Run old games on old hardware and learn new things!

Yocto Project Workshop at Embedded Recipes 2025

Michael Opdenacker

Root Commit

May 16, 2025

Michael Opdenacker

Embedded Linux consultant and trainer

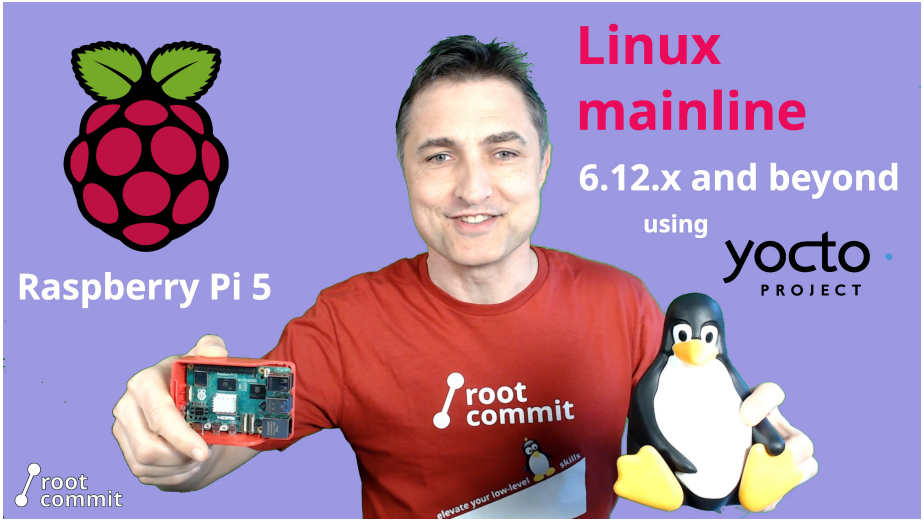
- https://rootcommit.com/about/michael-opdenacker/
- Former founder of Bootlin
- New founder of Root Commit
- Offering embedded Linux training courses with a focus on practical activities, interactivity and learning techniques. https://rootcommit.com/training/
- Sharing training materials under CC-BY-SA license
- Free Software enthusiast and advocate
- Real dream job: become a famous video maker, but not quite there yet 😉

How the Story Started

For the OE workshop 2025 in Brussels,
I wanted to test the latest mainline kernel on these boards:

- `genericarm64` machine
- BeagleBone Black
- BeaglePlay
- Raspberry Pi 5

This Led to My Most Viewed Video



https://www.youtube.com/watch?v=AbNCtm6CF4E

Making Videos is Fun With Free Software

root commit

- Recording: OBS — Open Broadcaster Software
  `https://obsproject.com/`
- Editing: Kdenlive
  `https://kdenlive.org/`
- Even using an Elgato Stream Deck for switching scenes:
  Very well supported on GNU/Linux with
  `https://github.com/StreamController/StreamController`

`https://www.youtube.com/watch?v=kIJ0j_6Jimk`

On the Desktop Too!

https://www.youtube.com/watch?v=zeDCODvmo14

Idea to Use Yocto on My Oldest Board Too

- Technical part: turned out to be easy
  Once I got my hand on the necessary cables
- Video marketing part: much more complicated
  Good to have a captive audience here 😏
- `https://www.youtube.com/shorts/w-N3yh5U8rw`

My Quick Kernel Recipe

root commit

- Very simple, just based on `kernel.bbclass`
- Supports one configuration per board:

```
~/tmp/linux-mainline > tree linux-mainline-6.14+git
linux-mainline-6.14+git
├── beagleboard
│   └── defconfig
├── raspberrypi0-2w-mainline
│   └── defconfig -> ../raspberrypi5-mainline/defconfig
├── raspberrypi2b-mainline
│   └── defconfig
├── raspberrypi3b-mainline
│   └── defconfig -> ../raspberrypi5-mainline/defconfig
├── raspberrypi3b-plus-mainline
│   └── defconfig -> ../raspberrypi5-mainline/defconfig
└── raspberrypi5-mainline
    └── defconfig

7 directories, 6 files
~/tmp/linux-mainline >
```

