APPLICATION NOTE

Renesas USB MCU and USB ASSP

LibUSB - A Complete RX USB Function and PC Host Solution

R01AN0492EJ0200
Rev. 2.00
Sep 30, 2012.

Introduction

You don’t necessarily have to follow a class specification to use USB. That is, in many fields it is not necessary to follow a certain USB class specification.

Certain professional domains mandate a certain class, and sometimes it is more convenient for a user to not have to do any installs for common device classes, where drivers are already installed on a system (a mouse, a keyboard). In such case, Renesas has support for many classes. For other situations, where an application is unique, simple, etc, there is no reason to follow a complicated class specification. In this case one is free to use a so called "Vendor" class, and for that, use "LibUSB". This can make development significantly faster. LibUSB is the “easy” choice for deploying a product.

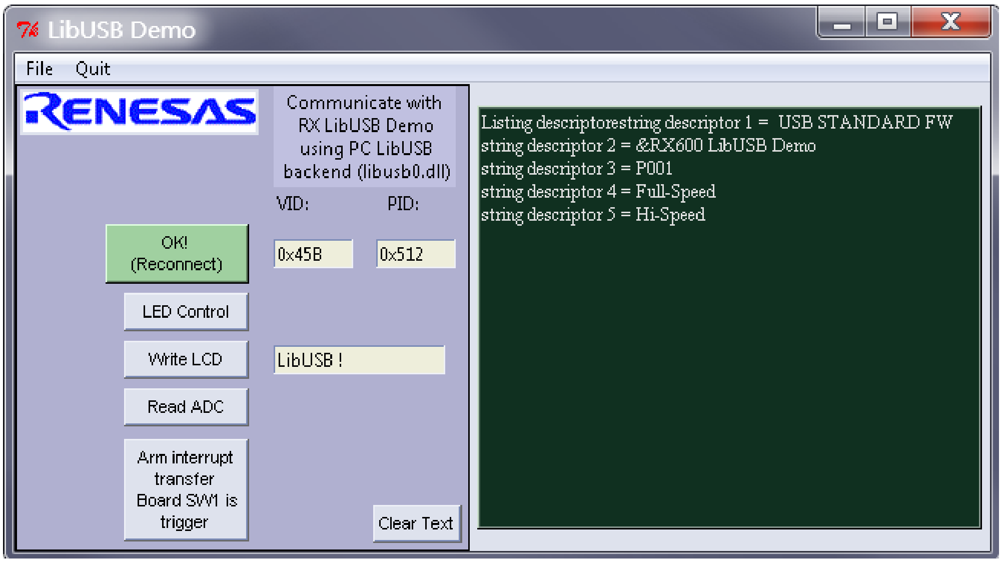
LibUSB is a PC driver library that enables user space application programs to communicate with USB devices. For example for Windows, this is described in the “libusb-win32” download available at https://sourceforge.net/apps/trac/libusb-win32

Even if a class must be used, LibUSB can be used to issue host-to-device Standard Control commands to easily during development test the target driver’s behavior. For example, iterate through enumeration step by step when developing the RX target code.

Summary: LibUSB is a great tool for getting prototype targets up and running fast, to exchange data with a USB host, without having to follow the rigid specs of a class, when not necessary. This document explains how to use the Peripheral LibUSB Demo on the RSK-RX63N, together with the PC host application *Renesas_libusb_host*.

Target Device

USB Full Speed: RX62N Group, RX621 Group, RX63N Group, RX631 Group.
USB High Speed: R8A66597 ASSP.



LibUSB Demo
File
Quit
RENESAS
Communicate with RX LibUSB Demo using PC LibUSB backend (libusb0.dll)
OK! (Reconnect)
LED Control
Write LCD
Read ADC
Arm interrupt transfer Board SW1 is trigger
Clear Text

VID: 0x45B
PID: 0x512

LibUSB !

```
Listing descriptorestring descriptor 1 =  USB STANDARD FW
string descriptor 2 = &RX600 LibUSB Demo
string descriptor 3 = P001
string descriptor 4 = Full-Speed
string descriptor 5 = Hi-Speed
```

Figure 1. In this document, LibUSB is explained, and a sample application is described.

Contents

1. Overview

This document explains using a standard PC host “LibUSB” driver together with an RX peripheral running the modified application level example code of the Renesas RX “Basic FW” (r01an0512ej_usb_basic).

1.1 Functions and Features

The RX peripheral LibUSB Demo source code will communicate with a LibUSB host PC application.

