

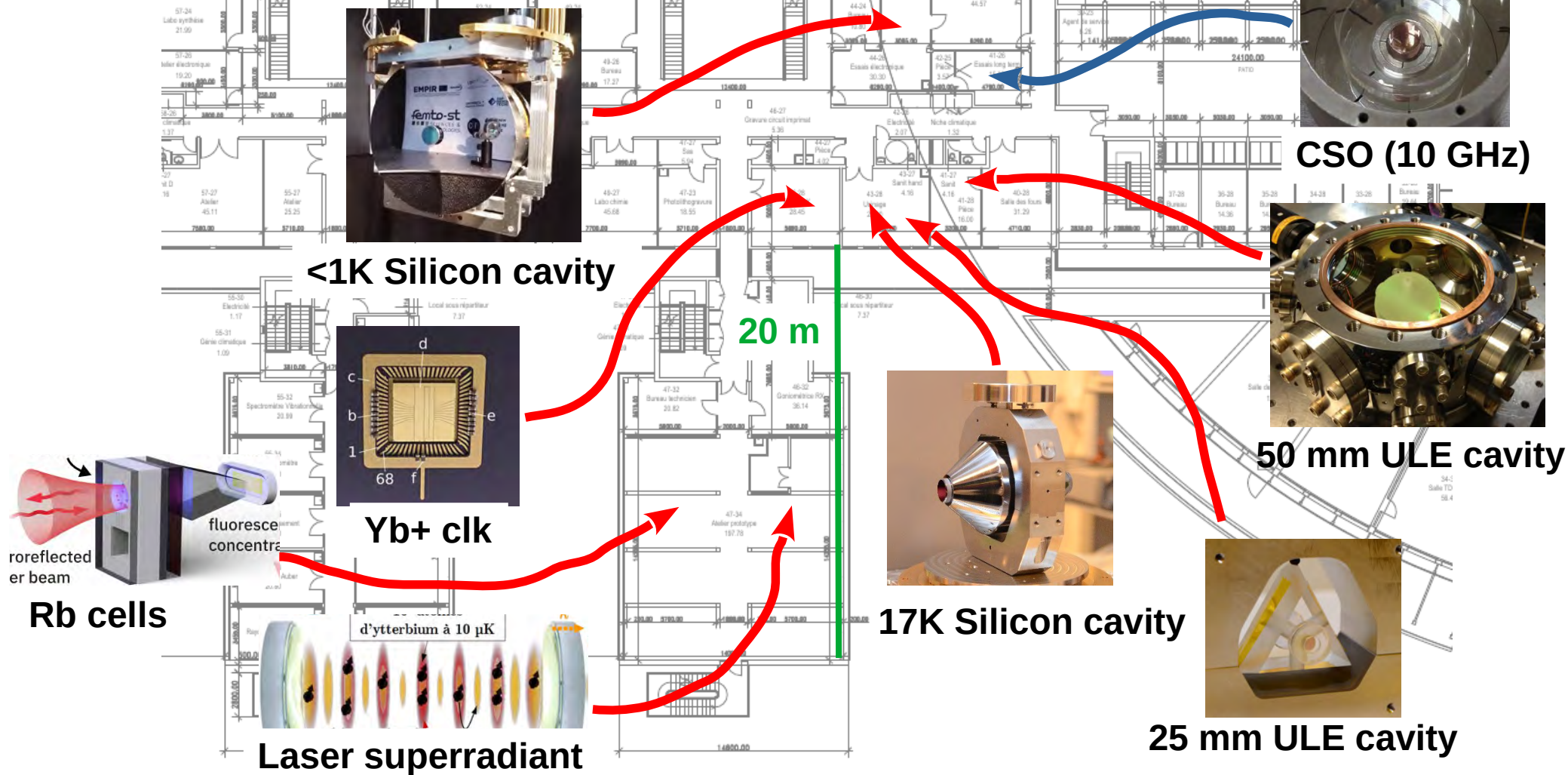
Local distribution of optical references based on digital electronic PLL

Martina Matusko, Shambo Mukherjee, Ivan Ryger, Séverine Denis, Gwenhaël Goavec-Merou, Jacques Millo, Remi Meyer, Clément Lacroûte, Émile Carry, Jean-Michel Friedt, Marion Delehay



AG Refimeve, LPL, le 03/10/2024

Local context

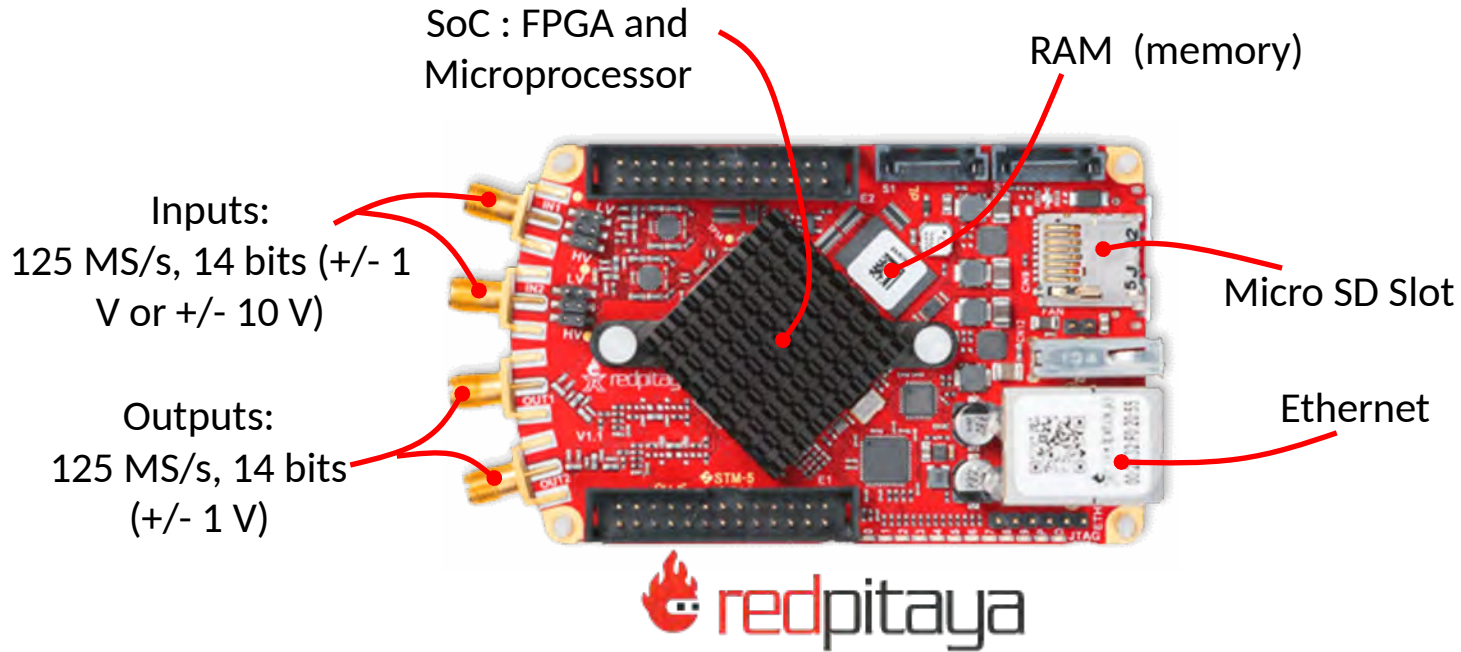


Outline

- Red Pitaya Platform (FPGA)
- Link stabilization with Red Pitaya
- Link stabilization with undersampling technic
- Others applications

Red Pitaya Platform

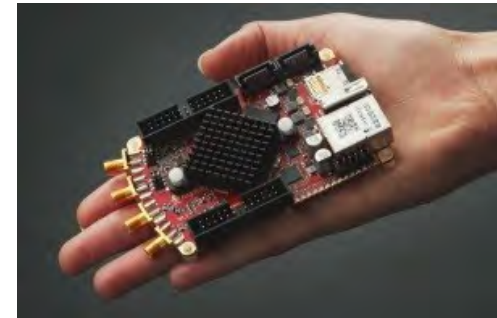
STEMLab 125-14



FPGA (Field-Programmable Gate Array):

- logic functions
- Multipliers (48 bits)
- Memory RAM
- DMA (CPU-FPGA data transfert)
- LookUp tables (LUT)

