



# Network Driver for GMAC MAC Devices User Guide

Version 1.20

For use with Network Driver for GMAC MAC Devices versions 2.03 and above

Exported on 11/08/2018

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

<b>1</b>	<b>System Overview.....</b>	<b>4</b>
1.1	Introduction .....	5
1.2	Feature Check .....	7
1.3	Packages and Documents .....	8
	Packages.....	8
	Documents .....	8
1.4	Change History .....	9
<b>2</b>	<b>Source File List .....</b>	<b>10</b>
2.1	API Header File .....	10
2.2	Configuration File.....	10
2.3	System Files.....	10
2.4	Version Files.....	11
2.5	Platform Support Package (PSP) Files.....	11
	Atmel SAM4E Xplained .....	11
	Atmel ATMSAMA5D3-Xplained.....	12
	Xilinx Zync 7000 and ZedBoard .....	13
<b>3</b>	<b>Configuration Options .....</b>	<b>14</b>
<b>4</b>	<b>Application Programming Interface .....</b>	<b>15</b>
4.1	gmac_eth_drv_init.....	15
4.2	Error Codes.....	16
<b>5</b>	<b>Integration.....</b>	<b>17</b>
5.1	OS Abstraction Layer .....	17
5.2	Utilities.....	17
5.3	PSP Porting .....	18
	Atmel SAMA5D3 Configuration File .....	19
	Xilinx Configuration File.....	20
	psp_gmac_eth_init .....	21
	psp_gmac_eth_start.....	22
	psp_gmac_eth_stop .....	23
	psp_gmac_eth_delete .....	24
	psp_gmac_get_buf .....	25

psp_gmac_set_speed.....	26
psp_isr_install.....	27
psp_isr_delete.....	28
psp_isr_enable.....	29
psp_isr_disable.....	30
psp_int_enable.....	31
psp_int_disable.....	32