- Alternative: `meta-linux-mainline` layer by Paul Barker
  `https://github.com/betafive/meta-linux-mainline`

linux-mainline_6.14.bb

```
KVER := "${PV}"
require linux-mainline.inc
SRC_URI:append:raspberrypi5-mainline = " file://defconfig"
SRC_URI:append:raspberrypi0-2w-mainline = " file://defconfig"
SRC_URI:append:raspberrypi3b-plus-mainline = " file://defconfig"
SRC_URI:append:raspberrypi3b-mainline = " file://defconfig"
SRC_URI:append:raspberrypi2b-mainline = " file://defconfig"
SRC_URI:append:beagleboard = " file://defconfig"
```

linux-mainline.inc

```
linux-mainline.inc
DESCRIPTION = "Mainline Linux kernel"
LICENSE = "GPL-2.0-only"
LIC_FILES_CHKSUM = "file://COPYING;md5=6bc538ed5bd9a7fc9398086aedcd7e46"
inherit kernel

S = "${WORKDIR}/git"
SRC_URI = "git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git;branch=linux-${KVER}.y;protocol=https"
SRCREV = "${AUTOREV}"
PV = "${KVER}+git"
PROVIDES = "virtual/kernel"
```

My U-Boot Recipe

root commit

- Quite simple too
- But no class for U-Boot, had to include files from the standard U-Boot recipe

u-boot-mainline_2024.07.bb

```
require recipes-bsp/u-boot/u-boot-common.inc
require recipes-bsp/u-boot/u-boot.inc
PROVIDES = "virtual/bootloader"
RPROVIDES:${PN} = "u-boot"

# U-Boot 2024.07
SRCREV = "3f772959501c99fbe5aa0b22a36efe3478d1ae1c"
UBOOT_MACHINE:beagleboard = "omap3_beagle_defconfig"

DEPENDS += "bc-native dtc-native gnutls-native
python3-pyelftools-native"
```

MACHINE Definition

root commit

beagleboard.conf

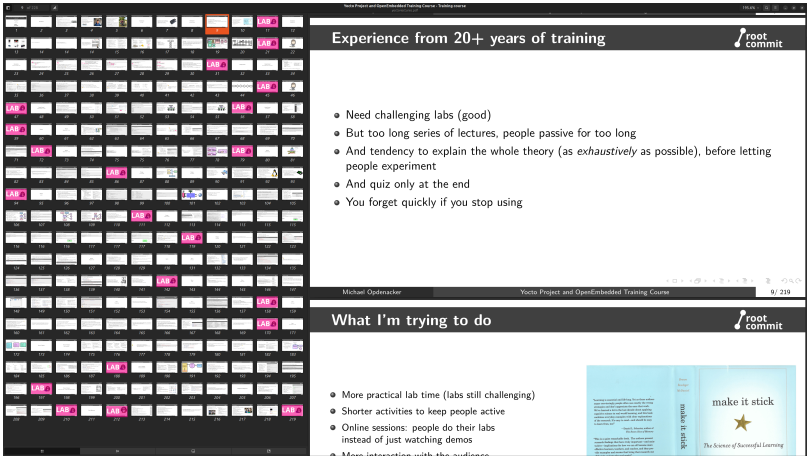
```
PREFERRED_PROVIDER_virtual/xserver ?= "xserver-xorg"
MACHINE_EXTRA_RRECOMMENDS = "kernel-modules"
EXTRA_IMAGEDEPENDS += "virtual/bootloader"
DEFAULTTUNE ?= "cortexa8hf-neon"
include conf/machine/include/arm/armv7a/tune-cortexa8.inc

IMAGE_FSTYPES += "tar.bz2 wic wic.bmap"
WKS_FILE ?= "beagleboard.wks"
MACHINE_ESSENTIAL_EXTRA_RDEPENDS += "kernel-image kernel-devicetree"
do_image_wic[depends] += "mtools-native:do_populate_sysroot dosfstools-native:do_populate_sysroot virtual/bootloader:do_deploy"

SERIAL_CONSOLES ?= "115200;ttyS2"

PREFERRED_PROVIDER_virtual/kernel ?= "linux-mainline"
PREFERRED_VERSION_linux-mainline ?= "6.14%"
KERNEL_IMAGETYPE = "zImage"
KERNEL_DEVICETREE = "ti/omap/omap3-beagle.dtb"

PREFERRED_PROVIDER_virtual/bootloader = "u-boot-mainline"
PREFERRED_VERSION_u-boot-mainline = "2024.07"
SPL_BINARY = "MLO"
UBOOT_SUFFIX = "img"

MACHINE_FEATURES = "usbgadget usbhost vfat alsa"

DTB_FILES = "omap3-beagle.dtb"
IMAGE_BOOT_FILES ?= "u-boot.${UBOOT_SUFFIX} ${SPL_BINARY} ${KERNEL_IMAGETYPE} ${DTB_FILES}"
```

