

Network Driver for Freescale EtherNet IP Devices User Guide

Version 1.10

For use with Network Driver for Freescale™ EtherNet
/IP™ Devices versions 1.03 and above

Date: 16-Jun-2017 10:42

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

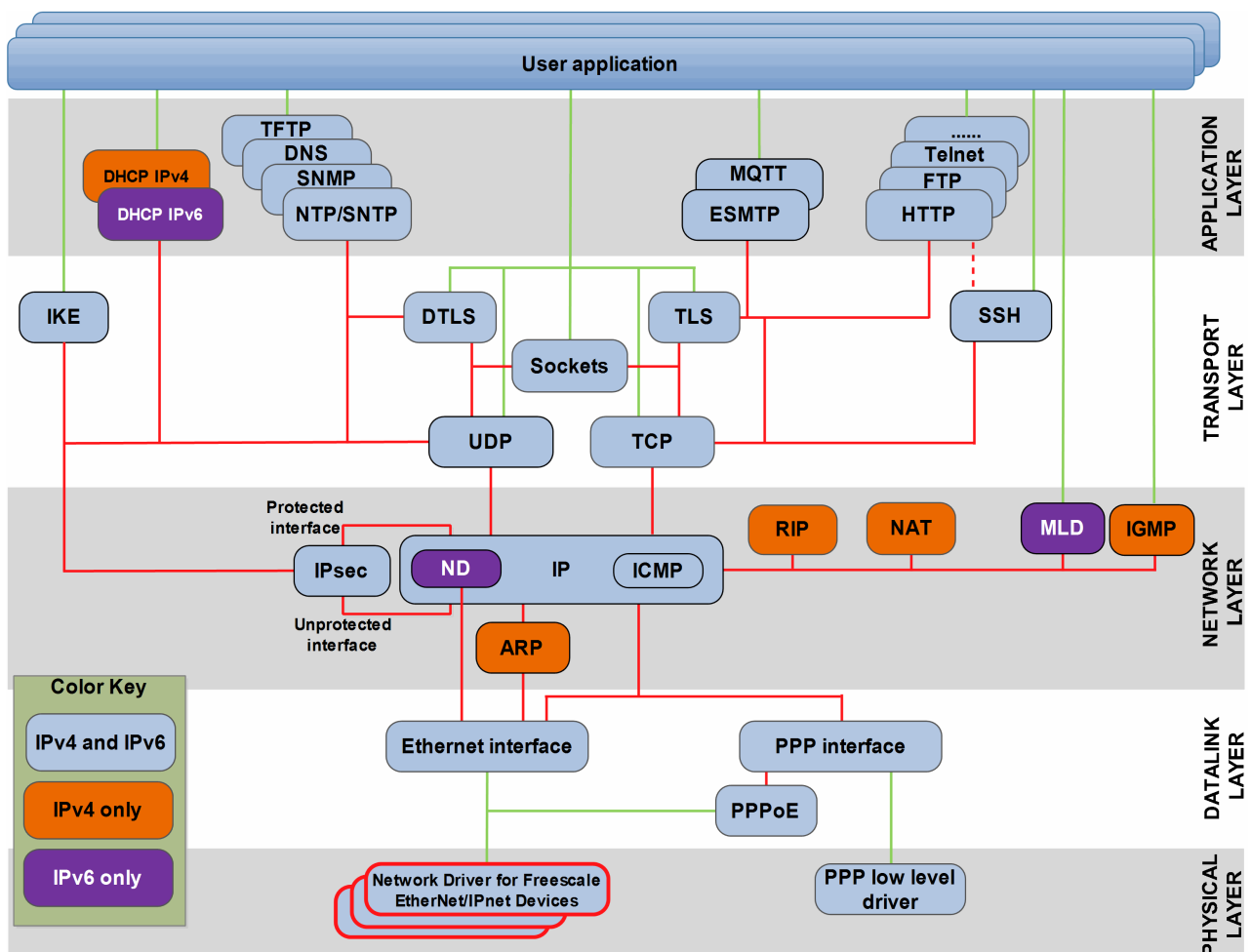
System Overview	3
Introduction	3
Feature Check	4
Packages and Documents	5
Packages	5
Documents	5
Change History	6
Source File List	7
API Header File	7
Configuration Files	7
System Files	7
Version Files	8
Platform Support Package (PSP) Files	8
Configuration Options	9
src/config/config_ethdriver_enet.h	9
config_eth_phy_ksz8xxx.h	10
config_eth_phy_ksz9xxx.h	10
Application Programming Interface	11
enet_eth_drv_init	11
Error Codes	12
Integration	13
OS Abstraction Layer	13
Utilities	13
PSP Porting	14
psp_enet_eth_init	15
psp_enet_eth_start	16
psp_enet_eth_stop	17
psp_enet_eth_delete	18
psp_enet_get_buf	19

