Example of directory structure for Packages

Stable: 31.03.2021 - 10:27 / Revision: 29.03.2021 - 13:41

A quality version of this page, accepted on *31 March 2021*, was based off this revision.

Contents

1 Article purpose

This article aims at proposing a way to organize, on the host PC, the software packages of the different Packages (Starter, Developer and Distribution) for a given release of the STM32MPU Embedded Software distribution.

The main objective of the proposed organization is to keep together the software packages corresponding to a given release because there are links between them. For example:

- Flashing the image from the Starter Package on the board is mandatory before modifying the source code from the Developer Package. Both the image and the source code must belong to the same software release.
- The SDK (Developer Package) and the image (Starter Package) have both been generated from the Distribution Package. A software release thus guarantees that there is no misalignment between the different software packages.

> The objective of this article is to describe one organization among all the possible organizations. Feel free to organize the delivered Packages in any other way that would better match your way of working.

> In practice, this article uses the release **STM32MP15-Ecosystem-v3.0.0** for the STM32MPU Embedded Software distribution as an example to illustrate the proposed organization. If you are using a different release, the names of the directories and files might differ.

The directories are shown in green, while the files are in black.

2 Creating the structure

- Create your <working directory> and assign a unique name to it (for example by including the release name):

```
PC $> mkdir STM32MP15-Ecosystem-v3.0.0
PC $> cd STM32MP15-Ecosystem-v3.0.0
```

- Create the first-level directories that will host the software packages delivered through the STM32MPU Embedded Software distribution release note:

```
PC $> mkdir Starter-Package
PC $> mkdir Developer-Package
PC $> mkdir Distribution-Package
```

- The resulting directory structure looks as follows:

```
STM32MP15-Ecosystem-v3.0.0  STM32MPU Embedded Software release
├── Developer-Package       Developer Package installation directory
├── Distribution-Package    Distribution Package installation directory
└── Starter-Package         Starter Package installation directory
```

Once all software packages have been installed according to the instructions given in the STM32MPU Embedded Software distribution release note, the resulting directory structure looks as follows:

```
STM32MP15-Ecosystem-v3.0.0                                  STM32MPU Embedded Software release
├── Developer-Package                                       Developer Package installation
directory
│   ├── SDK                                                 SDK for OpenSTLinux distribution
│   ├── STM32Cube_FW_MP1_V1.4.0                             STM32CubeMP1 Package
│   └── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31  Linux kernel, U-Boot, TF-A and OP-
TEE OS source code (OpenSTLinux distribution)
├── Distribution-Package                                    Distribution Package installation
directory
│   └── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31  OpenSTLinux distribution (full
source code and OpenEmbedded-based build framework)
└── Starter-Package                                         Starter Package installation
directory
    └── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31  Software image (binaries)
```

3 Focus on the Starter Package directory

The *Starter-Package* directory contains the software image for the STM32MPU Embedded Software distribution.

The trusted boot chain is the default solution delivered by STMicroelectronics. It includes the superset of features (for example, all Flash memory devices are supported). The basic boot chain is also upstreamed by STMicroelectronics, with a limited number of features (for example microSD card memory boot only). Refer to the Boot chain overview article for details.

