Libera

New trends in beam diagnostics (present and future platforms with Zynq SoC)

Peter Leban on behalf of I-Tech, DEELS, June 2016, Hamburg

Content

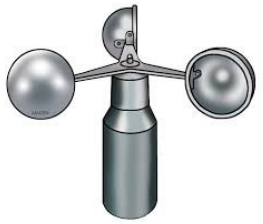
Peter's project

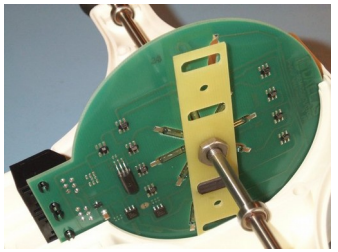
Company's projects

Weather Station Project
Step 1: Do It Yourself!

Bill of material

- Temperature sensor
- Humidity sensor
- Pressure sensor
- Rain gauge
- Wind sensor
- Fruit instrument





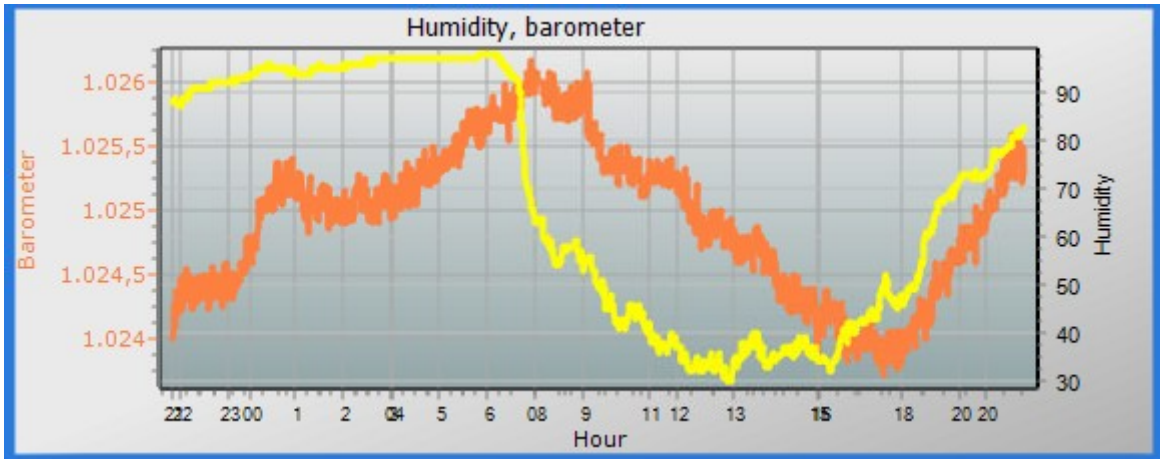
~100 – 200 EUR

Other

- Time
- A little knowledge

0 EUR, this is my hobby!

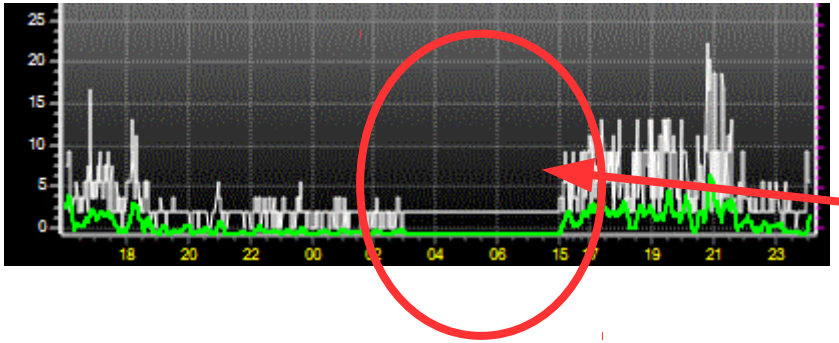
Hey, it works!



What next?

- I need to protect my temperature sensor from direct sun
- Would be nice to plot the data easier than parsing the files and using Excel or Matlab
- How to avoid occasional sensor communication hickups?
- I must make my weather data available online!

It works, but...



Communication link lost!

- Bought another temperature sensor...
- Change from fruit instrument to laptop for easier work
- Still many problems with link, overheating the sensor, wetting the sensor, etc.
- Reading the forums, checking other options, ...

I just don’t have enough spare time to do it!

Step 2. Buy the hardware.

