



OpenSTLinux distribution



Contents

1. OpenSTLinux distribution	3
2. ALSA overview	9
3. Category:Distribution Package	15
4. GStreamer overview	22
5. How to create a new open embedded layer	28
6. OpenEmbedded	34
7. OpenSTLinux BSP architecture overview	40
8. OpenSTLinux architecture overview	46
9. STM32MP15 OpenSTLinux release note	52
10. STM32MPU Embedded Software distribution	58
11. U-Boot overview	64
12. Wayland Weston overview	70



CLASSIFICATION: UNCLASSIFIED, SECURITY REVISION: UNCLASSIFIED, DATE: 11/16/20

A quality version of this page, approved on *17 November 2020*, was based off this revision.

Contents

1 What is the OpenSTLinux distribution?	4
2 Software architecture overview	5
3 OpenSTLinux concept	6
3.1 Layers	6
3.2 Machines	6
3.3 Images	6
3.4 Distros	7
3.5 Reference source code	7
4 How to get the software for this distribution	8
5 References	9



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.

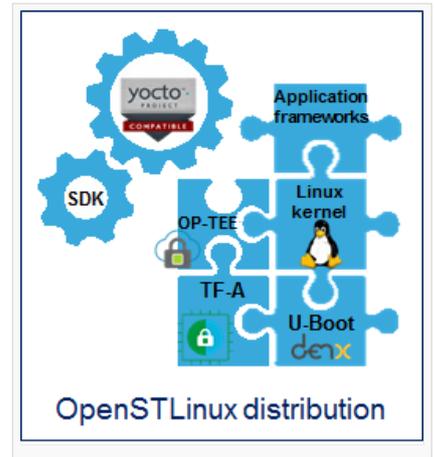
OpenSTLinux is a Linux[®] distribution based on the [OpenEmbedded](#) build framework.

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.



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.

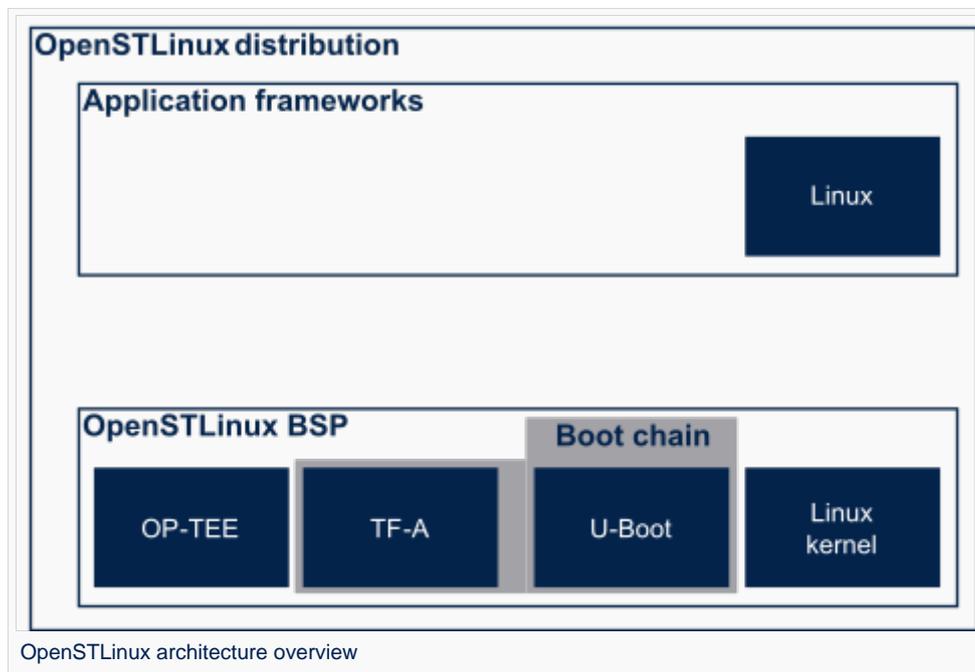


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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

Stable: 05.01.2021 - 15:38 / Revision: 04.01.2021 - 17:18

Contents

1 What is the OpenSTLinux distribution?	10
2 Software architecture overview	11
3 OpenSTLinux concept	12
3.1 Layers	12
3.2 Machines	12
3.3 Images	12
3.4 Distros	13
3.5 Reference source code	13
4 How to get the software for this distribution	14
5 References	15

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.

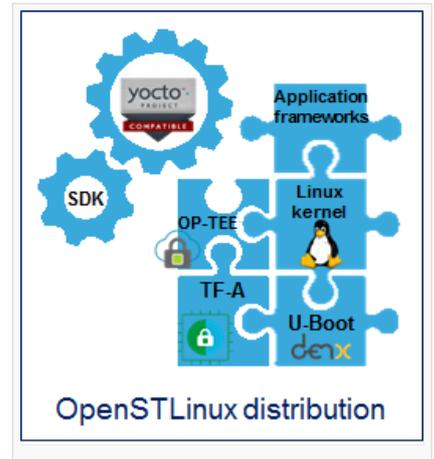
OpenSTLinux is a Linux[®] distribution based on the [OpenEmbedded](#) build framework.

