

Perl on embedded Linux with BuildRoot

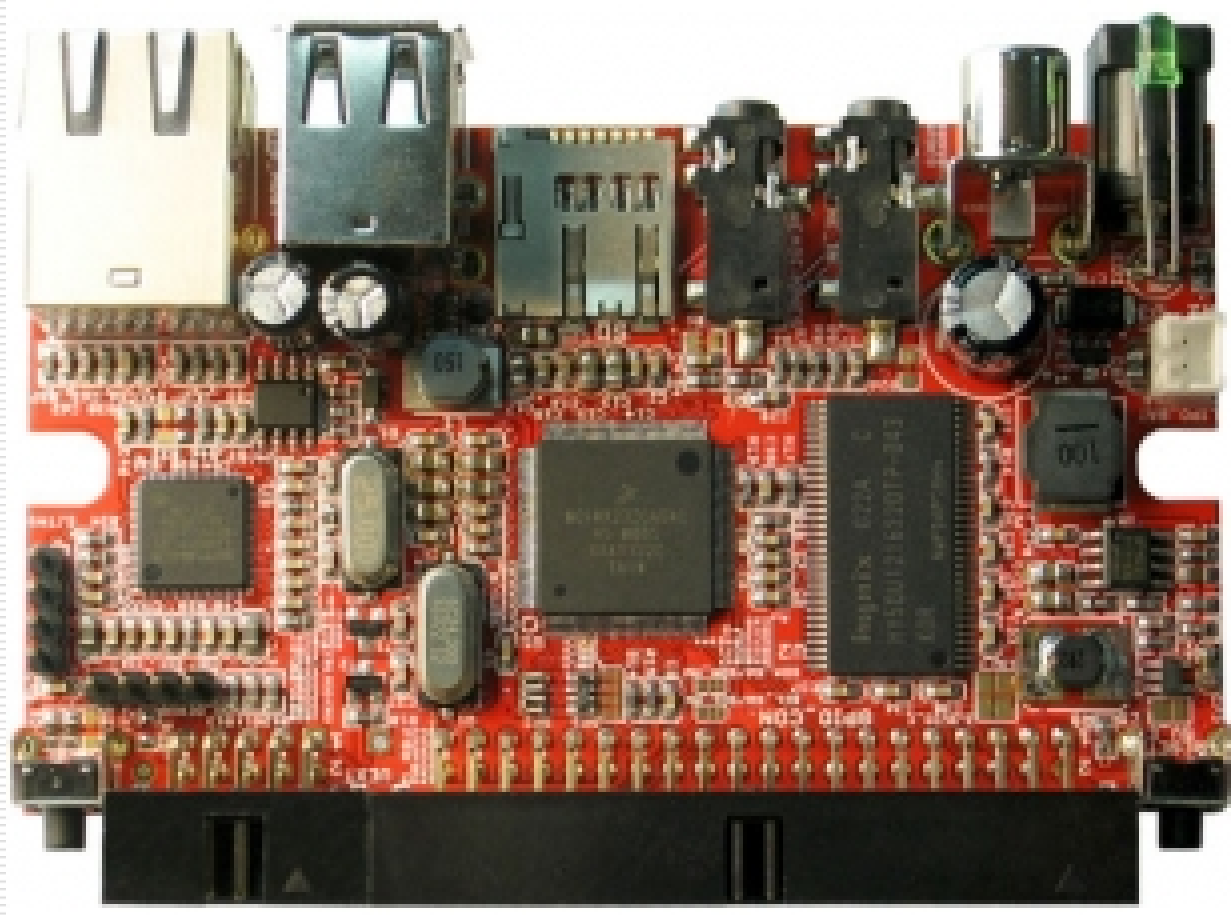
(40')



François Perrad

francois.perrad@gadz.org

OLinuXino iMX233 Board



OLinuXino iMX233 Board

- Features
 - Freescale i.MX233 @ 454MHz
 - ARM926J Core (ARMv5)
 - RAM 64 Mb
 - microSD card connector
 - Ethernet 100 Mbit
 - 2 USB Host
 - TV video output, 40 pin GPIO
 - [Arch Linux ARM](#) support
- [Olimex Ltd](#), a Bulgarian company
- Schematic Open Source (like Arduino)
- 45 €