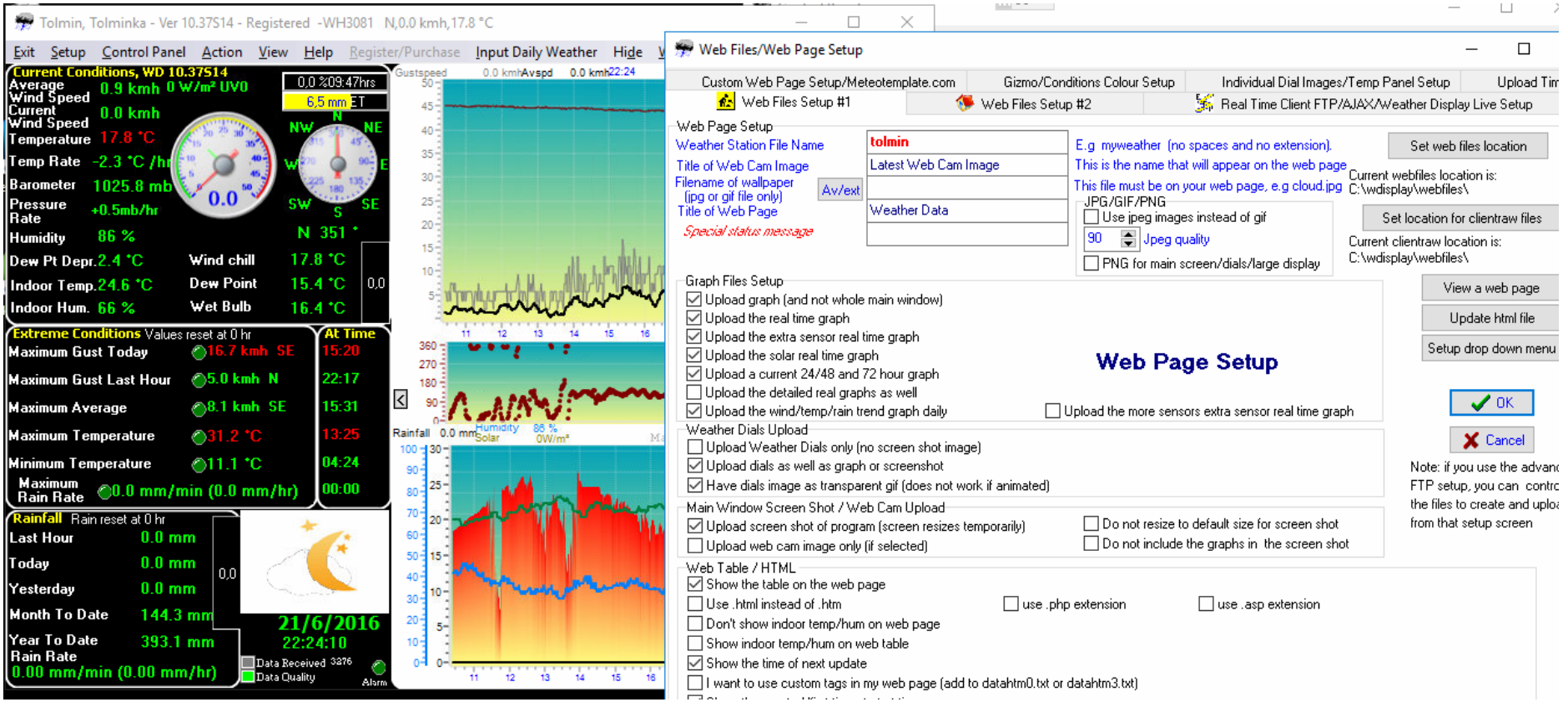
- I can actually buy a station for 100 EUR! Interesting... and it’s wireless :-)
- While I wait for delivery, I can build a little weather house. My father has some experience and can help me.

Nice... it works great.

But I don’t like the software. It’s so not logical and I can not upload to internet.

No problem. Many alternative options available.

