

Embedded Linux Development Using Yocto Project Cookbook *Second Edition*

Practical recipes to help you leverage the power of Yocto to
build exciting Linux-based systems

Alex González

Packt>

BIRMINGHAM - MUMBAI

Embedded Linux Development Using Yocto Project Cookbook

Second Edition

Copyright © 2018 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Gebin George
Acquisition Editor: Prachi Bisht
Content Development Editor: Dattatraya More
Technical Editor: Jovita Alva
Copy Editor: Safis Editing
Project Coordinator: Shweta H Birwatkar
Proofreader: Safis Editing
Indexer: Francy Puthiry
Graphics: Tania Dutta
Production Coordinator: Arvindkumar Gupta

First published: March 2015
Second edition: January 2018

Production reference: 1240118

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-78839-921-0

www.packtpub.com



GET

80%

OFF ANY PACKT
EBOOK AND VIDEO



Go to www.packtpub.com
and use this code in the checkout:

HBIOT80OFF

Packt

I dedicate this second edition to the loving memory of my mum whose example makes me constantly challenge myself.



`mapt.io`

Mapt is an online digital library that gives you full access to over 5,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Mapt is fully searchable
- Copy and paste, print, and bookmark content

PacktPub.com

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Foreword

Adoption of Linux continues to grow year by year, with the majority of growth in the area of embedded systems. While it is possible to build an embedded Linux operating system from scratch, it is hard work and error prone. Thankfully, there is a better way—using the BitBake build tool, the OpenEmbedded Core metadata and the Poky distribution, which, together, make up the Yocto Project.

Since its inception in 2010, the Yocto Project has progressed to become the de facto build system for a wide range of appliances and devices running Linux. Now, with the advent of the connected world known as the Internet of Things, the Yocto Project is taking on a key role in creating the backbone operating systems for devices that we rely on every day. So, the Yocto Project does not stand still, it has to continually evolve to cater for current generations of hardware and to support the tools and applications used in modern connected devices.

However, as anyone who has dipped a toe into the Yocto Project pool will know, the water gets deep very quickly. The flexibility of the Yocto Project means that you need to be aware of many concepts if you are to make best use of the tool. I heartily recommend this book as your lifesaver! Alex's thorough understanding of the topic, coupled with a practical approach to problem solving, makes this an easy-to-read, essential companion that will help you not only to keep your head above water but to become a proficient swimmer.

This second edition of *Embedded Linux Development Using Yocto Project Cookbook* follows in the same style and technical content as the first, but has been refreshed and extended to describe the current versions of the Yocto Project. Alex's deep understanding of the Yocto Project conjoins with his practical knowledge of the subject to produce a practical guide to the Yocto Project. The cookbook style allows you to dip in and out as needed to find answers to particular problems. Alex always adds pointers to more detailed descriptions of the problems covered, so that, step by step, you can build up a thorough understanding of the underlying principles. As an educator, I am always looking for books to recommend to my students. This is at the top of my list when teaching people about the Yocto Project.

Chris Simmonds

Founder of 2net.co.uk, author and teacher

Contributors

About the author

Alex González is a software engineering supervisor at Digi International and product owner of the Digi Embedded Yocto distribution. He started working professionally with embedded systems in 1999 and the Linux kernel in 2004, designing products for voice and video over IP networks, and followed his interests into machine-to-machine (M2M) technologies and the Internet of Things. Born and raised in Bilbao, Spain, Alex has an electronic engineering degree from the University of the Basque Country and he received his MSc in communication systems from the University of Portsmouth.

I would like to thank the Yocto and OpenEmbedded communities, whose dedication keep the Yocto Project running, and the people involved with the Freescale BSP community layer, whose joint work is the basis for this book.

Also, the amazing people who work with me on a daily basis and from whom I am constantly learning, and especially to Javier Viguera for his efforts in thoroughly reviewing the contents of the book.

About the reviewer

Javier Viguera has been a Linux fan since the mid-1990s, when he managed to install Slackware at home to avoid fighting for a two hour slot in the university's computer lab.

