



# STM32MP1 Distribution Package for Android



# STM32MP1 Distribution Package for Android

Stable: 16.04.2020 - 07:58 / Revision: 16.04.2020 - 07:48

This article describes how to obtain and use the **Distribution Package for Android** of the **STM32MPU Embedded Software** for any **STM32MP1 family** (**STM32MP15 boards**) development platform, in order to modify or add pieces of software, and to create the right Android distribution for the targeted product.

It lists some **prerequisites** in terms of knowledge and development environment, and gives **step-by-step** instructions to download and install the STM32MPU Embedded Software packages for this Package.

## Contents

1 Distribution Package content .....	2
2 Prerequisites .....	2
<b>2.1 Knowledge .....</b>	<b>3</b>
<b>2.2 Development setup .....</b>	<b>3</b>
3 Installing the STM32MPU distribution for Android .....	5
<b>3.1 Use manifest .....</b>	<b>5</b>
<b>3.2 Use packages .....</b>	<b>6</b>
<b>3.3 Use local mirror .....</b>	<b>6</b>
4 Building the STM32MPU distribution for Android .....	7
5 Flashing the built image .....	7
6 References .....	7

## 1 Distribution Package content

If you are not familiar with the **STM32MPU Embedded Software for Android** distribution and its **Packages**, please read the following articles:

- Which STM32MPU Embedded Software Package for Android better suits your needs (and especially the Distribution Package chapter)
- STM32MPU Embedded Software distribution for Android

In summary, this **Distribution Package** provides:

- An **Android build framework** (aka distribution builder)
- For the **STM32MPU distribution for Android** (development on Arm Cortex-A processor):
  - the BSP (Linux kernel, U-Boot, TF-A, optionally OP-TEE) pieces in **binary** (prebuilt). Several scripts are provided to load source code and re-build them
  - the application frameworks (including hardware abstraction code) pieces in **source code**
- For the **STM32Cube MPU Package** (development on Arm Cortex-M processor), all pieces of software in **source code**: BSP, HAL, middlewares and applications
- A toolset to tune the system for your needs, and to handle the built image (for example STM32CubeProgrammer to install the built image on the board).

## 2 Prerequisites



## 2.1 Knowledge

The STM32MP1 Distribution Package aims to create an Android distribution for the targeted product: solid knowledge of Linux and Android are recommended to make the most of this Package.

