A new web Slow Control & Monitoring system featuring SRS ZeroSuppression implementation

StefanoC*, S. L. Goentoro F. Costa, A. Zybel

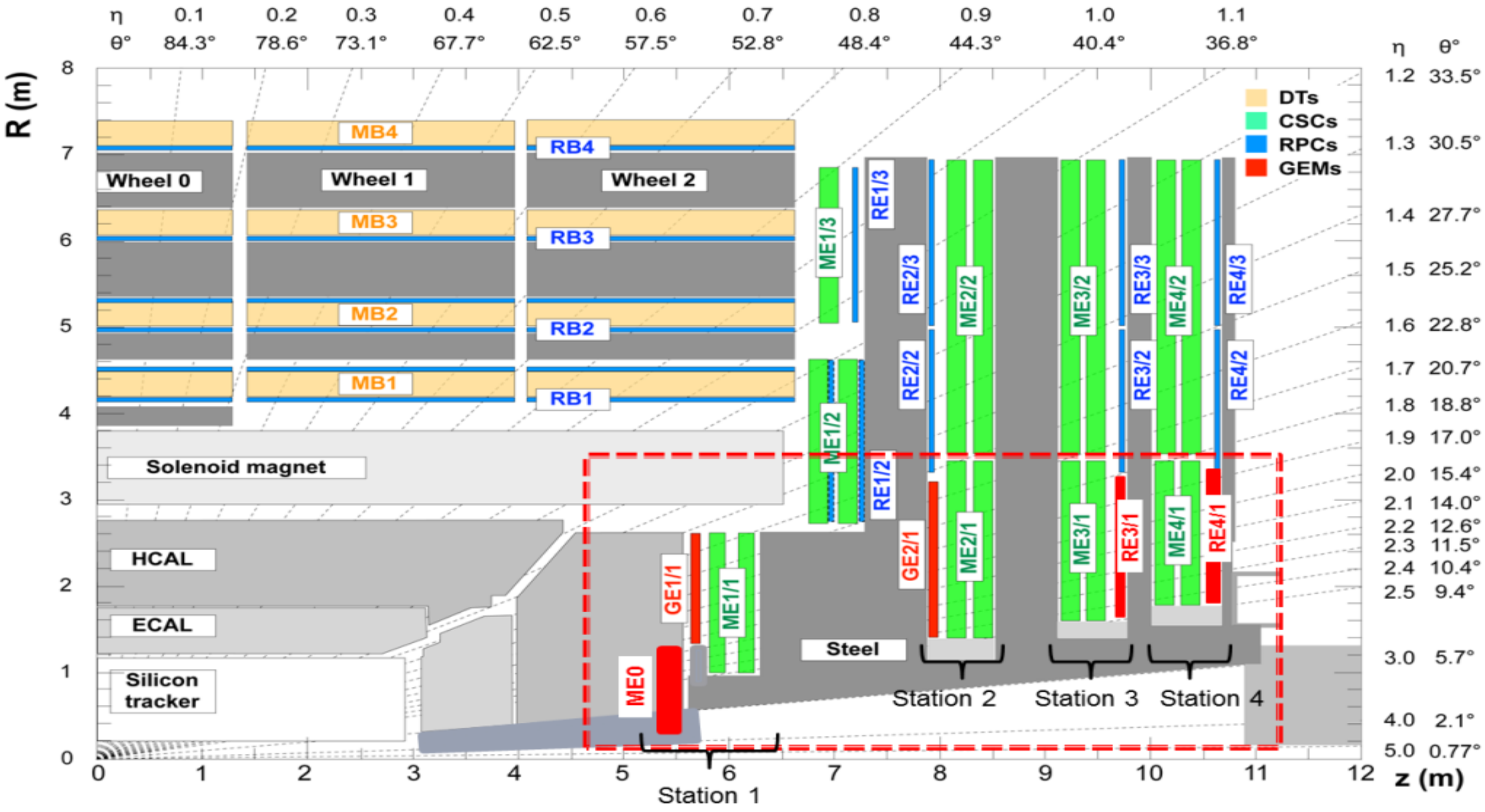
*on behalf of FIT team

The antefact: CMS Muon Upgrade

CMS detector: designed to detect and reconstruct muons with best precision

Phase-II CMS needs handles to cope with high rate between LS2 and LS3, the high-eta region (highest rate) is not detector redundant as all other regions.

