



OpenSTLinux distribution



A quality version of this page, approved on *20 February 2019*, was based off this revision.

Template:ArticleMainWriter

Template:ArticleApprovedVersion

Contents

1 What is the OpenSTLinux distribution?	3
2 Software architecture overview	4
3 OpenSTLinux concept	5
3.1 Layers	5
3.2 Machines	5
3.3 Images	6
3.4 Distros	6
3.5 Reference source code	6
4 How to get the software for this distribution	7
5 References	8



1 What is the OpenSTLinux distribution?

The OpenSTLinux distribution, running on the Arm[®] Cortex[®]-A processor(s), is a sub-part of the STM32MPU Embedded Software distribution.

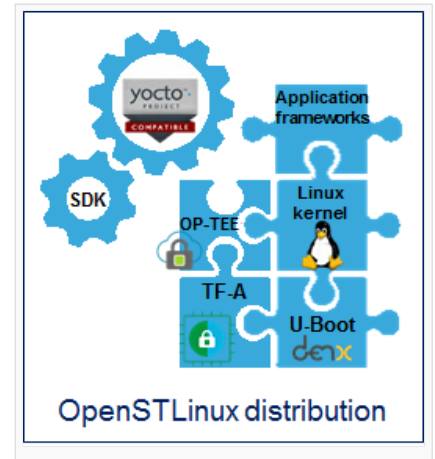
OpenSTLinux is a Linux[®] distribution based on the OpenEmbedded build framework.

It includes the following collection of software components:

- OpenSTLinux BSP (OP-TEE secure OS, boot chain and Linux kernel)
- Application frameworks such as the following Linux application frameworks (non-exhaustive list):
 - Wayland-Weston as a display/graphic framework
 - Gstreamer as a multimedia framework
 - Advanced Linux Sound Architecture (ALSA) libraries

As explained in the OpenEmbedded article, the files used to build an image are stored in layers that come from different sources, and that are configured for this image.

Only layers specific to the OpenSTLinux distribution (for example the OpenSTLinux Board Support Package layer) are detailed in the Layers chapter below. Community layers referenced in this article are not listed. [Template:ProductMarketingReviewsComments](#)



The Machines chapter introduces the machine metadata (the information needed to build a kernel for specific target boards), while the Images chapter specifies the images that contain the information needed to build the user space. The Distros chapter describes the available distros configurations.

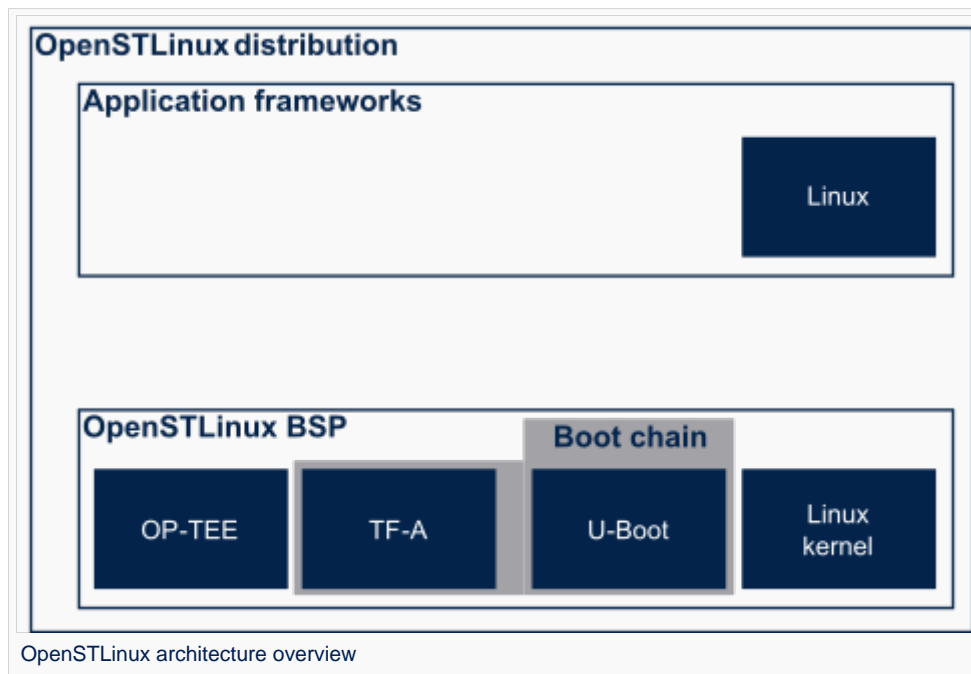


2 Software architecture overview

The figure below gives an overview of the OpenSTLinux architecture. You can obtain more details by clicking on one of the sub-levels.