With a master's degree in telecommunications engineering and a bachelor's degree in computer science, he is currently working at Digi International as an embedded software engineer.

Living in La Rioja, Spain, in his spare time Javier likes to see good classical movies. He is also a fan of airplanes and aviation.

I would like to thank the author Alex, for this opportunity. Also, and especially, my wife and daughter for supporting me during the review.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit authors.packtpub.com and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

| | |
|--|----|
| Preface | 1 |
| Chapter 1: The Build System | 6 |
| Introduction | 7 |
| Setting up the host system | 8 |
| Getting ready | 8 |
| How to do it... | 9 |
| How it works... | 10 |
| There's more... | 10 |
| See also | 11 |
| Installing Poky | 11 |
| Getting ready | 11 |
| How to do it... | 12 |
| How it works... | 13 |
| There's more... | 13 |
| See also | 14 |
| Creating a build directory | 14 |
| How to do it... | 14 |
| How it works... | 15 |
| There's more... | 16 |
| Building your first image | 16 |
| Getting ready | 16 |
| How to do it... | 17 |
| How it works... | 18 |
| There's more... | 20 |
| Explaining the NXP Yocto ecosystem | 21 |
| Getting ready | 21 |
| How to do it... | 21 |
| How it works... | 22 |
| There's more... | 23 |
| See also | 26 |
| Installing support for NXP hardware | 26 |
| Getting ready | 26 |
| How to do it... | 26 |
| How it works... | 27 |

| | |
|--|----|
| There's more... | 28 |
| See also | 29 |
| Building Wandboard images | 29 |
| How to do it... | 30 |
| How it works... | 30 |
| See also | 32 |
| Using the Toaster web interface | 33 |
| Getting ready | 34 |
| How to do it... | 34 |
| Administrator interface | 34 |
| Starting a build | 35 |
| Customizing images with Toaster | 36 |
| Building SDKs with Toaster | 37 |
| How it works... | 37 |
| There's more... | 39 |
| Running a Toaster Docker container | 39 |
| How to do it... | 40 |
| See also | 42 |
| Configuring network booting for a development setup | 42 |
| Getting ready | 42 |
| Installing a TFTP server | 43 |
| Installing an NFS server | 44 |
| How to do it... | 45 |
| Using Docker as a Yocto build system container | 47 |
| Getting ready | 48 |
| How to do it... | 49 |
| How it works... | 50 |
| See also | 51 |
| Sharing downloads | 51 |
| Getting ready | 52 |
| How to do it... | 53 |
| Sharing the shared state cache | 54 |
| How to do it... | 55 |
| There's more... | 55 |
| Setting up a package feed | 56 |
| Getting ready | 56 |
| Versioning packages | 57 |
| How to do it... | 58 |
| There's more... | 61 |
| Generating signed packages | 61 |

| | |
|---|----|
| Using signed package feeds | 62 |
| Creating a GNUPG key pair | 62 |
| Backing up your keys | 63 |
| See also | 63 |
| Using build history | 64 |
| How to do it... | 64 |
| How it works... | 65 |
| Looking at build history | 66 |
| There's more... | 67 |
| Working with build statistics | 68 |
| How to do it... | 69 |
| How it works... | 69 |
| There's more... | 72 |
| See also | 73 |
| Debugging the build system | 74 |
| Getting ready | 74 |
| Finding recipes | 74 |
| Dumping BitBake's environment | 75 |
| Using the development shell | 76 |
| How to do it... | 77 |
| Task log and run files | 77 |
| Adding logging to recipes | 78 |
| Looking at dependencies | 79 |
| Debugging dependencies | 82 |
| Debugging BitBake | 82 |
| Error reporting tool | 83 |
| Chapter 2: The BSP Layer | 85 |
| Introduction | 85 |
| Creating a custom BSP layer | 86 |
| How to do it... | 86 |
| How it works... | 87 |
| There's more... | 88 |
| Adding a new machine | 89 |
| Adding a custom device tree to the Linux kernel | 93 |
| Adding a custom U-Boot machine | 94 |
| Adding a custom formfactor file | 95 |
| Build your custom machine | 96 |
| See also | 96 |
| Adding a custom kernel and bootloader | 96 |
| Getting ready | 97 |
| Finding the Linux kernel source | 97 |
| Finding the U-Boot source | 98 |