- CPU: Dual-Core ARM Cortex-A9 MPCore
- FPGA: Xilinx Zynq 7010



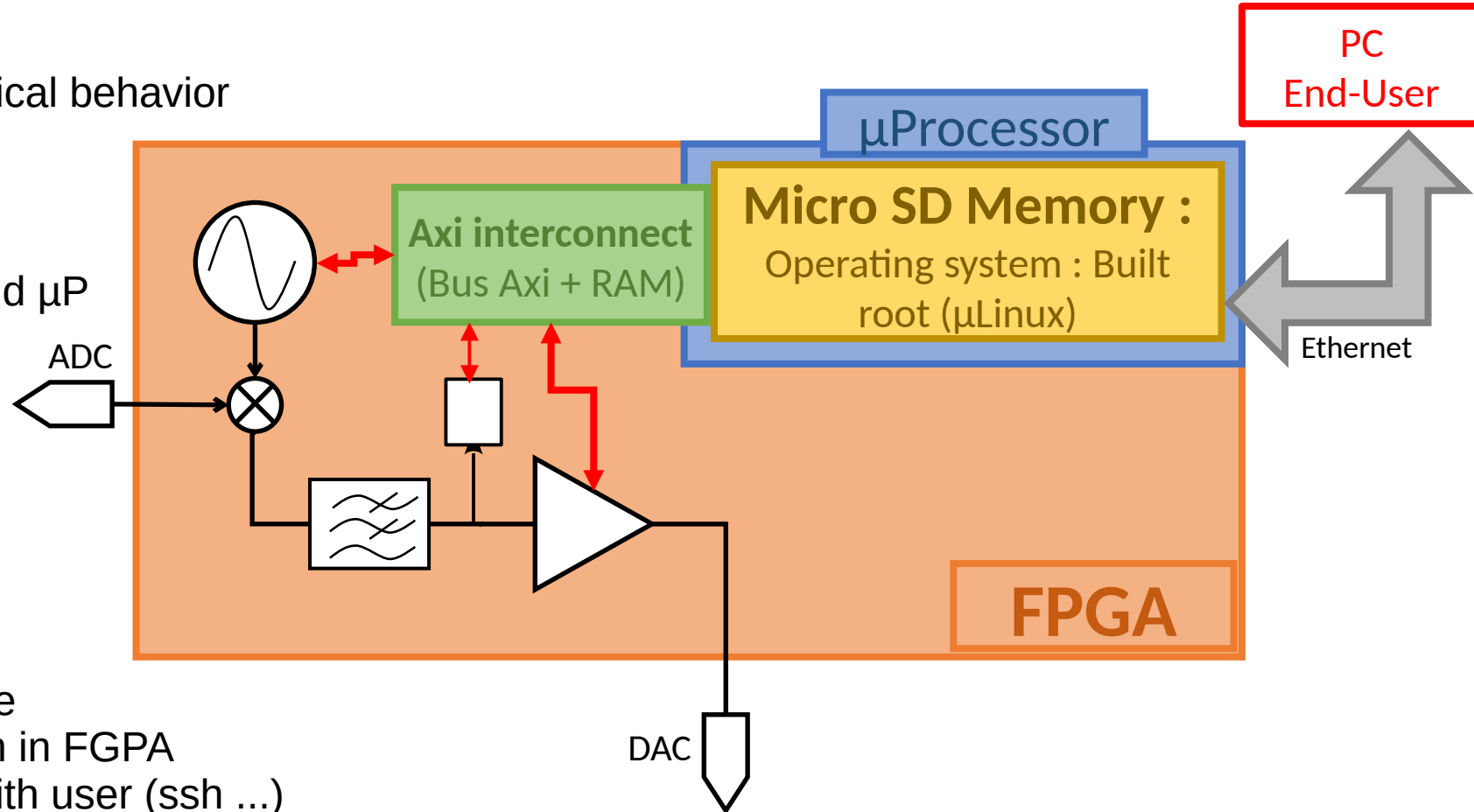
Red Pitaya Platform

overview

FPGA:
Configured for logical behavior

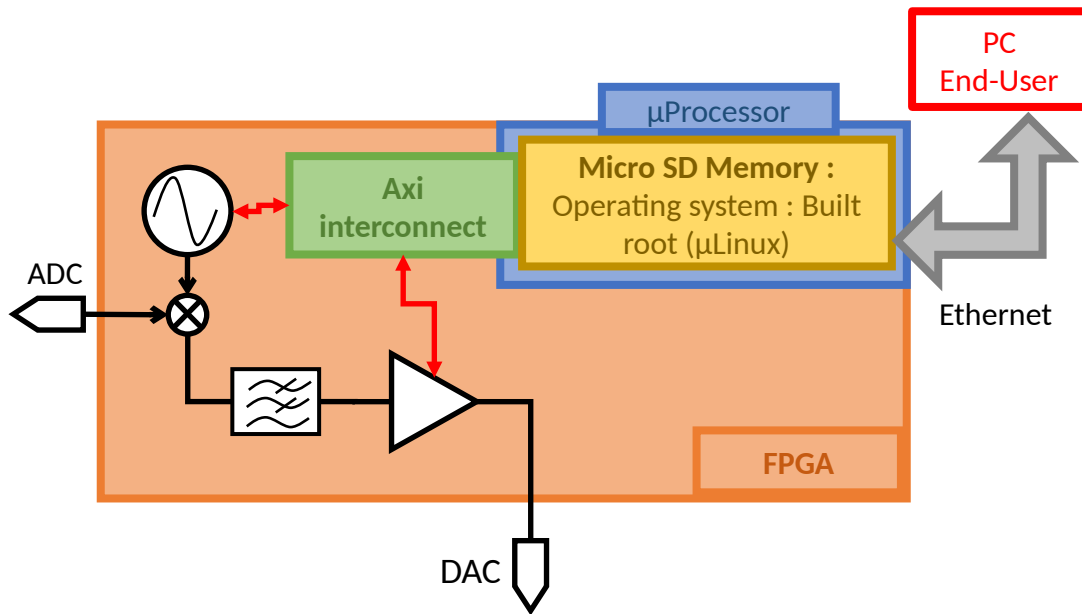
AXi interconnect:
Communication
between FPGA and μ P

μ Processor:
High level software
set/get information in FPGA
Communication with user (ssh ...)

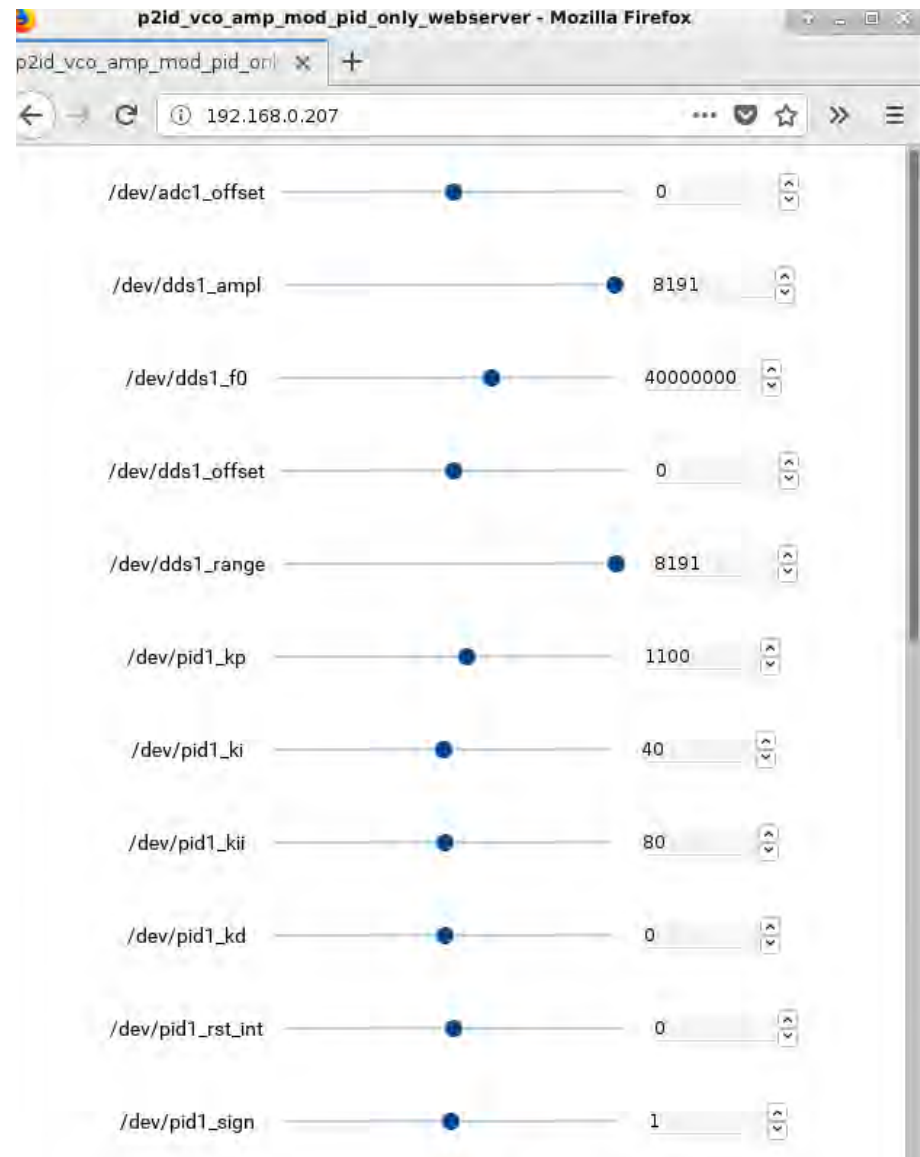


Red Pitaya Platform

User interface



```
liboscimp_fpga.shifter_set("/dev/shifterDyn_1", int(value))
```



