

# USB 3.0 Host Controller Driver

ISG-NK1-100024

Rev.0.0.1

Sep 9, 2011

## Linux USB3 driver porting manual

### Introduction

This manual is intended for engineers who wish to utilize the RENESAS USB3.0 host controllers,  $\mu$ PD72020x, in Linux environment.

### Target Device

USB3.0 Host Controller  $\mu$ PD720200,  $\mu$ PD720200A,  $\mu$ PD720201, and  $\mu$ PD720202

### Contents

1. Overview .....	2
2. Preparation .....	2
3. Port and modify driver .....	3
4. Kernel compile and install .....	13
5. Appendix: Function and Structure list .....	14

## 1. Overview

USB3.0 drivers for Renesas USB3 host controllers are supported natively in Linux kernel 2.6.31 or later. In using these drivers in old kernel, back porting is required. This document describes how to port USB3.0 driver to old kernel, using the example of porting the drivers from kernel 2.6.32 to kernel 2.6.20 & 2.6.28.

**Note:** Additional modification may be required when USB3.0 driver is newer than 2.6.32,

## 2. Preparation

### 2.1 Getting Linux kernel

- 1) Download the target kernel from Linux.org (<http://www.linux.org/dist/kernel.html>).
- 2) Put target kernel on the following directory.  
/user/src
- 3) Extract the compressed file with the following command. When it completes, all files including source code is placed in the new directory. In this case, all files are extracted on the new directory, "linux-2.6.32".

<pre># tar xvfj [Target kernel zip file] ex) # tar xvfj linux-2.6.32.tar.bz2</pre>
--

### 2.2 Location of USB3.0 driver source code

If you download the kernel later than 2.6.31, source code of USB 3.0 drivers is included in the following directory.

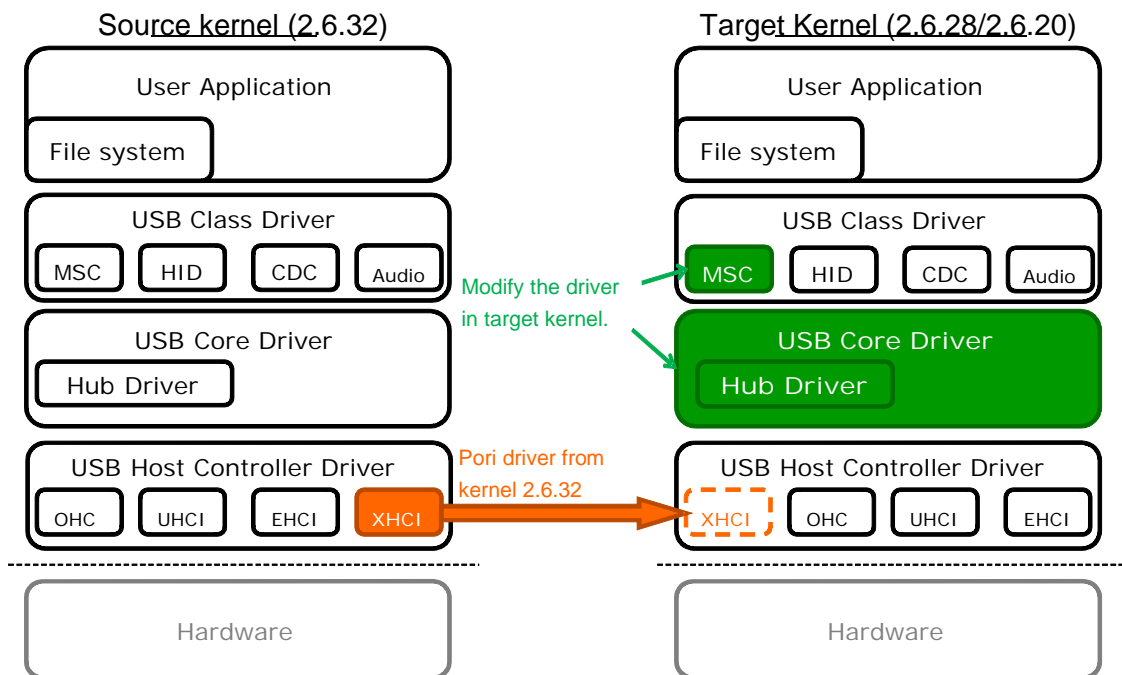
linux-2.6.32/drivers/usb/host

### 3. Port and modify driver

Just copying USB3.0 driver to old kernel (in this case, 2.6.20 or 2.6.28) will cause compile errors. To clean up these errors, the following modification is required.

- USB Core driver
- Structure like URB (USB Request Block) which USB driver stack handles

Fig1. Porting image



#### MEMO The changes in kernel 2.6.24

There were major changes in kernel 2.6.24.

- New function regarding Scatter/Gather DMA is supported
- Endpoint information is added in URB.

Especially, the changes in URB will cause errors in passing argument.

In porting USB3 driver to the kernel older than 2.6.24, it is required to care about these changes.

### 3.1 Porting to kernel 2.6.28

#### 3.1.1 Porting USB3.0 driver

It is possible to use USB3.0 driver from kernel 2.6.32. Just copy the source code to the following directory.

[Source directory (2.6.32)]

linux-2.6.32/drivers/usb/host

[Files to be copied]

pci-quirks.c  
 xhci-dbg.c  
 xhci-ext-caps.h  
 xhci-hcd.c  
 xhci-hub.c  
 xhci-mem.c  
 xhci-pci.c  
 xhci-ring.c  
 xhci.h

[Target directory (2.6.28)]

linux-2.6.28/drivers/usb/host

#### 3.1.2 Modifying USB core driver

It is necessary to modify USB core driver in kernel 2.6.28 as in below table. For more detail, please refer to source code.

[Directory]      linux-2.6.28/drivers/usb/core

File name	Modification
config.c	<p>◆ <b>Add new function</b>            find_next_descriptor_more()            usb_parse_ss_endpoint_companion()</p> <p>◆ <b>Modify function</b>            usb_parse_endpoint()</p>
devio.c	<p>◆ <b>Modify function</b>            proc_resetep()</p>
hcd.c	<p>◆ <b>Add new array</b>            static const u8 usb3_rh_dev_descriptor[18]            static const u8 ss_rh_config_descriptor[]</p> <p>◆ <b>Add new function</b>            usb_hcd_check_bandwidth()            usb_hcd_reset_endpoint()</p> <p>◆ <b>Modify function</b>            rh_call_control()            usb_add_hcd()</p>
hub.c	<p>◆ <b>Modify function</b>            static inline char *portspeed()            kick_khubd()            usb_kick_khubd()            hub_activate()            hub_configure()            usb_ep0_reinit()            hub_set_address()            hub_port_init()            hub_port_connect_change()</p>
message.c	<p>◆ <b>Add new function</b></p>

