

A Tango Device Server for Zynq XADC using AMP

Key words :

Tango Distributed Control System
OpenEmbedded Linux, Zynq XADC & OCM, AMP (asymmetric multi processor)

Participants :

Baptiste CAMUS	baptiste.camus@e.ujf-grenoble.fr
Nicolas GROS	nicolas.gros@e.ujf-grenoble.fr
Nicolas COUPIN	nicolas.coupin@e.ujf-grenoble.fr
Bertrand RIGAUD	bertrand.rigaud@e.ujf-grenoble.fr

Supervisor :

PONSARD Raphael [raphael.ponsard \(-at-\) ac-grenoble.fr](mailto:raphael.ponsard (-at-) ac-grenoble.fr) , [rponsard \(-at-\) gmail.com](mailto:rponsard (-at-) gmail.com)

University :

Université Joseph Fourier, IUT GEII, Grenoble, FRANCE

Category :

emb_student

Team number :

XIL-63895

Platform :

Zedboard

Youtube :

<https://youtu.be/J34uoM3a1pM>

Paper :

<https://docs.google.com/document/d/1glvFaKJogfreaWklSOiwTDw1JLJJtfZDZQDQ5kmlFsl>

Github:

<https://github.com/raph38130/HDL/tree/master/TangoAmperemeter>

[TANGO DCS AT THE ESRF](#)

[Forewords](#)

[Zedboard/Zynq uses case at the ESRF](#)

[OpenEmbedded toolchain](#)

[The TFL AmpereHourMeter](#)

[Hardware Design Vivado 2016.1](#)

[AXI GPIO](#)

[XADC](#)

[OLED IP](#)

[Asymmetric Multi Processor](#)

[Device tree dts](#)

[Bootargs](#)

[uEnv.txt](#)

[Linux kernel options](#)

[Zynq_remoteproc kernel modules](#)

[Cpu1app](#)

[Boot.s](#)

[Ld.script](#)

[Tango XADC Demo](#)

[Pogo wizard tools](#)

[Ubuntu tango linux vdi](#)

[Zedboard SD Image](#)

[Fat partition](#)

[Ext4 partition](#)

[Full image here](#)

[Starting Device Server on ZedBoard](#)

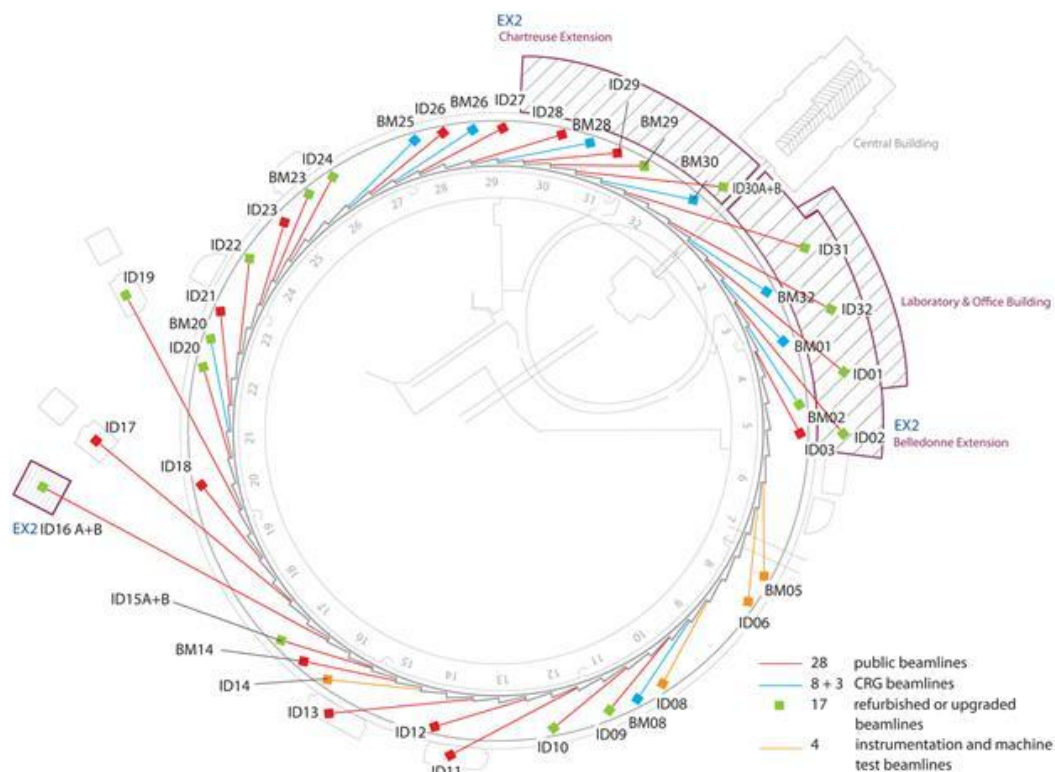
[Starting jive or jdraw synoptic on tango virtualbox](#)

