VECTOR

MICROSAR Ethernet Driver

Technical Reference

Infineon AURIX TriCore Tc3xx Ethernet Controller IP

Version 3.0.1

Authors	Benjamin Groebner
Status	Released

Document Information

History

Author	Date	Version	Remarks
Benjamin Groebner	2017-08-24	0.1.0	Creation of document
Benjamin Groebner	2018-01-04	0.2.0	TASK-60515 - Provide DrvETh_Tc3xx Beta version using TriBoard TC3X9 V2.0
Benjamin Groebner	2018-02-26	1.0.0	TASK-65966 - Review Integration and Core Update
Benjamin Groebner	2018-05-23	2.0.0	FEAT-3644 - Support QoS/FQTSS by DrvEth_Tc3xxEthAsr
Benjamin Groebner	2019-03-20	3.0.0	▶ ESCAN00101437 Possible memory violation in case of multi Eth driver configuration ▶ ESCAN00101808 Compiler warning: Calling function without prototype Appl_UnlockEndinit / Appl_LockEndinit ▶ ESCAN00101709 Missing overflow handling during global time retrieval
Benjamin Groebner	2019-04-04	3.0.1	▶ ESCAN00102625 Possible loss of interrupt could lead to stuck communication on Ethernet ▶ ESCAN00101663 Memory corruption possible if RX buffers are configured smaller than 1522 bytes

Reference Documents

No.	Source	Title	Version
[1]	AUTOSAR	AUTOSAR_SWS_EthernetDriver.pdf	4.1.1
[2]	Vector	TechnicalReference_Eth_Core.pdf	see delivery
[3]	Infineon	AURIX TC3xx User's Manual Part 1+2	1.0.0
[4]	Infineon	TC38x User's Manual Appendix	2.3.0
[5]	Vector	AN-ISC-8-1172_TriCore_ENDINIT_protection_handling.pdf	see delivery

Scope of the Document

This technical reference describes the specific use of the specific controller driver software. It supplements the general Ethernet driver technical reference [2].

Caution
We have configured the programs in accordance with your specifications in the questionnaire. Whereas the programs do support other configurations than the one specified in your questionnaire, Vector´s release of the programs delivered to your company is expressly restricted to the configuration you have specified in the questionnaire.

Contents

Illustrations

Tables

1 Component History

The component history gives an overview over the important milestones that are supported in the different versions of the component.

Component Version	New Features
0.01.00	Initial release of component (Beta)
0.02.00	Beta version supporting TriBoard TC3X9 V2.0
1.00.xx	QM release after review integration and core update
2.00.xx	FEAT-3644: Support QoS/FQTSS
3.00.xx	▶ Updated to DrvEth__coreEthAsr version 3.08.01 ▶ Support for centralized Eth_GeneralTypes.h (MSR4-R22)

Table 1-1 Component history

2 Introduction

This document describes the specific functionality of the driver either limiting or extending the function set described in [2].

In addition to that specific integration information for the driver itself and specific supported derivatives and use-cases beyond the general ones described in [2] are given.

3 Hardware Overview

The following table summarizes the supported hardware platforms and settings this Ethernet Driver can be used on. It also provides information on the documentation used during the development of the driver for detailed reference to hardware specific information.

Derivatives	TC3xx: TC38x, TC39x_StepB (NOT TC39x_StepA !)
Compiler	Tasking, GreenHills
Hardware manuals	[3] [4]
Errata sheets	-
Safety manuals	-
Specifics	-
Used registers (offset to base register offset)	Register offset < 0x2044 Details: Refer to the defines in Eth_30_Tc3xx_LL_regs.h
Operating modes	Refer to the states specified in the CAD
Hardware features related to independence or partitioning	None
Access mechanism	Memory mapped registers
Hardware diagnostics	none

Table 3-1 Hardware Overview

4 Functional Description

4.1 Features

The features listed in the following tables cover the specific functionality specified for the Eth_30_Tc3xx differing from the one defined in [2].

The following table gives a short overview over the functionality provided by the driver regarding [2]. For deviations and limitation see the following subsections.