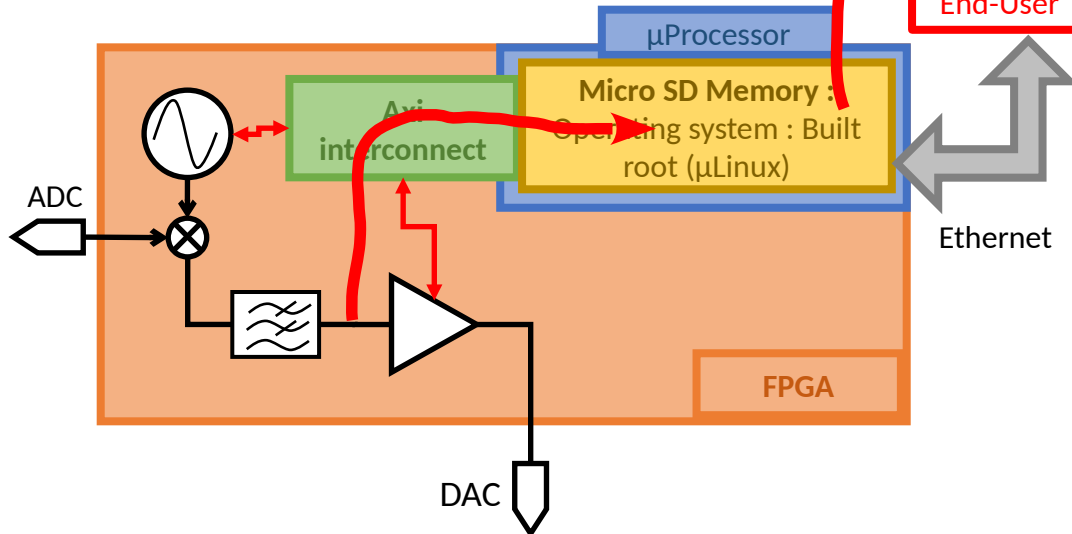
Red Pitaya Platform

Signal monitoring

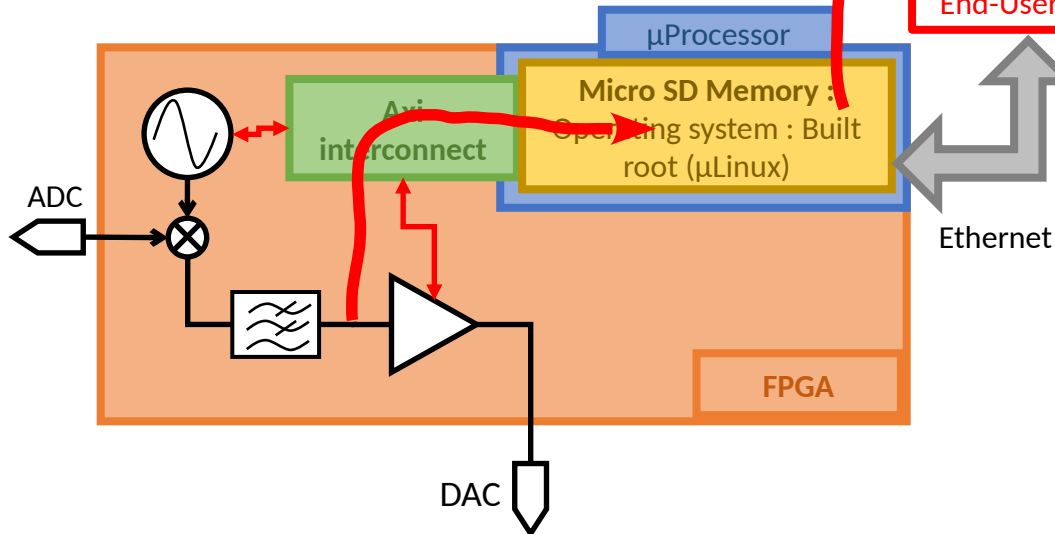
```
#!/usr/bin/env python
import zmq, time
context = zmq.Context()
sock = context.socket(zmq.PUB)
sock.bind("tcp://*:9901")
while True:
    time.sleep(0.05)
    with open('/dev/data_fast', 'rb') as f:
        sock.send(f.read(65536))
```

Server / publisher

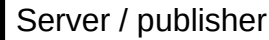
PC
End-User



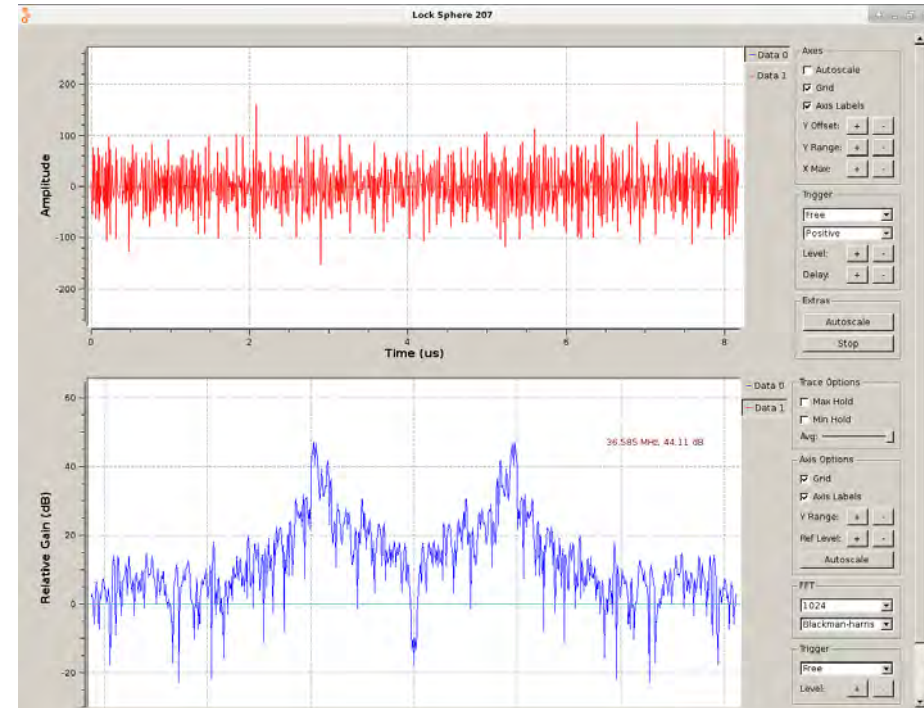
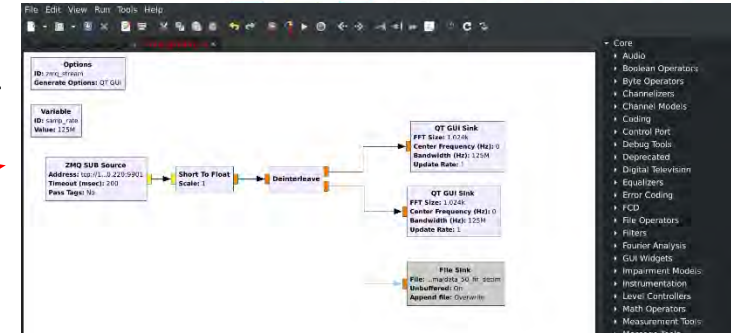
```
#!/usr/bin/env python
import zmq, time
context = zmq.Context()
sock = context.socket(zmq.PUB)
sock.bind("tcp://*:9901")
while True:
    time.sleep(0.05)
    with open('/dev/data_fast', 'rb') as f:
        sock.send(f.read(65536))
```



Client / Subscriber



Ethernet



Red Pitaya Platform

Software “suite”

<https://github.com/oscimp/oscimpDigital>

Open software

Available for few eval boards

Tutorial available

... but not so easy

GitHub - oscimp/oscimpDigital: OscillatorIMP ecosystem for the digital characterization of ultrastable oscillators and Software Defined Radio (SDR) frontend processing — Mozilla Firefox

Product Solutions Resources Open Source Enterprise Pricing Search or jump to... Sign in Sign up

oscimp / oscimpDigital Public

Code Issues 1 Pull requests Actions Projects Wiki Security Insights

master 1 Branch 0 Tags Go to file Code

File	Description	Commit	Time
app @ cba8475	* app 749c94c...cba8475 (1):	b607099	7 months ago
doc	add documentation according to Jack		3 months ago
fpga_ip @ a8477c6	* app 749c94c...cba8475 (1):		7 months ago
lib @ 8e66e54	corrected use of delayTempoReal_axi in liboscimp_fpga...		last year
linux_driver @ 3a483a6	* app 749c94c...cba8475 (1):		7 months ago
.gitignore	ignore: add vivado output directory		5 years ago
.gitmodules	add documentation submodule		7 months ago
LICENSE.md	Create LICENSE.md		4 years ago
README.md	documentation: tree structure		2 years ago
settings.sh.sample	Update settings.sh.sample		4 years ago

About

OscillatorIMP ecosystem for the digital characterization of ultrastable oscillators and Software Defined Radio (SDR) frontend processing

ecosystem fpga sdr buildroot redpitaya digital-signal-processing radio-frequency software-defined-radio linux-driver data-stream-processing plutossdr digital-characterization

Readme View license Activity Custom properties 48 stars 9 watching 6 forks

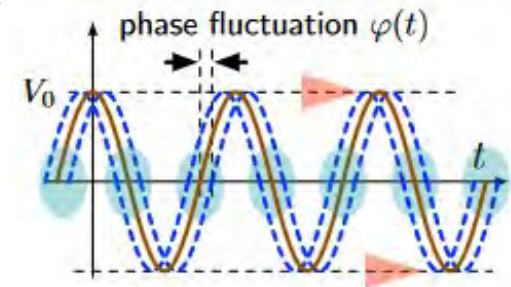
Outline

- Red Pitaya Platform (FPGA)
- Link stabilization with Red Pitaya
- Link stabilization with undersampling technic
- Others applications

femto-st
SCIENCE & TECHNOLOGIES

The diagram illustrates a fiber-based frequency stabilization system. A laser beam is split by an optical coupler (OC) into two paths. One path leads to a photodetector (PD1), and the other path leads to a fiber modulator (FM1). The beam from FM1 passes through an acousto-optic modulator (AOM) and another OC. The beam is then split again: one path goes through a fiber link (represented by red loops) to a third OC, which outputs to a user (User output) and a second fiber modulator (FM2). The other path from the third OC goes through an OC to a photodetector (PD2), which is connected to a frequency comb (FC/PNA). The output of PD1 is mixed with a reference frequency f_{demod} in a mixer (represented by a circle with an X). The resulting signal is filtered (represented by a trapezoid) and then processed by a proportional-integral (PI) controller (represented by a triangle). The output of the PI controller is a frequency f_{out} , which is fed back into the AOM. A curved arrow labeled "Thermal & mechanical perturbations" points from the fiber link to the AOM. An inset graph shows "phase fluctuation" with a sinusoidal wave V_0 and a dashed line indicating the fluctuation.