TANGO DCS AT THE ESRF

Forewords

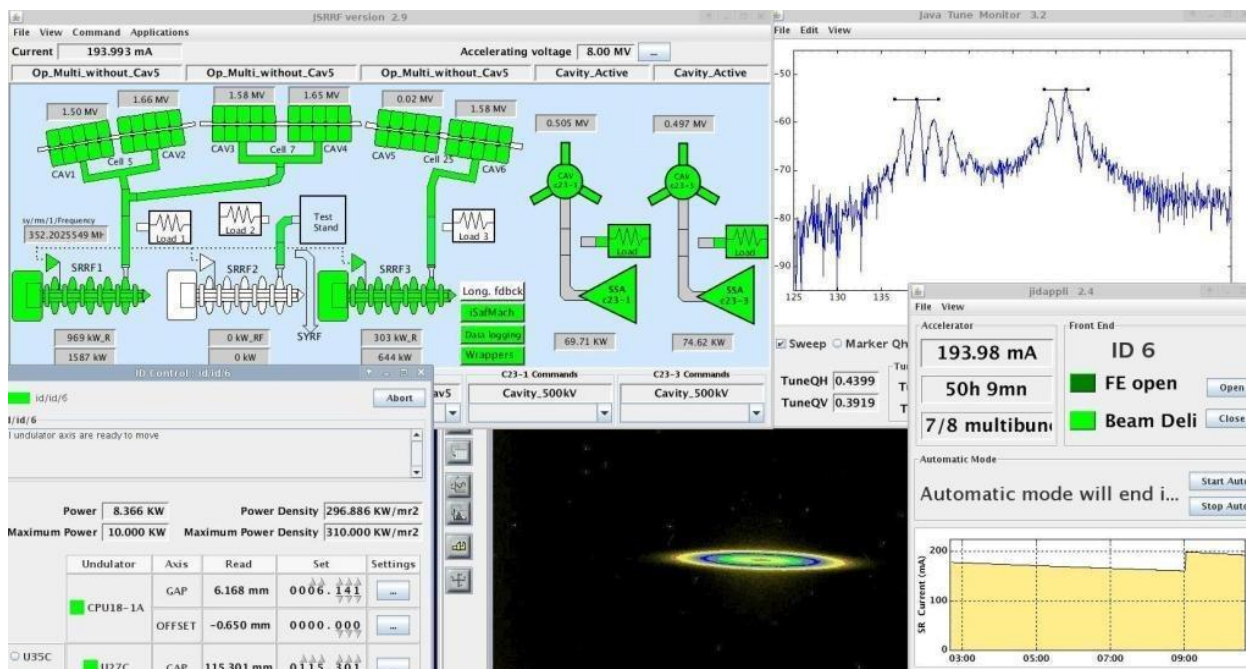
Tango distributed control system was elaborated at the ESRF, the european synchrotron radiation facility in Grenoble, France. The ESRF is a large research laboratory (1500 engineers, researchers), a kind of giant microscope to explore organic molecules or others using X rays of high energy (6 GeV) issued from an 850 meters circumference accelerator, in more than 40 beams lines operating simultaneously. See www.esrf.eu for more info.

