USBH internal peripheral

Contents

Stable: 25.09.2020 - 09:43 / Revision: 25.09.2020 - 09:37

Stable: 25.06.2020 - 00.43.1 Revision: 25.06.2020 - 00.37

Contents

1 Article purpose

The purpose of this article is to

- briefly introduce the USBH peripheral and its main features
- indicate the level of security supported by this hardware block
- explain how it can be allocated to the three runtime contexts and linked to the corresponding software components
- explain, when needed, how to configure the USBH peripheral.

2 Peripheral overview

The **USBH** peripheral is used to interconnect other systems with STM32 MPU devices, using USB standard.

2.1 Features

The **USBH** peripheral is a USB Host controller supporting high-speed (480 Mbit/s) using an **EHCI** controller, and full- and low-speeds (12 and 1.5 Mbit/s) through **OHCI** controller.
The **USBH** peripheral has two physical ports providing a **UTMI+** physical interface, mapped on an on-chip 2-port high-speed UTMI+ PHY.
It supports the standard registers used for low- and full-speed operation (**OHCI** model) and high-speed operation (**EHCI** model) and the power management feature called Link Power Management (LPM).

The supported standards are:

- *Universal Serial Bus Revision 2.0 Specification*[1], Revision 2.0, April 27, 2000
- *USB 2.0 Link Power Management Addendum Engineering Change Notice to the USB 2.0 specification*[2], July 16, 2007
- *Enhanced Host Controller Interface Specification for Universal Serial Bus*[3], Revision 1.0, March 12, 2002
- *EHCI v1.1 Addendum*[4], August 2008
- *Open Host Controller Interface Specification for USB*[5], Release 1.0a, September 14, 1999
- *USB 2.0 Transceiver Macrocell Interface (UTMI) Specification*[6], Version 1.05, March 29, 2001
- *UTMI+ Specification*[7], Revision 1.0, February 25, 2004

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

2.2 Security support

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

3 Peripheral usage and associated software

3.1 Boot time

The **USBH** peripheral is usually not used at boot time. But it may be used by the SSBL (see Boot chains overview), for example to boot a kernel stored on a USB key (mass storage).

3.2 Runtime

3.2.1 Overview

The **USBH** peripheral can be allocated to the Arm® Cortex®-A7 non-secure core to be used under Linux® with USB framework.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
High speed interface	USBH (USB Host)		Linux USB 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 by itself can be performed via the STM32CubeMX tool for all internal peripherals. It can then be manually completed (especially for external peripherals) according to the information given in the corresponding software framework article.

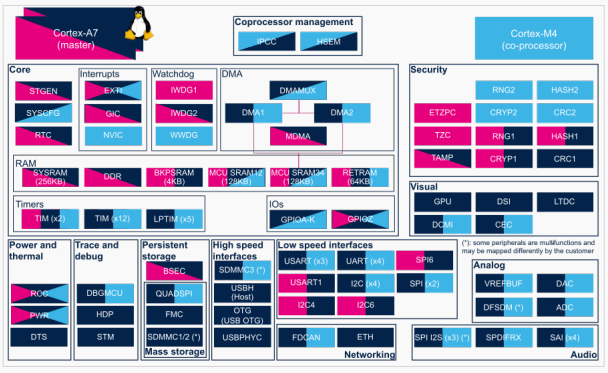
For Linux kernel configuration, please refer to USBH device tree configuration.

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.

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
.



Internal peripherals

Domain	Peripheral	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
High speed interface	USBH (USB Host)	USBH (USB Host)				

4 References

- Universal Serial Bus Revision 2.0 Specification
- ECN USB 2.0 Link Power Management Addendum
- Enhanced Host Controller Interface Specification for Universal Serial Bus
- Enhanced Host Controller Interface Specification: Addendum
- Open Host Controller Interface Specification for USB
- USB 2.0 Transceiver Macrocell Interface (UTMI) Specification
- UTMI+ Specification

USB Host (STM32 specific)

Microprocessor Unit

Enhanced Host Controller Interface

Open Host Controller Interface

USB 2.0 Transceiver Macrocell Interface

Second Stage Boot Loader

Open Portable Trusted Execution Environment

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Stable: 19.06.2020 - 12:38 / Revision: 12.06.2020 - 12:17

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Stable: 25.09.2020 - 09:43 / Revision: 25.09.2020 - 09:39

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer