ST Restriced – Subject to Non-Disclosure Agreement – Do not copy

STM32MP1 Platform trace and debug environment overview

ST Restriced – Subject to Non-Disclosure Agreement – Do not copy

Contents

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

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

A quality version of this page, approved on *30 March 2021*, was based off this revision.

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

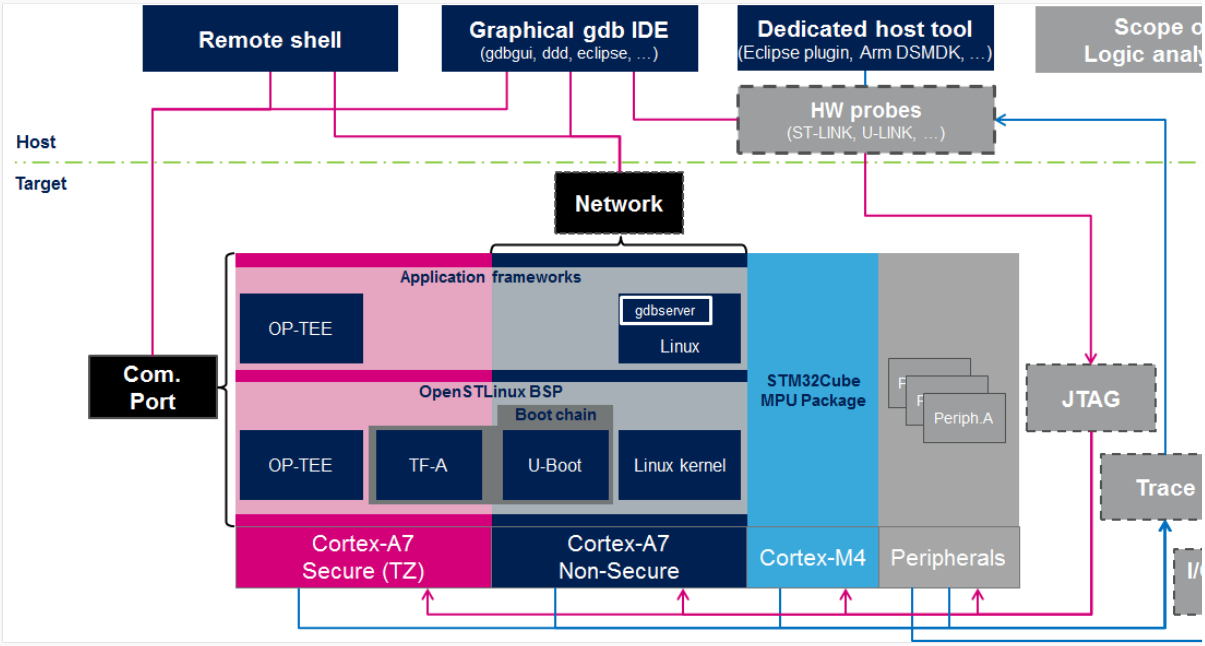
- The **STM32MPU Embedded Software** package (see STM32MPU Embedded Software architecture overview) that includes:
  - The **OpenSTLinux BSP** and **application frameworks** components, running on the Arm® Cortex®-A core
  - The **STM32Cube MPU Package** running on the Arm® Cortex®-M core
- The **STM32MPU 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:

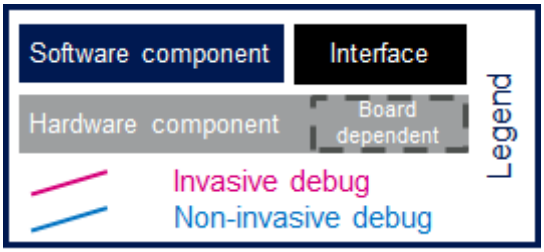
- **Invasive debug**: debug process that allows the control and monitoring of the processor. Most debug features are considered invasive because they enable you to halt the processor and modify its state.
- **Non-invasive debug**: debug process that allows the monitoring of the processor but not the 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.