Feature	Driver/Hardware support - = not available ■ = available on specific derivatives ■ = available
Time Synchronization	■
Quality of Service	■
FQTSS (Traffic Shaping)	■
Checksum Offloading	■
Receive Buffer Segmentation	- (not supported by this driver)

Table 4-1 Vector Driver Feature Set Support

4.1.1 Deviations

There are no deviations.

4.1.2 Additions/ Extensions

There are no additions or extensions.

4.1.3 Limitations

There are no limitations.

4.2 Initialization

In contrast to [2] the driver needs following specific initialization procedure:

During the execution of Eth_30_Tc3xx_Init() the Ethernet driver needs to configure the Ethernet clock register (ETH_CLC).

To be able to perform the configuration, the user must guarantee full write access to this register, which is protected by the watchdog-control-register. The user has to provide two functions to disable (Appl_UnlockEndinit()) and afterwards enable (Appl_LockEndinit()) the write protection by accessing the watchdog register. Please refer to Application Note [5] for details.

4.3 States

The driver doesn’t provide specific states.

4.4 Main Functions

The driver doesn’t provide additional main functions.

4.5 Error Handling

4.5.1 Development Error Reporting

The driver doesn’t provide additional development error reporting functionality.

4.5.2 Production Code Error Reporting

The driver doesn’t provide additional production error reporting functionality.

4.6 Checksum Offloading

The driver/hardware can offload the checksum calculation of following protocols:

Protocol	Driver/Hardware support - = not available ■ = available
ICMPv4	■
IPv4	■
UDP over IPv4	■
TCP over IPv4	■
ICMPv6	■
IPv6 with Routing Header Option	-
IPv6 with Destination Header Option	-
IPv6 with Hop by Hop Header Option	-
UDP over IPv6	■
TCP over IPv6	■

Table 4-2 Checksum Offloading Capability

4.7 Quality of Service (QoS)

Quality of Service is supported for transmission and reception of frames. For frame transmission, all four queues provided by the hardware are used. When a frame is sent, it is sorted into the corresponding transmission queue based on the VLAN-priority, before a DMA-request is triggered. When a frame is received, its VLAN-priority is analyzed by a separation filter in hardware and sorted into the corresponding reception queue. Queues with a higher priority are processed before queues with lower priority.

Note
Quality of Service can be applied on VLAN tagged traffic only because it relies on the priority encoded in the VLAN TCI (Tag Control Information).

4.7.1 Queue Priorities

The supported queue priorities can be identified in the configuration tool by the parameter `Priority` (see Table 4-3) and are assigned to the transmission queues as follows:

Queue	Priority
Q0	0
Q1	1
Q2	2
Q3	3

Table 4-3 TX queue to ring priority mapping

For receiving frames, the following priorities are assigned to the available reception queues:

Queue	Priority
Q0	0
Q1	1
Q2	2
Q3	3

Table 4-4 Rx queue to ring priority mapping

4.7.2 VLAN Priority Mapping

The driver provides the possibility to map VLAN-priorities by configuration to the transmission- and reception queues described in section 4.7.1. This means that for example after mapping VLAN-priority 7 to reception queue 3, all frames holding this VLAN-priority will be received with the highest priority. There is no further restriction to the mapping besides that each VLAN-Priority (0-7) must be assigned and that every queue except the queue with priority 0 needs an assigned VLAN-Priority.

A detailed description of how to configure the queues is not necessary because DaVinci Configurator completely guides through the setup.

4.8 FQTSS (Traffic Shaping)

The Tricore GETH contains three Traffic Shaping units. Table 4-5 shows which queues are available for traffic shaping.

Queue	Traffic Shaping Availability - = not available ■ =available
Q0 (Best effort)	-
Q1 (Network Control)	■
Q2 (Class B)	■
Q3 (Class A)	■

Table 4-5 Hardware support for Traffic Shaping

Figure 4-1 shows an example configuration, where a Bandwidth Limit of 12 000 000 Bit/s is configured for the third transmission queue. This is 1.2% of the Port Transmit Rate of 1000Mbit/s, which is configured in the transceiver settings. The type of the hardware queue with priority 3 is Traffic Class A. This information is hardware dependent and is set by DaVinci Configurator automatically. In this example VLAN priority 6 and 7 are mapped to the queue with priority 3. Thereby this traffic shaper is active for all traffic sent using VLAN priority 6 or 7, which are handled in queue 3.

