Which STM32MPU Embedded Software Package better suits your needs

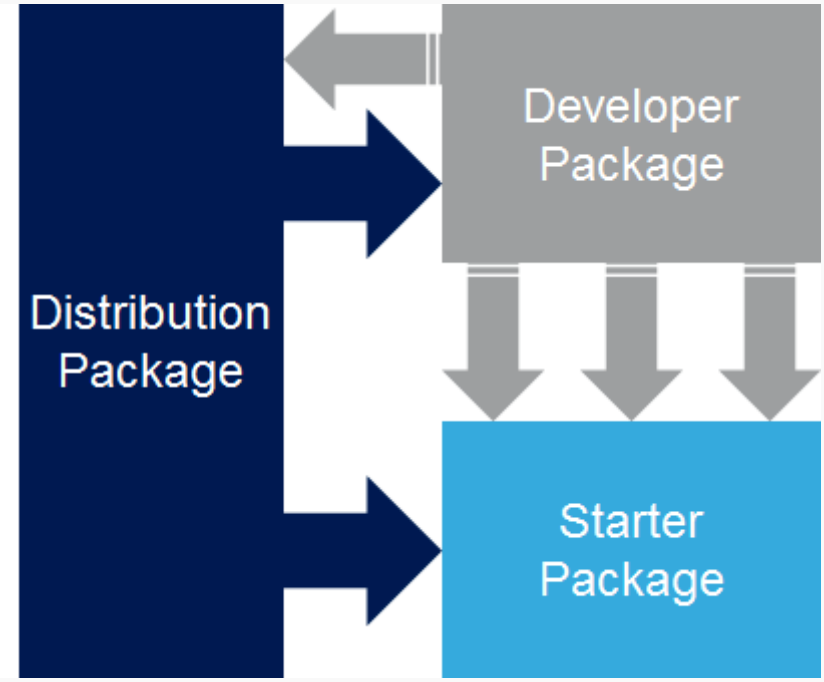
Contents

Stable: 23.07.2020 - 12:41 / Revision: 09.07.2020 - 13:18

A quality version of this page, approved on *23 July 2020*, was based off this revision.

The STM32MPU Embedded Software distribution for STM32 microprocessor platforms supports three software Packages:

- The **Starter Package** to quickly and easily start with any STM32MP microprocessor device. The Starter Package is generated from the Distribution Package.
- The **Developer Package** to add your own developments on top of the STM32MPU Embedded Software distribution, or to replace the Starter Package pre-built binaries. The Developer Package is generated from the Distribution Package.
- The **Distribution Package** to create your own Linux® distribution, your own Starter Package and your own Developer Package.



STM32MPU Embedded Software distribution Packages

This article describes in details the above Packages ("ways of working") and helps finding the one that best fits your use case. Packages are standalone, but they are not isolated one from the others: there are gateways between them.