# 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.
- [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](#).

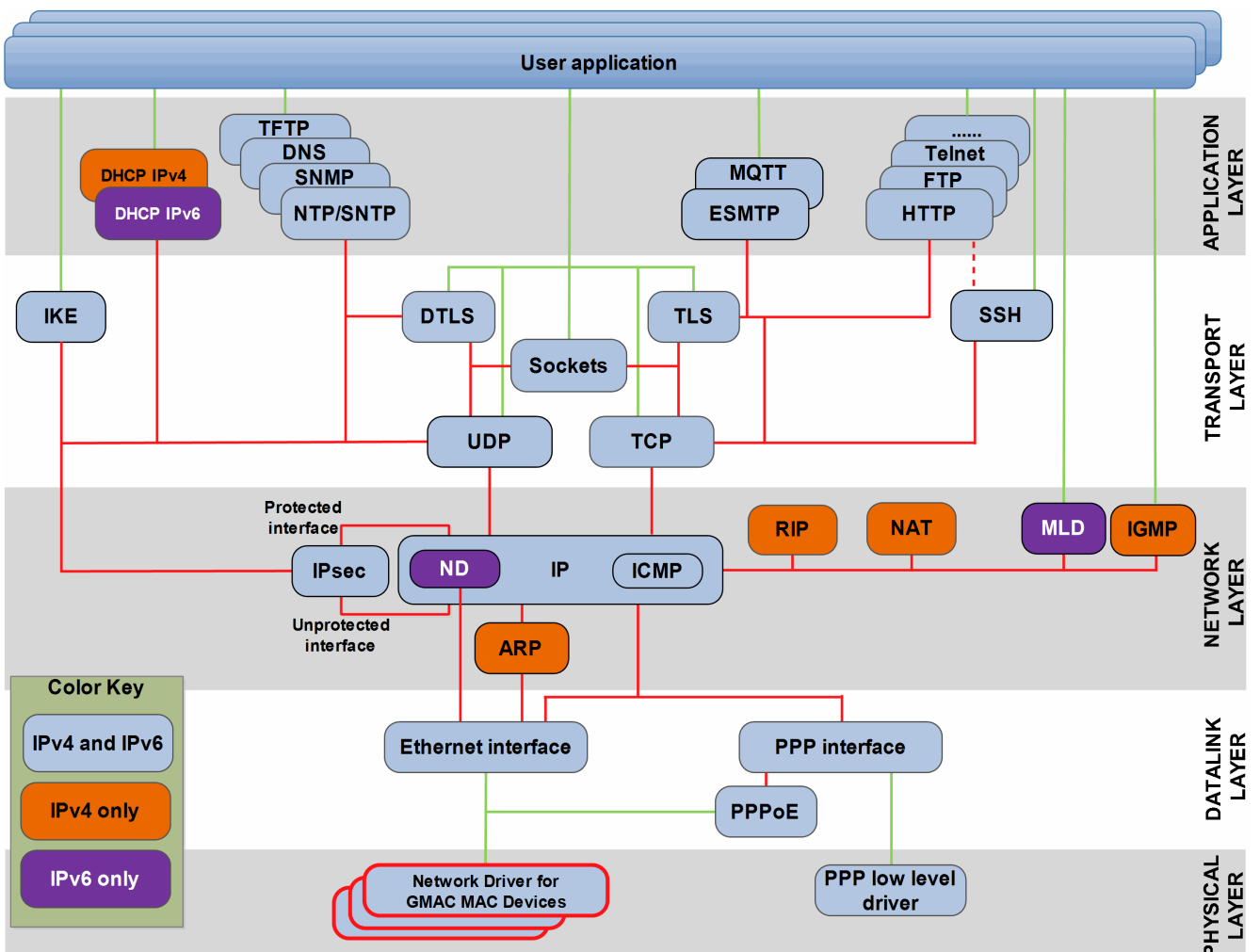
# 1.1 Introduction

This guide is for those who want to implement a network driver for Xilinx® Zynq® and Microchip Technology Inc. Atmel® SMART microcontrollers that have a GMAC MAC core.

This package was tested using the following combinations:

- Atmel® SAM4E devices and the Atmel® SAM4E Xplained evaluation board.
- Atmel® SAMA5D3 devices and the Atmel® ATMSAMA5D3 Xplained evaluation board.
- Xilinx Zynq® 7000 and the ZedBoard evaluation board.

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:**

- The KSZ8031 PHY provided is just a sample; if you need a different PHY, contact HCC Embedded.
- 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 Xilinx<sup>®</sup> Zynq<sup>®</sup> and Microchip Technology Inc. Atmel<sup>®</sup> SMART microcontrollers that have a GMAC MAC core. (The package was tested with Xilinx<sup>®</sup> Zynq<sup>®</sup> 7000 and Atmel<sup>®</sup> SAM4E and SAMA5D3 devices.)
- HCC provides fully tested reference drivers for this module.

## 1.3 Packages and Documents

### Packages

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

Package	Description
<b>hcc_base_doc</b>	This contains the two guides that will help you get started.
<b>nw_drv_base</b>	The network driver base package. This is the base system on which this driver is built.
<b>nw_drv_eth_gmac</b>	The Network Driver for GMAC MAC Devices package described in this document.
<b>oal_base</b>	The OS Abstraction Layer (OAL) package.
<b>psp_template_base</b>	The base Platform Support Package (PSP).
<b>mutil_timer</b>	The MISRA-compliant timer utility.

### 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 GMAC MAC Devices User Guide**

This is this document.



## 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 [Network Driver PDFs](#).
- For the history of changes made to the package code itself, see [History: nw\\_drv\\_eth\\_gmac](#).

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	2018-11-08	2.03	Added references to the ZedBoard evaluation board to <i>Introduction</i> and <i>Source Files</i> . Added <i>Xilinx Configuration File</i> .
1.10	2018-10-21	2.02	Added Atmel® ATMSAMA5D3 Xplained evaluation board to <i>Introduction</i> and <i>Source Files</i> . Added ISR and INT functions for ATMSAMA5D3 to <i>PSP Porting</i> .
1.00	2018-06-11	2.01	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_gmac.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 `gmac_eth_drv_init` function.

