Driving and virtualizing control systems: the Open Source approach used in WhiteRabbit

Javier Muñoz Mellid & Samuel Iglesias Gonsalvez

Academia-Industry Matching event on Technology of Controls for Accelerators and Detectors
November 2013

Agenda

- Knowing Igalia
- The value of the Open Source approach
- Open Source approach in WhiteRabbit
- A walk on technical details and demo

What is Igalia?

- Open Source Company
- 40 engineers and hackers working around the globe
- Hacking upstream in different technologies and communities
  - kernel (Linux)
  - virtualization (QEMU/KVM)
  - browsers/multimedia (WebKit, Blink, GStreamer...)
  - compilers (V8, JavaScriptCore...)
  - UI (GTK+,...)
  - documents (Evince, LibreOffice...)
  - distros (Debian, Tizen...)
  - automotive/IVI
- 13 years old now!
- www.igalia.com

Partnering

- Linux Foundation
  www.linuxfoundation.org/news-media/announcements/2011/04/igalia-joins-linux-foundation
- W3C
  www.igalia.com/nc/igalia-247/news/item/igalia-joins-the-world-wide-web-consortium
- Tizen Association
  www.igalia.com/nc/igalia-247/news/item/igalia-joins-the-tizen-association-partner-program/
- ...

What services and solutions provides Igalia?

- We are experts in Open Source Software solutions
- We help companies and institutions to achieve open developments aligned with their culture and business models
- We collaborate with customers through concrete and well-defined projects
- Usual customer's needs ranges from expert technical development, support and training to strengthen in-house teams or requiring a technical link between project and open communities

Can your project take advantage of an open approach?

- To take serious advantage of Open Source approach you should avoid using open and free software only (consuming). You should take in consideration producing it as WhiteRabbit community is doing!
- It requires understanding the 'upstreaming concept' in Open Source projects
- We will see the upstream concept quickly in order to illustrate it with the WhiteRabbit experience

What is upstreaming?

- Upstreaming refers to the practice of contributing source code that has been developed independently or in-house back to an open source project.
- Having code accepted upstream provides several benefits to the open source project, the companies submitting code to upstream projects, and the open source ecosystem in general.

Advantages of Upstream Alignment

- Accelerate Innovation
- Minimize R&D Costs
- Gain Additional Contributors
- Increase Code Quality
- Enable Faster Integration and Testing
- Influence the project's direction
- Provide stability to the open source project

CERN and Igalia collaboration (I)

- Started in 2012
- Collaboration between CERN's BE-CO-HT section and Igalia's Kernel/Virtualization team
- Worked around FMC/TDC board in the WhiteRabbit context
- Igalia was in charge of Linux kernel driver development, testing and quality for this board
- In this project we started to develop virtual hardware for an open virtualizator/emulator (QEMU/KVM)
- We introduced part of our collaboration with CERN based on automatic testing, continuous integration and quality for Linux drivers using virtual hardware in LinuxCon 2012
- In this conference, LinuxCon, Igalia promotioned some of the technologies used in CERN and the WhiteRabbit project as part of this collaboration

CERN and Igalia collaboration (II)

In summary, this collaboration related with control aspects raised value in the following aspects:

- Speeding up Linux driver development for one custom board designed in CERN
- Developed virtual hardware for automatic testing, continuous integration and safe tests
- Supported the collaboration's upstream process with code in different communities (Linux, QEMU, Open Hardware Repository and KiCAD mainly)
- We introduced this collaboration and experience in LinuxCon 2012. A very relevant conference in the industry
- Related to this collaboration, some of our customers found useful the open virtual hardware approach to reduce development times, working in parallel with hardware designers, redesigning firmware's interface or running agressive testing