File name	Modification
	usb_reset_endpoint() <b>◆ Modify function</b> usb_sg_init() usb_clear_halt () usb_enable_endpoint() usb_enable_interface() usb_set_interface() usb_reset_configuration() usb_set_configuration()
urb.c	<b>◆ Modify function</b> usb_submit_urb()
usb.c	<b>◆ Add include file</b> #include <linux/debugfs.h> <b>◆ Modify function</b> usb_release_dev() usb_dev_complete() usb_dev_resume() usb_dev_thaw() usb_dev_restore() struct usb_device *usb_alloc_dev()
hcd.h	<b>◆ Modify structure</b> hc_driver{} <b>◆ Add new function declaration</b> extern void usb_hcd_reset_endpoint() extern int usb_hcd_check_bandwidth()
hub.h	<b>◆ Add new define</b> #define USB_PORT_FEAT_SUPERSPEED 11
usb.h	<b>◆ Add new function declaration</b> extern void usb_enable_interface() <b>◆ Modify declaration</b> extern void usb_enable_endpoint()

### 3.1.3 Modifying USB storage class driver

It is necessary to modify USB storage class driver in kernel 2.6.28 as in below table. For more detail, please refer to source code.

[Directory]      **linux-2.6.28/drivers/usb/storage**

File name	Modification
transport.c	<p>◆ <b>Modify function</b></p> <p>usb_stor_clear_halt()</p>

### 3.1.4 Modifying header file

The following modification is required. Please refer to the source code.

[Directory]      **linux-2.6.28/include/linux**

File name	Modification
pci_ids.h	<p>◆ <b>Add new define</b></p> <p>#define PCI_CLASS_SERIAL_USB_XHCI      0x0c0330</p>
usb.h	<p>◆ <b>Add new structure</b></p> <p>struct usb_host_ss_ep_comp{} //Companion descriptor for SS device</p> <p>◆ <b>Add member in structure</b></p> <p>struct usb_host_endpoint{     struct usb_host_ss_ep_comp *ss_ep_comp;    /* For SS devices */</p> <p>struct usb_device{     u32 route;     int slot_id;</p> <p>struct urb{     struct usb_sg_request *sg;    /* (in) scatter gather buffer list */     int num_sgs;                  /* (in) number of entries in the sg list */</p> <p>◆ <b>Add new function declaration</b></p> <p>extern void usb_reset_endpoint()</p>

[Directory]      **linux-2.6.28/include/linux/usb**

File name	Modification
ch9.h	<p>◆ <b>Add new define</b></p> <ul style="list-style-type: none"> <li>• Descriptor type</li> </ul> <p>#define USB_DT_SS_ENDPOINT_COMP      0x30</p> <ul style="list-style-type: none"> <li>• Device/Interface class code</li> </ul> <p>#define USB_SUBCLASS_VENDOR_SPEC      0xff</p> <ul style="list-style-type: none"> <li>• Endpoint</li> </ul> <p>#define USB_ENDPOINT_SYNCTYPE      0x0c  #define USB_ENDPOINT_SYNC_NONE      (0 &lt;&lt; 2)  #define USB_ENDPOINT_SYNC_ASYNC      (1 &lt;&lt; 2)  #define USB_ENDPOINT_SYNC_ADAPTIVE      (2 &lt;&lt; 2)  #define USB_ENDPOINT_SYNC_SYNC      (3 &lt;&lt; 2)</p> <p>◆ <b>Add new structure</b></p> <p>struct usb_ss_ep_comp_descriptor{</p> <p>◆ <b>Modify structure</b></p> <p>struct usb_qualifier_descriptor{     USB_SPEED_SUPER,                  /* usb 3.0 */</p>

### 3.1.5 Modifying Makefile, Kconfig

Before compiling kernel, it is necessary to modify Makefile and Kconfig, shown in red ink.

**[Directory]**      **linux-2.6.28/drivers/usb/host**  
**[File name]**      **Makefile**

```
#
# Makefile for USB Host Controller Drivers
#

ifeq ($(CONFIG_USB_DEBUG),y)
    EXTRA_CFLAGS      += -DDEBUG
endif

isp1760-objs := isp1760-hcd.o isp1760-if.o

xhci-objs := xhci-hcd.o xhci-mem.o xhci-pci.o xhci-ring.o xhci-hub.o xhci-dbg.o

obj-$(CONFIG_USB_WHCI_HCD)  += whci/

obj-$(CONFIG_PCI)           += pci-quirks.o

obj-$(CONFIG_USB_EHCI_HCD)  += ehci-hcd.o
obj-$(CONFIG_USB_ISP116X_HCD) += isp116x-hcd.o
obj-$(CONFIG_USB_OHCI_HCD)  += ohci-hcd.o
obj-$(CONFIG_USB_UHCI_HCD)  += uhci-hcd.o
obj-$(CONFIG_USB_SL811_HCD) += sl811-hcd.o
obj-$(CONFIG_USB_SL811_CS)  += sl811_cs.o
obj-$(CONFIG_USB_U132_HCD)  += u132-hcd.o
obj-$(CONFIG_USB_R8A66597_HCD) += r8a66597-hcd.o
obj-$(CONFIG_USB_ISP1760_HCD) += isp1760.o
obj-$(CONFIG_USB_HWA_HCD)   += hwa-hc.o
obj-$(CONFIG_USB_XHCI_HCD) += xhci.o
```

**[Directory]**      **linux-2.6.28/drivers/usb/host**  
**[File name]**      **Kconfig**

```
#
# USB Host Controller Drivers
#
comment "USB Host Controller Drivers"
    depends on USB

config USB_XHCI_HCD
    tristate "xHCI HCD (USB 3.0) support (EXPERIMENTAL)"
    depends on USB && PCI && EXPERIMENTAL
    ---help---
    The eXtensible Host Controller Interface (xHCI) is standard for USB 3.0
    "SuperSpeed" host controller hardware.

    To compile this driver as a module, choose M here: the
    module will be called xhci-hcd.

config USB_XHCI_HCD_DEBUGGING
    bool "Debugging for the xHCI host controller"
    depends on USB_XHCI_HCD
    ---help---
    Say 'Y' to turn on debugging for the xHCI host controller driver.
    This will spew debugging output, even in interrupt context.
    This should only be used for debugging xHCI driver bugs.

    If unsure, say N.

config USB_C67X00_HCD
    tristate "Cypress C67x00 HCD support"
    depends on USB
    help
```

[Directory]      linux-2.6.28/drivers/usb/core  
[File name]     Makefile

```
#  
# Makefile for the kernel USB device drivers.  
#  
  
# Object files in subdirectories  
  
obj-$(CONFIG_USB)                += core/  
  
obj-$(CONFIG_USB_MON)           += mon/  
  
obj-$(CONFIG_PCI)                += host/  
obj-$(CONFIG_USB_EHCI_HCD)      += host/  
obj-$(CONFIG_USB_ISP116X_HCD)   += host/  
obj-$(CONFIG_USB_OHCI_HCD)      += host/  
obj-$(CONFIG_USB_UHCI_HCD)      += host/  
obj-$(CONFIG_USB_SL811_HCD)     += host/  
obj-$(CONFIG_USB_U132_HCD)      += host/  
obj-$(CONFIG_USB_R8A66597_HCD)   += host/  
obj-$(CONFIG_USB_HWA_HCD)       += host/  
obj-$(CONFIG_USB_XHCI_HCD)      += host/  
obj-$(CONFIG_USB_C67X00_HCD) += c67x00/
```



## 3.2 Porting to kernel 2.6.20

### 3.2.1 Porting and modifying USB3.0 driver

Different from porting to kernel 2.6.28, it is necessary to modify USB3.0 driver in porting to kernel 2.6.20, because there were differences in URB structure, argument passed from core driver, and function regarding scatter/gather.