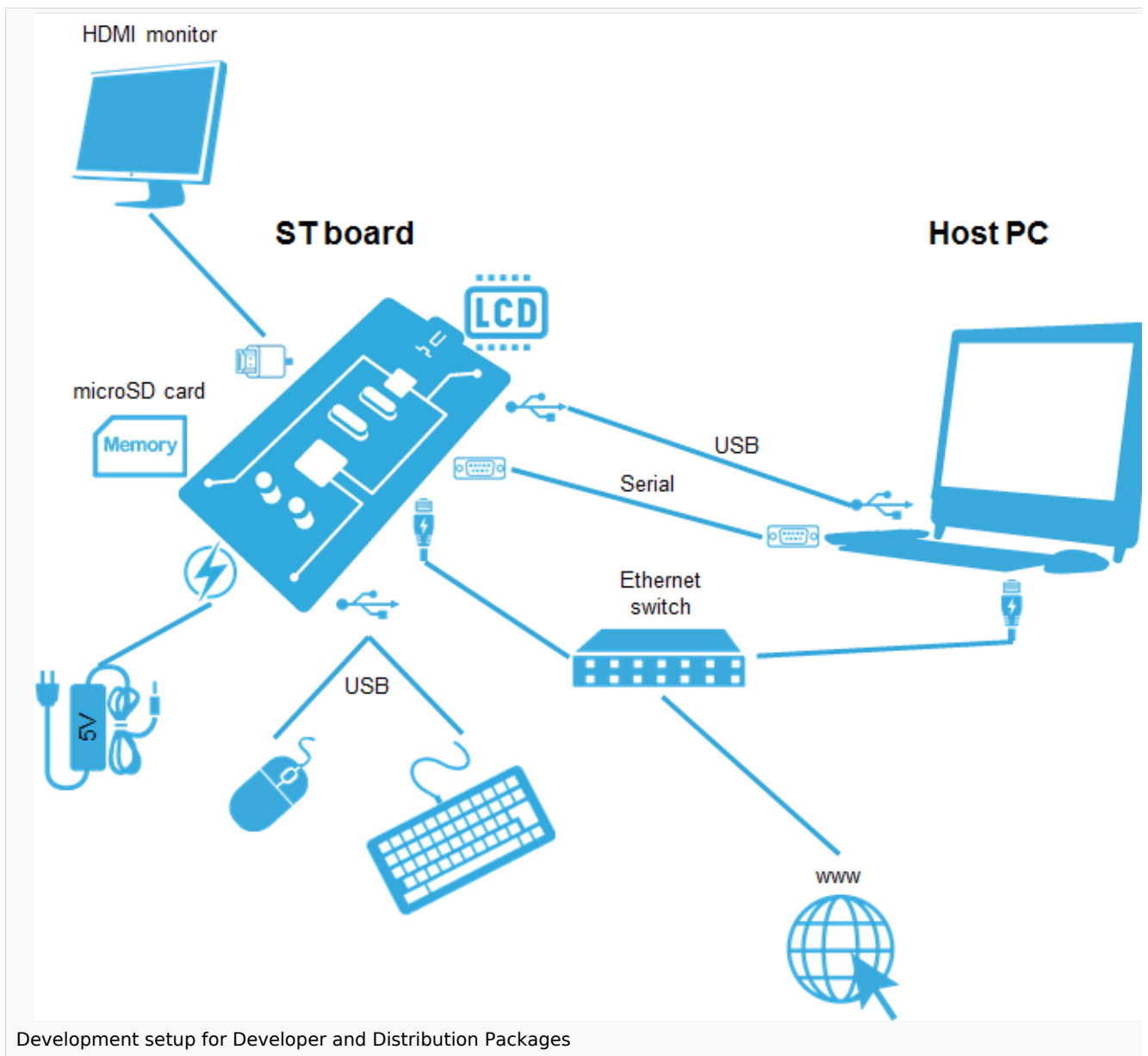
The STM32MPU distribution for Android is an Android distribution based on the Android build framework: a short introduction about Android is available in AOSP<sup>[1]</sup>.

Reading the [STM32MPU Embedded Software for Android architecture overview](#) is also highly recommended.

## 2.2 Development setup

The recommended setup for the development PC (host) is specified in the following article: [PC prerequisites](#).

Whatever the development platform (board) and development PC (host) used, the range of possible development setups is illustrated by the picture below.



Development setup for Developer and Distribution Packages

The following components are **mandatory**:

- Host PC for cross-compilation and cross-debugging, installed as specified above
- Board assembled and configured as specified in the associated Starter Package article
- Mass storage device (for example, microSD card) to load and update the software images (binaries)

The following components are **optional**, but **recommended**:

- A serial link between the host PC (through [Terminal program](#)) and the board for traces (even early boot traces), and access to the board from the remote PC (command lines)
- An Ethernet link between the host PC and the board for cross-development and cross-debugging through a local network. This is an alternative or a complement to the serial (or USB) link
- A display connected to the board, depending on the technologies available on the board: DSI LCD display, HDMI monitor (or TV) and so on
- A mouse and a keyboard connected through USB ports



**Additional optional** components can be added by means of the connectivity capabilities of the board: cameras, displays, JTAG, sensors, actuators, and much more.

## 3 Installing the STM32MPU distribution for Android

The software package is provided AS IS, and by downloading it, you agree to be bound to the terms of the [software license agreement \(SLA\)](#). The detailed content licenses can be found [here](#).

The installation relies on the `repo` command. If the Repo tool (a Google-built repository management tool that runs on top of Git) is not already installed and configured on the host PC, refer to the [PC prerequisites](#) article.

Create a working directory for the distribution (we recommended using the distribution name):

```
PC $> mkdir <working directory>
PC $> cd <working directory>
```

### 3.1 Use manifest

Initialize repo in the working directory:

```
PC $> repo init -u https://github.com/STMicroelectronics/android-manifest -b refs/tags
/<Tag> -m stm32mp1droid.xml
```

With

- `<Tag>` = **tag** selecting the required version of the STM32MPU distribution for Android™ (see following table)

STM32MPU distribution for Android™		
Tag	android-<version>	Release note
st-android-10.0.0-2020-02-21	android-10.0.0_r22	<a href="#">st-android-10.0.0-2020-02-21 release note</a>
st-android-9.0.0-2019-09-27	android-9.0.0_r46	<a href="#">st-android-9.0.0-2019-09-27 release note</a>

Synchronize the local project directories with the remote repositories specified in the manifest:

```
PC $> repo sync
```

This may take several hours.