Technical details and demo

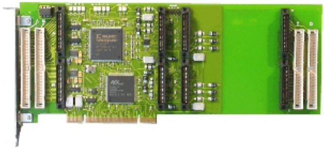
Our work in kernel

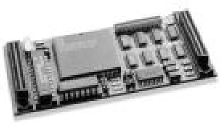
- Industry Pack bus
  - TEWS TPCI-200.
  - GE IP-OCTAL-232.
- FMC bus (WhiteRabbit project)
  - FMC TDC board

Industry Pack drivers

- Open Source drivers developed at CERN.
- Mainstream Linux kernel support added in 2012.
  - In Staging in May 2012.
  - Moved to mainstream in November 2012.
- Three drivers:
  - ipack bus driver
  - TEWS TPCI-200 carrier board
  - GE IP-OCTAL: 8 channel serial port device

Industry Pack boards

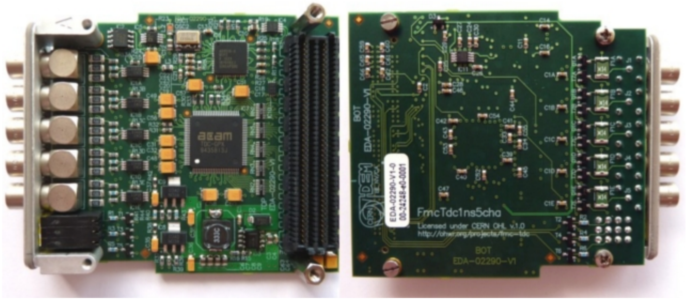




FMC TDC driver

- Hosted in OHWR.
  - `http://www.ohwr.org/projects/fmc-tdc-sw`
- Tested on a SPEC board.
- It uses ZIO framework to bring an I/O device to user-space.
- It uses the FMC bus driver to register the mezzanine in the system.
  - FMC bus is supported in mainline kernel as of June 2013.

FMC TDC board


acam
TDC-GPX
FmcTdc1ns5cha
Licensed under CERN OHL v1.0
http://ohwr.org/projects/fmc-tdc
BOT
EDA-02290-V1
EDA-02290-V1-0

Our work in virtualization

Virtualized models

- Industry Pack: Added to QEMU in 2012.
  - TEWS TPCI-200
  - GE IP-OCTAL-232
- FMC TDC
  - SPEC board (only needed bits)
  - FMC TDC board

FMC TDC board virtualization

- When developing the driver, we saw some problems:
  - No enough devices for all developers
  - FW is not mature enough
  - Which is the source of the error?

FMC TDC board virtualization

- So we created a virtual model of the device
  - Virtualized just some specific bits of SPEC
  - Same memory addresses and registers than the real FW
  - We can test error conditions to improve driver’s robustness
- Could it be possible to test the driver automatically?

FMC TDC board: Error conditions

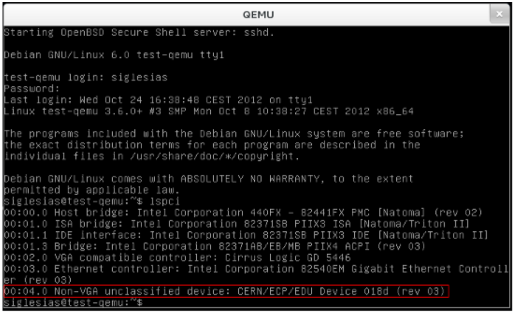
It was needed to test error conditions on the driver

- Generic stuff
  - The virtual board in being detected by the driver as a genuine device
  - Normal mode: no error injection
- DMA errors
- Set up different input pulse configurations

How we did that

- QEMU (`http://wiki.qemu.org`)
  - Open source machine emulator and virtualizer
  - Allow us to play with different setups
  - If has nice features: snapshot, shared folders, no screen, etc
- Buildbot (`http://trac.buildbot.net/`)
  - Continuous integration system designed to automate the build/test cycle
  - Python. Web interface to check the logs
- Our own testing suite
  - Based on Sam's experience working with PTS in the past
  - Needed more flexibility: different setups for the same test
  - Developed in Python, as PTS

