



Network Driver for STR912 Devices User Guide

Version 1.20

For use with STR912 Network Driver versions 2.06 and above

Exported on 11/06/2018

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

1	System Overview.....	3
1.1	Introduction	4
1.2	Feature Check	5
1.3	Device Description	6
1.4	Packages and Documents	7
	Packages.....	7
	Documents	7
1.5	Change History	8
2	Source File List	9
2.1	API Header File	9
2.2	Configuration File.....	9
2.3	Source Code Files.....	9
2.4	Platform Support Package (PSP) files	9
2.5	Version Files.....	10
3	Configuration Options	11
4	Application Programming Interface	12
4.1	str912_eth_drv_init	12
4.2	Error Codes.....	13
5	Integration.....	14
5.1	OS Abstraction Layer	14
5.2	PSP Porting	15
	psp_eth_str912_init.....	16
	psp_eth_str912_start.....	17
	psp_eth_str912_stop.....	18
	psp_eth_str912_delete.....	19

1 System Overview

This chapter contains the fundamental information for this module.

The component sections are as follows:

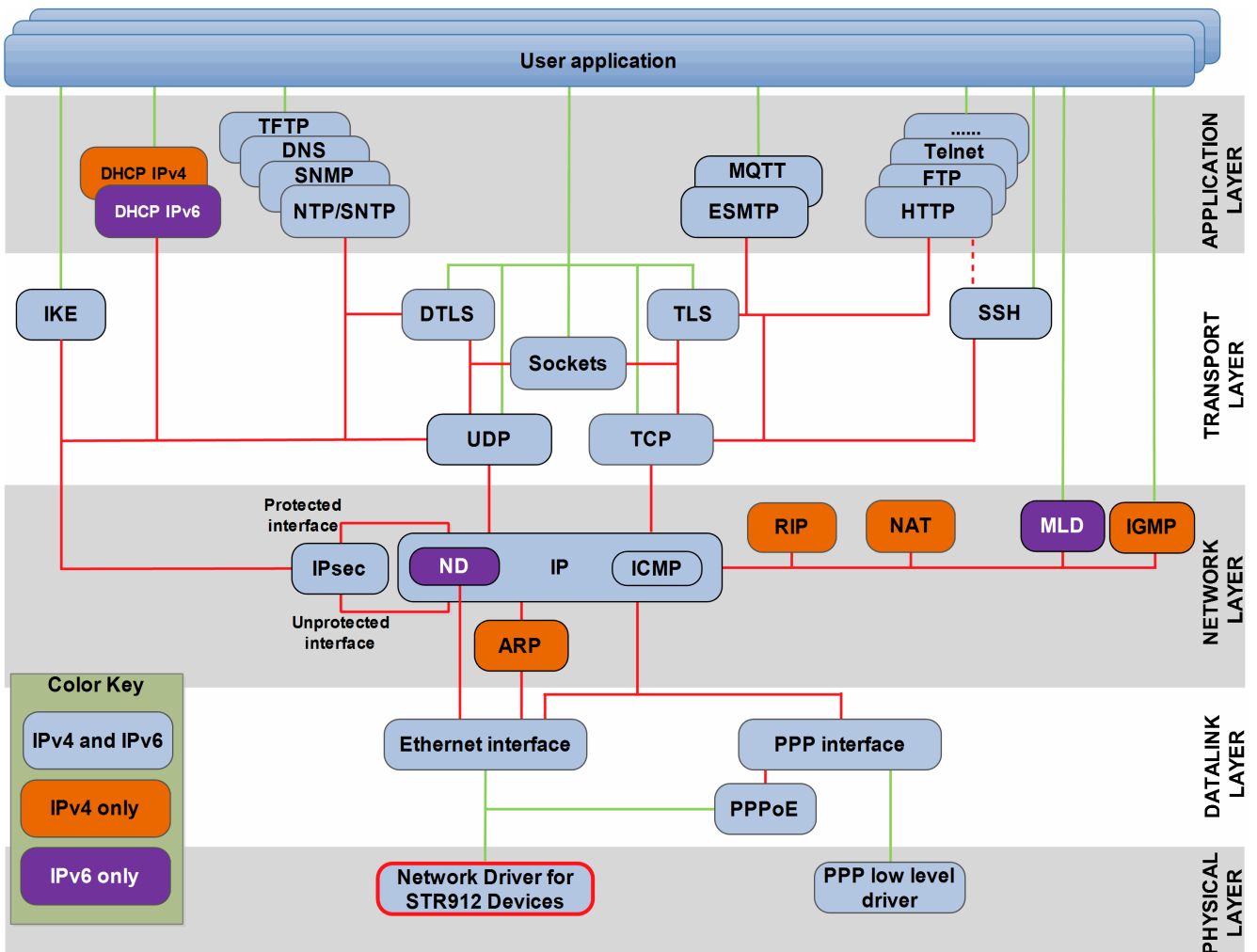
- [Introduction](#) – describes the main elements of the module. This section includes a diagram showing the position of the driver interface within HCC's TCP/IP stack.
- [Feature Check](#) – summarizes the main features of the module as bullet points.
- [Device Description](#) – summarizes the properties of the two Microchip devices.
- [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 [Network Driver PDFs](#).

1.1 Introduction

This guide is for those who want to implement a network driver for STR912xx on-chip ENET Ethernet controllers from STMicroelectronics.

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.
- Fully compatible with the HCC Network Driver Interface specification.
- Supports STR912xx on-chip ENET Ethernet controllers from STMicroelectronics.
- HCC provides fully tested reference drivers for this module.

