

Loopback Network Driver User Guide

Version 1.50

For use with Loopback Network Driver versions 1.06
and above

Date: 20-Feb-2018 17:17

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	4
Feature Check	5
Packages and Documents	6
Packages	6
Documents	6
Change History	7
Source File List	8
API Header File	8
Configuration File	8
Source Code Files	8
Version File	8
Configuration Options	9
Application Programming Interface	10
loopback_drv_init	10
Error Codes	11
Integration	12
PSP Porting	12

1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

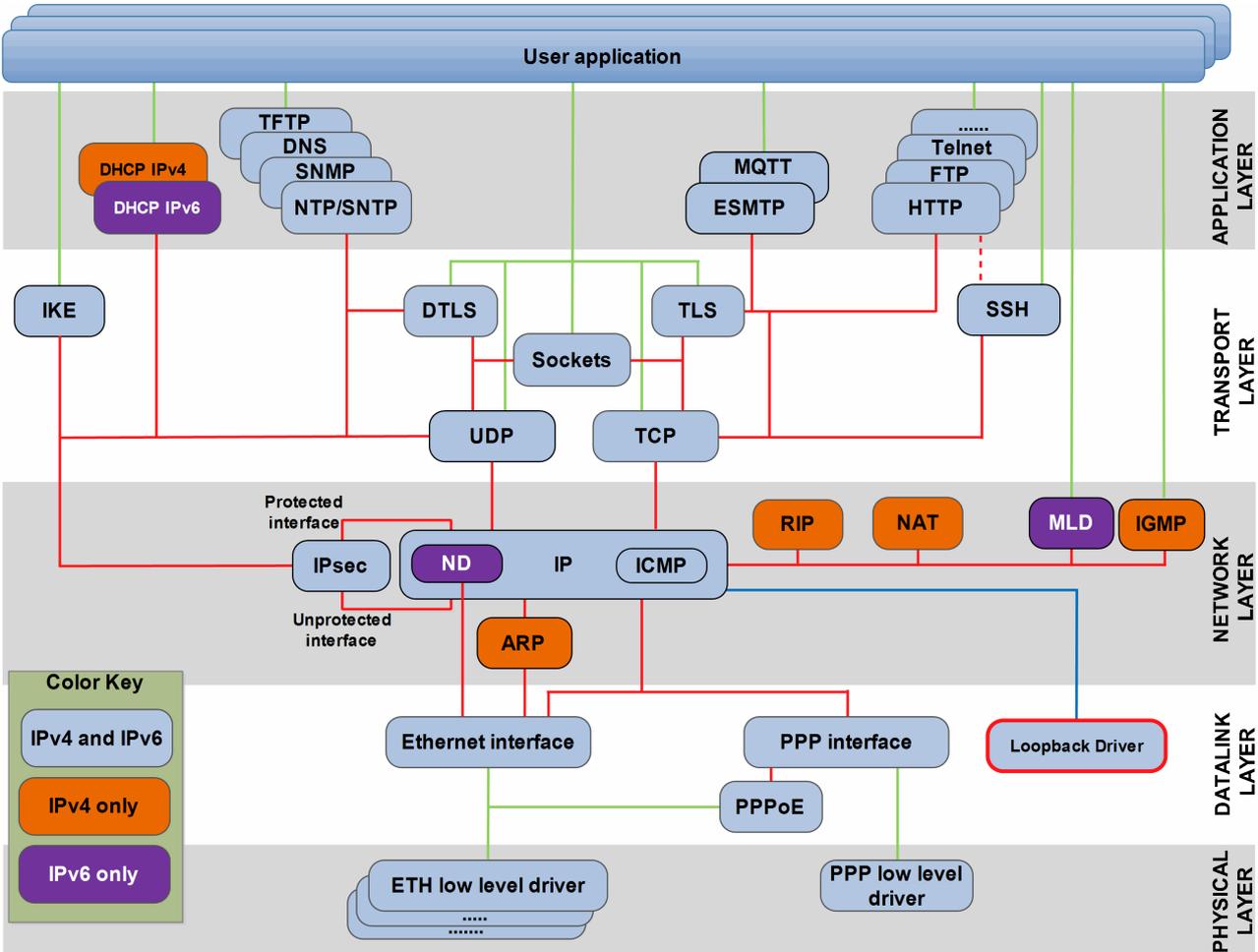
- [Introduction](#) – describes the main elements of the module. This section includes a diagram showing the position of the loopback driver within HCC's TCP/IP stack.
- [Feature Check](#) – summarizes the main features of the module as bullet points.
- [Packages and Documents](#) – the *Packages* section lists the packages that you need in order to use this module. The *Documents* section lists the relevant user guides.
- [Change History](#) – lists the earlier versions of this manual, giving the software version that each manual describes.