Universal Asynchronous Receiver/Transmitter

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

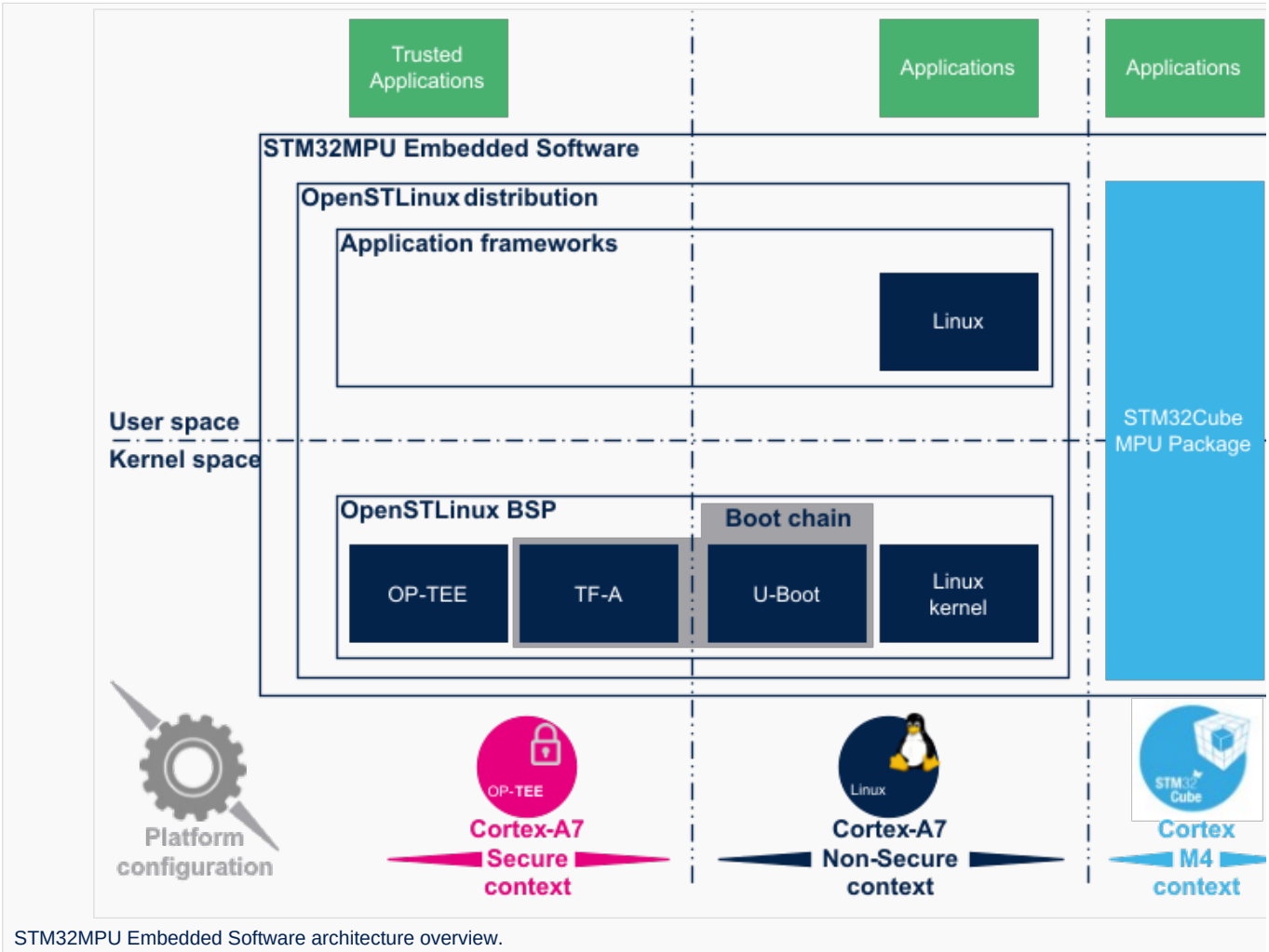
A quality version of this page, approved on *26 March 2021*, was based off this revision.

1 STM32MPU Embedded Software overview

The diagram below shows STM32MPU Embedded Software distribution main components:

- The **OpenSTLinux distribution**, running on the Arm® Cortex®-A, including:
  - The **OpenSTLinux BSP** with:
    - The boot chain based on TF-A and U-Boot.
    - The OP-TEE secure OS running on the Arm® Cortex®-A in secure mode.
    - The **Linux® kernel** running on the Arm® Cortex®-A in non-secure mode.
  - The **application frameworks** are composed of middlewares relying on the BSP and providing API, on **Linux** side, to run **Applications** that typically interact with the user via the display, the touchscreen, etc.
  - On **OP-TEE** side, the **Trusted Applications (TA)** relies on the OP-TEE core for secrets operations (not visible from the Linux and STM32Cube MPU Package)
- The **STM32Cube MPU Package** is running on the Arm® Cortex®-M: it is based on HAL drivers and middlewares, like other STM32 microcontrollers, completed with coprocessor management.

The figure below is clickable so that the user can directly jump to one of the sub-levels listed above.



STM32MPU Embedded Software architecture overview.


3rd Party
ST
Community
Legend

2 Open Source Software (OSS) philosophy

The **Open source software** source code is released under a license in which the copyright holder grants users the rights to study, change and distribute the software to anyone and for any purpose[1].
STMicroelectronics maximizes the using of open source software and contributes to those communities. Notice that, due to the software review life cycle, it can take some time before getting all developments accepted in the communities, so STMicroelectronics can also temporarily provide some source code on github[2], until it is merged in the targeted repository.

3 References

- https://en.wikipedia.org/wiki/Open-source_software
- STM32MP1 Distribution Package