Links to last year project [ZynqTangoGateway](#)



ESRF beam lines

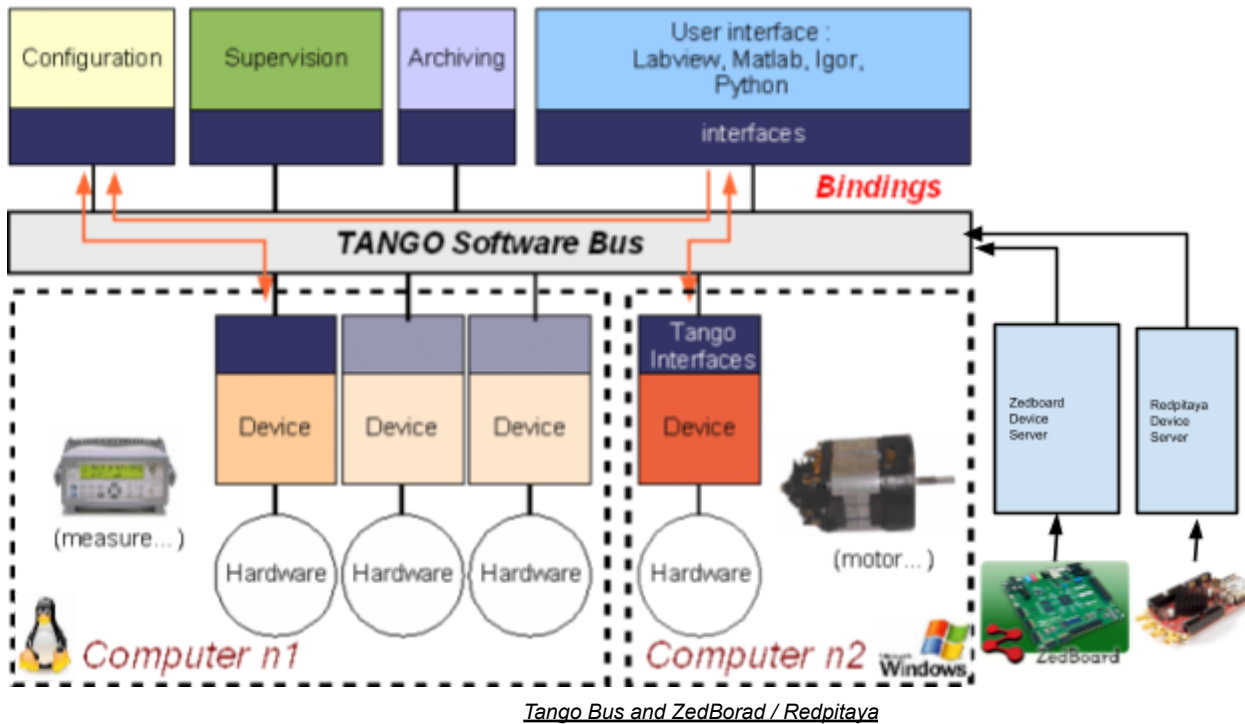
Tango control system is a 10 men / 10 years middleware piece of code, tools and paradigms, which aims at the 24/7 monitoring of more than 200 000 sensors & actuators in the facility. It features among other items : logging, events driven communication, seamless integration to NI Labview, Matlab, Igor scientific toolboxes or OPC-UA plc, EPICS bus, ... See www.tango-controls.org for info.



some Tango GUI and Tools

Tango is set of tools and protocol, with API in C++/Python/Java, free and open source, scalable from large synchrotron to small raspberry pi form factor embedded devices. It has many advantages over competition, either closed and expensive systems (ABB, SIEMENS, ...) or even its direct competitor in research labs EPICS. It uses an object oriented paradigm to hide device complexity and promotes design reuse.

There are already several hundreds Device Server classes, written in C++, Python or Java, ready to use for the most common devices used at ESRF, Soleil, Alba, Elletra, Desy, ... all free of charges and available... see : www.tango-controls.org/resources/device-classes/



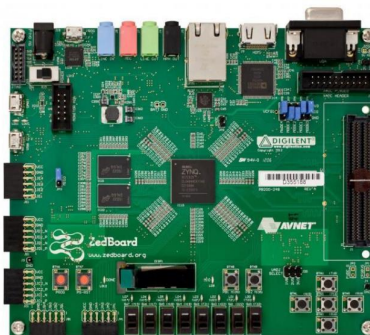
There is a vivid and growing community of professional software developers (15 synchrotrons or laser research facilities and a growing number of industrial company).

Tango team is proud to announce that the new SKA Telescope (square kilometer array telescope featuring antennas spread thoroughly South Africa and Australia) recently adopts Tango as its [control system framework](http://www.skatelescope.org). see www.skatelescope.org

Zedboard/Zynq uses case at the ESRF

Fpga are widely used at the ESRF for data high speed acquisition or very specific devices that can not be found COTS, as high precision motors controllers (IcePap), booster power controller, ... All devices, either in linac, booster, accelerator, or beam lines experimental hutches are intended to be monitored by Tango and so need their Device Server.