BuildRoot Project

- ❑ Created in 2001 by uClibc dev. for testing
- ❑ Around 2005, becomes an independant build system for embedded linux device
- ❑ Peter Korsgaard, maintainer since 2009
- ❑ Scheduled releases every 3 months
- ❑ 300+ contributors
- ❑ 1000+ packages
- ❑ Homepage : buildroot.net
- ❑ Well documented ([manual](#))
- ❑ Git repository:
<http://git.buildroot.net/buildroot/tree/>
- ❑ Mailing list : buildroot@busybox.net
- ❑ GPL v2 Licence

BuildRoot aims

- BR is a build system which configures/build all components of an embedded Linux system
 - Cross toolchain
 - Bootloader (U-Boot, Barebox, ...)
 - Linux kernel
 - Userland libraries & applications
 - Rootfs image
- from sources (fetched by BR), with your configuration choices and in a reproducible way
- The minimal system contains only the Linux Kernel and Busybox
- BuildRoot is easily usable and customizable

BuildRoot aims

- Many architectures :
 - x86, x86_64, ARM, MIPS, PowerPC, NIOS, microblaze
- Many toolchains :
 - Linaro, Sourcery CodeBench, ...
- Many C librairies (with internal toolchain) :
 - uClibc
 - glibc
 - eglibc
 - musl

BuildRoot Technos

- Makefile (gmake)
- Kconfig language
Documentation/kbuild/kconfig-language.txt
- Patch
- Bash
- Git (format-patch, send-email, ...)



Package example

□ **Config.in**

```
config BR2_PACKAGE_GZIP
```

```
bool "gzip"
```

```
depends on BR2_USE_WCHAR
```

```
help
```

```
Standard GNU compressor.
```

```
Provides things like gzip, gunzip, gzcat, etc.
```

```
http://www.gnu.org/software/gzip/gzip.html
```

```
comment "gzip needs a toolchain w/ wchar"
```

```
depends on !BR2_USE_WCHAR
```