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.



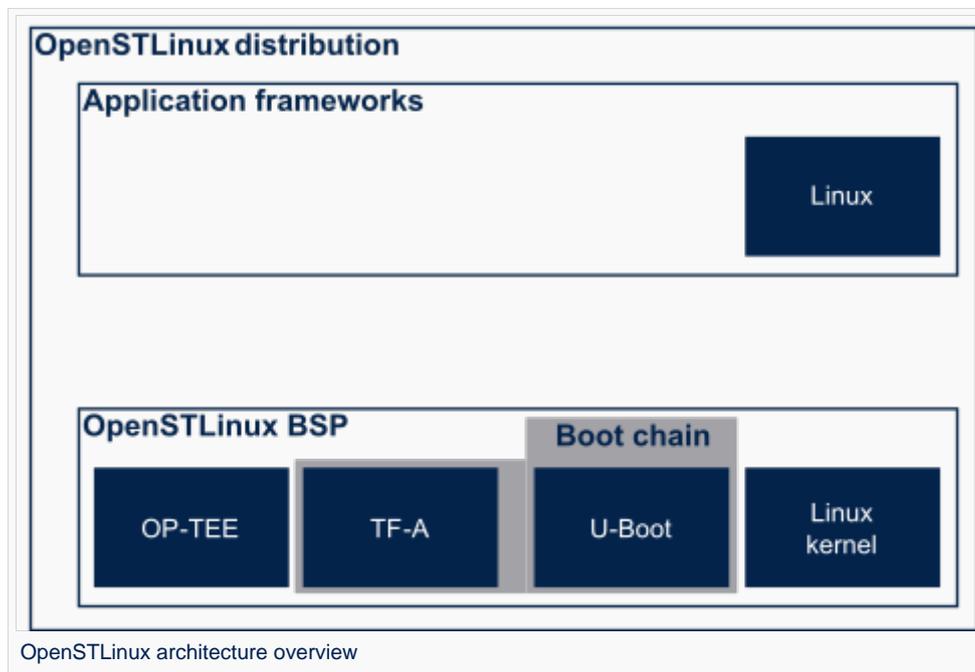
The [Machines](#) chapter introduces the machine metadata (the information needed to build a kernel for specific target boards), while the [Images](#) chapter specifies the images that contain the information needed to build the user space. The [Distros](#) chapter describes the available distros configurations.

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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

Stable: 17.06.2020 - 15:26 / Revision: 16.01.2020 - 13:44

Contents

1 What is the OpenSTLinux distribution?	16
2 Software architecture overview	17
3 OpenSTLinux concept	18
3.1 Layers	18
3.2 Machines	18
3.3 Images	18
3.4 Distros	19
3.5 Reference source code	19
4 How to get the software for this distribution	20
5 References	21



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.

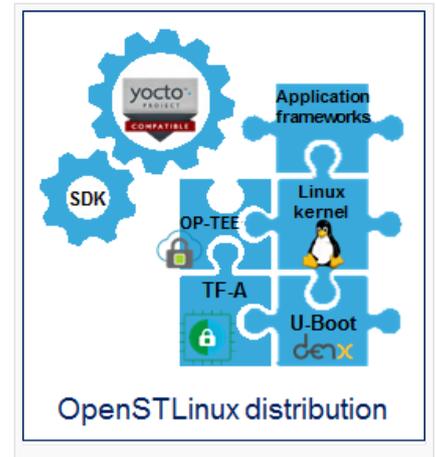
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

