oY2 - Yocto Project Expert

Advanced Yocto Project usage and adaptation

Objectives

- Using Yocto to develop components
- Customizing the BSP
- Building out of tree modules
- Setup Source cache

Labs are conducted QEMU ARM-based board
We use a recent version of Yocto

Prerequisite

- Good C programming skills (see our oL2 - C Language for Embedded MCUscourse)
- Knowledge of Linux Embedded systems (see our oD1 - Embedded Linuxcourse)
- Knowledge of Yocto Project Development (see our oY1 - Yocto Project Developmentcourse)
- Preferably knowledge of Linux user programming (see our oD0 - Linux User Mode Programmingcourse)

Course environment

- Theoretical course:
  - PDF course material (in English)
  - Course dispensed using the Teams video-conferencing system
  - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system
  - Quizzes are used to check trainee's assimilation of course content
- Practical activities :
  - Practical activities represent from 40% to 50% of course duration
  - One Online Linux PC per trainee for the practical activities
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
- Downloadable preconfigured virtual machine for post-course practical activities.

Duration

- Total: 12 hours
- 2 sessions, 6 hours +/- 30 min each (excluding break time)
- From 50% to 60% of training time is devoted to practical activities
- Some Labs may be completed between sessions and are checked by the trainer on the next session

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Session

Development process using the extensible SDK and devtool

- Using devtool to create a package and its recipe
- Using devtool to modify an existing package and recipe
- Using devtool to update a recipe to build a new version of a package

Exercise: Create, test and modify a recipe for an existing package using devtool

Develop and debug applications using SDK and eclipse

- Adding eclipse remote debug packages
- Configuring eclipse

Exercise: Create remote debugging session using eclipse

Writing tasks in python

- Introduction to python
- Using python in Yocto
  - The main bitbake classes
  - Defining variable values in Python
  - Writing tasks in Python

Exercise: Writing a task and customizing a recipe in Python

Porting Yocto

- Porting Yocto to a new board
- BSP architecture
  - Selecting and configuring u-boot recipe
  - Selecting and configuring kernel recipe
- Adding a new BSP layer (yocto-bsp create)

Exercise: Creating a new BSP layer

Second Session

BSP Development

- Adding a custom u-boot to Yocto
- Customizing the Yocto kernel recipe
  - Setting the default configuration
  - Adding patches
  - Specifying the kernel sources
- Configuring Linux Kernel
  - Using menuconfig
  - Using patches
  - Creating Configuration Fragments
  - Validating Configuration
- Kernel device tree

Exercise: Create u-boot and kernel recipes to use custom versions, test the result
Exercise: Patch kernel and activate new options using a fragment
Exercise: Create and use a new device tree

Out-of-Tree Modules

- Adding modules to image
- Creating an out-of-tree module
- Kernel modules with eSDK

Exercise: Build and test modules

Tailoring the build system

- Setting up a Yocto source cache
  - Local, per system, cache setup
  - Setting up a global, network wide, cache
- Customizing the build system
  - Using a prebuilt toolchain
  - Using a pre-compiled kernel
- Optimizing Yocto build times
  - Using prebuilt, binary, packages
  - Using shared compilation caches

Exercise: Setting up a global source cache
Exercise: Setting up an optimized build environment and rebuilding an image