Flash memory partitions (e.g. rootfs, bootfs...) are explained in the STM32MP15 Flash mapping article.

```
Starter-Package
└── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31
    └── images
        └── stm32mp1
            ├── flashlayout_st-image-weston                                  Flash layout
files (description of the partitions) for the supported Flash devices and boards
            │   ├── FlashLayout_emmc_stm32mp157c-ev1-optee.tsv               Flash layout
file for eMMC and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
            │   ├── FlashLayout_emmc_stm32mp157c-ev1-trusted.tsv             Flash layout
file for eMMC and trusted boot chain → STM32MP157C-EV1
            │   ├── FlashLayout_sdcard_stm32mp157c-dk2-basic.tsv             Flash layout
file for microSD card and basic boot chain → STM32MP157C-DK2
            │   ├── FlashLayout_sdcard_stm32mp157c-dk2-optee.tsv             Flash layout
file for microSD card and trusted boot chain, with OP-TEE OS → STM32MP157C-DK2
            │   ├── FlashLayout_sdcard_stm32mp157c-dk2-trusted.tsv           Flash layout
file for microSD card and trusted boot chain (recommended setup) → STM32MP157C-DK2
            │   ├── FlashLayout_sdcard_stm32mp157c-dk2-extensible.tsv        Flash layout
file for microSD card with no userfs partition but a rootfs partition extended to sdcard
size (recommended setup for package repository service) → STM32MP157C-DK2
            │   └── [...]
            ├── fip
            │   ├── fip-<board>-<boot-type>.bin                Image to flash
            ├── arm-trusted-firmware
            │   ├── tf-a-<board>-<boot-type>.stm32             Intermediate tf-a binary used
to build fip image
            ├── u-boot
            │   ├── u-boot-nodtb-soc.bin                       Intermediate tf-a binary used
to build fip image
            │   ├── u-boot-<board>-<boot-type>.dtb             Intermediate tf-a dtb used to
build fip image
            ├── optee
            │   ├── tee-header_v2-<board>.bin
            │   ├── tee-pageable_v2--<board>.bin
            │   ├── tee-pager_v2--<board>.bin
            ├── kernel
            │   ├── vmlinux                                    vmlinux copied in bootfs
partition
            ├── scripts
            │   └── create_sdcard_from_flashlayout.sh
            ├── st-image-bootfs-openstlinux-weston-stm32mp1.ext4             Binary for boo
tfs partition
            ├── st-image-bootfs-openstlinux-weston-stm32mp1.manifest
            ├── st-image-userfs-openstlinux-weston-stm32mp1.ext4             Binary for use
rfs partition
            ├── st-image-userfs-openstlinux-weston-stm32mp1.manifest
            ├── st-image-vendorfs-openstlinux-weston-stm32mp1.ext4           Binary for ven
dorfs partition
            ├── st-image-weston-openstlinux-weston-stm32mp1.ext4             Binary for roo
```

Example of directory structure for Packages

```
tfs partition
             ├── st-image-weston-openstlinux-weston-stm32mp1.license
             ├── st-image-weston-openstlinux-weston-stm32mp1-license_content.html
             ├── st-image-weston-openstlinux-weston-stm32mp1.manifest
             └── [...]
```

4 Focus on the Developer Package directory

The *Developer-Package* directory contains:

- The source code for the following OpenSTLinux software packages (development for Arm®Cortex®-A processor):
  - Linux® kernel
  - U-Boot
  - TF-A
  - OP-TEE OS
- The debug symbol files for Linux kernel, U-Boot, TF-A and OP-TEE OS
- The SDK (for cross-development on an host PC)
- The STM32Cube MPU Package (developed for Arm®Cortex®-M processor)

```
Developer-Package
├── SDK                                                                    SDK
for OpenSTLinux distribution: details in Standard SDK directory structure article
│   ├── environment-setup-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi  Environment setup
script for Developer Package
│   ├── site-config-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
│   ├── sysroots
│   │   ├── cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi                Target sysroot
(libraries, headers, and symbols)
│   │   │   └── [...]
│   │   └── x86_64-ostl_sdk-linux                                     Native sysroot
(libraries, headers, and symbols)
│   │       └── [...]
│   └── version-cortexa7t2hf-neon-vfpv4-ostl-linux-gnueabi
```

```
├── STM32Cube_FW_MP1_V1.4.0              STM32CubeMP1 Package: details in STM32CubeMP1
Package content article
│   ├── Drivers
│   │   ├── BSP                          BSP drivers for the supported STM32MPU boards
│   │   │   └── [...]
│   │   ├── CMSIS
│   │   │   └── [...]
│   │   └── STM32MP1xx_HAL_Driver        HAL drivers for the supported STM32MPU devices
│   │       └── [...]
│   ├── _htmresc
│   │   └── [...]
│   ├── License.md                       License types for the components
│   ├── Middlewares                      Middlewares used by the examples and applications
│   │   └── [...]
│   ├── package.xml
│   ├── Projects
│   │   ├── STM32CubeProjectsList.html   List of examples and applications for
STM32CubeMP1 Package
│   │   ├── STM32MP157C-DK2              Set of examples and applications → STM32MP157C-DK2
│   │   │   └── [...]
│   │   └── STM32MP157C-EV1              Set of examples and applications → STM32MP157C-EV1
│   │       └── [...]
│   ├── Readme.md
│   ├── Release_Notes.html               Release note for STM32CubeMP1 Package
│   └── Utilities
│       └── [...]
```

