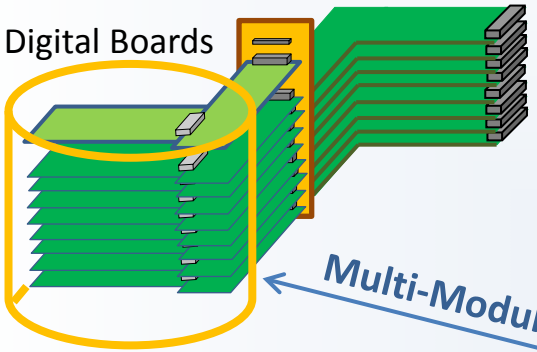
AGIPD 1M Firmware / Software

(Multi-Module System Status)

- Synchronization & Communication schemes
- Versatile Communicator (PPC, C++)
- Usage example: Pixel Reordering debugging
- Monitoring and Control (Device Server Implementation)
- Some Photos from hardware , heating measurement
- Summary , outlook

Multi-Module Front-End Readout (1M system)

Readout sequence , triggering, C&C information,.. distributed by two Control modules via dedicated signal lines to (8+8)Readout modules. According these signals Readout modules execute all proper data processing and send these data to Back-End.



Multi-Module FW/SW

Firmware
ADC data readout
Vetoing
Pre-processing image data
Transmit to Back End, Configure/drive ASICs (Periphery Commands)

FPGA VHDL

Software

16x10GbE

TB

PCs

DB

C&C

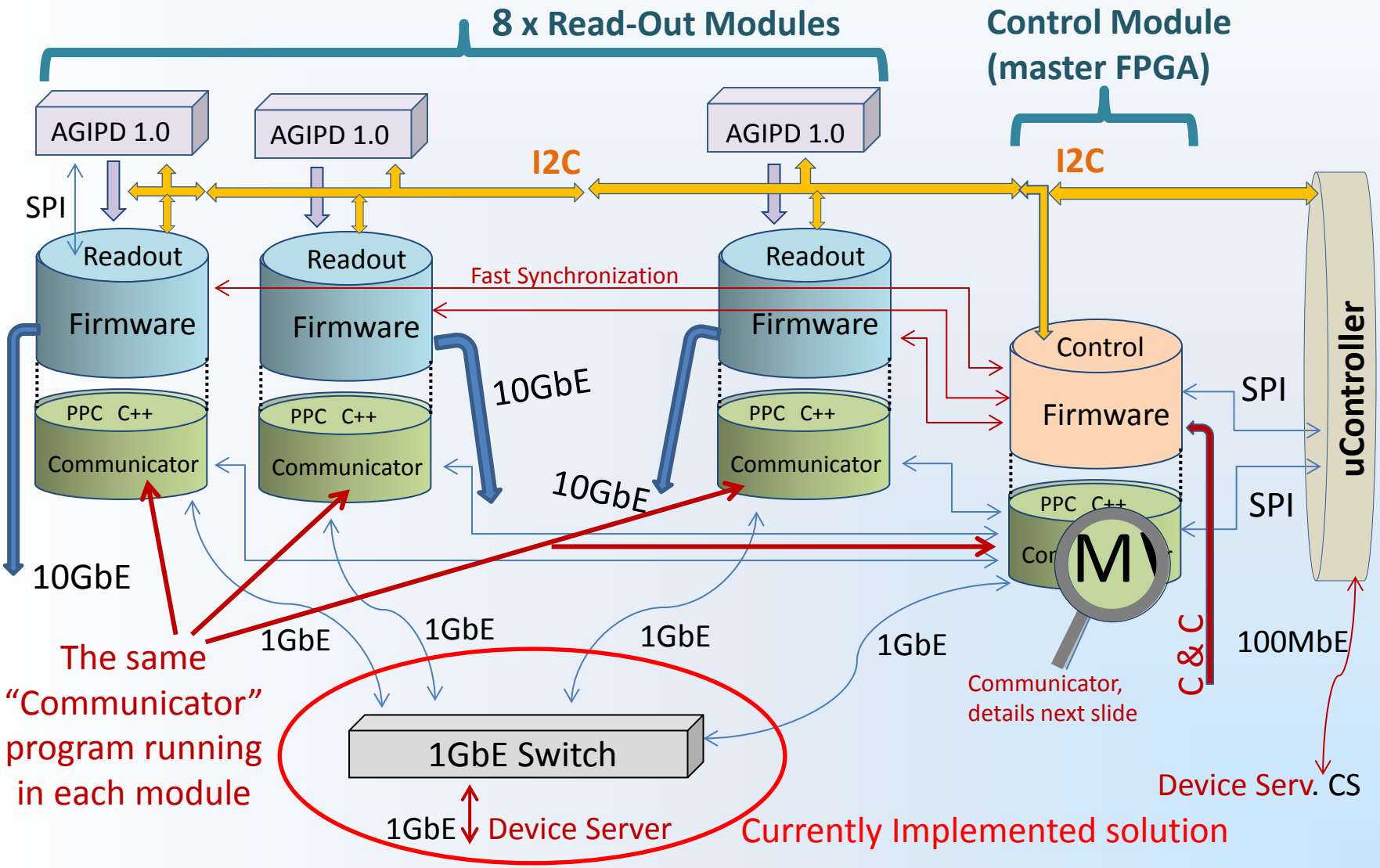
Next Slide

Firmware & Software
Clock & Control, Configure Readout FSM
Central SC, DB Access Power On/Off, CS
ADC control via SPI , I2C, Reset's

