TEXAS INSTRUMENTS

Chase Maupin,
Software Integration Manager

Henry Wiechman,
Software Marketing Manager

Denys Dmytriyenko,
Open Source Technologist

Texas Instruments

The Yocto Project: Changing the way embedded Linux software solutions are developed

Overview – The challenges

Enabling complex silicon devices such as systems-on-a-chip (SoC) with operating firmware and application software can be a challenge for equipment manufacturers who often are more comfortable with hardware than software issues. In many ways, the open-source movement improved the task of software development, but it also brought its own set of challenges. Fortunately, open-source embedded software development has continued to evolve, simplifying the development processes, enhancing the efficiency and allowing equipment manufacturers to more effectively differentiate the capabilities of their hardware. The Yocto Project™ is another phase in this evolution.

Previously, open-source embedded software developers faced a complex and often unstructured task. Before any applications could be run on new silicon, an operating system kernel and base file system were required, as well as a software shell, compiler, b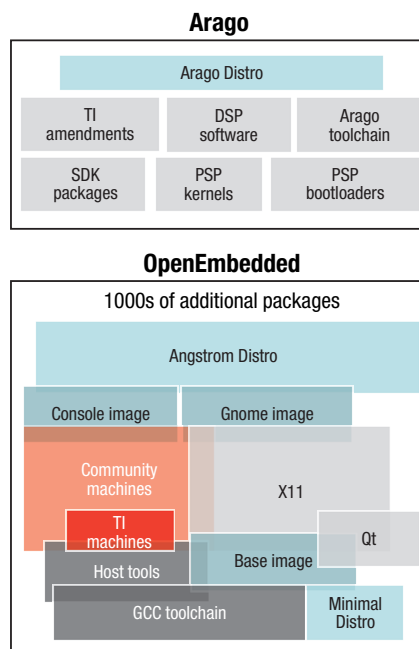ootloader and other components. Then packages of functional software targeted at the underlying hardware architecture had to be gathered, modified if necessary with open-source tools and integrated into a coherent Linux™ software distribution specific to the hardware platform.

Exacerbating the complexity of the entire process was the sometimes chaotic nature of the open-source world. A stable code foundation is critical for developing commercial software, but open-source code often fluctuates with the latest improvements and alterations, each of which can cause unexpected and unintended problems in extensive software systems. According to a 2012 annual survey of embedded developers conducted by Embedded Market Forecasters, two primary factors contributed to embedded developers' choice of Operating Systems (OS) in their designs: cost (44.6 percent) and the availability of source code (33.1 percent). These factors have contributed to the explosion of demand for Linux in embedded devices. Emerging from these conditions, the Yocto Project was founded to bring together hardware-agnostic tools, processes, templates and methods so that developers of embedded software could bring their products to market quickly with a competitive advantage through differentiated capabilities.

The Yocto Project and the OpenEmbedded Project

The Yocto Project (**www.yoctoproject.org**) is an open-source collaboration under the auspices of the Linux Foundation and composed of hardware manufacturers as well as open-source operating system suppliers. The project is made up of several sub-projects, including various tools, methods, meta-packages of executable code and, perhaps most importantly, a comprehensive framework for building embedded Linux software distributions.

Yocto Project grew out of the needs of commercial technology providers for greater stability from a maintained set of core packages and metadata, structured layering to provide code isolation, easy-to-use tooling, and clarity with regards to licensing issues and functional

Arago

Arago Distro		
TI amendments	DSP software	Arago toolchain
SDK packages	PSP kernels	PSP bootloaders

OpenEmbedded

1000s of additional packages

Angstrom Distro

Console image | Gnome image

Community machines | X11

TI machines | Qt

Host tools | Base image

GCC toolchain | Minimal Distro

Figure 1: What is OE-Classic?

expectations. Texas Instruments Incorporated (TI) and other companies joined the Yocto Project™ because it delivers these benefits and it gives their adopters the ability to focus more of their efforts and energy on advancing and differentiating their own products and systems by improving the productivity of open-source software development programs. Yocto Project has evolved in alignment with the OpenEmbedded (OE) Project, which was initiated to develop a framework for embedded Linux development, consisting of a file system, host tools and tool chain generation.