## 3.2 Use packages

If you have been provided with an archive `<STM32MPU_delivery_version>.tar.gz`, you need to follow those steps:

- Download the official Android version compatible with the provided package (with or without mirror):

```
PC $> repo init -u https://android.googlesource.com/platform/manifest -b android-  
<version>  
PC $> repo sync
```

For more info on available version please refer to the official page<sup>[2]</sup>.

This may take several hours.

- Download the archive depending on the required version (tag):

STM32MPU distribution for Android™			
Tag	Link	android- <version>	Release note
st-android-10.0.0- 2020-02-21	<a href="https://en.st-android-10.0.0-2020-02-21-source.tar.gz">en.st-android-10.0.0-2020-02-21-source.tar.gz</a>	android-10.0.0 _r22	<a href="#">st-android-10.0.0-2020-02-21 release note</a>
st-android-9.0.0- 2019-09-27	<a href="https://en.st-android-9.0.0-2019-09-27-source.tar.gz">en.st-android-9.0.0-2019-09-27-source.tar.gz</a>	android-9.0.0 _r46	<a href="#">st-android-9.0.0-2019-09-27 release note</a>

- Extract the archive inside the `<working directory>`:

```
PC $> tar xzf en.<Tag>-source.tar.gz
```

## 3.3 Use local mirror

It can be of interest to create a local mirror of the different repositories used.

A helper script `cachesetup` is available for this purpose, which creates / updates the mirrors based on the configuration file `device/stm/<STM32Series>/scripts/cache/android_cache.config`.

Update the cache configuration file (`android_cache.config`) depending on your needs. There are two possible kinds of cache:

- REPO mirror: mirror of list of modules in repositories based on provided manifest (for example Android AOSP)
- GIT mirror: mirror of one module (for example Linux kernel)

Take care that if you change the environment variable name, you have to adapt the associated scripts `load_xxxx` available in `device/stm/<STM32Series>-xxxx/source` directory.

Create the mirrors by executing the following commands:

```
PC $> source build/envsetup.sh  
PC $> cachesetup --new
```

Regular updating of your mirror is recommended by executing:



```
PC $> cachesetup
```

For more information on how to use this command, look at the cachesetup usage (execute `cachesetup -h`).

From this stage, you can use the local mirror instead of the remote repository by adding `--reference=$CACHE_DIR` to the repo init commands (the `CACHE_DIR` environment variable name is set in `android_cache.config` file).

```
PC $> repo init --reference=$CACHE_DIR -u https://github.com/STMicroelectronics/
/android-manifest -b refs/tags/<Tag> -m stm32mp1droid.xml
```

or

```
PC $> repo init --reference=$CACHE_DIR -u https://android.googlesource.com/platform
/manifest -b android-<version>
```

This copies all necessary files defined by the manifest from the mirror, instead of downloading it from the remote repository.

## 4 Building the STM32MPU distribution for Android

When the installation is complete, please refer to the [How to build STM32MPU distribution for Android](#) page to learn how to build it.

## 5 Flashing the built image

When the distribution has been built, please refer to the [How to populate boards for Android](#) page to learn how to flash your device.

## 6 References

- <https://source.android.com/devices/architecture>
- <https://source.android.com/setup/start/build-numbers#source-code-tags-and-builds>

also known as

Board support package

Das U-Boot -- the Universal Boot Loader (see [U-Boot\\_overview](#))

Trusted Firmware for Arm Cortex-A

Open Portable Trusted Execution Environment



## STM32MP1 Distribution Package for Android

---

Microprocessor Unit

Hardware Abstraction Layer

Android Open Source Project

Display Serial Interface (MIPI<sup>®</sup> Alliance standard)

High-Definition Multimedia Interface (HDMI standard)

debug and test protocol, named from the Joint Test Action Group that developed it

stm32mp1