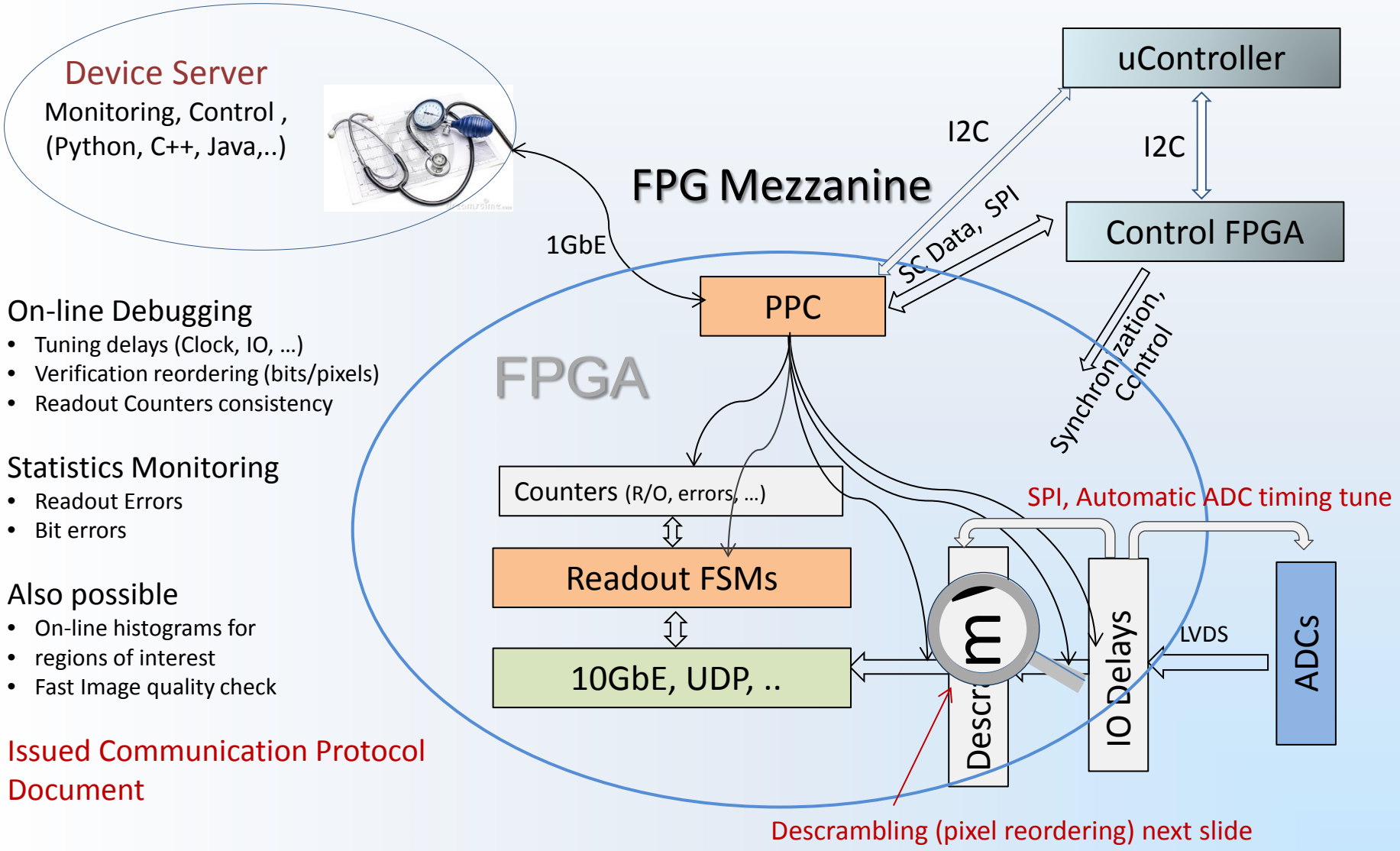
Local Control & Monitoring

Central Control, show online data, …… ..

Multi-Module Synchronization & Control communication schemes



Versatile Communicator (PPC, C++)



On-line Debugging

- Tuning delays (Clock, IO, …)
- Verification reordering (bits/pixels)
- Readout Counters consistency

