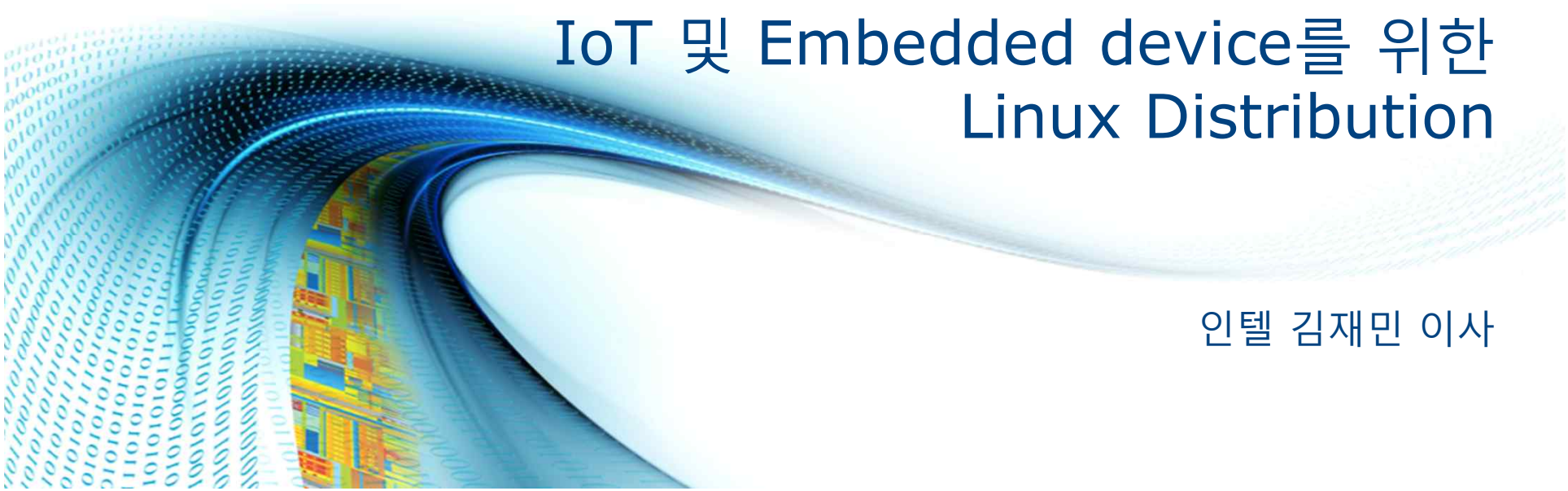


Yocto project

IoT 및 Embedded device를 위한 Linux Distribution

인텔 김재민 이사

Agenda

- What is Yocto Project?
- Why use the Yocto Project?
- Build System Overview
  - Recipe
  - Workflow
  - Layers
  - BSP
- Yocto and OIC
- Yocto and Intel open-hardware Boards
- References

Intel is not just a silicon provider, but

List of companies contributing to the Linux Kernel from Sept 2014 through Jan 2015

- From Linux Foundation

Company	Changes	Total
None	11,968	12.4%
Intel	10,108	10.5%
Red Hat	8,078	8.4%
Linaro	5,415	5.6%
Samsung	4,290	4.4%
Unknown	3,842	4.0%
IBM	3,081	3.2%
SUSE	2,890	3.0%

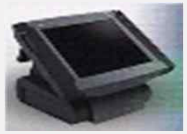
What is Yocto Project?

- Open source collaboration project focused on embedded Linux developers.
- Helps developers to build their own customized Linux distribution for embedded products.
- Provides high quality infrastructure, tools and methods for developers.
- Supported by embedded industry leaders across multiple architectures.
- Intended to provide a starting point for developers.
- Hosted by the Linux Foundation.

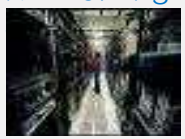
Simple Electronics

Internet of Things

Point of Sale



Networking &



Industrial

Yocto Project Official Website

www.yoctoproject.org

Who is the Yocto Project?

