How to create your own STM32MPU distribution for Android

Contents

Stable: 16.03.2021 - 10:00 / Revision: 04.11.2020 - 09:00

A quality version of this page, approved on *16 March 2021*, was based off this revision.

This article explains how you can create your own STM32MPU distribution for Android (adapted to your device) starting from an existing one based on STMicroelectronics reference boards.

It is performed in several steps:
1- Create your own device directory, and configure it depending on your device capability
2- Customize the <STM32Series> setup
3- Customize the <STM32Series> associated BSP (TF-A, U-Boot, OP-TEE, Linux)
4- Build your distribution
5- Flash your distribution
6- Debug your distribution

It is also possible to create your own device structure, getting back only the required <STM32Series> associated BSP (TF-A, U-Boot, OP-TEE or/and Linux kernel). For that purpose refer directly to the chapter Customize the BSP.

It is intended for Distribution Package users.

Contents

1 Prerequisites

The environment used must be installed using the right Distribution Package for your selected microprocessor device. See Distribution_Package to allow understanding how to get back the referenced distribution.

The STM32CubeProgrammer must be installed.

2 Create your device

You need to create your device folder in the Android distribution tree with the name of your device in `device/stm/<STM32Series>/<your device name>`.

Once created, you must put all necessary configuration files dedicated to your device, like it is done for STMicroelectronics reference <BoardId>.

- MakeFiles[1]
  - `device.mk`: declares the files and modules needed for the device
  - `aosp_<your device name>.mk`: creates a specific product definition makefile
  - `AndroidProducts.mk`: points to the products makefiles `aosp_<your device name>.mk`
  - `BoardConfig.mk`: contains board-specific configurations

- Overlays: some overlays need to be adapt to match your device configuration especially `frameworks/base/core/res/res/values/config.xml.`. They must be added in `device/stm/<STM32Series>/<your device name>/overlay`
- Configurations: depending on your device some configurations must be changed like for Bluetooth, WiFi, audio etc... Please have a look at How to customize the STM32MPU distribution for Android.

3 Customize the distribution setup

3.1 Adapt the Android distribution

A mechanism is in place to select required modules version and potentially apply some patches directly to the Android distribution source files.

It is recommended to take a look at the list of modifications performed. All explanations are available in Modify Android distribution chapter.

3.2 Adapt the flash layout

A configuration file is used to configure the layout. It is recommended to take a look at the current layout configuration and adapt it if needed.

All explanations are available in Layout configuration chapter.

3.3 Execute the setup

```
PC $> source ./build/envsetup.sh
PC $> <STM32Series>setup
```

The `<STM32Series>setup` (ex: stm32mp1setup for STM32MP1 series) script must be executed only once.

During the distribution setup, the required graphics libraries are loaded.

4 Customize the BSP

You can change various elements to customize the BSP to match your device.

Information

Every scripts used below need to be configured for your device.

You need to change on every `build_xxxx` (available in `device/stm/<STM32Series>-xxxx/source` directory) the values of the board name list `BOARD_NAME_LIST` with your device name and the associated board flavour list `BOARD_FLAVOUR_LIST` which are used to select the device tree (naming convention: *<STM32Series>-<board flavour>*)

4.1 Bootloaders

If you want to use the same bootloaders as on STMicroelectronics reference board (TF-A and U-Boot), you have first to load the source (please refer to Load the bootloader sources chapter).

You can update the TF-A and U-boot with your own device tree (it is possible to use the STM32CubeMX tool to help you creating it):

- TF-A device trees are available in `device/stm/<STM32Series>-bootloader/tf-a-<STM32Series>/fdts/` directory
- U-Boot device trees are available in `device/stm/<STM32Series>-bootloader/u-boot-<STM32Series>/arch/arm/dts/` directory

You can adapt the used U-boot defconfig available in `device/stm/<STM32Series>-bootloader/u-boot-<STM32Series>/configs/`:

- <STM32Series>_trusted_defconfig if you do not need any trusted execution environment

You have then to rebuild the bootloaders (please refer to Build the bootloaders chapter).

4.2 TEE

If you want to use the same trusted execution environment (OP-TEE), you have first to load the source (please refer to Load OP-TEE sources chapter).

You can update OP-TEE OS with your own device tree (it is possible to use the STM32CubeMX tool to help you creating it).

OP-TEE OS device trees are available in `device/stm/<STM32Series>-tee/optee_os-<STM32Series>/core/arch/arm/fdts/` directory.

You have then to rebuild the trusted execution environment (please refer to Build OP-TEE OS chapter).

You can also build the list of trusted applications required (please refer to Build Trusted Applications chapter).

4.3 Kernel

You can update the Linux kernel configuration, please refer to How to customize kernel for Android.

5 Build your distribution

To be able to build your customized distribution you need to

```
PC $> lunch aosp_<your device>-<build_type>
```

Available `build_type` values:

- `user`: to generate an end-user production image;
- `userdebug`: similar to an `user` build but with root access and debug capabilities;
- `eng`: development configuration with additional debugging tools.

Then build it

```
PC $> make -j
```

For more details please have a look at How to build STM32MPU distribution for Android.

6 Flash your distribution

From this stage you can flash the generated images on your device (please refer to Flashing the built image).

7 Debug your distribution

From this stage, there are various ways to debug your device (please refer to STM32MP1 Platform trace and debug environment overview for Android).

8 References

- https://source.android.com/setup/develop/new-device

Das U-Boot -- the Universal Boot Loader (see U-Boot_overview)