STM32MP1 Platform trace and debug environment overview

Contents

Stable: 30.03.2021 - 09:05 / Revision: 29.03.2021 - 20:37

Stable: 30.03.2021 - 09:39 | Revision: 29.03.2021 - 20:37

A quality version of this page, approved on *12 December 2019*, was based off this revision.

The block diagram below shows the **STM32MP1 Platform trace and debug environment** components and their possible interfaces:

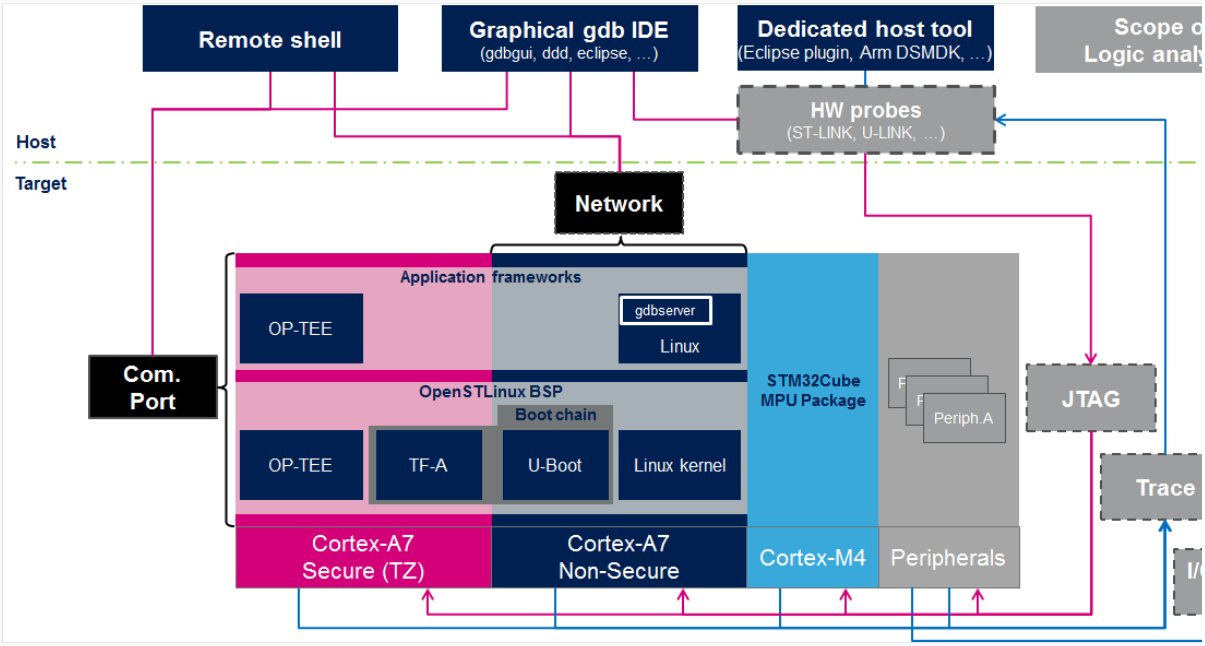
- The **STM32MPU Embedded Software** package (see STM32MPU Embedded Software architecture overview) that includes:
  - The **OpenSTLinux BSP** and **application frameworks** components, running on the Arm®Cortex®-A core
  - The **STM32Cube MPU Package** running on the Arm®Cortex®-M core
- The **STM32MPU peripherals** shared between Cortex®-A and Cortex®-M cores (such as GPIO, I2C and SPI)
- The **user interfaces or tools**, which allow to interact with different trace and debug Tools, such as:
  - The **remote shell** using terminal console
  - The **debugger tools** (such as GDB)
  - The **graphical IDE** (such as GDBGUI or SystemWorkbench)
- The **trace and debug interfaces or hardware paths** that provide access to trace and debug components through:
  - The **network** interface (e.g. Ethernet)
  - The **communication port** (e.g UART)
  - The hardware connector interfaces:
    - **JTag** port
    - **Trace** port to access ETM, STM, ITM and SWD
    - **I/O probes** to access HDP
- The **hardware probes** such as ST-Link.

This block diagram also illustrates the Arm® debugging modes:

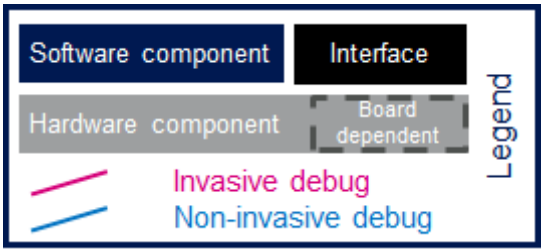
- **Invasive debug**: debug process that allows the control and monitoring of the processor. Most debug features are considered invasive because they enable you to halt the processor and modify its state.
- **Non-invasive debug**: debug process that allows the monitoring of the processor but not the control. The embedded trace macrocell (ETM) interface and the performance monitor registers are non-invasive debug features.

Click the figure below to directly jump to the component you want to trace, monitor or debug:

- By selecting a **hardware component**, you will be redirected to the corresponding hardware board article in order to check if the hardware connector is supported on your board.
- By selecting a **target software component**, you will be be redirected to an article that explains in details how to trace, monitor or debug this component.
- By selecting a **host software component**, you will be redirected to an article that explains how to use this remote tool.



STM32MP1 Platform trace and debug environment overview.



