



## How to compile the device tree with the Distribution Package

---

### How to compile the device tree with the Distribution Package



A quality version of this page, approved on *30 March 2021*, was based off this revision.

## Contents

1 Introduction .....	3
2 Creating a new open embedded layer for your demo .....	4
2.1 Update layer.conf file .....	4
2.2 Create the machine for your demo .....	5
2.2.1 Prepare the machine configuration file .....	5
2.2.2 Configure the machine configuration file for your demo .....	5
2.3 Associate EULA with the new demo machine .....	6
2.4 Move DeviceTree files and project coming from STM32CubeMX tool .....	6
2.5 Update the README file .....	6
2.6 Clean up useless content .....	6
3 Adding specific recipes and content necessary for your demo .....	8



### 1 Introduction

---

This article is intended for Yocto experts, or people who have some practical experience of the Yocto environment.

This section describes the steps needed to create and configure a demo layer using DeviceTree files from the STM32CubeMX tool, and to add and configure a machine similar to those already supported by the OpenSTLinux Distribution Package (in particular the machine delivered inside the existing STM32MP BSP layer 'addons').

Reminder: this addon-layer is deployed under the following path in the delivery : **<path of OpenSTLinux distribution delivery>/layers/meta-st/meta-st-stm32mp-addons/**



## 2 Creating a new open embedded layer for your demo

You first need to create a new layer. See the latest [How to create a new open embedded layer](#)

After creation, you have under `<path of OpenSTLinux distribution delivery>/layers/meta-st/`:

```
$ tree meta-my-demo-layer
meta-my-demo-layer
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
├── recipes-example
│   └── example
│       └── example.bb
3 directories, 4 files
```

### 2.1 Update layer.conf file

Open the layer.conf file and add the lines below for the licenses, demo layer path, and dependency with the STM32MP BSP layer 'addons' :

```
EULA_FILE_ST_stm32mpmydemo = "${LAYERDIR}/conf/eula/${MACHINE}"
EULA_FILE_ST_MD5SUM_stm32mpmydemo = "8b505090fb679839cefbcc784afe8ce9"

#Inform bitbake for adding another location to search for licenses
LICENSE_PATH += "${LAYERDIR}/files/licenses"

# Set a variable to get the STM32MP MX BSP location
STM32MP_MY_DEMO_BASE = "${LAYERDIR}"

# This should only be incremented on significant changes that may
# cause compatibility issues with other layers
LAYERVERSION_meta-my-demo-layer = "1"

LAYERDEPENDS_meta-my-demo-layer = "stm-st-stm32mp-mx"

# OpenEmbedded compatibility information
# This should only be incremented on significant changes that will
# cause compatibility issues with other layers
LAYERVERSION_meta-my-demo-layer = "1"
LAYERSERIES_COMPAT_meta-my-demo-layer = "dunfell"
```

#### Information

LAYERSERIES\_COMPAT must be aligned with the version of OpenEmbedded used.  
Please refer to <https://wiki.yoctoproject.org/wiki/Releases>



## 2.2 Create the machine for your demo

### 2.2.1 Prepare the machine configuration file

- Copy the machine delivered inside the existing STM32MP BSP layer 'addons' into your demo layer

```
$ cp <path of OpenSTLinux distribution delivery>/layers/meta-st/meta-st-stm32mp-addons
/conf/machine/stm32mp1-mx.conf <path of OpenSTLinux distribution delivery>/layers/meta-st
/meta-my-demo-layer/conf/machine/stm32mp1-demo.conf
```

- Open stm32mp1-demo.conf and update the line below

```
#@NEEDED_BSPLAYERS: layers/meta-openembedded/meta-oe layers/meta-openembedded/meta-python
layers/meta-st/meta-st-stm32mp-addons
```

- Replace STM32MP\_MX\_BASE by **STM32MP\_MY\_DEMO\_BASE**
- Add these lines after the series of includes:

```
# Define specific common machine name
MACHINEOVERRIDES .= ":stm32mpmydemo"
```

### 2.2.2 Configure the machine configuration file for your demo

In the customer machine file, move to *User machine customization sections* paragraph to configure your machine:

- **Boot Scheme**

To select your boot scheme configuration(s), comment and uncomment the *BOOTSCHHEME\_LABELS* lines.

- **Boot Device Choice**

To select your boot device configuration(s), comment and uncomment the *BOOTDEVICE\_LABELS* lines.

- **Support Feature Choice**

To select additional features to enable on board, uncomment the "MACHINE\_FEATURES" proposed lines.