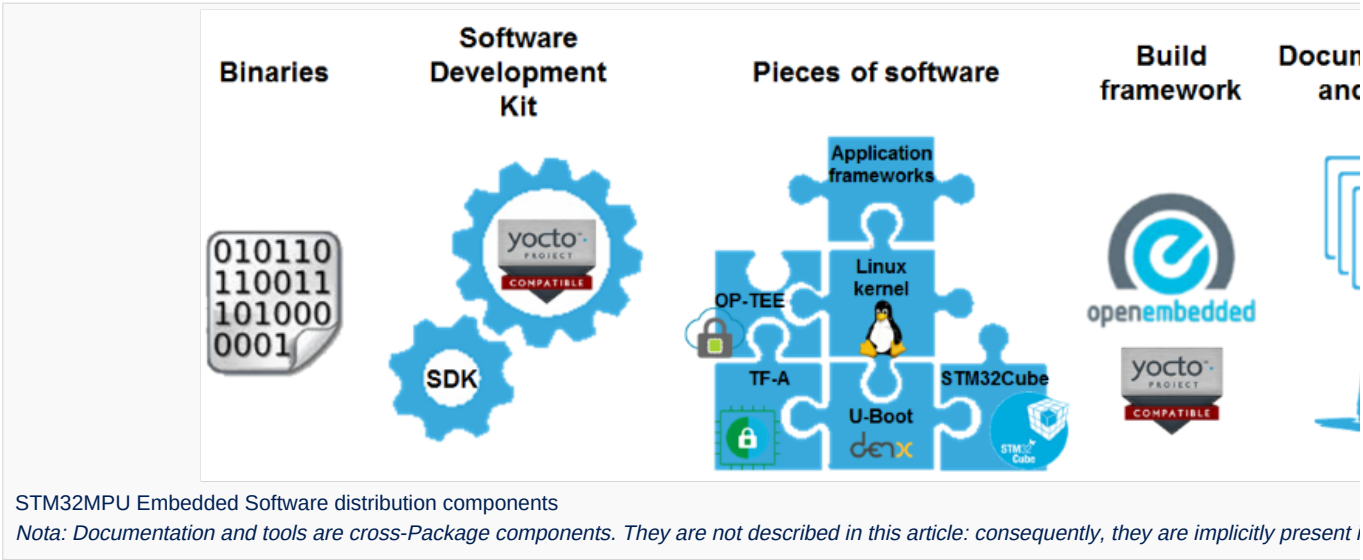
This article starts by an overview of the STM32MPU Embedded Software distribution components.
It concludes by providing an example of production cycle based on the three Packages.

Contents

1 Overview of STM32MPU Embedded Software distribution components

The STM32MPU Embedded Software distribution includes the following components (as illustrated in the figure below):

- Binaries: software images
- Software development kit (SDK):
  - Cross-development toolchain
  - Libraries, headers and symbols
  - Environment setup scripts
- Source code for:
  - The OpenSTLinux distribution (U-Boot, TF-A, OP-TEE, Linux® kernel, application frameworks) running on Arm® Cortex®-A processor(s)
  - The STM32CubeMP1 Package running on Arm® Cortex®-M processor(s)
- Build framework for embedded Linux® (based on OpenEmbedded that is the recommended build system of the Yocto Project):
  - BitBake: a make-like build tool
  - Recipes: a recipe specifies how a particular package is built
  - Layers: a layer is a collection of recipes and/or configurations used on top of the OpenEmbedded core
- Documentation and tools such as:
  - The present wiki
  - The STM32CubeProgrammer
  - Integrated development environments (IDE)



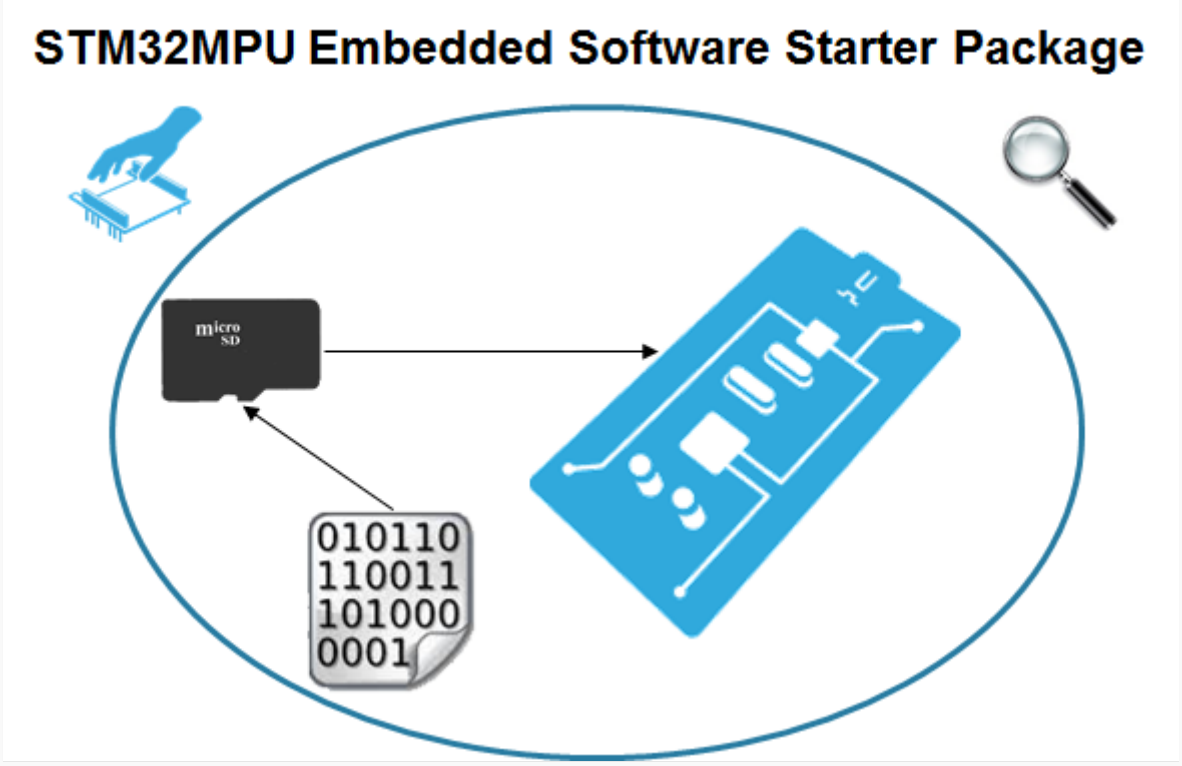
STM32MPU Embedded Software distribution components