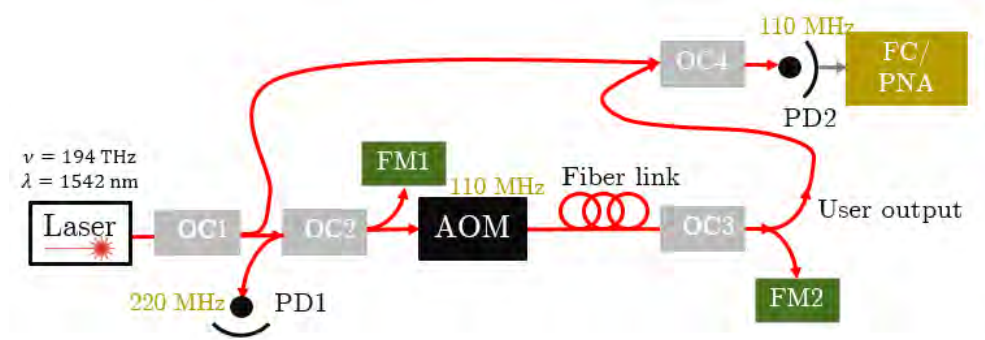
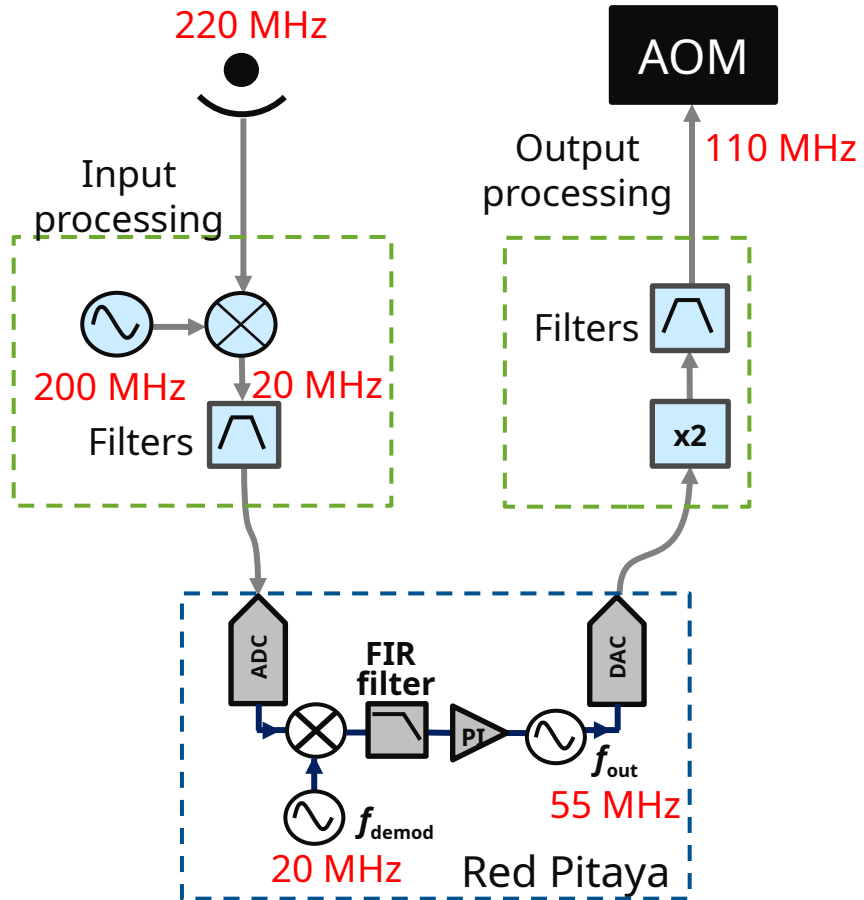
OC: optical coupler
FM: Faraday mirror
AOM: acousto-optic modulator
PD: photodiode
FC: frequency counter
PNA: phase-noise analyzer
ADC: analog-to-digital converter
FIR: finite impulse response
PI: proportional-integral control
DAC: digital-to-analog converter



<https://rubiola.org>

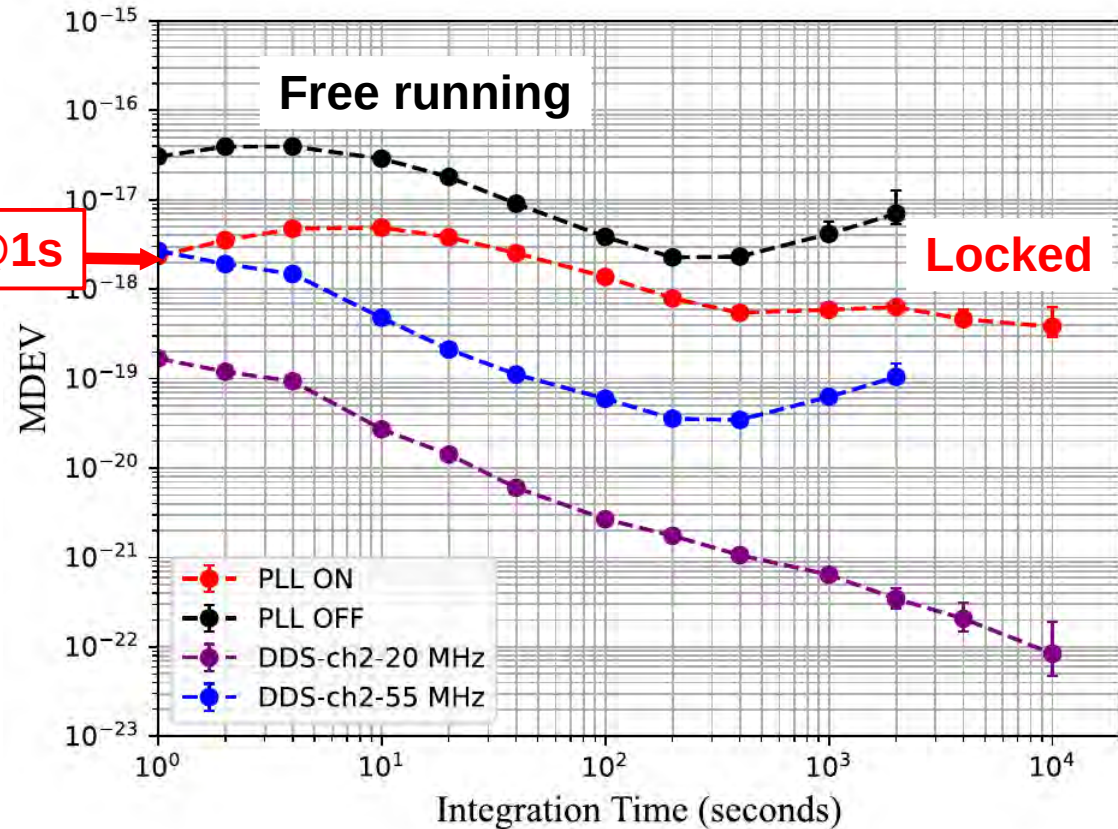
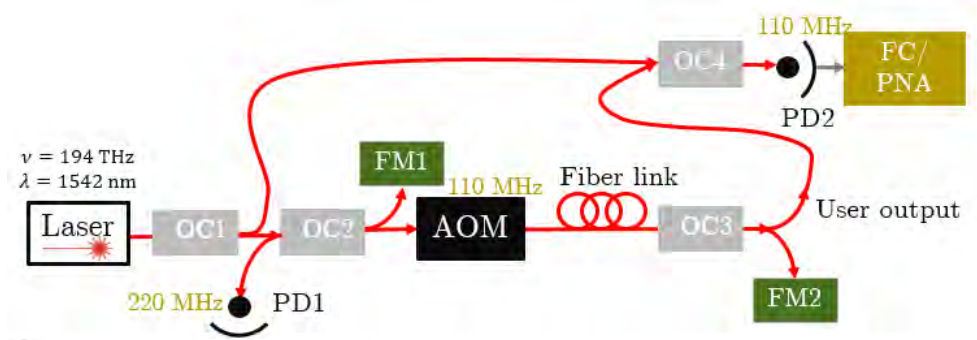
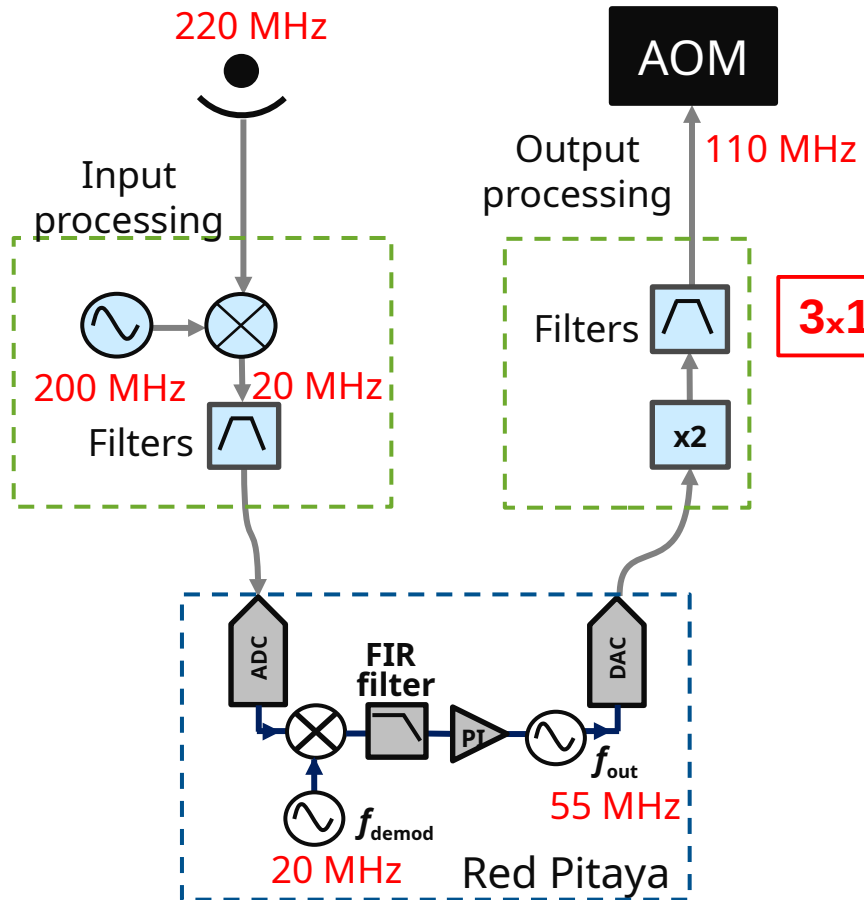
Fiber link

