AMD XILINX

Embedded Design with PetaLinux Tools

EMBD-PLNX (v1.0)

Course Specification

Course Description

This course provides embedded systems developers experience with creating an embedded Linux system targeting AMD Xilinx SoCs using the PetaLinux tools.

The course provides experience with:

- Using open-source embedded Linux components
- Using the PetaLinux tool design flow
- Creating and debugging an application
- Building the environment and booting the system using the Arm® processors available in AMD Xilinx SoCs
- Customizing the root file system
- Configuring the Linux environment and network components
- Developing custom hardware and custom drivers

The primary focus is on embedded Linux development in conjunction with the AMD Xilinx tool flow.

What's New for 2022.1

- More information on Yocto and sysroot
- Yocto setting changes (board/board variants)
- Device tree binary (DTB) setting changes
- All labs have been updated to the latest software versions

Level – Embedded Software 4

Course Details

Course Duration – 16 hours – 2 days

Course Part Number – EMBD-PLNX

Who Should Attend? – Embedded software developers interested in customizing a kernel using PetaLinux on the Arm processors available in AMD Xilinx SoCs

Prerequisites

- *Designing FPGAs Using the Vivado Design Suite 1*
  - Introduction to FPGA Design
- *Designing FPGAs Using the Vivado Design Suite 2*
  - Designing with the IP Integrator
  - Creating and Packaging Custom IP
- *Embedded Systems Software Development*
  - Software development for embedded systems course

Software Tools

- PetaLinux Tools 2022.1
- Vivado® Design Suite 2022.1
- Vitis™ unified software platform 2022.1

Hardware

- Architecture: Zynq® UltraScale+™ MPSoC
- Demo board: Zynq UltraScale+ MPSoC ZCU104 board or Versal AI Core Series VCK190 Evaluation Kit

* This course focuses on the Zynq UltraScale+ MPSoC. Check with your local Authorized Training Provider for the specifics of the in-class lab board or other customizations.

After completing this comprehensive training, you will have the necessary skills to:

- Explain what an embedded Linux kernel is
- Create a PetaLinux project to configure and build an image
- Create a working Arm processor-based Linux system using the Vivado Design Suite and PetaLinux tools
- List various hardware interfacing options available for the Arm processor
- Describe the Linux device driver architecture
- Build custom hardware cores and device drivers using the user space I/O (UIO) framework

Course Outline

Day 1

- **Introduction to Embedded Linux**
  Introduces embedded Linux, including a brief architectural overview, as well as some of the reasons for its rising popularity as an embedded OS. Also introduces the concept of toolchains and cross-compilation. {Lecture}
- **Embedded Linux Components**
  Describes the various components required for embedded Linux platforms (including the kernel image, root file system, and boot loaders) and how the components affect the booting of Linux on these platforms. {Lecture, Lab}
- **Driving the PetaLinux Tool**
  Covers the functionality, inputs, and outputs of the PetaLinux tools as well as the project directory structure generated by the PetaLinux tools. Basic PetaLinux commands are also introduced. {Lecture, Lab}
- **Yocto Project Overview**
  Introduces the Yocto Project, which encompasses a build system designed for embedded Linux. {Lecture}
- **PetaLinux Tool Design Flow**
  Provides a brief description of the PetaLinux tool design flow and describes in detail various PetaLinux commands (including petalinux-create, petalinux-config, petalinux-build, petalinux-package, and petalinux-boot) and their example use cases. {Lecture}
- **PetaLinux Application Development**
  Introduces core concepts for developing, customizing, and running software applications in an embedded Linux environment. {Lecture, Lab}
- **Customizing the Project**
  Analyzes different configuration options provided by the PetaLinux tool for firmware version, rootfs type, boot image storage, and primary flash partition. Also describes external file system boot configuration. {Lecture}
- **Customizing the Root File System**
  Provides a brief description on customizing the rootfs for embedded Linux components such as libraries, applications, modules, layers, recipes, and packages. {Lecture}
- **Networking and TCP/IP**
  Discusses how the TCP/IP networking stack can be used to improve productivity during embedded product development by supporting network data communication, network control/status management, and firmware and hardware upgrades. {Lecture, Lab}

- **PetaLinux Booting and Packaging**

  Describes how to package and then boot a PetaLinux image via QEMU, SD card, JTAG, and TFTP. {Lecture}

Day 2

- **PetaLinux Application Debugging**

  Describes how to debug software applications running on an Arm processor using the system debugger (TCF agent) or GNU debugger (GDB). {Lecture, Lab}
- **Upgrading the Workspace**

  Describes the petalinux-upgrade command and how to upgrade PetaLinux project software components without changing the host tool components. {Lecture}
- **Basic Hardware Design Process with the Vivado Design Suite**

  Describes the complete board bring-up process, which includes the hardware design as well as Linux image creation for the hardware. {Lecture, Lab}
- **Linux Device Drivers Overview**

  Provides a brief overview on Linux device drivers and their requirements. Also describes what a device tree is and how it is generated. {Lecture}
- **User Space I/O and Loadable Kernel Modules**

  Introduces two lightweight approaches for accessing the physical memory of devices from user space: direct access through the `dev/mem` virtual device and the user space I/O framework. Also covers the role and usage loadable kernel modules. {Lecture, Lab}
- **Custom Hardware Development**

  Describes the Create and Package IP Wizard and how it can be used to create a variety of architectural options for interfacing a system with custom processing hardware. {Lecture, Lab}
- **Custom Driver Development**

  Discusses device driver options to match custom hardware devices and how to use the provided interfaces to read and write to the devices. {Lecture, Lab}
- **PetaLinux: Advanced Configurations**

  Reviews how modify advanced configuration settings using the PetaLinux tool. These configurations include including selecting the Linux components for the build, enabling automatic configuration for a selected component, customizing how the Linux system interacts with the underlying hardware platform. {Lecture}