Network Driver for Renesas Devices User Guide

Version 1.40

For use with Network Driver for Renesas Devices module versions 1.04 and above

Date: 15-Jun-2017 17:30

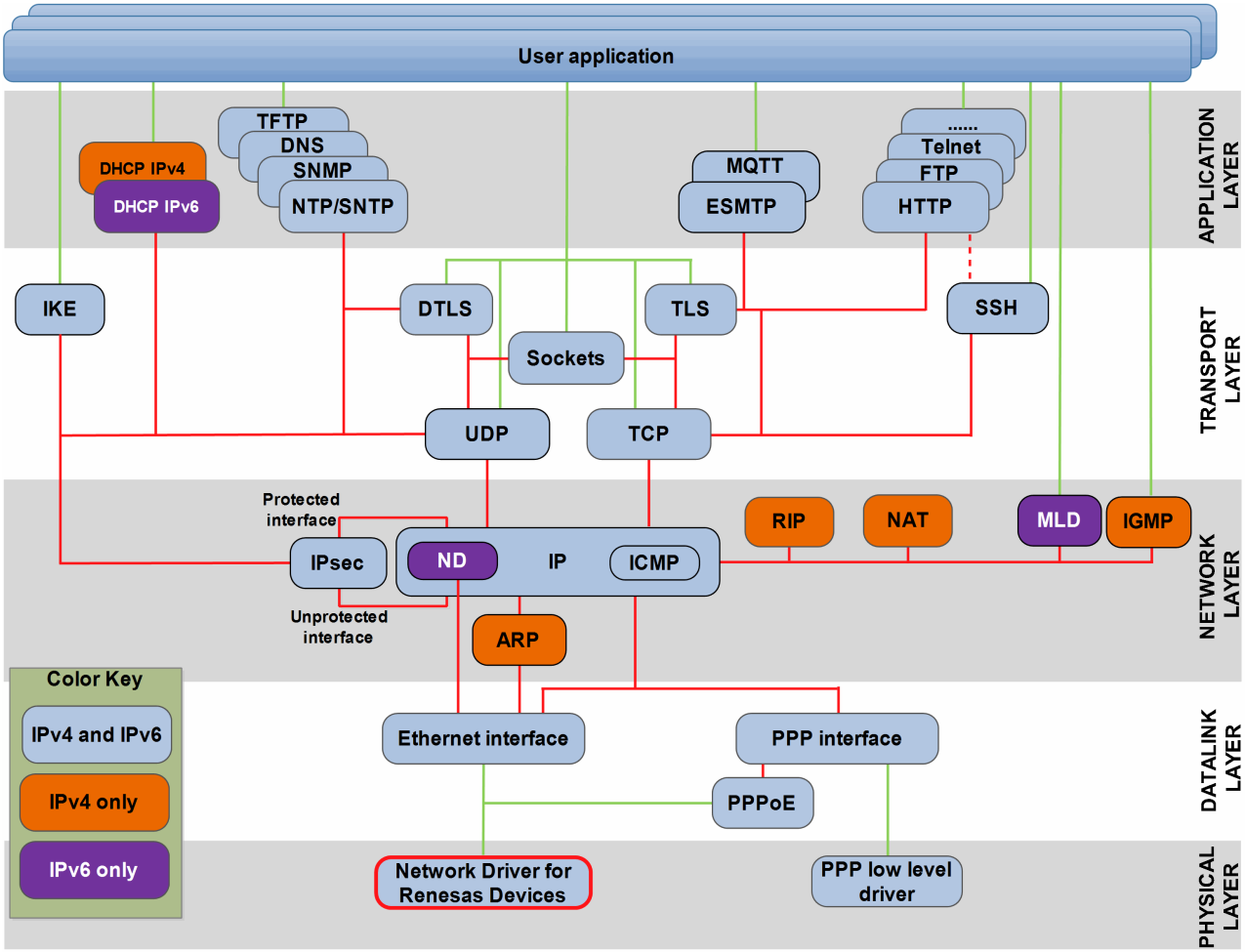
Table of Contents

1 System Overview

1.1 Introduction

This guide is for those who want to implement a network driver for Renesas standalone Ethernet controllers. Renesas MCUs supported include those in the SuperH SH-2A and SH7260 series.