CMS needs to increase robustness at high-eta



The antefact: CMS Muon Upgrade

CMS detector: designed to detect and reconstruct muons with best precision

Phase-II CMS needs handles to cope with high rate between LS2 and LS3, tracking trigger won't be installed until ~2023

High eta region has vacancies after LS1

- Good opportunity to install new GEM detectors,
- Original foreseen RPC design is not sustainable at high rates

High eta region requires mature technology

- High rate capability O(MHz/cm^2)
- Good time resolution: triggering
- Good spatial resolution O(100μm): tracking

Objectives of GEM Muon Upgrade

- Sustain triggering at current thresholds up to $|\eta|=2.4$
- Increase offline muon identification coverage to $|\eta|=3.5-4$
- Maintain existing envelope by preventing or addressing aging effects

SRS@GEM assembly sites productions

GEM DETECTOR PRODUCTION FOR THE CMS HIGH-ETA UPGRADE

Flexible detector production schema, construction load balanced among several assembly sites: *(INFN-Frascati, INFN-Bari, FIT, GHENT, SEOUL, BARC)*.

WHY SRS?

- SRS has been chosen by the collaboration and being implemented in all CMS GEM assembly sites to perform detector gain uniformity with APV25.
- A single GE11 detector has >3k channels, SRS is a cost-effective readout system featuring also a great support by CERN RD51.