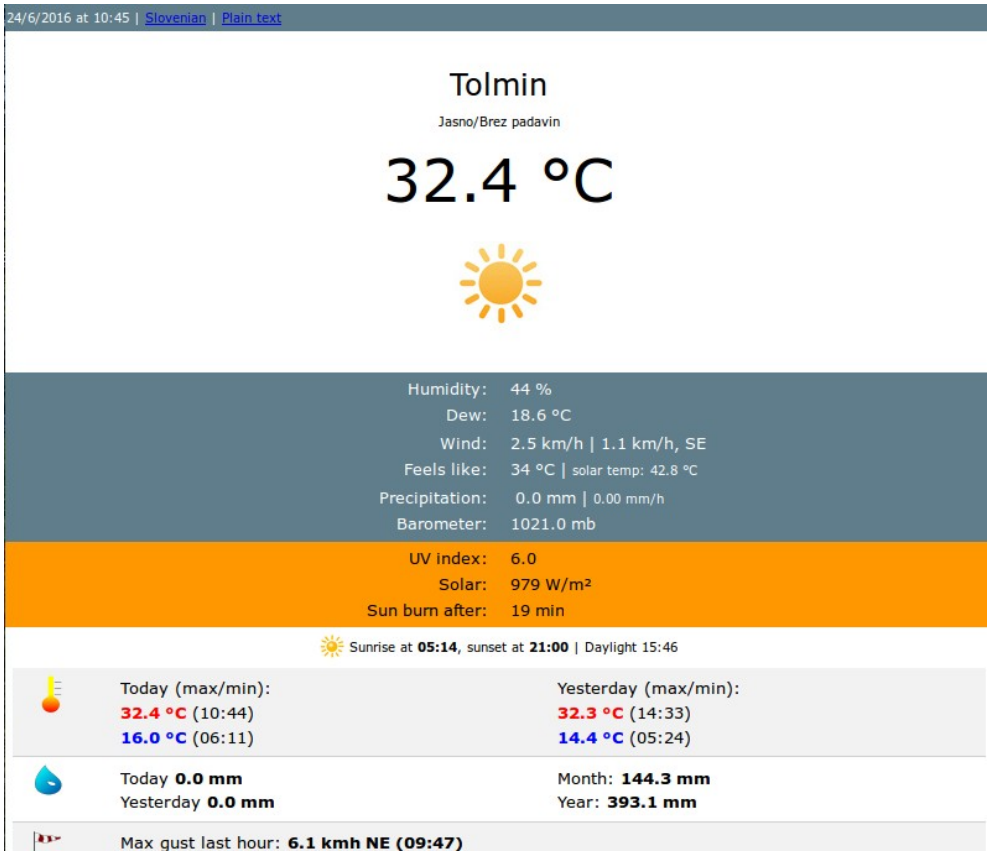
Step 3. Play with software.

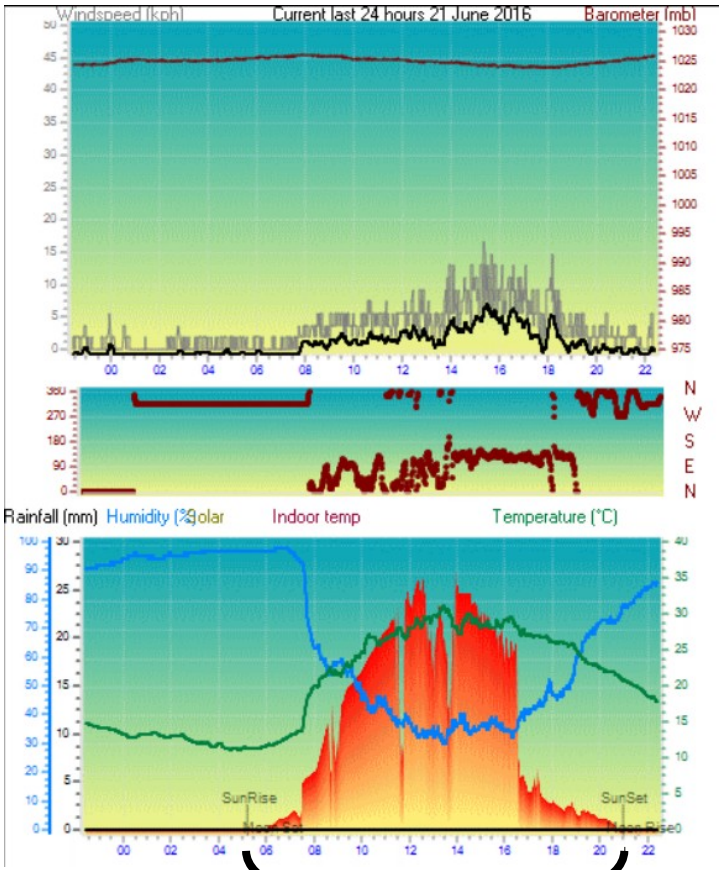


Tons of parameters, files, reports, forecasts, upload options, webcam, time-lapse webcam, solar display, lightning counter, etc.

+ I can add a dozen of extra Dallas sensors

Step 4. This was my project





Longest day at my lattitude.