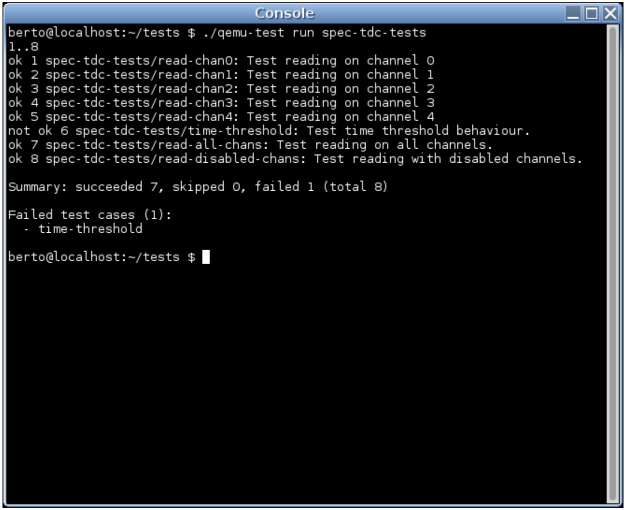
The result (I)



```
Starting OpenBSD Secure Shell server: sshd.

Debian GNU/Linux 6.0 test-qemu tty1

test-qemu login: siglesias
Password:
Last login: Wed Oct 24 16:38:48 CEST 2012 on tty1
Linux test-qemu 3.6.0+ #3 SMP Mon Oct 8 10:38:27 CEST 2012 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
siglesias@test-qemu:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371SB PIIX3 IDE [Natoma/Triton II]
00:01.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 03)
00:02.0 VGA compatible controller: Cirrus Logic GD 5446
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controll
er (rev 03)
00:04.0 Non-VGA unclassified device: CERN/ECP/EDU Device 018d (rev 03)
siglesias@test-qemu:~$
```

```
00:04.0 Non-VGA unclassified device: CERN/ECP/EDU Device 018d (rev 03)
```

The result (II)

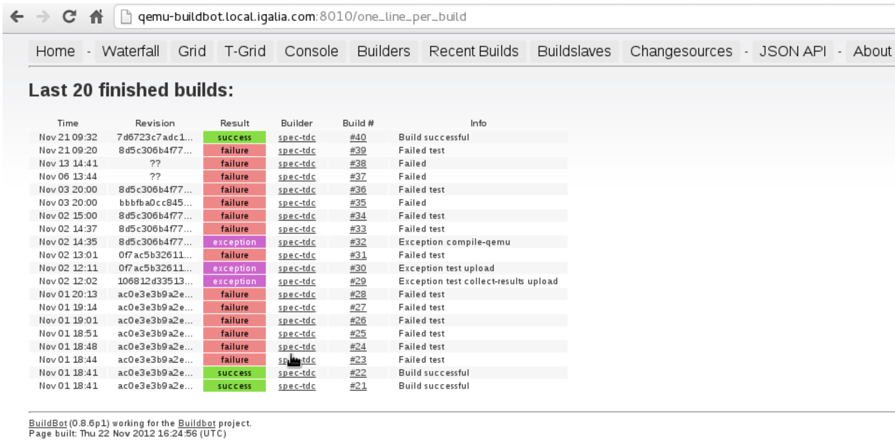


```
berto@localhost:~/tests $ ./qemu-test run spec-tdc-tests
1..8
ok 1 spec-tdc-tests/read-chan0: Test reading on channel 0
ok 2 spec-tdc-tests/read-chan1: Test reading on channel 1
ok 3 spec-tdc-tests/read-chan2: Test reading on channel 2
ok 4 spec-tdc-tests/read-chan3: Test reading on channel 3
ok 5 spec-tdc-tests/read-chan4: Test reading on channel 4
not ok 6 spec-tdc-tests/time-threshold: Test time threshold behaviour.
ok 7 spec-tdc-tests/read-all-chans: Test reading on all channels.
ok 8 spec-tdc-tests/read-disabled-chans: Test reading with disabled channels.

Summary: succeeded 7, skipped 0, failed 1 (total 8)

Failed test cases (1):
  - time-threshold

berto@localhost:~/tests $
```