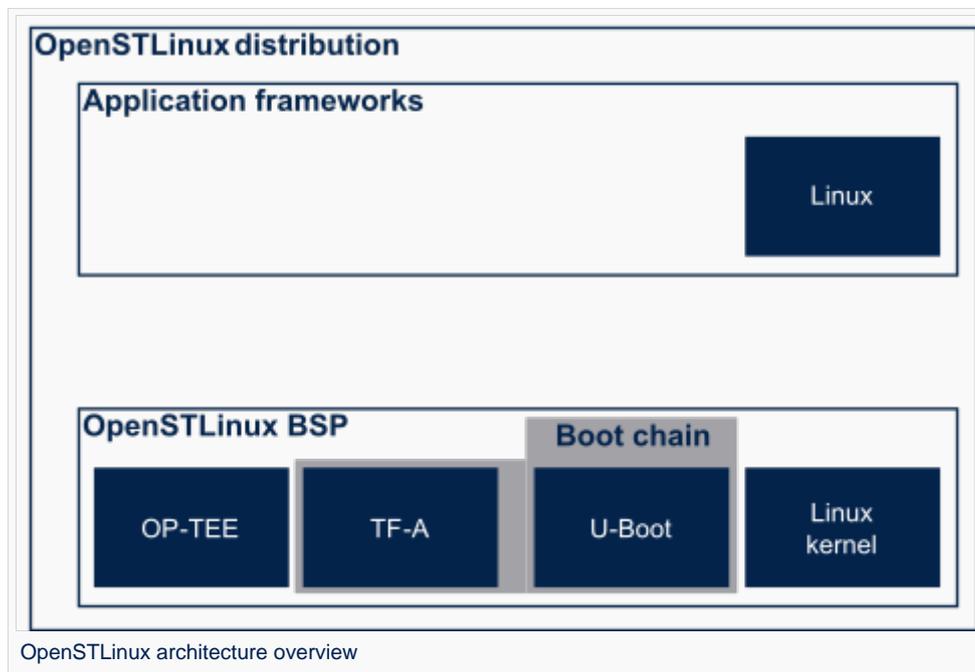


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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



Pages in category "Distribution Package"

The following 6 pages are in this category, out of 6 total.

- [How to add a customer application](#)
- [How to create your own machine](#)
- [How to cross-compile with the Distribution Package](#)
- [How to customize the Linux kernel](#)
- [STM32MP1 Distribution Package](#)
- [STM32MP1 Distribution Package for Android](#)

Stable: 23.03.2021 - 10:38 / Revision: 16.03.2021 - 17:17

Contents

1 What is the OpenSTLinux distribution?	23
2 Software architecture overview	24
3 OpenSTLinux concept	25
3.1 Layers	25
3.2 Machines	25
3.3 Images	25
3.4 Distros	26
3.5 Reference source code	26
4 How to get the software for this distribution	27
5 References	28



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.

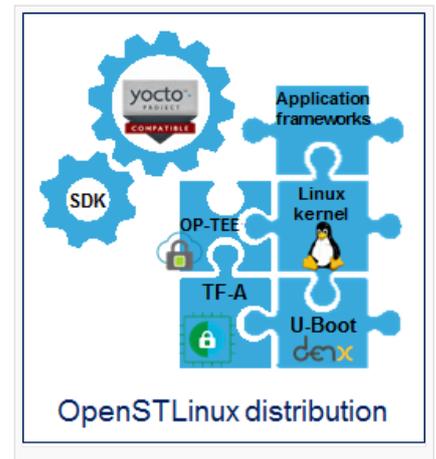
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

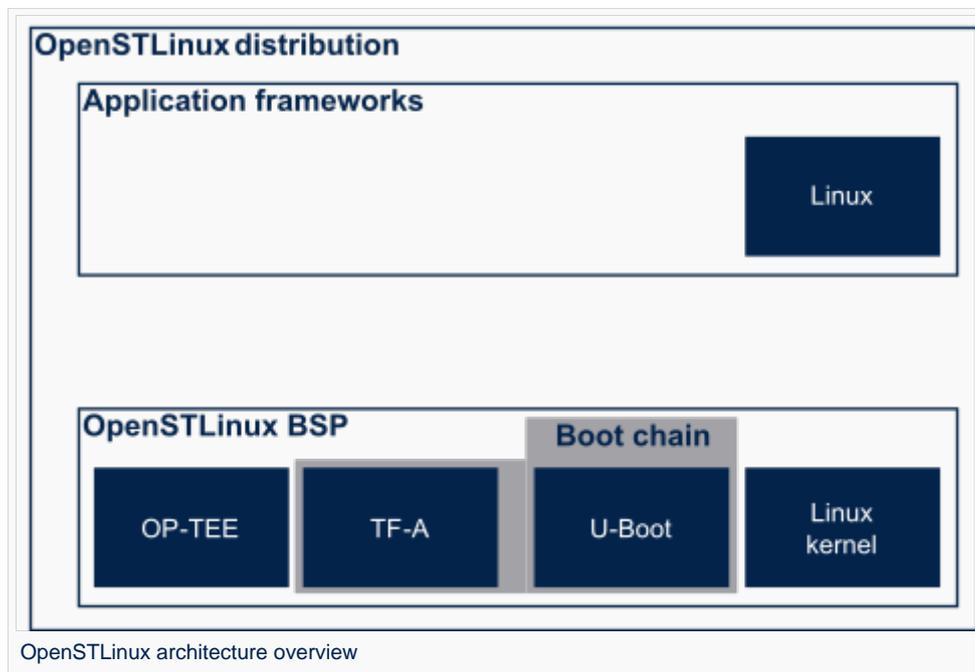


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 19.10.2020 - 14:17 / Revision: 19.10.2020 - 14:16

Contents

1 What is the OpenSTLinux distribution?	29
2 Software architecture overview	30
3 OpenSTLinux concept	31
3.1 Layers	31
3.2 Machines	31
3.3 Images	31
3.4 Distros	32
3.5 Reference source code	32
4 How to get the software for this distribution	33
5 References	34



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.