The **Linux application frameworks** (or Linux user space components) that rely on the OpenSTLinux BSP, mainly come from open-source communities.

OpenEmbedded core components (community layers) combined with STMicroelectronics-specific layers (for example the BSP layer) give a consistent starting point to develop customer applications based on a standardized interface.





3 OpenSTLinux concept

To build an OpenSTLinux based software, a combination of machine, image and distro must be selected. Available machines, images and distros are listed below.

More information on how to compile and use an OpenSTLinux distribution can be found in the **Distribution Package** articles available in [category:Distribution Package](#).

3.1 Layers

This chapter describes the layers developed by STMicroelectronics. [Template:ProductMarketingReviewsComments](#)

STMicroelectronics' strategy is to organize the layers in order to split BSP descriptions from applications and frameworks. By doing this, any BSP can be combined with any framework, or no framework at all.

Layer	Description
meta-st-stm32mp	BSP Layer for stm32mp
meta-st-openstlinux	OpenSTLinux layer - framework/image settings

For the detailed content of the layers, please check [STM32MP15_OpenSTLinux_release_note#Detailed delivery content](#)

In addition, in OpenSTLinux distribution, a layer (meta-st-stm32mp-addons) has been created to manage STM32CubeMX integration.

If necessary, you can also create your own layer. The procedure to do so is explained in [How_to_create_a_new_open_embedded_layer](#).

Layer	Description
meta-st-stm32mp-addons	BSP Layer addons for stm32mp (CubeMX Machine)
meta-my-custo-layer	framework addons/customization

3.2 Machines

Machine	Description
stm32mp1	Machine configuration for STM32MP1 microprocessor device boards
stm32mp1-eval	Machine configuration for STM32MP1 evaluation board
stm32mp1-disco	Machine configuration for STM32MP1 disco board



3.3 Images

Image	Description
Official images	
st-image-weston	OpenSTLinux weston image with basic Wayland support (if enabled in distro)
Supported images	
st-image-core	OpenSTLinux core image
Proposed images as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-x11	ST example of image based on X11
st-example-image-xfce	ST example of image based on XFCE framework
st-example-image-gtk	ST example of image based on GTK framework

3.4 Distros

For further details on distributions (distros), read "Creating a distribution" ^[1]

Distro	Description
openstlinux-eglfs	OpenSTLinux featuring eglfs - no X11, no Wayland
openstlinux-weston	OpenSTLinux featuring Weston/Wayland
openstlinux-x11	OpenSTLinux featuring X11
nodistro	*** DEFAULT OPENEMBEDDED SETTING : DISTRO is not defined ***

3.5 Reference source code

The TF-A, U-Boot, OP-TEE and kernel components have been configured by default to use **tarball + patches** for source code.

If you prefer to use **github**[®] as a source code reference, you need to update your local.conf file

Refer to the section 'Configure STM32MP default version to github' in local.conf :

```
# =====
# Configure STM32MP default version to github
# =====
#STM32MP_SOURCE_SELECTION_pn-linux-stm32mp = "github"
#STM32MP_SOURCE_SELECTION_pn-optee-os-stm32mp = "github"
#STM32MP_SOURCE_SELECTION_pn-tf-a-stm32mp = "github"
#STM32MP_SOURCE_SELECTION_pn-u-boot-stm32mp = "github"
```

Then just uncomment the line to switch the component to github[®] for source code.



4 How to get the software for this distribution

The OpenSTLinux distribution is available through the three Packages (Starter, Developer, and Distribution) of the STM32MPU Embedded Software ecosystem.



5 References

- <http://www.yoctoproject.org/docs/2.4/mega-manual/mega-manual.html#creating-your-own-distribution> Creating a distribution
Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))