

DHCP Server User Guide

Version 1.40

For use with DHCP Server versions 1.02 and above

Date: 16-Jan-2017 12:17

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

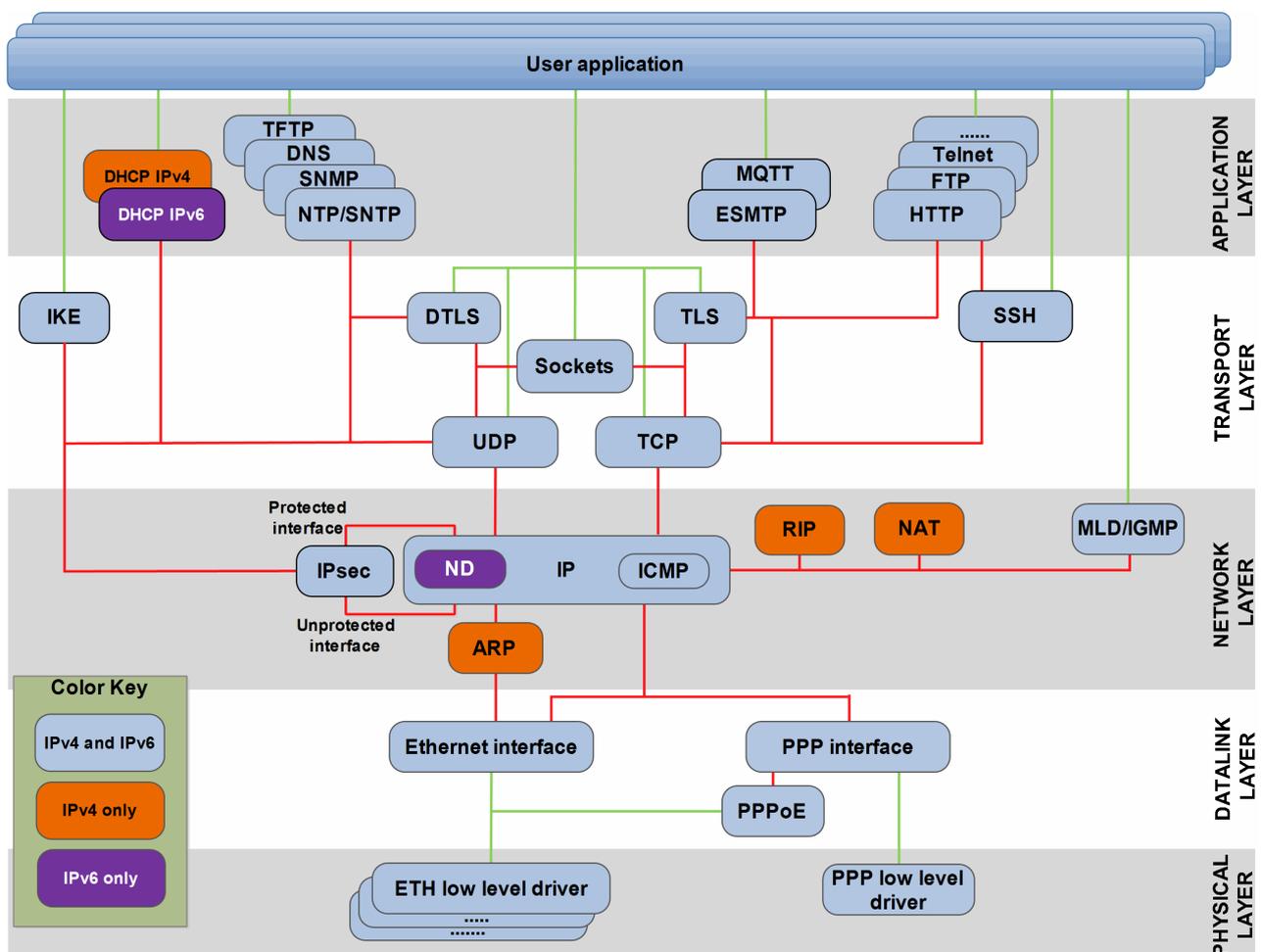
System Overview	3
Introduction	3
Feature Check	4
Packages and Documents	4
Packages	4
Documents	4
Change History	5
Source File List	6
API Header File	6
Configuration File	6
DHCP System	6
Version File	6
Configuration Options	7
Application Programming Interface	8
Module Management	8
dhcps_init	9
dhcps_start	10
dhcps_stop	11
dhcps_delete	12
dhcps_set_config	13
Error Codes	14
Types and Definitions	15
t_dhcps_config	15
Integration	16
OS Abstraction Layer	16
Utilities	16
PSP Porting	17