- SRS is adopted for the crucial task of commissioning and validation of detector performance before installation at CMS.

WHY NEW SOFTWARE TOOLS?

- Production sites needs:
  - Common hardware/software/procedures.
  - CMS and detector customized software features/protocols.

  → Development of Slow Control and Monitoring system **SCRIBE** (but flexible enough to be adapted/customized by other groups..)
- Introduction of ZeroSuppression feature is mandatory to reduce file dimensions and CPU analysis time.

  → Development of analysis framework compatible with ZeroSuppression **AMORE ZS**

Configuring/data-taking of the SRS in a nutshell (1/2)

Requirements for CMS GEM detector assembly sites

- Tool easy to use, quick learning curve
  1) During detector mass production no time for manual file editing and/or complex procedures, no time for recovery of bad configurations/runs.
  2) Data-taking often performed by students, no time for expert training.
- Zero Suppression is needed to reduce file dimensions and CPU analysis time
  1. Gain uniformity for all 3k strips mean ~15M events, without ZS rawfiles >>100T
  2. Assembly sites do not have cluster for data-analysis, AMORE has to run on a simple pc
- Light, modular x-compatible, open-source
  1) All software should be like that isn't??

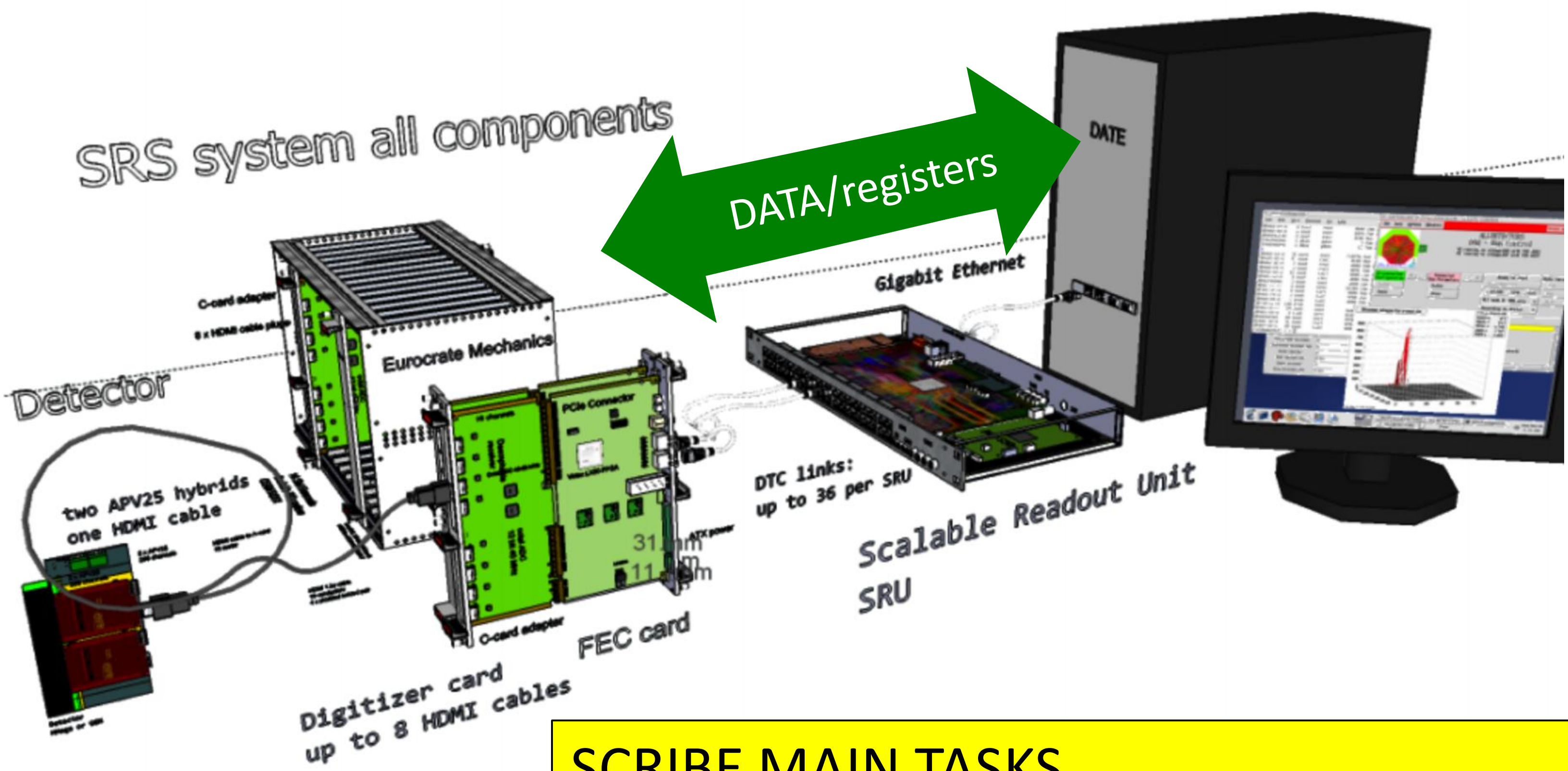
1) SCRIBE (Slow Control and Run Initialization byte-wise environment

