



# Building RT image with Yocto

Pierre Ficheux ([pierre.ficheux@smile.fr](mailto:pierre.ficheux@smile.fr))

02/2018



## Disclaimer

- Poor English speaker !
- But “good” French speaker and writer :-)

“Loin du français je meurs”  
(Louis-Ferdinand Céline)



\$ whoami

- Embedded Linux developer, writer and teacher
- CTO @ Smile ECS (Embedded & Connected Systems)
- Last book about embedded Linux (in french !)

Pierre Fichoux



EYROLLES

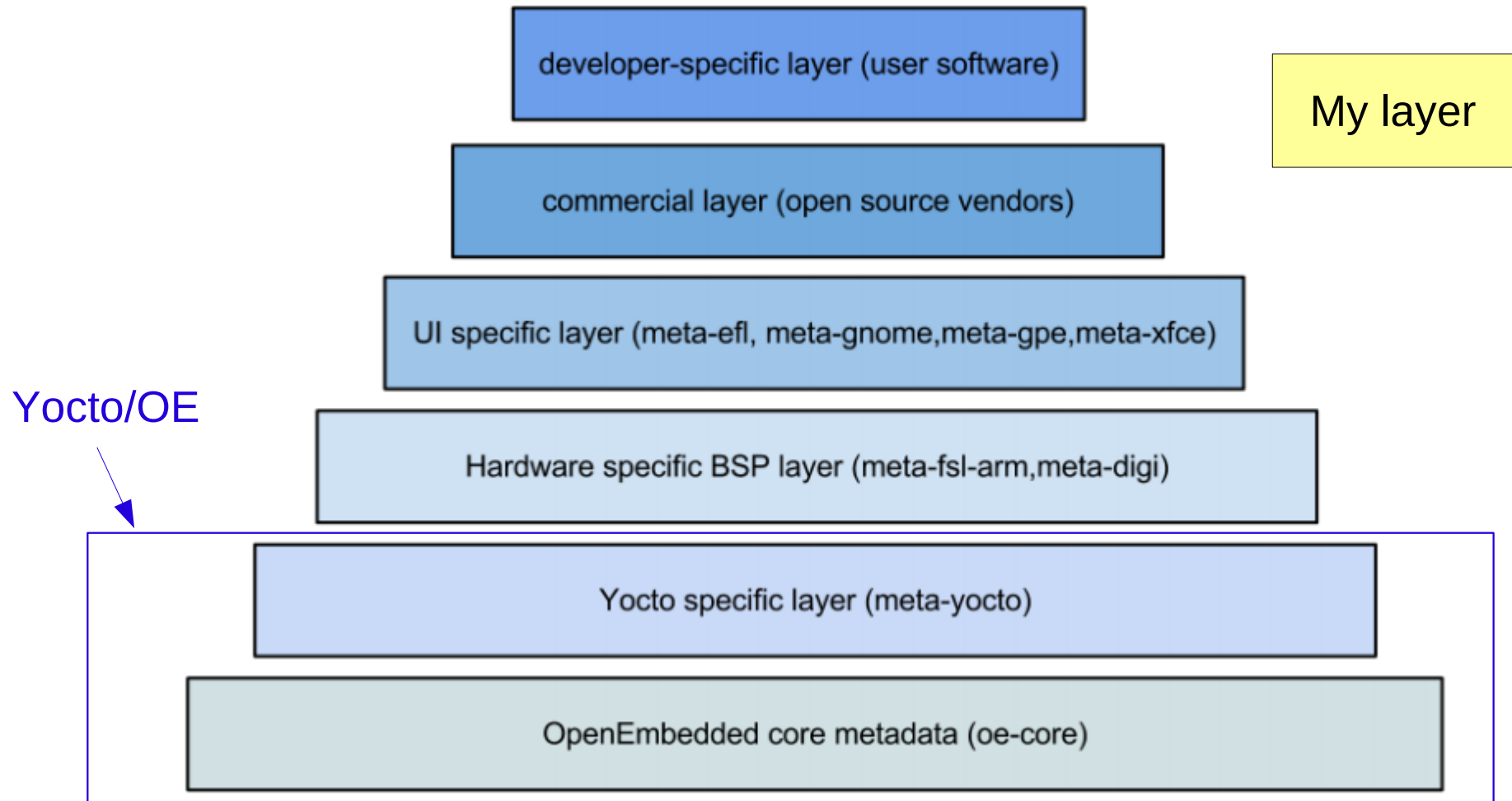


## Yocto quick history

- Open Embedded started in 2003 (OpenZaurus)
- Buildroot was not smart enough (static, no packages)
- Recipes = “how to cross-compile X for target Y”
- BitBake program to run OE recipes
  - Inspired by “Portage” (Gentoo)
  - written in Python
- Yocto started in 2010 (by Intel & friends) including
  - OE (core)
  - BitBake
  - Poky (reference distro)
  - E-Glibc (Glibc since 2014)
  - Smart documentation !!