1 System Overview

1.1 Introduction

This guide is for those who want to implement a Dynamic Host Control Protocol (DHCP) Server module as part of their TCP/IP stack. The DHCP Server module is used to configure dynamically IP addresses and other information that is needed for Internet communication.

The DHCP Server module is part of the HCC MISRA-compliant TCP/IP stack, as shown below, and is designed specifically for use with it. (In this diagram green lines show interfaces available to users of the stack, red lines show interfaces internal to the TCP/IP system.)



The DHCP server manages a pool of IP addresses and information about client configuration parameters such as default gateway, domain name, and name servers. When a DHCP client connects to a network, it sends a broadcast query to a DHCP server, requesting the necessary information. If the request is valid, the server assigns the client an IP address, a lease (the length of time the allocation is valid), and other IP configuration parameters, such as the subnet mask and the default gateway.

The query is typically sent straight after booting and must complete before the client can initiate IP-based communication with other hosts. Upon disconnection the IP address is returned to the pool for use by another computer. In this way many computers can use the same IP address in a short time.

1.2 Feature Check

The main features of the system are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Complies with the HCC MISRA-compliant TCP/IP stack.
- Designed for integration with both RTOS and non-RTOS based systems.
- Compliant with [RFC 2131](#).
- The number of clients is configurable.
- Address leasing is configurable.

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>ip_app_dhcps</code>	The DHCP server package described in this manual.
<code>mip_base</code>	The TCP/IP stack base package.
<code>mip_udp</code>	The UDP 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 TCP/IP IPv4 Stack System User Guide, HCC TCP/IP Dual Stack System User Guide

These are the core documents that describe the complete TCP/IP system. The second manual covers both IPv4 and IPv6 systems.

HCC DHCP Server User Guide

This is this document.

1.4 Change History

This section includes recent changes to this product. For a list of all the changes, refer to the file **hcc/history/ip/ip_app_dhcps.txt** in the distribution package.

Version	Changes
1.02	Updated to work with IP base major version 5. Corrected client IP address verification in DHCP INFORM request.
1.01	Initial release.

2 Source File List

The following sections describe 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 files except the configuration file.

2.1 API Header File

The file `src/api/api_ip_app_dhcps.h` is the only file that should be included by an application using this module. For details of these API functions, see [Application Programming Interface](#).

2.2 Configuration File

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

2.3 DHCP System

The file `src/ip/apps/dhcps/dhcps.c` is the main source code file. **This file should only be modified by HCC.**

2.4 Version File

The file `src/version/ver_ip_app_dhcps.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.

3 Configuration Options

All system configuration options are set in the file `src/config/config_ip_app_dhcps.h`. This section lists the available configuration options and their default values.

DHCP_TASK_STACK_SIZE

The DHCP server task stack size. The default value is 1024.

DHCPS_MAX_CLIENT_NUM

The maximum number of parallel connected DHCP clients. The default value is 3.

DHCPS_MIN_LEASE_TIME

The minimum lease time in seconds. The default value is 3600.

DHCP_MAX_LEASE_TIME

The maximum lease time in seconds. The default value is (24 * 3600).

DHCPS_PING_TIMEOUT

The maximum time for a ping timeout in seconds. The default value is 1.

DHCPS_CLIENT_ID_MAX_LEN

The maximum length of the Client identification field. The default value is 16.