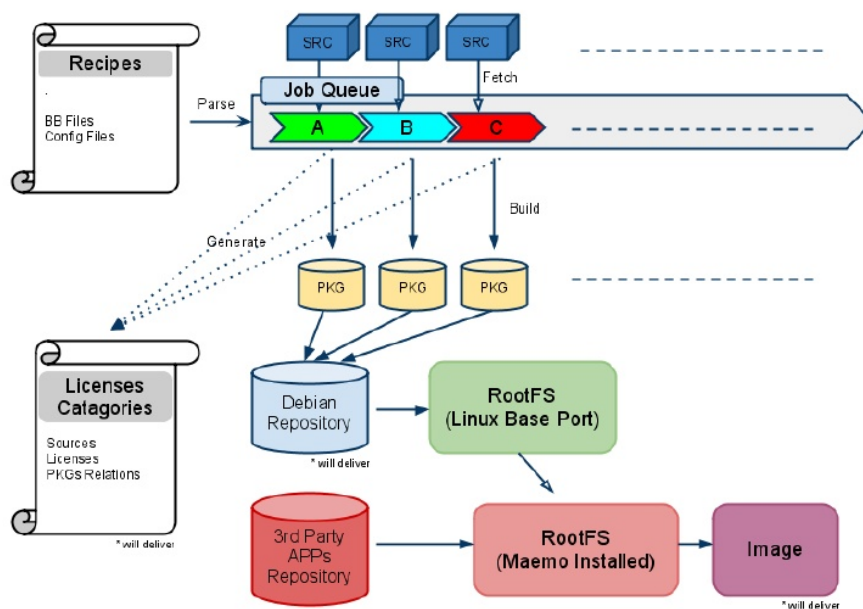
Xilinx Zynq meets perfectly these requirements, offering in one chip the best of 2 worlds : Fpga (for data acquisition) and the ARM cpu dual core hosting linux, and its high level/networking stack, ...



xilinx avnet ZedBoard

OpenEmbedded toolchain

We used Openembedded toolchain to build our dedicated Linux appliance. We find on GitHub all required bitbake recipes except a few : omniORB4, zeromq (old version) and of course Tango. But it was easy to write it from scratch because Tango already uses autoconf as its packaging tools.



OpenEmbedded toolchain

The TFL AmpereHourMeter

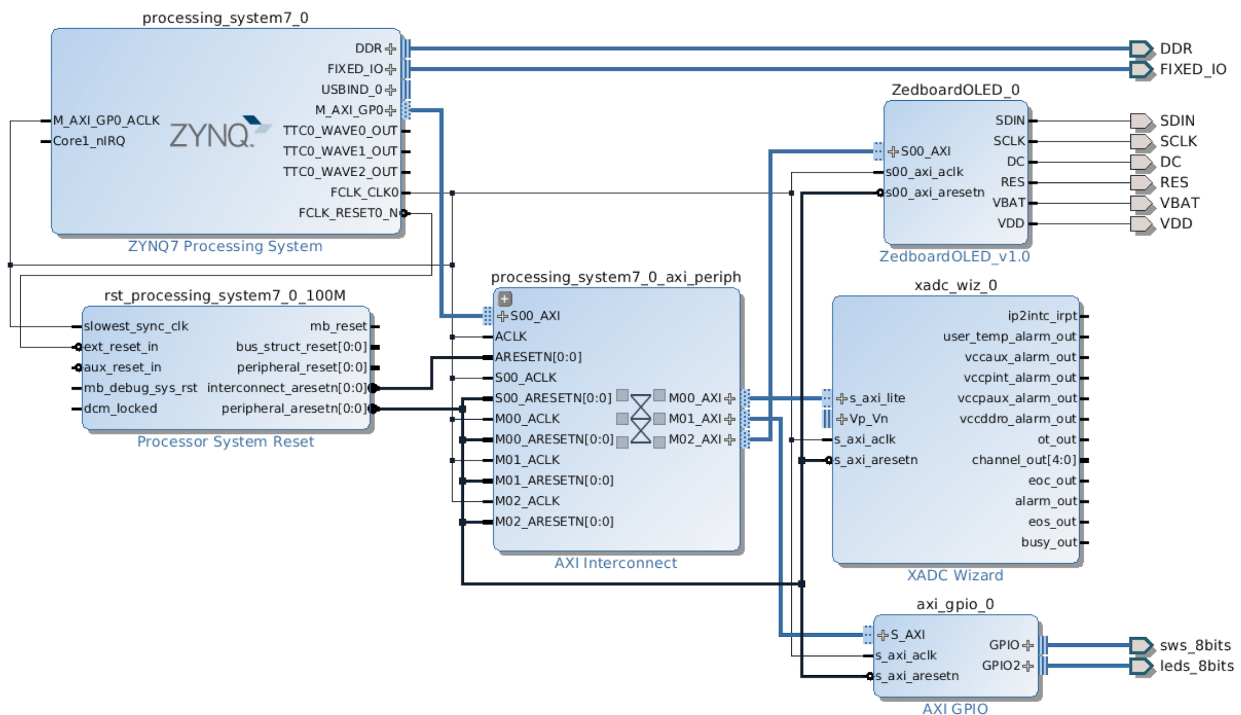
The ESRF Diagnostic Group is revamping its aging current monitoring system in TFL1 & TFL2 (Transfer Line 1 and 2, between Booster synchrotron and Storage Ring). Current is converted to a -10V 10V signal by a dedicated device, the PCT (Proportional Current Transformer). The new system should be integrated to the main ESRF's control system (TANGO DCS).