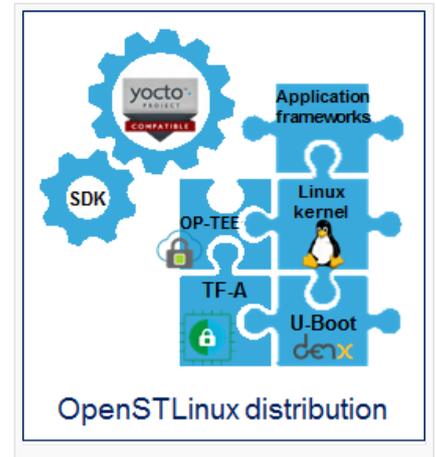
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

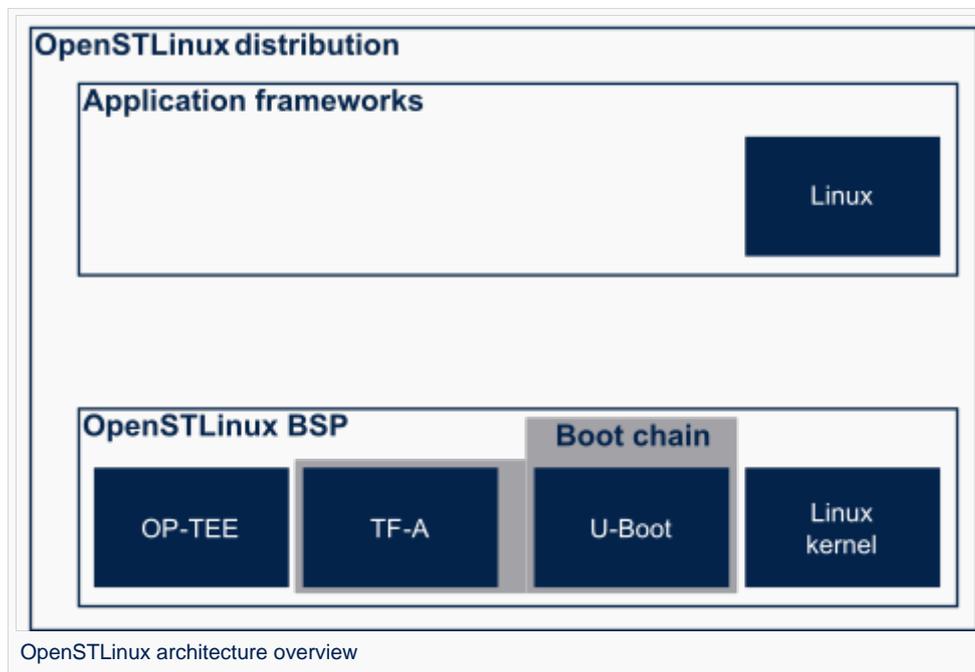


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 09.12.2020 - 17:46 / Revision: 09.12.2020 - 13:33

Contents

1 What is the OpenSTLinux distribution?	35
2 Software architecture overview	36
3 OpenSTLinux concept	37
3.1 Layers	37
3.2 Machines	37
3.3 Images	37
3.4 Distros	38
3.5 Reference source code	38
4 How to get the software for this distribution	39
5 References	40

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.

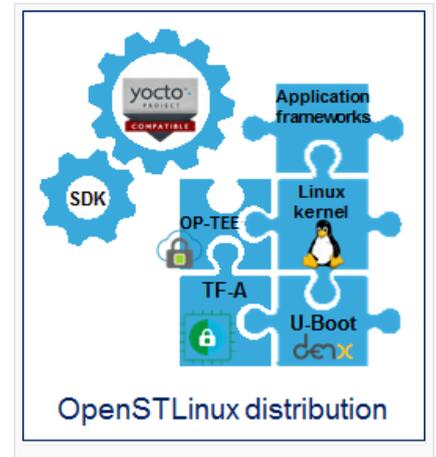
OpenSTLinux is a Linux[®] distribution based on the [OpenEmbedded](#) build framework.

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.



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.