Nota: Documentation and tools are cross-Package components. They are not described in this article: consequently, they are implicitly present i

Each of the three Packages described below is made of some of the above components.

2 Starter Package

This Package allows to quickly and easily get any STM32 microprocessor development platform up and running.
An STM32 microprocessor development platform (such as STM32MP157x-EV1 Evaluation board) and a microSD card (populated with the software image delivered for this platform) are all what is required to discover the platform capabilities. No integrated development environment (IDE), no software development kit (SDK) are needed: just use the development platform as a "micro-PC" running Linux®.
The software image contains firmware delivered as examples with regards to the STM32Cube MPU Package (if the considered development platform includes an Arm® Cortex®-M processor).



Starter Package components

What are the main use cases that can be covered with this Package?

- Evaluation of the STM32 microprocessor device and development platform (such as internal peripherals and performance)
- Execution of the Cube M4 firmware projects delivered within each STM32MPU Embedded Software distribution Starter Package (such as the firmware available on the Arm® Cortex®-M processor

  Please read STM32CubeMP1 Package#Starter Package for STM32CubeMP1 article to get started with running Cube M4 firmware
- Direct development on the development platform of simple user examples in different languages (Shell, Python...)

For more information on the Starter Package depending on your development platform, refer to the corresponding Starter Package article named ***Your_development_platform* - Starter Package**.
Example: STM32MP15 Evaluation boards - Starter Package for the STM32MP15 Evaluation boards.
All *Starter Package* articles belong to the Category:Starter Package.

Switch to the next Package if you need to:

- modify or tune any piece of software of the STM32MPU Embedded Software distribution (e.g. Linux® kernel)
- integrate pieces of software from open-source communities or third parties
- develop and integrate your own applications
- benefit from a software development kit (e.g. for powerful cross-development toolchain)
- take advantage of an integrated development environment (IDE).