4 Application Programming Interface

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

4.1 Module Management

The functions are the following:

Function	Description
dhcps_init()	Initializes the module and allocates the required resources.
dhcps_start()	Starts the module.
dhcps_stop()	Stops the module.
dhcps_delete()	Deletes the module and releases the resources it used.
dhcps_set_config()	Sets the DHCP server configuration of an interface.

dhcps_init

Use this function to initialize the module and allocate the required resources.

Note: You must call this function before any other.

Format

```
t_dhcps_ret dhcps_init (void)
```

Arguments

Argument

None.

Return Values

Return value	Description
DHCPS_SUCCESS	Successful execution.
DHCPS_ERR_OS	Error during allocation of OS resource.
DHCPS_ERROR	Operation failed.
Else	See Error Codes .

dhcps_start

Use this function to start the module.

Note: You must call **dhcps_init()** before this to initialize the module.

Format

```
t_dhcps_ret dhcps_start ( void )
```

Arguments

Argument

None.

Return values

Return value	Description
DHCPS_SUCCESS	Successful execution.
DHCPS_ERROR	Operation failed.
Else	See Error Codes .

dhcps_stop

Use this function to stop the module.

Format

```
t_dhcps_ret dhcps_stop ( void )
```

Arguments

Argument
None.

Return values

Return value	Description
DHCPS_SUCCESS	Successful execution.
DHCPS_ERROR	Operation failed.
Else	See Error Codes .

dhcps_delete

Use this function to delete the module and release the associated resources.

Format

```
t_dhcps_ret dhcps_delete ( void )
```

Arguments

Argument
None.

Return values

Return value	Description
DHCPS_SUCCESS	Successful execution.
DHCPS_ERR_OS	Error during delete of OS resource.
DHCPS_ERROR	Operation failed.
Else	See Error Codes .

dhcps_set_config

Use this function to set the DHCP server configuration of an interface.

Format

```
t_dhcps_ret dhcps_set_config (
    const t_ip_ifc_hdl      ifc_hdl,
    const t_dhcps_config * const p_dhcps_config )
```

Arguments

Argument	Description	Type
ifc_hdl	The interface handle.	t_ip_ifc_hdl
p_dhcps_config	A pointer to the new configuration.	t_dhcps_config *

Return values

Return value	Description
DHCPS_SUCCESS	Successful execution.
DHCPS_ERROR	Operation failed.
Else	See Error Codes .

4.2 Error Codes

The table below lists the return codes that may be generated by the API calls.

Error code	Value	Meaning
DHCPS_SUCCESS	0U	Successful execution.
DHCPS_ERROR	1U	Operation failed.
DHCPS_ERR_IP_ADDR	2U	IP address error.
DHCPS_ERR_NO_MORE_ENTRY	3U	No more entries available.
DHCPS_ERR_INVALID_PARAM	4U	Invalid parameter.
DHCPS_ERR_INVALID_CONFIG	5U	Invalid configuration.
DHCPS_ERR_OS	6U	Error during allocation of OS resource.

4.3 Types and Definitions

t_dhcps_config

The *t_dhcps_config* structure defines the DHCP server configuration:

Element	Type	Description
dsc_ipaddr_min	uint32_t	The DHCP server IP address range minimum.
dsc_ipaddr_max	uint32_t	The DHCP server IP address range maximum.
dsc_netmask	uint32_t	The netmask address.
dsc_gateway	uint32_t	The gateway address.
dsc_dns1_addr	uint32_t	The first DNS server address.
dsc_dns2_addr	uint32_t	The second DNS server address.

5 Integration

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

5.1 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 uses the following OAL components:

OAL Resource	Number Required
Tasks	1
Mutexes	1
Events	1

5.2 Utilities

The DHCP 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 TCP/IP modules.

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 *HCC Base Platform Support Package User Guide*.

The module makes use of the following standard PSP functions:

Function	Package	Element	Description
psp_memcmp()	psp_base	psp_string	Compares two blocks of memory.
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	Element	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.
PSP_WR_BE32	psp_base	psp_endianness	Writes a 32 bit value to be stored as big-endian to a memory location.