1 System Overview

1.1 Introduction

This guide is for those who want to implement a network driver for EtherNet/IP™ devices from Freescale™ Semiconductor Inc (now Qualcomm®). These include i.MX 6 and all Ethernet-capable Kinetis microcontrollers.

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, including full MISRA compliance.
- Fully compatible with the HCC Network Driver Interface specification.
- Supports EtherNet/IP™ devices from Freescale™ Semiconductor Inc (now Qualcomm®). These include i.MX 6 and all Ethernet-capable Kinetis microcontrollers.
- HCC provides fully tested reference drivers for this module.

1.3 Packages and Documents

Packages

The table below lists the packages which 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 this driver is built.
<code>nw_drv_eth_enet</code>	The Network Driver for Freescale™ EtherNet/IP™ Devices package.

Documents

For an overview of HCC's 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 Freescale EtherNet/IP Devices 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 Freescale EtherNet IP Devices User Guide](#).
- For the history of changes made to the package code itself, see [History: nw_drv_eth_enet](#).

The current version of this manual is 1.10. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.10	2017-06-16	3.07	New <i>Change History</i> format.
1.00	2017-04-25	3.07	First online version.

2 Source File List

This section describes 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 and PSP files.

2.1 API Header File

The file `src/api/api_ethdriver_enet.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 `enet_eth_drv_init` function.

2.2 Configuration Files

These files in the directory `src/config` contain all the configurable parameters of the system. Configure these as required. For details of these options, see [Configuration Options](#).

File	Description
<code>config_eth_phy_ksz8xxx.h</code>	KSZ8xxx Ethernet driver configuration.
<code>config_eth_phy_ksz9xxx.h</code>	KSZ9xxx Ethernet driver configuration.
<code>config_ethdriver_enet.h</code>	Ethernet driver configuration.

2.3 System Files

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

File	Description
<code>eth_enet.c</code>	Ethernet driver code.
<code>eth_phy_ksz8xxx.c</code>	KSZ8xxx Ethernet driver code.
<code>eth_phy_ksz8xxx_reg.h</code>	KSZ8xxx registers header file.
<code>eth_phy_ksz9xxx.c</code>	KSZ9xxx Ethernet driver code.
<code>eth_phy_ksz9xxx_reg.h</code>	KSZ9xxx registers header file.

There are files for specific devices in folders under `src`, for example `src/psp_imx6`.

2.4 Version Files

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

File	Description
ver_eth_phy_ksz9xxx.h	KSZ9xxx driver version number.
ver_ethdriver_enet.h	Ethernet driver version number.
ver_psp_eth_phy_ksz8xxx.h	KSZ8xxx driver version number.
ver_psp_eth_phy_ksz9xxx.h	KSZ9xxx driver version number.

2.5 Platform Support Package (PSP) Files

These files provide functions the core code needs to call, depending on the hardware. There is a folder named **psp_xxx** for each device type that HCC has implemented.