This Tango device server should monitor and integrate the current in the ESRF storage ring as done by a Dose Meter (high energy X-Ray) controller. For security reasons, the interlock system that ensure human safety (if this current grows too much, it will shutdown all the facility) can't rely on an underlying non deterministic linux and must be run as a standalone application, on a dedicated ARM cpu core.

The ESRF Electronic Group was investigating feasibility of a multi purpose Zynq SoC platform called DANCE for their devices. It was recently decided to use high performance Xilinx Kintex7 Fpga and QSeven SMB instead of Zynq SoC.

Hardware Design Vivado 2016.1

A reference design (tcl script) can be found [here](#) on github, featuring gpio for leds and switches, internal adc access, and oled display.



AXI_GPIO

an axi_gpio instance :

Gpio0 : Switches

Gpio1 : Leds

XADC

an internal xadc instance :

The CPU Temperature sensor is used for testing purpose. Any external signal could be use

OLED IP

We are using an IP from Qatar university. Thanks for sharing.

<https://github.com/ama142/ZedboardOLED-v1.0-IP>

(Driver needs some rewriting)

Asymmetric Multi Processor

As explain above, we have to go AMP for safety reasons, so that cpu1app could be certified by ASN (french nuclear security authority).

Many thanks to HenriChoi and his amazing [blog](#)

Device tree dts

```
remoteproc@1 {
    compatible = "xlnx,zynq_remoteproc";
    reg = < 0x1FE00000 0x200000 >;
    interrupt-parent = <&ps7_scugic_0>;
    interrupts = < 0 37 0 0 38 0 >;
    firmware = "cpu1app.elf";
    ipino = <0>; //The only free ipino
    vring0 = <0>; //MUST BE 0
    vring1 = <0>; //MUST BE ZERO
};
xadc@f8007100 {
    compatible = "xlnx,zynq-xadc-1.00.a", "xlnx,ps7-xadc-1.00.a"; //CAUTION !!! zynq-xadc
    reg = <0xf8007100 0x20>;
    interrupts = <0 7 4>;
    interrupt-parent = <&ps7_scugic_0>;
    clocks = <&clkc 12>;
};
remoteproc@1 {
    compatible = "xlnx,zynq_remoteproc";
    reg = < 0x1FE00000 0x200000 >;
    interrupt-parent = <&ps7_scugic_0>;
    interrupts = < 0 37 0 0 38 0 >;
    firmware = "cpu1app.elf";
    ipino = <0>; //The only free ipino
    vring0 = <0>;
    vring1 = <0>;
};
ps7_ocmc_0: ps7-ocmc@f800c000 {
    compatible = "xlnx,ps7-ocmc-1.00.a", "xlnx,zynq-ocmc-1.0"; //CAUTION !!! zynq-ocmc
    interrupt-parent = <&ps7_scugic_0>;
    interrupts = <0 3 4>;
    reg = <0xf800c000 0x1000>;
} ;
```

Bootargs

Maxcpus= if required CORE number could be select here
but not required as remoteproc will allocate/deallocate core automatically
mem=0x1FE00000

0000 0000 --- 1FE0 0000 linux core9
1FE0 0000--- 2000 0000 standalone core1

uEnv.txt

[Configuration file](#) for SD u-boot

Flatened Device tree blob address is set before standalone core1 memory area

Linux kernel options

select

- Industrial I/O (iio)
- remoteproc

A linux kernel .config file can be found [here](#)

Zynq_remoteproc kernel modules

Thanks HenryChoi...