Statistics Monitoring

- Readout Errors
- Bit errors

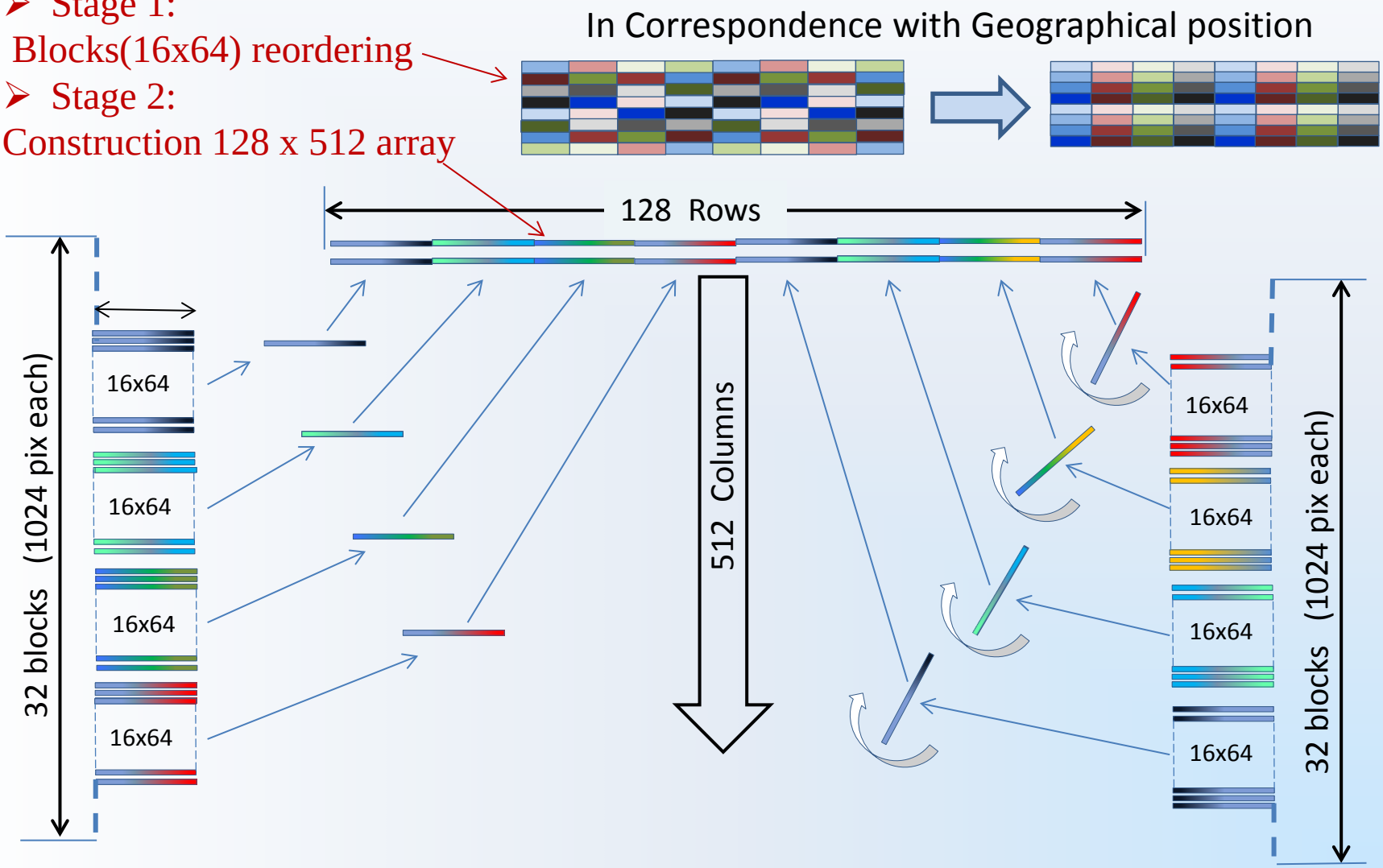
Also possible

- On-line histograms for
- regions of interest
- Fast Image quality check

Issued Communication Protocol Document

Example: Help to debug Pixel Reordering

- Stage 1:
  Blocks(16x64) reordering
- Stage 2:
  Construction 128 x 512 array



Descrambling on the fly

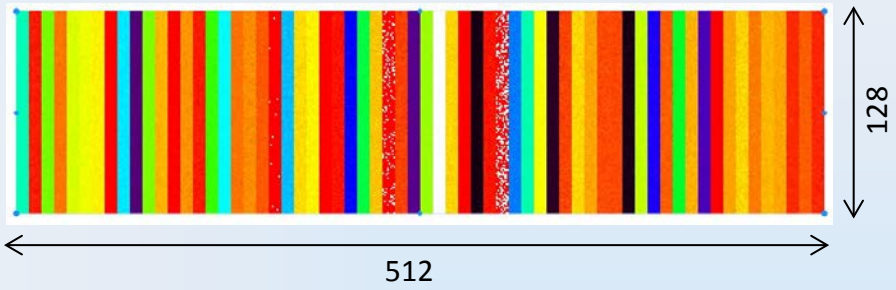
Input Data Stream (Output From ADC) => Array [512] [128]

Direct implementation of the algorithm have written below based on the multiple loops requires significant time and memory resources

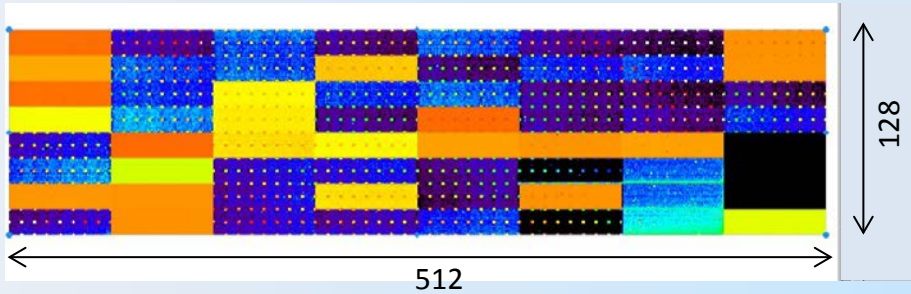
```
For (ASIC# = 1;  ASIC# < 9;  ASIC#  ++)
{ For (COLUMN# = 0;  COLUMN# < 64;  COLUMN# ++)
   { BR_OFFSET_BOT = (ASIC# - 1)*4;   // Read Bottom Sub Column
{  For (BRAM# = BR_OFFSET_BOT;  BRAM # < BR_OFFSET_BOT+4;  BRAM #++)
{ PIX_OFFSET = 16*COLUMN#
  For (PIX# = PIX_OFFSET;  PIX # < PIX_OFFSET +16;  PIX #++)
{
DATA_OUT = BRAM[BRAM#][PIX#];
}
} // PIX#
} //  BRAM#

    BR_OFFSET_TOP =  (ASIC#+8) *4 - 1;    // Read Top Sub Column
{  For (BRAM#  = BR_OFFSET_TOP;  BRAM #  > BR_OFFSET_TOP -  4;  BRAM #  - -)
{ PIX_OFFSET = 1023 - 16*COLUMN#;
  For (PIX# = PIX_OFFSET;  PIX #  > PIX_OFFSET - 16;  PIX # - -);
{
DATA_OUT = BRAM[BRAM#][PIX#];
}
 } // PIX#
   } //  BRAM#
      } // COLUMN#
  }  // ASIC#
```

Using " Vector based " FPGA algorithms the full descrambling has required just two clock cycles and performed on the fly with only several clocks latency.

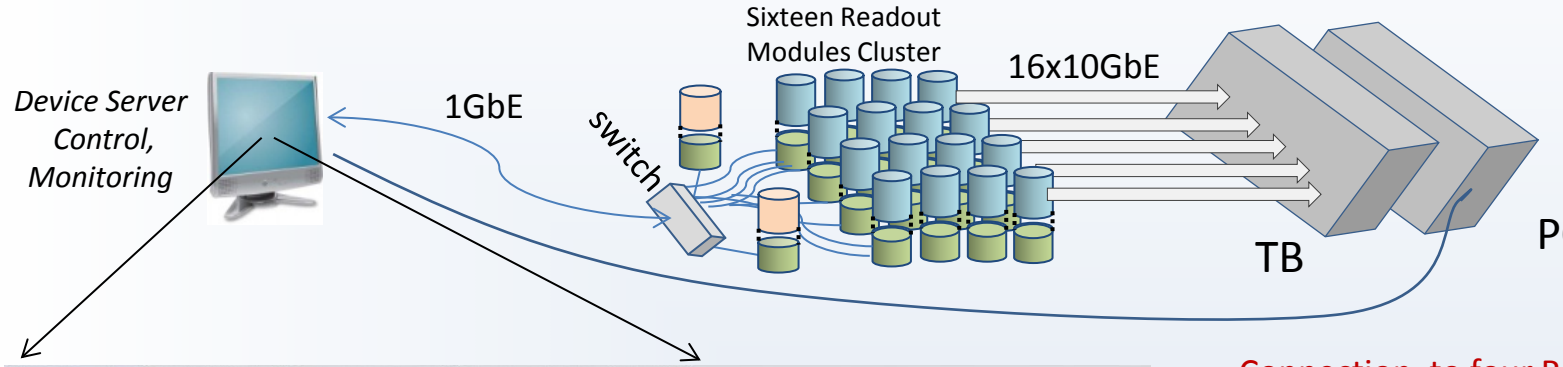
Input => 64 data blocks (1024 pixels each)

