



STM32MP1 Platform trace and debug environment overview for Android

STM32MP1 Platform trace and debug environment overview for Android



Contents

1. STM32MP1 Platform trace and debug environment overview for Android	3
2. Main Page	3



Stable: 17.11.2021 - 10:46 / Revision: 17.11.2021 - 15:58

The content format pdf is not supported by the content model wikitext.

[Return to Main Page](#)

Stable: 17.11.2021 - 10:46 / Revision: 17.11.2021 - 15:58

You do not have permission to edit this page, for the following reasons:

- The action you have requested is limited to users in one of the groups: **Administrators**, **Editors**, **Reviewers**, **Selected_editors**, **ST_editors**.
- 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**

You can view and copy the source of this page.

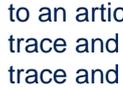


The block diagram below shows the **STM32MP1 Platform trace and debug environment for Android** components and their possible interfaces. Click the block diagram to directly jump to one of the sub-levels listed below:

- * The "STM32MPU Embedded Software" package (see [STM32MPU Embedded Software for Android architecture overview](#)) that comprises:
 - ** the "STM32MPU distribution for Android" running on the Arm Cortex-A and 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 core in Secure mode.
 - **** the "Linux kernel" running on the Arm Cortex-A core in Non-secure mode.
 - *** the "application frameworks" composed of middleware components relying on the BSP and providing:
 - **** "OP-TEE" APIs to run "Trusted Applications (TA)" that allow manipulating secrets (not visible from the Linux and STM32Cube MPU Package).
 - **** "Android" APIs to run "applications" that typically interact with the user via a display or a touchscreen.
 - ** the "STM32Cube MPU Package" runs on the Arm Cortex-M core: like other STM32 microcontrollers, it is based on HAL drivers and middleware components. It is completed with the [coprocessor management](#).
- * The "STM32MPU peripherals" shared between Cortex-A and Cortex-M cores (such as GPIO, I2C and SPI).
- * The "user interfaces or tools", which allow interacting with different trace and debug Tools, such as:
 - ** The "remote shell" using terminal console
 - ** The "Android host tools" (such as Android Studio)
 - ** 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 controlling and monitoring 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 monitoring the processor but not controlling it. 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:

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

[STM32MP1 Platform trace and debug environment overview for Android](#)  800px center STM32MP1 Platform trace and debug environment overview for Android...
[How to get Terminal/Remote shell](#) rect 220 4 461 72 [Android tracing, monitoring and debugging | Android host tools](#) rect 473 4 746 72 [GDB](#) rect 789 90 1064 160 [Category:HW probes](#) rect 594 181 731 248 [Getting started with ST boards | Network link](#) rect 256 312 386 385 [OP-TEE - How to debug | OP-TEE](#) rect 661 312 793 385 [Android tracing, monitoring and debugging | Linux applications](#) rect 256 459 386 531 [OP-TEE - How to debug | OP-TEE](#) rect 401 459 516 531 [TF-A - How to debug | TF-A](#) rect 537 459 650 531 [U-Boot - How to debug | U-Boot](#) rect 661 459 793 531 [Linux tracing, monitoring and debugging | Linux kernel](#) rect 94 380 229 445

[Getting started with ST boards | ComPort connector](#) rect 1101 389 1237 459 [Getting started with ST boards | JTag connector](#) rect 1179 485 1316 554 [Getting started with ST boards | Trace connector](#) rect 1258 586 1407 656 [Getting started with ST boards | IOProbe connector](#) rect 800 270 939 559 [STM32CubeMP1 Package | STM32CubeMP1 Package](#) } [File:STM32MP1 Platform trace and debug environment overview legend.png](#) center link= <noinclude> [Category:Trace and debug tools](#) 1 [Category:Android](#) { [PublicationRequestId](#) 13245 | 2019-09-12 } </noinclude>

Templates used on this page:

- [Template:Highlight](#) (view source)
- [Template:Info](#) (view source)
- [Template:STDarkBlue](#) (view source)

[Return to Main Page.](#)