Example of directory structure for Packages

```
└── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31     Source code for OpenSTLinux
distribution
    ├── images
    │   └── stm32mp1                                   Debug symbol files installation
directory
    │       ├── tf-a-bl2-optee.elf                     Debug symbol file for TF-A, with OP-
TEE OS → trusted boot firmware stage
    │       ├── tf-a-bl2-trusted.elf                   Debug symbol file for TF-A → trusted
boot firmware stage
    │       ├── tf-a-bl32-trusted.elf                  Debug symbol file for TF-A → runtime
software stage
    │       ├── u-boot-stm32mp157a-dk1-optee.elf       Debug symbol file for U-Boot, with OP-
TEE OS → STM32MP157A-DK1
    │       ├── u-boot-stm32mp157a-dk1-trusted.elf     Debug symbol file for U-Boot →
STM32MP157A-DK1
    │       ├── u-boot-stm32mp157c-dk2-optee.elf       Debug symbol file for U-Boot, with OP-
TEE OS → STM32MP157C-DK2
    │       ├── u-boot-stm32mp157c-dk2-trusted.elf     Debug symbol file for U-Boot →
STM32MP157C-DK2
    │       ├── u-boot-stm32mp157c-ev1-optee.elf       Debug symbol file for U-Boot, with OP-
TEE OS → STM32MP157C-EV1
    │       ├── u-boot-stm32mp157c-ev1-trusted.elf     Debug symbol file for U-Boot →
STM32MP157C-EV1
    │       ├── vmlinux                                Debug symbol file for Linux kernel
    │       └── [...]
```

```
    └── sources
        └── arm-ostl-linux-gnueabi
            ├── FIP_artifacts
            ├── fip/                          fip images of Stater Package
            ├── arm-trusted-firmware/         intermediate tf-a binaries from Stater Package
(needed to build any fip image
            ├── optee/                        intermediate optee binaries from Stater Package
(needed to build any fip image
            └── u-boot/                       intermediate u-boot binaries from Stater Package
(needed to build any fip image
```

```
            ├── linux-5.10.10
            │   ├── [*].patch                 ST patches for Linux kernel
            │   ├── fragment-[*].config       ST configuration fragments for Linux kernel
            │   ├── linux-5.10.10              Linux kernel source code directory
            │   ├── linux-5.10.10.tar.xz
            │   ├── README.HOW_TO.txt         Helper file for Linux kernel management: referenc
e for Linux kernel build
            │   └── series
```

```
            ├── optee-os-stm32mp-3.12.0.r1  OP-TEE OS installation directory
            │   ├── [*].patch                ST patches for OP-TEE OS
            │   ├── optee-os-stm32mp-3.12.0.r1
            │   ├── Makefile.sdk             Makefile for the OP-TEE OS compilation
            │   ├── optee-os-stm32mp-3.12.0.r1-r0.tar.gz         OP-TEE OS source code
directory
            │   ├── README.HOW_TO.txt        Helper file for OP-TEE OS management: reference
 for OP-TEE OS build
            │   └── series
```

```
            .
```

```
├── tf-a-stm32mp-2.4.r1           TF-A installation directory
│   ├── [*].patch                 ST patches for TF-A
│   ├── tf-a-stm32mp-2.4.r1  TF-A source code directory
│   ├── Makefile.sdk              Makefile for the TF-A compilation
│   ├── README.HOW_TO.txt         Helper file for TF-A management: reference
for TF-A build
│   ├── series
│   └── tf-a-stm32mp-2.4.r1-r0.tar.gz
```

```
└── u-boot-stm32mp-2020.10.r1  U-Boot installation directory
    ├── [*].patch              ST patches for U-Boot
    ├── Makefile.sdk           Makefile for the U-Boot compilation
    ├── README.HOW_TO.txt      Helper file for U-Boot management: reference
for U-Boot build
    ├── series
    ├── u-boot-stm32mp-2020.10.r1  U-Boot installation directory        U-
Boot source code directory
    └── u-boot-stm32mp-2020.01-r0.tar.gz
```

Appendix A shows the structure of the Linux kernel, U-Boot, TF-A and OP-TEE OS installation directories after these software packages have been built.