Protocols like SRP (Stream Reservation Protocol) in an AVB context need this feature to manage the reserved streams.

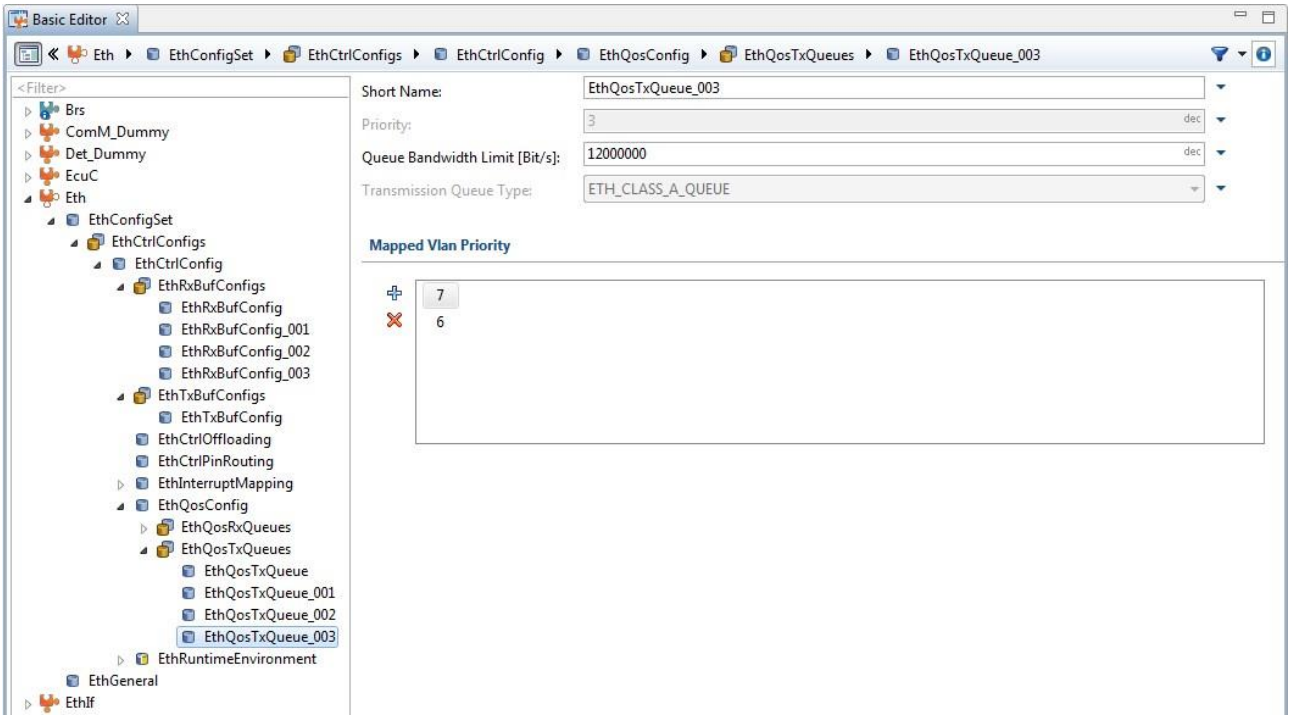

Figure 4-1 Example configuration for Traffic Shaping

5 Integration

This chapter gives necessary information for the integration of the MICROSAR Eth_30_Tc3xx into an application environment of an ECU.

5.1 Scope of Delivery

The delivery of the Eth_30_Tc3xx in any case contains the files which are described in see [2].

The chapters 5.1.1 and 5.1.2 define the additional files that are delivered for this specific driver:

5.1.1 Static Files

The driver doesn’t provide additional static files.

5.1.2 Dynamic Files

The driver doesn’t provide additional dynamic files.