Add a message queue with this [patch](#)

[Start](#) and [stop](#) scripts for standalone cpu1app

Check core status and usage in sysfs with :

```
root@zedboard-zynq7:~# cat /sys/devices/system/cpu/offline
1
root@zedboard-zynq7:~# cat /sys/devices/system/cpu/present
0-1
root@zedboard-zynq7:~# cat /sys/devices/system/cpu/online
0
```

Cpu1app

In Bsp configuration, choose standalone/core1

Needs an extra .rtable region entry in ELF

Source code [here](#) on github :

```
Disable MMU for OCM
XADC init
OLED init
Endless Loop:
    Xdac read
    Oled write
    OCM write
```

Boot.s

No more hack required

Ld.script

Set memory for core1

Add section .rtable

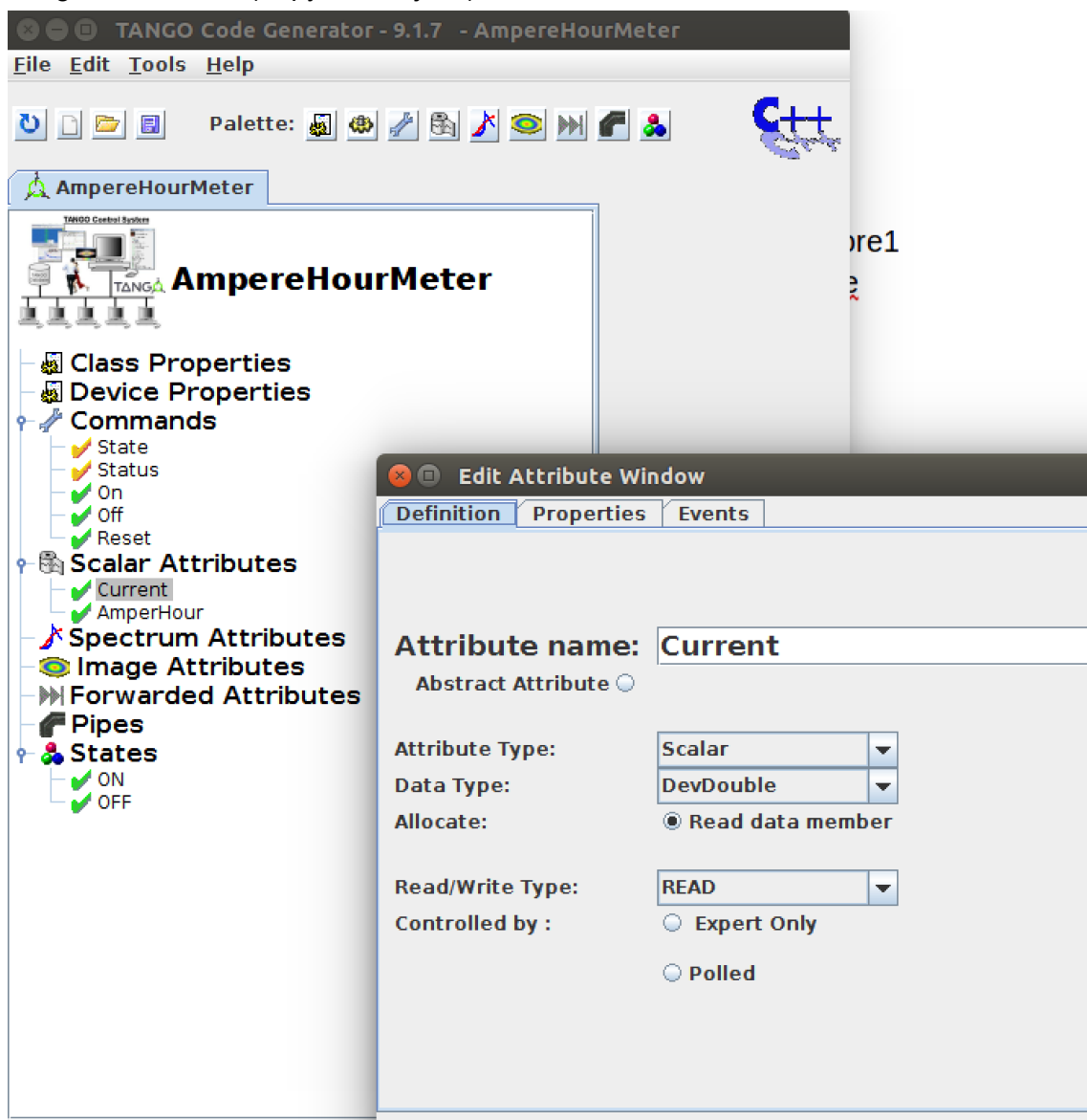
Code is [here](#)