- Graphical interface works through dynamic web-app
  - Multi-client, x-platform, x-device
  - Easy to learn (self-explained)
  - Some immediate feedback with real-time plots

2) AMORE ZS

- AMORE is the analysis framework that processes the rawdata files, we developed an adapted data-unpacker to understand new rawdata file structure. The new AMORE features:
  - Same output files as before (seamless)
  - No pedestal needed
  - Reduced CPU time needed for processing time (x100 faster)

Configuring/data-taking of the SRS in a nutshell (2/2)



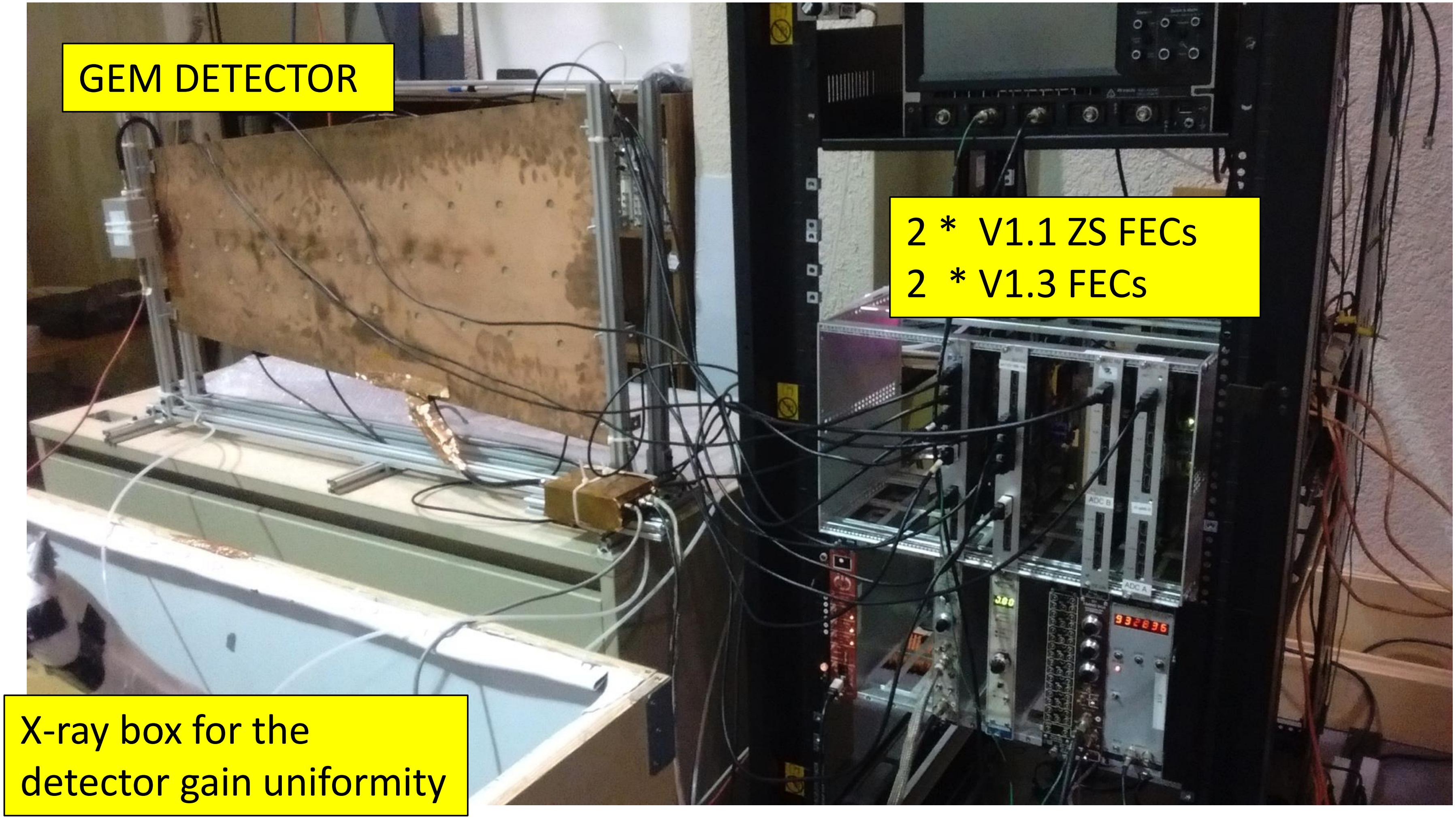
SCRIBE MAIN TASKS

- Easy change/check/monitor one or more register values.
- Easy data-taking

AMORE ZS MAIN TASKS

- Easy data-analysis
- Production of the smallest possible rootfile out of the DATE rawfile