5.2 Exclusive Areas

The driver doesn’t provide additional exclusive areas.

5.3 Memory Mapping

This section describes the specific memory mapping that must be applied for the buffers and descriptors of the driver.

Derivative	Section	Alignment	Memory
TC3xx	Buffer-Section	16 Byte	Scratch-Pad-RAM
	Descriptor-Section	16 Byte	System-RAM

Table 5-1 Memory Mapping

Caution
Previous table can’t be assumed to be applicable for any derivative supported by the driver.

Please see the derivatives respective reference manual to ensure that no specific restrictions apply for it.

5.4 Interrupts

5.4.1 Interrupt Mapping

This section describes the mapping between the interrupt events given by the Ethernet Controller and used by the driver and their respective mapping to interrupt sources of the Interrupt Controller integrated on the derivative.

Derivative	Interrupt Source	Interrupt Event
TC3xx	Ethernet Interrupt Queue 0 Receive	Ethernet RX on Queue 0
	Ethernet Interrupt Queue 0 Transmit	Ethernet TX on Queue 0
	Ethernet Interrupt Queue 1 Receive	Ethernet RX on Queue 1
	Ethernet Interrupt Queue 1 Transmit	Ethernet TX on Queue 1
	Ethernet Interrupt Queue 2 Receive	Ethernet RX on Queue 2
	Ethernet Interrupt Queue 2 Transmit	Ethernet TX on Queue 2
	Ethernet Interrupt Queue 3 Receive	Ethernet RX on Queue 3
	Ethernet Interrupt Queue 3 Transmit	Ethernet TX on Queue 3

Table 5-2 Interrupt Mapping

Note
Previous table doesn't contain all supported derivatives.

If the used derivative isn't listed in the table, please see the respective reference manual of the derivative to retrieve an applicable mapping.

