**Four elements of Embedded Linux**

-------------------------------------------------

**Toolchain :**

The "Compiler" and "other tools" needed to create code for your target device. Everything else depends on the toolchain.

**Bootloader:**

The "Program" that initializes the board and loads the Linux Kernel.

**Kernel:**

This is the "heart of the system". Managing system resources and interfacing with hardware.

**Root File System:**

"Contains the libraries and programs" that are run once the kernel has completed its initialization.

One more element can be collection of programs specific to your embedded application which make the device do whatever it is supposed to do, be it weigh groceries, display movies, control a robot or fly a drone.

**What is Yocto?**

----------------------

Yocto is the smallest SI metric system prefix

Like 'm' stands for mili (10^-3)

So is yocto ( y = 10^-24 )

**What is Yocto Project?**

---------------------------------

Yocto project provides open source, high quality infrastructure and tools to help developers create their own custom Linux distribution for any hardware architecture.

**History**

-----------

Founded in 2010, in an effort to reduce work duplication, provide resources and information catering to both new and experienced users.

**Collaboration of**

many hardware manufacturers

open source operating systems vendors

electronic companies

Yocto is also a project working group of the Linux Foundation.

**Input Output of Yocto Projects**

---------------------------------------

To understand the outcome provided by the Yocto Project, we can use the analogy of computing machine.

**Input :**

Set of data that describes what we want, that is our specification

(Kernel Configuration, Hardware Name, Packages/Binaries to be installed)

**Output:**

Linux Based Embedded Product

(Linux Kernel, Root File System, Bootloader, Device Tree, Toolchain)

**Setting up Build Machine**

------------------------------------

**Prerequisites:**

1. 50 Gbytes of free disk space

2. Run a supported Linux distribution (i.e. recent releases of Fedora, openSUSE, CentOS, Debian or Ubuntu)

3. Git 1.8.3.1 or higher

tar 1.27 or higher

Python 3.4.0 or higher

Packages and package installation vary depending on your development system.

(\*) Install the required packages for Yocto to Work from

https://www.yoctoproject.org/docs/latest/ref-manual.html#ubuntu-packages

https://docs.yoctoproject.org/ref-manual/system-requirements.html#ubuntu-and-debian

All products will be installed

$ sudo apt-get install gawk wget git-core diffstat unzip textinfo gcc-multilib \

build-essential chrpath socat cpio python python3 python3-pip python3-pexcept \

xz-utils debianutils iputils-ping python3-git pthon3-jinja2 libgl1-mesa \

libsdll1.2-dev pylint3 xterm

$ sudo apt install gawk wget git diffstat unzip texinfo gcc build-essential \

chrpath socat cpio python3 python3-pip python3-pexpect xz-utils debianutils \

iputils-ping python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3 \

xterm python3-subunit mesa-common-dev zstd liblz4-tool

$ sudo apt install make python3-pip

$ sudo pip3 install sphinx sphinx\_rtd\_theme pyyaml

**What is Poky?**

**--------------------**

Poky is a “reference distribution of Yocto Project”. The word "reference" is used to mean "example" in this context.

Yocto Project uses Poky to build images (Kernel, System and application software) for targeted hardware.

At the technical level it is a combined repository of the components.

Bitbake

OpenEmbedded Core

meta-yocto-bsp

Documentation

**Note:** Poky does not contain binary files. It is a working example of how to build your own custom Linux Distribution from source.

**What is difference between Poky and Yocto?**

----------------------------------------------------------------

The exact difference between Yocto and Poky is Yocto refers to the organization (like one would refer to 'canonical', the company behind Ubuntu) and Poky refers to the actual bits downloaded.(analogous to 'Ubuntu')

**What is MetaData?**

---------------------------

**Non Yocto:** A set of data that describes and gives information about other data.

**Yocto World:**

Metadata refers to the “build instructions”

“Commands” and “data” used to indicate what versions of software are used

Where they are obtained from

Changes of additions to the software itself ( pacthes ) which are used to fix bugs or customize the software for use in a particular situation

**Metadata is collection of**

Configuration files (.conf)

Recipes (.bb and .bbappend)

Classes (.bbclass)

Includes (.inc)

**OpenEmbedded Project**

----------------------------------

From http://www.openembedded.org/wiki/Main\_Page

OpenEmbedded offers a best-in-class “cross-compile environment”. It allows developers to create a complete Linux distribution for Embedded Systems.

**What is the difference between OpenEmbedded and the Yocto Project?**

------------------------------------------------------------------------------------------------------

The Yocto Project and OpenEmbedded share a core collection of metadata called openembedded-core.

However, the two organizations remain separate, each with its own focus.

OpenEmbedded “provides a comprehensive set of metadata” for a wide variety of architectures, features, and applications

Not a reference distribution

Designed to be the foundation for others

The Yocto Project focuses on providing powerful, easy-to-use, interoperable, well-tested tools, metadata, and board support packages (BSPs) for a core set of architectures and specific boards.

**OpenEmbedded-Core (oe-core)**

---------------------------------------------

The Yocto Project and OpenEmbedded have agreed to work together and share a common core set of metadata (recipes, classes and associated file): oe-core

**What is BitBake?**

-------------------------

Bitbake is a core component of the Yocto Project.

It is basically performs the same functionality as of make.

It's a task scheduler that parses python and shell script mixed core.

The code parsed generates and runs tasks, which are basically a set of steps ordered according to code's dependencies.

It reads recipes and follows them by fetching packages, building them and incorporating the results into bootable images.

It keeps track of all tasks being processed in order to ensure completion, maximizing the use of processing resources to reduce build time and being predictable

**meta-yocto-bsp**

-----------------------

A Board Support Package (BSP) is a collection of information that defines how to support a particular hardware devices, set of devices, or hardware platform.

The BSP includes information about the hardware features present on the device and kernel configuration information along with any additional hardware drivers required.

The BSP also lists any additional software components required in addition to a generic Linux software stack for both essential and optional platform features.

The meta-yocto-bsp layer in Poky maintains several BSPs such as the beaglebone, EdgeRouter, and generic versions of both 32-bit and 64-bit IA machines.

**Machines supported**

-----------------------------

Texas Instruments BeagleBone (beaglebone)

Freescale MPC8315E-RDB (mpc8315e-rdb)

Intel x86-base PCs and devices (genericx86 and genericx86-64)

Ubiquiti Networks EdgeRouter Lite (edgerouter)

**Note:** To develop on different hardware, you will need to complement Poky with hardware-specific Yocto layers.

**Others**

---------

meta-poky, which is Poky-specific metadata

Documentation, which contains the Yocto Project source files used to make the set of user manuals.

**Conclusions**

------------------

**Poky includes**

Some OE components(oe-core)

Bitbake

Demo-BSP's

Helper scripts to setup environment

Emulator QEMU to test the Image