The driver's location within HCC's TCP/IP stack is shown below. (In this diagram green lines show interfaces available to users of the TCP/IP stack, red lines show internal TCP/IP interfaces.)



Note: Although every attempt has been made to simplify the system's use, you need a good understanding of the requirements of the systems you are designing in order to obtain the maximum practical benefits. HCC Embedded offers hardware and firmware development consultancy to help you implement your system.

1.2 Feature Check

The main features of the network driver are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Designed for integration with both RTOS and non-RTOS based systems.
- Conforms to the HCC Coding Standard.
- Supports all Renesas 20/SuperH-2A, SH7260, and RX63 controllers.

1.3 Packages and Documents

Packages

The table below lists the packages which you need in order to use this module.

Package	Description
hcc_base_doc	This contains the two guides that will help you get started.
nw_drv_base	The network driver base package. This is the base system on which the Renesas driver is built.
nw_drv_eth_renesas	The Network Driver for Renesas package.

Documents

For an overview of the HCC TCP/IP stack software, 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 Network Driver User Guide

This document describes the network driver base system.

HCC Network Driver for Renesas 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: Network Driver for Renesas User Guide.
- For the history of changes made to the package code itself, see History: nw_drv_eth_renesas.

The current version of this manual is 1.40. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.40	2017-06-15	1.04	New *Change History* format.
1.30	2017-04-20	1.04	Added *Change History*. New functions in *PSP Porting*.
1.20	2017-03-29	1.02	Changes to TCP Stack diagram.
1.10	2017-01-16	8.06	Changes to TCP Stack diagram.
1.00	2016-07-14	8.01	First online version.

2 Source File List

This section lists and describes all the source code files included in the system. These files follow HCC Embedded's 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 files except the configuration file and PSP files.

2.1 API Header File

The file **src/api/api_ethdriver_renesas.h** should be included by any application using the system. This is the only file that should be included by an application using this module. It defines the eth_drv_renesas_init() function.

2.2 Configuration File

The file **src/config/config_ethdriver_renesas.h** contains all the configurable parameters of the system. Configure these as required. For details of the options, see Configuration Options.

2.3 System Files

The following files are in the directory **src/driver/network/ethernet/renesas**. **These files should only be modified by HCC**.

File	Description
eth_renesas.c	Ethernet driver code.
eth_renesas_reg.h	Registers header file.
eth_phy_lan87xx.c	LAN 8720 header file.
eth_phy_lan87xx_reg.c	LAN 8720 registers header file.

2.4 Version Files

These files in the directory **src/version** contain the version numbers of components of this module. The version number is checked by all modules that use a component to ensure system consistency over upgrades.

File	Description
ver_ethdriver_renesas.h	Module version file
ver_psp_eth_mii.h	Media Independent Interface (MII) version file.
ver_psp_eth_phy.h	Ethernet PSP version file
ver_psp_eth_phy_lan87xx.h	LAN 8720 version file.

2.5 Platform Support Package (PSP) Files

These files provide functions the core code needs to call, depending on the hardware. The following generalized PSP implementation files are in the directory **src/psp**. These provide templates for you to produce your own PSP. There is also a directory named **psp_rx63n** with files specifically for the RX63 device type implemented by HCC.

> **Note:** You may need to modify these PSP implementations for your specific microcontroller and development board; see PSP Porting for details.

File	Description
include/psp_eth_mii.h	Media Independent Interface (MII) header file.
include/psp_eth_phy.h	Ethernet PSP header file.

3 Configuration Options

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

HCC_RENESAS_LINK_STA_POLL_INTERVAL

The link timer interval in ms. The default is 500.

HCC_RENESAS_BUF_SIZE

The Ethernet buffer size. The default is (16 * 1024).

HCC_RENESAS_BUF_ALIGN_PWR_2

The buffer alignment as a power of 2. The default is 5.

HCC_RENESAS_MAX_TX_BUFD

The maximum number of EMAC TX descriptors to use per interface. The default is 4.

HCC_RENESAS_MAX_RX_BUFD

The maximum number of EMAC RX descriptors to use per interface. The default is 4.