| | |
|--|-----|
| Developing using a Git repository fork | 99 |
| How to do it... | 99 |
| How it works... | 100 |
| Building the U-Boot bootloader | 102 |
| Getting ready | 102 |
| How to do it... | 103 |
| There's more... | 104 |
| Describing Linux's build system | 107 |
| How to do it... | 107 |
| How it works... | 108 |
| There's more... | 108 |
| Configuring the Linux kernel | 109 |
| Getting ready | 109 |
| How to do it... | 110 |
| Using Yocto to configure the Linux kernel | 112 |
| There's more... | 113 |
| Building the Linux kernel | 114 |
| How to do it... | 114 |
| Using Yocto to build the Linux kernel | 116 |
| There's more... | 117 |
| Building external kernel modules | 118 |
| Getting ready | 119 |
| How to do it... | 121 |
| There's more... | 122 |
| Module auto-loading | 123 |
| See also | 123 |
| Debugging the Linux kernel and modules | 123 |
| How to do it... | 123 |
| How it works... | 124 |
| There's more... | 126 |
| Using dynamic debug | 126 |
| Rate-limiting debug messages | 128 |
| See also | 128 |
| Debugging the Linux kernel booting process | 128 |
| How to do it... | 129 |
| How it works... | 129 |
| Dumping the kernel's printk buffer from the bootloader | 129 |
| There's more... | 131 |
| Using the kernel function tracing system | 132 |
| Getting ready | 133 |

| | |
|--|-----|
| How to do it... | 133 |
| How it works... | 134 |
| There's more... | 135 |
| Filtering function traces | 136 |
| Enabling trace options | 137 |
| Using the function tracer on oops | 137 |
| Getting a stack trace for a given function | 138 |
| Configuring the function tracer at boot | 138 |
| See also | 138 |
| Managing the device tree | 138 |
| Getting ready | 139 |
| How to do it... | 139 |
| How it works... | 140 |
| The compatible property | 142 |
| The Wandboard device tree file | 142 |
| Defining buses and memory-addressable devices | 144 |
| There's more... | 145 |
| Modifying and compiling the device tree in Yocto | 145 |
| See also | 146 |
| Debugging device tree issues | 146 |
| How to do it... | 146 |
| How it works... | 146 |
| Looking at the device tree from U-Boot | 146 |
| Looking at the device tree from the Linux kernel | 148 |
| Chapter 3: The Software Layer | 149 |
| Introduction | 150 |
| Exploring an image's contents | 150 |
| Getting ready | 150 |
| How to do it... | 151 |
| How it works... | 152 |
| Adding a new software layer | 152 |
| Getting ready | 153 |
| How to do it... | 153 |
| How it works... | 154 |
| There's more... | 155 |
| See also | 158 |
| Selecting a specific package version and provider | 158 |
| How to do it... | 159 |
| How do we select which provider to use? | 159 |
| How do we select which version to use? | 160 |
| How do we select which version not to use? | 160 |

| | |
|---|-----|
| Adding supported packages | 160 |
| Getting ready | 161 |
| How to do it... | 161 |
| How it works... | 162 |
| There's more... | 164 |
| Configuring packages | 164 |
| Adding new packages | 165 |
| Getting ready | 165 |
| How to do it... | 166 |
| How it works... | 167 |
| Package licensing | 167 |
| Fetching package contents | 168 |
| Specifying task overrides | 169 |
| Configuring packages | 171 |
| Splitting into several packages | 172 |
| Setting machine-specific variables | 172 |
| Adding data, scripts, or configuration files | 173 |
| How to do it... | 173 |
| There's more... | 174 |
| Managing users and groups | 175 |
| Getting ready | 175 |
| How to do it... | 176 |
| There's more... | 177 |
| Using the sysvinit initialization manager | 178 |
| Getting ready | 179 |
| How to do it... | 180 |
| Using the systemd initialization manager | 181 |
| Getting ready | 182 |
| How to do it... | 183 |
| There's more... | 184 |
| Installing systemd unit files | 185 |
| See also | 187 |
| Installing package installation scripts | 187 |
| Getting ready | 187 |
| How to do it... | 187 |
| How it works... | 188 |
| Reducing the Linux kernel image size | 189 |
| How to do it... | 189 |
| How it works... | 190 |
| Reducing the root filesystem size | 194 |