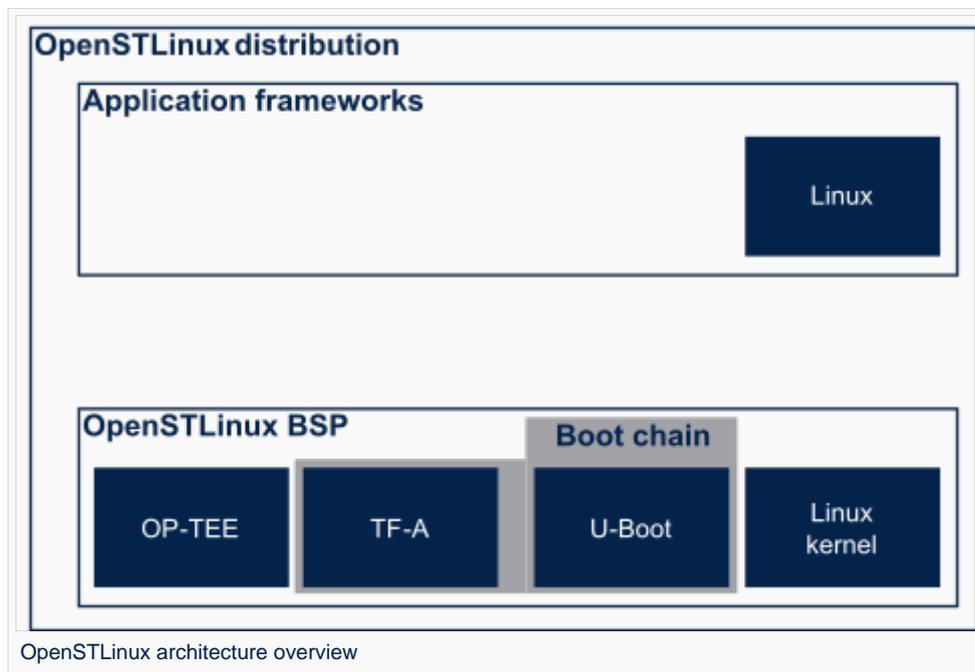


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 15.10.2019 - 09:53 / Revision: 15.10.2019 - 09:41

Contents

1 What is the OpenSTLinux distribution?	41
2 Software architecture overview	42
3 OpenSTLinux concept	43
3.1 Layers	43
3.2 Machines	43
3.3 Images	43
3.4 Distros	44
3.5 Reference source code	44
4 How to get the software for this distribution	45
5 References	46



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.

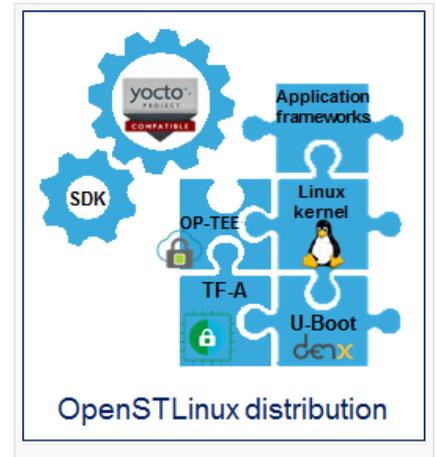
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

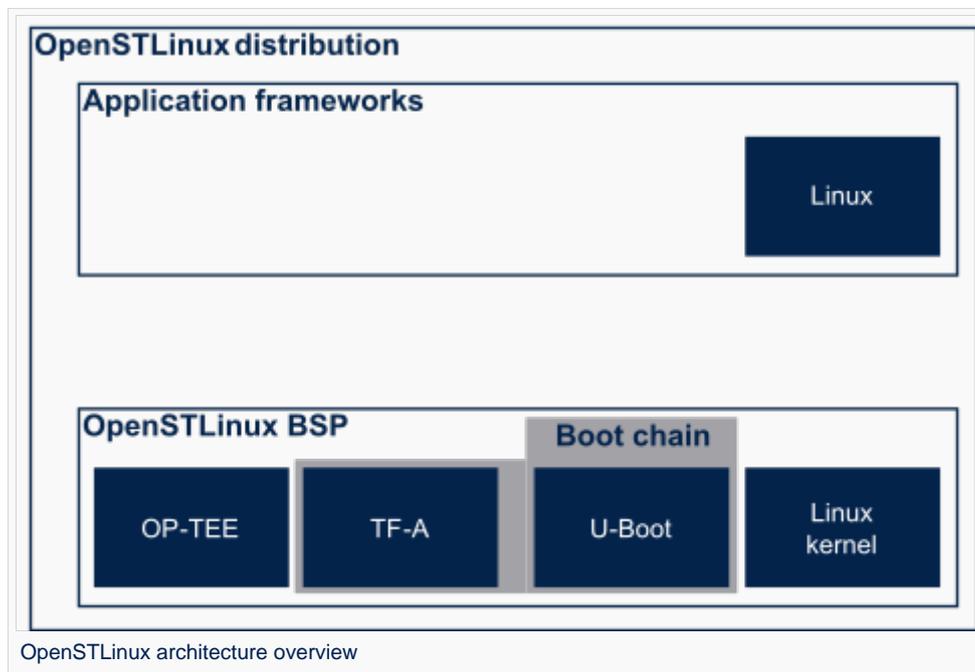


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 26.03.2021 - 13:18 / Revision: 12.03.2021 - 11:26

Contents

1 What is the OpenSTLinux distribution?	47
2 Software architecture overview	48
3 OpenSTLinux concept	49
3.1 Layers	49
3.2 Machines	49
3.3 Images	49
3.4 Distros	50
3.5 Reference source code	50
4 How to get the software for this distribution	51
5 References	52



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.