1.2 Related Documents

1. Universal Serial Bus Revision 2.0 specification
   http://www.usb.org/developers/docs
2. Renesas Electronics USB Devices
   http://www.renesas.com/prod/usb/
3. Renesas USB MCU and USB ASSP USB Basic Firmware Application Note (Document No. R01AN0512EJ)
4. RX62N Group, RX621 Group User's Manual: Hardware (Document No. R01UH0033EJ)
5. RX62N Group Renesas Starter Kit + for RX62N User's Manual (Document No. REJ10J2198-0201)
6. RX630 Group User's Manual: Hardware (Document No. R01UH0040EJ)
7. RX630 Group Renesas Starter Kit for RX630 User's Manual (Document No. R20UT0292EG0100)
8. RX63N Group, RX631 Group User’s Manual: Hardware (Document No. R01UH0041EJ)
9. RX63N Group Renesas Starter Kit + for RX63N User’s Manual
10. R8A66597 Datasheet (Document No. REJ03F0229)

1.3 Terms and Acronyms

The following terms and acronyms are used in this document:

• ASSP	:	Application Specific Standard Produce
• GUI	:	Guided User Interface.
• LibUSB backend		The Windows USB driver libusb0_x.x.dll, downloaded together with libusb-win32 if you are using Windows.
• non-OS	:	USB basic firmware for OS-less system
• pstd	:	Prefix of function and file for Peripheral USB-BASIC-F/W
• RSK		Renesas Starter Kit
• RTOS	:	USB basic firmware for uITRON system
• R8A66597	:	Renesas Hi-Speed USB2.0 ASSP R8A66597 board. Use in combination with RX62N-RSK.
• SW1/SW2/SW3	:	User switches on the RX62N-RSK and RX63N-RSK and RX630-RSK.
• USB	:	Universal Serial Bus
• USB Basic FW	:	USB basic firmware for Renesas USB device (non-OS& RTOS)

2. Physical Configuration

The peripheral LibUSB Demo source code allows the USB PC host application *Renesas_libusb_host* to manipulate the connected RSK board via USB.

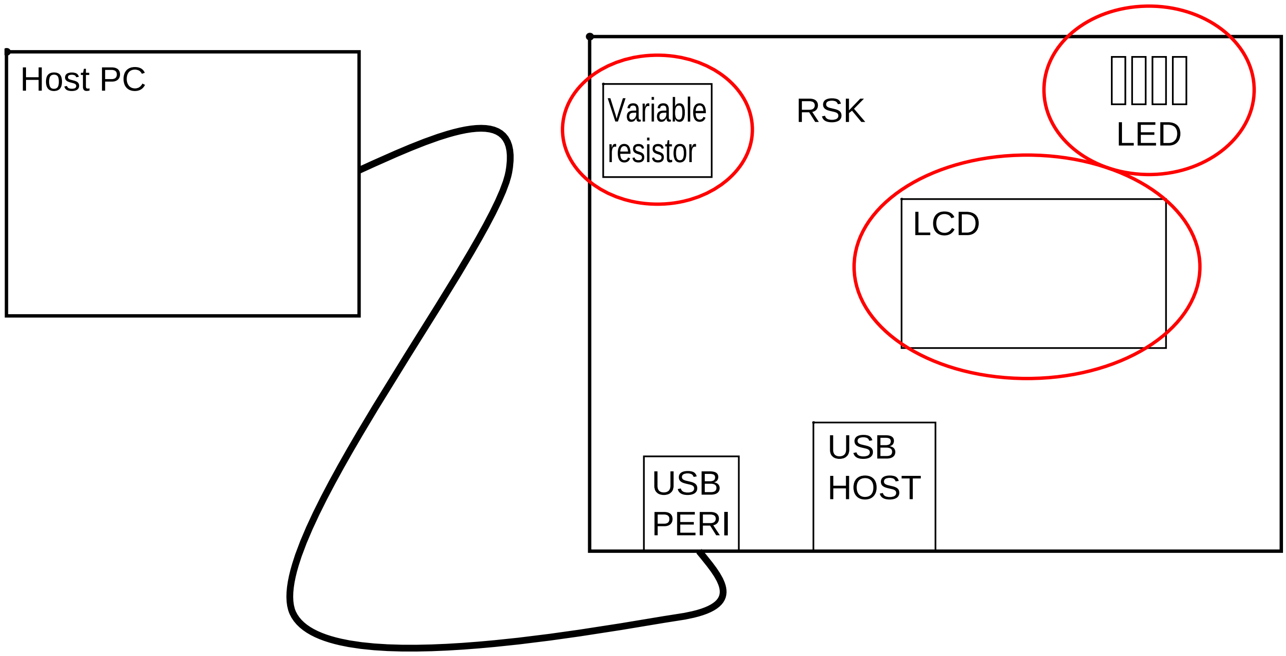


Figure 2. System physical setup.

3. LibUSB Demo Description

- Enumeration occurs when the target RX board is connected to the USB host (a PC in most cases).
- When the PC USB host application *Renesas_libusb_host* is run a connection via the PC LibUSB backend to the RX target occurs if host finds a matching VID and PID. By default in the firmware, VID is 0x45B, and PID is 0x512. *This must be changed for the peripheral by the customer when developing a product!*
- LED demo: The RX board LED light pattern is changed when the LED command is sent from the host.
- LCD demo: The RX board LCD display is changed using user input from host when the LCD command is sent from the host.
- AD demo: The RX board AD value is sent to the host PC when the read ADC command is sent from the host. The value of the variable resistor connected to the AD as shown in the above figure is read and sent to the host PC.