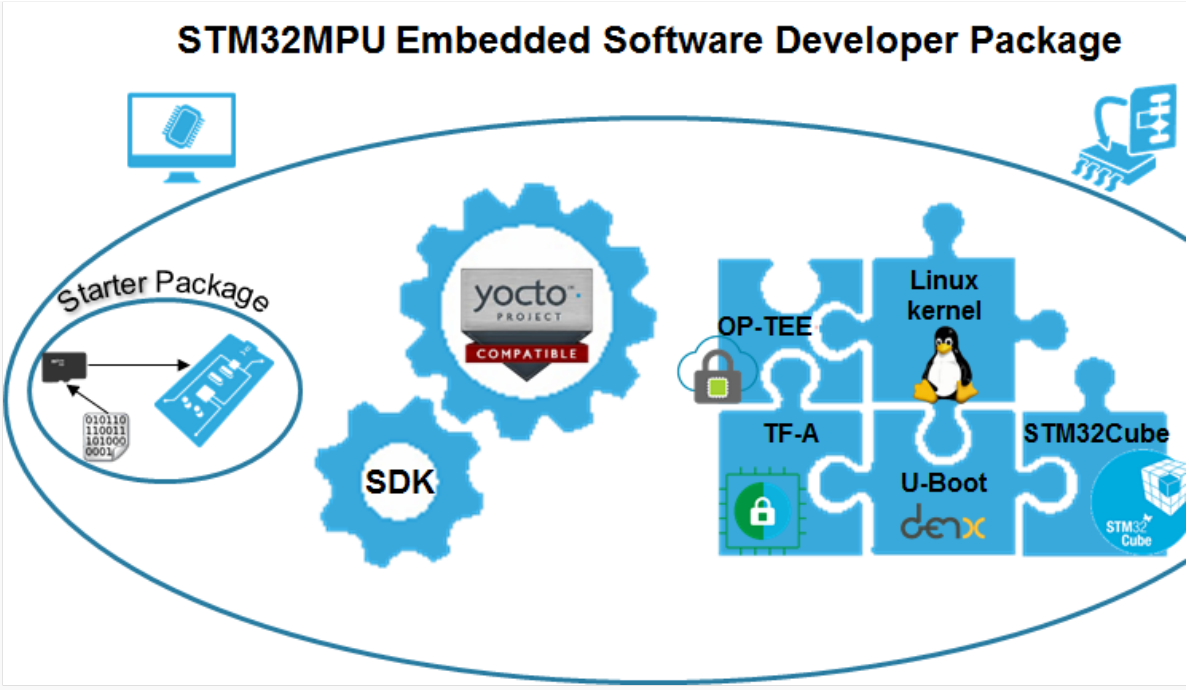
3 Developer Package

This Package allows the modification of some pieces of software of the STM32MPU Embedded Software distribution, as well as the addition of your applications in your platform software image.
The Developer Package provides on top of the Starter Package:

- a software development kit (SDK) for cross-development on an host PC
- the following pieces of software of the OpenSTLinux distribution in **source code**:
  - U-Boot
  - Trusted Firmware-A (TF-A)
  - Linux® kernel
  - Optionally, Open source Trusted Execution Environment (OP-TEE)
- The STM32Cube MPU Package in **source code** (if the considered development platform includes an Arm® Cortex®-M processor)
- Initialization code generator (STM32CubeMX). This tool makes it possible to
  - generate device tree for OpenSTLinux distribution
  - generate the peripheral initialization C code and IDE project creation files for STM32Cube MPU package
- An Eclipse-based integrated development environment (IDE) - STM32CubeIDE
  - STM32Cube firmware projects are supported by *STM32CubeIDE*,
  - Projects files generated by STM32CubeMX can be directly opened by this IDE to continue Cube firmware development,
  - STM32CubeIDE integrates STM32CubeMX for an easier user experience,
  - IDE is ***optional*** for OpenSTLinux development since all actions can be achieved through a command line.

Information

The Developer Package doesn't include the **application frameworks** (Linux® user space applications, U-Boot applications and OP-TEE user space applications) in **source code**.



Developer Package components

What are the main use cases that can be covered with this Package?