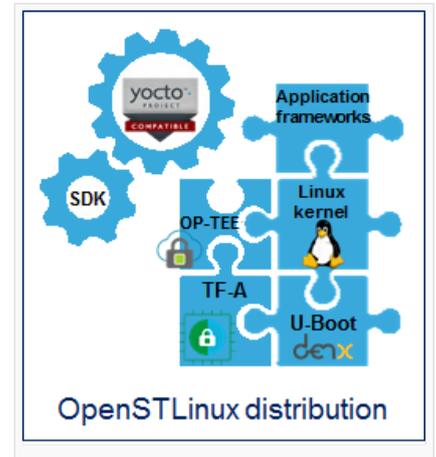
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

[OpenSTLinux architecture overview](#)



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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 17.11.2021 - 16:43 / Revision: 17.11.2021 - 09:37

Contents

1 What is the OpenSTLinux distribution?	53
2 Software architecture overview	54
3 OpenSTLinux concept	55
3.1 Layers	55
3.2 Machines	55
3.3 Images	55
3.4 Distros	56
3.5 Reference source code	56
4 How to get the software for this distribution	57
5 References	58



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.

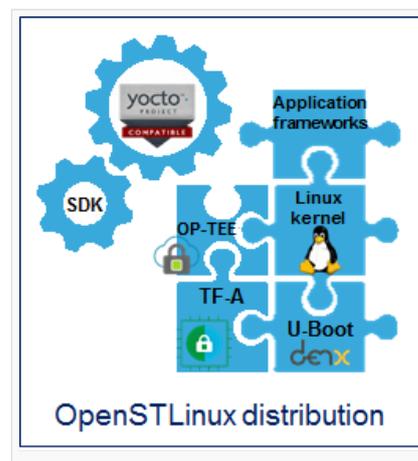
OpenSTLinux is a Linux[®] distribution based on the [OpenEmbedded](#) build framework.

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.



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.

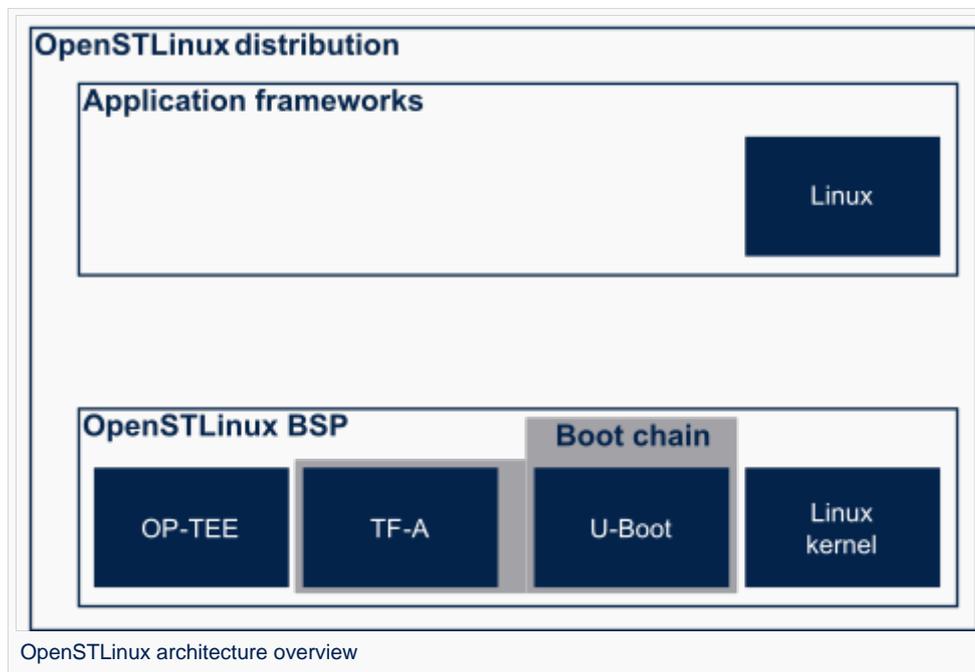


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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

Stable: 15.04.2020 - 14:24 / Revision: 15.04.2020 - 14:21

Contents

1 What is the OpenSTLinux distribution?	59
2 Software architecture overview	60
3 OpenSTLinux concept	61
3.1 Layers	61
3.2 Machines	61
3.3 Images	61
3.4 Distros	62
3.5 Reference source code	62
4 How to get the software for this distribution	63
5 References	64



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.