1.3 Device Description

The driver supports a range of STR912 products. This table summarizes the properties of some of these devices:

	STR912FAW32	STR912FAZ46	STR912FAW47
Flash	256 + 32 Kb	1024 + 128 Kb	2048 + 128 Kb
RAM	64 Kb	96 Kb	96 Kb

1.4 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.
nw_drv_base	The network driver base package. This is the base system on which the STR912 network driver is built.
nw_drv_eth_str912	The network driver for STR912 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 STR912 Devices User Guide

This is this document.

1.5 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_eth_str912](#).

The current version of this manual is 1.20. The full list of versions is as follows:

Manual version	Date	Software version	Reason for change
1.20	2018-11-06	2.06	Corrected values of MAX_RX_DESC and MAX_RX_DESC in <i>Configuration Options</i> .
1.10	2017-06-16	2.06	New <i>Change History</i> format.
1.00	2017-04-25	2.06	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_str912.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 functions, see [Application Programming Interface](#).

2.2 Configuration File

The file `src/config/config_ethdriver_str912.h` contains all the configurable parameters of the system. Configure these as required. For details of these options, see [Configuration Options](#).

2.3 Source Code Files

The following files are in the directory `src/driver/network/ethernet/str912`. **These files should only be modified by HCC.**

File	Description
<code>eth_phy_ste100p.c</code>	Implements ETH PHY functions for STE100P.
<code>eth_phy_ste100p_reg.h</code>	STE100P header file.
<code>eth_str912.c</code>	Implements the Ethernet driver for STR912

2.4 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 for other devices.

Note:

- You must modify these PSP implementations for your specific microcontroller and development board; see [PSP Porting](#) for details.
- In the package these files are offset to avoid overwriting an existing implementation. Copy them to the root **hcc** directory for use.

File	Description
include/psp_eth_mii.h	Media Independent Interface (MII) header file.
include/psp_eth_phy.h	Ethernet PSP header file for STE100P.
target/eth/psp_eth_str912.c	Functions source code.
target/eth/psp_eth_str912.h	Functions header file.
target/include/hcc_str912_reg.h	Registers file.

2.5 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_ethdriver_str912.h	STR912 version number.
ver_psp_eth_mii.h	MII version number.
ver_psp_eth_phy.h	Ethernet PSP version number.
ver_psp_eth_phy_ste100p.h	STE100P version number.
ver_psp_eth_str912.h	Functions files version number.
ver_psp_proc_reg.h	Registers file version number.

3 Configuration Options

Set the system configuration options in the file `src/config/config_ethdriver_str912.h`. This section lists the available options and their default values.

STR912_HCLK_FREQ

The clock frequency. The default value is (`SCU_GetHCLKFreqValue()` * 1000).

STR912_ETHERNET_BUF_SIZE

The Ethernet buffer size. The default value is 16384.

ETH_PHY_DEV_ADDR

The physical device address. The default value is 0x15.

MAX_RX_DESC

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

MAX_TX_DESC

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

MAX_FRAME_SIZE

The maximum frame size. The default value is 1536.

MAC_ADDRESS

The MAC address. The default value is { 0x00, 0xA2, 0x92, 0x00, 0x92, 0x05 }.

LINK_STA_POLLED

Keep the default of 1 to poll link status. If this is set to 0, this is handled using interrupts.

LPC_LINK_STA_POLL_INTERVAL

The poll interval in milliseconds, used if the previous parameter is enabled. The default value is 500.

ETH_PHY_RMII

Keep the default of 1 if RMII mode is used. Otherwise, set it to 0.

RX_FEED_ENABLE

Keep the default of 1 to enable RX feed mode (meaning the IP stack provides RX buffers). Otherwise, set it to 0.

4 Application Programming Interface

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

4.1 str912_eth_drv_init

Use this function to initialize the network driver.

Format

```
t_nwdriver_ret str912_eth_drv_init (  
    const 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 the error codes that may be generated by the API call:

Error code	Value	Meaning
NWDRIVER_SUCCESS	0	Successful execution.
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 module uses the OS Abstraction Layer (OAL) that allows it to run seamlessly with a wide variety of RTOSes, or without an RTOS.

The module uses the following OAL components:

OAL Resource	Number Required
Tasks	0
Mutexes	1
Events	0
ISRs	1

5.2 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 *Platform Support Package (PSP) Base User Guide*.

The module makes use of the following standard PSP macros:

Macro	Package	Element	Description
PSP_RD_LE16	psp_base	psp_endianness	Reads a 16 bit value stored as little-endian from a memory location.
PSP_RD_LE32	psp_base	psp_endianness	Reads a 32 bit value stored as little-endian from a memory location.

The module makes use of the following standard PSP function:

Function	Package	Element	Description
psp_memcpy()	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_str912.c** file.

Function	Description
psp_eth_str912_init()	Initializes the hardware for the Ethernet driver.
psp_eth_str912_start()	Starts the driver.
psp_eth_str912_stop()	Stops the driver.
psp_eth_str912_delete()	Deletes the driver, releasing associated resources.

These functions are described in the following sections.

psp_eth_str912_init

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

Format

```
t_nwdriver_ret psp_eth_str912_init ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_str912_start

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

Format

```
t_nwdriver_ret psp_eth_str912_start ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_str912_stop

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

Format

```
t_nwdriver_ret psp_eth_str912_stop ( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

psp_eth_str912_delete

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

Format

```
t_nwdriver_ret psp_eth_str912_delete( void )
```

Arguments

None.

Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.