1.1 Introduction

This guide is for those who want to implement a loopback network driver as part of the HCC TCP/IP stack.

When the IP loopback driver is enabled, packets addressed to/received from IPv4 address 127.0.0.1 or IPv6 address ::1 are looped back to the source. This driver is useful for testing when working with the real remote application is not possible.

The loopback driver's position in the HCC TCP/IP stack is shown below; it is designed specifically for use with this stack. (In this diagram green lines show interfaces available to users of the stack, red lines show interfaces internal to the TCP/IP system.)



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

1.2 Feature Check

The main features of the loopback 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 including full MISRA compliance.

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>nw_drv_base</code>	The network driver base package. This is the base system on which the loopback driver is built.
<code>nw_drv_loopback</code>	The loopback driver package.

Additional packages

Other packages may be provided to work with the loopback driver. Examples include specific media drivers for particular targets, and PSP extensions for particular targets.

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 Loopback Network Driver User Guide

This is this document.

1.4 Change History

This section describes past changes to this manual.

- To view or download manuals, see [Network Driver PDFs](#).
- For the history of changes made to the package code itself, see [History: nw_drv_loopback](#).

The current version of this manual is 1.50. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.50	2018-02-20	1.06	Added LOOPBACK_DRV_HW_ADDR_ENABLE configuration option.
1.40	2017-06-15	1.05	New <i>Change History</i> format.
1.30	2017-03-29	1.05	Changes to TCP Stack diagram.
1.20	2017-01-17	1.05	Changes to TCP Stack diagram.
1.10	2016-04-22	1.05	Modified <i>Introduction</i> .
1.00	2015-12-22	1.05	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.

2.1 API Header File

The file `src/api/api_nwdriver_loopback.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. For details of the single function, see [Application Programming Interface](#).

2.2 Configuration File

The file `src/config/config_nwdriver_loopback.h` contains all the configurable parameters of the system. Configure these as required. This is the only file in the module that you should modify. For details of these options, see [Configuration Options](#).

2.3 Source Code Files

The file `src/driver/network/loopback/nwdriver_loopback.c` contains the source code. **This file should only be modified by HCC.**

2.4 Version File

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

3 Configuration Options

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

LOOPBACK_DRV_COUNT

The maximum number of loopback drivers. The default value is 1.

LOOPBACK_DRV_BUFFER_SIZE

The buffer size for loopback drivers. The default value is (16 * 1024).

LOOPBACK_DRV_MTU_SIZE

The size of the Maximum Transmission Unit (MTU). The default value is 1514.

LOOPBACK_DRV_RX_FEED

This controls how the receive buffers are added.

- Set this to zero to use the IP stack to add the receive buffers.
- Keep the default value of 1 if you want the driver to request the buffers.

LOOPBACK_DRV_RX_COUNT

The receive count. The default value is 4.

LOOPBACK_DRV_HW_ADDR_ENABLE

The default of 1 adds support for the `p_nwfn_set_hw_addr()` and `p_nwfn_get_hw_addr()` functions. Set this to zero if you want to disable these.

4 Application Programming Interface

This section describes all the Application Programming Interface (API) functions. It includes all the functions that are available to an application program.

4.1 loopback_drv_init

Use this function to initialize the loopback driver.

Format

```
t_nwdriver_ret loopback_drv_init (
    const uint32_t      param,
    t_nwdriver * * const p_lbdriver )
```

Arguments

Parameter	Description	Type
param	The driver parameter.	uint32_t
p_lbdriver	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 all 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 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 *HCC Base Platform Support Package User Guide*.

The module makes use of the following standard PSP functions:

Function	Package	Element	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.