Advisory Board and Technical Leadership

- Organized under the Linux Foundation
- Individual Developers
- Embedded Hardware Companies
- Semiconductor Manufacturers
- Embedded Operating System Vendors
- OpenEmbedded / LTSI Community

THE LINUX FOUNDATION

Member Organizations

Supporting Organizations

https://www.yoctoproject.org/ecosystem

© 2013 The Linux Foundation. All rights reserved.

Why use the Yocto Project?

- It’s not an embedded Linux distribution. It creates a custom for you.
- Small footprints.
- It's a complete embedded Linux development environment with tools, metadata, and documentation.
- Automatically creates an application development SDK customized for each specific devices. (compiler, performance, debug, power analysis, Eclipse)

Why use the Yocto Project?

- Develop using one common Linux OS for all major architecture. (IA, ARM, PowerPC, MIPS, etc)
- Start with a validated collection of software and libraries.
- Flexible framework allowing you to reuse your software stack.
- Enables easy transition from Proof of Concept(POC) to supported Commercial Linux with no loss of optimizations, code or design.
- Proprietary code can be included in build structure within a separate layer, which can be kept private.

Yocto Project Build System Overview

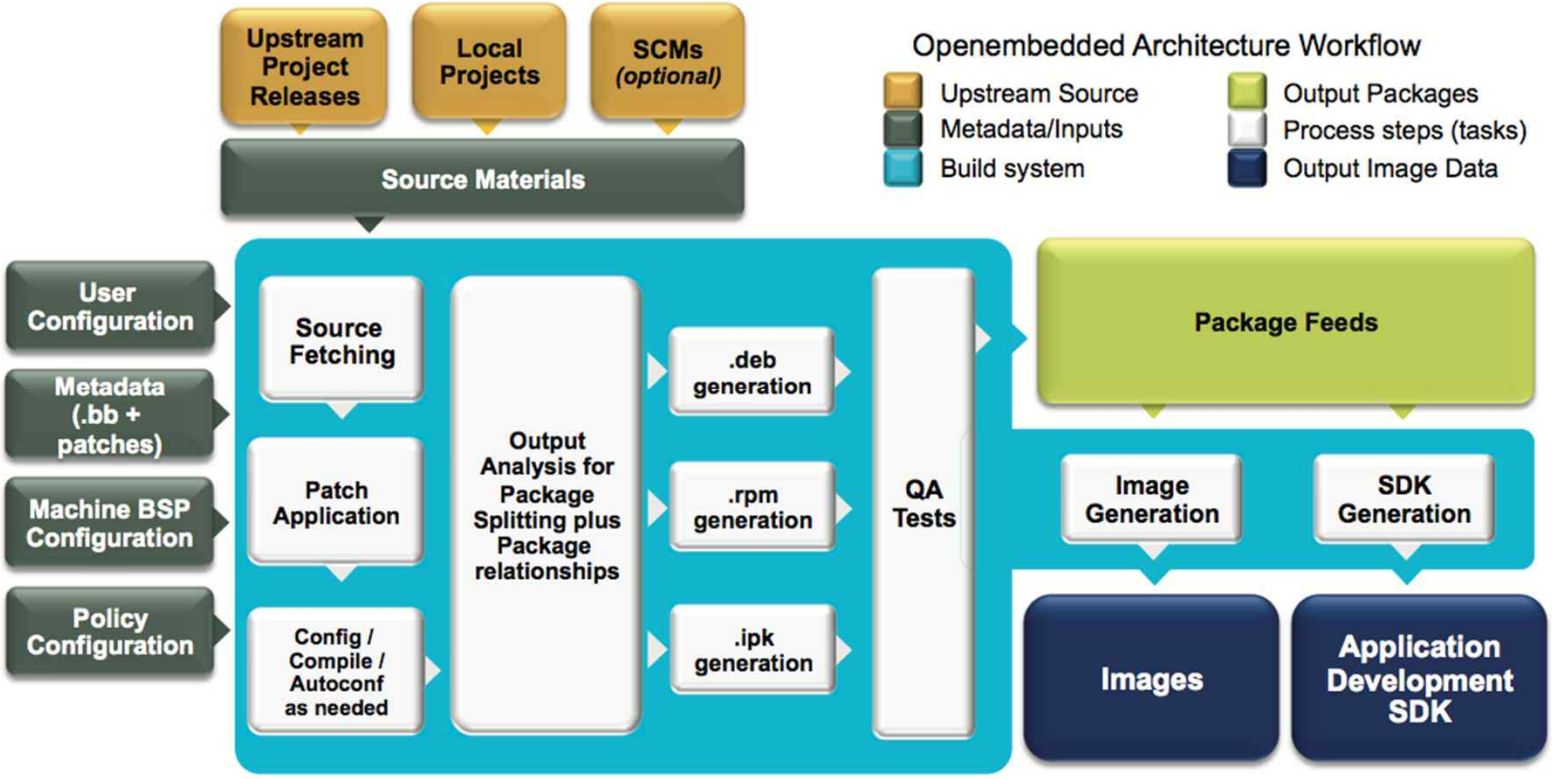
Poky = BitBake + Metadata

- Poky – build system used by the Yocto Project
- BitBake – a task executor and scheduler
- Metadata – task definitions
  - Configuration (.conf) – global definitions of variables
  - Classes (.bbclass) – encapsulation and inheritance of build logic, packaging, etc.
  - Recipes (.bb) – the logical units of software/images to build

Recipe

- A recipe is a set of instructions for building packages, including
  - Where to obtain the upstream sources and which patches to apply
  - Dependencies (on libraries or other recipes)
  - Configuration/compilation options
  - Define what files go into what output packages
  - License