| | |
|--|-----|
| How to do it... | 194 |
| How it works... | 196 |
| Memory-based root filesystems | 198 |
| Getting ready | 199 |
| How to do it... | 199 |
| How it works... | 200 |
| There's more... | 201 |
| Securing the root filesystem | 202 |
| Getting ready | 202 |
| How to do it... | 203 |
| Setting up SELinux | 204 |
| Setting up SMACK | 207 |
| How it works... | 207 |
| Looking into SELinux | 208 |
| Looking into SMACK | 209 |
| See also | 211 |
| Releasing software | 211 |
| Getting ready | 212 |
| How to do it... | 213 |
| There's more... | 215 |
| See also | 215 |
| Analyzing your system for compliance | 215 |
| How to do it... | 215 |
| There's more... | 216 |
| Working with open source and proprietary code | 217 |
| How to do it... | 217 |
| How it works... | 218 |
| The U-Boot bootloader | 218 |
| The Linux kernel | 218 |
| glibc | 218 |
| musl | 218 |
| BusyBox | 219 |
| The Qt framework | 219 |
| The X Windows system | 220 |
| There's more... | 220 |
| See also | 221 |
| Chapter 4: Application Development | 222 |
| Introduction | 223 |
| Introducing toolchains | 223 |
| Getting ready | 224 |

| | |
|---|-----|
| How to do it... | 225 |
| Preparing an SDK | 226 |
| Getting ready | 227 |
| How to do it... | 227 |
| How it works... | 230 |
| Customizing standard and extensible SDKs | 230 |
| Adding packages to the SDKs | 231 |
| Using the extensible SDK | 232 |
| Getting ready | 232 |
| Exploring the extensible SDK contents | 235 |
| How to do it... | 236 |
| How it works... | 241 |
| The workspace layer | 241 |
| Recipe development workflow | 242 |
| Adding a recipe to the workspace layer | 242 |
| Removing a recipe from the workspace layer | 243 |
| Modifying a recipe in the workspace layer | 244 |
| Building and testing your recipe | 244 |
| Updating a recipe with your changes | 245 |
| Finishing work on a recipe | 245 |
| There's more... | 245 |
| Using the Eclipse IDE | 246 |
| Getting ready | 246 |
| How to do it... | 248 |
| There's more... | 252 |
| See also | 253 |
| Developing GTK+ applications | 253 |
| Getting ready | 253 |
| How to do it... | 255 |
| There's more... | 264 |
| Using the Qt Creator IDE | 266 |
| Getting ready | 267 |
| How to do it... | 268 |
| Developing Qt applications | 274 |
| Getting ready | 274 |
| How to do it... | 276 |
| There's more... | 282 |
| Describing workflows for application development | 283 |
| How to do it... | 283 |
| How it works... | 283 |
| External development | 283 |

| | |
|--|-----|
| Working directory development | 284 |
| External source development | 287 |
| Working with GNU make | 288 |
| How to do it... | 289 |
| See also | 290 |
| Working with the GNU build system | 290 |
| Getting ready | 291 |
| How to do it... | 292 |
| See also | 292 |
| Working with the CMake build system | 293 |
| Getting ready | 293 |
| How to do it... | 293 |
| See also | 294 |
| Working with the SCons builder | 294 |
| Getting ready | 294 |
| How to do it... | 295 |
| See also | 295 |
| Developing with libraries | 296 |
| Getting ready | 296 |
| Building a static library | 297 |
| Building a shared dynamic library | 299 |
| How to do it... | 299 |
| How it works... | 302 |
| There's more... | 303 |
| See also | 303 |
| Working with the Linux framebuffer | 303 |
| Getting ready | 303 |
| How to do it... | 304 |
| How it works... | 306 |
| There's more... | 307 |
| See also | 311 |
| Using the X Windows system | 311 |
| Getting ready | 311 |
| How to do it... | 312 |
| There's more... | 314 |
| See also | 314 |
| Using Wayland | 315 |
| Getting ready | 315 |
| How to do it... | 316 |
| There's more... | 317 |

