

# Network Driver for ENC624J600 Devices User Guide

Version 1.50

For use with Network Driver for ENC624J600 versions  
1.04 and above

**Date:** 16-Jun-2017 10:36

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
Device Description	4
Packages and Documents	5
Packages	5
Documents	5
Change History	6
Source File List	7
API Header File	7
Configuration File	7
System Files	7
Version File	7
Platform Support Package (PSP) Files	8
Configuration Options	9
Application Programming Interface	10
eth_drv_encx24_init	10
Error Codes	11
Integration	12
OS Abstraction Layer	12
Utilities	12
PSP Porting	13
psp_eth_encx24_init	14
psp_eth_encx24_start	15
psp_eth_encx24_stop	16
psp_eth_encx24_delete	17

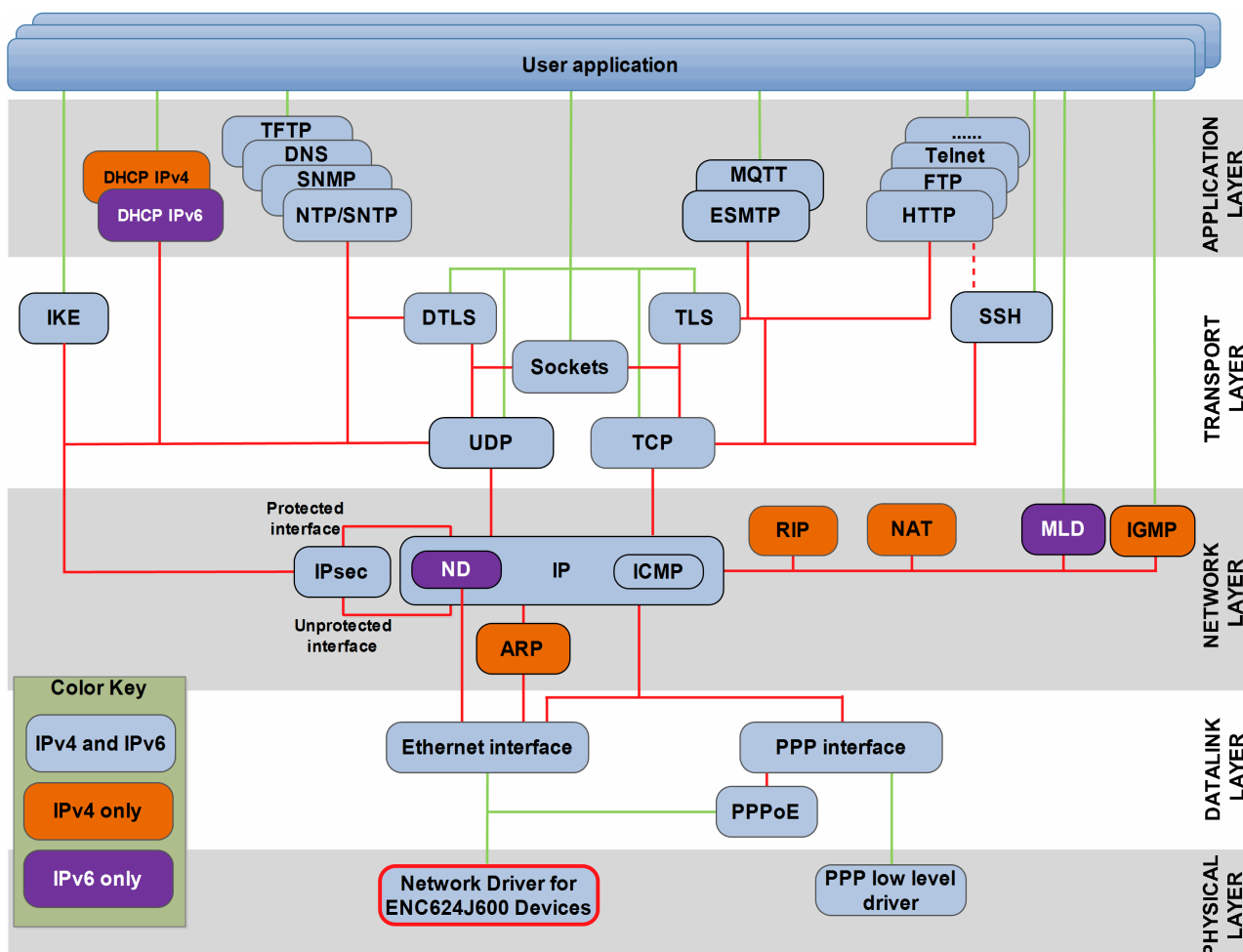
# 1 System Overview

## 1.1 Introduction

This guide is for those who want to implement a network driver for the Microchip ENC624J600 standalone Ethernet controller.

The ENC624J600 is a standalone 10/100 Base-T Ethernet interface controller with integrated MAC & PHY, hardware cryptographic security engines, and a factory pre-programmed unique MAC address. A 24 Kbyte on-chip RAM buffer is available for TX and RX operations. The device has a Serial Peripheral Interface (SPI) or a flexible parallel interface. It can serve as an Ethernet network interface for any microcontroller equipped with SPI or a standard parallel port.