With deep roots in the open-source movement, the first instantiation of Open Embedded – OpenEmbedded-Classic (OE-Classic) began as a highly unstructured environment with relatively little hierarchical architecture to it. As OE-Classic grew with the addition of packages, recipes, metadata and other code, the flatness or egalitarian nature of the environment made it more difficult to use effectively and efficiently. Problems with the scalability and extensibility of OE-Classic became apparent as developers adopted it and added to it. For example, TI created its Arago overlay to help filter, organize, and customize the OE-Classic layer. However, this overlay was unique to TI-developed releases. With no lines of demarcation between the various logical segments of OE-Classic or between various vendor implementations, code alterations or new submissions that were meant to affect only a limited area of OE-Classic would often have unexpected ramifications on other segments. These and other factors contributed to the infrequency and instability of OE-Classic releases.

With this as a backdrop, the OE community and several technology providers, including TI, began seeking solutions that balanced the creativity of the open-source community with the objectives of commercial software development. The result was a new OE collaboration knows as OpenEmbedded Core (OE-Core). With the advent of OE-Core, TI has modified the Arago overlay into a compatible layer as supported by OE-Core, called meta-Arago, which provides the same functionality while being compatible with solutions from other vendors.
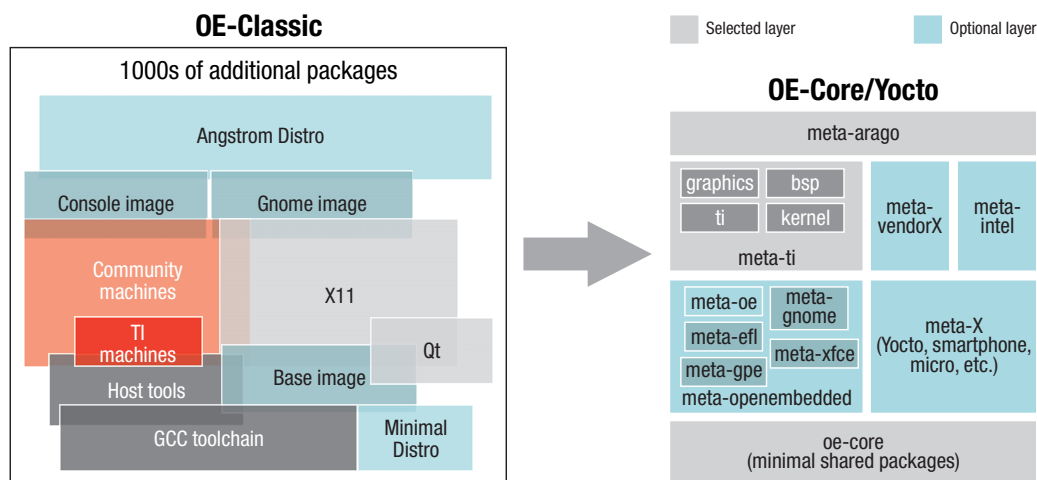
OE-Classic

Selected layer	Optional layer

OE-Core/Yocto

Figure 2: A layered approach

Growing numbers of adopters

The cleanliness of OE-Core and its alignment with Yocto Project™ as a Linux™ foundation project are convincing a significant number of developers to migrate in this direction. For example, layers and structure have been introduced to offer a degree of isolation among the various packages, metadata, recipes and tools that make up the environment. This separation minimizes unforeseen consequences when changes, additions or submissions occur. In particular, OE-Core with TI's meta-Arago overlay streamlines the development process on TI platforms so that the power of the underlying hardware can be more effectively applied. In addition, Yocto/OE-Core has incorporated capabilities that are important to commercial open-source developers, such as license verification. Previously, recipes might not list the license they operated under, but this has become a requirement of Yocto/OE-Core. This provides a level of confidence to developers since they can be assured that they know under which licenses the packages in their distributions will be operating. In addition, tools are available that will automatically monitor the license status of packages and recipes and filter out those that do not operate under the license specified by the developer. So, for example, the programming team might decide to avoid packages licensed under GPLv3. These tools would automatically exclude such packages from the distribution.

These benefits have attracted the support of many developers, as well as commercial technology providers like TI. As this support has continued to grow, the momentum generated for Yocto/OE-Core has been self-perpetuating. When new developers and commercial concerns become active members of the community, all members benefit because they are able to take advantage of the contributions and the creative solutions of thousands of developers. This, in turn, attracts more developers to the community.

