RX Family

R01AN2297EJ0132
Rev.1.32
May 15, 2026

Sample Program using USB Peripheral Human Interface Device Class Driver (PHID) for USB Mini Firmware to communicate via USB with USB Host Firmware Integration Technology

Introduction

This document describes the following sample firmware: USB Peripheral Human Interface Devices Class Driver for USB Mini Firmware using Firmware Integration Technology. The sample firmware is referred to below as the PHID.

When developing an actual software, be sure to use the “USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology Application Note” (Document number: R01AN2166) together with the user’s manual for each MCU (Hardware). In addition, also refer to the " USB Peripheral Human Interface Device Class Driver (PHID) for USB Mini Firmware using Firmware Integration Technology Application Note” (Document number: R01AN2171), if necessary. “USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology Application Note” (Document number: R01AN2166) is located in the "reference_documents" folder within the package.

Target Device

RX111 Group
RX113 Group
RX231 Group
RX23W Group
RX261 Group

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

Contents

1. **Introduction** 2
2. **Software Configuration** 4
3. **Setup** 5
4. **Sample Application** 8
5. **Class Request** 20
6. **Using RI600V4 project with CS+** 21
7. **Using the e^2 studio project with CS+** 25

1. Introduction

1.1 Functions

The PHID conforms to the USB human interface device (HID) class specification and implements communication with HID devices.

The PHID provides the following functionalities:

1. Operation as a Full-Speed (12 Mbps) device.
2. It is recognized as a HID device when connected to a USB host, and transfers data as a virtual mouse or virtual keyboard.

1.2 FIT Module Configuration

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

Table 1-1 FIT Module Configuration

FIT Module	Folder Name
Board Support Package Module Using Firmware Integration Technology	r_bsp
RX Family USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology	r_usb_basic_mini
RX Family USB Peripheral Human Interface Devices Class Driver(HID) for USB Mini Firmware using Firmware Integration Technology	r_usb_phid_mini
RX Family LPC (Low Power Consumption) Module Firmware Integration Technology	r_lpc_rx
RX Family IRQ Module Firmware Integration Technology	r_irq_rx

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

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

1.3 Note

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

1.4 Operating Confirmation Environment

The environment required for the PHID to operate is described below:

Table 1-2 Operation Confirmation Environment

Item	Contents
C compiler	Renesas Electronics C/C++ compiler for RX Family (The option "-lang=C99" is added to the default setting of IDE)
	GCC for Renesas RX (The option "-std=gnu99" is added to the default setting of IDE)
	IAR C/C++ Compiler for Renesas RX
Real-Time OS	FreeRTOS RI600V4
Endian	Little Endian, Big Endian
USB Driver Revision Number	Rev.1.32
Using Board	Renesas Starter Kit for RX111 Renesas Starter Kit for RX113 Renesas Starter Kit for RX231 Renesas Solution Starter Kit for RX23W EK-RX261
Host Environment	The operation of this USB Driver module connected to the following OSes has been confirmed. 1. Windows® 11

2. Software Configuration

2.1 Module Configuration

The PHID comprises a HID class driver as well as a device drivers. In response to data transfer requests from the APL, it transfers data to the USB host, via the PCD.

Figure 2-1 shows the module configuration of the PHID, and Table 2-1 lists the functions of the modules.

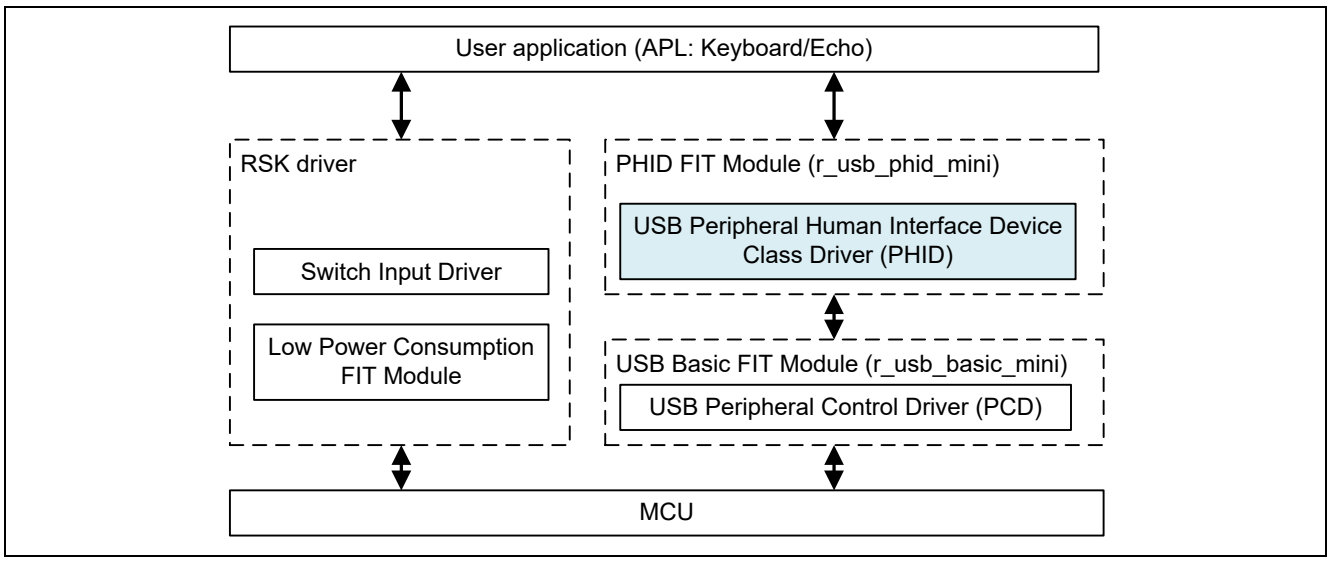


Figure 2-1 Module Configuration

Table 2-1 Function of Moudles

Module Name	Function
APL	Sample application program
RSK driver	Sample application for using the peripheral functions on the RSK board.
PHID (r_usb_phid_mini)	Interprets requests from the HID host. Reports APL key operation information to the HID host, via the PCD.
PCD (r_usb_basic_mini)	USB Peripheral Control Driver