5 Focus on the Distribution Package directory

The *Distribution-Package* directory contains all the OpenEmbedded layers required to get the source code of any STM32MPU Embedded Software component, as well as a build framework based on OpenEmbedded.

```
Distribution-Package
└── openstlinux-5.10-dunfell-mp1-21-03-31  OpenSTLinux distribution
    └── layers
        ├── meta-openembedded  Collection of layers for the OpenEmbedded-Core universe (Op
enEmbedded standard)
        │   └── [...]
        ├── meta-qt5           QT5 layer for OpenEmbedded (standard)
        │   └── [...]
```

```
        ├── meta-st
        │   ├── meta-st-openstlinux                    STMicroelectronics layer that
contains the settings of the frameworks and images for the OpenSTLinux distribution
        │   │   ├── recipes-st
        │   │   │   ├── images
        │   │   │   │   ├── st-image-core.bb          Core image for OpenSTLinux
distribution
        │   │   │   │   └── st-image-weston.bb        Weston image with basic Wayland
support for OpenSTLinux distribution: recommended setup
        │   │   │   └── packagegroups
        │   │   │       └── [...]
        │   │   └── [...]
```

```
        │   ├── meta-st-stm32mp                    STMicroelectronics layer that contains
the description of the BSP for the STM32 MPU devices
        │   │   ├── recipes-bsp
        │   │   │   ├── alsa                       Recipes for ALSA control configuration
        │   │   │   │   └── [...]
        │   │   │   ├── drivers                    Recipes for Vivante GCNANO GPU kernel
drivers
        │   │   │   │   └── [...]
        │   │   │   ├── trusted-firmware-a         Recipes for TF-A
        │   │   │   │   └── [...]
        │   │   │   └── u-boot                     Recipes for U-Boot
        │   │   │       └── [...]
        │   │   ├── recipes-extended
        │   │   │   ├── m4projects                 Recipes for STM32Cube MPU Package
within the OpenSTLinux distribution
        │   │   │   │   └── [...]
        │   │   │   └── [...]
        │   │   ├── recipes-graphics
        │   │   │   ├── gcnano-userland            Recipes for Vivante libraries OpenGL
ES, OpenVG and EGL (multi backend)
        │   │   │   │   └── [...]
        │   │   │   └── [...]
        │   │   ├── recipes-kernel
        │   │   │   └── linux                      Recipes for Linux kernel
        │   │   │       └── [...]
        │   │   └── linux-firmware                 Recipes for Linux firmwares (example,
Bluetooth firmware)
        │   │       └── [...]
```

```
        │   │   ├── recipes-st
        │   │   │   └── images
        │   │   │       ├── st-image-bootfs.bb    Recipes for the bootfs partition binary
        │   │   │       ├── st-image-userfs.bb    Recipes for the userfs partition binary
        │   │   │       └── st-image-vendorfs.bb  Recipes for the vendorfs partition
binary
        │   │   └── [...]
```

```
        │   ├── meta-st-stm32mp-addons  STMicroelectronics layer that helps managing the
STM32CubeMX integration
        │   │   └── [...]
        │   └── scripts
        │       ├── envsetup.sh         Environment setup script for Distribution Package
        │       └── [...]
```

```
        ├── meta-timesys       Timesys layer for OpenEmbedded (standard)
        │   └── [...]
        └── openembedded-core  Core metadata for current versions of OpenEmbedded
(standard)
            └── [...]
```

Appendix B shows the structure of the build directory.