Red Pitaya



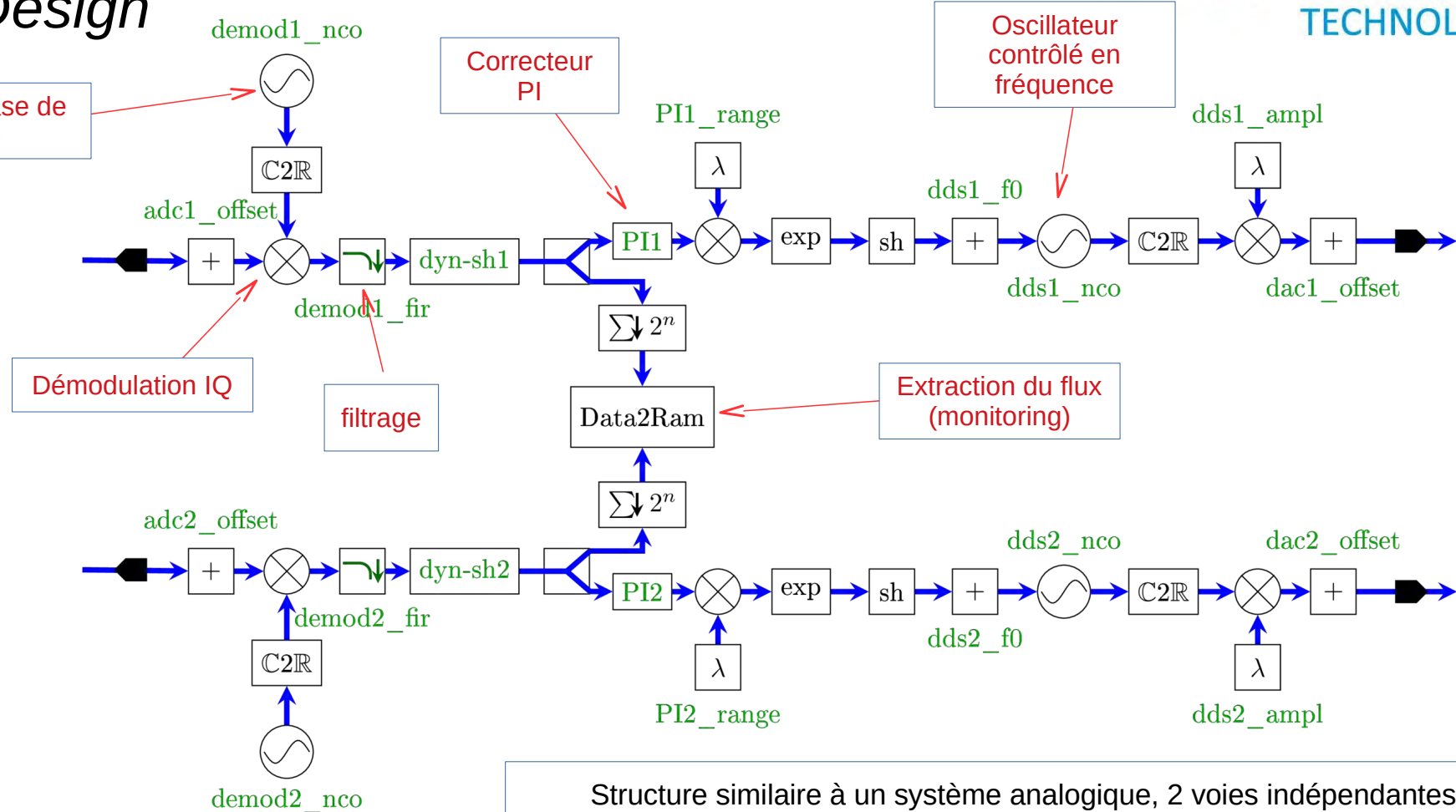
Fiber link

Red Pitaya



Fiber link

FPGA Design

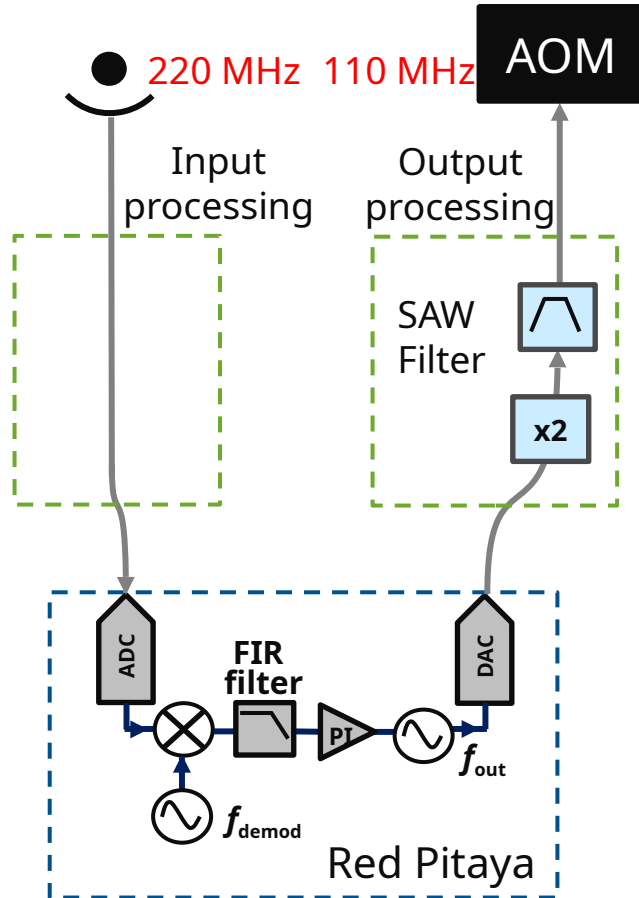


Outline

- Red Pitaya Platform (FPGA)
- Link stabilization with Red Pitaya
- Link stabilization with undersampling technic
- Others applications

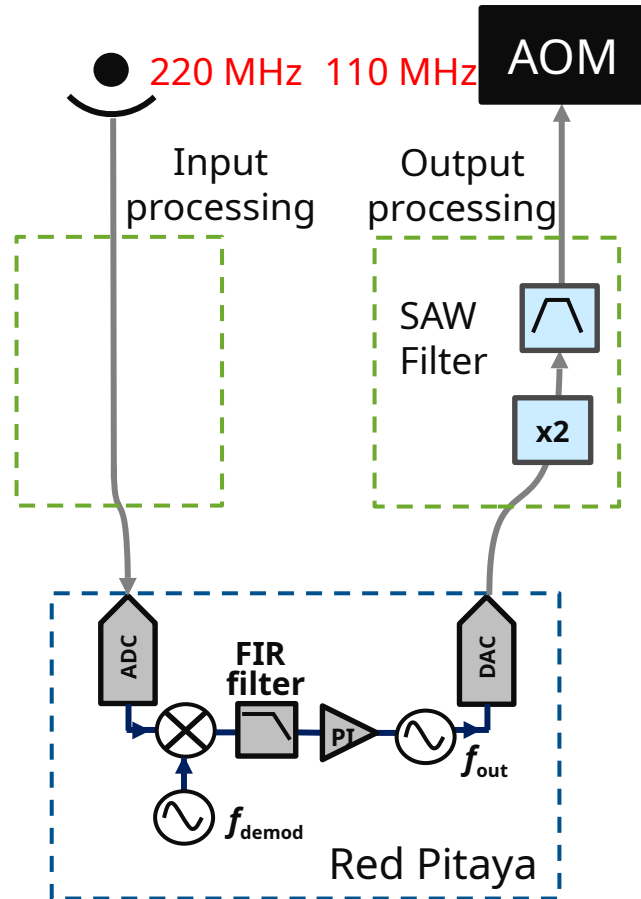
Undersampling - 1

STEMlab 125-14

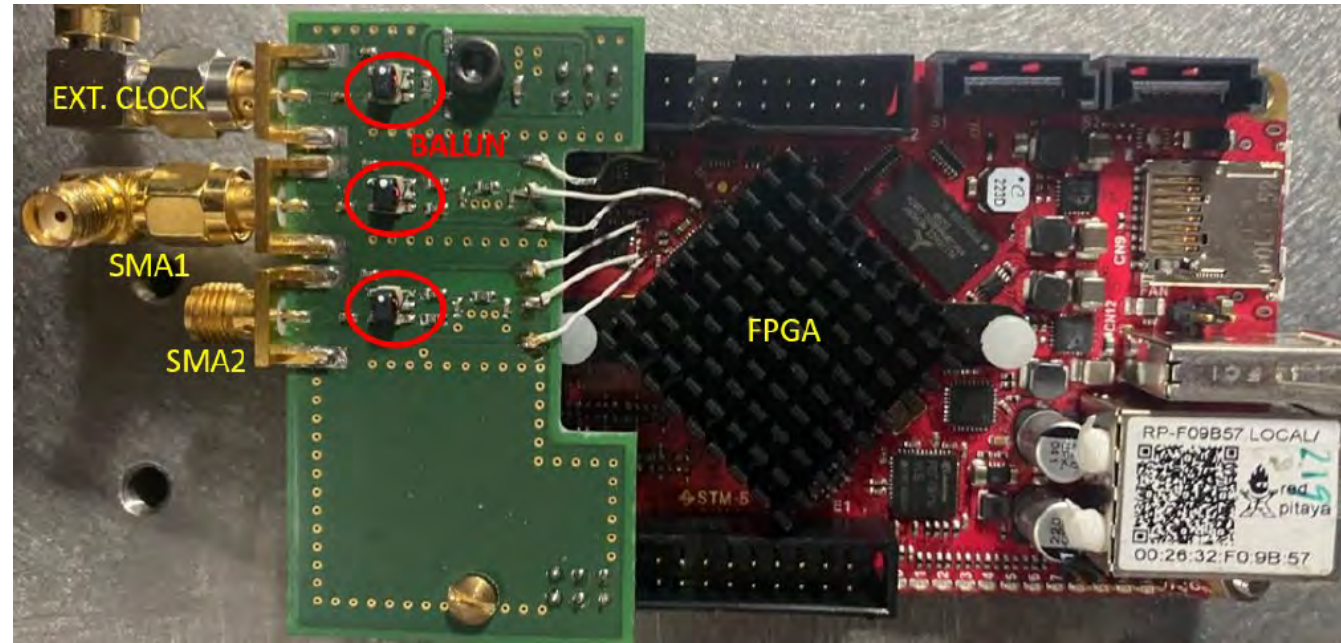


Undersampling - 1

STEMlab 125-14



Bypass anti-aliasing filters
Connect external clock
« extension board » (balun / differential input)