Package example

□ `gzip.mk`

`GZIP_VERSION = 1.6`

`GZIP_SOURCE = gzip-$(GZIP_VERSION).tar.xz`

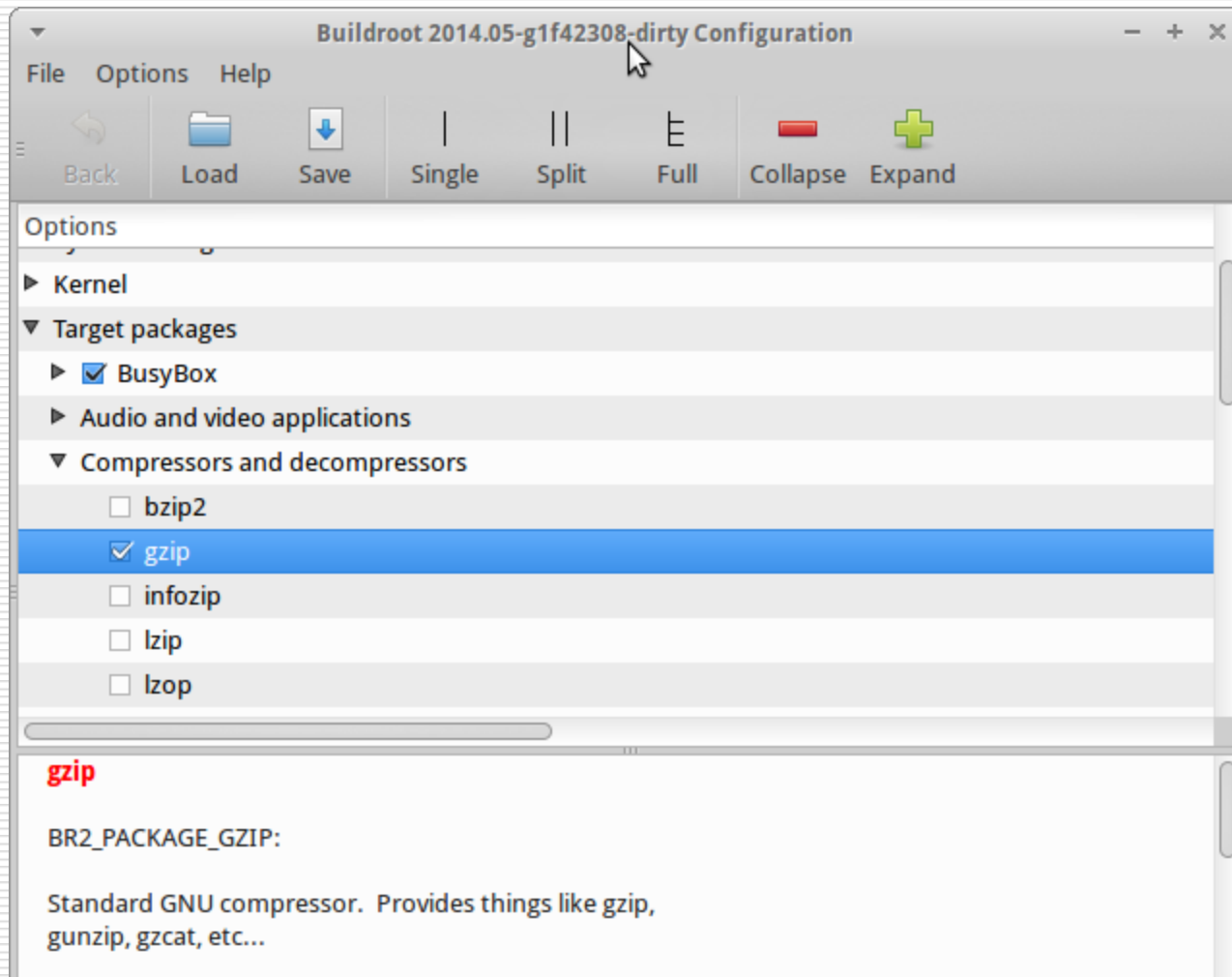
`GZIP_SITE = $(BR2_GNU_MIRROR)/gzip`

`GZIP_LICENSE = GPLv3+`

`GZIP_LICENSE_FILES = COPYING`

`$(eval $(autotools-package))`

Package example



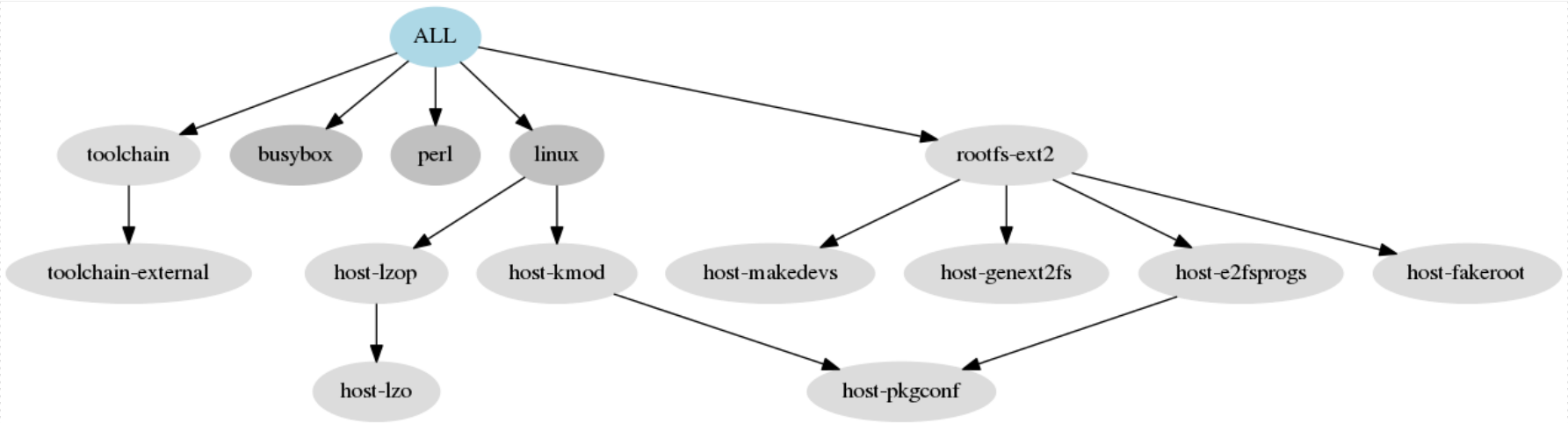
Perl History in BR

- 2009.02 : microperl 5.8.8
- 2012.05 : microperl 5.12.4
- 2012.11 : perl 5.16.1
- 2013.02 : perl 5.16.2
- 2013.05 : perl 5.16.3
- 2013.11 : perl 5.18.1
- 2014.02 : perl 5.18.2
- 2014.05 : CPAN infrastructure