6 Appendix A: directory structure after build (Developer Package)

Provided you have followed the recommendations of the *README.HOW_TO.txt* helper files to build the Linux kernel, the U-Boot and the TF-A, then the following new directories and files contain the build outputs.

```
Developer-Package
├── stm32mp1-openstlinux-5.10-dunfell-mp1-21-03-31  Source code for OpenSTLinux
distribution
│   └── sources
│       └── arm-ostl-linux-gnueabi
│           ├── FIP_artifacts                    FIP images to flash
│           │   ├── fip/                                FIP images to flash, need to edit
tsv file to point on that fip images
│           │   ├── arm-trusted-firmware/               Intermediate binaries of tf-a
required to build fip images
│           │   ├── u-boot/                             Intermediate binaries of u-boot
required to build fip images
│           │   ├── optee/                              Intermediate binaries of  optee
required to build fip images
```

```
│           ├── linux-5.10.10
│           │   ├── build                                Linux kernel build directory
│           │   │   ├── install_artifact
│           │   │   │   ├── lib
│           │   │   │   │   └── modules
│           │   │   │   │       └── X.Y.Z
│           │   │   │   │           └── [...]            Kernel modules hierarchy (*.ko
files)
│           │   │   │   └── boot
│           │   │   │       ├── stm32mp157a-dk1[*].dtb  Linux kernel device tree blob
files for bootfs partition → STM32MP157A-DK1
│           │   │   │       ├── stm32mp157c-dk2[*].dtb  Linux kernel device tree blob
files for bootfs partition → STM32MP157C-DK2
│           │   │   │       ├── stm32mp157c-ev1[*].dtb  Linux kernel device tree blob
files for bootfs partition → STM32MP157C-EV1
│           │   │   │       ├── uImage                  Linux kernel binary image file
(with U-Boot wrapper) for bootfs partition
│           │   │   │       └── [...]
│           │   │   ├── vmlinux                         Debug symbol file for Linux kernel
│           │   │   └── [...]
│           │   └── [...]
```

```
│           ├── optee-os-stm32mp-3.12.0.r1                                    OP-
TEE OS installation directory
│           │   ├── build                                                     OP-TEE
 OS build directory
│           │   │   ├── tee-[header/pageable/pager]_v2-stm32mp157a-dk1-optee.stm32  Binari
es for OP-TEE OS runtime services (trusted boot chain) → STM32MP157A-DK1
│           │   │   ├── tee-[header/pageable/pager]_v2-stm32mp157c-dk2-optee.stm32  Binari
es for OP-TEE OS runtime services (trusted boot chain) → STM32MP157C-DK2
│           │   │   ├── tee-[header/pageable/pager]_v2-stm32mp157c-ev1-optee.stm32  Binari
es for OP-TEE OS runtime services (trusted boot chain) → STM32MP157C-EV1
│           │   │   └── [...]
│           │   └── [...]
```

Example of directory structure for Packages

```
│           ├──tf-a-stm32mp-2.4.r1                                  TF-A installation directory
│           │   ├── build                                           TF-A build directory
│           │   │   ├── optee                                       TF-A, with OP-TEE OS
│           │   │   │   ├── tf-a-bl2-optee.elf                      Debug symbol file for TF-A
, with OP-TEE OS → trusted boot firmware stage
│           │   │   │   ├── tf-a-stm32mp157a-dk1-optee.stm32        TF-A binary for FSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157A-DK1