### 2.2 Configuration File

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

### 2.3 System Files

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

File	Description
<code>eth_gmac.c</code>	Ethernet driver code.
<code>eth_phy_ksz8031.c</code>	KSZ8031 Ethernet PHY driver code.
<code>eth_phy_ksz8031_reg.h</code>	KSZ8031 registers header file.

**Note:** The files for the KSZ8031 Ethernet PHY are provided as a tested sample. Others can be provided on request; contact HCC for details.

## 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
<b>ver_ethdriver_gmac.h</b>	Ethernet driver version number.
<b>ver_eth_phy_ksz8031.h</b>	KSZ8031 PHY driver version number.

## 2.5 Platform Support Package (PSP) Files

These files provide functions the core code needs to call, depending on the hardware.

**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.

### Atmel SAM4E Xplained

The following implementation files for the Atmel® SAM4E Xplained evaluation board are in the directory **src/psp\_atsam4e\_xplained**.

File	Description
<b>include/psp_eth_mii.h</b>	Media Independent Interface (MII) header file.
<b>include/psp_eth_phy.h</b>	Ethernet PHY header file.
<b>target/eth/psp_eth_gmac.h</b>	PSP functions header file.
<b>target/eth/psp_eth_gmac.c</b>	PSP functions source code.

The PSP also contains the following version files in the directory **src/version**:

File	Description
<b>ver_psp_eth_gmac.h</b>	PSP version.
<b>ver_psp_eth_mii.h</b>	MII version.

File	Description
<b>ver_psp_eth_phy.h</b>	Ethernet PHY version.

## Atmel ATMSAMA5D3-Xplained

The following implementation files for the Atmel® ATMSAMA5D3 Xplained evaluation board are in the directory **src/psp\_atsama5d3\_xplained**.

File	Description
<b>config/config_eth_phy_ksz9xxx.h</b>	Configuration file for KSZ9xxx devices.
<b>config/config_ethdriver_gmac.h</b>	This has the same options as the main configuration file, but with <a href="#">defaults appropriate to this device</a> .

The following files are in the directory **psp**:

File	Description
<b>include/psp_eth_mii.h</b>	Media Independent Interface (MII) header file.
<b>include/psp_eth_phy.h</b>	Ethernet PHY header file.
<b>target/eth/psp_eth_gmac.h</b>	PSP functions header file.
<b>target/eth/psp_eth_gmac.c</b>	PSP functions source code.
<b>target/isr/psp_isr.h</b>	ISR functions header file.
<b>target/isr/psp_isr.c</b>	ISR functions source code.

The PSP also contains the following version files in the directory **src/version**:

File	Description
<b>ver_psp.h</b>	PSP version.
<b>ver_psp_eth_gmac.h</b>	GMAC PSP version.
<b>ver_psp_eth_mii.h</b>	MII version.
<b>ver_psp_eth_phy.h</b>	Ethernet PHY version.
<b>ver_psp_eth_phy_ksz9xxx.h</b>	KSZ9xxx Ethernet PHY version.
<b>ver_psp_isr.h</b>	ISR version.

## Xilinx Zync 7000 and ZedBoard

The following implementation files for the Xilinx<sup>®</sup> Zync<sup>®</sup> 7000 and the ZedBoard are in the directory **src/psp\_zynq7000\_zedboard**:

File	Description
<b>config/config_ethdriver_gmac.h</b>	This has the same options as the main configuration file, but with <a href="#">defaults appropriate to this device</a> .
<b>config/config_psp_eth_gmac.h</b>	Use this to set an RMI interface instead of MII.

The following files are in the directory **psp**:

File	Description
<b>include/psp_eth_mii.h</b>	Media Independent Interface (MII) header file.
<b>include/psp_eth_phy.h</b>	Ethernet PHY header file.
<b>target/eth/psp_eth_gmac.h</b>	PSP functions header file.
<b>target/eth/psp_eth_gmac.c</b>	PSP functions source code.
<b>target/include/component_gmac.h</b>	Atmel API definition for GMAC.
<b>target/isr/psp_isr.h</b>	ISR functions header file.
<b>target/isr/psp_isr.c</b>	ISR functions source code.