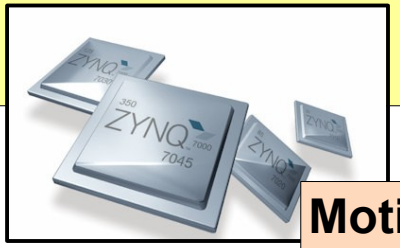
http://freeweb.t-2.net/vreme_tmin

The hobby and the fun part seen through the eyes of the company:

The need for a new instrument, new application

You have a Zynq

Motivation for an engineer
- Programmable logic and the CPU in one chip
- Easy to program, easy to build “an instrument”
- Cheap
- Fun



Motivation for a company
- Programmable logic and the CPU in one chip
- Easy to program, easy to build “an instrument”
- Cheap to produce an instrument

Why would an engineer actually buy an instrument instead of build one?

Because ... but

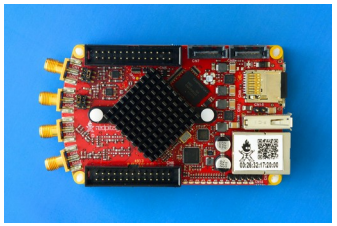
- It's fun to discover, measure, play with the prototype board...**but** what / when is the end point?
- Continuous improvement is essential **but** could be painful (maintenance of several (prototype) versions)
- One (prototype) board works **but** it is not series production
- Make the instrument work with my Control System **but** is probably not compatible with others
- Chips are cheap **but** not always result in building a cheap instrument (development time is more expensive)

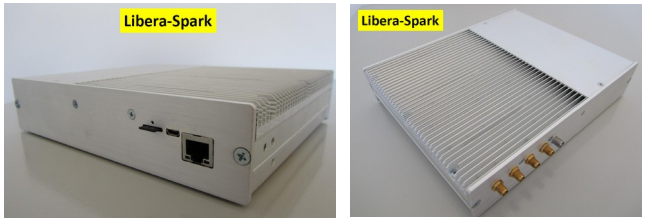
There are the mythbusters that break the buts, but...

...they are not very common and they don't admit all the pain :-)

How it started (2013 – 2014)

	Red Pitaya	**Booster BPM**
ADC	2 inputs, 125 MS/s, 14 bit 4 inputs, 100 kS/s, 12 bit	4 inputs, 125 MS/s, 14 bit
DAC	2 outputs, 125 MS/s, 14 bit 4 outputs, 100 kS/s, 12 bit	Not available
SoC	Zynq 7010, 512 MB DDR3	Zynq 7020, 512 MB DDR3





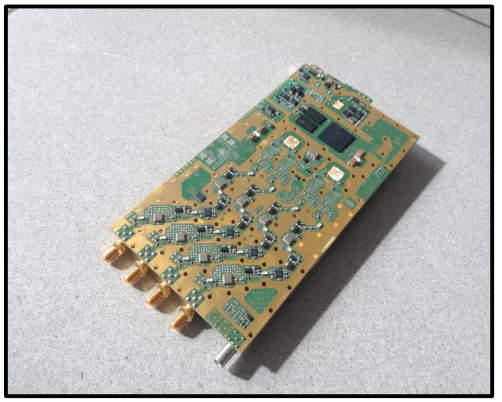
Courtesy of Kees Scheidt, ESRF

+ all source code

Platform with (almost) generic HW and SW interface (2015-2016)

Booster ring / Storage ring BPM

ESRF The European Synchrotron · Argonne NATIONAL LABORATORY · Siam Photon · 自然科学研究機構 分子科学研究所 極端紫外光研究施設 · diamond



Single pass BPM

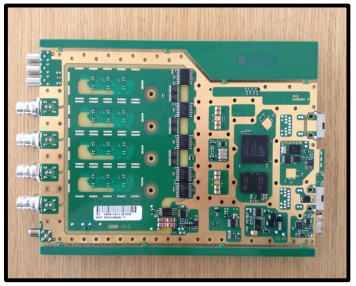
HZDR · PEKING UNIVERSITY · HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF · ALBA · ADAM Applications of Detectors and Accelerators to Medicine

Digitizer

University of Twente The Netherlands · KOREA INSTITUTE OF RADIOLOGICAL & MEDICAL SCIENCES · PEKING UNIVERSITY

Beam loss monitor

ESRF The European Synchrotron · Siam Photon · DESY · NSRRC · Institute of High Energy Physics Chinese Academy of Sciences · SESAME · SACLA



Photon BPM

SSRF Shanghai Synchrotron Radiation Facility · NSRRC · NORTHWESTERN UNIVERSITY

All are similar...

but...

are still different

What if the original software is no good? (the 1st but)

- C++ code examples provided in a Virtualbox image.