4. USB Host

This section will show how to get the LibUSB driver and host LibUSB Demo running from the host's perspective (the PC). Also how to create your own INF-file.

4.1 Run the Demo

1. Copy the RX board specific code from ..\Workspace\HwResourceForUSB_RX6xyz to ..\Workspace\HwResourceForUSB.
2. If using HEW: Switch to the appropriate debug session (e.g. with RSK63n use E1E20_63N).
3. Compile, download and run the target board.
4. Connect an A-to-Mini-B USB cable between PC and RX target board.

Install the LibUSB PC driver libusb.dll

5. Windows should show the "Found New HW Wizard". Point to the ..\LibUSB_Host_PC folder which contains INF-files for the default demo.
   You must later create your own INF file, for your product with your VID and PID. See 'Create Your Proprietary Product INF-file' below.
6. If *libusb_x.sys* is not already on your PC, point to the folder ..*\LibUSB_Demo\LibUSB_Host_PC\libusb-win32-bin-1.2.x.0\bin*
7. The Found New HW Wizard should copy the driver to e.g. ..*/Windows/system32*. If you are having any problems installing the driver, see separate section "Windows 'Found New Hardware Wizard' (application note).

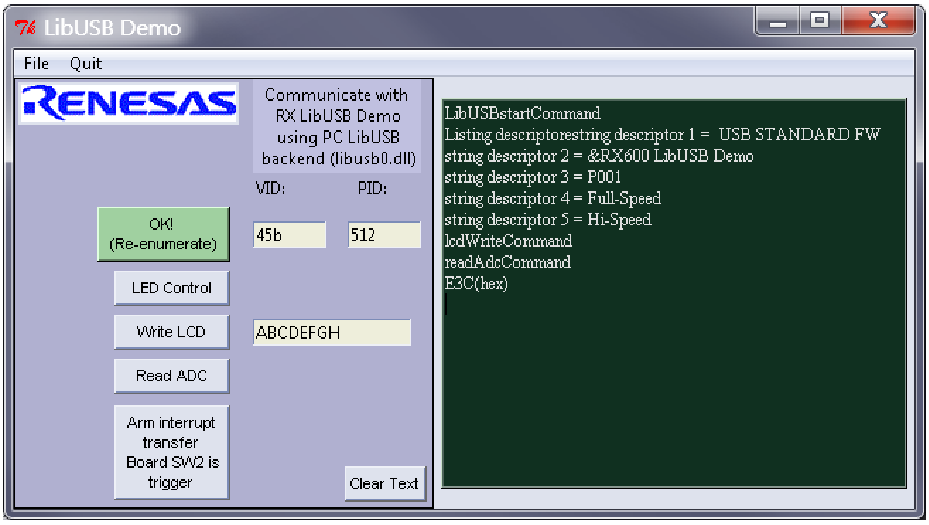
Run the host demo program

8. Run ..*\LibUSB_HostPC\host_exe\Renesas_libusb_host\dist\Renesas_libusb_host.exe.*

   The host will manipulate in order the LEDs, the LCD, then read the ADC, in a loop.

9. If you prefer using a GUI, run ..*\LibUSB_HostPC\host_exe\Renesas_libusb_host_gui\dist\Renesas_libusb_host_gui.exe*

   Connect with the target board using the "Connect" button with VID and PID set as in file *r_usb_vendor_descriptor.c*. Then, manipulate the LED, the display, and read the board ADC using the control buttons in the GUI.


LibUSB Demo
File
Quit
RENESAS
Communicate with RX LibUSB Demo using PC LibUSB backend (libusb0.dll)
OK! (Re-enumerate)
LED Control
Write LCD
Read ADC
Arm interrupt transfer Board SW2 is trigger
Clear Text

VID: 45b

PID: 512

ABCDEFGH

```
LibUSBstartCommand
Listing descriptorestring descriptor 1 =  USB STANDARD FW
string descriptor 2 = &RX600 LibUSB Demo
string descriptor 3 = P001
string descriptor 4 = Full-Speed
string descriptor 5 = Hi-Speed
lcdWriteCommand
readAdcCommand
E3C(hex)
```

Figure 3. The commands to control the LED, the display, etc from the host PC user application, are implemented via the PC host "LibUSB backend", that is, the driver libusb.dll.

4.2 Create Your Own Product INF-file

1. Change the VID and PID number in *..\Workspace\SmplMain\APL\r_usb_vendor_descriptor.c*. Compile, connect, download, and run.
2. Run *..\libusb-win32-bin-1.2.4.0\bin\inf-wizard.exe* and follow the GUI. The RX with your new VID and PID should show up. Enter desired Manufacturer and Device.
3. Enumeration: When Windows shows the "Found New HW Wizard", point to the above created INF file.
4. If libusb0.sys is needed, point to the folder *..\LibUSB_Demo\LibUSB_Host_PC\libusb-win32-bin-1.2.6.0\bin*
5. The Found New HW Wizard should copy the driver to e.g. Windows/system32.