Build System Workflow



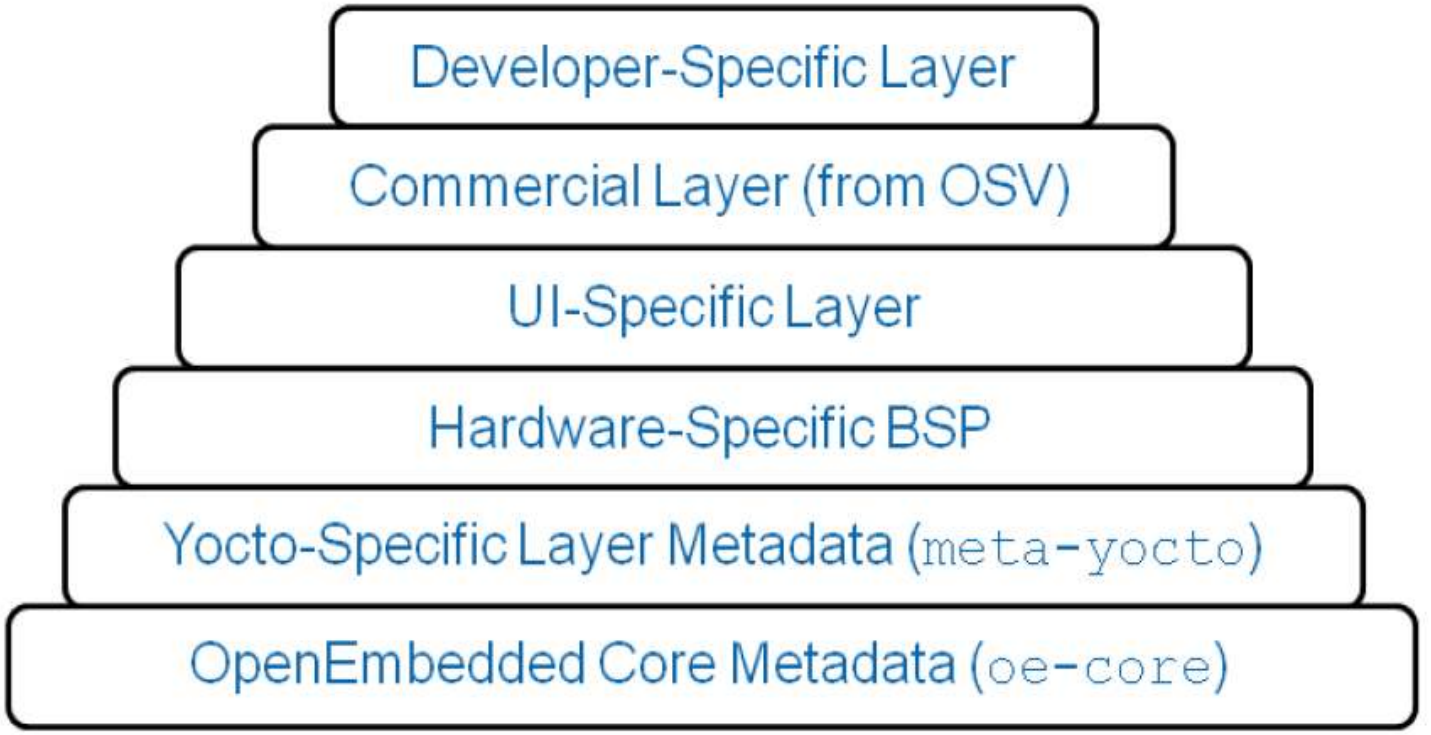
Standard Recipe Build Steps

- Building recipes involves executing the following functions, which can be overridden when needed for customizations
  - do_fetch
  - do_unpack
  - do_patch
  - do_configure
  - do_compile
  - do_install
  - do_package

Layers

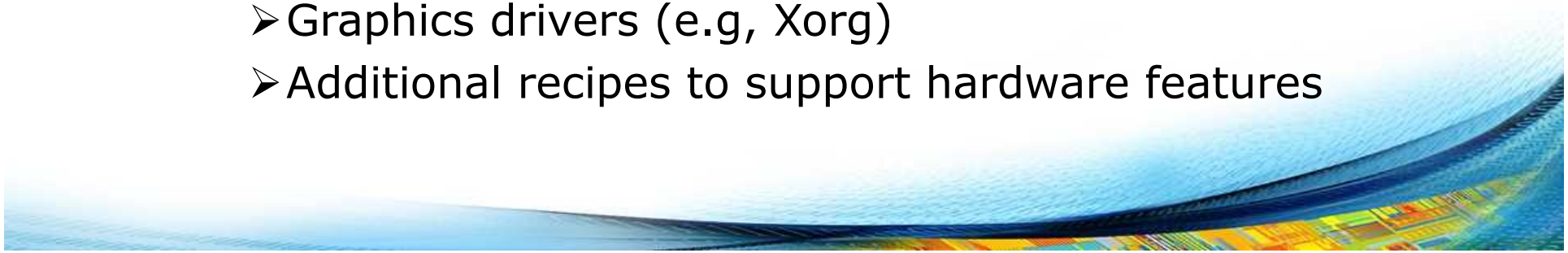
- The Yocto Project build system is composed of layers
- A layer is a logical collection of recipes representing the core, a Board Support Package(BSP), or an application stack