# Yocto layers





- Recipes (`.bb`)
- Extended recipes (`.bbappend`)
- Based on classes (`.bbclass`) → `inherit`
- Include files (`.inc`) → `include` / `require`
- Configuration files (`.conf`)
- Everything external should be in `meta-<whatever>`
- Very few number of mainline targets (13)
  - QEMU-\* (x86, ARM, MIPS, PPC)
  - BeagleBone Black
  - PPC
  - Generic x86 + x86\_64
  - EdgeRouter
- Most real platforms need a `meta-<hw-type>` layer



## Yocto + Pi 3 « in a nutshell »

- Installing Poky (Yocto reference distro)

```
$ git clone -b <branch> git://git.yoctoproject.org/poky
```

- Installing Raspberry Pi layer (meta-raspberrypi)

```
$ cd poky
```

```
$ git clone -b <branch> git://git.yoctoproject.org/meta-raspberrypi
```

- Creating work directory

```
$ source oe-init-build-env rpi3-build
```

- Adding Pi layer to `bblayers.conf`

```
$ bitbake-layers add-layer ../meta-raspberrypi
```

- Adding machine type to `local.conf`

```
echo "MACHINE = \"raspberrypi3\"" >> conf/local.conf
```

- Building minimal image

```
$ bitbake core-image-minimal
```

- Writing Micro-SD

```
$ sudo dd if=core-image-minimal-raspberrypi3.rpi-sdimg of=/dev/mmcblk0
```



- Several talks about it (FOSDEM, etc.)
- 2 ways
  - PREEMPT\_RT (single kernel patch)
  - RTAI / Xenomai (co-kernel approach - Cobalt, kernel patch + libs)
  - Xenomai can work on top PREEMPT\_RT (Mercury)
- PREEMPT\_RT is “the official Linux RT patch” since 2015 (Thomas Gleixner, Steven Rostedt)
- RTAI is a fork for RTLinux (Paolo Mantegazza)
- Xenomai is a fork for RTAI (Philippe Gerum)
- Co-kernel is more complex to use but more efficient (2x+)
  - Hardware support
  - Specific (RT) kernel interface (RTDM)
  - Application design (migration problem)





## RT and “build systems”

- Yocto and Buildroot are “build systems”
- Easy to build RT image in Buildroot
  - PREEMPT\_RT is a kernel patch
  - Xenomai / RTAI support available
- PREEMPT\_RT support available in Yocto
  - Dedicated kernel recipe = linux-yocto-rt
  - Image recipe = core-image-rt (depending on linux-yocto-rt)
  - Only for test (?)

```
$ grep COMPATIBLE linux/linux-yocto-rt_4.12.bb COMPATIBLE_MACHINE = "(qemux86|  
qemux86-64|qemuarm|qemuppc|qemumips)"      !!!!
```

- Support for Xenomai in meta-eldk (very old)



- Updating recipe (.bb) with a .bbappend
- Default logo in meta/recipes-core/psplash

```
meta/recipes-core/psplash/  
├─ files  
│   └─ psplash-init  
│       └─ psplash-poky-img.h  
└─ psplash_git.bb
```

- Yocto logo in meta-poky/recipes-core/psplash

```
meta-poky/recipes-core/psplash  
├─ files  
│   └─ psplash-poky-img.h ← new logo file !  
└─ psplash_git.bbappend ← FILESETRAPATHS_prepend_poky := "${THISDIR}/files:"
```

- Enabling I<sup>2</sup>C for Pi in rpi\_config\_git.bbappend

```
do_deploy_append() {  
    # Enable i2c by default  
    echo "dtparam=i2c_arm=on" >> ${DEPLOYDIR}/bcm2835-bootfiles/config.txt  
}
```



- Just “extend” kernel for PREEMPT\_RT
  - Create a new layer meta-<rt-test-name>
  - Add a recipes-kernel/linux-<board>-rt directory

```
meta-article-bis/  
├─ conf  
│   └─ layer.conf  
└─ recipes-kernel  
    └─ linux-rpi3  
        ├── files  
        │   ├── defconfig ← kernel config file  
        │   └─ patch-4.4.50-rt63.patch ← PREEMPT_RT patch  
        └─ linux-rpi3_4.4.bbappend
```

- PREEMPT\_RT kernel branch available for some boards (BBB)
- Specific layer is needed for Xenomai



- More difficult as :
  - Need to run `prepare-kernel.sh` script to apply kernel patch (l-pipe)
  - Need to install user-space files
- New layer meta-xenomai
  - Kernel recipe `linux-xenomai-<board>`
  - User-space recipe (Autotools based)
- Kernel recipe adds `do_prepare_kernel()` function for patch (Cobalt support only)
- Executed before `do_configure()` (use `addtask`)



- Add *meta-xenomai* layer path

```
$ cd <path>/poky
```

```
$ git clone https://github.com/pficheux/meta-xenomai.git
```

```
$ cd rpi3-build
```

```
$ bitbake-layers add-layer ../meta-xenomai
```

- Add Xenomai support to `local.conf` or dedicated image recipe

```
PREFERRED_PROVIDER_virtual/kernel = "linux-xenomai-<board>"
```

```
IMAGE_INSTALL_append = " xenomai rt-tests"
```



Xenomai, TBD

- Test on Yocto 2.3 + 2.4
- New boards (mostly done by end-users)



# References

- Yocto layer meta-xenomai <https://github.com/pficheux/meta-xenomai>
- Xenomai project <http://xenomai.org/>
- PREEMPT\_RT Wiki [https://rt.wiki.kernel.org/index.php/Main\\_Page](https://rt.wiki.kernel.org/index.php/Main_Page)
- Embedded Linux Systems with the Yocto Project par Rudolf J. Streif <https://www.pearson.com/us/higher-education/program/Streif-Embedded-Linux-Systems-with-the-Yocto-Project/PGM275649.html>