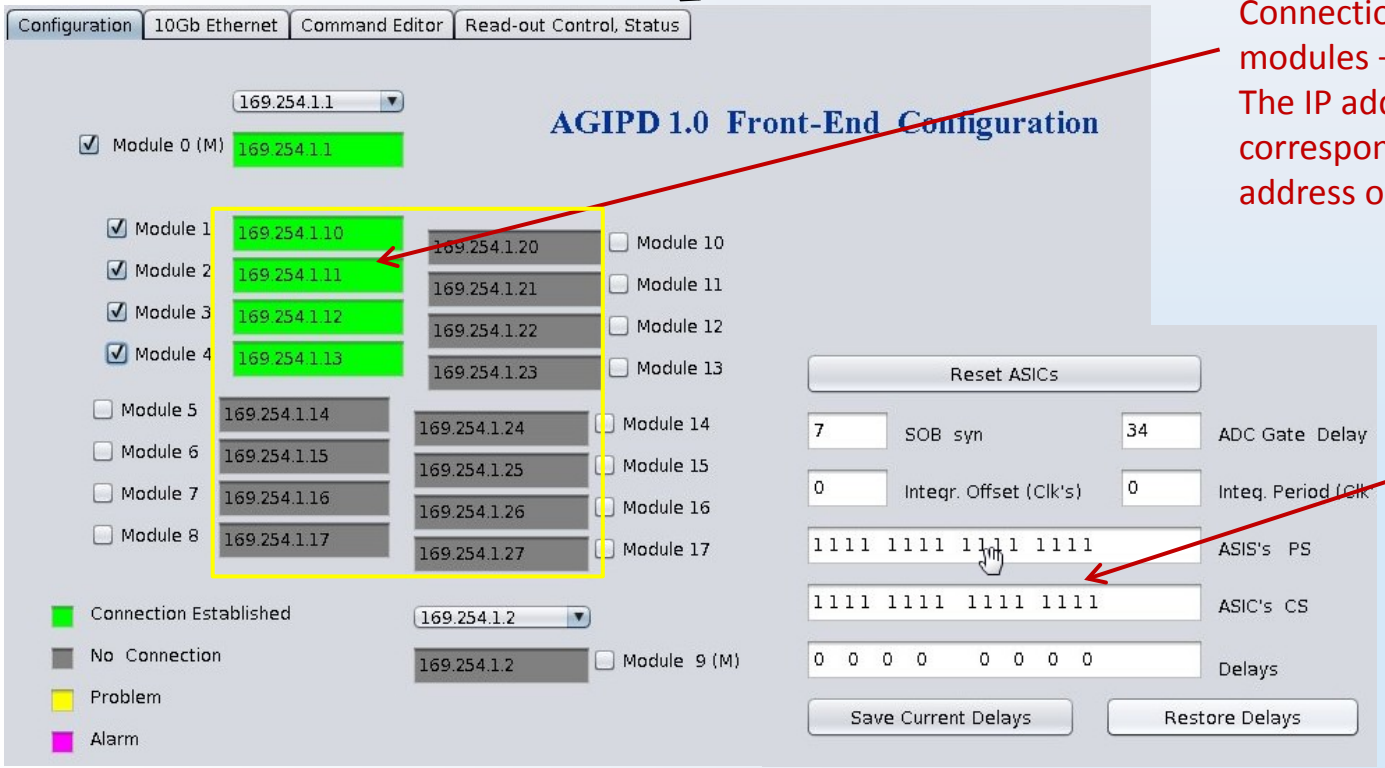


The same 64 data blocks after descrambling



Monitoring & Control for up to Sixteen Readout Modules

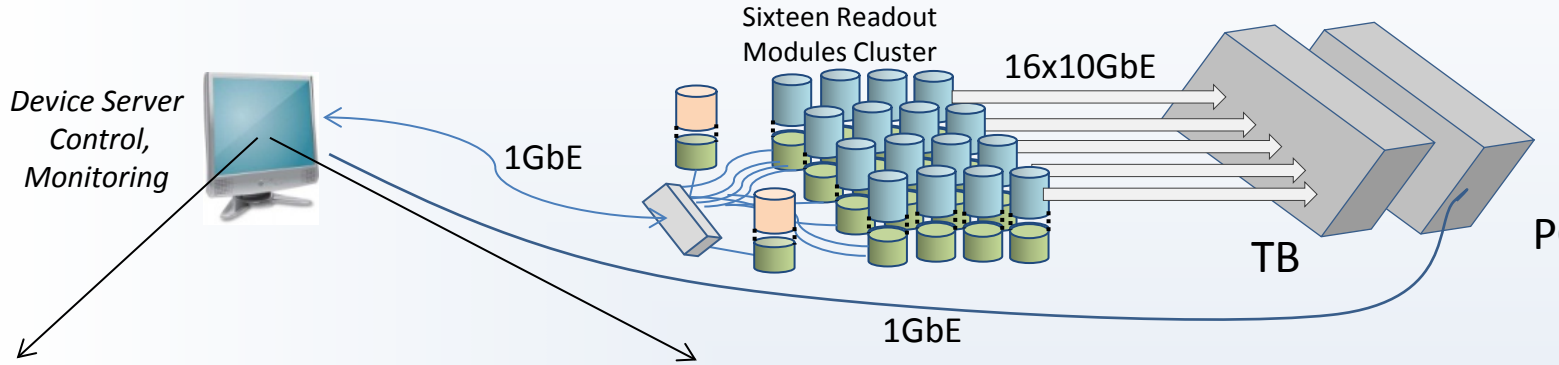




Connection to four Readout modules + Master established
The IP addresses automatically correspond to Geographical address of the Module