The result (III)

- `http://tdc-buildbot.igalia.com`
- Username: buildbot
- Password: qemulinux

The result (IV)



qemu-buildbot.local.igalia.com:8010/one_line_per_build

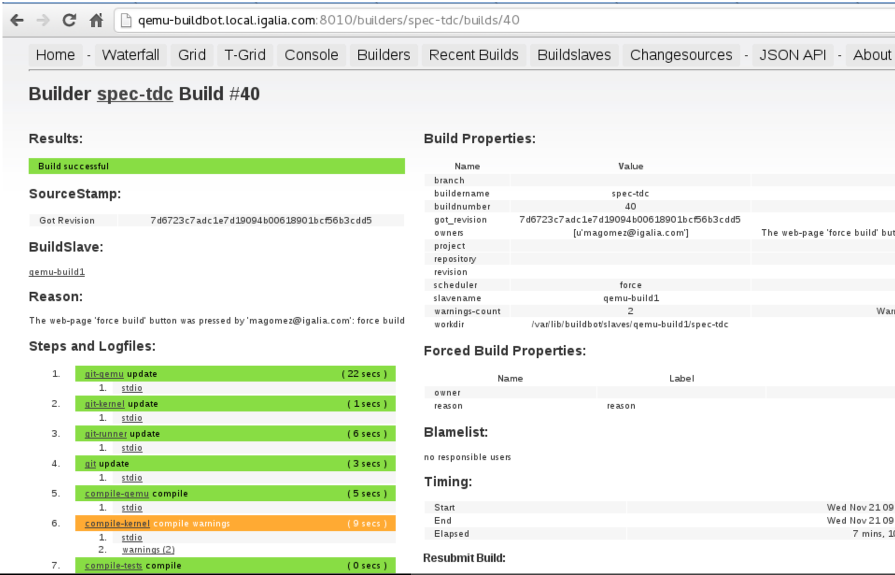
Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About

Last 20 finished builds:

Time	Revision	Result	Builder	Build #	Info
Nov 21 09:32	7d6723c7adc1...	success	spec-tdc	#40	Build successful
Nov 21 09:20	8d5c306b4f77...	failure	spec-tdc	#39	Failed test
Nov 13 14:41	??	failure	spec-tdc	#38	Failed
Nov 06 13:44	??	failure	spec-tdc	#37	Failed
Nov 03 20:00	8d5c306b4f77...	failure	spec-tdc	#36	Failed test
Nov 03 20:00	bbbfba0cc845...	failure	spec-tdc	#35	Failed
Nov 02 15:00	8d5c306b4f77...	failure	spec-tdc	#34	Failed test
Nov 02 14:37	8d5c306b4f77...	failure	spec-tdc	#33	Failed test
Nov 02 14:35	8d5c306b4f77...	exception	spec-tdc	#32	Exception compile-qemu
Nov 02 13:01	0f7ac5b32611...	failure	spec-tdc	#31	Failed test
Nov 02 12:11	0f7ac5b32611...	exception	spec-tdc	#30	Exception test upload
Nov 02 12:02	106812d33513...	exception	spec-tdc	#29	Exception test collect-results upload
Nov 01 20:13	ac0e3e3b9a2e...	failure	spec-tdc	#28	Failed test
Nov 01 19:14	ac0e3e3b9a2e...	failure	spec-tdc	#27	Failed test
Nov 01 19:01	ac0e3e3b9a2e...	failure	spec-tdc	#26	Failed test
Nov 01 18:51	ac0e3e3b9a2e...	failure	spec-tdc	#25	Failed test
Nov 01 18:48	ac0e3e3b9a2e...	failure	spec-tdc	#24	Failed test
Nov 01 18:44	ac0e3e3b9a2e...	failure	spec-tdc	#23	Failed test
Nov 01 18:41	ac0e3e3b9a2e...	success	spec-tdc	#22	Build successful
Nov 01 18:41	ac0e3e3b9a2e...	success	spec-tdc	#21	Build successful

