

LYE - Learning Yocto Embedded

Nhat-Trieu Huynh-Pham

March 23, 2025

Abstract

This document provides an overview of my journey to learn **Yocto Project** from the ground up, covering key concepts, development workflows, and practical insights gained along the way.

1 Introduction

The history of the **Yocto Project** begins with the need to create a coherent and flexible environment for building software for embedded devices that could support a diversity of hardware architectures and application requirements. Over the years, the Yocto Project has gained broad recognition and support in the industry, becoming the standard for creating custom operating systems for a wide range of devices, from simple gadgets to sophisticated industrial and commercial customized systems.

The **Yocto Project** is not an operating system itself, but a set of tools that allow developers to precisely configure, compile, and deploy their versions of Linux, tailored to the specific requirements of their projects. So, as you can see, it is the perfect solution for many embedded developers! [\[Stu24\]](#)

The rest of this document is structured as follows:

- **Section 2** introduces essential terms related to the Yocto Project.
- **Section 3** details the step-by-step process of building a Yocto-based system.
- **Section 4** presents some examples of customizing a Yocto build to bring-up/port the **Raspberry Pi** hardware.
- **Section 5** provides a final example for hands-on practice.

2 Terms

2.1 BitBake

BitBake is the main build tool of Yocto, responsible for processing configuration files and generating OS images. It is similar to Makefile but more flexible.

2.2 Recipe (Công thức - .bb)

A recipe describes how to compile and install a specific software package. For example, a recipe for `busybox` specifies how to fetch the source code, compile it, and package it.

2.3 Layer

A layer is a collection of recipes, configurations, and patches organized to facilitate management. Examples include:

- `meta-yocto`: The main layer of Yocto.
- `meta-raspberrypi`: A layer supporting Raspberry Pi.
- `meta-custom`: A user-defined custom layer.

2.4 Machine

The target hardware specification for Yocto, defined using the `MACHINE` environment variable. Examples:

- `raspberrypi4-64` for Raspberry Pi 4.
- `qemuarm64` for ARM 64-bit emulation.

2.5 Distro (Bản phối)

A set of policies and configurations that define a specific OS distribution. For example, `poky` is Yocto's default distribution.

2.6 Poky

A reference distribution in Yocto, including BitBake and a set of essential layers. `Poky` helps create a minimal embedded Linux system.

2.7 Image (Hình ảnh hệ điều hành)

The final output of Yocto, containing the kernel, root filesystem, and bootloader. Common image types include:

- `core-image-minimal`: A minimal OS.
- `core-image-full-cmdline`: A full command-line OS.
- `core-image-sato`: A graphical OS.

2.8 SDK (Software Development Kit)

A toolkit for developing applications for Yocto-based systems. It can be generated using:

```
bitbake -c populate_sdk <image-name>
```

2.9 Toolchain

Is a set of tools (compiler, linker, debugger) to compile software for embedded systems.

2.10 devtool

A rapid development tool in Yocto that allows adding, editing, and testing recipes.

2.11 Toaster

Is a web interface to manage Yocto's build system.

3 Build Steps

4 Customization example

5 Final example to practice

References

[Stu24] Scythe Studio. Yocto linux - build your own embedded linux distribution, February 27 2024.

Accessed: 2025-03-23.