Board support package

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Cortex®

Microprocessor Unit

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Serial Peripheral Interface

GNU dedugger, a portable debugger that runs on many Unix-like systems

(Software)Integrated development/design/debugging environment

Universal Asynchronous Receiver/Transmitter

Embedded Trace Macrocell

System Trace Module

Instruction Trace Macrocell

Serial Wire Debug

Hardware Debug Port

spelling for older versions of STLink, ST in-circuit debugger and programmer for the STM8 and STM32 microcontroller families

Stable: 26.03.2021 - 11:32 / Revision: 12.03.2021 - 11:07

The block diagram below shows the **STM32MP1 Platform trace and debug environment** components and their possible interfaces:

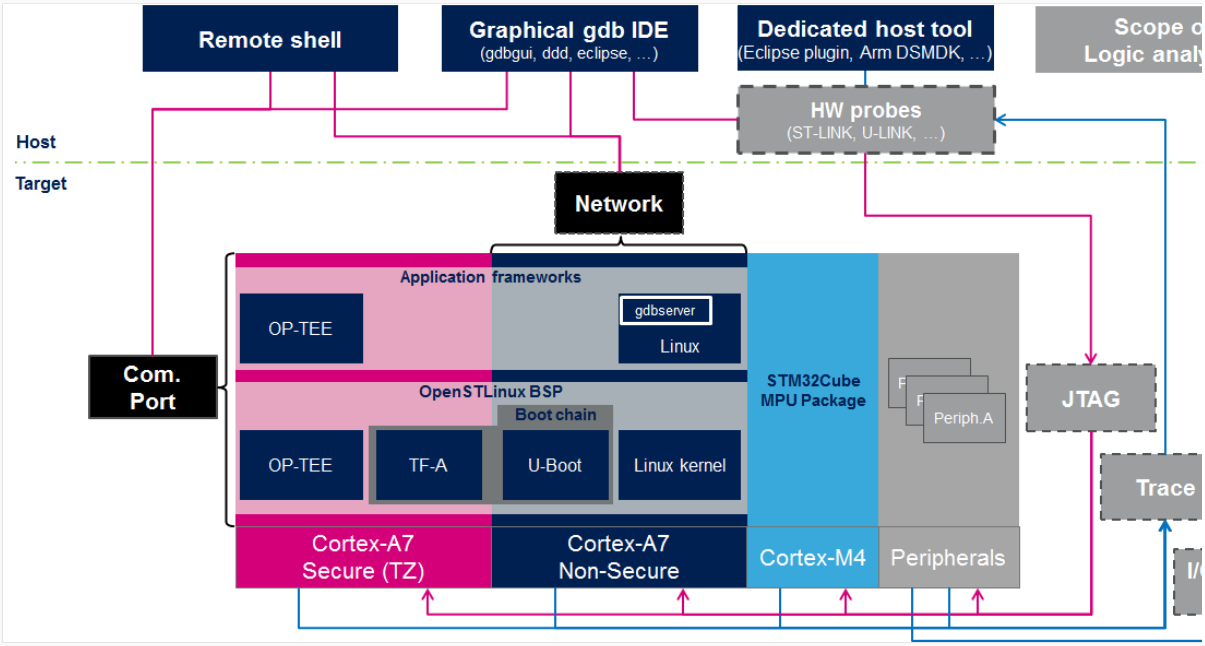
- The **STM32MPU Embedded Software** package (see STM32MPU Embedded Software architecture overview) that includes:
  - The **OpenSTLinux BSP** and **application frameworks** components, running on the Arm®Cortex®-A core
  - The **STM32Cube MPU Package** running on the Arm®Cortex®-M core
- The **STM32MPU peripherals** shared between Cortex®-A and Cortex®-M cores (such as GPIO, I2C and SPI)
- The **user interfaces or tools**, which allow to interact with different trace and debug Tools, such as:
  - The **remote shell** using terminal console
  - The **debugger tools** (such as GDB)
  - The **graphical IDE** (such as GDBGUI or SystemWorkbench)
- The **trace and debug interfaces or hardware paths** that provide access to trace and debug components through:
  - The **network** interface (e.g. Ethernet)
  - The **communication port** (e.g UART)
  - The hardware connector interfaces:
    - **JTag** port
    - **Trace** port to access ETM, STM, ITM and SWD
    - **I/O probes** to access HDP
- The **hardware probes** such as ST-Link.

This block diagram also illustrates the Arm® debugging modes:

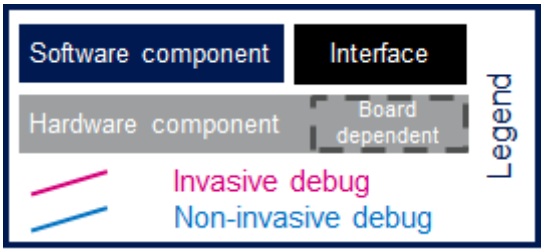
- **Invasive debug**: debug process that allows the control and monitoring of the processor. Most debug features are considered invasive because they enable you to halt the processor and modify its state.
- **Non-invasive debug**: debug process that allows the monitoring of the processor but not the control. The embedded trace macrocell (ETM) interface and the performance monitor registers are non-invasive debug features.

Click the figure below to directly jump to the component you want to trace, monitor or debug:

- By selecting a **hardware component**, you will be redirected to the corresponding hardware board article in order to check if the hardware connector is supported on your board.
- By selecting a **target software component**, you will be be redirected to an article that explains in details how to trace, monitor or debug this component.
- By selecting a **host software component**, you will be redirected to an article that explains how to use this remote tool.



STM32MP1 Platform trace and debug environment overview.



Board support package

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Cortex®

Microprocessor Unit

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Serial Peripheral Interface

GNU dedugger, a portable debugger that runs on many Unix-like systems

(Software)Integrated development/design/debugging environment

Universal Asynchronous Receiver/Transmitter

Embedded Trace Macrocell

System Trace Module

Instruction Trace Macrocell

Serial Wire Debug

Hardware Debug Port

spelling for older versions of STLink, ST in-circuit debugger and programmer for the STM8 and STM32 microcontroller families