SRS developments at FIT:
SRS stable setup for detector gain uniformity

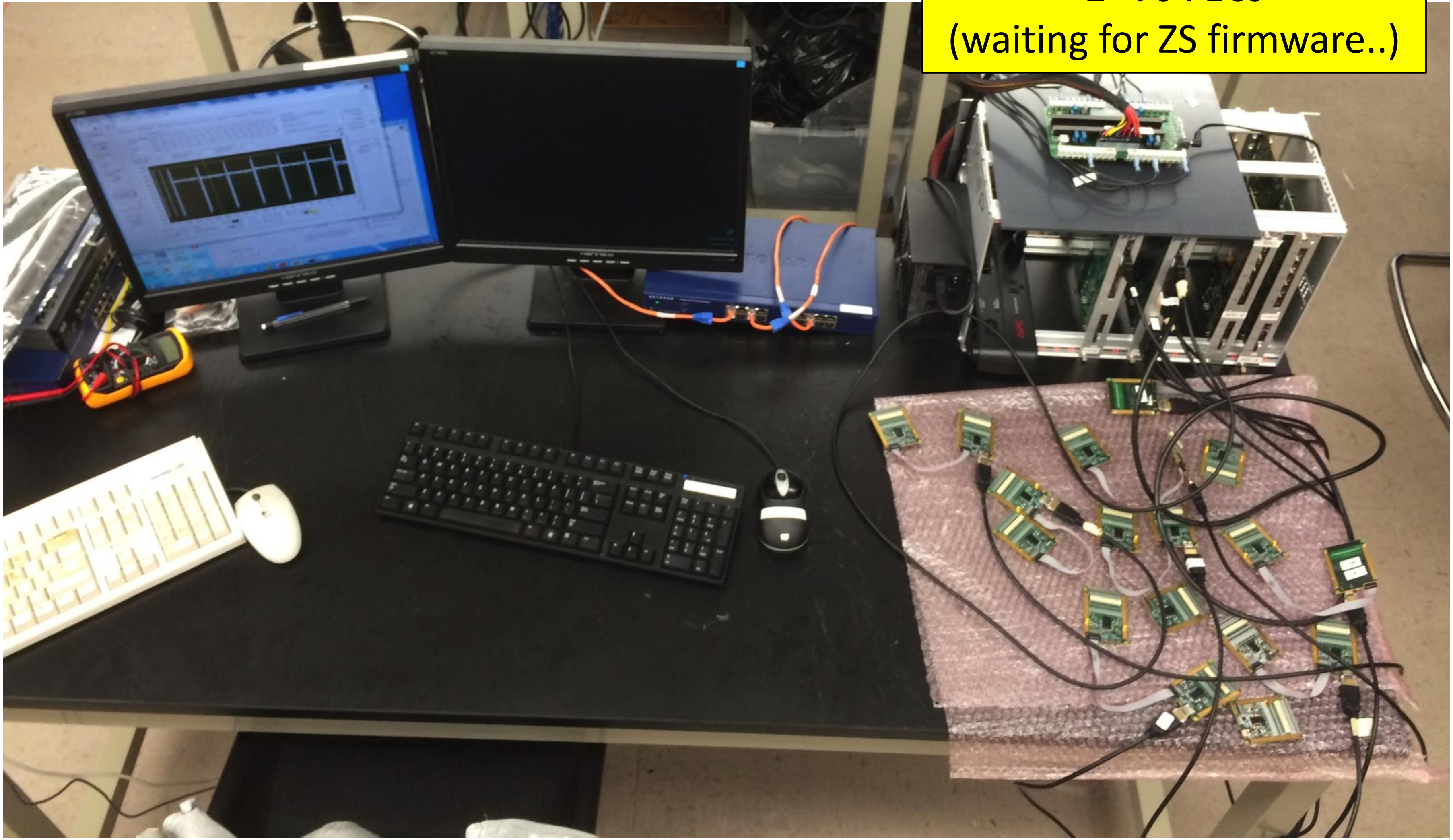


GEM DETECTOR

2 * V1.1 ZS FECs
2 * V1.3 FECs

X-ray box for the
detector gain uniformity

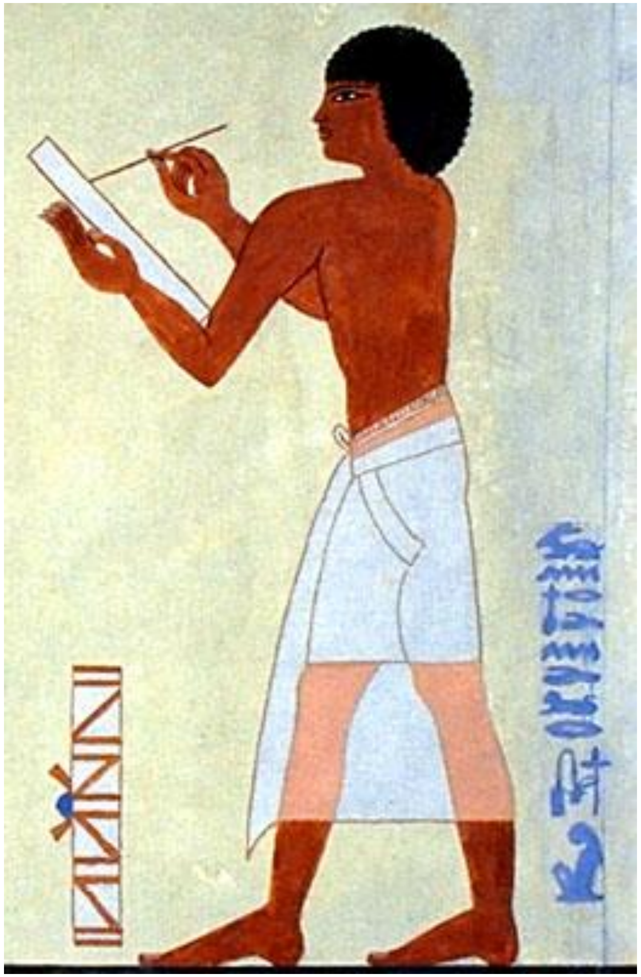
SRS developments at FIT:
SRS experimental setup for detector gain uniformity



2*V6 FECs
(waiting for ZS firmware..)