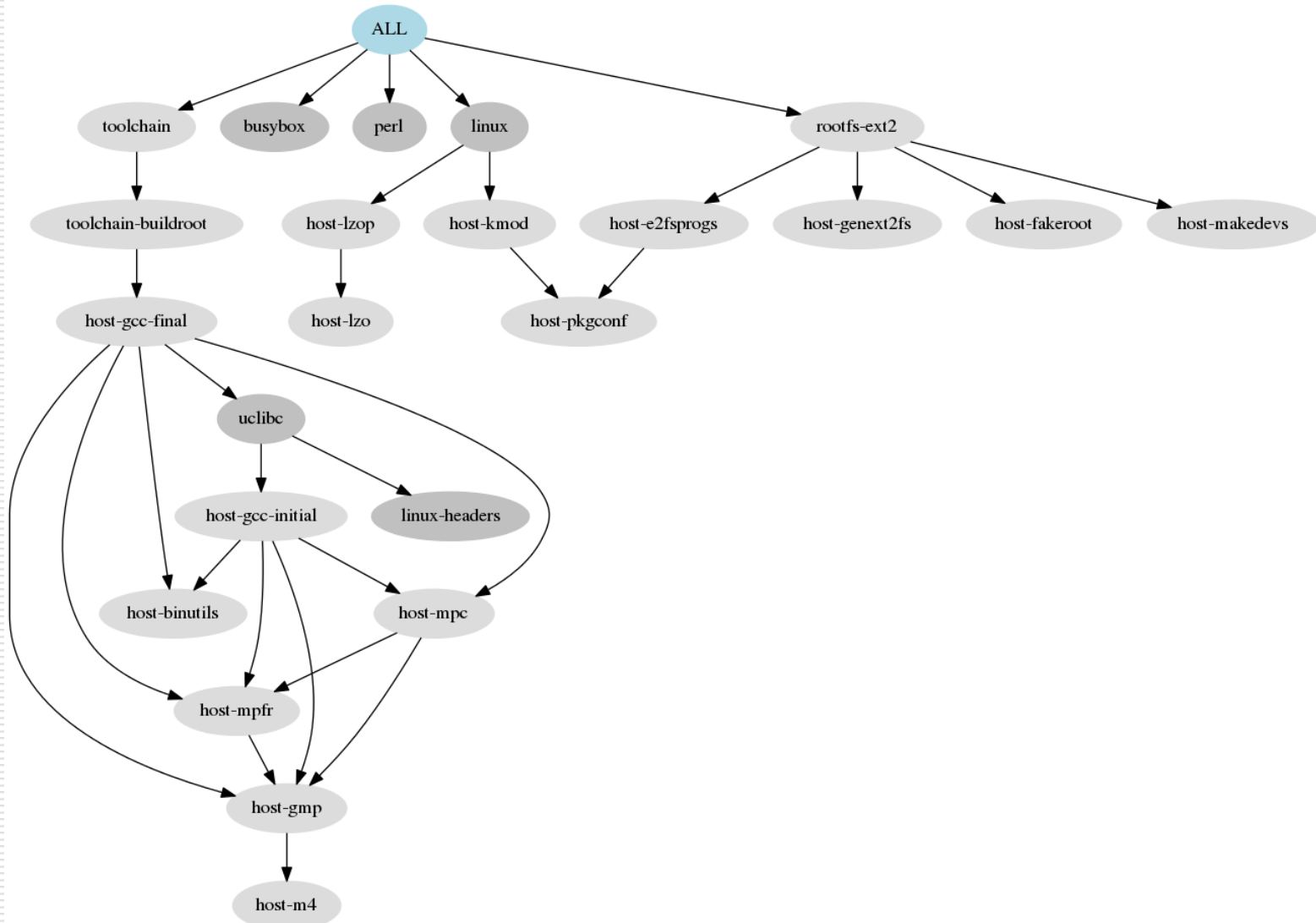
Perl-Cross project

- Perl-Cross provides alternative **configure** script (**bash**), top-level **Makefile**, and some auxiliary files
- Started in 2009
- Used in BR since 2012, with series 5.16 and 5.18
- Alex Suykov
 - single author (and single point of failure)
 - 5.20 ? 5.2x ?
- Homepage <http://arsv.github.io/perl-cross/>
- Previously on <http://perlcross.berlios.de/>

Perl with BR



Perl with BR



CPAN infrastructure

- `package/pkg-perl.mk`
 - target & host package
 - `Makefile.PL` & `Build.PL`
 - Configure, Build, Install
 - Use BR infrastructure for download, extract, patch

- `support/scripts/scancpan`
 - Populates Perl packages with dependencies and metadata fetched from <https://metacpan.org/>
 - Perl script using `MetaCPAN-API-Tiny`