3. Setup

3.1 Hardware

3.1.1 Example Operating Environment

Figure 3-1 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

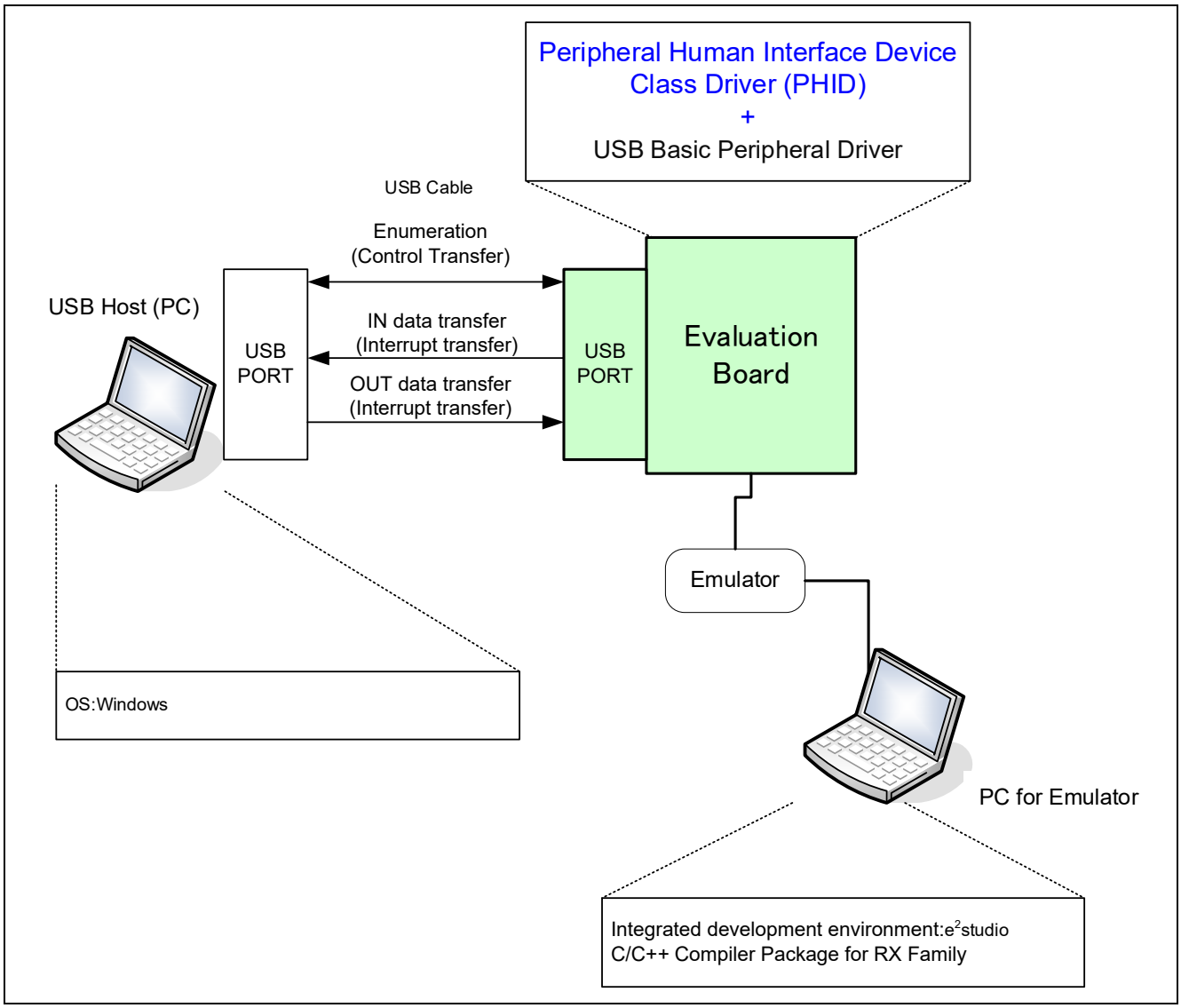


Figure 3-1 Example Operating Environment

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

Table 3-1 Evaluation Board on which PHID operation has been verified

MCU	Evaluation Board
RX111	RSKRX111
RX113	RSKRX113
RX231	RSKRX231
RX23W	RSSKRX23W
RX261	EK-RX261

3.1.2 RSK / RSSK / EK Setting

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

Table 3-2 **Jumper Setting**

RSK / RSSK	Jumper Setting
RSKRX111	J12: Shorted Pin2-3
RSKRX113	J12: Shorted Pin2-3
RSKRX231	J15: Shorted Pin2-3
RSSKRX23W	J5: Shorted Pin1-2
EK-RX261	J18: Shorted Pin2-3, J19: Shorted Pin2-3

Note:

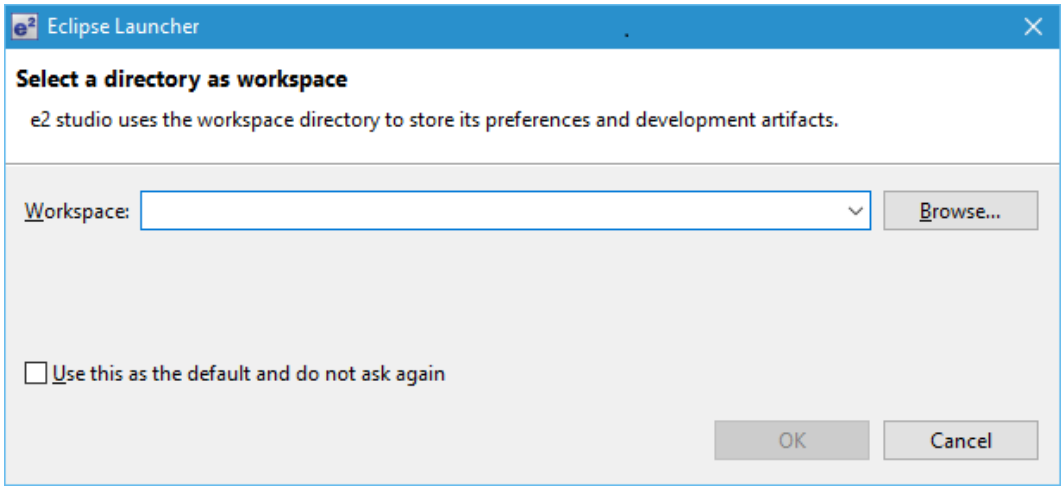
For the detail of RSK/RSSK setting, refer to the user's manual of RSK/RSSK.

3.2 Software

1) Setup e^2 studio

a) Start e^2 studio

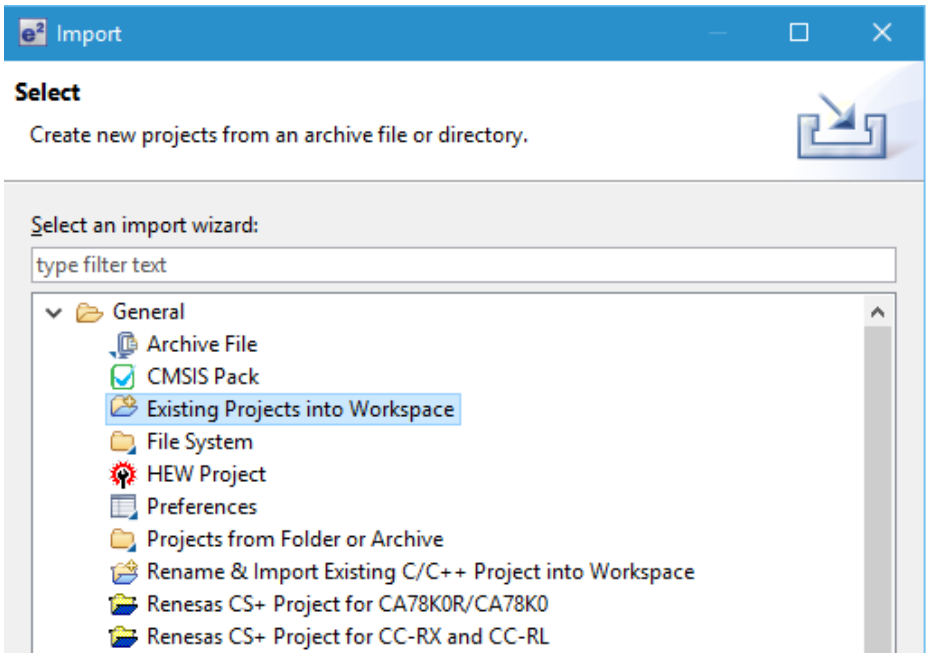
b) If you start up e^2 studio at first, the following dialog is displayed. Specify the folder to store the project in this dialog.



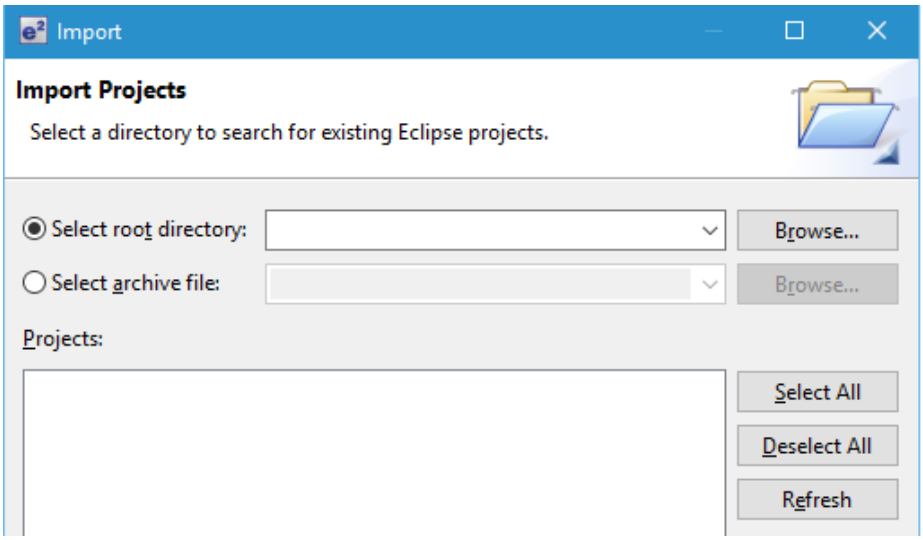
2) Import the project to the workspace

a) Select [File] > [Import]

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



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



d) Click "Finish".