Please modify the source code of kernel 2.6.32 as in below and copy them to “linux-2.6.20/drivers/usb/host” in same way as chapter [3.1.1](#).

File name	Modification
xhci-hcd.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> xhci_urb_enqueue() xhci_urb_dequeue()</li> </ul>
xhci-pci.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> xhci_pci_setup()</li> </ul>
xhci-ring.c	<ul style="list-style-type: none"> <li>◆ <b>Add include file</b> #include &lt;asm/types.h&gt;</li> <li>◆ <b>Modify function</b> handle_stopped_endpoint() handle_tx_event() prepare_transfer() count_sg_trbs_needed() queue_bulk_sg_tx()</li> </ul>
xhci.h	<ul style="list-style-type: none"> <li>◆ <b>Modify function declaration</b> int xhci_urb_enqueue() int xhci_urb_dequeue()</li> </ul>
xhci-dbg.c	No need to modify.
xhci-ext-caps.h	No need to modify.
xhci-hub.c	No need to modify.
xhci-mem.c	No need to modify.

The following modifications are also required for the files in kernel 2.6.20.

**[Directory]**      **linux-2.6.20/drivers/usb/host**

File name	Modification
pci-quirks.c	<ul style="list-style-type: none"> <li>◆ <b>Add new function</b> handshake() quirk_usb_handoff_xhci()</li> <li>◆ <b>Add include file</b> #include "xhci-ext-caps.h"</li> <li>◆ <b>Modify function</b> quirk_usb_early_handoff()</li> </ul>

### 3.2.2 Modifying USB core driver

It is necessary to modify USB core driver in kernel 2.6.20 as in below table. For more detail, please refer to source code.

[Directory]      **linux-2.6.20/drivers/usb/core**

File name	Modification
config.c	<ul style="list-style-type: none"> <li>◆ <b>Add new function</b> find_next_descriptor_more() usb_parse_ss_endpoint_companion()</li> <li>◆ <b>Modify function</b> usb_parse_endpoint()</li> </ul>
devio.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> proc_resetep()</li> </ul>
hcd.c	<ul style="list-style-type: none"> <li>◆ <b>Add new array</b> static const u8 usb3_rh_dev_descriptor[18] static const u8 ss_rh_config_descriptor[]</li> <li>◆ <b>Add new function</b> usb_hcd_check_bandwidth() usb_hcd_disable_endpoint() usb_hcd_reset_endpoint()</li> <li>◆ <b>Modify function</b> rh_call_control() usb_hcd_submit_urb() usb_add_hcd()</li> </ul>
hub.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> static inline char *portspeed() hub_configure() ep0_reinit() hub_set_address() hub_port_init() hub_port_connect_change()</li> </ul>
message.c	<ul style="list-style-type: none"> <li>◆ <b>Add new function</b> usb_reset_endpoint()</li> <li>◆ <b>Modify function</b> usb_sg_init() usb_clear_halt () usb_enable_endpoint() usb_enable_interface() usb_set_interface() usb_reset_configuration() usb_set_configuration()</li> </ul>
urb.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> usb_submit_urb()</li> </ul>
usb.c	<ul style="list-style-type: none"> <li>◆ <b>Add include file</b> #include &lt;linux/debugfs.h&gt;</li> <li>◆ <b>Modify function</b> usb_release_dev() struct usb_device *usb_alloc_dev()</li> </ul>
hcd.h	<ul style="list-style-type: none"> <li>◆ <b>Modify structure</b> hc_driver{}</li> <li>◆ <b>Add new function declaration</b> extern void usb_hcd_reset_endpoint() extern int usb_hcd_check_bandwidth()</li> </ul>

File name	Modification
hub.h	<ul style="list-style-type: none"> <li>◆ <b>Add new define</b> #define USB_PORT_FEAT_SUPERSPEED 11</li> </ul>
usb.h	<ul style="list-style-type: none"> <li>◆ <b>Add new function declaration</b> extern void usb_enable_endpoint()</li> </ul>

### 3.2.3 Modifying USB storage class driver

It is necessary to modify USB storage class driver in kernel 2.6.20 as in below table. For more detail, please refer to source code.

[Directory]      **linux-2.6.20/drivers/usb/storage**

File name	Modification
transport.c	<ul style="list-style-type: none"> <li>◆ <b>Modify function</b> usb_stor_clear_halt()</li> </ul>

### 3.2.4 Modifying header file

The following modification is required. Please refer to the source code.

[Directory]      **linux-2.6.20/include/linux**

File name	Modification
pci_ids.h	<ul style="list-style-type: none"> <li>◆ <b>Add new define</b> #define PCI_CLASS_SERIAL_USB_XHCI 0x0c0330</li> </ul>
usb.h	<ul style="list-style-type: none"> <li>◆ <b>Add new structure</b> struct usb_host_ss_ep_comp{} //Companion descriptor for SS device</li> <li>◆ <b>Add member in structure</b> struct usb_host_endpoint{}     struct usb_host_ss_ep_comp *ss_ep_comp; /* For SS devices */</li> <li>struct usb_device{}     u32 route;     int slot_id;</li> <li>struct urb{}     struct usb_host_endpoint *ep; /* (internal) pointer to endpoint */     struct usb_sg_request *sg; /* (in) scatter gather buffer list */     int num_sgs; /* (in) number of entries in the sg list */</li> <li>◆ <b>Add new function declaration</b> extern void usb_reset_endpoint()</li> <li>◆ <b>Add new function</b> static inline int usb_endpoint_num() static inline int usb_endpoint_xfer_control()</li> </ul>
usb_ch9.h	<ul style="list-style-type: none"> <li>◆ <b>Add new define</b> <ul style="list-style-type: none"> <li>• Descriptor type #define USB_DT_SS_ENDPOINT_COMP 0x30 #define USB_DT_SS_EP_COMP_SIZE 6</li> <li>• Device/Interface class code #define USB_SUBCLASS_VENDOR_SPEC 0xff</li> </ul> </li> <li>◆ <b>Add new structure</b> struct usb_ss_ep_comp_descriptor{}  <ul style="list-style-type: none"> <li>◆ <b>Add member in structure</b> struct usb_qualifier_descriptor{}     USB_SPEED_SUPER, /* usb 3.0 */</li> </ul> </li> </ul>

File name	Modification
kernel.h	<b>◆ Add new define</b> #define upper_32_bits(n) ((u32)(((n) >> 16) >> 16)) #define lower_32_bits(n) ((u32)(n))

### 3.2.5 Modifying Makefile, Kconfig

Same modifications are required as in section [3.1.5](#).