4.3 Following the Demo in the Debugger

Set a breakpoint in file *r_usb_vendor_papl.c* at
usb_plibusb_LibUsbCmdChk();

When a command to a demo endpoint is received, the break will trigger and you can follow the demo on the target.

4.4 Creating Your Own Host Application

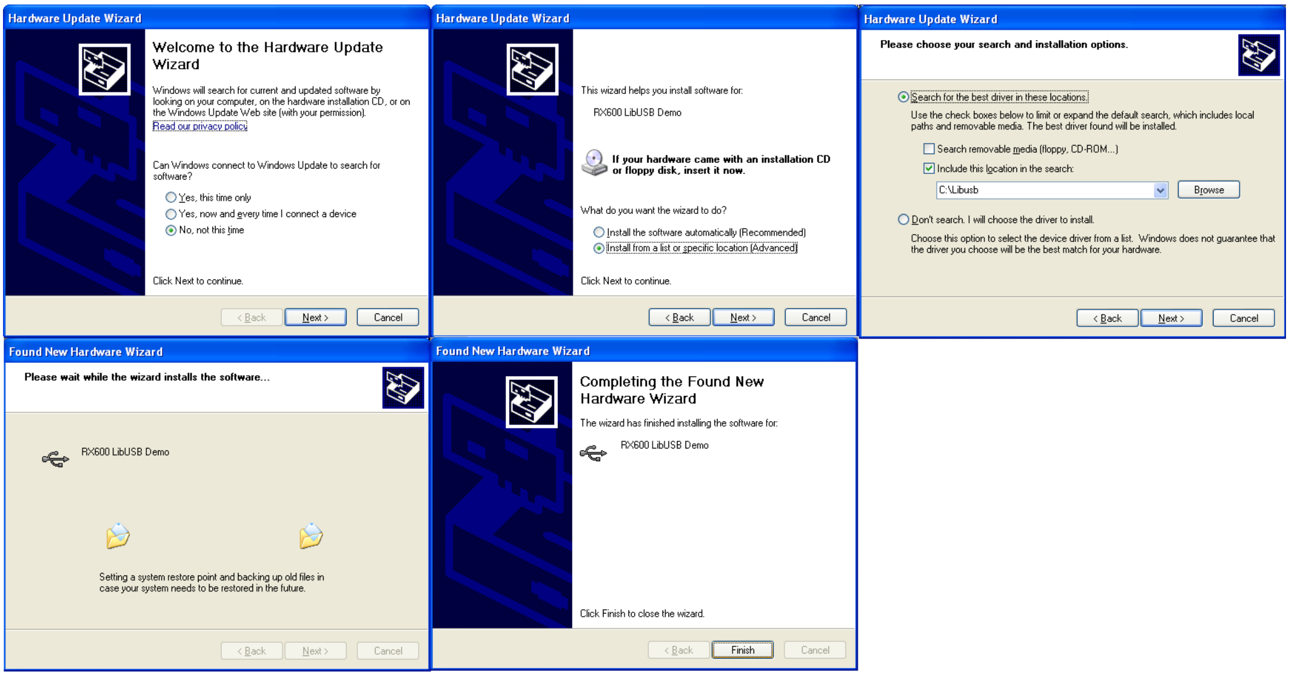
Python

There is no need to purchase a license to use Python. See the included files *Renesas_pyusb.txt* and *Renesas_libusb_host .py* for instructions how to run Python and PyUSB. Once these are installed, the Python code for PC is easy to change. Run the code by right-clicking on Renesas_Host_LibUSB.py and selecting "Edit with Idle".

4.5 Windows 'Found New Hardware Wizard'

Do the following in detail to learn how this works, to understand how to implement it for customer doing this for your new product, or if you are having any difficulty with the LibUSB driver.

1. Start Windows Device Manager (type "device manager " into the Windows Run menu) to keep track of enumeration status. Keep it open.
2. Restart the target by disconnecting and reconnecting the cable. We will follow the 'Found New HW Wizard'. (The correct driver (libusb0.sys) will be found automatically by Windows if the vendor and product ids were used by Windows at some time in the past.)
3. If the 'Found New HW Wizard' does not appear, right click on 'Lib USB Demonstration' in the Device Manager and click on 'Update Driver.' Then follow these same steps.
4. In the Windows 'Found New HW wizard', select "No, not this time' (Win7; Browse my computer…), then 'Install from a specific location' (Win7; Let me pick…).
5. If you saved your INF-file to Window's INF-directory and are given a choice of several INF-files, identify yours by matching manufacture name.
6. To be completely specific, select 'Don't search, I will choose driver…' then 'Have disk', to make <u>sure</u> we pick the right INF-file.
7. The LibUSB driver for windows; libusb0.sys was probably already installed if you ran the INF Wizard, to the *c:\Windows\system32* folder, but if asked, select 'Browse' and point to directory *..\LibUSB_Host_PC\libusb-win32-bin-1.2.5.0\bin\x86.*
8. Press OK and Finish.



Hardware Update Wizard

Welcome to the Hardware Update Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).
Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.