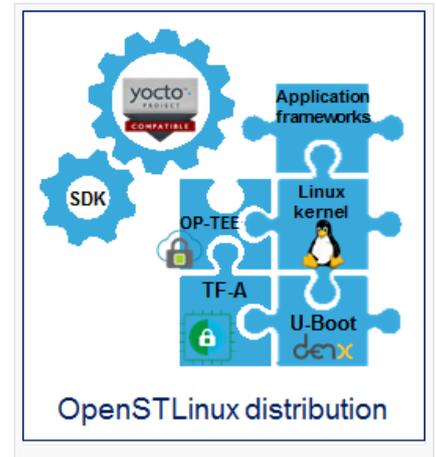
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



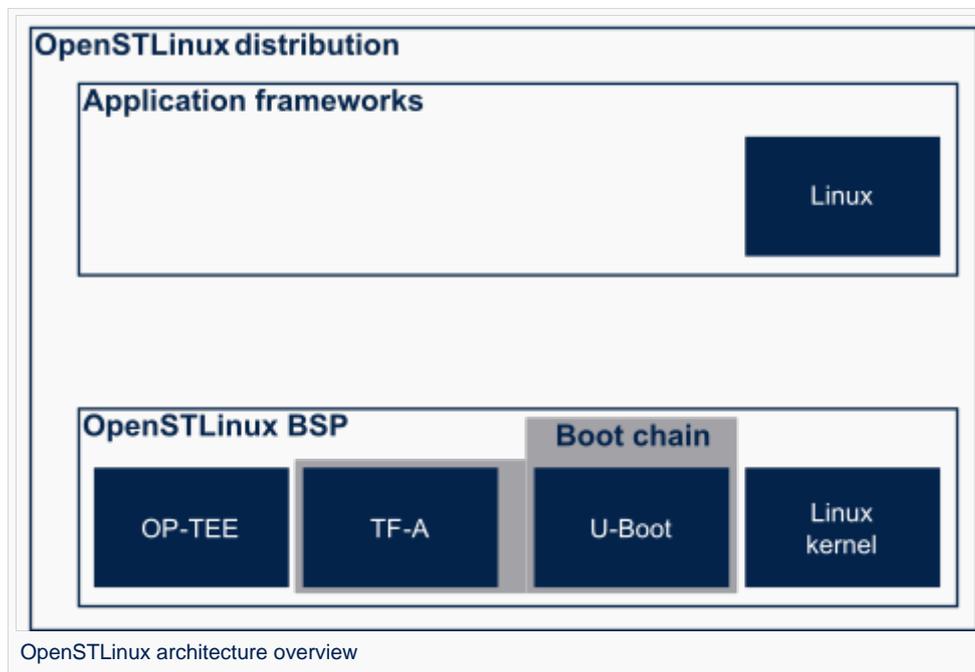
The [Machines](#) chapter introduces the machine metadata (the information needed to build a kernel for specific target boards), while the [Images](#) chapter specifies the images that contain the information needed to build the user space. The [Distros](#) chapter describes the available distros configurations.

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- Creating a distribution

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

Stable: 19.10.2021 - 13:54 / Revision: 19.10.2021 - 13:54

Contents

1 What is the OpenSTLinux distribution?	65
2 Software architecture overview	66
3 OpenSTLinux concept	67
3.1 Layers	67
3.2 Machines	67
3.3 Images	67
3.4 Distros	68
3.5 Reference source code	68
4 How to get the software for this distribution	69
5 References	70



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.

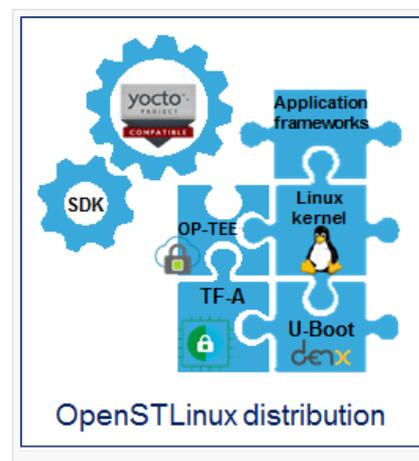
OpenSTLinux is a Linux® distribution based on the [OpenEmbedded](#) build framework.

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.



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.