## 4. Kernel compile and install

After porting and modifying USB3.0 driver, next steps are to make configuration, compile, and install. This example uses PC with Fedora Linux.

### 4.1 Initialize the environment

```
# make mrproper
```

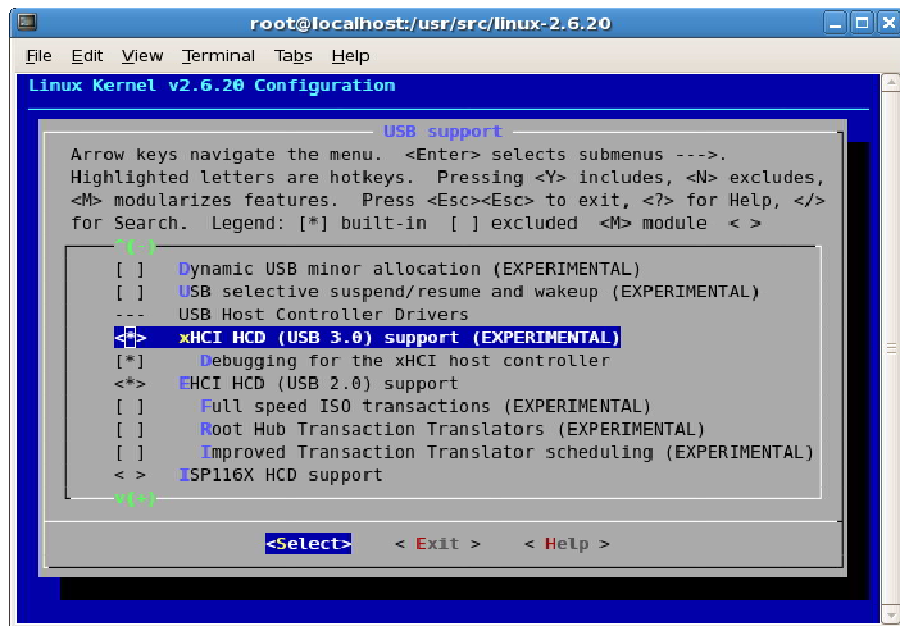
### 4.2 Make configuration

With the following command, you can configure the system.

```
# make menuconfig
```

To enable USB 3.0 driver, please select [Device drivers] -> [USB support] and choose 'y' or 'm' for [xHCI HCD (USB3.0) support]. To enable debug message of USB3.0 driver, please also choose 'y' for [Debugging for the xHCI ~], but this may affect USB transfer speed. If you need to test transfer speed, it is recommended to disable all debug function in [Kernel hacking] -> [Kernel Debugging].

Fig2. Kernel configuration menu



### 4.3 Kernel compile

After making configuration, you can compile with the following step.

```
# make dep
# make clean
# make
```

### 4.4 Install

After making configuration, you can compile with the following step.

```
# make modules_install
# make install
```

## 5. Appendix: Function and Structure list

After porting and modifying USB3.0 driver, next steps are to make configuration, compile, and install. This example uses PC with Fedora Linux.

### 5.1 Core driver (drivers/usb/core)

#### 5.1.1 config.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
plural	plural	plural
find_next_descriptor_more	-	-
find_next_descriptor	find_next_descriptor	find_next_descriptor
usb_parse_ss_endpoint_companion	-	-
usb_parse_endpoint	usb_parse_endpoint	usb_parse_endpoint
usb_release_interface_cache	usb_release_interface_cache	usb_release_interface_cache
usb_parse_interface	usb_parse_interface	usb_parse_interface
usb_parse_configuration	usb_parse_configuration	usb_parse_configuration
usb_destroy_configuration	usb_destroy_configuration	usb_destroy_configuration
usb_get_configuration	usb_get_configuration	usb_get_configuration

## 5.1.2 hcd.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
usb_busmap	usb_busmap	usb_busmap
is_root_hub	is_root_hub	-
ascii2desc	ascii2utf	ascii2utf
rh_string	rh_string	rh_string
rh_call_control	rh_call_control	rh_call_control
usb_hcd_poll_rh_status	usb_hcd_poll_rh_status	usb_hcd_poll_rh_status
rh_timer_func	rh_timer_func	rh_timer_func
rh_queue_status	rh_queue_status	rh_queue_status
rh_urb_enqueue	rh_urb_enqueue	rh_urb_enqueue
usb_rh_urb_dequeue	usb_rh_urb_dequeue	usb_rh_urb_dequeue
usb_host_authorized_default_show	usb_host_authorized_default_show	-
usb_host_authorized_default_store	usb_host_authorized_default_store	-
usb_bus_init	usb_host_init	usb_host_init
-	usb_host_cleanup	usb_host_cleanup
-	usb_bus_init	usb_bus_init
usb_register_bus	usb_register_bus	usb_register_bus
usb_deregister_bus	usb_deregister_bus	usb_deregister_bus
register_root_hub	register_root_hub	register_root_hub
usb_calc_bus_time	usb_calc_bus_time	usb_calc_bus_time
usb_hcd_link_urb_to_ep	usb_hcd_link_urb_to_ep	-
usb_hcd_check_unlink_urb	usb_hcd_check_unlink_urb	-
usb_hcd_unlink_urb_from_ep	usb_hcd_unlink_urb_from_ep	urb_unlink
hcd_alloc_coherent	hcd_alloc_coherent	usb_claim_bandwidth
hcd_free_coherent	hcd_free_coherent	usb_release_bandwidth
map_urb_for_dma	map_urb_for_dma	-
unmap_urb_for_dma	unmap_urb_for_dma	-
usb_hcd_submit_urb	usb_hcd_submit_urb	usb_hcd_submit_urb
unlink1	unlink1	unlink1
usb_hcd_unlink_urb	usb_hcd_unlink_urb	usb_hcd_unlink_urb
usb_hcd_giveback_urb	usb_hcd_giveback_urb	usb_hcd_giveback_urb
usb_hcd_flush_endpoint	usb_hcd_flush_endpoint	-
usb_hcd_check_bandwidth	-	usb_check_bandwidth
usb_hcd_disable_endpoint	usb_hcd_disable_endpoint	usb_hcd_endpoint_disable
usb_hcd_reset_endpoint	-	-
usb_hcd_synchronize_unlinks	usb_hcd_synchronize_unlinks	-
usb_hcd_get_frame_number	usb_hcd_get_frame_number	usb_hcd_get_frame_number
hcd_bus_suspend	hcd_bus_suspend	hcd_bus_suspend
hcd_bus_resume	hcd_bus_resume	hcd_bus_resume
hcd_resume_work	hcd_resume_work	-
usb_hcd_resume_root_hub	usb_hcd_resume_root_hub	usb_hcd_resume_root_hub
usb_bus_start_enum	usb_bus_start_enum	usb_bus_start_enum
usb_hcd_irq	usb_hcd_irq	usb_hcd_irq
usb_hc_died	usb_hc_died	usb_hc_died
usb_create_hcd	usb_create_hcd	usb_create_hcd
hcd_release	hcd_release	hcd_release
usb_get_hcd	usb_get_hcd	usb_get_hcd
usb_put_hcd	usb_put_hcd	usb_put_hcd
usb_add_hcd	usb_add_hcd	usb_add_hcd
usb_remove_hcd	usb_remove_hcd	usb_remove_hcd
usb_hcd_platform_shutdown	usb_hcd_platform_shutdown	usb_hcd_platform_shutdown
usb_mon_register	usb_mon_register	usb_mon_register
usb_mon_deregister	usb_mon_deregister	usb_mon_deregister
-	-	usb_enable_root_hub_irq