- Modification and tuning of the pieces of software delivered as source code (see above list):
  - Code modification
  - Efficient development cycle (compile, push on target, debug)
  - Tuning of the configuration options (e.g. for the Linux® kernel)
  - Adaptation of the device tree files for the microprocessor device and development platform
- Development of the firmware running on the Arm® Cortex®-M processor

  Please read STM32CubeMP1 Package#Developer Package for STM32CubeMP1 article to get started with building and running Cube M4 firmware
- Addition of your own Linux® applications or Linux® application frameworks
- Addition of your own trusted applications (TA) (if OP-TEE is part of the software release)
- Addition of your own U-Boot applications

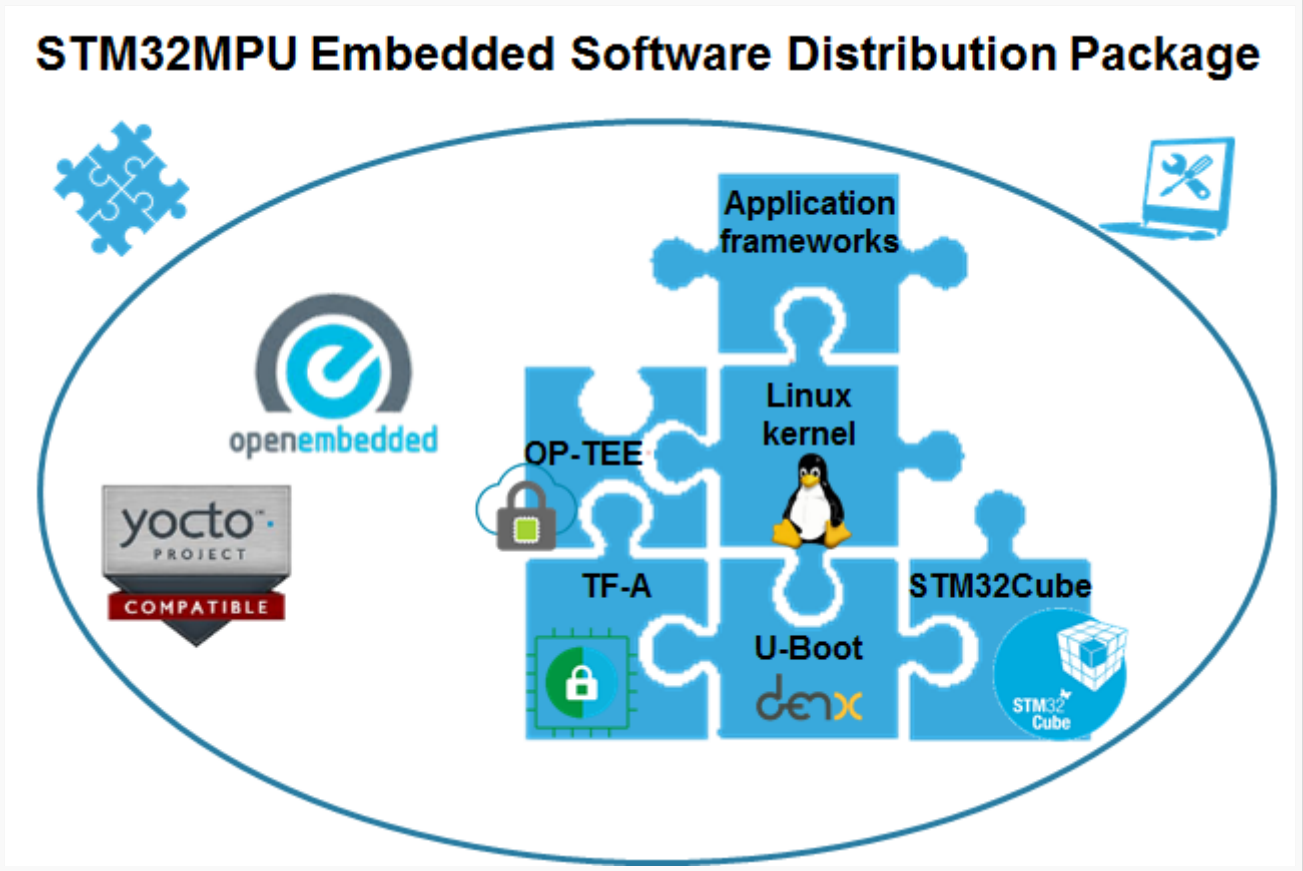
For more information on the Developer Package for a given STM32 microprocessors Series, refer to the corresponding Developer Package article named ***STM32_microprocessors_series*** **Developer Package**.
Example: STM32MP1 Developer Package for the STM32MP1 microprocessors Series.
All *Developer Package* articles belong to the Category:Developer Package.