MACHINE Definition — Notes

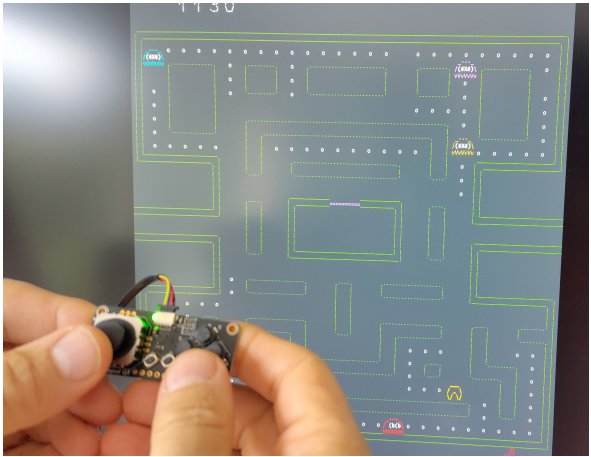
root commit

- Reused from an old, no longer supported layer
- More complicated, but can include all settings put otherwise in `conf/local.conf`

Developing My Free OE / Yocto Course

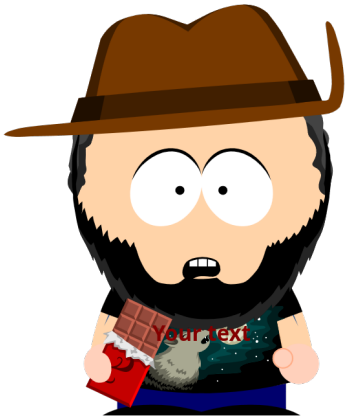


Creative-Commons BY-SA-4.0 materials: `https://rootcommit.com/training/yocto`
220 pages of lectures, 18 practical labs, on TI BeaglePlay and QEMU



Anonymous Person in This Room Loving ASCII Games

root commit



Talk proposals with ASCII games are guaranteed to be accepted

Recipes for the ASCII Games

root commit

- `devtool` really made it easy to create recipes for old ascii games
- MyMan: `https://github.com/kragen/myman/`

  Last modified 16 years ago (qualified!)
  - Managed to compile it by adding Autotools to the sources: `https://github.com/michaelopdenacker/myman`
  - Then developed a patch to drive the game from a joystick
- vitetris: `http://victornils.net/tetris/`
  - Typically 6 years old, but still maintained 😮
  - Hand made Makefile, not targeting cross-compilation
  - Had to define custom tasks

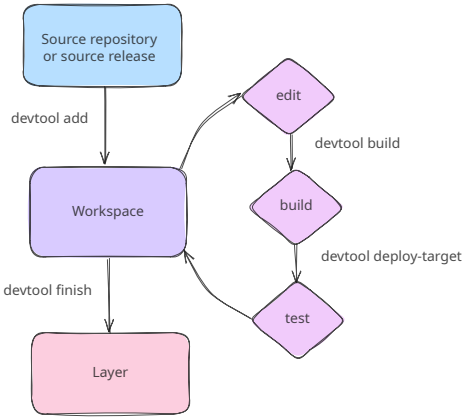
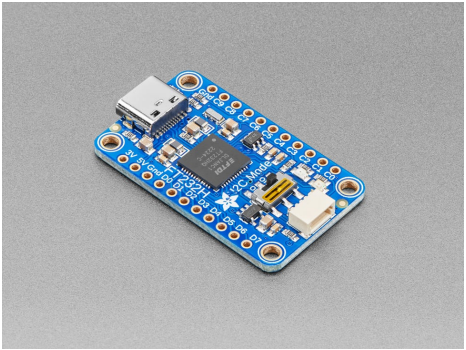


Diagram created with `https://excalidraw.com/` — Open Source of course!

Issues With BeagleBoard — From 2008

root commit

- Was looking great — about to solder expansion headers
- But I2C pins are 1.8V 😲
  Same issue on my Panda Board.
- Didn't have time to find level shifters
- Also had an Adafruid USB-I2C device
  But only supported in userspace 😱
  Found an experimental kernel driver. Haven't managed to fully fix it yet (devices detected though).