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

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

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

4. Sample Application

4.1 Application Specifications

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

1. **Keyboard Mode: Keyboard Functionality**

   When the RSK/RSSK/EK is connected to the USB host, the USB host recognizes the RSK/RSSK/EK as a keyboard. The RSK/RSSK/EK functions as a keyboard, and keyboard data is sent to the USB host by the interrupt IN transfers.

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

   The RSK/RSSK/EK 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 10.

4.2 Overview of Application Processing (for Non-OS)

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

4.2.1 Initial setting

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

4.2.2 Main loop (Keyboard mode)

In keyboard mode, processing is performed to send information about the switchs(buttons) on the RSK/RSSK/EK board to the USB host. When the RSK/RSSK/EK (HID device) is connected to the USB host (PC), the RSK/RSSK/EK 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 **4.6.1, Switch specification**.

1. When enumeration with the USB host completes, the USB host sends a 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_REQUST*. 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_REQUST_COMPLETE*. The APL performs processing to make request information settings, etc.
3. The APL confirms whether a switch(button) on the RSK/RSSK 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*, is 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, APL sets *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/RSSK/EK) to low-power mode. For information on low-power mode, refer to **4.4, 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.

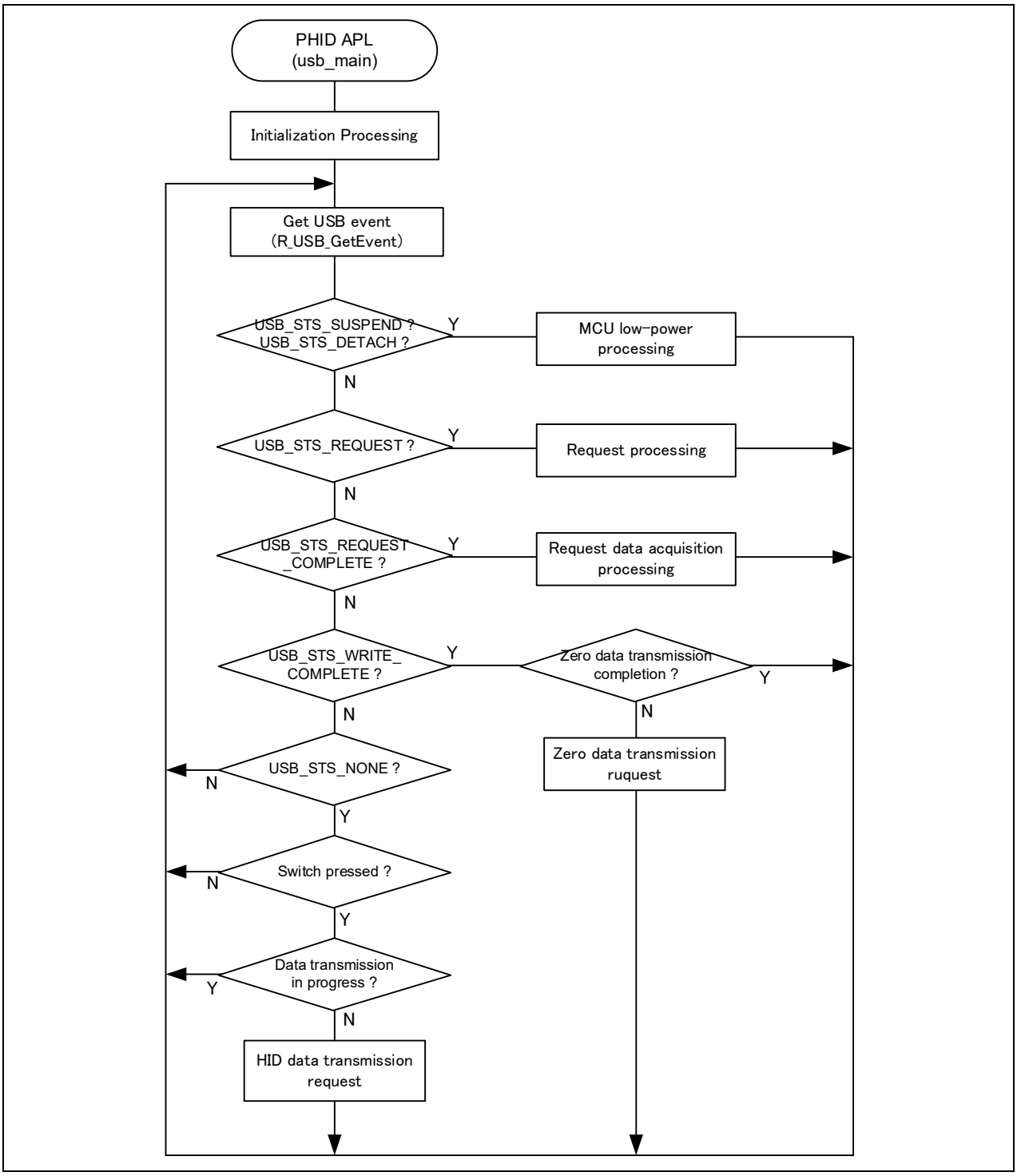


Figure 4-1 Main Loop Processing (Keyboard mode)

4.2.3 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 HID device (RSK/RSSK/EK) to low-power mode. For information on low-power mode, refer to **4.4, 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.