ASICs, ADC control via I2C, SPI

10Gb Ethernet Configuration



Configuration | 10Gb Ethernet | Command Editor | Read-out Control, Status

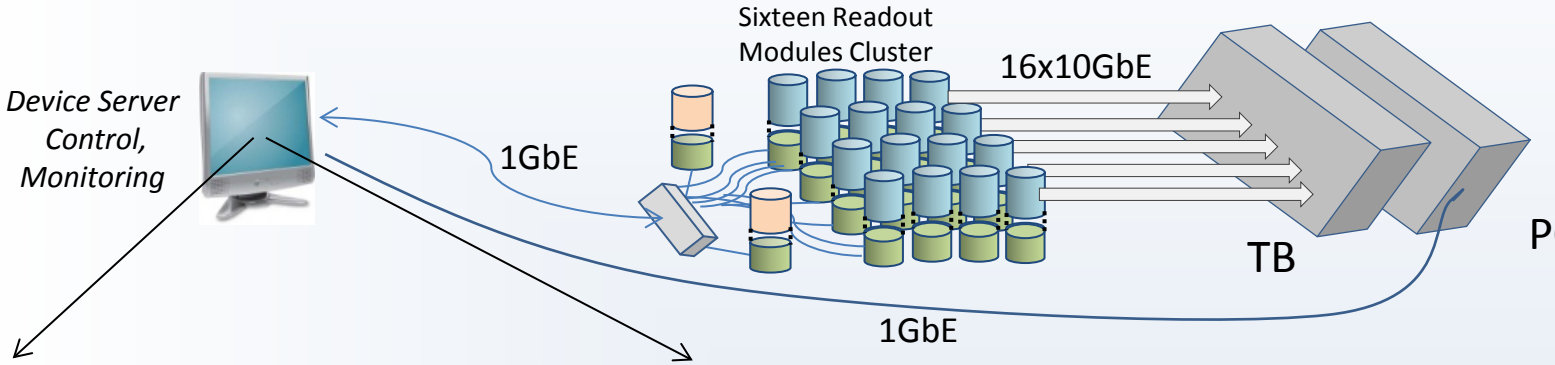
Front-End Modules Ethernet Addresses

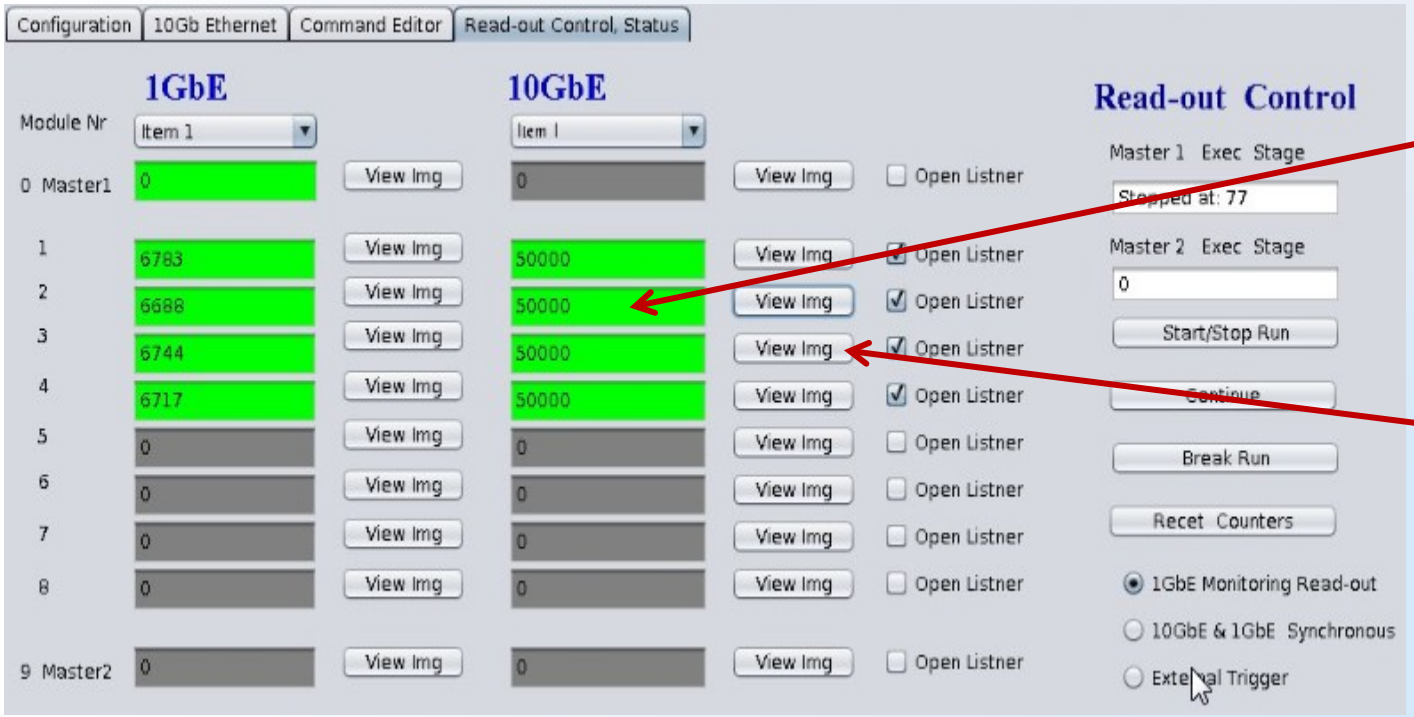
Send to H/W | Save All Current Addresses | Restore All Addresses

	MAC Addresses		MAC Addresses		IP Addresses		IP Addresses	
Src0 M	00:17:BD:A3:F1:45	OK	Src9 M	21:43:56:bf:de:13	196.254.1.2	OK	196.254.5.8	OK
Dst0 M	01:18:BE:A4:F2:44	OK	Dst9 M	21:43:56:bf:de:13	196.254.1.4	OK	196.254.5.8	OK
Src1	00:17:BD:A3:F1:47	OK	Src10	21:43:56:bf:de:13	196.254.1.5	OK	196.254.5.8	OK
Dst1	C8:1F:66:D9:E8:0D	OK	Dst10	21:43:56:bf:de:13	196.254.1.4	OK	196.254.5.8	OK
Src2	00:17:BD:A3:F1:66	OK	Src11	21:43:56:bf:de:13	196.254.3.2	OK	196.254.5.8	OK
Dst2	C8:1F:66:D9:E8:0F	OK	Dst11	21:43:56:bf:de:13	196.254.3.6	OK	196.254.5.8	OK
Src3	01:18:BE:A4:F2:77	OK	Src12	21:43:56:bf:de:13	196.254.5.7	OK	196.254.5.8	OK
Dst3	00:0A:F7:41:2D:50	OK	Dst12	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Src4	21:43:56:bf:de:13	OK	Src13	21:43:56:bf:de:13	196.254.7.8	OK	196.254.5.8	OK
Dst4	00:0A:F7:41:2D:52	OK	Dst13	21:43:56:bf:de:13	196.254.7.10	OK	196.254.5.8	OK
Src5	21:43:56:bf:de:13	OK	Src14	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Dst5	21:43:56:bf:de:13	OK	Dst14	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Src6	21:43:56:bf:de:13	OK	Src15	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Dst6	21:43:56:bf:de:13	OK	Dst15	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Src7	21:43:56:bf:de:13	OK	Src16	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Dst7	21:43:56:bf:de:13	OK	Dst16	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Src8	21:43:56:bf:de:13	OK	Src17	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK
Dst8	21:43:56:bf:de:13	OK	Dst17	21:43:56:bf:de:13	196.254.5.8	OK	196.254.5.8	OK