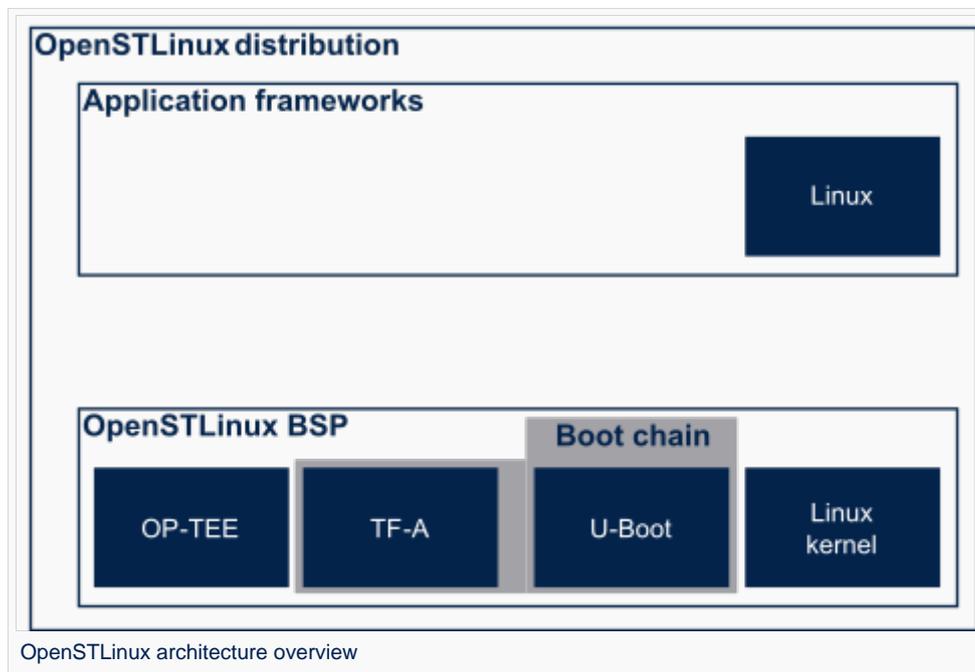


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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

Stable: 19.03.2021 - 16:08 / Revision: 16.03.2021 - 17:20

Contents

1 What is the OpenSTLinux distribution?	71
2 Software architecture overview	72
3 OpenSTLinux concept	73
3.1 Layers	73
3.2 Machines	73
3.3 Images	73
3.4 Distros	74
3.5 Reference source code	74
4 How to get the software for this distribution	75
5 References	76

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.

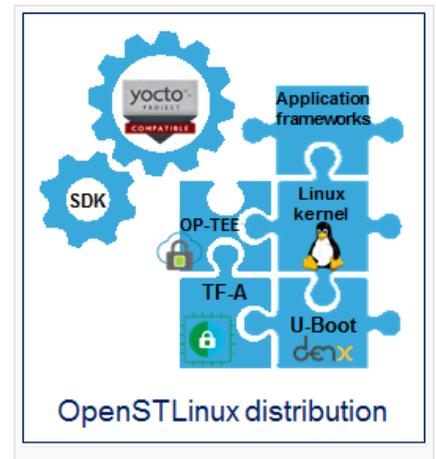
OpenSTLinux is a Linux[®] distribution based on the [OpenEmbedded](#) build framework.

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.



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.

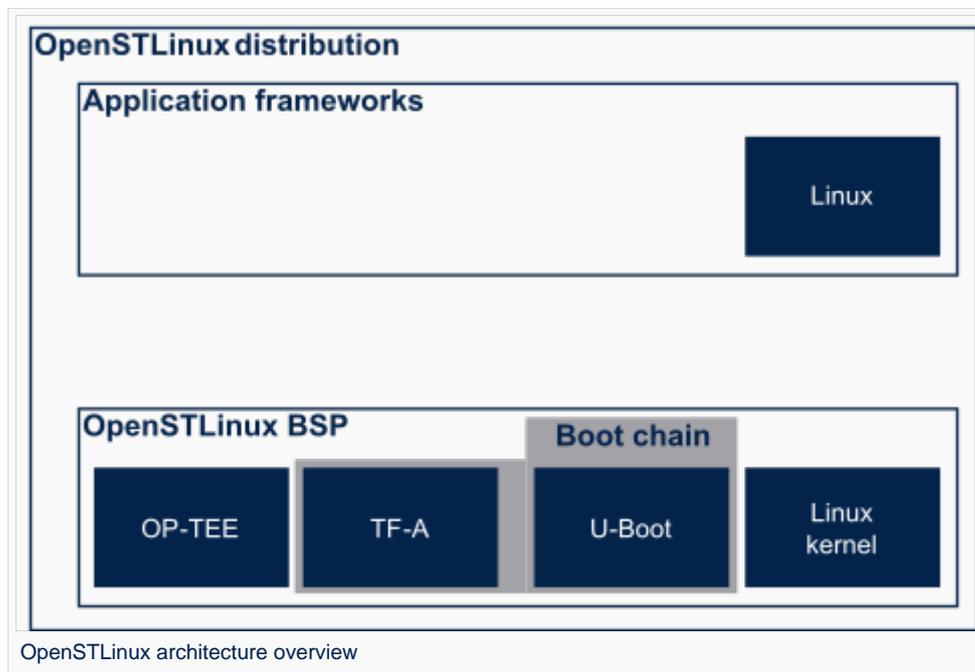


2 Software architecture overview

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.

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	



Image	Description
Official images	
st-image-core	OpenSTLinux core image
Images, proposed as example only	
st-example-image-qt	ST example of image based on QT framework
st-example-image-qtwayland	ST example of image based on QT framework (available for ecosystem release v2.1.0)

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
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 distribution.



5 References

- [Creating a distribution](#)

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