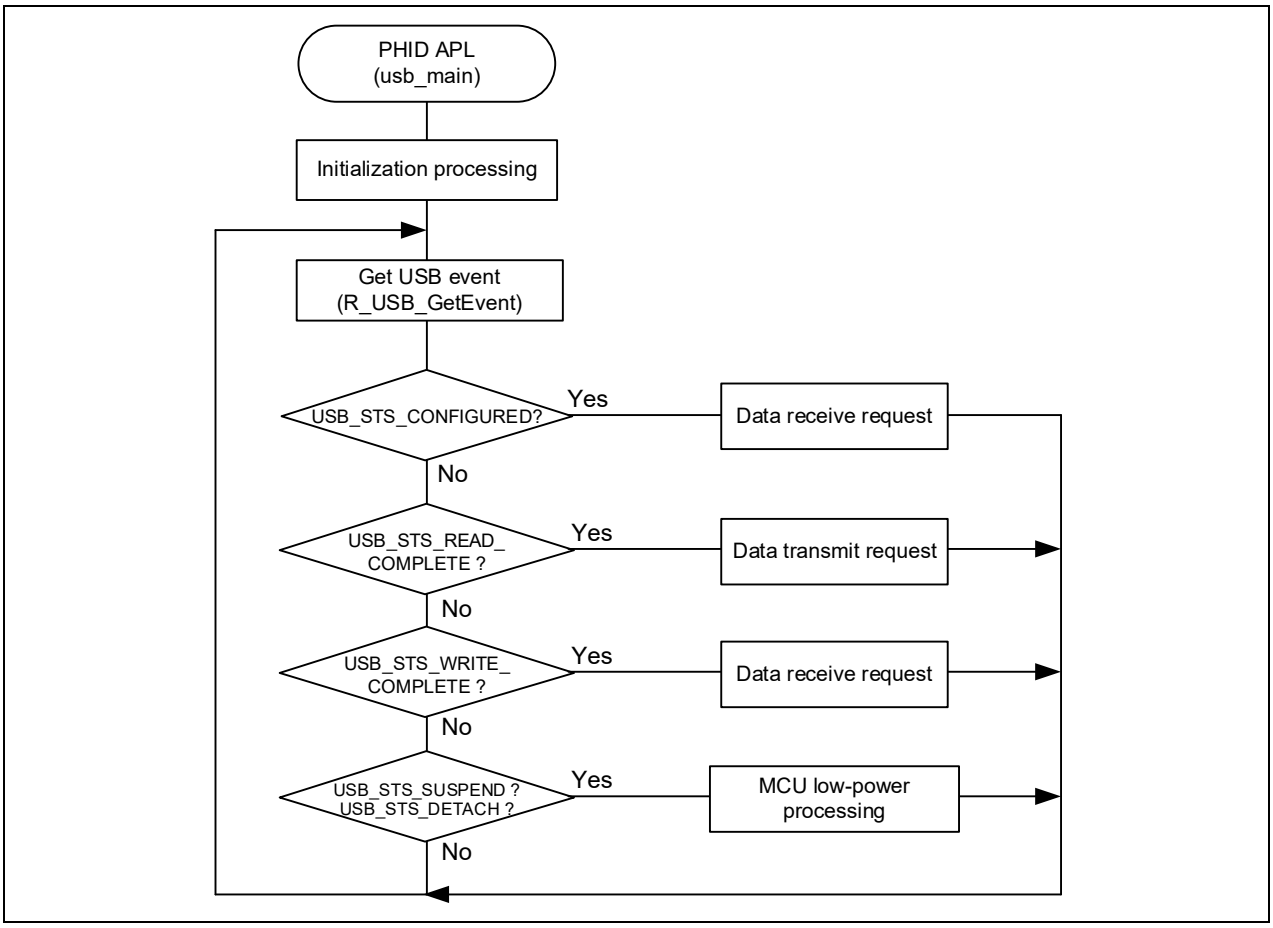


Figure 4-2 Main Loop Processing (Echo mode)

4.3 Overview of Application Processing (for RTOS)

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

4.3.1 Initial setting

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

4.3.2 Main loop (Keyboard mode)

In keyboard mode, processing is performed to send information about the switchs(buttons) on the board to the USB host. When the board (HID device) is connected to the USB host (PC), the board 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 **4.6.1, Switch specification**.

1. When a USB-related event has completed, the USB driver calls the callback function (*usb_apl_callback*). In the callback function (*usb_apl_callback*), the application task (APL) is notified of the USB completion event using the real-time OS functionality.
2. In APL, information regarding the USB completion event was notified from the callback function is retrieved using the real-time OS functionality.
3. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_REQUEST*, APL performs processing in response to the received request.
4. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_REQUEST_COMPLETE*, APL performs processing to make request information settings, etc.
5. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_WRITE_COMPLETE*, APL checks whether the transmission completion data is zero data or not.Note that the transmission processing is performed in step 7, below.
   (1). If the transmission completion data is not zero data, APL performs a data transmission request to send zero data (8 bytes) by calling the *R_USB_Write* function. (In keyboard mode it is necessary to send zero data to inform the USB host that the key input has been released.)
   (2). If the transmission completion data is zero data, APL sets *NO_WRITING* to the status management variable.
6. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_SUSPEND* or *USB_STS_DETACH*, APL performs processing to transition the board (HID device) to low-power mode. For information on low-power mode, refer to **4.4, MCU Low power consumption processing**.
7. APL checks whether switch(button) on the RSK is pressed or not after the above processing.If the switch 2(SW2) or switch 3(SW3) is pressed, 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.

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

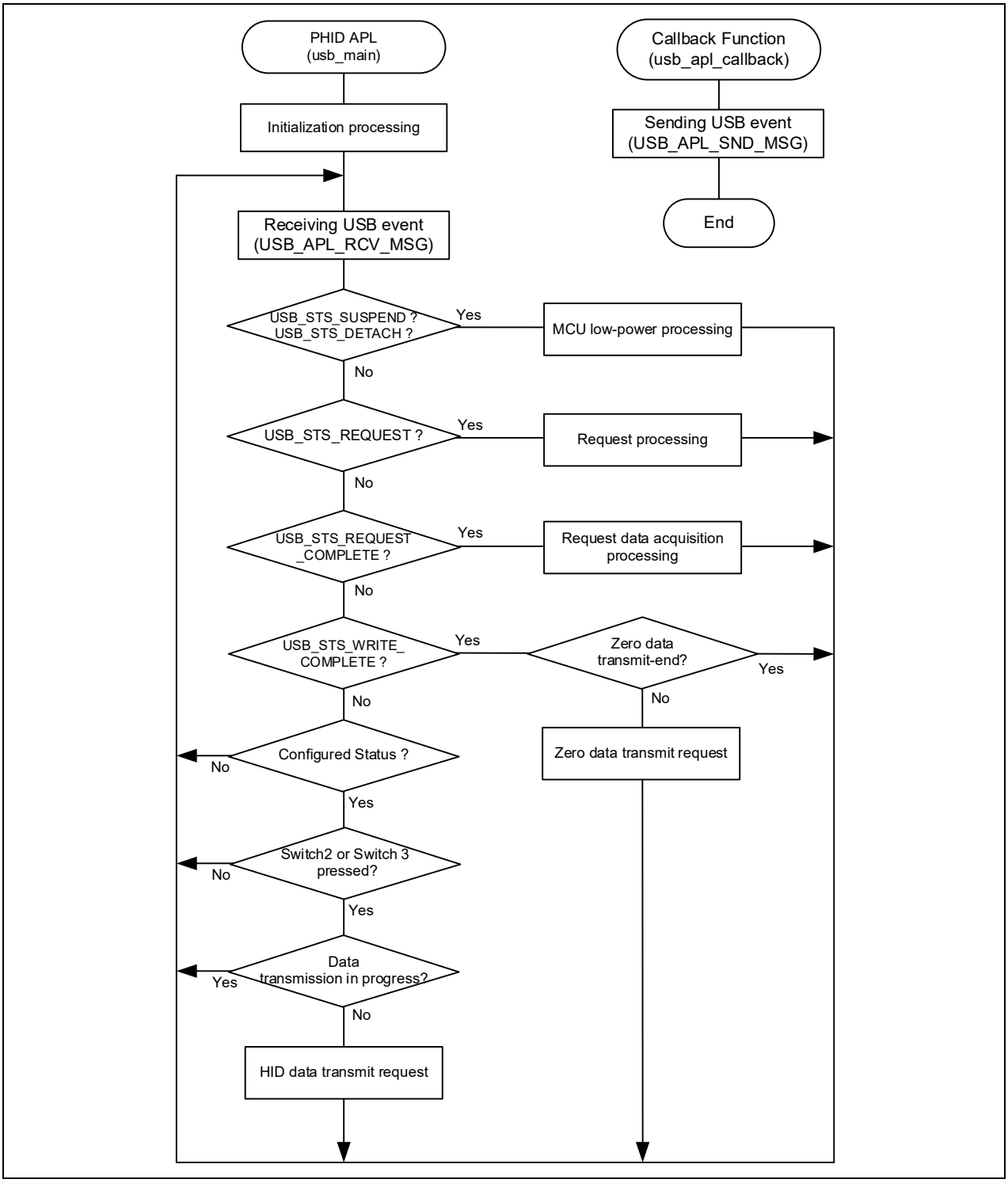


Figure 4-3 Main Loop Processing (Keyboard mode)

4.3.3 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 a USB-related event has completed, the USB driver calls the callback function (*usb_apl_callback*). In the callback function (*usb_apl_callback*), the application task (APL) is notified of the USB completion event using the real-time OS functionality.
2. In APL, information regarding the USB completion event was notified from the callback function is retrieved using the real-time OS functionality.
3. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_CONFIGURED*, APL performs a data reception request to receive data transmitted from the USB Host by calling the *R_USB_Read* function.
4. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_READ_COMPLETE*, APL performs a data transmission request to send USB Host the reception data by calling the *R_USB_Write* function.
5. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_REQUEST*, APL performs processing in response to the received request.
6. If the USB completion event (the *event* member of the *usb_ctrl_t* structure) retrieved in step 2 above is *USB_STS_SUSPEND* or *USB_STS_DETACH*, APL performs processing to transition the HID device (RSK) to low-power mode. For information on low-power mode, refer to **4.4, MCU Low power consumption processing**.
7. The above processing is repeated.