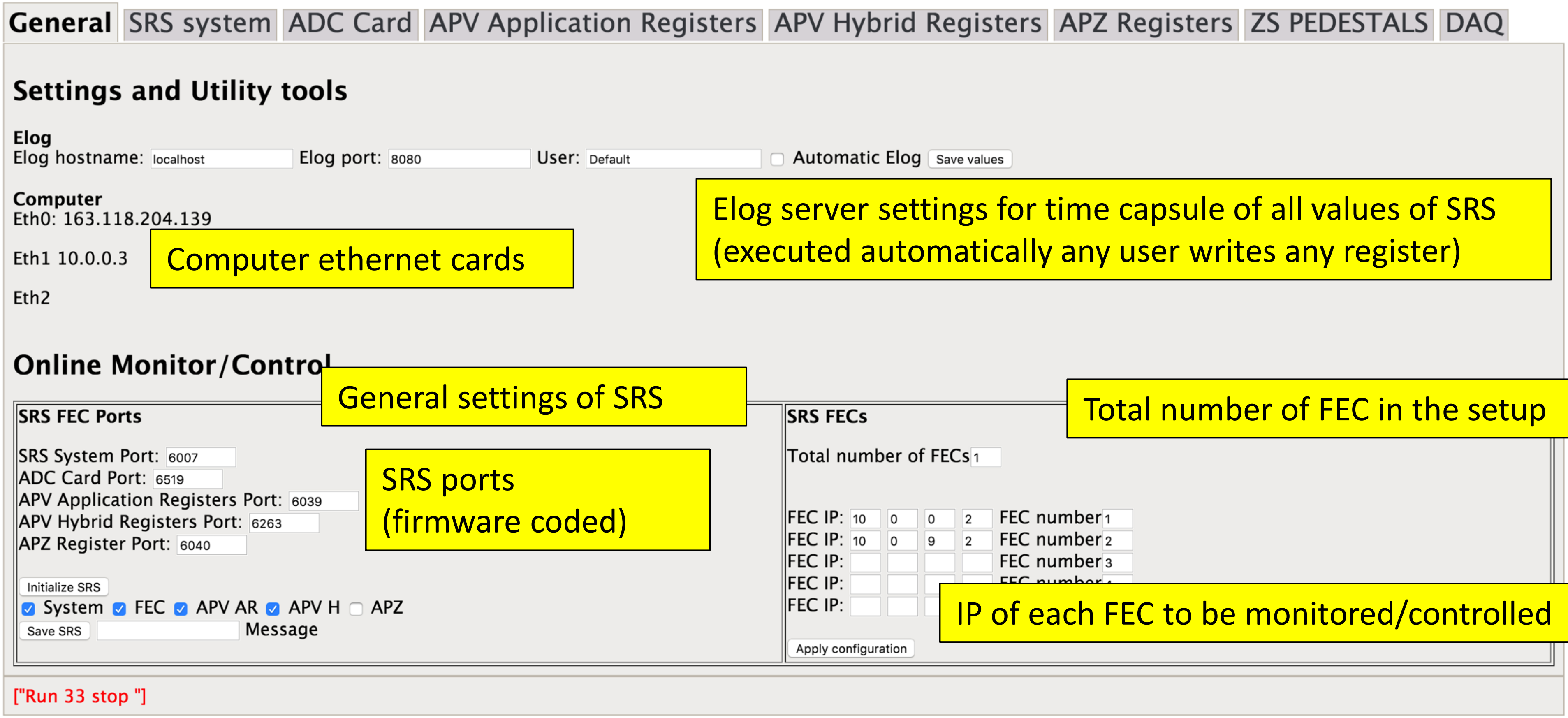
SCRIBE

Slow Control and Run Initialization byte-wise environment



Configuring the SRS

Slow Control & Run Initialization Byte-wise Environment



General | SRS system | ADC Card | APV Application Registers | APV Hybrid Registers | APZ Registers | ZS PEDESTALS | DAQ

Settings and Utility tools

Elog
Elog hostname: localhost Elog port: 8080 User: Default □ Automatic Elog [Save values]

Computer
Eth0: 163.118.204.139

Eth1 10.0.0.3

Eth2

Online Monitor/Control

SRS FEC Ports

SRS System Port: 6007
ADC Card Port: 6519
APV Application Registers Port: 6039
APV Hybrid Registers Port: 6263
APZ Register Port: 6040

[Initialize SRS]
☑ System ☑ FEC ☑ APV AR ☑ APV H □ APZ
[Save SRS] Message

SRS FECs

Total number of FECs 1

FEC IP: 10 0 0 2 FEC number 1
FEC IP: 10 0 9 2 FEC number 2
FEC IP: FEC number 3
FEC IP: FEC number
FEC IP:

[Apply configuration]

["Run 33 stop "]

Elog server settings for time capsule of all values of SRS (executed automatically any user writes any register)