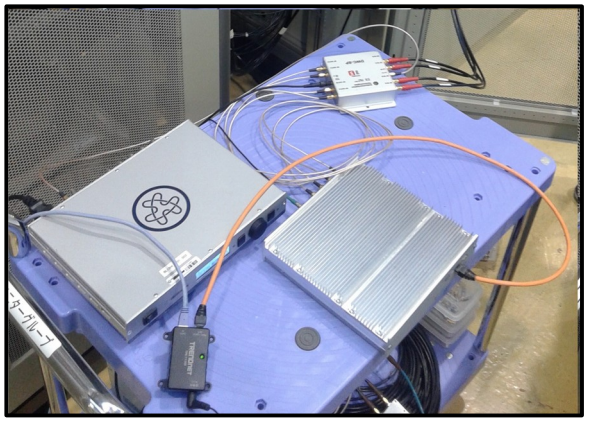
  BNL RHIC: Smooth and painless integration to the control system (no EPICS, no TANGO)

  HZDR: OPC-UA server on the device. Stream out data as UDP.

- Source code is available
- TANGO interface optimization with support from the ESRF

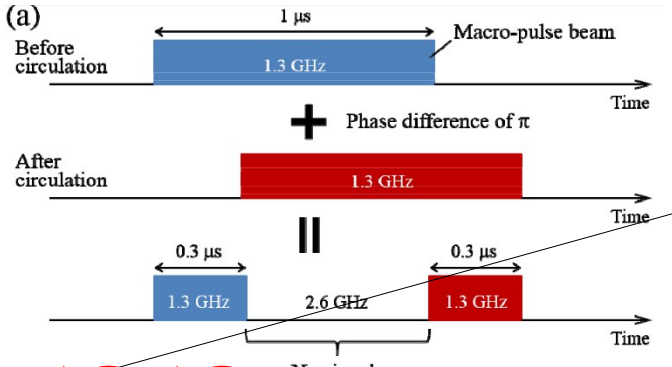
Beam in the compact ERL at KEK

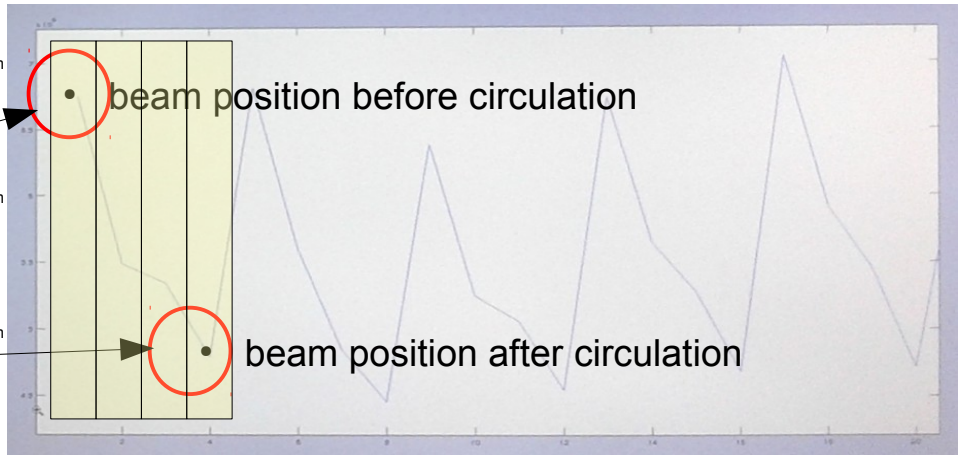
Could the processing be slightly modified? (the 2nd but)



Pulses at 1.3 GHz, 1 us macro-pulse

- Down conversion to 500 MHz
- Multiple processing windows





Requests, expectations, questions
(many more ...buts)