Perl package example

□ Config.in

```
config BR2_PACKAGE_PERL_DATETIME
    bool "perl-datetime"
    select BR2_PACKAGE_PERL_DATETIME_LOCALE
    select BR2_PACKAGE_PERL_DATETIME_TIMEZONE
    select BR2_PACKAGE_PERL_PARAMS_VALIDATE
    select BR2_PACKAGE_PERL_TRY_TINY
    help
        A date and time object
```

□ perl-datetime.mk

```
PERL_DATETIME_VERSION = 1.10
PERL_DATETIME_SOURCE = DateTime-$(PERL_DATETIME_VERSION).tar.gz
PERL_DATETIME_SITE = $(BR2_CPAN_MIRROR)/authors/id/D/DR/DROLSKY/
PERL_DATETIME_DEPENDENCIES = perl host-perl-module-build perl-
    datetime-locale perl-datetime-timezone perl-params-validate perl-
    try-tiny
PERL_DATETIME_LICENSE = artistic_2

$(eval $(perl-package))
```


Dancer2 demo

- Busybox, Dropbear (ssh), ntp, Perl
- Dancer2, GD, DateTime
 - 74 target packages
 - 21 host packages
 - ExtUtils::MakeMaker, Module::Build, Module::Build::Tiny
- 60.3 MB used on disk
- Dancer2 is a good test for the BR infrastructure, but not the best Web framework for this kind of target

Dancer2 demo

Perl is dancing on eLinux Buildroot
You've joined the dance floor!

[About your application's environment](#)

uname -srm

```
Linux 3.13.6 armv5tejl
```

cat /proc/cpuinfo

```
processor       : 0
model name     : ARM926EJ-S rev 5 (v5l)
Features      : swp half fastmult edsp java
CPU implementer : 0x41
CPU architecture: 5TEJ
CPU variant   : 0x0
CPU part      : 0x926
CPU revision  : 5

Hardware      : Freescale MXS (Device Tree)
Revision     : 0000
Serial       : 0000000000000000
```

df -h

Filesystem	Size	Used	Available	Use%	Mounted on
/dev/root	78.1M	62.7M	11.5M	85%	/
devtmpfs	29.1M	0	29.1M	0%	/dev
tmpfs	20.1M	0	20.1M	0%	/dev/shm

Components

- [Dancer2](#)
- [Perl5](#)
- [perl-cross](#)
- [Buildroot](#)
- [Busybox](#)
- [Linux](#)
- [Olimex](#) OLinuXino
- [Freescale](#) i.MX233
- [ARM](#) 926

Sources

on [Github](#) by [fperrad](#)
and with the [recipe](#).

Demo recipe

- All stuff on Github :
<https://github.com/fperrad/br/tree/dancer2>

- Build

```
$ make defconfig
```

```
BR2_EXTERNAL=dancer2
```

```
BR2_DEFCONFIG=dancer2/configs/olimex_imx233_olinuxino_defco
```

```
$ make
```

- Flashing in SD card

```
$ umount /dev/mmcblk0p2
```

```
$ sudo dd if=output/images/imx23_olinuxino_dev_linux.sb  
bs=512 of=/dev/mmcblk0p1 seek=4
```

```
$ sudo dd if=output/images/rootfs.ext2 of=/dev/mmcblk0p2  
bs=512
```

```
$ sync
```

Working with Qemu system

- Working without hardware target
- BR has many defconfig for Qemu

```
$ make qemu_x86_defconfig
```

```
$ make gconfig
```

```
$ make
```

```
$ qemu-system-i386 -M pc -kernel  
output/images/bzImage -drive  
file=output/images/rootfs.ext2,if=ide  
-append root=/dev/sda -net  
nic,model=rtl8139 -net user
```

- Next step ? VirtualBox or VmWare

Testing on Target

- BuildRoot doesn't handle the test step
- My way :
 - Copy the whole build tree (130 MB) on the target
 - Then :
 - (remember no **make**, no toolchain, ...)
 - `$ cd t`
 - `$ ln -s ../perl perl`
 - `$./perl harness`
 - 256 MB of RAM required
 - Impossible on real target OLinuXino
 - With **qemu-system-arm**
 - Take 6 hours or more
 - Many patches for test suite
 - merged in perl 5.20.0

Bibliography / Webography

- <http://www.buildroot.net/>
- <https://www.olimex.com/Products/OLinuXino/iMX233/>
- <http://archlinuxarm.org/>
- <http://arsv.github.io/perl-cross/>
- <https://github.com/fperrad/br/tree/dancer2>