## 5.1.3 hub.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
usb_hub	usb_hub	usb_hub
usb_hub::char	usb_hub::char	usb_hub::char
portspeed	portspeed	portspeed
hdev_to_hub	hdev_to_hub	hdev_to_hub
get_hub_descriptor	get_hub_descriptor	get_hub_descriptor
clear_hub_feature	clear_hub_feature	clear_hub_feature
clear_port_feature	clear_port_feature	clear_port_feature
set_port_feature	set_port_feature	set_port_feature
set_port_led	set_port_led	set_port_led
led_work	led_work	led_work
get_hub_status	get_hub_status	get_hub_status
get_port_status	get_port_status	get_port_status
hub_port_status	hub_port_status	hub_port_status
kick_khubd	kick_khubd	kick_khubd
usb_kick_khubd	usb_kick_khubd	usb_kick_khubd
hub_irq	hub_irq	hub_irq
hub_clear_tt_buffer	hub_clear_tt_buffer	hub_clear_tt_buffer
hub_tt_work	hub_tt_kevent	hub_tt_kevent
usb_hub_clear_tt_buffer	usb_hub_tt_clear_buffer	usb_hub_tt_clear_buffer
hub_power_on	hub_power_on	hub_power_on
hub_hub_status	hub_hub_status	hub_hub_status
hub_port_disable	hub_port_disable	hub_port_disable
hub_port_logical_disconnect	hub_port_logical_disconnect	hub_port_logical_disconnect
hub_activation_type	hub_activation_type	-
hub_activate	hub_activate	hub_activate
hub_init_func2	hub_init_func2	-
hub_init_func3	hub_init_func3	-
hub_quiescing_type	hub_quiescing_type	-
hub_quiesce	hub_quiesce	hub_quiesce
hub_pre_reset	hub_pre_reset	hub_pre_reset
hub_post_reset	hub_post_reset	hub_post_reset
hub_configure	hub_configure	hub_configure
hub_release	hub_release	-
hub_disconnect	hub_disconnect	hub_disconnect
hub_probe	hub_probe	hub_probe
hub_ioctl	hub_ioctl	hub_ioctl
find_port_owner	-	locktree
usb_hub_claim_port	-	-
usb_hub_release_port	-	-
usb_hub_release_all_ports	-	-
usb_device_is_ow ned	-	-
recursively_mark_NOTATTACHED	recursively_mark_NOTATTACHED	recursively_mark_NOTATTACHED
usb_set_device_state	usb_set_device_state	usb_set_device_state
choose_address	choose_address	choose_address
release_address	release_address	release_address
update_address	update_address	-
usb_stop_pm	usb_stop_pm	-
usb_disconnect	usb_disconnect	usb_disconnect
show_string	show_string	show_string
announce_device	announce_device	-
usb_configure_device_otg	usb_configure_device_otg	-
usb_configure_device	usb_configure_device	-
usb_new_device	usb_new_device	usb_new_device
usb_deauthorize_device	usb_deauthorize_device	-
usb_authorize_device	usb_authorize_device	-
hub_is_w usb	hub_is_w usb	hub_is_w usb
hub_port_wait_reset	hub_port_wait_reset	hub_port_wait_reset
hub_port_reset	hub_port_reset	hub_port_reset
check_port_resume_type	check_port_resume_type	-
usb_port_suspend	usb_port_suspend	usb_port_suspend
finish_port_resume	finish_port_resume	finish_port_resume
usb_port_resume	usb_port_resume	usb_port_resume
remote_wakeup	remote_wakeup	remote_wakeup
hub_suspend	hub_suspend	hub_suspend
hub_resume	hub_resume	hub_resume
hub_reset_resume	hub_reset_resume	-
usb_root_hub_lost_power	usb_root_hub_lost_power	usb_root_hub_lost_power
hub_port_debounce	hub_port_debounce	hub_port_debounce
usb_ep0_reinit	usb_ep0_reinit	ep0_reinit
hub_set_address	hub_set_address	hub_set_address
hub_port_init	hub_port_init	hub_port_init
check_highspeed	check_highspeed	check_highspeed
hub_power_remaining	hub_power_remaining	hub_power_remaining
hub_port_connect_change	hub_port_connect_change	hub_port_connect_change
hub_events	hub_events	hub_events
hub_thread	hub_thread	hub_thread
usb_hub_init	usb_hub_init	usb_hub_init
usb_hub_cleanup	usb_hub_cleanup	usb_hub_cleanup
descriptors_changed	descriptors_changed	config_descriptors_changed
usb_reset_and_verify_device	usb_reset_and_verify_device	-
usb_reset_device	usb_reset_device	usb_reset_device
usb_queue_reset_device	-	usb_reset_composite_device
-	-	hub_port_suspend
-	-	hub_port_resume
-	-	usb_resume_root_hub