Adafruit FT232H breakout:
`https://www.adafruit.com/product/2264`

Issues With BeagleBone (white) — From 2011

root commit

- Looking good — 3.3V I2C
- But serial port only through USB (onboard FTDI chip)
- Good idea at the time, but with a USB mini-AB connector 😱
- No time to make the other serial ports work

→ Will run the demo on BeaglePlay instead

Tweaking the Device Tree of BeaglePlay

root commit

u-boot-ti-staging_2025.01%.bbappend

```
FILESEXTRAPATHS:prepend := "${THISDIR}/files:"
SRC_URI += "file://0001-k3-am625-beagleplay.dts-connect-Adafruit-mini-I2C-ga.patch"
```

files/0001-k3-am625-beagleplay.dts-connect-Adafruit-mini-I2C-ga.patch

```
From b72870f4b87e2f103ab8a7ef451e85ad5526f182 Mon Sep 17 00:00:00 2001
From: Michael Opdenacker <michael.opdenacker@rootcommit.com>
Date: Tue, 15 Apr 2025 05:16:21 +0200
Subject: [PATCH] k3-am625-beagleplay.dts: connect Adafruit mini I2C gamepad

Signed-off-by: Michael Opdenacker <michael.opdenacker@rootcommit.com>
---
 dts/upstream/src/arm64/ti/k3-am625-beagleplay.dts | 7 +++++++-
 1 file changed, 6 insertions(+), 1 deletion(-)

diff --git a/dts/upstream/src/arm64/ti/k3-am625-beagleplay.dts b/dts/upstream/src/arm64/ti/k3-am625-beagleplay.dts
index 9d8f7be53d0..2642b02046d 100644
--- a/dts/upstream/src/arm64/ti/k3-am625-beagleplay.dts
+++ b/dts/upstream/src/arm64/ti/k3-am625-beagleplay.dts
@@ -564,8 +564,13 @@
 &mcu_i2c0 {
        pinctrl-names = "default";
        pinctrl-0 = <&i2c_qwiic_pins_default>;
-       clock-frequency = <100000>;
+       clock-frequency = <400000>;
        status = "okay";
+
```

Kas File to Build Everything

root commit

beagleplay-gaming.yml

```
header:
  version: 14

machine: beagleplay-gaming
distro: asciigaming
target: core-image-gaming

repos:
  poky:
    url: git://git.yoctoproject.org/poky.git
    commit: dcb242eb1926f393fb087388752328 24e69e11e4
    layers:
      meta:
  meta-arm:
    url: https://git.yoctoproject.org/git/meta-arm
    commit: 3cadb81ffaa9f03b92e302843cb22a9cd41df34b
    layers:
      meta-arm:
      meta-arm-toolchain:
  meta-ti:
    url: https://git.yoctoproject.org/git/meta-ti
    commit: 05609ed6e6441c5549496e31b6a28da3a105a7bf
    layers:
      meta-ti-bsp:
  meta-openembedded:
    url: https://git.openembedded.org/meta-openembedded
    commit: edd1a1e284fdcb80cd48d411f235d47f23bc27ae
    layers:
      meta-oe:
  meta-homebrew:
    url: file:///home/mike/yocto-labs/meta-homebrew
    commit: c3ad510802246f1e0d4630186f28b714262d5ebd
  meta-mainline:
    url: file:///home/mike/yocto-labs/meta-mainline
    commit: 7668accbbb44815e985c3e84ac66e943781b5402
```

Yocto and Old Hardware: Key Takeaways

- Super easy when mainline supports a board
  Also true with recent boards with early Linux support
- Only one (Kas) file to share to let others build an image too
- U-Boot is a little less easy, as you have to use U-Boot recipe includes
- `devtool` is great to create recipes and applications at the same time
- Can be worth it to add Autotools to old recipes
- Old hardware has old connectors, didn't realize
- And sometimes obsolete operating conditions (like 1.8V),
  making it hard to use recent accessories.

→Yocto is great to keep old hardware busy... and people too!

Questions? Comments?

- mo@rootcommit.com
- XMPP: omichael@conversations.im
- Signal: rootcommit.01

`https://rootcommit.com/pub/conferences/2025/yocto-workshop-nice/yocto-old-hardware/`

- Sources (LaTeX):
`https://gitlab.com/rootcommit/yocto-old-hardware/`