- **Specific firmwares and kernel modules configuration**

This section allows user to configure some specificities related to its board hardware.

- *KERNEL\_MODULE\_AUTOLOAD* you may need to feed this variable with the list of kernel modules that need to be loaded at boot time, such as 'goodix' for current touch-screen used on STM32MP157C-EV1 evaluation board.

- *BLUETOOTH\_LIST* in case you enable the bluetooth feature for your machine, you should set, at least, the firmware module to use for your hardware (e.g. 'linux-firmware-bluetooth-bcm4343' for STM32MP157C-DK2 discovery board).

- *WIFI\_LIST* in case you enable the wifi feature for your machine, you should set, at least, the firmware module to use for your hardware (e.g.'linux-firmware-bcm43430' for STM32MP157C-DK2 discovery board).

- **CubeMX Project config**

You have to uncomment and configure the following variables to set your CubeMX project:

- *CUBEMX\_DTB* name of CubeMX generated device tree files, without file extension (e.g. stm32mp157c-<ProjectName>-mx)

- *CUBEMX\_PROJECT* path of CubeMX generated device tree files relative to layer path folder (e.g. mx/STM32MP157C-EV1/my-demo/DeviceTree/my-demo)



In order to give a better view on how to configure these variables, some machine samples are provided to show how to set-up a disco and eval board cubeMX machine: refer to conf/machine/examples from meta-st-stm32mp-addons layer.

### 2.3 Associate EULA with the new demo machine

Copy the eula folder delivered inside the existing STM32MP BSP layer 'addons' into your demo layer

```
$ cp -rf <path of OpenSTLinux distribution delivery>/layers/meta-st/meta-st-stm32mp-addons
/conf/eula/ <path of OpenSTLinux distribution delivery>/layers/meta-st/meta-my-demo-layer
/conf/.
```

Then replace the existing symbolic link with the machine used for your demo:

```
$ rm stm32mp1-mx
$ ln -s ST_EULA_SLA stm32mp1-demo
```

### 2.4 Move DeviceTree files and project coming from STM32CubeMX tool

The principle is that the user generates devicetree files for the targeted demo from the STM32CubeMX tool.

#### Warning

**Most of the time, generated devicetree files - mainly user sections - must be reworked by the end user for compilation and functional purposes. Each demo is delivered with an application note that describe changes applied on STM32CubeMX devicetree files**

These files are then moved into the "mx" folder created into your demo layer : <path of OpenSTLinux distribution delivery>/layers/meta-st/meta-my-demo-layer/mx/

Sub-folders are created and populated with the generated devicetree files:

```
mx/${CUBEMX_PROJECT}/kernel
mx/${CUBEMX_PROJECT}/u-boot
mx/${CUBEMX_PROJECT}/tf-a
mx/${CUBEMX_PROJECT}/optee-os
```

With **CUBEMX\_PROJECT** that is equal to the value defined inside the machine used for the demo.

### 2.5 Update the README file

Please update the README file with the information needed for building and executing the demo.

### 2.6 Clean up useless content

You can delete the content of the recipes-example folder created by the create-layer command.



After making all of the updates, your demo layer should be similar to:

```

$ tree meta-my-demo-layer
meta-my-demo-layer
├── conf
│   ├── eula
│   │   ├── ST_EULA_SLA
│   │   ├── stm32mp1-demo -> ST_EULA_SLA
│   │   └── ...
│   ├── layer.conf
│   └── machine
│       └── stm32mp1-demo.conf
├── COPYING.MIT
├── mx
│   └── STM32MP157C-EV1
│       └── my-demo
│           ├── DeviceTree
│           │   └── my-demo
│           │       ├── kernel
│           │       │   └── stm32mp157c-my-demo.dts
│           │       ├── tf-a
│           │       │   ├── stm32mp157c-my-demo.dts
│           │       │   └── stm32mp15-mx.h
│           │       ├── u-boot
│           │       │   ├── stm32mp157c-my-demo.dts
│           │       │   ├── stm32mp157c-my-demo-u-boot.dtsi
│           │       │   └── stm32mp15-mx.h
│           │       └── optee-os
│           │           └── stm32mp157c-my-demo.dts
└── README

```



### 3 Adding specific recipes and content necessary for your demo

---

Examples of further add-on components:

- Recipes for installing distro-specific configuration files
- Any image recipes specific to user distribution
- A *psplash append file* for a branded splash screen
- Any other append files to make custom changes

Some other added components (\*bb) are more specific: images, system services, and so on (a non-exhaustive list is shown below):

- Recipes-core for *psplash screen*, *systemd services*
- Recipes-samples for example images

...