Yocto technology

The Yocto Project has sought to strike a balance between the dynamism of the collaborative open-source community based on shared learning and epitomized by OE-Classic and one tempered by the practical

requirements of commercial software development, such as timely, cost-effective and stable releases of the environment. Ultimately, complete software systems can be delivered to the marketplace faster and more efficiently.

As a shell or framework, Yocto Project™ provides an environment within which Linux distributions tailored to any platform can be built quickly and effectively. One of Yocto Project's principal tools is BitBake, a build engine which merges recipes, packages and metadata into a new Linux distribution that can be optimized to the hardware, while remaining compatible with the rest of the open-source community. Yocto Project also has tools for generating a Software Development Kit (SDK) tailored to each distribution.

One of the important aspects of the Yocto Project is its hierarchical maintenance architecture. Because of the well-defined structure of Yocto Project, maintainers can be assigned to oversee the maintenance of a certain section or sub-section, which may be part of a much larger segment or layer of Yocto Project. This sort of personal accountability helps to ensure that Yocto/OE-Core will remain an efficient development environment. In addition, submissions to Yocto Project are reviewed for compliance to certain standards of quality. When code meets the project's compliance standards, it receives a compliance badge so that developers in the community can be assured of its authenticity and quality.

TI and Yocto Project

Because of the significant value that Yocto Project brings developers, TI has become a corporate supporter of the project. In particular, TI has committed to making ongoing and continuous contributions in terms of manpower and leadership. In fact, the company has designated members of its technical staff to be contributors to the Yocto Project. Denys Dmytriyenko, for example, is a meta-TI and meta-Arago maintainer, responsible for not only monitoring code submissions, but also functioning as a member of the Yocto/OE board of directors, attending conferences and overall ensuring that TI technology will function effectively within Yocto Project.

By aligning with Yocto/OE-Core, TI is leveraging the well-defined foundational layers of Yocto Project. Moreover, the latest TI SDKs are based on the OE-Core file system, which gives developers the ability to effectively tailor software distributions to the requirements of a particular application.

Already, TI has developed significant Yocto/OE-Core resources for developers. For its Sitara™ SoC platforms, TI has provided a Linux reference distribution utilizing its meta-Arago overlay to Yocto/OE-Core as well as a supporting SDK. With the Sitara Linux distribution, SDK developers are able to take full advantage of the computational power of all of the ARM® general-purpose cores in Sitara devices. The easy-to-use out-of-box experience of Arago lets developers begin working on TI platforms almost immediately. Along with its TI-specific modifications and packages, Arago contains components that would not be appropriate for Yocto/OE, but are useful to developers using TI SoCs. TI also provides a board support package (BSP) layer called meta-ti for TI devices, which supports any of several popular Linux distributions, including Angstrom.

Arago provides a host of useful tools and functions that simplify software development. For example, Arago filters and sets values to preferred versions of packages that are known to work, eliminating much of

the guess work about which package version to use. Another useful tool provided by TI as part of the Arago project is a script file that allows users to configure their environment and share that configuration with others. This means that once a configuration is generated for an SDK release from TI, it can be shared with developers using that software to set up their own system to duplicate and then customize the software build. In this way developers do not have to worry about the details of configuring their system and instead focus on what packages they want to build into their own distribution.

Developers can adopt Arago and efficiently differentiate their systems on top of it or they can modify Arago to meet their particular needs. In this way, the entire build process is simplified without sacrificing the benefits of individually optimized applications.

Conclusions

The pressure to bring high-quality software systems to market quickly has never been so great as it is today. With constant pressure on profit margins and rapidly closing windows of opportunity in the marketplace, equipment manufacturers invariably expect that software developers will be able to shorten their development cycles and, at the same time, provide differentiated capabilities that give their products a competitive advantage. To meet these pressures head-on, programmers require a robust and scalable open-source software foundation like Yocto/OE-Core. TI's significant contributions in the past indicate the firm commitment the company has made to leveraging the unique abilities of Yocto/OE-Core and continuing its effective evolution in the future.

For more information, go to **http://arago-project.org** and **www.ti.com/sitaralinux**.

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Products		Applications	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		