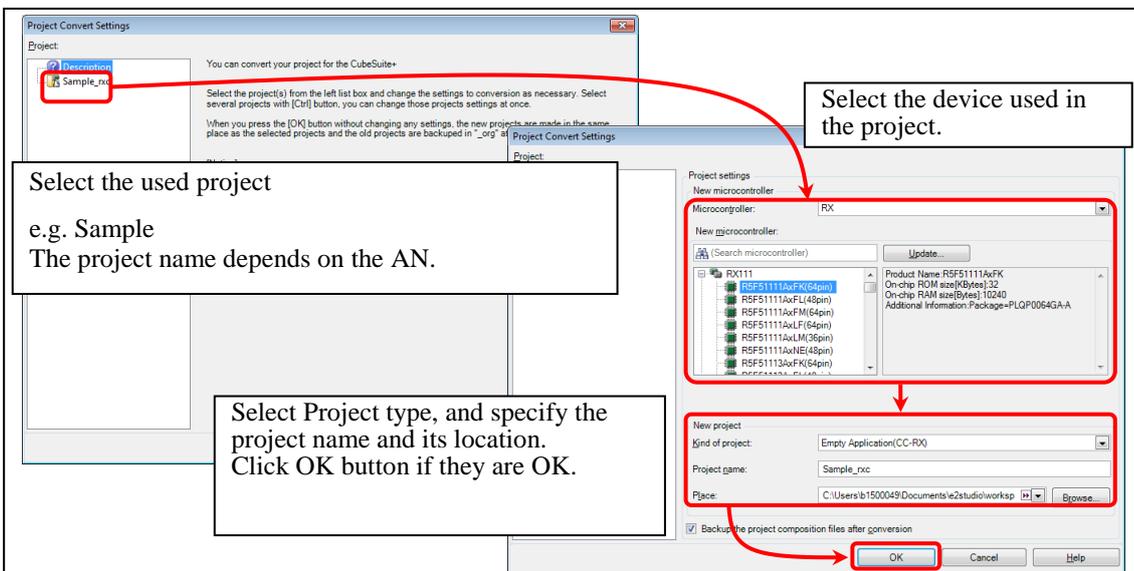
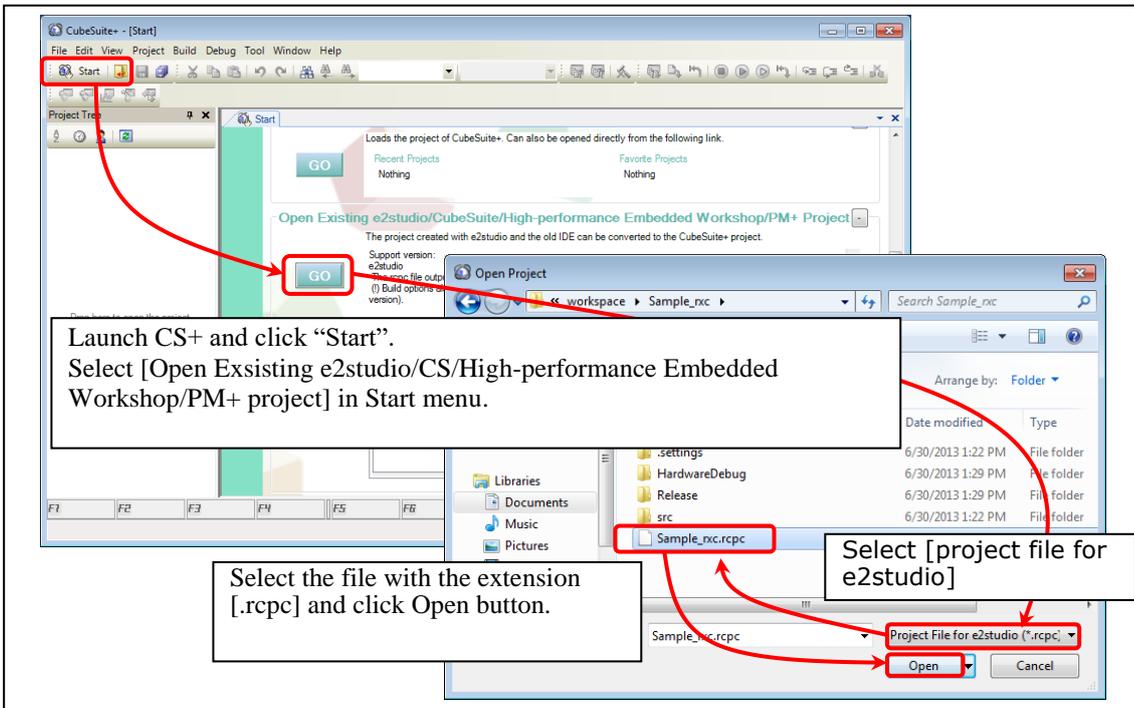


Figure 10-1 Using the e² studio project with CS+

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	Aug.10.11	—	Add Target Device RX630, R8A66597
		25	Add the information on RX630 and R8A66597 (Hi-Speed USB) 5.Peripheral HID Sample Application (APL) change to Application - Unused SW1 - SW description is changed to the function switch description
2.00	Sep.30.12	—	Revision of the document by firmware upgrade
2.10	Apr.1.13	—	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	—	Supporting USB Host and Peripheral Interface Driver application note(Document No.R01AN3293EJ)
2.31	Sep 30, 2017	—	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)



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