## 5.1.4 message.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
api_context	api_context	-
usb_api_blocking_completion	usb_api_blocking_completion	usb_api_blocking_completion
usb_start_wait_urb	usb_start_wait_urb	usb_start_wait_urb
usb_internal_control_msg	usb_internal_control_msg	usb_internal_control_msg
usb_control_msg	usb_control_msg	usb_control_msg
usb_interrupt_msg	usb_interrupt_msg	usb_interrupt_msg
usb_bulk_msg	usb_bulk_msg	usb_bulk_msg
sg_clean	sg_clean	sg_clean
sg_complete	sg_complete	sg_complete
usb_sg_init	usb_sg_init	usb_sg_init
usb_sg_wait	usb_sg_wait	usb_sg_wait
usb_sg_cancel	usb_sg_cancel	usb_sg_cancel
usb_get_descriptor	usb_get_descriptor	usb_get_descriptor
usb_get_string	usb_get_string	usb_get_string
usb_try_string_workarounds	usb_try_string_workarounds	usb_try_string_workarounds
usb_string_sub	usb_string_sub	usb_string_sub
usb_get_langid	-	-
usb_string	usb_string	usb_string
usb_cache_string	usb_cache_string	usb_cache_string
usb_get_device_descriptor	usb_get_device_descriptor	usb_get_device_descriptor
usb_get_status	usb_get_status	usb_get_status
usb_clear_halt	usb_clear_halt	usb_clear_halt
create_intf_ep_devs	-	-
remove_intf_ep_devs	-	-
usb_disable_endpoint	usb_disable_endpoint	usb_disable_endpoint
usb_reset_endpoint	-	-
usb_disable_interface	usb_disable_interface	usb_disable_interface
usb_disable_device	usb_disable_device	usb_disable_device
usb_enable_endpoint	usb_enable_endpoint	usb_enable_endpoint
usb_enable_interface	usb_enable_interface	usb_enable_interface
usb_set_interface	usb_set_interface	usb_set_interface
usb_reset_configuration	usb_reset_configuration	usb_reset_configuration
usb_release_interface	usb_release_interface	release_interface
usb_if_uevent	usb_if_uevent	-
find_iad	find_iad	-
__usb_queue_reset_device	-	-
usb_set_configuration	usb_set_configuration	usb_set_configuration
set_config_request	set_config_request	set_config_request
driver_set_config_work	driver_set_config_work	driver_set_config_work
cancel_async_set_config	-	-
usb_driver_set_configuration	usb_driver_set_configuration	usb_driver_set_configuration

## 5.1.5 urb.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
urb_destroy	urb_destroy	urb_destroy
usb_init_urb	usb_init_urb	usb_init_urb
usb_alloc_urb	usb_alloc_urb	usb_alloc_urb
usb_free_urb	usb_free_urb	usb_free_urb
usb_get_urb	usb_get_urb	usb_get_urb
usb_anchor_urb	usb_anchor_urb	-
usb_unanchor_urb	usb_unanchor_urb	-
usb_submit_urb	usb_submit_urb	usb_submit_urb
usb_unlink_urb	usb_unlink_urb	usb_unlink_urb
usb_kill_urb	usb_kill_urb	usb_kill_urb
usb_poison_urb	usb_poison_urb	-
usb_unpoison_urb	usb_unpoison_urb	-
usb_kill_anchored_urbs	usb_kill_anchored_urbs	-
usb_poison_anchored_urbs	usb_poison_anchored_urbs	-
usb_unpoison_anchored_urbs	-	-
usb_unlink_anchored_urbs	usb_unlink_anchored_urbs	-
usb_wait_anchor_empty_timeout	usb_wait_anchor_empty_timeout	-
usb_get_from_anchor	usb_get_from_anchor	-
usb_scuttle_anchored_urbs	usb_scuttle_anchored_urbs	-
usb_anchor_empty	usb_anchor_empty	-

## 5.1.6 usb.c

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
usb_ifnum_to_if	usb_ifnum_to_if	usb_ifnum_to_if
usb_altnum_to_altsetting	usb_altnum_to_altsetting	usb_altnum_to_altsetting
find_interface_arg	find_interface_arg	find_interface_arg
__find_interface	__find_interface	__find_interface
usb_find_interface	usb_find_interface	usb_find_interface
usb_release_dev	usb_release_dev	usb_release_dev
usb_dev_uevent	usb_dev_uevent	-
ksuspend_usb_init	ksuspend_usb_init	ksuspend_usb_init
ksuspend_usb_cleanup	ksuspend_usb_cleanup	ksuspend_usb_cleanup
usb_dev_prepare	usb_dev_prepare	-
usb_dev_complete	usb_dev_complete	-
usb_dev_suspend	usb_dev_suspend	usb_autosuspend_work
usb_dev_resume	usb_dev_resume	-
usb_dev_freeze	usb_dev_freeze	-
usb_dev_thaw	usb_dev_thaw	-
usb_dev_poweroff	usb_dev_poweroff	-
usb_dev_restore	usb_dev_restore	-
usb_devnode	-	-
usb_bus_is_wusb	usb_bus_is_wusb	-
usb_alloc_dev	usb_alloc_dev	usb_alloc_dev
usb_get_dev	usb_get_dev	usb_get_dev
usb_put_dev	usb_put_dev	usb_put_dev
usb_get_intf	usb_get_intf	usb_get_intf
usb_put_intf	usb_put_intf	usb_put_intf
usb_lock_device_for_reset	usb_lock_device_for_reset	usb_lock_device_for_reset
match_device	match_device	match_device
usb_find_device	usb_find_device	usb_find_device
usb_get_current_frame_number	usb_get_current_frame_number	usb_get_current_frame_number
__usb_get_extra_descriptor	__usb_get_extra_descriptor	__usb_get_extra_descriptor
usb_buffer_alloc	usb_buffer_alloc	usb_buffer_alloc
usb_buffer_free	usb_buffer_free	usb_buffer_free
usb_buffer_map_sg	usb_buffer_map_sg	usb_buffer_map_sg
usb_buffer_unmap_sg	usb_buffer_unmap_sg	usb_buffer_unmap_sg
usb_disabled	usb_disabled	usb_disabled
usb_bus_notify	-	-
usb_debugfs_init	-	-
usb_debugfs_cleanup	-	-
usb_init	usb_init	usb_init
usb_exit	usb_exit	usb_exit