│           │   │   │   ├── tf-a-stm32mp157c-dk2-optee.stm32        TF-A binary for FSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157C-DK2
│           │   │   │   ├── tf-a-stm32mp157c-ev1-optee.stm32        TF-A binary for FSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157C-EV1
│           │   │   │   └── [...]
│           │   │   └── trusted                                     TF-A, without OP-TEE OS
│           │   │       ├── tf-a-bl2-trusted.elf                    Debug symbol file for TF-A
 → trusted boot firmware stage
│           │   │       ├── tf-a-bl32-trusted.elf                   Debug symbol file for TF-A
 → trusted boot firmware stage
│           │   │       ├── tf-a-stm32mp157a-dk1-trusted.stm32      TF-A binary for FSBL
partition (trusted boot chain) → STM32MP157A-DK1
│           │   │       ├── tf-a-stm32mp157c-dk2-trusted.stm32      TF-A binary for FSBL
partition (trusted boot chain) → STM32MP157C-DK2
│           │   │       ├── tf-a-stm32mp157c-ev1-trusted.stm32      TF-A binary for FSBL
partition (trusted boot chain) → STM32MP157C-EV1
│           │   │       └── [...]
│           │   └── [...]
```

```
│           └── u-boot-stm32mp-2020.10.r1                           U-Boot installation
directory
│               ├── build-basic                                     U-Boot build directory
for basic boot chain
│               │   ├── u-boot-spl.stm32-stm32mp157a-dk1-basic      U-Boot binary for FSBL
partition (basic boot chain) → STM32MP157A-DK1
│               │   ├── u-boot-spl.stm32-stm32mp157c-dk2-basic      U-Boot binary for FSBL
partition (basic boot chain) → STM32MP157C-DK2
│               │   ├── u-boot-spl.stm32-stm32mp157c-ev1-basic      U-Boot binary for FSBL
partition (basic boot chain) → STM32MP157C-EV1
│               │   ├── u-boot-stm32mp157a-dk1-basic.img            U-Boot binary for SSBL
partition (basic boot chain) → STM32MP157A-DK1
│               │   ├── u-boot-stm32mp157c-dk2-basic.img            U-Boot binary for SSBL
partition (basic boot chain) → STM32MP157C-DK2
│               │   ├── u-boot-stm32mp157c-ev1-basic.img            U-Boot binary for SSBL
partition (basic boot chain) → STM32MP157C-EV1
│               ├── build-optee                                     U-Boot build directory
for trusted boot chain, with OP-TEE OS
│               │   ├── u-boot-stm32mp157a-dk1-optee.elf            Debug symbol file for U-
Boot, with OP-TEE OS → STM32MP157A-DK1
│               │   ├── u-boot-stm32mp157a-dk1-optee.stm32          U-Boot binary for SSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157A-DK1
│               │   ├── u-boot-stm32mp157c-dk2-optee.elf            Debug symbol file for U-
Boot, with OP-TEE OS → STM32MP157C-DK2
│               │   ├── u-boot-stm32mp157c-dk2-optee.stm32          U-Boot binary for SSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157C-DK2
│               │   ├── u-boot-stm32mp157c-ev1-optee.elf            Debug symbol file for U-
Boot, with OP-TEE OS → STM32MP157C-EV1
│               │   ├── u-boot-stm32mp157c-ev1-optee.stm32          U-Boot binary for SSBL
partition (trusted boot chain), with OP-TEE OS → STM32MP157C-EV1
│               │   └── [...]
│               ├── build-trusted                                   U-Boot build directory
for trusted boot chain
│               │   ├── u-boot-stm32mp157a-dk1-trusted.elf          Debug symbol file for U-
Boot → STM32MP157A-DK1
│               │   ├── u-boot-stm32mp157a-dk1-trusted.stm32        U-Boot binary for SSBL
partition (trusted boot chain) → STM32MP157A-DK1
```

