DHCP Client for IPv4 User Guide

Version 1.40

For use with Dynamic Host Control Protocol (DHCP) Client for IPv4 versions 2.23 and above

Exported on 12/07/2018

Table of Contents

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 this module 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.

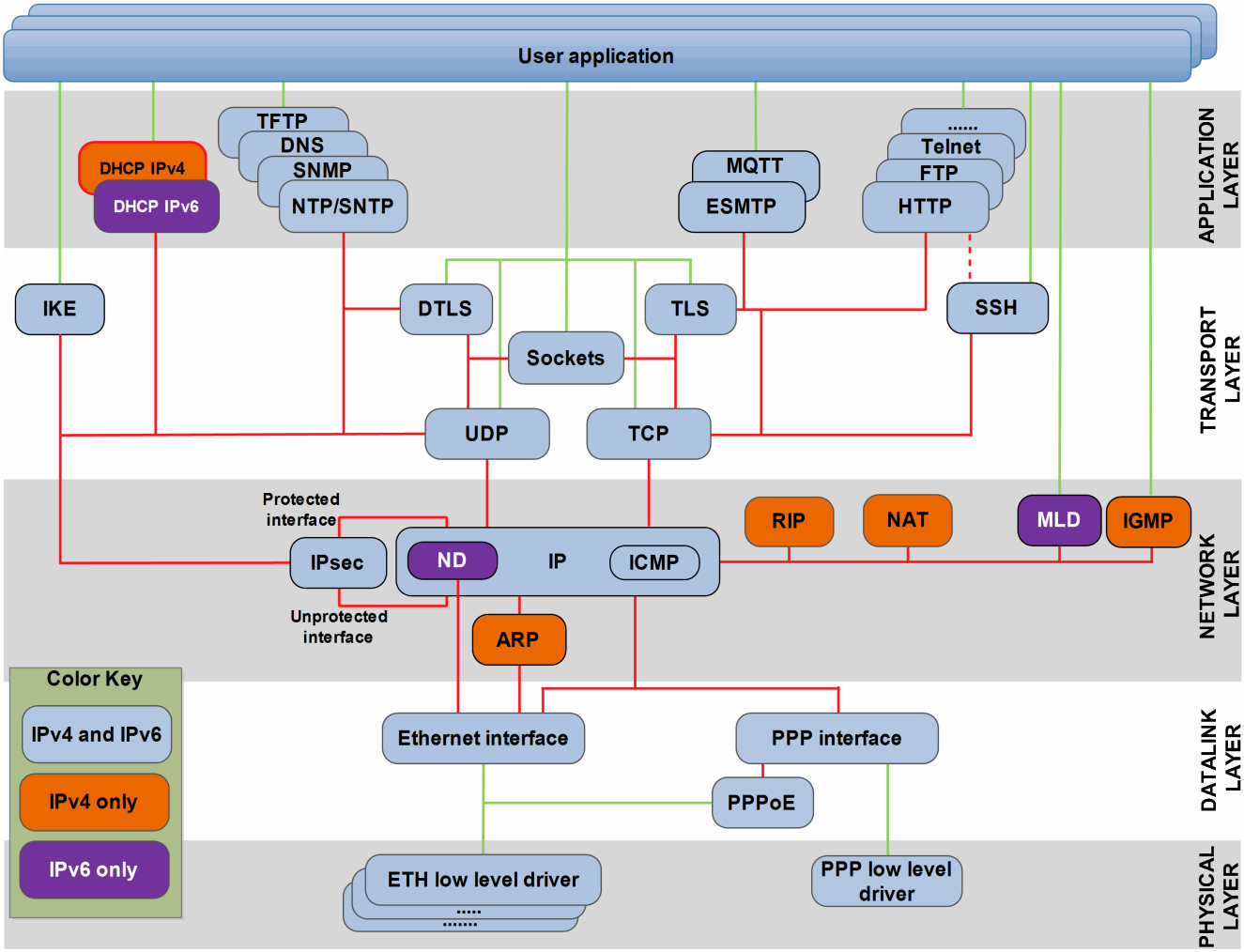
Note: To download this manual as a PDF, see TCP/IP PDFs.

1.1 Introduction

This guide is for those who want to implement a Dynamic Host Control Protocol (DHCP) client module as part of their TCP/IP stack. The DHCP Client for IPv4 module is used by a client (a computer or other device) to get an IP address automatically from a remote DHCP server.

This client module supports IPv4 addresses. It can be used alone in an IPv4 system or in the HCC dual TCP/IP stack alongside the DHCP client for IPv6.

The DHCP Client for IPv4 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.)



When a DHCP-configured client connects to a network, it sends a broadcast query to a DHCP server, requesting necessary information. If the request is valid, the server assigns the device 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 may use the same IP address in a short time.

Use of vendor class identifiers is supported, allowing a DHCP client to tell a server their vendor type and configuration. The information is a vendor-specific string, interpreted where possible by the server. Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration.

Servers that cannot interpret the vendor-specific information sent by a client **must** ignore it. Other servers should respond to the client.