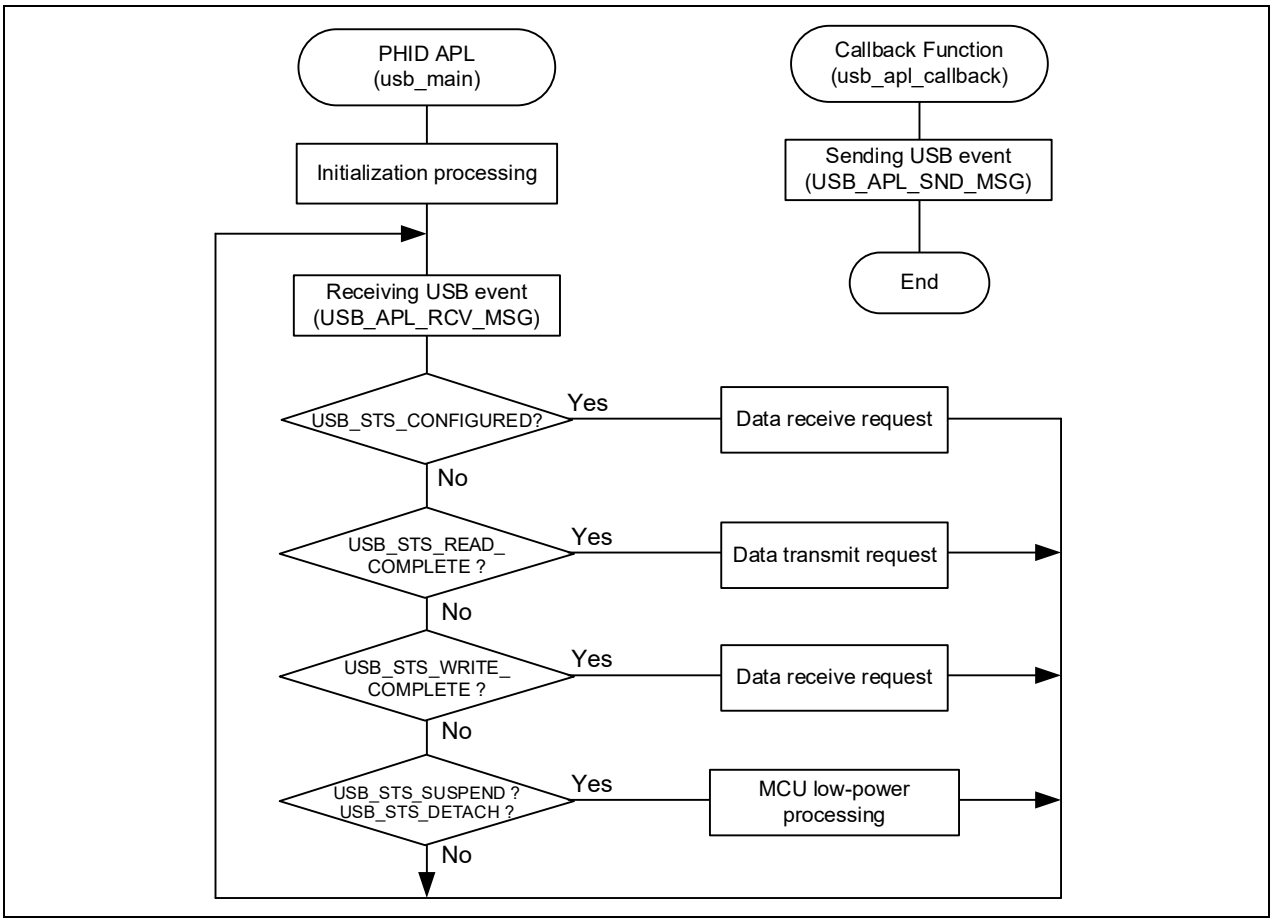


Figure 4-4 Main Loop Processing (Echo mode)

4.4 MCU Low power consumption processing

MCU low-power processing occurs when the conditions in Table 4-1 or Table 4-2 are met, causing a transition to low-power mode. To enable this processing, specify *USB_APL_ENABLE* to *USB_SUPPORT_LPW* definition in the *r_usb_phid_apl_config.h file*.

1. Non-OS

Table 4-1 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/RSSK) 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/RSSK) attaches to the USB host.
2. When a suspend signal sent by the USB host is received while the HID device (RSK/RSSK) 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.

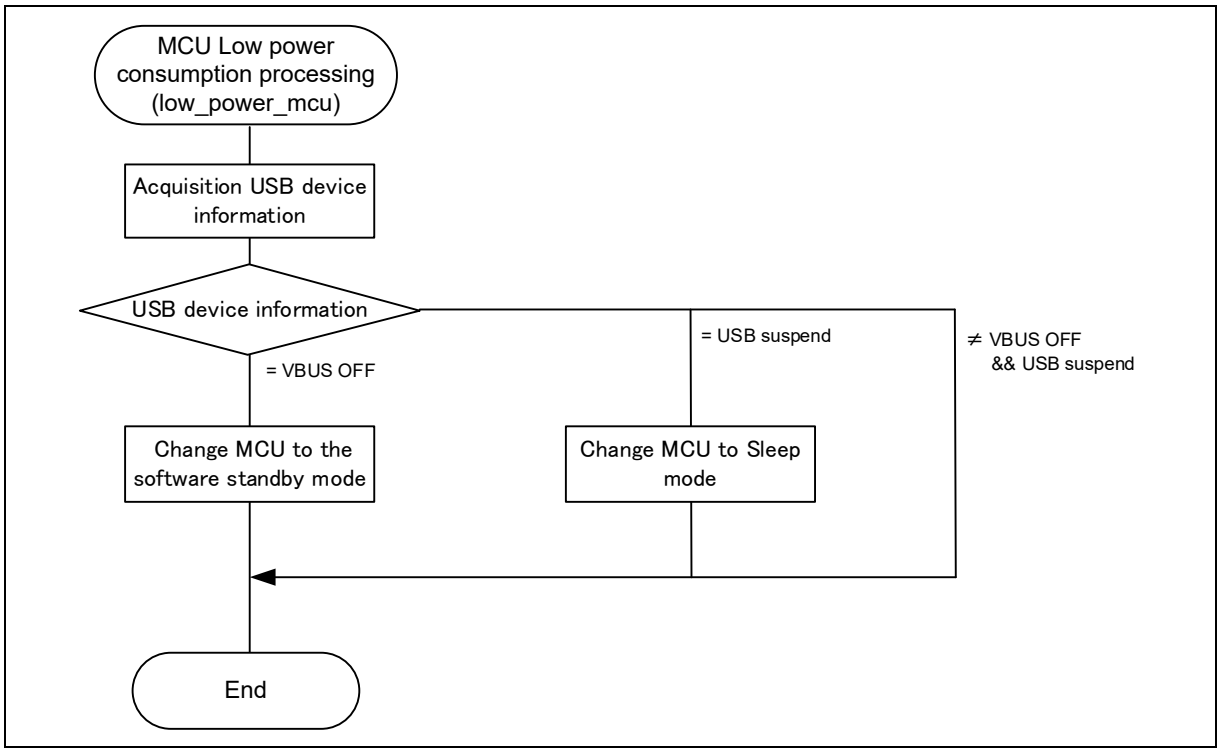


Figure 4-5 Flowchart of MCU Low Power Consumption Processsing

2. RTOS (FreeRTOS only)

Table 4-2 Conditions for Transition to Low-Power Mode

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

(1). When the board 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 board (HID device) attaches to the USB host.