The PSP also contains the following version files in the directory **src/version**:

File	Description
<b>ver_psp.h</b>	PSP version.
<b>ver_psp_eth_gmac.h</b>	GMAC PSP version.
<b>ver_psp_eth_mii.h</b>	MII version.
<b>ver_psp_eth_phy.h</b>	Ethernet PHY version.
<b>ver_psp_isr.h</b>	ISR version.

## 3 Configuration Options

Set the system configuration options in the file `src/config/config_ethdriver_gmac.h`, as described below. This section lists the available options and their default values.

### **GMAC\_ETHERNET\_BUF\_SIZE**

The Ethernet buffer size. The default is 16384.

### **ETH\_ISR\_ID**

The interrupt ID, GMAC\_IRQn. The default value is 44.

### **ETH\_ISR\_PRIO**

The Link ISR priority, defined in case it is used. The default value is 5.

### **ETH\_PHY\_DEV\_ADDR**

The External PHY address. The default value is 0x00.

### **ETH\_LINK\_STA\_POLL\_INTERVAL**

The poll interval in milliseconds. The default value is 100.

### **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, 0xA0, 0x91, 0xFB, 0x93, 0xCD }.

## 4 Application Programming Interface

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

### 4.1 gmac\_eth\_drv\_init

Use this function to initialize the network driver.

#### Format

```
t_nwdriver_ret gmac_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 the error codes that may be generated by the API and PSP functions:

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
<b>psp_membar()</b>	psp_base	psp_membar	Memory barrier.
<b>psp_memcpy()</b>	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\_gmac.c** file. The ISR and INT functions are in the **psp/target/isr/psp\_isr.c** file.

Function	Description
<b>psp_gmac_eth_init()</b>	Initializes the hardware for the Ethernet driver.
<b>psp_gmac_eth_start()</b>	Starts the driver.
<b>psp_gmac_eth_stop()</b>	Stops the driver.
<b>psp_gmac_eth_delete()</b>	Deletes the driver, releasing associated resources.
<b>psp_gmac_get_buf()</b>	Gets the address of the Ethernet buffer.
<b>psp_gmac_set_speed()</b>	Sets the Ethernet speed.

The following are only for use with the Atmel<sup>®</sup> ATMSAMA5D3 Xplained and ZedBoard evaluation boards:

<b>psp_isr_install()</b>	Installs an ISR.
<b>psp_isr_delete()</b>	Deletes an ISR.
<b>psp_isr_enable()</b>	Enables an ISR.
<b>psp_isr_disable()</b>	Disables an ISR.
<b>psp_int_enable()</b>	Enables global interrupts.
<b>psp_int_disable()</b>	Disables global interrupts.

These functions are described in the sections below that follow the two configuration files.

## Atmel SAMA5D3 Configuration File

The file **src/config/config\_ethdriver\_gmac.h**, in the **psp\_atsama5d3\_xplained** directory has its own configuration options file with default values appropriate for the device. This section lists the available options and their default values.

### **GMAC\_ETHERNET\_BUF\_SIZE**

The Ethernet buffer size. The default is (16384 \* 10).

### **ETH\_ISR\_ID**

The interrupt ID, GMAC0. The default value is 34.

### **ETH\_ISR\_PRIO**

The Link ISR priority. Five priority bits are implemented, so this is the lowest priority. The default value is 240.

### **ETH\_PHY\_DEV\_ADDR**

The External PHY address. The default value is 0x07.

### **ETH\_LINK\_STA\_POLL\_INTERVAL**

The poll interval in milliseconds. The default value is 100.

### **MAX\_RX\_DESC**

The maximum number of RX descriptors. The default value is 40.

### **MAX\_TX\_DESC**

The maximum number of TX descriptors. The default value is 40.

### **MAC\_ADDRESS**

The MAC address of the driver. The default value is { 0x00, 0xA0, 0x91, 0xFB, 0x93, 0xCD }.

## Xilinx Configuration File

The file **src/config/config\_ethdriver\_gmac.h**, in the **psp\_zynq7000\_zedboard** directory has its own configuration options file with default values appropriate for the device. This section lists the available options and their default values.

### **GMAC\_ETHERNET\_BUF\_SIZE**

The Ethernet buffer size. The default is 262144.

### **ETH\_ISR\_ID**

The interrupt ID, GMAC0. The default value is 61.

### **ETH\_ISR\_PRIO**

The Link ISR priority. Five priority bits are implemented, so this is the lowest priority. The default value is ( `configSYSTEM_INTERRUPT_PRIORITY` ).

### **ETH\_PHY\_DEV\_ADDR**

The External PHY address. The default value is 0x00.

### **ETH\_LINK\_STA\_POLL\_INTERVAL**

The poll interval in milliseconds. The default value is 100.

### **MAX\_RX\_DESC**

The maximum number of RX descriptors. The default value is 64.

### **MAX\_TX\_DESC**

The maximum number of TX descriptors. The default value is 64.

### **MAC\_ADDRESS**

The MAC address of the driver. The default value is { 0x00, 0xA0, 0x91, 0xFB, 0x93, 0xCD }.

## psp\_gmac\_eth\_init

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

### Format