HCC_RENESAS_MAC_ADDRESS

The MAC address of the driver. The default is { 0x00, 0xA0, 0x91, 0x00, 0x93, 0xCD }.

HCC_RENESAS_DATA_LE

The endianness. The default of 1 means data is little-endian. Change this to 0 for big-endian.

4 Application Programming Interface

This section describes the single Application Programming Interface (API) function and the error codes it may return.

4.1 eth_drv_renesas_init

Use this function to initialize the network driver.

Format

```
t_nwdriver_ret eth_drv_renesas_init (
    const uint32_t          param,
    t_nwdriver * * const    p_ethdriver )
```

Arguments

Parameter	Description	Type
param	The driver parameter.	uint32_t
p_ethdriver	Where to write the pointer to the driver.	t_nwdriver * *

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

4.2 Error Codes

This table lists the error codes that may be generated by the API calls:

Error code	Value	Meaning
NWDRIVER_SUCCESS	0	Execution successful.
NWDRIVER_ERROR	1	Operation failed.

5 Integration

This section describes all aspects of the network driver that require integration with your target project. This includes porting and configuration of external resources.

5.1 OS Abstraction Layer

The network driver uses the OS Abstraction Layer (OAL) that allows it to run seamlessly with a wide variety of RTOSes, or without an RTOS.

The network driver uses the following OAL components:

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

5.2 Utilities

The code creates and uses a single timer in the **hcc_timer** module. The **hcc_timer** module is included in your system when you install the base network driver module.

5.3 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. For full details of its functions and macros, see the *Platform Support Package (PSP) Base User Guide*.

The module makes use of the following standard PSP functions:

Function	Package	Component	Description
psp_memcpy()	psp_base	psp_string	Copies a block of memory. The result is a binary copy of the data.
psp_memset()	psp_base	psp_string	Sets the specified area of memory to the defined value.

The module makes use of the following standard PSP macros:

Macro	Package	Component	Description
PSP_RD_BE16	psp_base	psp_endianness	Reads a 16 bit value stored as big-endian from a memory location.
PSP_RD_BE32	psp_base	psp_endianness	Reads a 32 bit value stored as big-endian from a memory location.

The module makes use of the following functions that must be provided by the PSP. These are designed for you to port them easily to work with your hardware solution. The package includes samples in the **include /psp_eth_rx63n.c** file.

Function	Description
psp_eth_renesas_init()	Initializes the hardware for the Ethernet driver.
psp_eth_renesas_start()	Starts the driver.
psp_eth_renesas_stop()	Stops the driver.
psp_eth_renesas_delete()	Deletes the driver, releasing associated resources.
psp_eth_renesas_dsc_mem_alloc()	Allocates a unit ID-related buffer.
psp_eth_renesas_dsc_mem_free()	Releases a unit ID-related buffer.

These functions are described in the following sections.

psp_eth_renesas_init

This function is provided by the PSP to initialize the Ethernet driver.

Format

```
t_nwdriver_ret psp_eth_renesas_init ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_renesas_start

This function is provided by the PSP to start the Ethernet driver.

Format

```
t_nwdriver_ret psp_eth_renesas_start ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_renesas_stop

This function is provided by the PSP to stop the Ethernet driver.

Format

```
t_nwdriver_ret psp_eth_renesas_stop ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_renesas_delete

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

Format

```
t_nwdriver_ret psp_eth_renesas_delete( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_renesas_dsc_mem_alloc

This function is provided by the PSP to allocate a unit ID-related buffer.

Format

```
t_nwdriver_ret psp_eth_renesas_dsc_mem_alloc (
    uint8_t       uid,
    uint32_t *    p_size,
    uint8_t * *   p_mem  )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t
p_size	Where to write the buffer size.	uint32_t *
p_mem	On return, a pointer to the buffer memory.	uint8_t * *

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
ETHDRIVER_FAILED	Operation failed.

psp_eth_renesas_dsc_mem_free

This function is provided by the PSP to release a unit ID-related buffer.

Format

```
t_nwdriver_ret psp_eth_renesas_dsc_mem_free ( uint8_t uid )
```

Arguments

Parameter	Description	Type
uid	The unit ID.	uint8_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
ETHDRIVER_FAILED	Operation failed.