The following general PSP implementation files are in the directory **src/psp**. These provide templates for you to produce your own PSP for other devices.

Note: You must 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.
target/eth/psp_eth_enet.h	Functions header file.
target/eth/psp_eth_enet.c	Functions source code.
target/include/hcc_imx6_regs.h	The device registers header file for the i.MX6.
target/include/hcc_imx6_regs.h	The device registers header file for the Kinetis K60.

3 Configuration Options

Set the system configuration options in the files in **src/config**, as described below. This section lists the available configuration options and their default values.

3.1 src/config/config_ethdriver_enet.h

ETHERNET_BUF_SIZE

The Ethernet buffer size. The default is 16512.

ETH_ISR_ID

The interrupt ID. The default value is 92.

ETH_ISR_PRIO

The Link ISR priority, defined in case it is used . The default value is 20.

ETH_PHY_ADDR

The physical address. The default value is 0.

ETH_PHY_CONN

The mode: Media-Independent Interface (MII), Reduced Media-Independent Interface (RMII), or Reduced Gigabit Media-Independent Interface (RGMII). The default value is ETH_PHY_CONN_RMII.

ETH_LINK_STA_POLL_INTERVAL

The poll interval in milliseconds. The default value is 1000.

MAX_RX_DESC

The maximum number of RX descriptors. The default value is 4.

MAX_TX_DESC

The maximum number of TX descriptors. The default value is 4.

MAC_ADDRESS

The MAC address of the driver. The default value is { 0x00, 0xA2, 0x92, 0x00, 0x92, 0xAC }.

ETH_DUPLEX

This specifies whether the connection is duplex or not. The default value is ETH_PHY_AUTONEG, meaning this is set by auto-negotiation.

ETH_SPEED

The Ethernet speed. The default value is ETH_PHY_AUTONEG, meaning this is set by auto-negotiation.

3.2 config_eth_phy_ksz8xxx.h

This file contains no options.

3.3 config_eth_phy_ksz9xxx.h

This file contains no options, but requires you to include the ETH driver configuration file in it. For example, the file may include this line:

```
#include "config_ethdriver_enet.h"
```

4 Application Programming Interface

This section describes the single API function and the error codes it may return.

4.1 enet_eth_drv_init

Use this function to initialize the network driver.

Format

```
t_nwdriver_ret enet_eth_drv_init (
    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 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 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	1
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 *HCC Base Platform Support Package User Guide*.

The module makes use of the following standard PSP function:

Function	Package	Element	Description
<code>psp_memcpy()</code>	psp_base	psp_string	Copies a block of memory. The result is a binary copy of the data.

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 `psp/target/eth/psp_eth_enet.c` file.

Function	Description
<code>psp_enet_eth_init()</code>	Initializes the hardware for the Ethernet driver.
<code>psp_enet_eth_start()</code>	Starts the driver.
<code>psp_enet_eth_stop()</code>	Stops the driver.
<code>psp_enet_eth_delete()</code>	Deletes the driver, releasing associated resources.
<code>psp_enet_get_buf()</code>	Get the address of the Ethernet buffer.

These functions are described in the following sections.

psp_enet_eth_init

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

Format

```
t_nwdriver_ret psp_enet_eth_init ( uint32_t module_id )
```

Arguments

Parameter	Description	Type
module_id	Where to store the buffer address.	uint32_t

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_enet_eth_start

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

Format

```
t_nwdriver_ret psp_enet_eth_start ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_enet_eth_stop

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

Format

```
t_nwdriver_ret psp_enet_eth_stop ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_enet_eth_delete

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

Format

```
t_nwdriver_ret psp_enet_eth_delete ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_enet_get_buf

This function is provided by the PSP to get the address of the Ethernet buffer.

Format

```
t_nwdriver_ret psp_enet_get_buf ( uint8_t * * const pp_buf )
```

Arguments

Parameter	Description	Type
pp_buf	Where to store the buffer address.	uint8_t **

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.