< Back | Next > | Cancel

Hardware Update Wizard

This wizard helps you install software for:

RX600 LibUSB Demo

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)
● Install from a list or specific location (Advanced)

Click Next to continue.

< Back | Next > | Cancel

Hardware Update Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

□ Search removable media (floppy, CD-ROM...)
☑ Include this location in the search:

C:\Libusb | Browse

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

< Back | Next > | Cancel

Found New Hardware Wizard

Please wait while the wizard installs the software...

RX600 LibUSB Demo

Setting a system restore point and backing up old files in case your system needs to be restored in the future.

< Back | Next > | Cancel

Found New Hardware Wizard

Completing the Found New Hardware Wizard

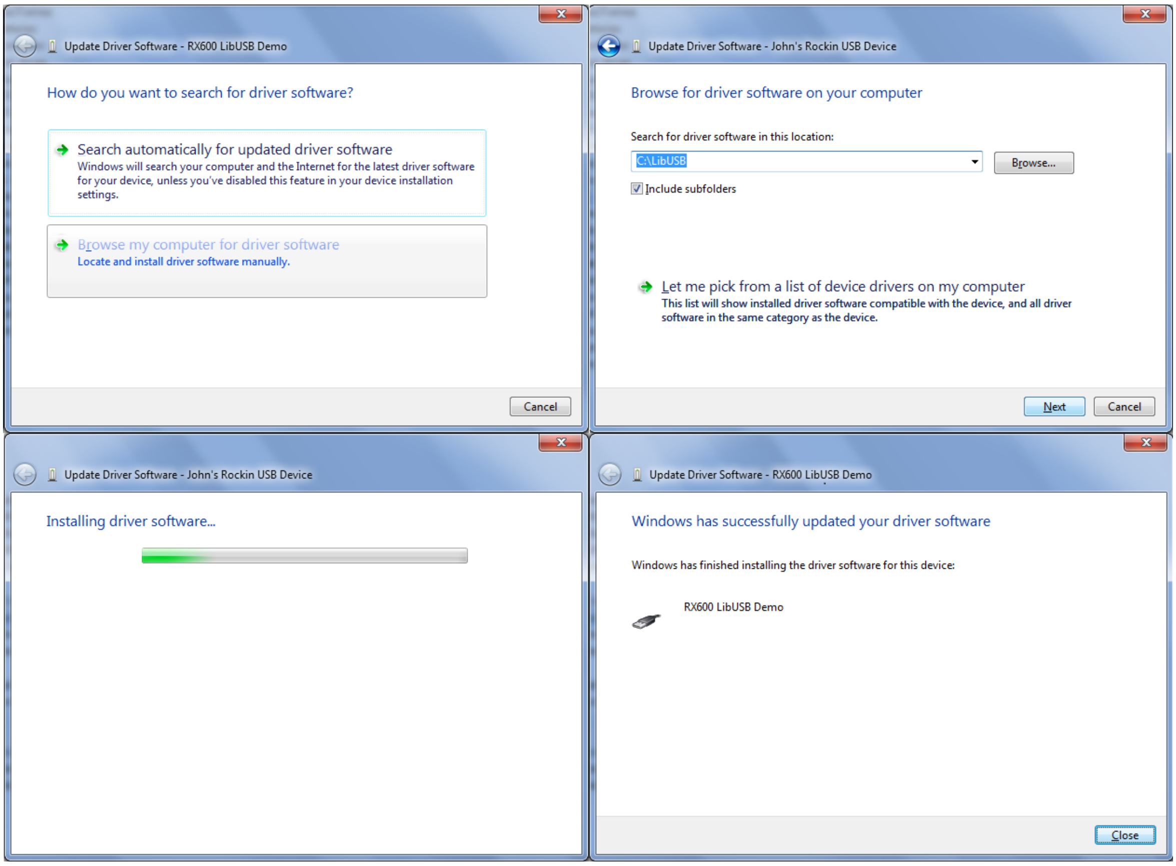
The wizard has finished installing the software for:

RX600 LibUSB Demo

Click Finish to close the wizard.

< Back | Finish | Cancel

Figure 4. If something goes wrong, restart the target firmware, uninstall the driver by right-clicking on the device in Device Mgr, then redo the procedure. Example above is for WinXP. Elow for Win7.



Update Driver Software - RX600 LibUSB Demo

How do you want to search for driver software?

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

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

Cancel

Update Driver Software - John's Rockin USB Device

Browse for driver software on your computer

Search for driver software in this location:

C:\LibUSB | Browse...

☑ Include subfolders

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

Next | Cancel

Update Driver Software - John's Rockin USB Device

Installing driver software...