## 5.1.7 hc\_driver structure (hcd.h)

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
const char*description	const char*description	const char*description
const char*product_desc	const char*product_desc	const char*product_desc
size_thcd_priv_size	size_thcd_priv_size	size_thcd_priv_size
irqreturn_t(*irq) (struct usb_hcd *hcd)	irqreturn_t(*irq) (struct usb_hcd *hcd)	irqreturn_t(*irq) (struct usb_hcd *hcd)
int flags	int flags	int flags
#defineHCD_MEMORY0x0001/* HC regs use memory	#defineHCD_MEMORY0x0001/* HC regs use	#defineHCD_MEMORY0x0001/* HC regs use
#defineHCD_LOCAL_MEM0x0002/* HC needs local	#defineHCD_LOCAL_MEM0x0002/* HC needs	-
#defineHCD_USB110x0010/* USB 1.1 */	#defineHCD_USB110x0010/* USB 1.1 */	#defineHCD_USB110x0010/* USB 1.1 */
#defineHCD_USB20x0020/* USB 2.0 */	#defineHCD_USB20x0020/* USB 2.0 */	#defineHCD_USB20x0020/* USB 2.0 */
#defineHCD_USB30x0040/* USB 3.0 */	-	-
#defineHCD_MASK0x0070	-	-
int(*reset) (struct usb_hcd *hcd)	int(*reset) (struct usb_hcd *hcd)	int(*reset) (struct usb_hcd *hcd)
int(*start) (struct usb_hcd *hcd)	int(*start) (struct usb_hcd *hcd)	int(*start) (struct usb_hcd *hcd)
int(*pci_suspend)(struct usb_hcd *hcd)	int(*pci_suspend) (struct usb_hcd *hcd, pm_message_t message)	int(*suspend) (struct usb_hcd *hcd, pm_message_t message)
int(*pci_resume)(struct usb_hcd *hcd, bool hibernated)	int(*pci_resume) (struct usb_hcd *hcd)	int(*resume) (struct usb_hcd *hcd)
void(*stop) (struct usb_hcd *hcd)	void(*stop) (struct usb_hcd *hcd)	void(*stop) (struct usb_hcd *hcd)
void(*shutdown) (struct usb_hcd *hcd)	void(*shutdown) (struct usb_hcd *hcd)	void(*shutdown) (struct usb_hcd *hcd)
int(*get_frame_number) (struct usb_hcd *hcd)	int(*get_frame_number) (struct usb_hcd *hcd)	int(*get_frame_number) (struct usb_hcd *hcd)
int(*urb_enqueue)(struct usb_hcd *hcd, struct urb *urb, gfp_t mem_flags)	int(*urb_enqueue)(struct usb_hcd *hcd, struct urb *urb, gfp_t mem_flags)	int(*urb_enqueue) (struct usb_hcd *hcd, struct usb_host_endpoint *ep, struct urb *urb, gfp_t mem_flags)
int(*urb_dequeue)(struct usb_hcd *hcd, struct urb *urb, int status)	int(*urb_dequeue)(struct usb_hcd *hcd, struct urb *urb, int status)	int(*urb_dequeue) (struct usb_hcd *hcd, struct urb *urb)
void (*endpoint_disable)(struct usb_hcd *hcd, struct usb_host_endpoint *ep)	void (*endpoint_disable)(struct usb_hcd *hcd, struct usb_host_endpoint *ep)	void (*endpoint_disable)(struct usb_hcd *hcd, struct usb_host_endpoint *ep)
void (*endpoint_reset)(struct usb_hcd *hcd, struct usb_host_endpoint *ep)	-	-
int(*hub_status_data) (struct usb_hcd *hcd, char *buf)	int(*hub_status_data) (struct usb_hcd *hcd, char *buf)	int(*hub_status_data) (struct usb_hcd *hcd, char *buf)
int(*hub_control) (struct usb_hcd *hcd, u16 typeReq, u16 w Value, u16 w Index, char *buf, u16 w Length)	int(*hub_control) (struct usb_hcd *hcd, u16 typeReq, u16 w Value, u16 w Index, char *buf, u16 w Length)	int(*hub_control) (struct usb_hcd *hcd, u16 typeReq, u16 w Value, u16 w Index, char *buf, u16 w Length)
int(*bus_suspend)(struct usb_hcd *)	int(*bus_suspend)(struct usb_hcd *)	int(*bus_suspend)(struct usb_hcd *)
int(*bus_resume)(struct usb_hcd *)	int(*bus_resume)(struct usb_hcd *)	int(*bus_resume)(struct usb_hcd *)
int(*start_port_reset)(struct usb_hcd *, unsigned port_num)	int(*start_port_reset)(struct usb_hcd *, unsigned port_num)	int(*start_port_reset)(struct usb_hcd *, unsigned port_num)
void(*relinquish_port)(struct usb_hcd *, int)	void(*relinquish_port)(struct usb_hcd *, int)	-
int(*port_handed_over)(struct usb_hcd *, int)	int(*port_handed_over)(struct usb_hcd *, int)	-
-	-	void(*hub_irq_enable)(struct usb_hcd *)
void(*clear_tt_buffer_complete)(struct usb_hcd *, struct usb_host_endpoint *)	-	-
int(*alloc_dev)(struct usb_hcd *, struct usb_device *)	-	-
void(*free_dev)(struct usb_hcd *, struct usb_device *)	-	-
int (*add_endpoint)(struct usb_hcd *, struct usb_device *, struct usb_host_endpoint *)	-	-
int (*drop_endpoint)(struct usb_hcd *, struct usb_device *, struct usb_host_endpoint *)	-	-
int(*check_bandwidth)(struct usb_hcd *, struct usb_device *)	-	-
void(*reset_bandwidth)(struct usb_hcd *, struct usb_device *)	-	-
int(*address_device)(struct usb_hcd *, struct usb_device *udev)	-	-
int(*update_hub_device)(struct usb_hcd *, struct usb_device *hdev, struct usb_tt *tt, gfp_t mem_flags)	-	-

## 5.2 Header file (include/linux)

### 5.2.1 usb.h

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
usb_host_ss_ep_comp	-	-
usb_host_endpoint	usb_host_endpoint	usb_host_endpoint
usb_host_interface	usb_host_interface	usb_host_interface
usb_interface_condition	usb_interface_condition	usb_interface_condition
usb_interface	usb_interface	usb_interface
usb_get_intfdata	usb_get_intfdata	usb_get_intfdata
usb_set_intfdata	usb_set_intfdata	usb_set_intfdata
usb_interface_cache	usb_interface_cache	usb_interface_cache
usb_host_config	usb_host_config	usb_host_config
usb_devmap	usb_devmap	usb_devmap
usb_bus	usb_bus	usb_bus
usb_device	usb_device	usb_device
usb_autopm_enable	usb_autopm_enable	usb_autopm_enable
usb_autopm_disable	usb_autopm_disable	usb_autopm_disable
usb_mark_last_busy	usb_mark_last_busy	-
usb_make_path	usb_make_path	usb_make_path
usb_dynids	usb_dynids	usb_dynids
usb_dynid	usb_dynid	-
usbdrv_wrap	usbdrv_wrap	usbdrv_wrap
usb_driver	usb_driver	usb_driver
usb_device_driver	usb_device_driver	usb_device_driver
usb_class_driver	usb_class_driver	usb_class_driver
usb_register	usb_register	usb_register
usb_iso_packet_descriptor	usb_iso_packet_descriptor	usb_iso_packet_descriptor
usb_anchor	usb_anchor	-
init_usb_anchor	init_usb_anchor	-
usb_fill_control_urb	usb_fill_control_urb	usb_fill_control_urb
usb_fill_bulk_urb	usb_fill_bulk_urb	usb_fill_bulk_urb
usb_fill_int_urb	usb_fill_int_urb	usb_fill_int_urb
usb_urb_dir_in	usb_urb_dir_in	-
usb_urb_dir_out	usb_urb_dir_out	-
__create_pipe	__create_pipe	__create_pipe
usb_maxpacket	usb_maxpacket	usb_maxpacket
-	usb_endpoint_num	-
-	usb_endpoint_type	-
-	usb_endpoint_dir_in	usb_endpoint_dir_in
-	usb_endpoint_dir_out	usb_endpoint_dir_out
-	usb_endpoint_xfer_bulk	usb_endpoint_xfer_bulk
-	usb_endpoint_xfer_control	-
-	usb_endpoint_xfer_int	usb_endpoint_xfer_int
-	usb_endpoint_xfer_isoc	usb_endpoint_xfer_isoc
-	usb_endpoint_is_bulk_in	usb_endpoint_is_bulk_in
-	usb_endpoint_is_bulk_out	usb_endpoint_is_bulk_out
-	usb_endpoint_is_int_in	usb_endpoint_is_int_in
-	usb_endpoint_is_int_out	usb_endpoint_is_int_out
-	usb_endpoint_is_isoc_in	usb_endpoint_is_isoc_in
-	usb_endpoint_is_isoc_out	usb_endpoint_is_isoc_out