Example of directory structure for Packages

```
│                   │   ├── u-boot-stm32mp157c-dk2-trusted.elf      Debug symbol file for U-
Boot → STM32MP157C-DK2
│                   │   ├── u-boot-stm32mp157c-dk2-trusted.stm32    U-Boot binary for SSBL
partition (trusted boot chain) → STM32MP157C-DK2
│                   │   ├── u-boot-stm32mp157c-ev1-trusted.elf      Debug symbol file for U-
Boot → STM32MP157C-EV1
│                   │   ├── u-boot-stm32mp157c-ev1-trusted.stm32    U-Boot binary for SSBL
partition (trusted boot chain) → STM32MP157C-EV1
│                   │   └── [...]
│                   └── [...]
```

7 Appendix B: directory structure after build (Distribution Package)

Provided you have followed the build method explained in OpenSTLinux distribution, then the following new directories contain the build outputs.

As long as you did not modify the source code:

- the files in STPurple are the same as the ones available in the **Starter Package**: flash layout, binaries for *bootfs*, *rootfs*, *userfs* and *vendorfs* partitions
- the files in grey are the same as the ones available in the **Starter and Developer Packages**: binaries for *FSBL* and *SSBL* partitions, and for OP-TEE OS runtime services
- the files in orange are the same as the ones available in the **Developer Package**: Linux kernel image and device tree blobs, and debug symbol files

```
Distribution-Package/openstlinux-5.10-dunfell-mp1-21-03-31 /build-openstlinuxweston-
stm32mp/tmp-glibc/deploy
├── images
│   └── stm32mp1
│       ├── flashlayout_st-image-weston                                  Flash layout
files (description of the partitions) for the supported flash devices
│       │   ├── FlashLayout_emmc_stm32mp157c-ev1-optee.tsv               Flash layout file
for eMMC and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   ├── FlashLayout_emmc_stm32mp157c-ev1-trusted.tsv             Flash layout file
for eMMC and trusted boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_nand-4-256_stm32mp157c-ev1-optee.tsv         Flash layout file
for NAND Flash and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   ├── FlashLayout_nand-4-256_stm32mp157c-ev1-trusted.tsv       Flash layout file
for NAND Flash and trusted boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-emmc_stm32mp157c-ev1-optee.tsv           Flash layout file
for NOR Flash (and eMMC) and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-emmc_stm32mp157c-ev1-trusted.tsv         Flash layout file
for NOR Flash (and eMMC) and trusted boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-nand-4-256_stm32mp157c-ev1-optee.tsv     Flash layout file
for NOR Flash (and NAND Flash) and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-nand-4-256_stm32mp157c-ev1-trusted.tsv   Flash layout file
for NOR Flash (and NAND Flash) and trusted boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-sdcard_stm32mp157c-ev1-optee.tsv         Flash layout file
for NOR Flash (and microSD card) and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   ├── FlashLayout_nor-sdcard_stm32mp157c-ev1-trusted.tsv       Flash layout file
for NOR Flash (and microSD card) and trusted boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_sdcard_stm32mp157a-dk1-basic.tsv             Flash layout file
for microSD card and basic boot chain → STM32MP157A-DK1
│       │   ├── FlashLayout_sdcard_stm32mp157a-dk1-optee.tsv             Flash layout file
for microSD card and trusted boot chain, with OP-TEE OS → STM32MP157A-DK1
│       │   ├── FlashLayout_sdcard_stm32mp157a-dk1-trusted.tsv           Flash layout file
for microSD card and trusted boot chain (recommended setup) → STM32MP157A-DK1
│       │   ├── FlashLayout_sdcard_stm32mp157c-dk2-basic.tsv             Flash layout file
for microSD card and basic boot chain → STM32MP157C-DK2
│       │   ├── FlashLayout_sdcard_stm32mp157c-dk2-optee.tsv             Flash layout file
for microSD card and trusted boot chain, with OP-TEE OS → STM32MP157C-DK2
│       │   ├── FlashLayout_sdcard_stm32mp157c-dk2-trusted.tsv           Flash layout file
for microSD card and trusted boot chain (recommended setup) → STM32MP157C-DK2
│       │   ├── FlashLayout_sdcard_stm32mp157c-ev1-basic.tsv             Flash layout file
for microSD card and basic boot chain → STM32MP157C-EV1
│       │   ├── FlashLayout_sdcard_stm32mp157c-ev1-optee.tsv             Flash layout file
```

Example of directory structure for Packages

```
for microSD card and trusted boot chain, with OP-TEE OS → STM32MP157C-EV1
│       │   └── FlashLayout_sdcard_stm32mp157c-ev1-trusted.tsv          Flash layout file
for microSD card and trusted boot chain (recommended setup) → STM32MP157C-EV1
│       │   └── [...]
│       ├── scripts
│       │   └── create_sdcard_from_flashlayout.sh
```

```
│       ├── st-image-bootfs-openstlinux-weston-stm32mp1.ext4            Binary for bootfs
partition
│       ├── st-image-userfs-openstlinux-weston-stm32mp1.ext4            Binary for userfs
partition
│       ├── st-image-vendorfs-openstlinux-weston-stm32mp1.ext4          Binary for vendorf
s partition
│       ├── st-image-weston-openstlinux-weston-stm32mp1.ext4            Binary for rootfs
partition
│       ├── fip/                                                       FIP images flashed
according flashlayout selected
│       ├── uImage                                                      Linux kernel
binary image file (with U-Boot wrapper) for bootfs partition
│       ├── vmlinux                                                     Debug symbol file
for Linux kernel
│       └── [...]
└── [...]
```

Software development kit (A programming package that enables a programmer to develop applications for a specific platform.)

Linux® is a registered trademark of Linus Torvalds.

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

Trusted Firmware for Arm® Cortex®-A

Open Portable Trusted Execution Environment

Operating System

Flash memories combine high density and cost effectiveness of EPROMs with the electrical erasability of EEPROMs. For this reason, the Flash memory market is one of the most exciting areas of the semiconductor industry today and new applications requiring in system reprogramming, such as cellular telephones, automotive engine management systems, hard disk drives, PC BIOS software for Plug & Play, digital TV, set top boxes, fax and other modems, PC cards and multimedia CD-ROMs, offer the prospect of very high volume demand.

former spelling for e•MMC ('e' in italic)

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

Cortex®

Microprocessor Unit

Board support package

Cortex Microcontroller Software Interface Standard

Hardware Abstraction Layer

Firmware Image Package is a packaging format used by TF-A

Advanced Linux sound architecture

Graphics Processing Units

Open Graphics Library (See http://www.opengl.org/ for more details)

Open Vector Graphics (See http://www.khronos.org/openvg/ for more details)

Khronos Native Platform Graphics Interface (See http://www.khronos.org/egl/ for more details)

First Stage Boot Loader

Second Stage Boot Loader

Flash memory shortened to gain space in titles, tables and block diagrams