Update Driver Software - RX600 LibUSB Demo

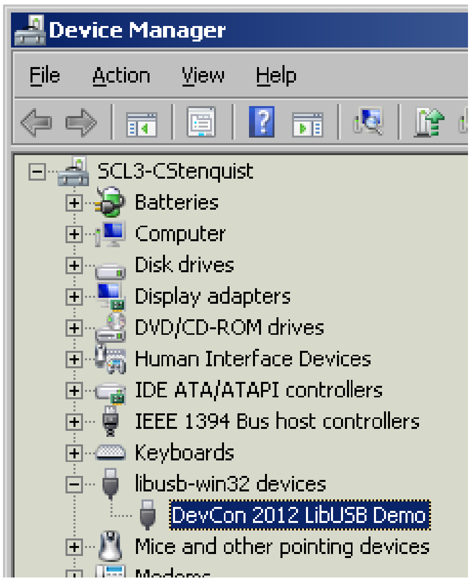
Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

RX600 LibUSB Demo

Close

9. Check that Windows has enumerated the RX properly. In Windows Device Manager you should see the RX LibUSB peripheral. See step 1 and Figure 5.



- SCL3-CStenquist
  - Batteries
  - Computer
  - Disk drives
  - Display adapters
  - DVD/CD-ROM drives
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - IEEE 1394 Bus host controllers
  - Keyboards
  - libusb-win32 devices
    - DevCon 2012 LibUSB Demo
  - Mice and other pointing devices
  - Modems

Figure 5. The RX LibUSB peripheral should shows up in Device Manager, listed under the "libusb-win32 device" class.

5. USB Peripheral

5.1 Software Configuration

LibUSB Demo is implemented using the USB Basic Firmware. There is a separate manual for the Basic FW as noted in Related Documents above. This manual must be studied to set up the code for the RX MCU device you are using. For example, the HwResources folders in the source code must be set up accordingly. Here we will focus on the LibUSB specific changes made to the Basic FW package to implement the LibUSB Demo.

5.2 Block Diagram

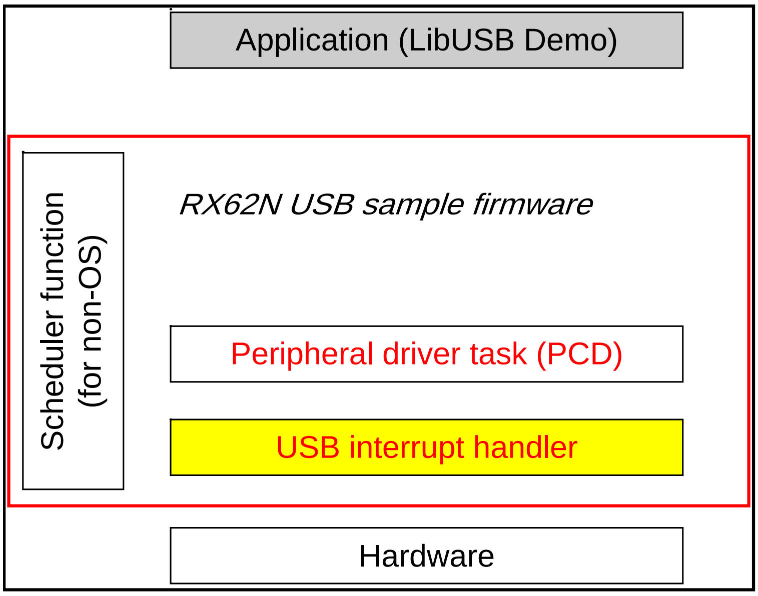

Application (LibUSB Demo)
Scheduler function (for non-OS)
RX62N USB sample firmware
Peripheral driver task (PCD)
USB interrupt handler
Hardware

Figure 6. Software block diagram

5.3 Code Change vs. USB Basic FW

The table below lists the most important functions that were added to for the LibUSB Demo. The function *usb_pvendor_apl_task* was also modified slightly to not invoke read and write of demo endpoints until the user (host) asks for data. Other functions, e.g. to read ADC and were also added.

File	Code	Outline
SmplMain\ APL\ r_usb_vendor_papl.c	usb_plibusb_ LibUsbCmdChk()	Demonstrates usage of bulk OUT endpoint: Host writes to EP to control LED Host writes to EP to control LCD Demonstrates usage of bulk IN endpoint: Host reads EP to get ADC value
SmplMain\ APL\ r_usb_vendor_papl.c	usb_vendor_int_ in_test_func()	Demonstrates usage of interrupt IN endpoint: Host reads switch press. (Not used in primary demo.)
SmplMain\ APL\ r_usb_vendor_descriptor.c	usb_gpvendor_smpl _eptbl1[]	Pipe Information Table, and EP descriptors changed: PIPE1 <=> EP1 OUT PIPE2 <=> EP2 IN

5.4 Pipe (Endpoint) Information Table

The Pipe Information Table is where endpoint numbers are selected for the demo. This is also where peripheral pipes are paired with endpoints for the peripheral. (The host selects its own pipes for these endpoints.) This is done in the array *usb_gpvendor_smpl_eptbl1* in file *..\WorkSpace\SmplMain\APL\r_usb_vendor_descriptor.c*. Here, for the LibUSB demo, Endpoint 1 is used and paired with Pipe1, EP2 is paired with Pipe 2. EP3 is paired with Pipe 6 and is an interrupt endpoint that is not used in the primary demo, but is used in conjunction with a switch on the board and a response output in the host demo to show how it is used. The interrupt timing period can be decreased by setting bInterval in the endpoint descriptor in the same file (5.5.1).