Online 10Gb Ethernet configuration depending on which Back-End system is used

Read-out status

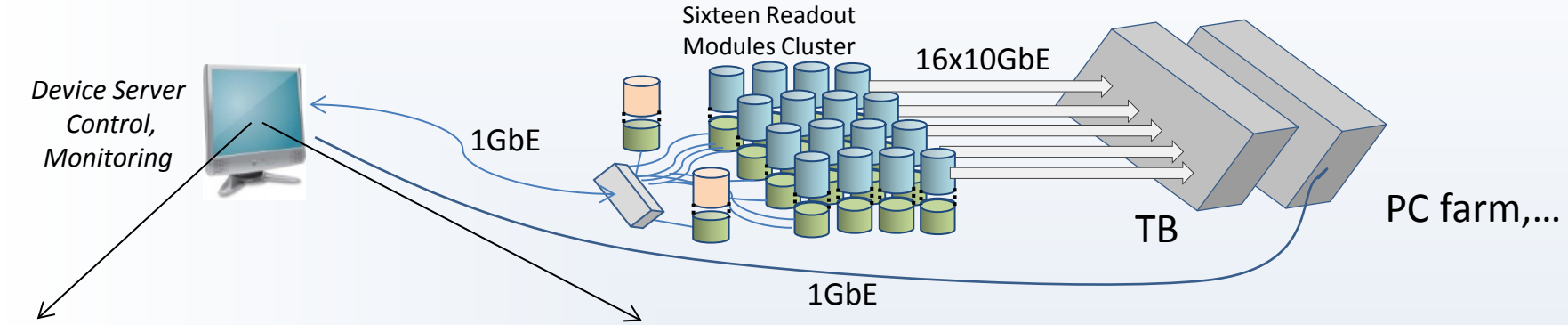


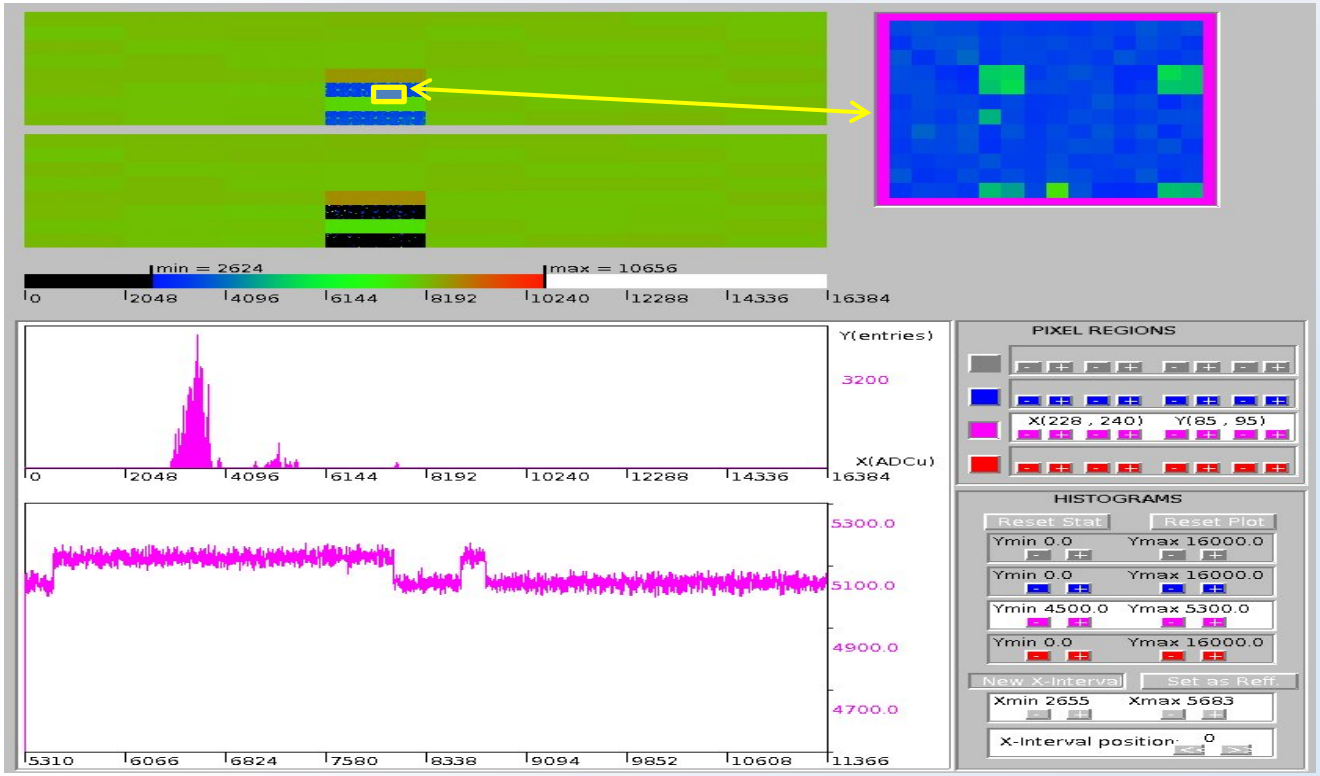


4 x 50000 images transmitted via 4 x 10GbE and recorded to Disc (No one img loss)