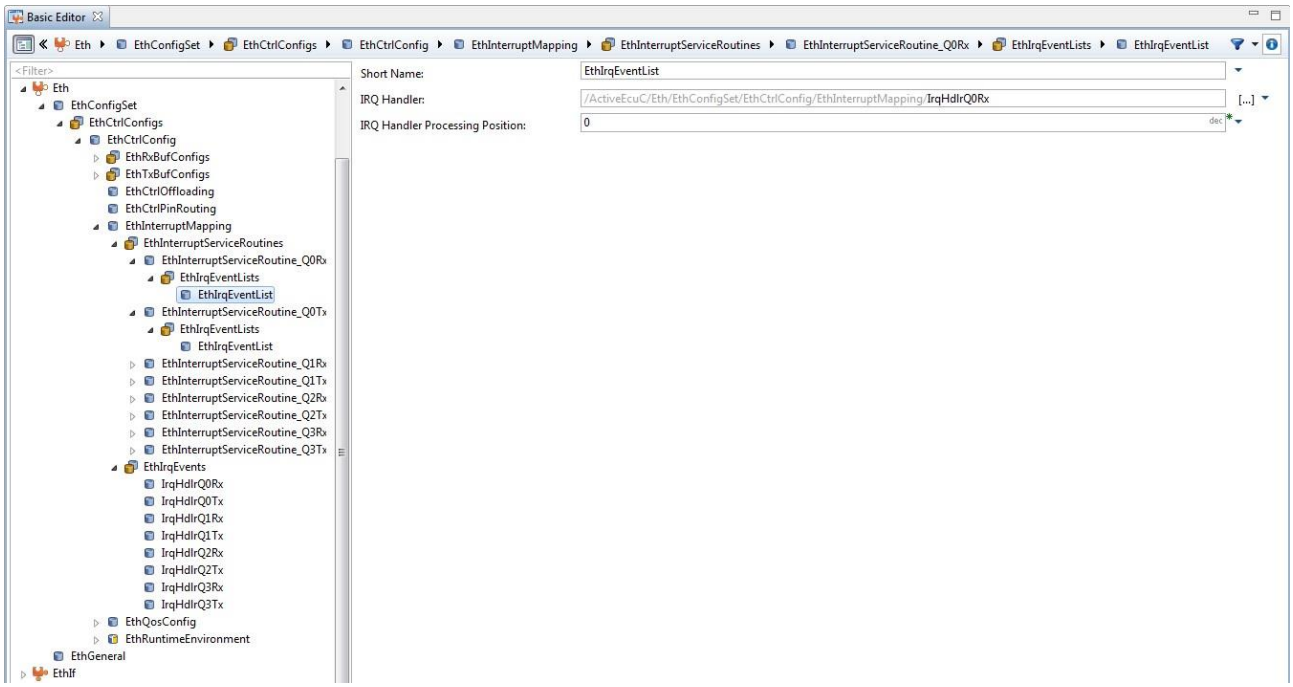

Figure 5-1 Interrupt Mapping in DaVinci Configurator Pro

5.4.2 Interrupt Prioritization

The driver doesn't need specific prioritization of interrupts.

Caution
Regarding the upper layers of the MICROSAR Stack (TcpIp, SoAd) it must be ensured that the Receive -Interrupts of the available QoS queues are all mapped to the same priority so they aren't able to interrupt each other. (Same requirement applies to the Transmit-Interrupts)

Caution
Previous table can't be assumed to be applicable for any derivative supported by the driver.

Please see the derivatives respective reference manual to ensure that no specific restrictions apply for it.

5.5 Operational Preconditions

This section describes the preconditions that must be fulfilled for either a specific use-case or a specific derivative for the driver to be operational after module initialization.

5.5.1 ETH_GPCTL register and port configuration

The TriCore Ethernet MAC provides different general purpose I/O lines that can be chosen alternatively. Depending on the I/O line and the kind of interface (RGMII, RMII or MII), the ports and the ETH_GPCTL register must be adapted by the user.

This can be done in the DaVinci Configurator Pro.

For Infineon Triboards with TC389 the following settings need to be applied:

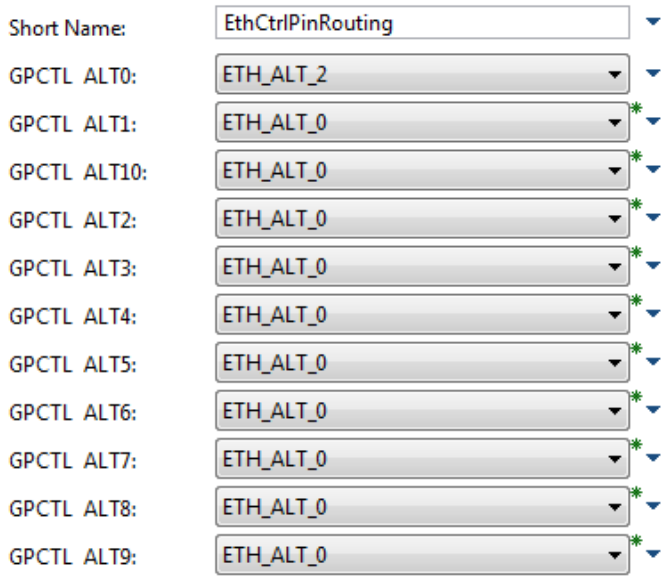

Figure 5-2 GPCTL – Pin Routing

6 API Description

The driver doesn’t extend the API.

7 Configuration

The driver doesn’t need specific configuration.

8 Glossary and Abbreviations

8.1 Glossary

Term	Description
DaVinci Configurator PRO	Generation tool for MICROSAR basis software

Table 8-1 Glossary

8.2 Abbreviations

Abbreviation	Description
API	Application Programming Interface
AUTOSAR	Automotive Open System Architecture
BSW	Basis Software
ECU	Electronic Control Unit
IP	Intellectual Property – Used to describe a specific Ethernet MAC of a vendor
ISR	Interrupt Service Routine
MICROSAR	Microcontroller Open System Architecture (the Vector AUTOSAR solution)

Table 8-2 Abbreviations

9 Contact

Visit our website for more information on

- News
- Products
- Demo software
- Support
- Training data
- Addresses

www.vector.com