5.5 Descriptors

This section describes the default descriptors of the LibUSB Demo source code, which are in file *r_usb_vendor_descriptor.c*. File *r_usb_usrconfig.h* contains the most typical fields changed by the product designer, but of course any field is to be changed subject to the end product purpose.

5.5.1 Device

Field	Nr bits	Value	Description
bLength	8	0x12	Descriptor size is 18 bytes.
bDescriptorType	8	0x01	DEVICE descriptor type.
bcdUSB	16	0x0200	USB specification version 2.00.
bDeviceClass	8	0xFF	Device Class is vendor-specific.
bDeviceSubClass	8	0xFF	Device Subclass is vendor-specific.
bDeviceProtocol	8	0xFF	Device Protocol is vendor-specific.
bMaxPacketSize0	8	0x40	Maximum packet size for endpoint zero is 64.
idVendor	16		Vendor ID 0x45B. *Customer must later change this*.
idProduct	16		Product ID 0x512. *Customer must later change this*.
bcdDevice	16	0x0100	Device release number is 2.00.
iManufacturer	8	0x01	Manufacturer index in string descriptor is 1.
iProduct	8	0x02	Product index in string descriptor is 2.
iSerialNumber	8	0x03	Serial number index in string descriptor is 3.
bNumConfigurations	8	0x01	Device has one possible configuration.

5.5.2 Configuration

Field	Nr bits	Value	Description
bLength	8	0x09	Descriptor size is 9 bytes.
bDescriptorType	8	0x02	CONFIGURATION descriptor type.
wTotalLength	16	0x0037	The total length of data for this configuration.
bNumInterfaces	8	0x02	This configuration supports two interfaces; 0 and 1.
bConfigurationValue	8	0x01	Configuration nr 1 - the only configuration.
iConfiguration	8	0x00	String descriptor index describing this configuration (All string descriptors, starting with the first, belong to this configuration).
bmAttributes	8	0xC0	Configuration characteristics.
bMaxPower	8	0x32	Maximum power 100 mA.

5.5.3 Interface

Field	Nr bits	Value	Description
bLength	8	0x09	Descriptor size is 9 bytes.
bDescriptorType	8	0x04	INTERFACE descriptor type.
bInterfaceNumber	8	0x00	The number of this interface is 0.
bAlternateSetting	8	0x00	No alternate setting.
bNumEndpoints	8	0x02	The number of endpoints used by this interface is 3.
bInterfaceClass	8	0xFF	Vendor-specific.
bInterfaceSubClass	8	0xFF	Vendor-specific.
bInterfaceProtocol	8	0xFF	Vendor-specific.
iInterface	8	0x00	String descriptor index describing this interface. Example string descriptors are common to bother interfaces.

5.5.4 Endpoints

Field	Nr bits	Value	Description
bLength	8	0x07	Descriptor size is 7 bytes.
bDescriptorType	8	0x05	ENDPOINT descriptor type.
bEndpointAddress	8	0x01	**OUT endpoint number 1**.
bmAttributes	8	0x02	Type – BULK only.
wMaxPacketSize	16	0x0040	Maximum packet size for this endpoint is 64 bytes.
bInterval	8	0x01	The polling interval is every 1 ms (every frame).

Field	Nr bits	Value	Description
bLength	8	0x07	Descriptor size is 7 bytes
bDescriptorType	8	0x05	ENDPOINT descriptor type
bEndpointAddress	8	0x82	**IN endpoint number 2**.
bmAttributes	8	0x02	Type – BULK only
wMaxPacketSize	16	0x0040	Maximum packet size for this endpoint is 64 bytes.
bInterval	8	0x01	The polling interval is every 1 ms (every frame).

Field	Nr bits	Value	Description
bLength	8	0x07	Descriptor size is 7 bytes
bDescriptorType	8	0x05	ENDPOINT descriptor type
bEndpointAddress	8	0x83	**IN endpoint number 3**.
bmAttributes	8	0x03	Type – INTERRUPT only.
wMaxPacketSize	16	0x0010	Maximum packet size for this endpoint is 16 bytes.
bInterval	8	0x04	The polling interval is every 4 ms (every 4 frames).

5.6 Application details

The application layer programmer should mainly have to deal with file *..\src\SmplMain\APL\**r_usb_vendor_papl.c***. To understand application level fundamentals for using the underlying USB Basic FW, a “must read” is *General operation using ANSI API* in the Basic FW application note; “USB Basic Host and Peripheral firmware”. See related documents.

5.6.1 Peripheral Demo Processing

The LibUSB application top level user control is described in 3.

When the application receives PIPE1 data whose first byte is a command (1, 2, or 4 in the demo), the demo application layer takes action depending on the command. As a user application programmer, you can see the action take place starting with execution of function *usb_plibusb_LibUsbCmdChk*.