This diagram shows the driver within the HCC TCP/IP 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 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.
- Fully compatible with the HCC Network Driver Interface specification.
- Compatible with Microchip ENC624J600 and also ENC424J600 devices.

## 1.3 Device Description

This table summarizes the properties of the two Microchip devices:

	ENC424J600	ENC624J600
<b>Pins</b>	44	64
<b>Speed</b>	10/100 Mbps	10/100 Mbps
<b>SRAM</b>	24Kb	24Kb
<b>SPI</b>	Yes	Yes
<b>Parallel interface</b>	8 bit multiplexed	8 and 16 bit multiplexed 8 and 16 bit demultiplexed

## 1.4 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 the ENC624J600 driver is built.
<code>nw_drv_eth_encx24</code>	The Network Driver for ENC624J600 Devices 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 ENC624J600 Devices User Guide

This is this document.

## 1.5 Change History

This section describes past changes to this manual.

- To view or download earlier manuals, see [Archive: Network Driver for ENC624J600 Devices User Guide](#).
- For the history of changes made to the package code itself, see [History: nw\\_drv\\_eth\\_enc24x](#).

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	2017-06-16	1.04	New <i>Change History</i> format.
1.40	2017-03-29	1.04	Changes to TCP Stack diagram.
1.30	2017-01-17	1.04	Changes to TCP Stack diagram.
1.20	2016-03-02	1.04	Changes to TCP Stack diagram.
1.10	2016-02-19	1.02	Modified <i>Introduction</i> . Added <i>Device Description</i> .
1.00	2015-12-21	1.02	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_encx24.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_encx24_init` function.

### 2.2 Configuration File

The file `src/config/config_ethdriver_encx24.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 folder `src/driver/network/ethernet/encx24`. **These files should only be modified by HCC.**

File	Description
<code>eth_encx24.c</code>	Ethernet driver code.
<code>eth_encx24_phy.c</code>	Ethernet driver code.
<code>eth_encx24_reg.h</code>	Header file.

### 2.4 Version File

The file `src/version/ver_ethdriver_encx24.h` contains the version number of this module. The version number is checked by all modules that use a module to ensure system consistency over upgrades.

## 2.5 Platform Support Package (PSP) Files

These files provide functions the core code needs to call, depending on the hardware. The following general PSP implementation files are in the directory **src/psp**. These provide templates for you to produce your own PSP.

**Note:** You must modify these PSP implementations for your specific microcontroller and development board; see [PSP Porting](#) for details.

File	Description
<code>include/psp_eth_mii.h</code>	Media Independent Interface (MII) header file.
<code>include/psp_eth_phy.h</code>	Ethernet PSP header file.
<code>target/eth/psp_eth_encx24.c</code>	Source code.
<code>target/eth/psp_eth_encx24.h</code>	Header file.



## 3 Configuration Options

Set the system configuration options in the file `src/config/config_ethdriver_encx24.h`. This section lists the available configuration options and their default values.

### **HCC\_ENC24\_LINK\_STA\_POLL\_INTERVAL**

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

### **HCC\_ENC24\_AUTONEG\_TIMEOUT**

The time in ms the driver can wait for auto-negotiation to complete before it times out. The default is 3000.

### **HCC\_ENC24\_BUF\_ALIGN\_PWR\_2**

The buffer alignment as a power of 2. The default is 1.

### **HCC\_ENC24\_MAX\_TX\_BUFD**

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

### **HCC\_ENC24\_USE\_USER\_MAC\_ADDRESS**

Set this to 1 to use a user-defined MAC address or leave the chip's original MAC address untouched. The default is 0.

### **HCC\_ENC24\_MAC\_ADDRESS**

The MAC address of the driver. The default is { 0x00, 0xA0, 0x91, 0x00, 0x92, 0xCD }.

### **HCC\_ENC24\_MAX\_FRAME\_LEN**

The maximum frame length. The default is 346.

### **HCC\_ENC24\_ISR\_ID**

The ISR ID. The default is `PSP_ETH_ENC24_ISR_ID`.

## 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\_encx24\_init

Use this function to initialize the network driver.

#### Format

```
t_nwdriver_ret eth_drv_encx24_init (
    const uint32_t      param,
    t_nwdriver * * const p_ethdriver )
```

#### Arguments

Parameter	Description	Type
param	The driver parameter. This is used as the unit ID passed to all <b>psp_spi_xxx()</b> function calls.	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 *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 **include** /`psp_eth_encx24.c` file.

Function	Description
<code>psp_eth_encx24_init()</code>	Initializes the hardware for the Ethernet driver.
<code>psp_eth_encx24_start()</code>	Starts the driver.
<code>psp_eth_encx24_stop()</code>	Stops the driver.
<code>psp_eth_encx24_delete()</code>	Deletes the driver, releasing associated resources.

These functions are described in the following sections.

## psp\_eth\_encx24\_init

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

### Format

```
t_nwdriver_ret psp_eth_encx24_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\_encx24\_start

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

### Format

```
t_nwdriver_ret psp_eth_encx24_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\_encx24\_stop

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

### Format

```
t_nwdriver_ret psp_eth_encx24_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\_encx24\_delete

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

### Format

```
t_nwdriver_ret psp_eth_encx24_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.