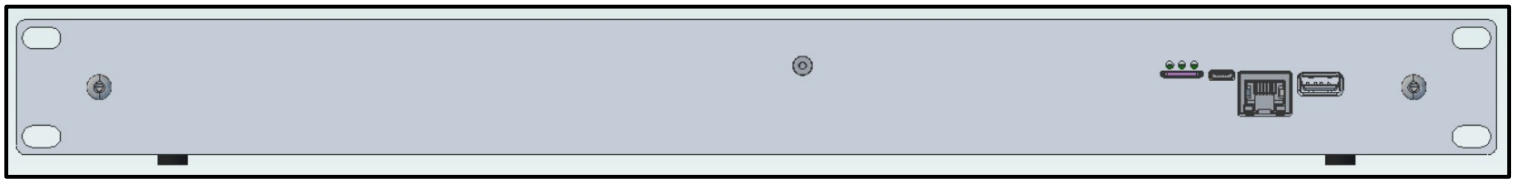
- **Faster sampling rate: bunch-by-bunch BPM, cavity BPM, Beam loss monitor**
- **Optical links**
- **More input channels**
- **Analog output(s)**
- **More I/O interfaces (digital)**
- **Exchangeable front-end**
- **Better long-term stability performance**

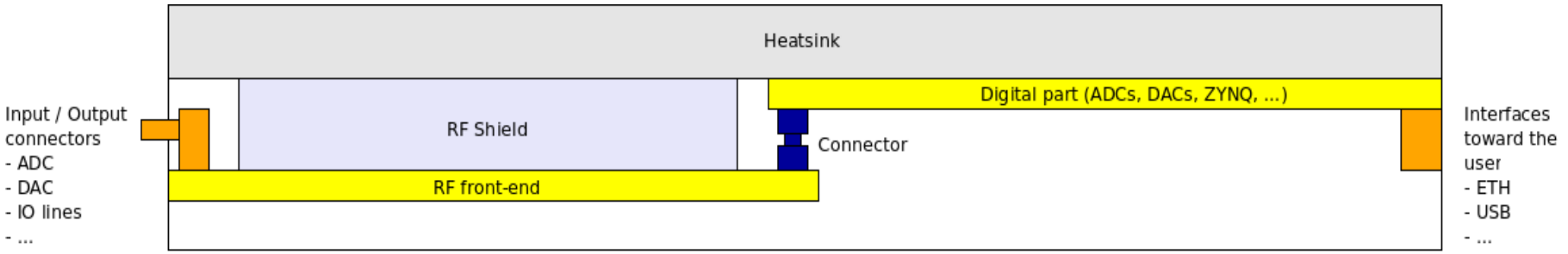
ongoing discussions with PAL, INFN, SPring-8, KEK, ESRF

Future platforms (digitizers)

	8 IN / 2 OUT	**4 IN / (1 OUT?)**
ADC	250 MS/s, 14 bit 4x LTC2123 or 4x ADS42JB49	500 MS/s, 16 bit 4x ADS54J54 or 4x ADS54J66
DAC	2x 250 MS/s, 16 bit	?
SoC	Zynq 7035	Zynq 7035
	Digitizer 2x cavity BPM 2x stripline BPM	Digitizer Beam loss monitor cavity BPM Bunch by bunch (BPM)

8-channel version





The front-end is exchangeable

8-channel version (continuation...)

Digitizer | **Strip-line BPM** | **Cavity BPM**

Exchangeable front-end

Buffer length: >100 µs
Trigger rate: 100 Hz
External reference clock
50 Ω input termination
50 Ω output termination
I/O LEMO connectors
Network attached device
PoE or external supply
19” width

No processing is specified.
Left to user.