Fiber link

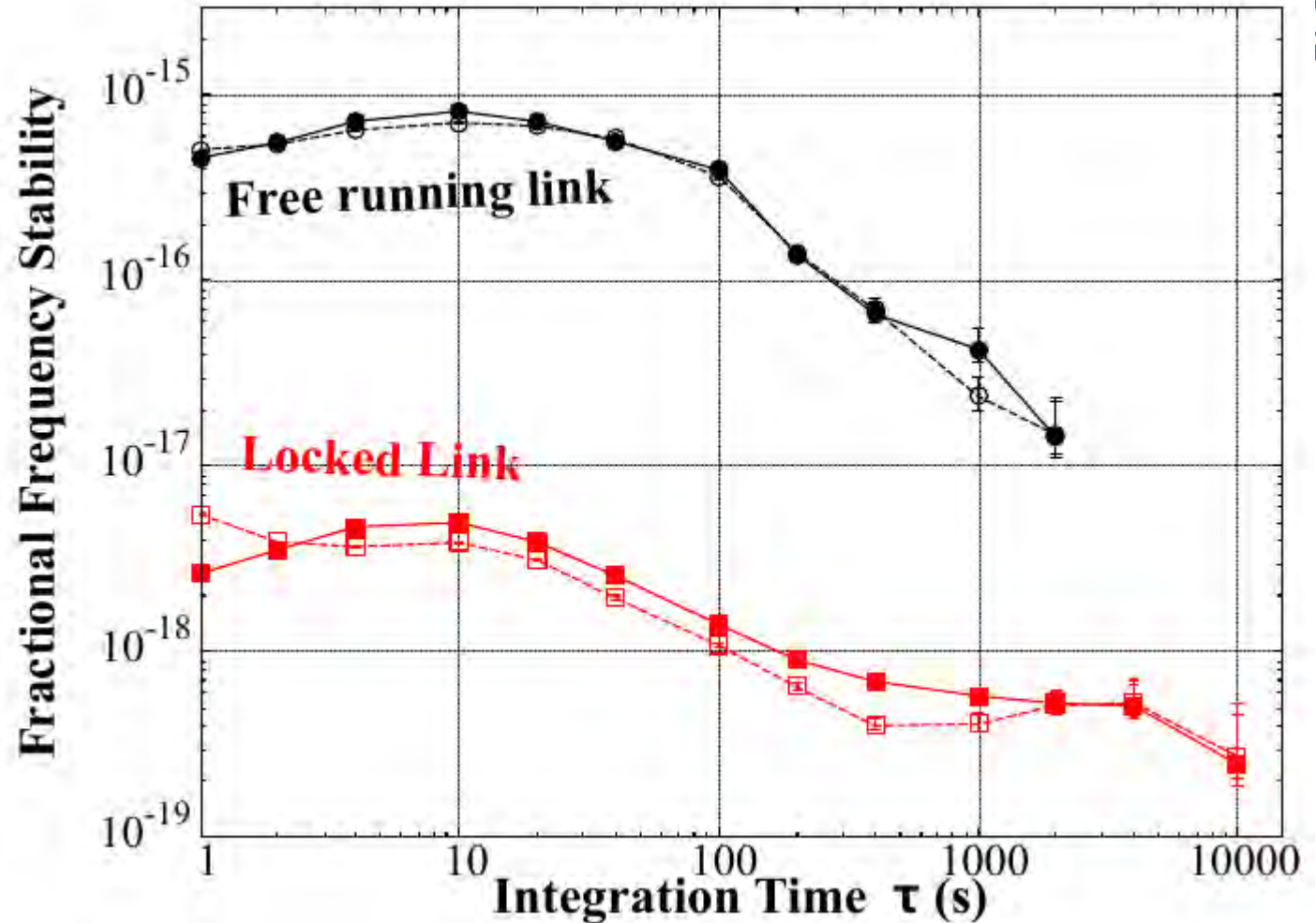
Performances

□ Undersampling

$4 \times 10^{-18} @ 1s$

■ Red Pitaya (regular)

$3 \times 10^{-18} @ 1s$



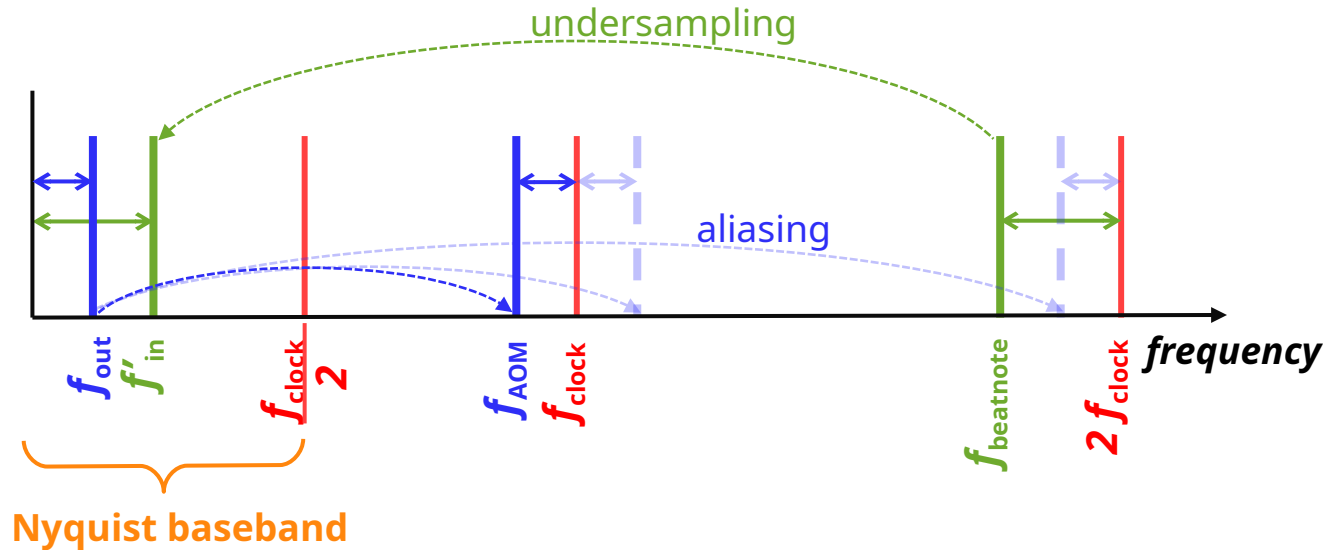
Undersampling

Principle

- signal at ADC
- signal at DAC

Nyquist: $f'_{in} = |f_{in} - n \times f_{clock}|, 0 < f'_{in} < f_{clock}/2$

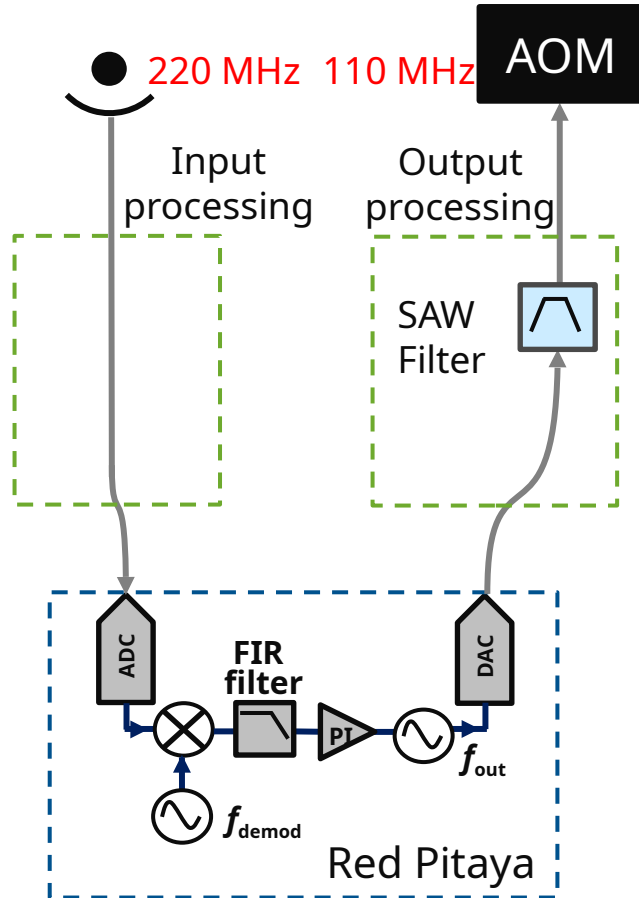
Aliasing: $n \times f_{clock} \pm f_{out}$



