

USB Device Low Level Driver for Renesas 20 User Guide

Version 1.10

For use with USB Device Low Level Driver for Renesas 20
versions 2.09 and above

Date: 16-Jun-2017 17:22

All rights reserved. This document and the associated software are the sole property of HCC Embedded. Reproduction or duplication by any means of any portion of this document without the prior written consent of HCC Embedded is expressly forbidden.

HCC Embedded reserves the right to make changes to this document and to the related software at any time and without notice. The information in this document has been carefully checked for its accuracy; however, HCC Embedded makes no warranty relating to the correctness of this document.

Table of Contents

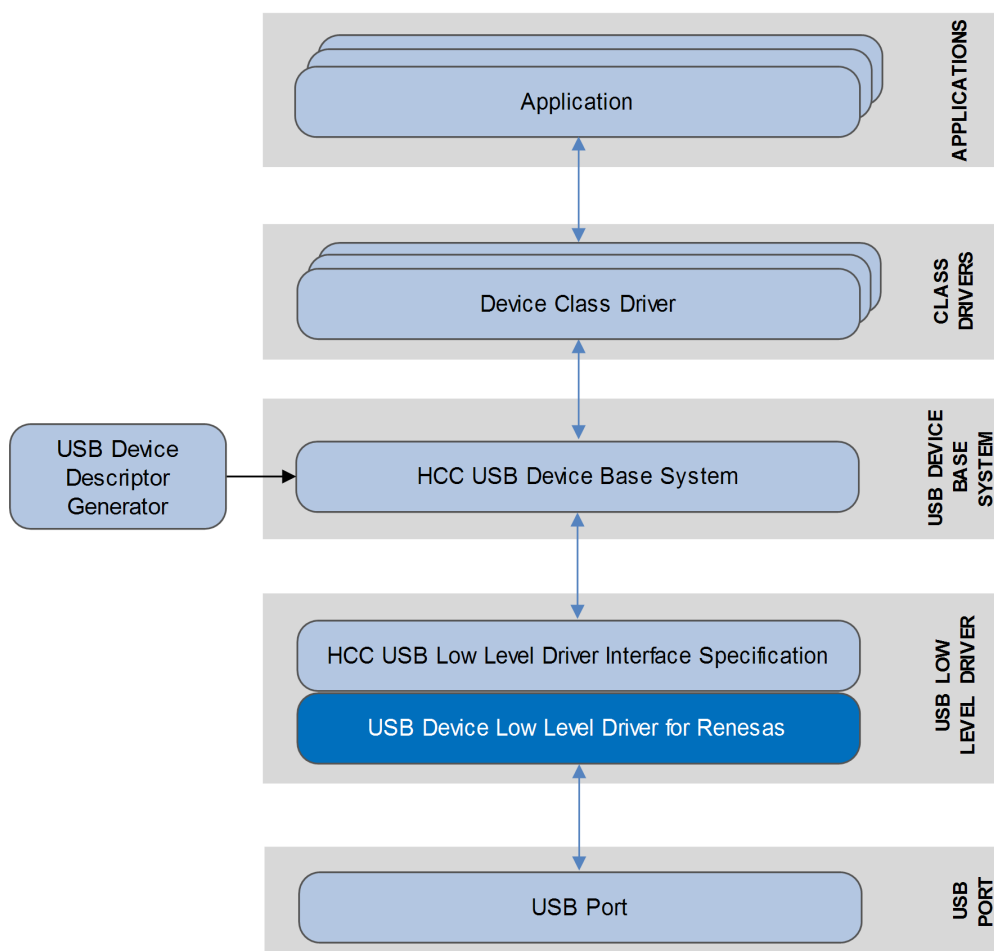
System Overview	3
Introduction	3
Feature Check	4
Packages and Documents	5
Packages	5
Documents	5
Change History	6
Source File List	7
Configuration File	7
Source Code	7
Version File	7
Platform Support Package (PSP) Files	8
Configuration Options	9
Integration	10
OS Abstraction Layer	10
PSP Porting	10
psp_usbd_hw_init	11
psp_usbd_hw_start	12
psp_usbd_hw_stop	13
psp_usbd_hw_delete	14

1 System Overview

1.1 Introduction

This guide is for those who want to configure and use the HCC Embedded Low Level Driver for Renesas 20 module with HCC's USB device stack. This module provides a USB device driver for Renesas processors with USB 2.0 controllers (the SH and RX families). The driver can handle all USB transfer types and, in conjunction with the USB device stack, can be used with any USB device class driver.

This package provides a low level driver for a USB stack, as shown below.



The low level driver is always started automatically by the USB device stack. The driver is linked to the stack at compile time because each low level driver uses the same function names. This also means that only one driver can run in a system.

1.2 Feature Check

The main features of the low level driver are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Conforms to HCC's USB Device Low Level Driver Specification.
- Integrated with the HCC USB device stack and all its class drivers.
- Supports all Renesas 20/SH-2A controllers.
- Supports all USB transfer types: control, bulk, interrupt, and isochronous.

1.3 Packages and Documents

Packages

The table below lists the packages that you need in order to use this module:

Package	Description
<code>hcc_base_doc</code>	This contains the two guides that will help you get started.
<code>usbd_base</code>	The USB device base package. Its source code includes the USB Driver device core.
<code>usbd_drv_renesas20</code>	The Renesas 20 low level driver package described by this document.

Documents

For an overview of HCC's embedded USB stacks, see [Product Information](#) on the main HCC website.

Readers should note the points in the [HCC Documentation Guidelines](#) on the HCC documentation website.

HCC Firmware Quick Start Guide

This document describes how to install packages provided by HCC in the target development environment. Also follow the *Quick Start Guide* when HCC provides package updates.