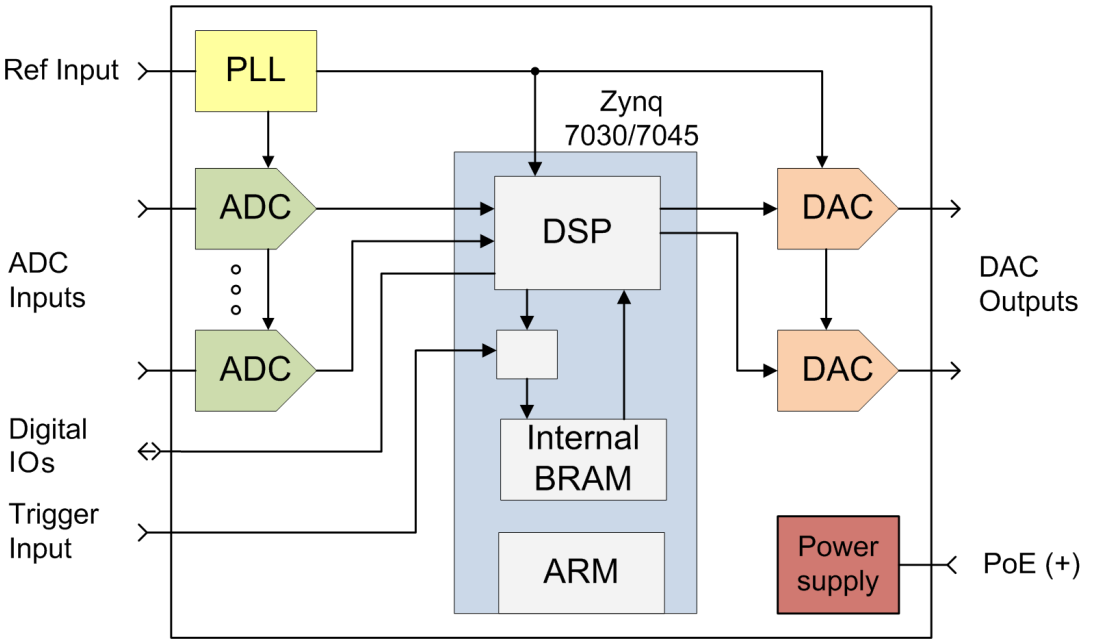
EPICS TANGO C++

4-channel version

Digitizer

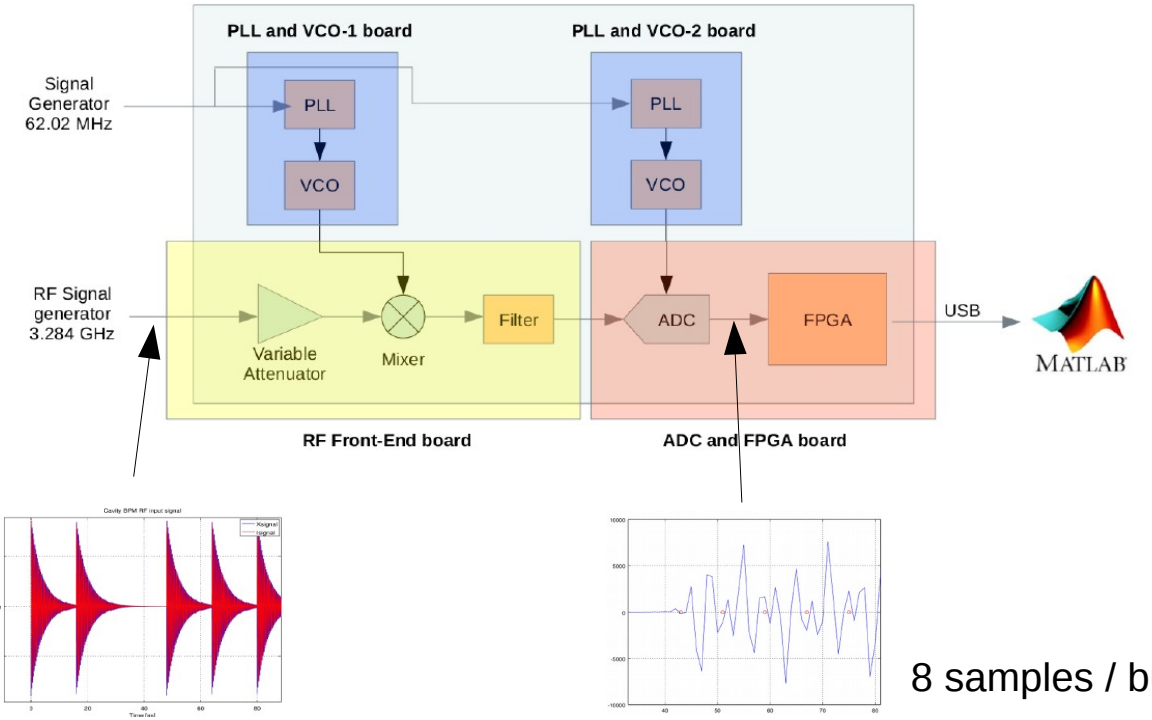
Cavity BPM

Buffer length: 50 µs
Trigger rate: 100 Hz
External reference clock
50 Ω input termination
I/O LEMO connectors
Network attached device
PoE or external supply
9.5” width