Computer ethernet cards

General settings of SRS

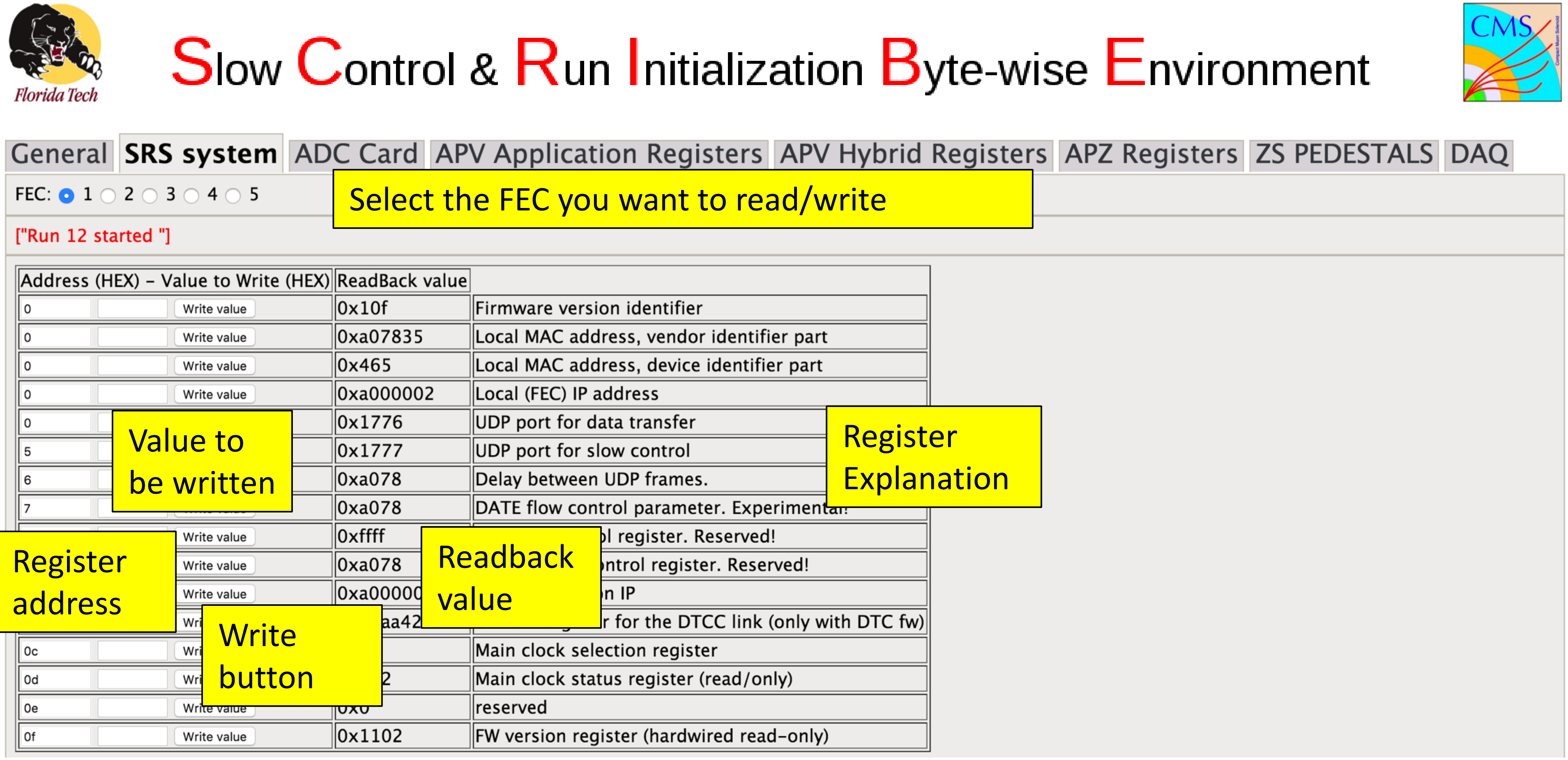
SRS ports (firmware coded)

Total number of FEC in the setup

IP of each FEC to be monitored/controlled

INFO MESSAGE: Configuration, data start/stop/SRS writing into registers, reading etc etc (This is shown on all pages so user knows the status/action takes)

Configuring the SRS



Slow Control & Run Initialization Byte-wise Environment

General | SRS system | ADC Card | APV Application Registers | APV Hybrid Registers | APZ Registers | ZS PEDESTALS | DAQ

FEC: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

["Run 12 started "]