HCC Source Tree Guide

This document describes the HCC source tree. It gives an overview of the system to make clear the logic behind its organization.

HCC Embedded USB Device Base System User Guide

This document defines the USB device base system upon which the complete USB stack is built.

HCC USB Device Low Level Driver for Renesas 20 User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: USB Device Low Level Driver for Renesas 20 User Guide](#).
- For the history of changes made to the package code itself, see [History: usbd_drv_renesas](#).

The current version of this manual is 1.10. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.10	2017-06-16	2.09	New <i>Change History</i> format.
1.00	2015-04-24	2.09	First release.

2 Source File List

This section describes all the source code files included in the system. These files follow the HCC Embedded standard source tree system, described in the [HCC Source Tree Guide](#). All references to file pathnames refer to locations within this standard source tree, not within the package you initially receive.

Note: Do not modify any of these files except the configuration file and PSP files.

2.1 Configuration File

The file `src/config/config_usbd_renesas20.h` contains all the configurable parameters. Configure these as required. For details of these options, see [Configuration Options](#).

2.2 Source Code

These files in the directory `src/usb-device/usb-drivers` are the source code files. **These files should only be modified by HCC.**

File	Description
<code>usbd_dev.h</code>	USB driver-specific header file.
<code>usbd_hw_renesas20.h</code>	Header file for functions.
<code>usbd_hw_renesas20.c</code>	Source code.

2.3 Version File

The file `src/version/ver_usbd_renesas20.h` contains the version number of this module. This version number is checked by all modules that use this module to ensure system consistency over upgrades.

2.4 Platform Support Package (PSP) Files

There are two sets of files, in directories named **src/psp_rx63/target** and **src/psp_sh7264/target**. These provide functions and elements the core code may need to use, depending on the hardware.

Note: These are PSP implementations for the specific microcontroller and development board; you may need to modify these to work with a different microcontroller and/or board. See [PSP Porting](#) for details.

The files are as follows:

File	Description
<code>include/hcc_xxxx_regs.h</code>	Register definitions.
<code>usb-renesas20/psp_usb-renesas20.c</code>	Function source code.
<code>usb-renesas20/psp_usb-renesas20.h</code>	Function header file.

3 Configuration Options

Set the system configuration options in the file `src/config/config_usbd_renesas20.h`. This section lists the available configuration options and their default values.

NO_OF_HW_EP

The number of hardware endpoints on the device, including EP0. The default is 9.

Endpoint and Pipe pairs

Note that only the following combinations are valid:

- Pipe1, Pipe2 – for Bulk or Isochronous endpoints, double buffering is possible.
- Pipe3, Pipe4, Pipe5 – for Bulk endpoints, double buffering is possible.
- Pipe6, Pipe7, Pipe8, Pipe9 – for Interrupt endpoints; no double buffering is possible.

The defaults are as follows:

```
#define USE_PIPE_FOR_EP1 6
#define USE_PIPE_FOR_EP2 1
#define USE_PIPE_FOR_EP3 2
#define USE_PIPE_FOR_EP4 3
#define USE_PIPE_FOR_EP5 4
#define USE_PIPE_FOR_EP6 USBD_EP_TYPE_NOTUSED
#define USE_PIPE_FOR_EP7 USBD_EP_TYPE_NOTUSED
#define USE_PIPE_FOR_EP8 USBD_EP_TYPE_NOTUSED
```

USB_D_RX_IT_PRIO

The RX priority. The default is 1.

USB_D_HIGH_SPEED

Set this to 1 if the USB controller is a high speed controller. The default is 0.

USB_D_PORT_USED

Set this to 1 if the port is used. The default is 0.

USB_D_RENESAS_BIGEND_FIFO_ACCESS

Set this to 1 if big-endian FIFO port control is needed. The default is 0.

4 Integration

This section specifies the elements of this package that need porting, depending on the target environment.

4.1 OS Abstraction Layer

All HCC modules use the OS Abstraction Layer (OAL) that allows the module to run seamlessly with a wide variety of RTOSes, or without an RTOS.

This module requires the following OAL elements:

OAL Resource	Number Required
Tasks	0
Mutexes	0
Events	1
ISRs	1

4.2 PSP Porting

The Platform Support Package (PSP) is designed to hold all platform-specific functionality, either because it relies on specific features of a target system, or because this provides the most efficient or flexible solution for the developer.

The module makes use of the following PSP functions, provided by the PSP to perform particular tasks. Their design makes it easy for you to port them to work with your hardware solution. The package includes samples for the RX63 and SH7264 families in the **psp_usbd_renesas20.c** files.

Function	Description
psp_usbd_hw_init()	Initializes the device.
psp_usbd_hw_start()	Starts the device.
psp_usbd_hw_stop()	Stops the device.
psp_usbd_hw_delete()	Deletes the device, releasing the associated resources.

These are described in the sections which follow.

psp_usbd_hw_init

This function is provided by the PSP to initialize the device.

Note: Call this function first.

Format

```
int psp_usbd_hw_init ( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.

psp_usbd_hw_start

This function is provided by the PSP to start the device.

Note: Call `psp_usbd_hw_init()` before this.

Format

```
int psp_usbd_hw_start ( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.

psp_usbd_hw_stop

This function is provided by the PSP to stop the device.

Format

```
int psp_usbd_hw_stop ( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.

psp_usbd_hw_delete

This function is provided by the PSP to delete the device, releasing the associated resources.

Format

```
int psp_usbd_hw_delete( void )
```

Arguments

None.

Return Values

Return value	Description
USB_SUCCESS	Successful execution.
USB_ERROR	Operation failed.