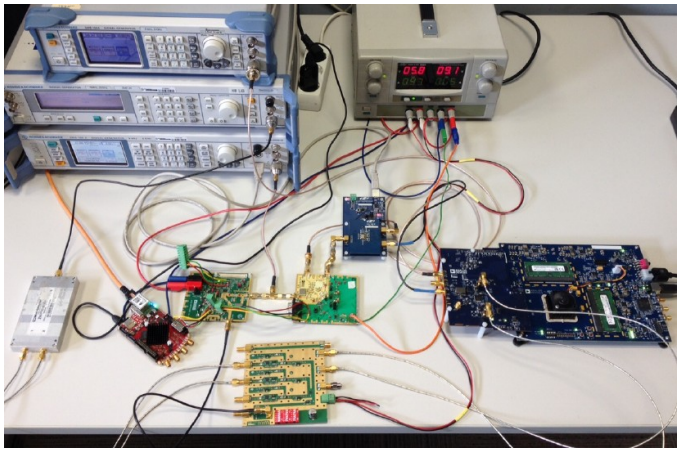


SO-DIMM supported optionally for large buffer storage

4-channel version (cavity BPM...)



8 samples / bunch



Development board (Virtex7)
Analog front-end
Red Pitaya

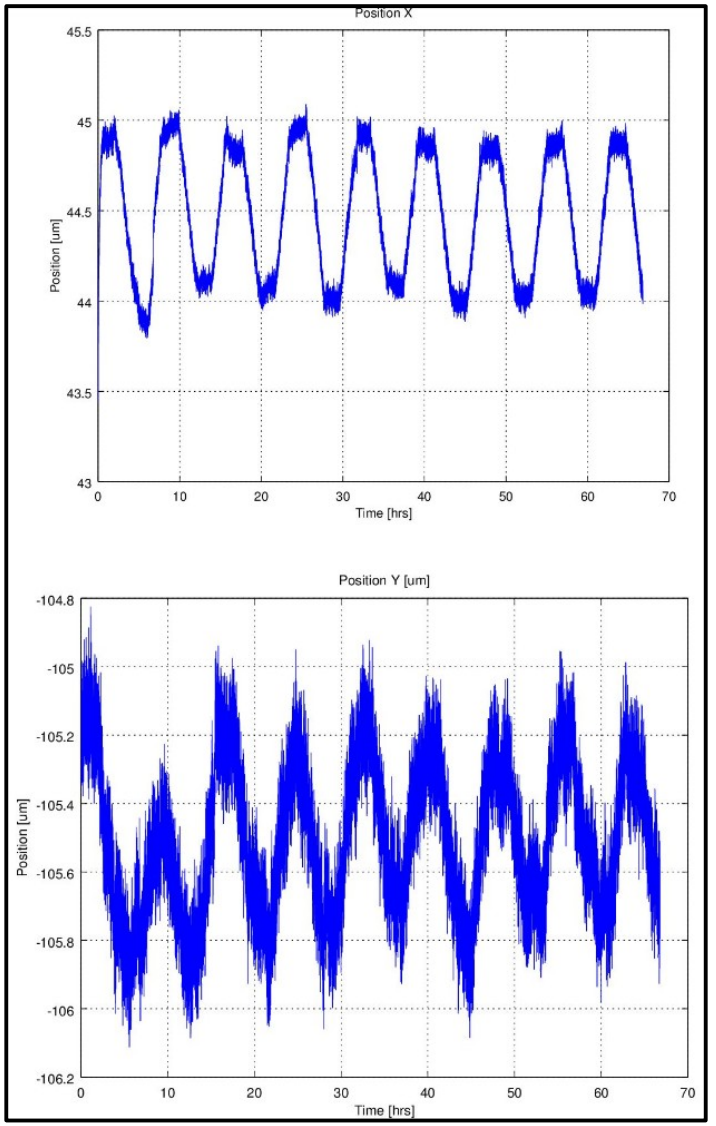
Long-term stability optimization

Present performance: ~1.5 µm/K
A measure is Libera Brilliance(+): 0.15 µm/K, typically

Prototype implementation in the Spark ERXR
- Tests in the temperature chamber
- 20 – 30 °C cycling
- 2 h at each temperature with 2 h transition

Measured dependence: 0.07 µm/K

To be evaluated for the EBS



Before conclusion

- Trying to keep with the PoE(+)
- Take into account the possible 'but' words during low-level design
- Not stick to strict application (e.g. BLM) but offer a clean »digitizer« version in parallel
- Keep the software interface backward compatible
- Continue with SoC for compact instruments
- Looking for Ultrascale for the more complex instruments

Conclusion

- Fast(er) modifications, new application development
- A step closer to engineers – source code, example code
- **Not an »I-Tech« project but a user project**
- Specifications for the future platforms are available but still in iteration phase, some orders already placed
- All invited to join the discussion and development