Tango XADC Demo

There is a learning curve to master Tango DCS concepts and tool. But some graphic tools can be used without prior knowledge.

Pogo wizard tools

This GUI tool generates C++ (or python or java) source code skeleton.



Only 1 line of code to add manually in the skeleton to expose data on the network...

```
void AmpereHourMeter::read_Current(Tango::Attribute &attr)
{
    DEBUG_STREAM << "AmpereHourMeter::read_Current(Tango::Attribute &attr) entering... " << endl;
    /*----- PROTECTED REGION ID(AmpereHourMeter::read_Current) ENABLED START -----*/
    //    Set the attribute value
    *attr_Current_read=((float*)ocm);
    attr.set_value(attr_Current_read);

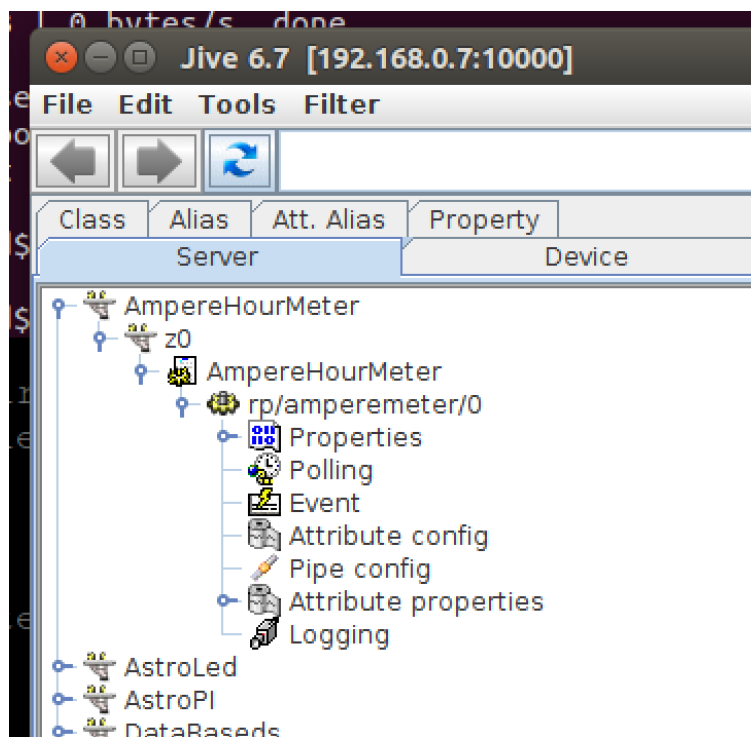
    /*----- PROTECTED REGION END -----*/      //    AmpereHourMeter::read_Current
}
```