The above procedure is illustrated in the next flowchart.

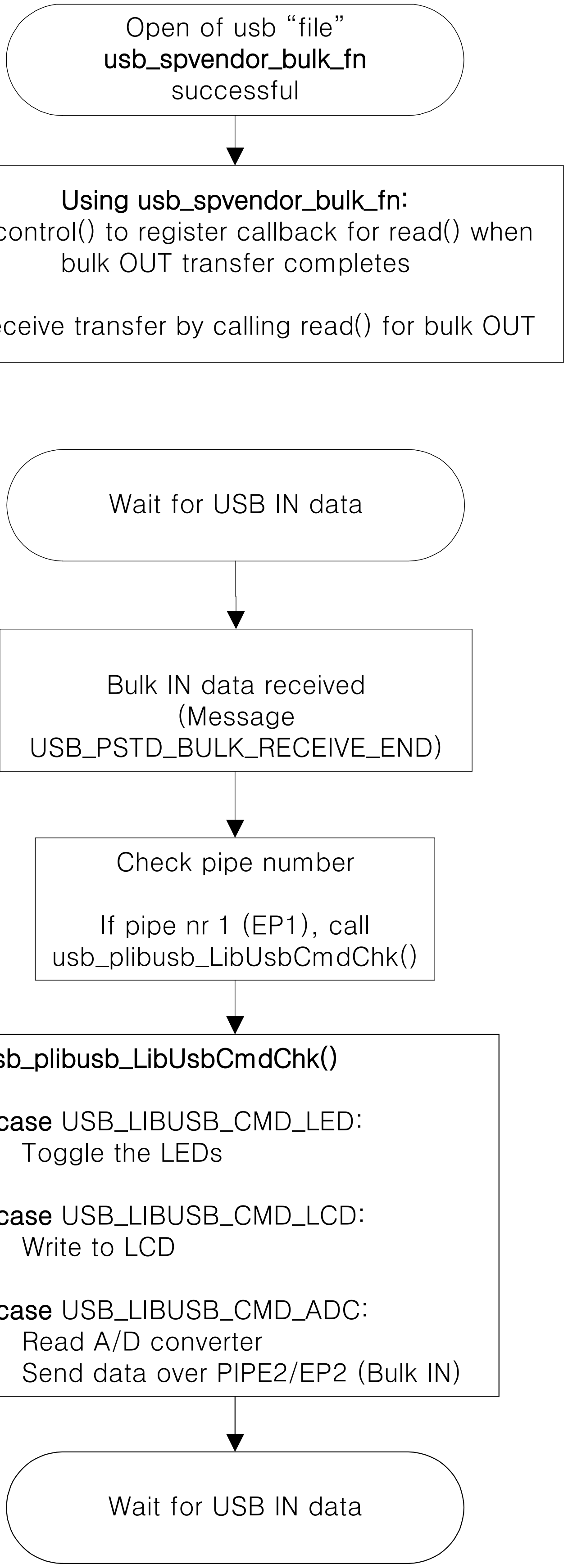


Figure 7. LibUSB application level processing flow. See file r_usb_vendor_papl.c, procedure usb_pvendor_apl_task.

5.7 Command Protocol Details

The LibUSB Demo has three application level commands; "Toggle LED," "Read ADC," and "Set LCD." The first byte in the host PIPE1 transfer contains the command number. The second and subsequent bytes are not used. See the table below for details. For more info, see file *..\src\SmplMain\APL\r_usb_vendor_papl.c*, function *usb_plibusb_LibUsbCmdChk*.

5.7.1 Toggle LED

Host	Direction Host-Per.	LibUSB Application
First byte of host OUT transfer = Command byte: 0x01	→	Received in PIPE1.
		Command processing Turns off the LED if it is currently on. Turns on the LED if it is currently off. Only the first byte is recognized as a command. The 2nd and subsequent bytes are not used.

5.7.2 Read ADC

Host		LibUSB Application
First byte of host OUT transfer = Command byte: 0x02	→	Received in PIPE1.
		Command processing Reads AD data. Only the first byte is recognized as a command. The 2nd and subsequent bytes are not used.
	←	Returns AD data (2-4 bytes). The first byte has the following meaning: 0: OK. 1: NOT OK. The 3 bytes starting at the 2nd byte contain the AD value in little endian.

5.7.3 Set LCD

Host		LibUSB Application
First byte of host OUT transfer = Command byte: 0x04	→	Received in PIPE1.
Seconded byte = Line number of LCD: This can zero (0x00) or one (0x01) A larger number clears the display		Command processing

6. 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

		Description	
Rev.	Date	Page	Summary
1.00	Mar 07, 2011	—	First edition issued
2.00	Sep 30, 2012	All	For USBv2.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU 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 accord 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 an MPU or MCU 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.

   Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be 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, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

RENESAS

Renesas Electronics Corporation

http://www.renesas.com

SALES OFFICES

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-651-700, Fax: +44-1628-651-804

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, 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 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

© 2012 Renesas Electronics Corporation. All rights reserved.

Colophon 2.2