```
t_nwdriver_ret psp_gmac_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\_gmac\_eth\_start

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

### Format

```
t_nwdriver_ret psp_gmac_eth_start ( void )
```

### Arguments

None.

### Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

## psp\_gmac\_eth\_stop

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

### Format

```
t_nwdriver_ret psp_gmac_eth_stop ( void )
```

### Arguments

None.

### Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

## psp\_gmac\_eth\_delete

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

### Format

```
t_nwdriver_ret psp_gmac_eth_delete ( void )
```

### Arguments

None.

### Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.



## psp\_gmac\_get\_buf

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

### Format

```
t_nwdriver_ret psp_gmac_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.

## psp\_gmac\_set\_speed

This function is provided by the PSP to set the Ethernet speed to 10 Mbps, 100 Mbps, or 1000 Mbps.

### Format

```
t_nwdriver_ret psp_gmac_set_speed ( uint32_t speed )
```

### Arguments

Parameter	Description	Type
speed	The speed to use.	uint32_t

### Return Values

Return value	Description
NWDRIVER_SUCCESS	Successful execution.
NWDRIVER_ERROR	Operation failed.

## psp\_isr\_install

This function is provided by the PSP to install an ISR.

### Format

```
int psp_isr_install (
    const oal_isr_dsc_t *   isr_dsc,
    oal_isr_id_t *         isr_id )
```

### Arguments

Parameter	Description	Type
isr_dsc	A pointer to the ISR descriptor.	oal_isr_dsc_t *
isr_id	On return, a pointer to the ISR ID.	oal_isr_id_t *

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

## psp\_isr\_delete

This function is provided by the PSP to delete an ISR.

### Format

```
int psp_isr_delete ( oal_isr_id_t isr_id )
```

### Arguments

Parameter	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

## psp\_isr\_enable

This function is provided by the PSP to enable an ISR.

### Format

```
int psp_isr_enable ( oal_isr_id_t isr_id )
```

### Arguments

Parameter	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

## psp\_isr\_disable

This function is provided by the PSP to disable an ISR.

### Format

```
int psp_isr_disable ( oal_isr_id_t isr_id )
```

### Arguments

Parameter	Description	Type
isr_id	The ISR ID.	oal_isr_id_t

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

## psp\_int\_enable

This function is provided by the PSP to enable global interrupts.

### Format

```
int psp_int_enable ( void )
```

### Arguments

None.

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.

## psp\_int\_disable

This function is provided by the PSP to disable global interrupts.

### Format

```
int psp_int_disable ( void )
```

### Arguments

None.

### Return Values

Return value	Description
OAL_SUCCESS	Successful execution.
OAL_ERROR	Operation failed.