If you click "View Img" button => next slide

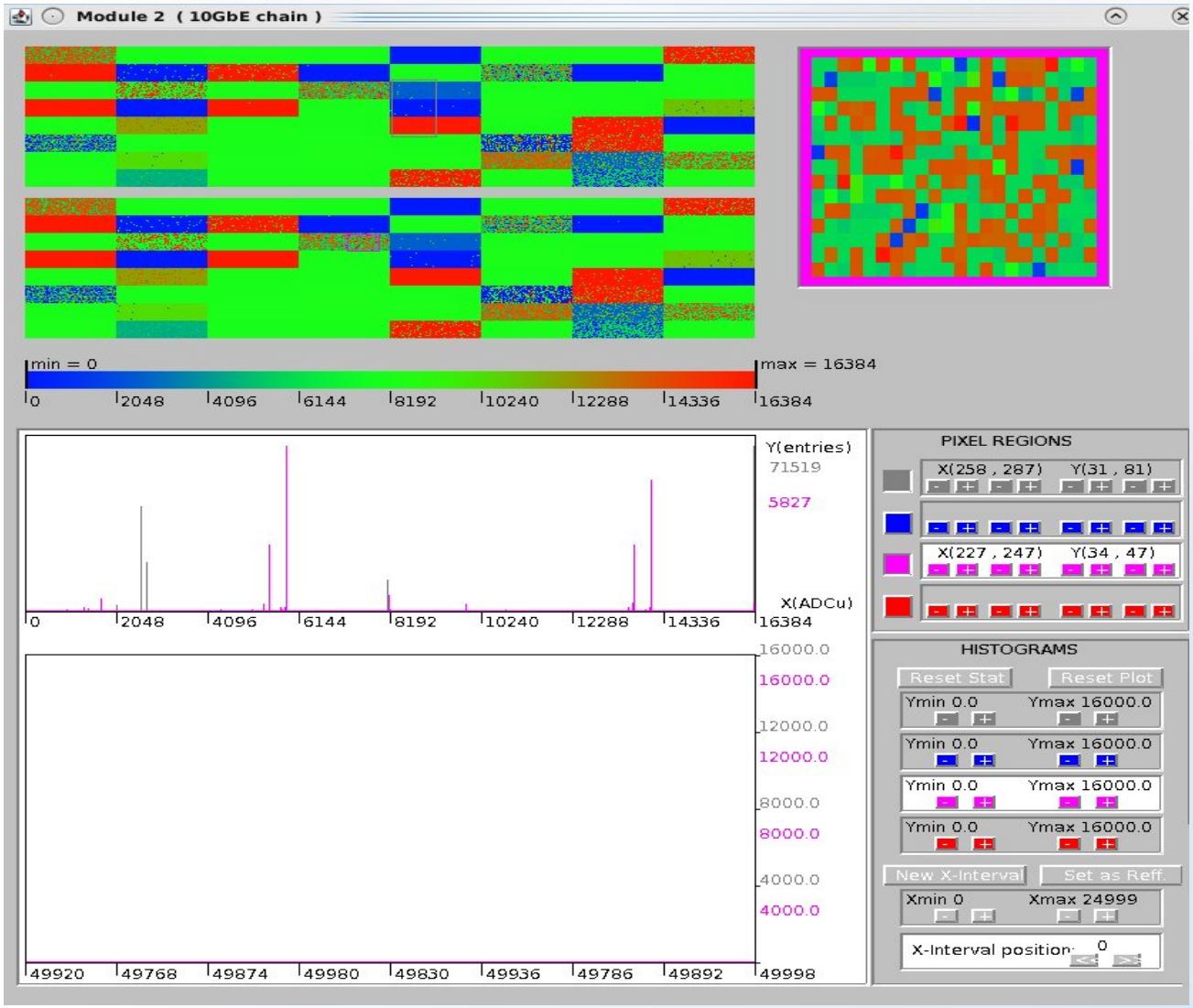
On-Line Imaging





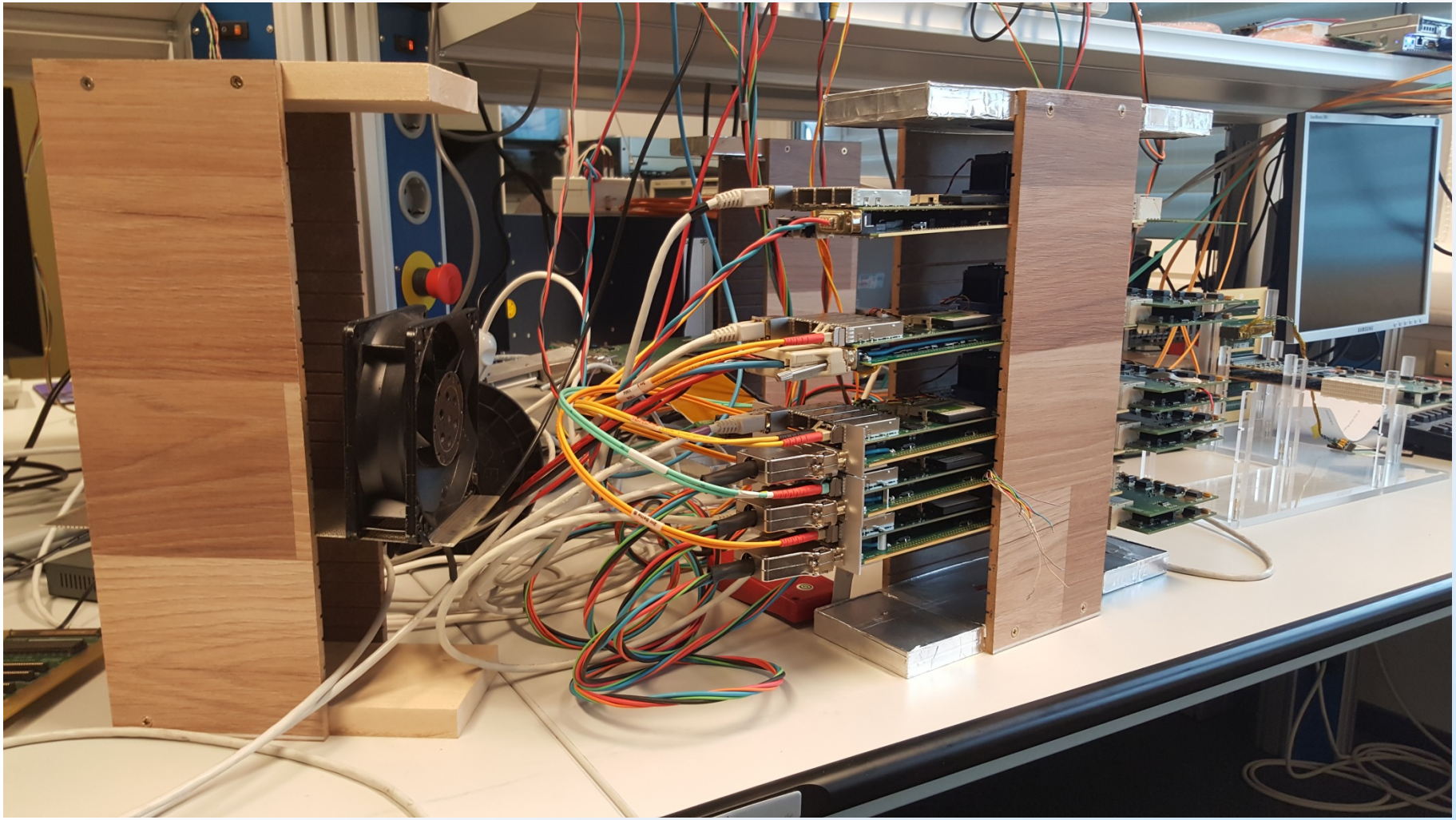
Online Imaging with selection of Regions of interest, Histogramming, Plotting ADC values versus img/time, …

Problem with ADC data (example)

