



# Yocto Project and OpenEmbedded training

## On-line seminar

<b>Title</b>	<b>Yocto Project and OpenEmbedded development training</b>
<b>Overview</b>	<p>Understanding the Yocto Project          Using it to build a root filesystem and run it on your target          Writing and extending recipes          Creating layers          Integrating your board in a BSP          Creating custom images          Application development with the Yocto Project SDK</p>
<b>Materials</b>	<p>Check that the course contents correspond to your needs:  <a href="https://bootlin.com/doc/training/yocto">https://bootlin.com/doc/training/yocto</a>.</p>
<b>Duration</b>	<p><b>Four</b> half days - 16 hours (4 hours per half day).          80% of lectures, 20% of practical demos.</p>
<b>Trainer</b>	<p>One of the engineers listed on          80% of lectures, 20% of practical demos.</p>
<b>Language</b>	<p>Oral lectures: English          Materials: English.</p>
<b>Audience</b>	<p>Companies and engineers interested in using the Yocto Project to build their embedded Linux system.</p>
<b>Prerequisites</b>	<p><b>Familiarity with embedded Linux</b> as covered in our embedded Linux training (<a href="https://bootlin.com/training/embedded-linux/">https://bootlin.com/training/embedded-linux/</a>)</p> <p><b>Familiarity with UNIX or GNU/Linux commands</b>          People lacking experience on this topic may get trained by themselves, for example with our freely available on-line slides: <a href="https://bootlin.com/blog/command-line/">https://bootlin.com/blog/command-line/</a></p>
<b>Required equipment</b>	<ul style="list-style-type: none"> <li>• Computer with the operating system of your choice, with the Google Chrome or Chromium browser for videoconferencing.</li> <li>• Webcam and microphone (preferably from an audio headset)</li> <li>• High speed access to the Internet</li> </ul>
<b>Materials</b>	<p>Electronic copies of presentations, demo instructions and data.</p>



## Hardware

STMicroelectronics STM32MP157A-DK1 Discovery board

- STM32MP157A (dual Cortex-A7) CPU from STMicroelectronics
- USB powered
- 512 MB DDR3L RAM
- Gigabit Ethernet port
- 4 USB 2.0 host ports
- 1 USB-C OTG port
- 1 Micro SD slot
- On-board ST-LINK/V2-1 debugger
- Arduino Uno v3-compatible headers
- Audio codec
- Misc: buttons, LEDs



## Half day 1

### Lecture - Introduction to embedded Linux build systems

- Overview of an embedded Linux system architecture
- Methods to build a root filesystem image
- Usefulness of build systems

### Lecture - Overview of the Yocto Project and the Poky reference system

- Organization of the project source tree
- Building a root filesystem image using the Yocto Project

### Demo - First Yocto Project build

- Downloading the Poky reference build system
- Building a system image



### Lecture - Using Yocto Project - basics

- Organization of the build output
- Flashing and installing the system image

### Demo - Flashing and booting

- Flashing and booting the image on the board

## Half day 2

---

### Lecture - Using Yocto Project - advanced usage

- Configuring the build system
- Customizing the package selection

### Demo - Using NFS and configuring the build

- Configuring the board to boot over NFS
- Learn how to use the `PREFERRED_PROVIDER` mechanism

### Lecture - Writing recipes - basics

- Writing a minimal recipe
- Adding dependencies
- Development workflow with *bitbake*

### Demo - Adding an application to the build

- Writing a recipe for *nInvaders*
- Adding *nInvaders* to the final image



## Lecture - Writing recipes - advanced features

- Extending and overriding recipes
- Adding steps to the build process
- Learn about classes
- Analysis of examples
- Logging
- Debugging dependencies

## Half day 3

---

### Demo - Learning how to configure packages

- Extending a recipe to add configuration files
- Using `ROOTFS_POSTPROCESS_COMMAND` to modify the final rootfs
- Studying package dependencies

### Lecture - Layers

- What layers are
- Where to find layers
- Creating a layer

### Demo - Writing a layer

- Learn how to write a layer
- Add the layer to the build
- Move *nInvaders* to the new layer

### Lecture - Writing a BSP

- Extending an existing BSP
- Adding a new machine
- Bootloaders
- Linux and the linux-yocto recipe
- Adding a custom image type

### Demo - Implementing the kernel changes

- Extend the kernel recipe to add the nunchuk driver
- Configure the kernel to compile the nunchuk driver
- Play *nInvaders*



## Half day 4

### Lecture - Creating a custom image

- Writing an image recipe
- Adding users/groups
- Adding custom configuration
- Writing and using package groups recipes

### Demo - Creating a custom image

- Writing a custom image recipe
- Adding *nInvaders* to the custom image

### Lecture - Creating and using an SDK

- Understanding the purpose of an SDK for the application developer
- Building an SDK for the custom image

### Demo - Experimenting with the SDK

- Building an SDK
- Using the Yocto Project SDK

### Questions and Answers

- Questions and answers with the audience about the course topics
- Extra presentations if time is left, according what most participants are interested in.