Undersampling - 2

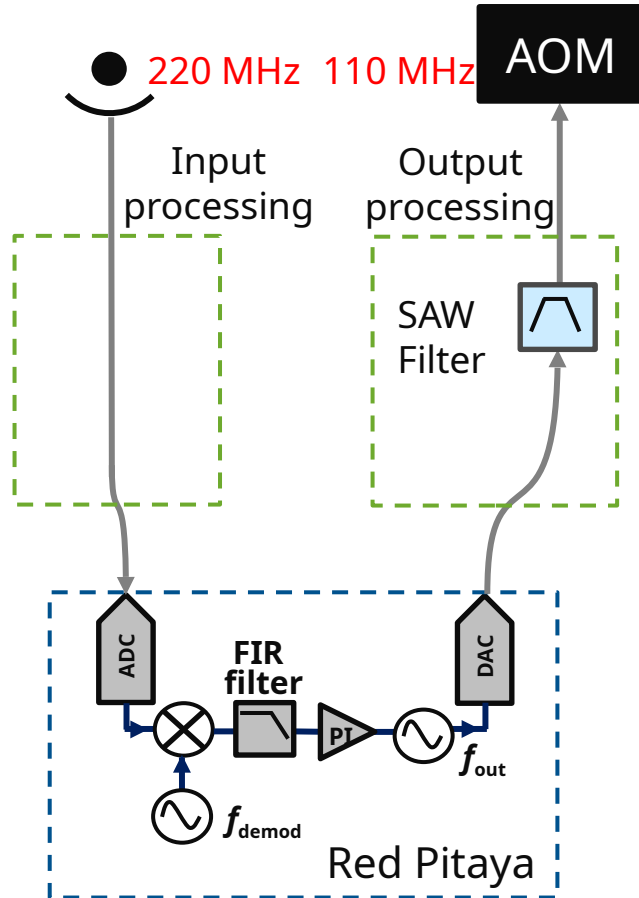
SDRlab 122-16

No anti-aliasing filters
16 bit ADC
14 bit DAC
Clock at 122.88 MHz



Undersampling - 2

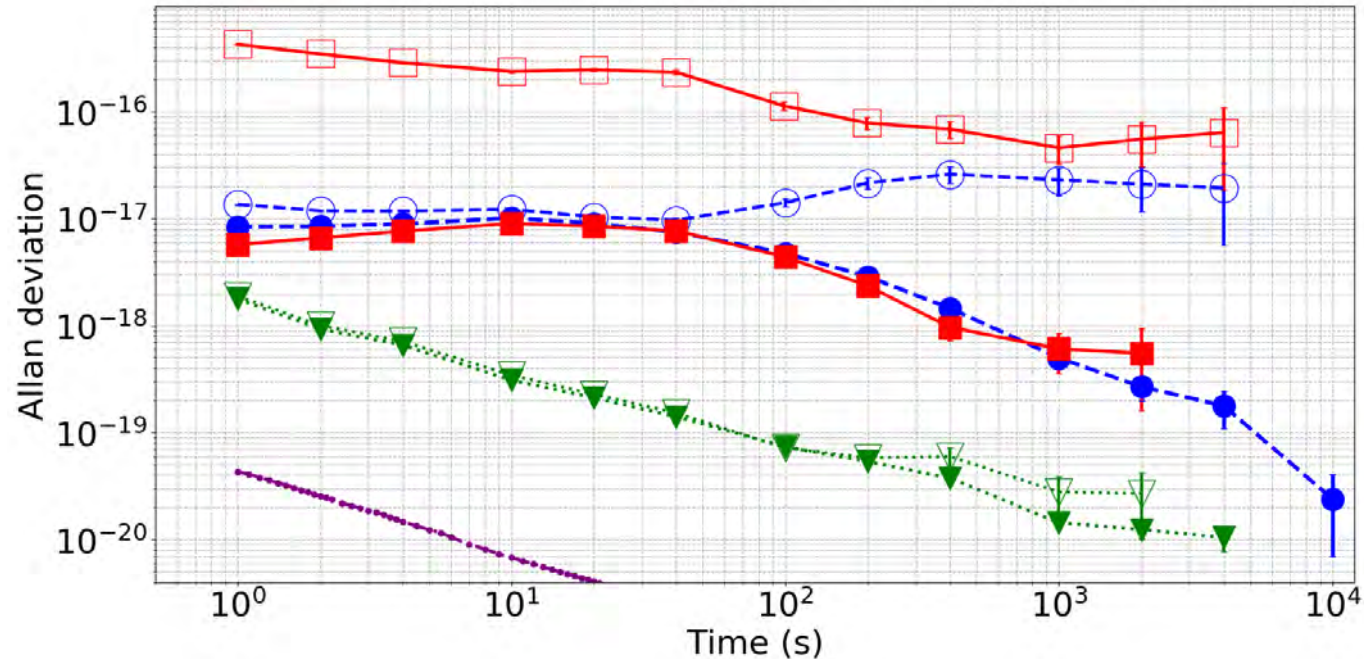
SDRlab 122-16



No anti-aliasing filters
16 bit ADC
14 bit DAC
Clock at 122.88 MHz

open / closed loop

- : with 90 m link
- : with 1 m link
- ▽ : no physical system
- : hydrogen maser

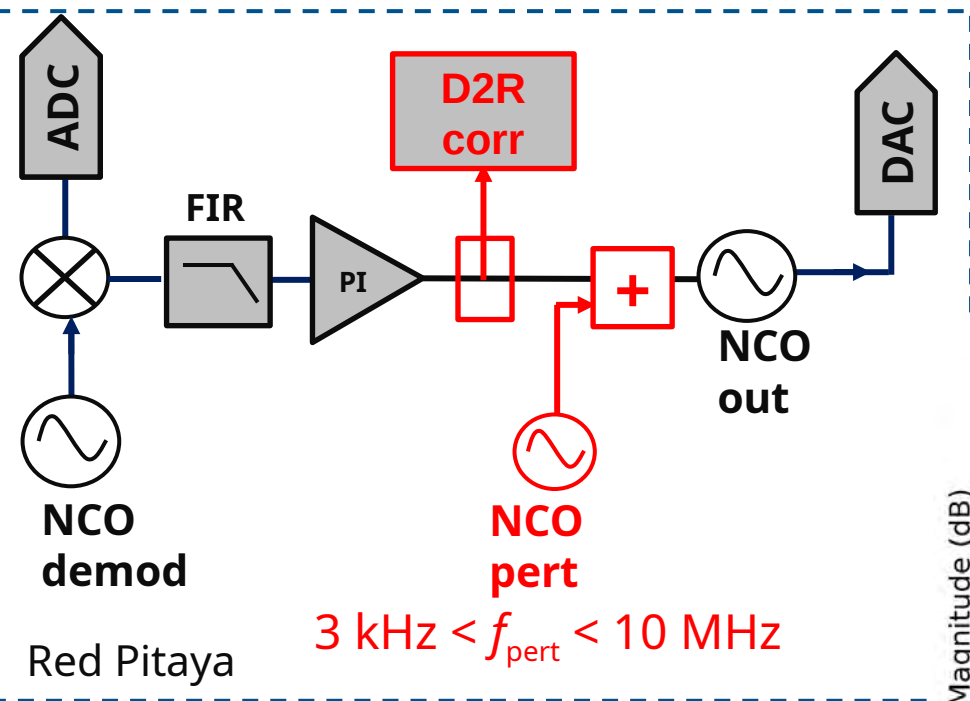


Outline

- Red Pitaya Platform (FPGA)
- Link stabilization with Red Pitaya
- Link stabilization with undersampling technic
- Others applications

PLL characterization

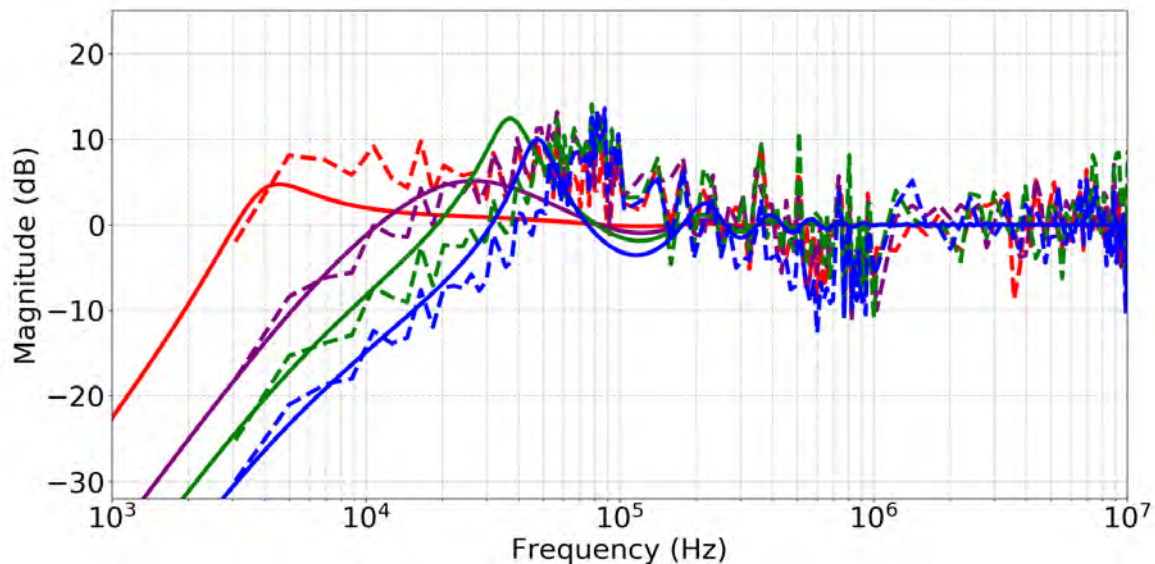
Tune PI gain



Red Pitaya adapted design

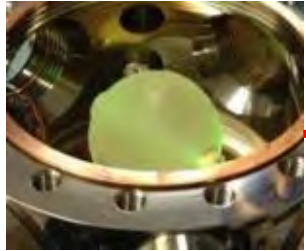
D2R: data-to-ram acquisition, NCO: numerically cor
oscillator. FIR: finite impulse response filter.

Red curves: lowest PI gains,
purple curves: 5×red-,
green curves: 10×red-,
blue: 20×red-curve PI gains

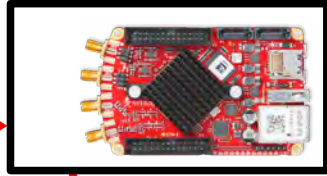


Application à la suppression de dérive de cavité

Laser stabilisé



PLL optique

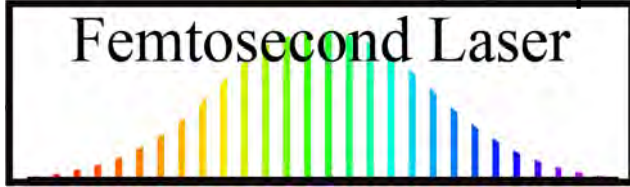


Lien optique

Vers Paris

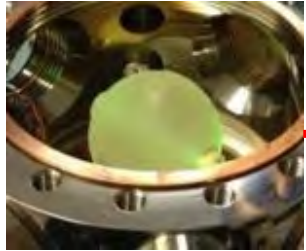
Stabilisation taux de rep.

Femtosecond Laser

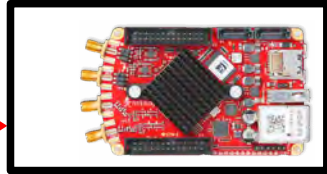


Application à la suppression de dérive de cavité

Laser stabilisé



PLL optique

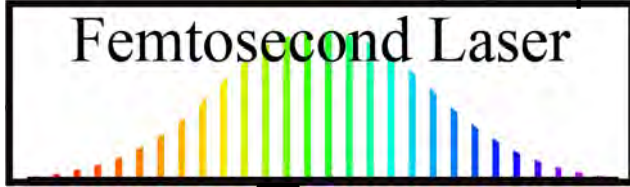


Lien optique

Vers Paris

Stabilisation taux de rep.

Femtosecond Laser



250MHz

H-Maser

150MHz

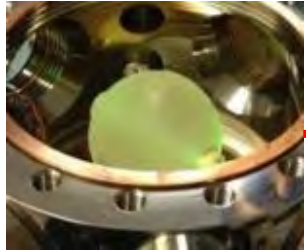
100MHz

Ethernet

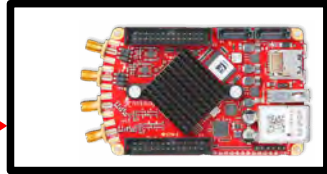
Sous échantillonnage
Démodulation
Correcteur PI

Application à la suppression de dérive de cavité

Laser stabilisé



PLL optique



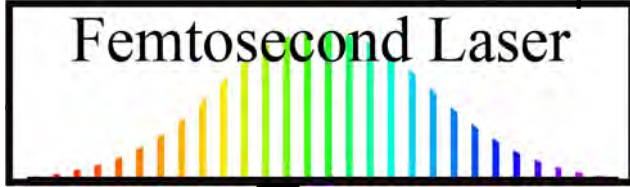
Corrections
de phase

Lien optique

Vers Paris

Stabilisation taux de rep.