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

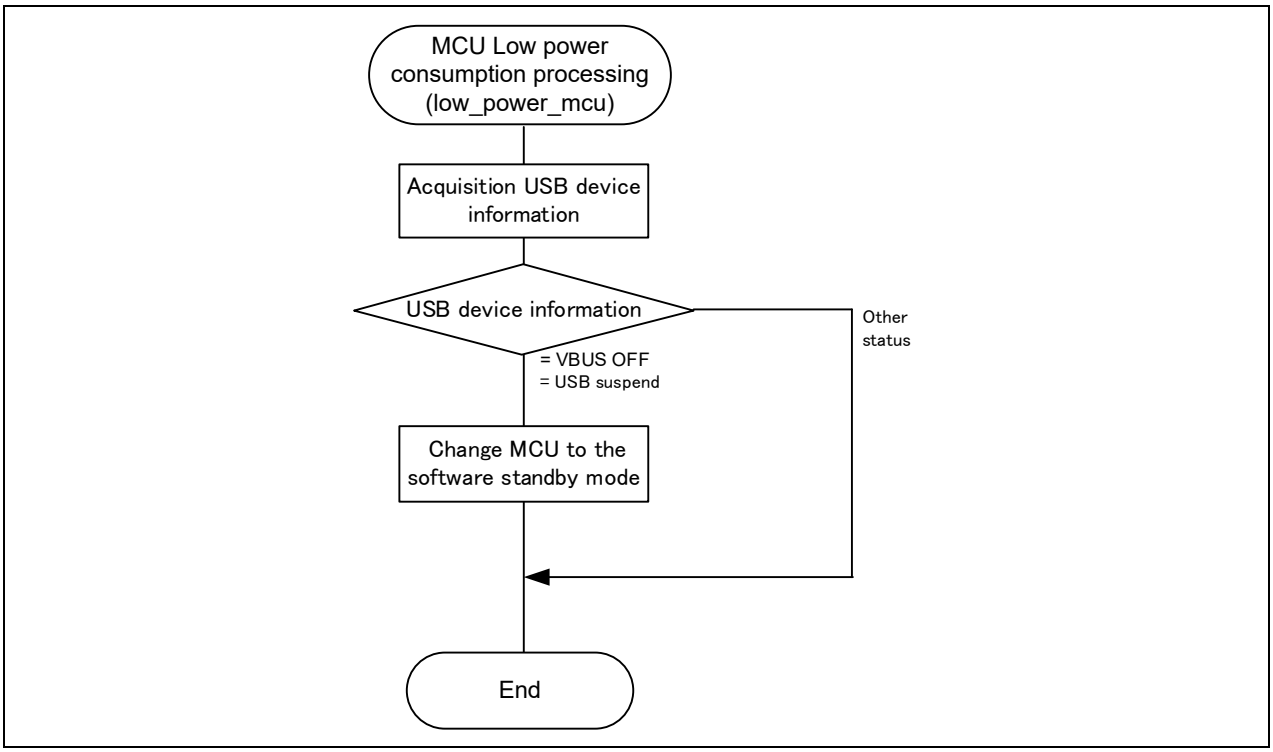


Figure 4-6 Flowchart of MCU Low Power Consumption Processsing

4.5 Configuration File for the application program (r_usb_phid_apl_config.h)

Make settings for the definitions listed below.

1. 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
```

2. Low-Power Function Definition

Specify whether or not the low-power function will be used. If the low-power function will be used, speify *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
```

3. USB_SUPPORT_RTOS Definition

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

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

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 the application note *USB Basic Mini Host and Peripheral Driver (USB Mini Firmware) using Firmware Integration Technology* (Document number. R01AN2166).

4.6 Keyboard operation

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

4.6.1 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.

1. RSK

Switch Number	Operation
Switch1(SW1)	The remote wakeup signal is sent to USB Host.
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.

2. RSSK

Switch Number	Operation
Switch1(SW1)	One of the key codes for characters “a” to “z” or “Enter” is reported to the host each time SW is pressed.
Switch2(SW2)	The remote wakeup signal is sent to USB Host.

3. EK

Switch Number	Operation
Switch1(SW1)	The remote wakeup signal is sent to USB Host.
Switch2(SW2)	One of the key codes for characters “a” to “z” or “Enter” is reported to the host each time SW is pressed.

Note:

1. The follow shows the port allocated to the switch (RSK/EK: Switch1, RSSK: Switch2).

	Port	Remarks
RX111	P30	
RX113	PA4	Using low-power function
	PJ0	Not uisng low-power function
RX231	P31	
RX23W	P30	
RX261	PD0	

2. The low power consumption mode is exited by pressing the switch (RSK: Switch1, RSSK: Switch2). This application program uses the following IRQ interrupt to exit the low power consumption mode.

	Port	IRQ
RX111	P30	IRQ0
RX113	PA4	IRQ5
RX231	P31	IRQ1
RX23W	P30	IRQ0
RX261	PD0	IRQ0

3. By default RSKRX113, the following setting are required since SW1 is assigned to PJ0.
   (1). After removing the option link resistance (0 ohm) mounted on R112, mount the resistance on R111.
   (2). After removing the option link resistance (0 ohm) mounted on R57, mount the resistance on R56.

4.6.2 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 4-3 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

4.7 Descriptor

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

5. Class Request

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

Table 5-1 Supported Basic Requests and HID Class Requests

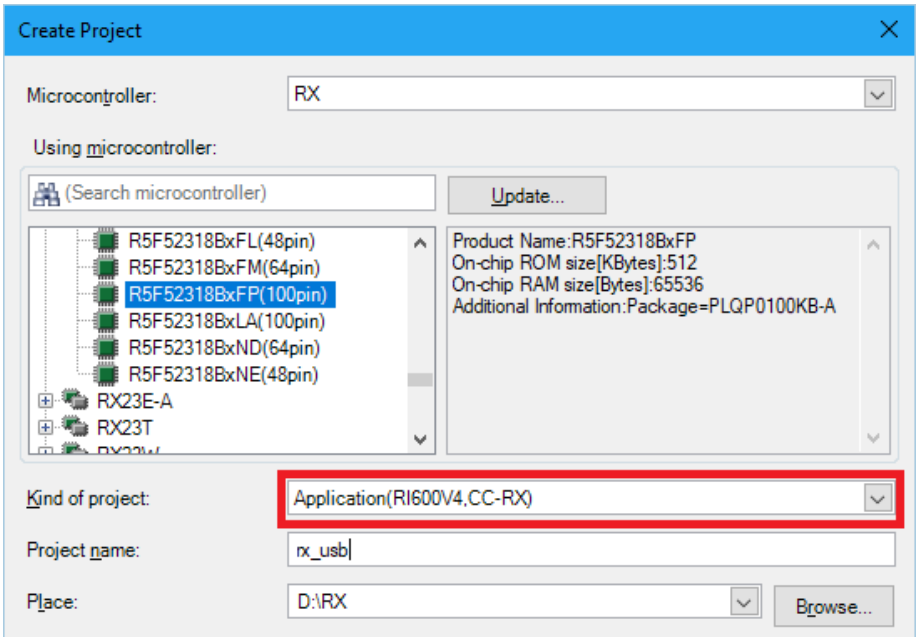
Request	Code	Description	Supported
Get Report	0x01	Sends a report to the USB Host	YES
Set Report	0x09	Receives a report from the USB Host	YES
Get Idle	0x02	Sends a duration (time) to the USB Host	YES
Set Idle	0x0A	Receives a duration (time) from the USB Host	YES
Get Protocol	0x03	Sends a protocol to the USB Host	NO
Set Protocol	0x0B	Receives a protocol from the USB Host	NO
Get Report Descriptor	Standard	Sends a report descriptor to the USB Host	YES
Get HID Descriptor	Standard	Sends an HID descriptor to the USB Host	YES

6. Using RI600V4 project with CS+

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

6.1 New Project Creation

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

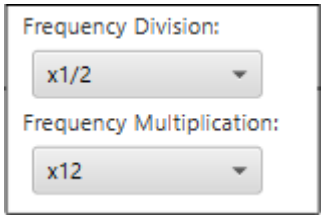


6.2 Launch Smart Configurator

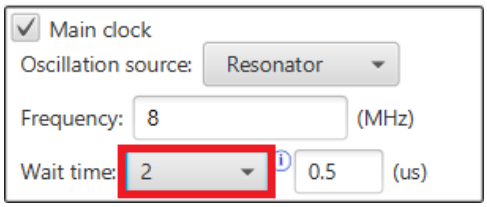
1. Clock Setting (Select "Clocks" tab)

