ETH internal peripheral

Stable: 11.02.2019 - 13:21 / Revision: 15.01.2019 - 11:22

Contents

1 Article purpose

The purpose of this article is to:

- briefly introduce the Ethernet peripheral and its main features
- indicate the level of security supported by this hardware block
- explain how each instance can be allocated to the two runtime contexts and linked to the corresponding software components
- explain, when necessary, how to configure the Ethernet peripheral.

2 Peripheral overview

The Ethernet peripheral (ETH) is based on Synopsys DesignWare® Ethernet GMAC IP, which enables the host to communicate data using the Gigabit Ethernet protocol (IEEE 802.3) at 10, 100 and 1000 Mbps.
The peripheral is composed of three main layers: the gigabit ethernet media access controller (GMAC), the MAC transaction layer (MTL), and the MAC DMA controller (MDC). The driver used to drive the ETH is Stmmac.

2.1 Features

The Ethernet peripheral main features are the following:

- Compliance with IEEE 802.3 specifications
- Support for IEEE 1588-2002 and IEEE 1588-2008 standards for precision networked clock synchronization

  IEEE 802.3-az for Energy Efficient Ethernet (EEE)
  IEEE 802.3x flow control automatic transmission of zero-quanta pause frame on flow control input de-assertion.
  IEEE 802.1Q VLAN tag detection for reception frames
  AMBA 2.0 for AHB Master/Slave ports and AMBA 3.0 for AXI Master/Slave ports

- Configurability allowing to support data transfer rates of 10/100/100 Mbps, 10/100 Mbps only or 1000 Mbps only
- Support for multiple TCP/IP offload functions

Refer to STM32MP15 reference manuals for the complete features list, and to the software components, introduced below, to see which features are implemented.

2.2 Security support

The ETH is a **non-secure** peripheral.

3 Peripheral usage and associated software

3.1 Boot time

The Ethernet peripheral can be used at boot time by SSBL (by UBoot with tftp protocol for image loading). See How_to_boot_the_kernel_via_TFTP_from_U-Boot for more details.

3.2 Runtime

3.2.1 Overview

The Ethernet peripheral can be allocated to the Arm® Cortex®-A7 non-secure core to be controlled in Linux® by the NetDev Framework.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
		Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	

Networking	ETH		Linux netdev/ethernet framework		

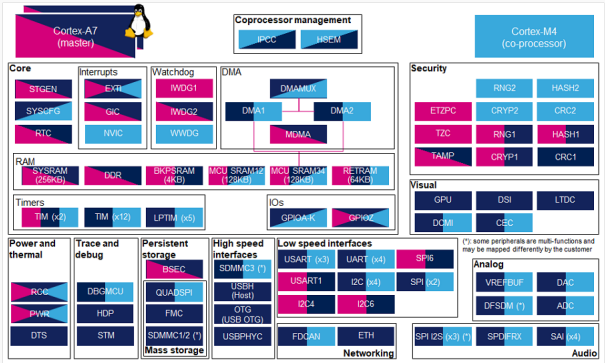
3.2.3 Peripheral configuration

The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article. When the Ethernet peripheral is assigned to the Linux® OS, it is configured through the device tree according to the information given in the Ethernet device tree configuration article.

3.2.4 Peripheral assignment

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- ☐ means that the peripheral can be assigned (☑) to the given runtime context.
- ✓ is used for system peripherals that cannot be unchecked because they are statically connected in the device.



Internal peripherals

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals.

Domain	Peripheral	Runtime allocation				Comment
		Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)	
Networking	ETH	ETH		☐		Assignment (single choice)

4 How to go further

Use this paragraph to add more information and introduce other documentation such as Application Notes (AN)

5 References

Ethernet

Direct Memory Access

Advanced High-performance Bus

Second Stage Boot Loader

Open Portable Trusted Execution Environment

Operating System