Switch to the next Package if you need to:

- modify or tune the (U-Boot, OP-TEE and Linux®) application frameworks of the OpenSTLinux distribution
- create your own distribution (e.g. to integrate and share pieces of software from open source communities or third parties)
- generate your own software development kit (SDK) and images

4 Distribution Package

This Package allows the creation of a new distribution with a final objective of productization.
It includes the source code of all the pieces of software of the STM32MPU Embedded Software distribution (including the application frameworks), plus a build framework based on OpenEmbedded (aka distribution builder). A host PC is required to operate this Package (a Linux® host PC is even strongly recommended).



Distribution Package components

What are the main use cases that can be covered with this Package?

- Creation of your own Linux® distribution
  - to tune the system configuration options (such as memory sizes or debug options)
  - to share all the developments within your team (software baseline)
  - to prepare the final software for productization
- Integration of the following software in this distribution:
  - your developments made thanks to the Developer Package
  - pieces of software coming from open source communities or third parties
  - Integrate your Cube M4 firmware projects in your own distribution (binary or source code)

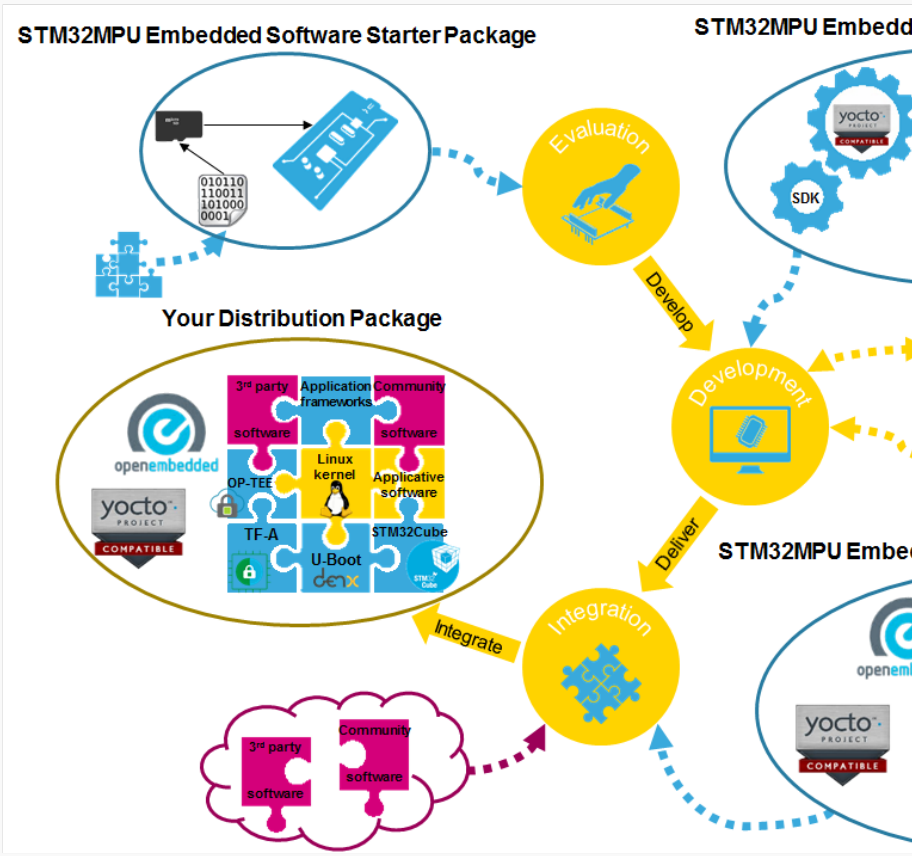
    Please read STM32CubeMP1 Package#Distribution Package for STM32CubeMP1 article to understand how Cube M4 firmware projects have been installed in STM32MPU Embedded software distribution
- Generation of your own Starter Package images
- Generation of your own Developer Package SDK

Generally speaking, few dedicated people from the project team use this Distribution Package to create and regularly enrich the project Linux® distribution and generate the project Developer Package. Most of developers will use this Developer Package for their developments, since it is a light weighted and efficient (short development cycle) environment.