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

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



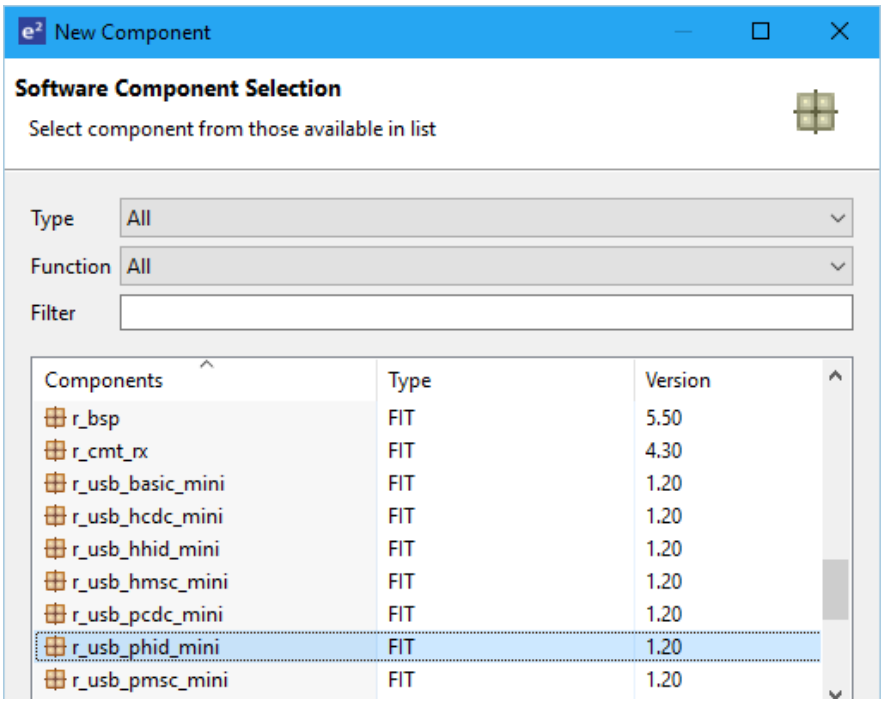
(2). Set the minimum value to the wait time for the main clock.



2. Component Setting (Select "Components" tab)

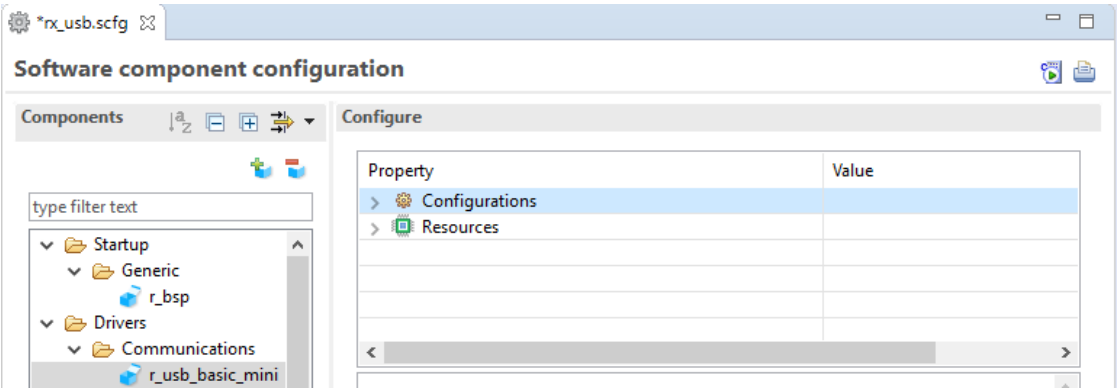
(1). Import the USB FIT module

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



(2). Configuration Setting

a. r_usb_basic_mini

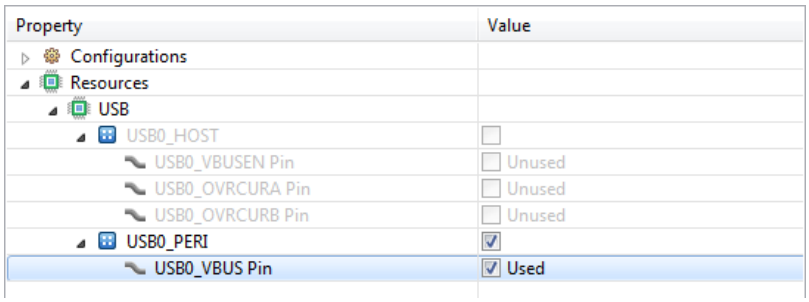


(a). Configurations

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

(b). Resources

Check the check box for USBx_VBUS pin.

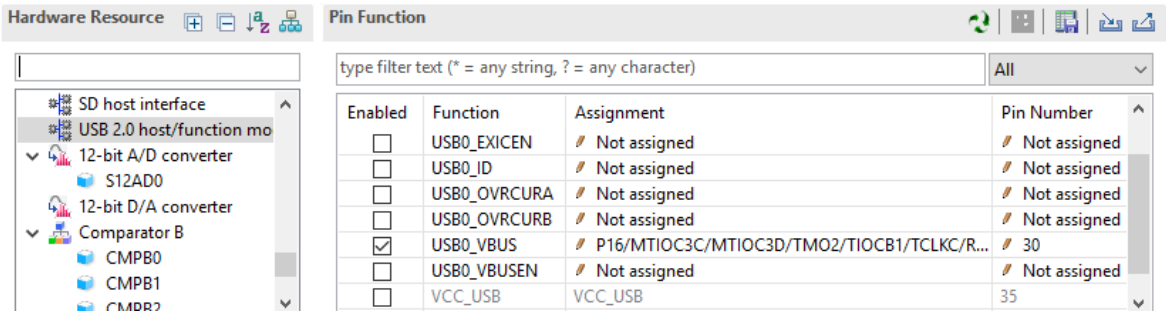


b. r_usb_phid_mini

Refer to chapter "Configuration" in *USB Peripheral Human Interface Devices Class Driver (PHID) for USB Mini Firmware Firmware Integration Technology* application note (Document number: R01AN2171).

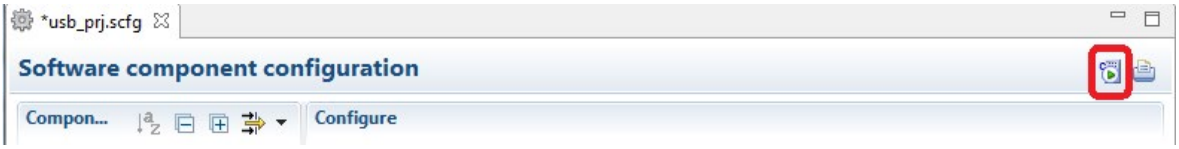
3. Pin Setting (Select "Pins" tab)

Select the port for USB pin match the user system.



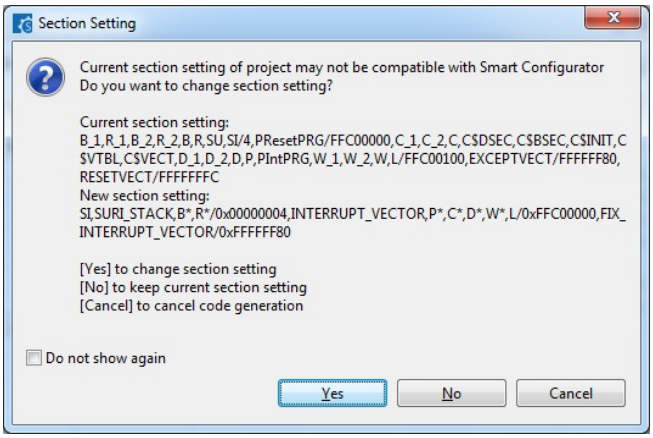
4. Generate Code

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



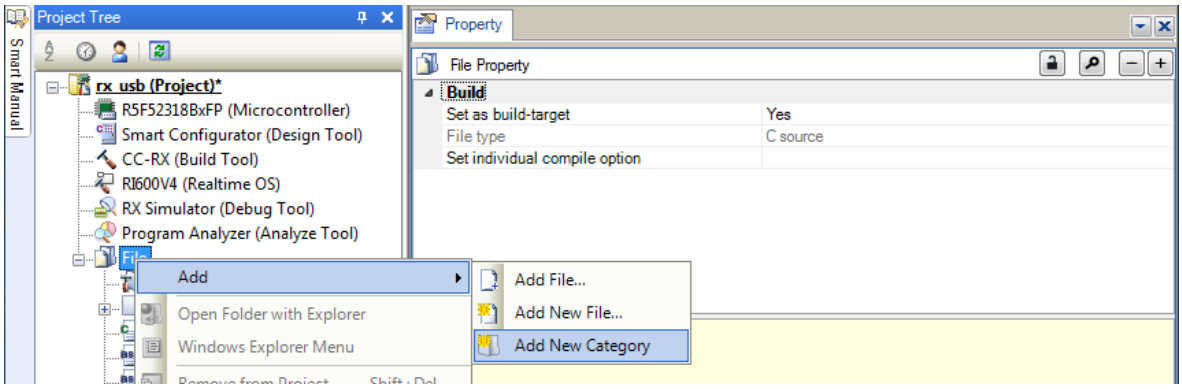
Note:

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



6.3 Add the application program and the configuration file

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



Note:

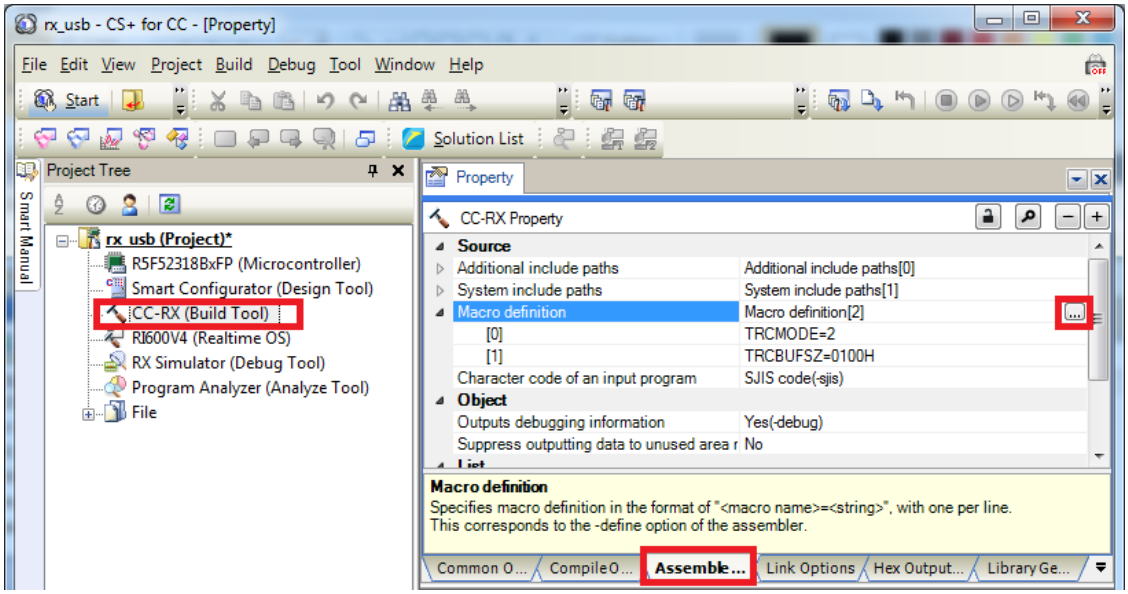
Remove the "task.c" file and "sample.cfg" created in "*<ProjectDir>*" folder by CS+.

6.4 Remote Macro Definition

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

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

1. TRCMODE = 2
2. TRCBUFSZ = 0100H



6.5 Build Execution

Excecute the build and generate the binary target program.

7. Using the e² studio project with CS+

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

[Note]

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

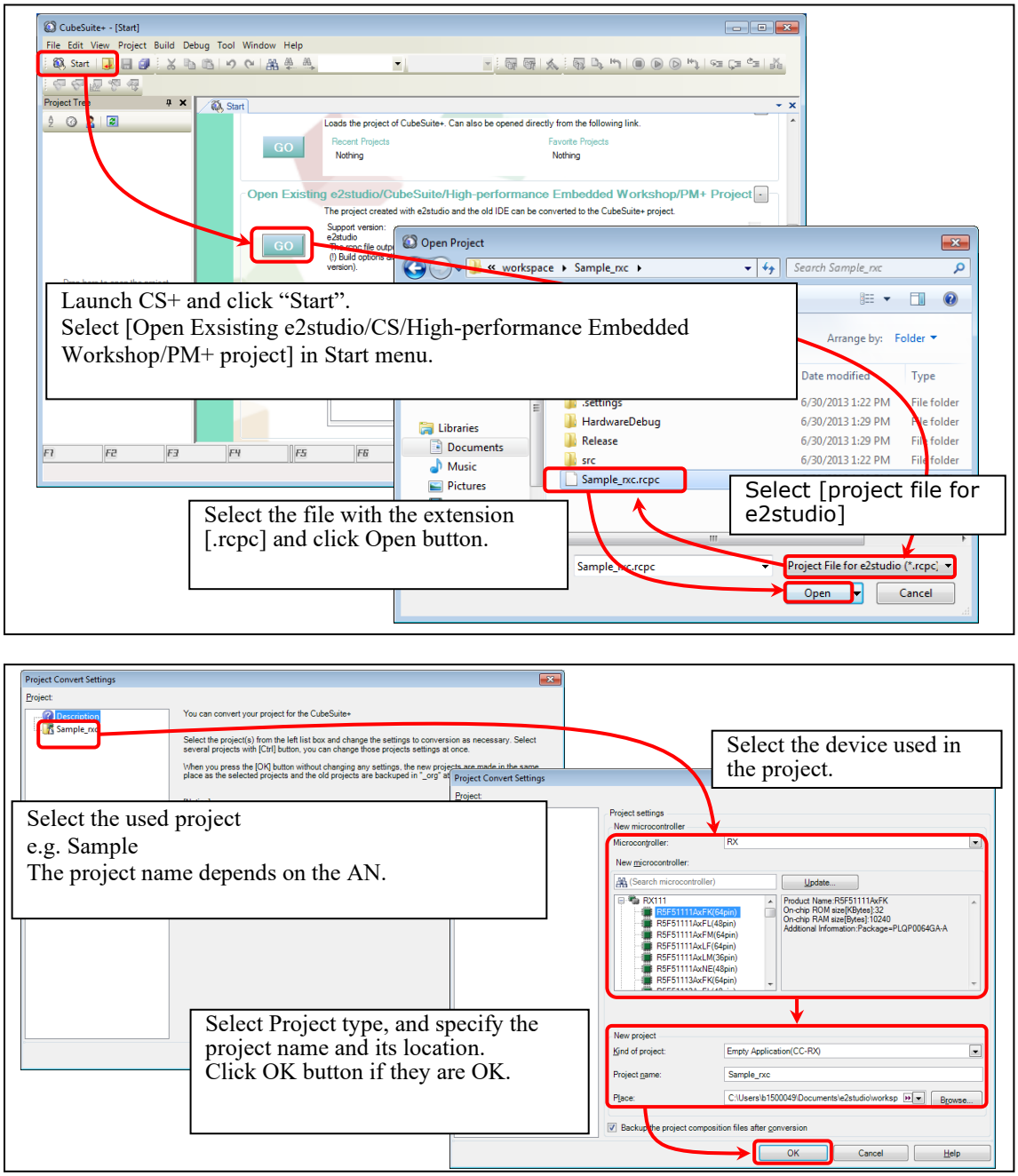


Figure 7-1 Using the e² studio project with CS+

Website and Support

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

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

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

Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Dec 1, 2014	—	First Edition Issued.
1.01	Jun 1, 2015	—	RX231 is added in Target Device.
1.02	Dec 28, 2015	—	1. The sample application program using API for OUT transfer is added. 2. Checked the operation with Windows® 10.
1.10	Nov 30, 2018	—	1. The following chapter has beed added. (1). 3.1.2 RSK/RSSK Setting 2. The following chapter has beed changed. (1). 4. Sample Application
1.12	Jun 30, 2019	—	RX23W is added in Target Device.
1.20	Jun 1, 2020	—	Supported the real-time OS.
1.30	Jul 31, 2024	—	RX261 is added in Target Device.
1.32	May 15, 2026	—	Included spdx file and json file in the package.

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/.

© 2026 Renesas Electronics Corporation. All rights reserved.