BuildBot (0.8.6p1) working for the Buildbot project.
Page built: Thu 22 Nov 2012 16:24:56 (UTC)

The result (V)


qemu-buildbot.local.igalia.com:8010/builders/spec-tdc/builds/40
Home - Waterfall Grid T-Grid Console Builders Recent Builds Buildslaves Changesources - JSON API - About

Builder spec-tdc Build #40

Results:

Build successful

SourceStamp:

Got Revision	7d6723c7adc1e7d19094b00618901bcf56b3cdd5

BuildSlave:

qemu-build1

Reason:

The web-page 'force build' button was pressed by 'magomez@igalia.com': force build

Steps and Logfiles:

1. git-qemu update (22 secs)
   1. stdio
2. git-kernel update (1 secs)
   1. stdio
3. git-runner update (6 secs)
   1. stdio
4. git update (3 secs)
   1. stdio
5. compile-qemu compile (5 secs)
   1. stdio
6. compile-kernel compile warnings (9 secs)
   1. stdio
   2. warnings (2)
7. compile-tests compile (0 secs)

Build Properties:

Name	Value	
branch		
buildername	spec-tdc	
buildnumber	40	
got_revision	7d6723c7adc1e7d19094b00618901bcf56b3cdd5	
owners	[u'magomez@igalia.com']	The web-page 'force build' but
project		
repository		
revision		
scheduler	force	
slavename	qemu-build1	
warnings-count	2	War
workdir	/var/lib/buildbot/slaves/qemu-build1/spec-tdc	

Forced Build Properties:

Name	Label	
owner		
reason	reason	

Blamelist:

no responsible users

Timing:

Start	Wed Nov 21 09
End	Wed Nov 21 09
Elapsed	7 mins, 1

Resubmit Build:

Results

- We could test different conditions:
  - Choose memory size, virtual machine image, network, etc
  - Different kernel versions
  - As is uses continuous integration -> Detect which commit added the error
- Based on Open Source technologies
  - Can be modified to validate complex systems before they are done

Links (I)

- CERN/Igalia collaboration
  - www.igalia.com/kernel
  - blogs.igalia.com/jmunhoz/blog/2012/12/14/follow-the-white-rabbit-working-with-cern.html
- Industry Pack drivers
  - blogs.igalia.com/siglesias/2012/11/23/fmc-tdc-driver
- FMC TDC driver
  - FMC projects SPEC, FMC TDC and FMC TDC software
  - blogs.igalia.com/siglesias/2012/11/23/fmc-tdc-driver/

Links (II)

- CI, testing suites and virtual hardware (QEMU)
  - blogs.igalia.com/berto/2012/10/03/industrypack-qemu-and-linuxcon/
  - blogs.igalia.com/berto/2012/11/28/qemu-and-open-hardware-spec-and-fmc-tdc/
  - events.linuxfoundation.org/.../pdf/lceu2012_garcia.pdf
  - blogs.igalia.com/magomez/2012/11/28/continuous-integration-and-testing-driver-development-and-virtual-hardware-the-fmc-tdc-experience/

- Fixing bugs upstream in KiCAD
  - blogs.igalia.com/jaragunde/2013/01/21/introducing-kicad-because-open-hardware-needs-open-tools/
  - blogs.igalia.com/jaragunde/2013/01/30/kicad-bug-squashing-round-1/
  - blogs.igalia.com/jaragunde/2013/02/14/kicad-bug-squashing-round-2/

Questions & Answers

Thank you!