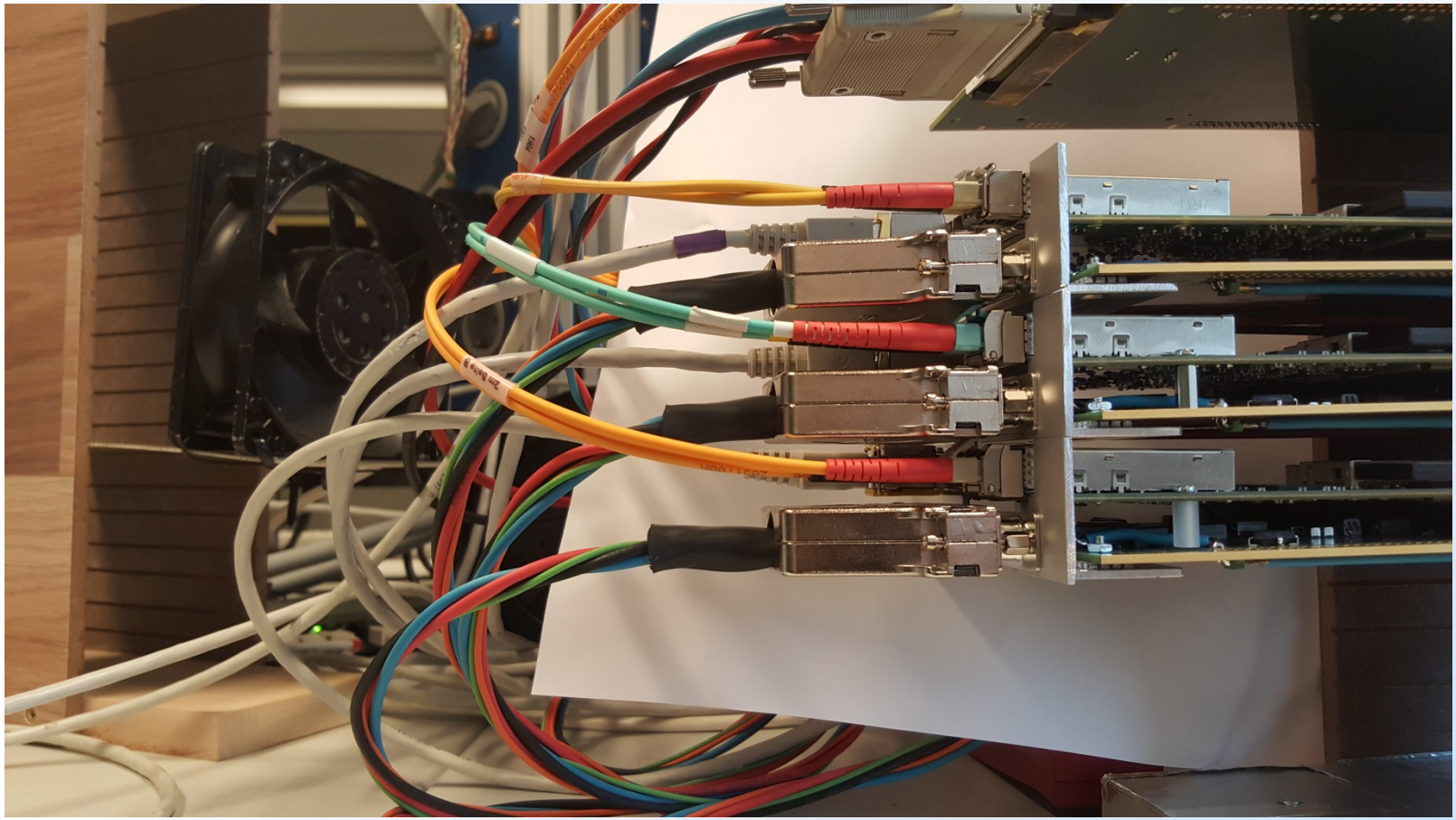


Analogue and Digital PSs were switched ON in wrong order

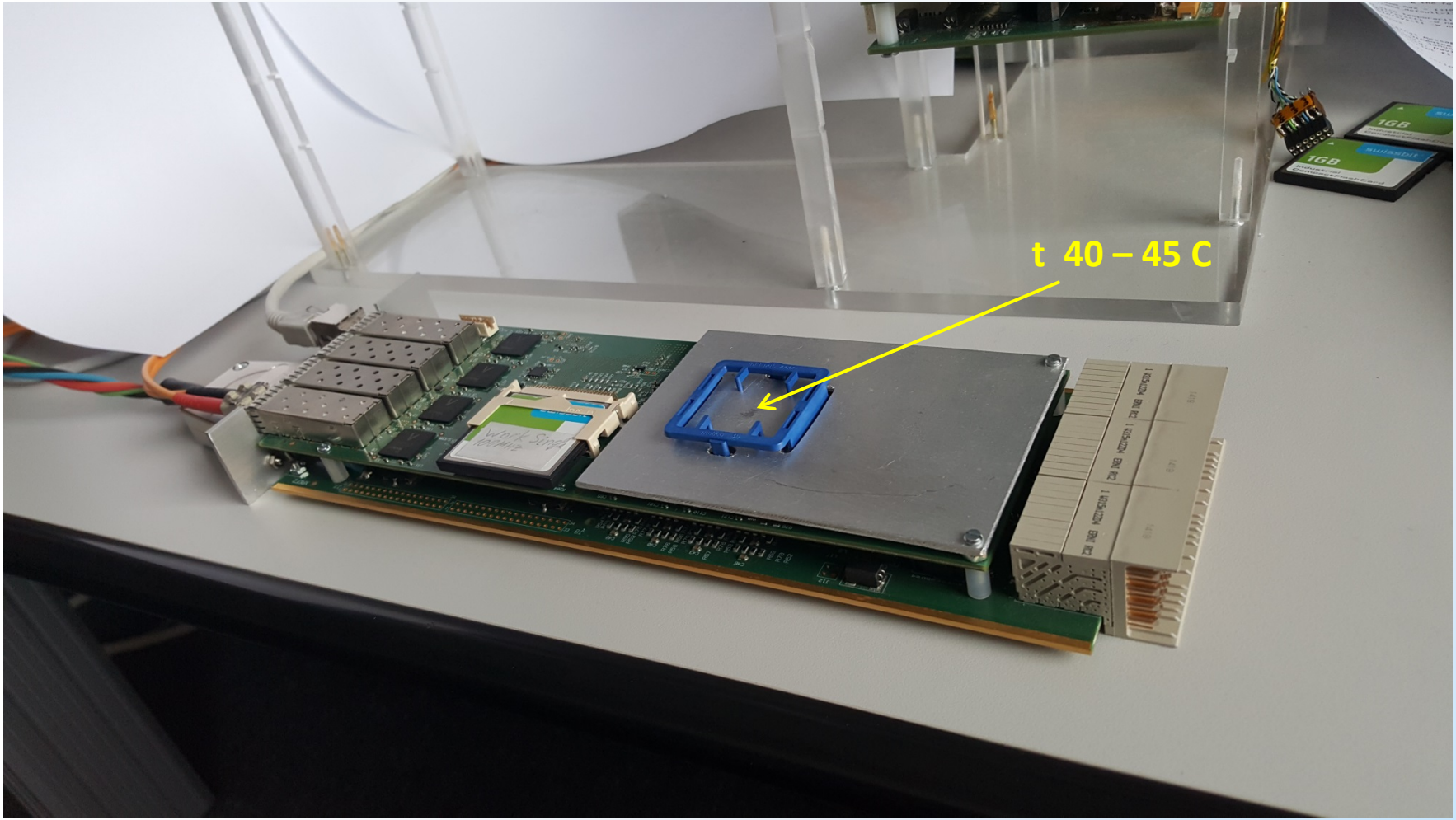
FEA Lab Test Bench



FEA Lab Test Bench



Some Heating test (measured in stack)



Summary , Outlook

Basic environment for Commissioning Start-up is available

- ✓ Full online data flow monitoring (statistic, errors, data corruption, ..)
- ✓ Real time parameters adjustment for different systems

  (Ethernet addresses, readout modes, delays, automatic ADC tuning, …)
- ✓ Dedicated Readout configuration regardless number of used modules and type of Back-up system
- ✓ Versatile communication protocol (already implemented in TANGO server , now moving to KARABO stuff)

Nearest and Future Tasks:

- o More Readout carrier boards are coming soon => testing 8 -16 modules system
- o Hardware integration with C&C system with relevant F/W update.
- o In the Readout FSM interpreter implement new readout macro-command operated in burst mode to fulfil XFEL bunch - veto frequency requirements.
- • Remote (via Ethernet) exchange FPGA boot files (new versions, dedicated read out, …)

Thanks !