Femtosecond Laser



250MHz

H-Maser

150MHz

100MHz

Ethernet

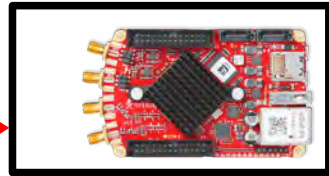
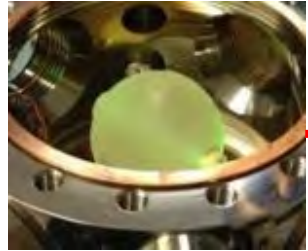
Sous échantillonnage
Démodulation
Correcteur PI

Zero MQ protocol

Application à la suppression de dérive de cavité

Laser stabilisé

PLL optique



Corrections
de phase

Lien optique

Vers Paris

Stabilisation taux de rep.

Femtosecond Laser



250MHz

H-Maser

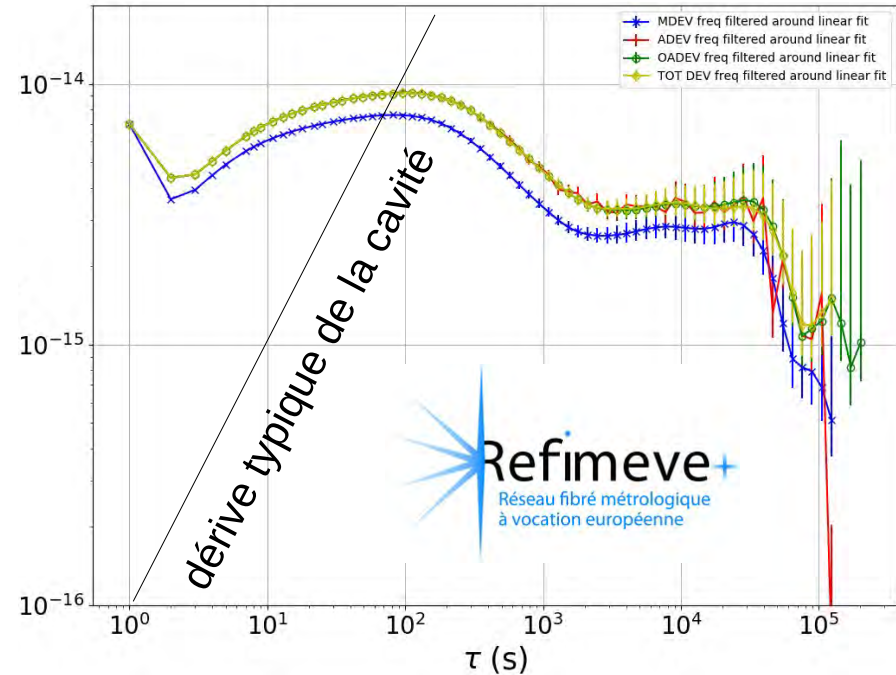
150MHz

Ethernet

Sous échantillonnage
Démodulation
Correcteur PI

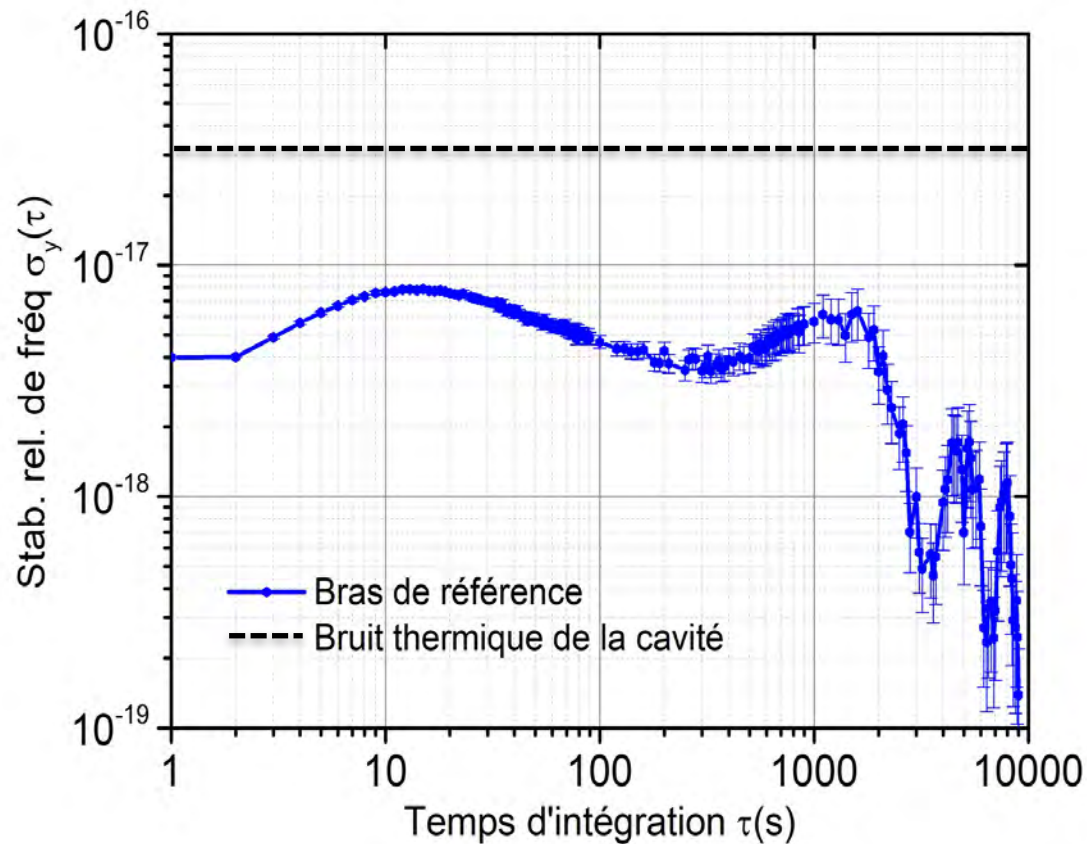
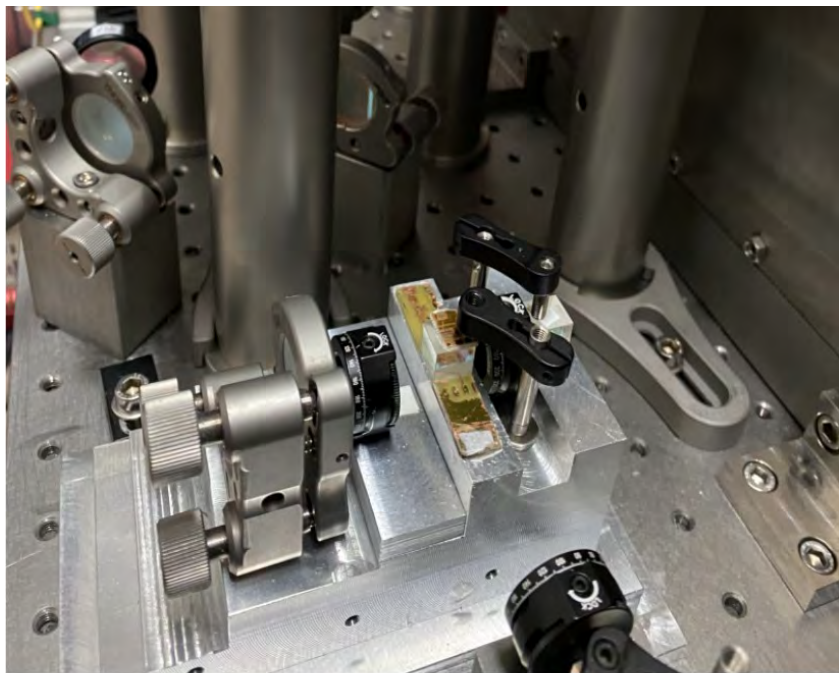
Zero MQ protocol

Comparaison SYRTE – FEMTO-ST



Free space ref arm

Characterization

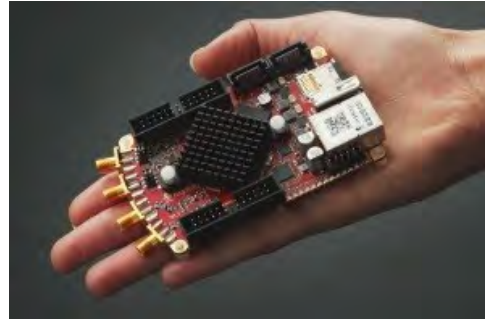
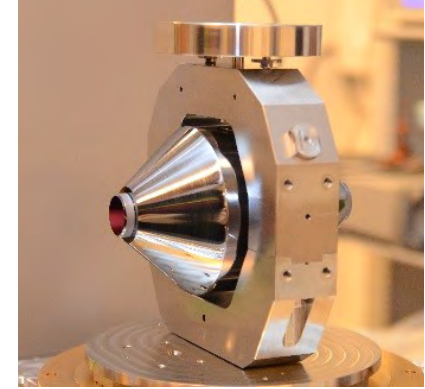


Conclusions

Link stabilization below 10^{-17} ==> compatible with our needs (silicon cryo cavity)

Software « suite » open : *oscimpDigital* in *github*

FPGA also used for PDH lock, Phase detection, RAM stabilization, apply sequences in CPT clock ...



-
- E. Cantin et al., New Journal of Physics 23, 053027 (2021).
K. Predehl et al., Science 336, 441 – 444 (2012).
N. R. Newbury et al., Opt. Lett. 32, 3056–3058 (2007).
A. C. Cárdenas Olaya et al, (EFTF) (2016) pp. 1–4.
A. Tourigny-Plante et al, RSI 89 (2018), 10.1063/1.5039344.
S. Mukherjee et al, IEEE Trans Ultrason Ferroelectr Freq Control. 69, 878–885 (2022).