## 5.2.2 ch9.h (usb\_ch9.h)

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
usb_ctrlrequest	usb_ctrlrequest	usb_ctrlrequest
usb_descriptor_header	usb_descriptor_header	usb_descriptor_header
usb_device_descriptor	usb_device_descriptor	usb_device_descriptor
usb_config_descriptor	usb_config_descriptor	usb_config_descriptor
usb_string_descriptor	usb_string_descriptor	usb_string_descriptor
usb_interface_descriptor	usb_interface_descriptor	usb_interface_descriptor
usb_endpoint_descriptor	usb_endpoint_descriptor	usb_endpoint_descriptor
usb_endpoint_num	-	-
usb_endpoint_type	-	-
usb_endpoint_dir_in	-	-
usb_endpoint_dir_out	-	-
usb_endpoint_xfer_bulk	-	-
usb_endpoint_xfer_control	-	-
usb_endpoint_xfer_int	-	-
usb_endpoint_xfer_isoc	-	-
usb_endpoint_is_bulk_in	-	-
usb_endpoint_is_bulk_out	-	-
usb_endpoint_is_int_in	-	-
usb_endpoint_is_int_out	-	-
usb_endpoint_is_isoc_in	-	-
usb_endpoint_is_isoc_out	-	-
usb_ss_ep_comp_descriptor	-	-
usb_qualifier_descriptor	usb_qualifier_descriptor	usb_qualifier_descriptor
usb_otg_descriptor	usb_otg_descriptor	usb_otg_descriptor
usb_debug_descriptor	usb_debug_descriptor	usb_debug_descriptor
usb_interface_assoc_descriptor	usb_interface_assoc_descriptor	usb_interface_assoc_descriptor
usb_security_descriptor	usb_security_descriptor	usb_security_descriptor
usb_key_descriptor	usb_key_descriptor	usb_key_descriptor
usb_encryption_descriptor	usb_encryption_descriptor	usb_encryption_descriptor
usb_bos_descriptor	usb_bos_descriptor	usb_bos_descriptor
usb_dev_cap_header	usb_dev_cap_header	usb_dev_cap_header
usb_wireless_cap_descriptor	usb_wireless_cap_descriptor	usb_wireless_cap_descriptor
usb_ext_cap_descriptor	usb_ext_cap_descriptor	-
usb_wireless_ep_comp_descriptor	usb_wireless_ep_comp_descriptor	usb_wireless_ep_comp_descriptor
usb_handshake	usb_handshake	usb_handshake
usb_connection_context	usb_connection_context	usb_connection_context
usb_device_speed	usb_device_speed	usb_device_speed
usb_device_state	usb_device_state	usb_device_state

## 5.2.3 ch9.h (usb\_ch9.h)

Kernel 2.6.32	Kernel 2.6.28	Kernel 2.6.20
struct kref kref;	struct kref kref;	struct kref kref;
void *hcpriv;	void *hcpriv;	void *hcpriv;
atomic_t use_count;	atomic_t use_count;	atomic_t use_count;
atomic_t reject;	u8 reject;	u8 reject;
int unlinked;	int unlinked;	-
struct list_head urb_list;	struct list_head urb_list;	struct list_head urb_list;
struct list_head anchor_list;	struct list_head anchor_list;	-
struct usb_anchor *anchor;	struct usb_anchor *anchor;	-
struct usb_device *dev;	struct usb_device *dev;	struct usb_device *dev;
struct usb_host_endpoint *ep;	struct usb_host_endpoint *ep;	-
unsigned int pipe;	unsigned int pipe;	unsigned int pipe;
int status;	int status;	int status;
unsigned int transfer_flags;	unsigned int transfer_flags;	unsigned int transfer_flags;
void *transfer_buffer;	void *transfer_buffer;	void *transfer_buffer;
dma_addr_t transfer_dma;	dma_addr_t transfer_dma;	dma_addr_t transfer_dma;
struct usb_sg_request *sg;	-	-
int num_sgs;	-	-
u32 transfer_buffer_length;	int transfer_buffer_length;	int transfer_buffer_length;
u32 actual_length;	int actual_length;	int actual_length;
unsigned char *setup_packet;	unsigned char *setup_packet;	unsigned char *setup_packet;
dma_addr_t setup_dma;	dma_addr_t setup_dma;	dma_addr_t setup_dma;
int start_frame;	int start_frame;	int start_frame;
int number_of_packets;	int number_of_packets;	int number_of_packets;
int interval;	int interval;	int interval;
int error_count;	int error_count;	int error_count;
void *context;	void *context;	void *context;
usb_complete_t complete;	usb_complete_t complete;	usb_complete_t complete;
struct usb_iso_packet_descriptor iso_frame_desc[0];	struct usb_iso_packet_descriptor iso_frame_desc[0];	struct usb_iso_packet_descriptor iso_frame_desc[0];
-	-	spinlock_t lock;

**Revision Record**

Rev.	Date	Description	
		Page	Summary
0.01	Sep 9th, 2011	-	First edition issued.(ISG-NK1-100024)

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
  2. 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.
  3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
  4. 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.
  5. When exporting the 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. You should not use Renesas Electronics products or the 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. 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.
  6. 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.
  7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. 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 categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. 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 an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheet or data book, etc.  
 "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.  
 "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.  
 "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
  8. 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.
  9. 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 system 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. 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.
  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.



### SALES OFFICES

### Renesas Electronics Corporation

<http://www.renesas.com>

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**  
 Dukas Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.  
 Tel: +44-1628-585-100, Fax: +44-1628-585-900

**Renesas Electronics Europe GmbH**  
 Arcadiastrasse 10, 40472 Düsseldorf, Germany  
 Tel: +49-211-6503-0, 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.**  
 7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.  
 Tel: +886-2-8175-9600, Fax: +886-2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**  
 1 HarbourFront Avenue, #06-10, Keppel Bay Tower, Singapore 098632  
 Tel: +65-6213-0200, Fax: +65-6278-8001

**Renesas Electronics Malaysia Sdn.Bhd.**  
 Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jin 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 Laviel' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
 Tel: +82-2-558-3737, Fax: +82-2-558-5141