Structure of Yocto Project



Board Support Packages

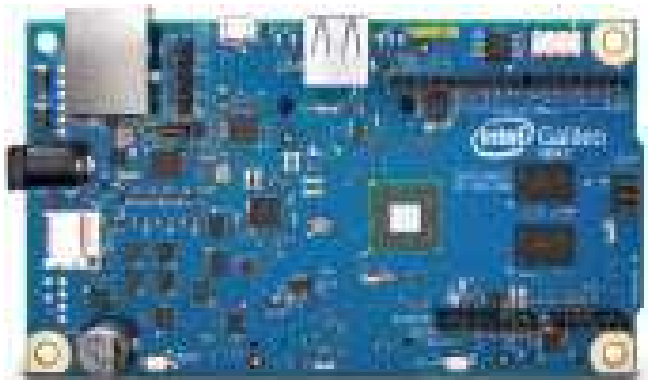
- BSPs are layers to enable support for specific hardware platforms
- Defines machine configuration for the “board”
- Adds machine-specific recipes and customizations
  - Kernel config
  - Graphics drivers (e.g, Xorg)
  - Additional recipes to support hardware features



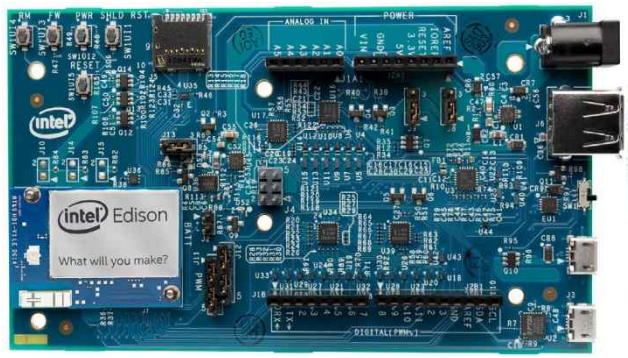
Yocto and OIC(Iotivity)

http://git.yoctoproject.org/cgit/cgit.cgi/meta-oic/about/

Yocto and Intel open-hardware Boards



Galileo



Edison

MinnowBoard MAX

Yocto supports Galileo and MinnowBoard Max

https://www.yoctoproject.org/blogs/jefro/2014/galileo-and-minnowboard-projects

Galileo and MinnowBoard Projects

Submitted by jefro on Thu, 2014-04-03 14:28

Many regular readers are familiar with the new Galileo board from Intel. This open-hardware board is based on the Quark SoC and provides Arduino compatibility with very high performance and great power management. (More information about the Galileo)

It may not be quite as commonly known that the Galileo is fully supported in the Yocto Project. The software downloads page for the Galileo contains drivers and images as well as links to the Galileo BSP and a full YP-based development environment. (Some community advice on using YP with Galileo)

This smaller board complements CircuitCo's Atom-based MinnowBoard project, which this week announced a new board called the MinnowBoard Max. The Minnows are also very well supported in YP.

Edison + Iotivity

https://wiki.iotivity.org/running_sample_codes_in_iotivity_0.9_sdk_on_edison

Yocto images

- For Intel Galileo

  https://software.intel.com/en-us/blogs/2015/03/04/creating-a-yocto-image-for-the-intel-galileo-board-using-split-layers

- For Intel Edison Board

  https://software.intel.com/en-us/iot/hardware/edison/downloads

- For MinnowBoard MAX

  http://www.elinux.org/Minnowboard:MinnowMaxYoctoProject

References

- Yocto Project Official Website
  - (https://www.yoctoproject.org/)
- Yocto Project Documentation for the Latest Release
  - (https://www.yoctoproject.org/documentation)
- Yocto Project Git Repository
  - (http://git.yoctoproject.org/cgit/cgit.cgi/)
- Getting Started with the Yocto Project - New Developer Screencast Tutorial
  - (https://www.yoctoproject.org/tools-resources/videos/getting-started-yocto-project-new-developer-screencast-tutorial)

Participate in Yocto Project

- Yocto Project is an open source project, and aims to deliver an open standard for the embedded Linux community and industry.
- Development is done in the public mailing lists:
  - openembedded-core@lists.openembedded.org
  - poky@yoctoproject.org
  - meta-intel@yoctoproject.org
  - linux-yocto@yoctoproject.org
- Bug reports and feature requests:
  - http://bugzilla.yoctoproject.org


yocto
PROJECT