For more information on the Distribution Package for a given STM32 microprocessor Series, refer to the corresponding Distribution Package article named ***STM32MP_microprocessors_series*** **Distribution Package**.
Example: STM32MP1 Distribution Package for the STM32MP1 microprocessors Series. All *Distribution Package* articles belong to the Category:Distribution Package.

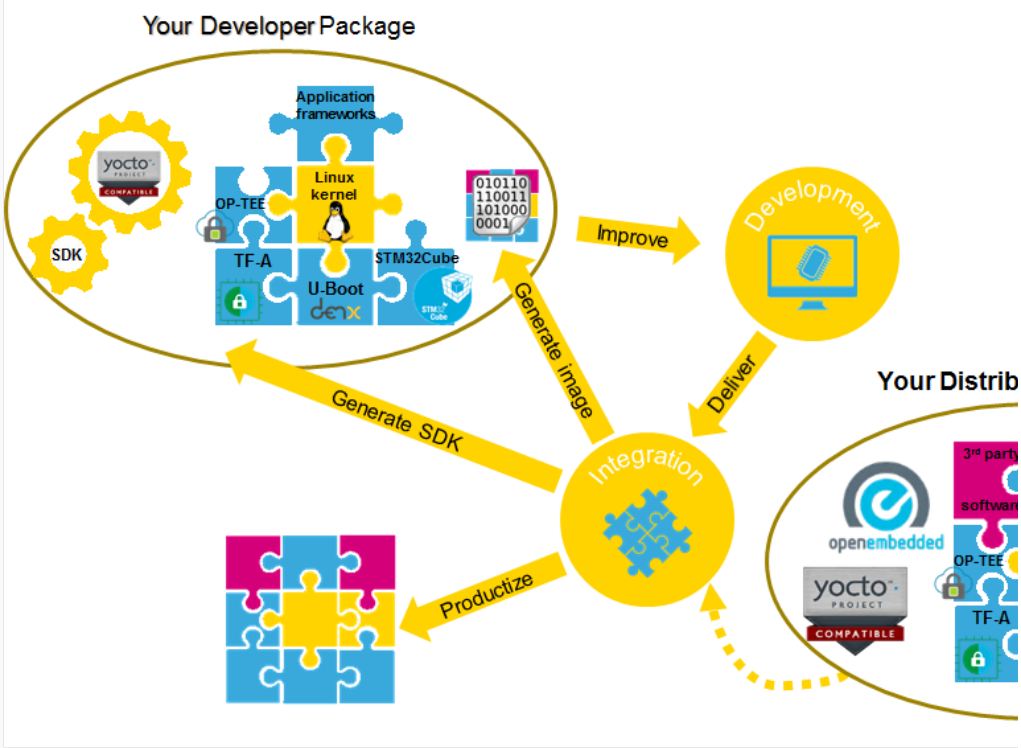
5 Example of production cycle

The following figures give an **example** of gateways between the different Packages, that might be seen as a production cycle.



STM32MPU Embedded Software distribution: evaluation, development and integration

In the example given above, the developers use the Developer Package of the STM32MPU Embedded Software distribution to create an add-on piece of software (e.g. Linux® application on top of the Linux® application frameworks) and modify the Linux® kernel. Their deliveries are integrated on top of the Distribution Package of the STM32MPU Embedded Software distribution, together with pieces of software from an open source community and a third party. The result is your own Distribution Package.



Your distribution: generating, improving and productizing

In the example given above, your new images (binaries) and your new Developer Package are generated from this new Distribution Package. The developers base their additional developments on this new Developer Package. The production cycle is then initiated.

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