Ubuntu tango linux vdi

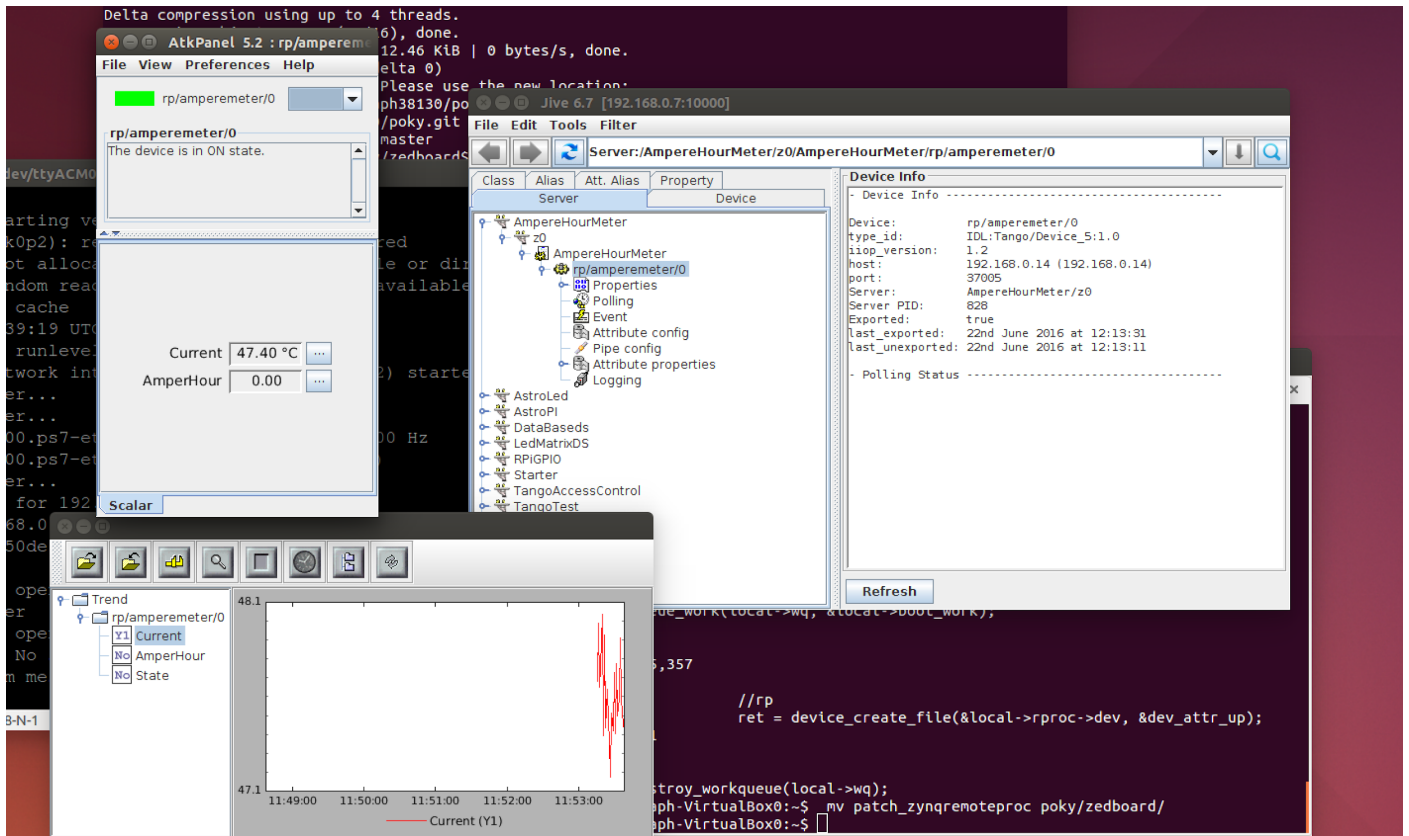
Download an ubuntu tango linux [here](#) from main Tango website

Caution :

- Virtualbox Network configuration : use bridged mode
- With jive, create DS wizard, add AmpereHourMeter Device server, instance z0, class AmpereHourMeter, device lgm/ampereometer/z0



Wizard is in Tools Menu, Click o- to open tree.



Device is ON (green), Monitoring Current (here cpu temp)

Zedboard SD Image

Fat partition

BOOT.bin
Amp.dtb
uEnv.txt
ulImage.bin

Ext4 partition

File system install with dd

Full image here

```
sudo dd bs=1M if=image.img of=/dev/sdc
```

```
gparted /dev/sdc // select Check, Apply
```

Starting Device Server on ZedBoard

Log into linux as root (no passwd)

Check network connectivity (pinging linuxbox)

```
/etc/hosts  
127.0.0.1 zedboard-zynq7
```

```
./start  
export TANGO_HOST=linuxbox:10000  
AmpereHourMeter z0&
```

Starting jive or jdraw synoptic on tango virtualbox

jive &

***Merci beaucoup à toute l'équipe de l'ESRF et notamment :
Jean Michel Chaize, Antonin Broquet, Andy Goetz
sans lesquels rien n'aurait été possible !***