1.2 Feature Check

The main features of the system are the following:

- Conforms to the HCC Advanced Embedded Framework.
- Compliant with the HCC MISRA-compliant TCP/IP stack.
- Supports IPv4 addresses.
- Supports vendor classes.
- Designed for integration with both RTOS and non-RTOS based systems.
- Compliant with RFC 2131 and RFC 2132.
- Has optional support for Fully Qualified Domain Names (FQDNs).

1.3 Packages and Documents

Packages

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

Package	Description
hcc_base_doc	This contains the two guides that will help you get started.
ip_app_dhcp_v4	The DHCP client for IPv4 package described in this manual.
ip_base_v4	The TCP/IP IPv4 base package.
mip_udp	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 Dual Stack System User Guide

This is the core document that describes the complete TCP/IP stack. It covers both IPv4 and IPv6 systems.

HCC DHCP Client for IPv4 User Guide

This is this document.

HCC DHCP Client for IPv6 User Guide

This document describes the similar DHCP client that supports IPv6 addresses.

1.4 Change History

This section describes past changes to this manual.

- To download this manual or a PDF describing an earlier software version, see TCP/IP PDFs.
- For the history of changes made to the package code itself, see History: ip_app_dhcp_v4.

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	2018-12-07	2.23	Added API section for handling vendor class identifier. Added DHCP_VENDOR_ID_MAX_LEN configuration option.
1.30	2017-06-20	2.19	New *Change History* format.
1.20	2017-03-28	2.18	Updated network diagram.
1.10	2017-01-18	2.18	Updated network diagram.
1.00	2016-04-05	2.16	First online version.

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_dhcp.h** is the only file that should be included by an application using this module. For details of the API function and callback, see Application Programming Interface.

2.2 Configuration File

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

2.3 DHCP Client System

These files are in the directory **src/ip/apps/dhcp**. **These files should only be modified by HCC**.

File	Description
dhcp.c	DHCP client source code.
dhcp.h	Header file.

2.4 Version File

The file **src/version/ver_ip_app_dhcp.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

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

DHCP_TASK_STACK_SIZE

The DHCP client task configuration. The default value is 1024.

DHCP_TIMER_PERIOD

The timer period in milliseconds. The default value is 100.

DHCP_MSG_EXPIRY

The DHCP message expiry time in milliseconds. The default value is 10000.

DHCP_LEASE_TIME

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

DHCP_FQDN_ASCII_ENC

Set this to 1 for ASCII encoding of the domain name used for FQDNs in the DHCP request phase. The default value is 0.

DHCP_CLI_ID_HW_ADDR

Keep this at the default of 1 to use the hardware address for the client identifier. To use FQDNs, set it to 0.

DHCP_VENDOR_ID_MAX_LEN

The maximum length of the vendor class identifier used during a DHCP request. See the dhcp_set_vendor_class_request() function.

4 Application Programming Interface

This section describes the single Application Programming Interface (API) function and the callback it uses.

4.1 dhcp_set_vendor_class_request

Use this function to set the vendor class identifier and callback function to be used during DHCP negotiation.

This allows a DHCP client to tell a server their vendor type and configuration. The information is a vendor-specific string, interpreted where possible by the server. Vendors may choose to define specific vendor class identifiers to convey particular configuration or other identification information about a client. For example, the identifier may encode the client's hardware configuration.

Servers that cannot interpret the vendor-specific information sent by a client MUST ignore it. Servers that respond should only use option 43 to return the vendor-specific information to the client. See RFC 2132 sections 8.4 and 9.13 for full information.

Format

```
t_ip_ret dhcp_set_vendor_class_request (
    const uint8_t * const        p_vendor_class_id,
    const uint8_t                len,
    const t_dhcp_vendor_ntf_fn   dhcp_vendor_ntf_fn )
```

Arguments

Arguments	Description	Type
p_vendor_class_id	The vendor class identifier to be sent during the DHCP request.	uint8_t *
len	The length of *vendor_class_id*.	uint8_t
dhcp_vendor_ntf_fn	The function to call after vendor-specific information arrives from a DHCP server.	t_dhcp_vendor_ntf_fn

Return Values

Return value	Description
IP_SUCCESS	Successful execution.
IP_ERR_INVALID_SIZE	Configuration option DHCP_VENDOR_ID_MAX_LEN is too small for the given vendor class ID.

4.2 t_dhcp_vendor_ntf_fn

This *typedef* defines the callback function to call after vendor-specific information is received from a DHCP server.

Format

```
typedef void ( * t_dhcp_vendor_ntf_fn )(
    const t_ip_ifc_hdl   ifc_hdl,
    const uint8_t *      const p_vendor_specific_info,
    const uint8_t        len )
```

Arguments

Arguments	Description	Type
ifc_hdl	The handle of the IP interface that received the vendor-specific information.	t_ip_ifc_hdl
p_vendor_specific_info	A pointer to the vendor-specific information (Code 43).	uint8_t *
len	The length of the vendor-specific information.	uint8_t

Return Values

None.

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	2
Events	1

The DHCP task is started automatically by the IP stack if DHCP is enabled (that is, if the IP_DHCP_ENABLE configuration option in the IPv4 base package is set).

The DHCP task function is named **dhcp_task()**.

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_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.