| | |
|---|-----|
| See also | 317 |
| Adding a web browser application | 318 |
| Getting ready | 318 |
| How to do it... | 318 |
| There's more... | 320 |
| See also | 320 |
| Adding Python applications | 320 |
| Getting ready | 321 |
| How to do it... | 322 |
| There's more... | 324 |
| Integrating the Open Java Development Kit | 325 |
| Getting ready | 325 |
| How to do it... | 325 |
| How it works... | 326 |
| There's more... | 327 |
| See also | 327 |
| Integrating Java applications | 327 |
| Getting ready | 328 |
| How to do it... | 328 |
| How it works... | 329 |
| There's more... | 330 |
| See also | 330 |
| Integrating Node.js applications | 331 |
| Getting ready | 331 |
| How to do it... | 333 |
| How it works... | 333 |
| There's more... | 336 |
| See also | 337 |
| Running Docker application containers | 337 |
| Getting ready | 338 |
| How to do it... | 338 |
| How it works... | 340 |
| See also | 341 |
| Chapter 5: Debugging, Tracing, and Profiling | 342 |
| Introduction | 343 |
| Analyzing core dumps | 343 |
| Getting ready | 344 |
| How to do it... | 344 |

| | |
|--|-----|
| How it works... | 346 |
| See also | 349 |
| Native GDB debugging | 349 |
| Getting ready | 349 |
| How to do it... | 350 |
| There's more... | 352 |
| Cross GDB debugging | 353 |
| Getting ready | 353 |
| How to do it... | 354 |
| There's more... | 356 |
| Using strace for application debugging | 357 |
| Getting ready | 357 |
| How to do it... | 357 |
| How it works... | 358 |
| See also | 358 |
| Using the kernel's performance counters | 359 |
| Getting ready | 359 |
| How to do it... | 360 |
| How it works... | 361 |
| There's more... | 363 |
| See also | 363 |
| Using static kernel tracing | 363 |
| Getting ready | 364 |
| How to do it... | 364 |
| How it works... | 367 |
| There's more... | 370 |
| See also | 371 |
| Using dynamic kernel tracing | 371 |
| Getting ready | 372 |
| How to do it... | 373 |
| There's more... | 376 |
| See also | 379 |
| Using dynamic kernel events | 379 |
| Getting ready | 379 |
| How to do it... | 380 |
| How it works... | 382 |
| There's more... | 383 |
| Exploring Yocto's tracing and profiling tools | 384 |
| Getting ready | 384 |

| | |
|--|-----|
| How to do it... | 385 |
| There's more... | 385 |
| Tracing and profiling with perf | 386 |
| Getting ready | 386 |
| How to do it... | 388 |
| How it works... | 394 |
| Reading tracing data | 395 |
| There's more... | 398 |
| Profile charts | 400 |
| Using perf as a strace substitute | 400 |
| See also | 400 |
| Using SystemTap | 401 |
| Getting ready | 401 |
| How to do it... | 402 |
| How it works... | 403 |
| See also | 404 |
| Using LTTng | 404 |
| Getting ready | 404 |
| How to do it... | 405 |
| How it works... | 408 |
| Extending application profiling | 409 |
| There's more... | 411 |
| See also | 412 |
| Using blktrace | 412 |
| Getting ready | 412 |
| How to do it... | 413 |
| How it works... | 415 |
| There's more... | 418 |
| Appendix A: Other Books You May Enjoy | 419 |
| Leave a review - let other readers know what you think | 421 |
| Index | 422 |