Address (HEX) – Value to Write (HEX)			ReadBack value	
0		Write value	0x10f	Firmware version identifier
0		Write value	0xa07835	Local MAC address, vendor identifier part
0		Write value	0x465	Local MAC address, device identifier part
0		Write value	0xa000002	Local (FEC) IP address
0			0x1776	UDP port for data transfer
5			0x1777	UDP port for slow control
6			0xa078	Delay between UDP frames.
7			0xa078	DATE flow control parameter. Experimental
		Write value	0xffff	l register. Reserved!
		Write value	0xa078	ntrol register. Reserved!
		Write value	0xa00000	n IP
		Wri	aa42	r for the DTCC link (only with DTC fw)
0c		Wri		Main clock selection register
0d		Wri	2	Main clock status register (read/only)
0e		Write value	0x0	reserved
0f		Write value	0x1102	FW version register (hardwired read-only)

Select the FEC you want to read/write

Value to be written

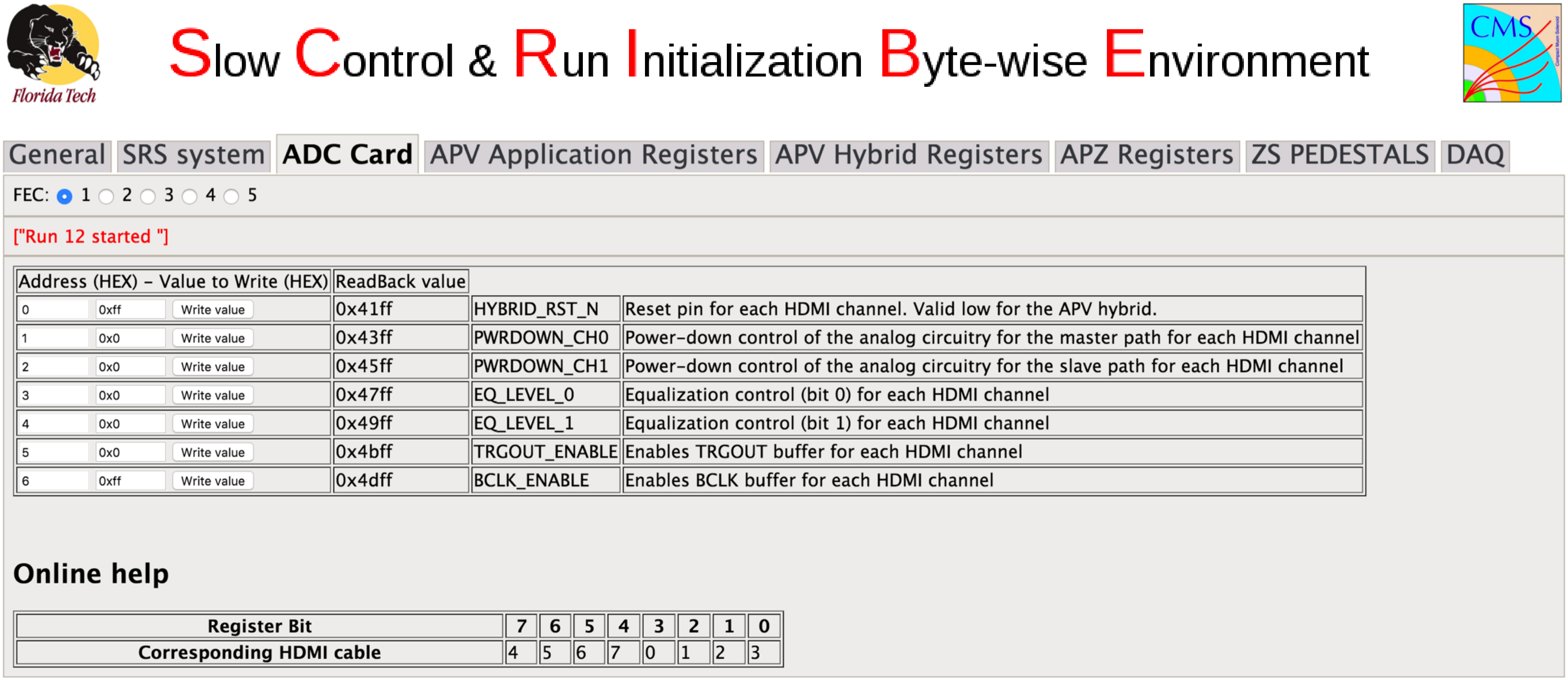
Register Explanation

Register address

Readback value

Write button

Configuring the SRS



Slow Control & Run Initialization Byte-wise Environment

General | SRS system | **ADC Card** | APV Application Registers | APV Hybrid Registers | APZ Registers | ZS PEDESTALS | DAQ

FEC: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

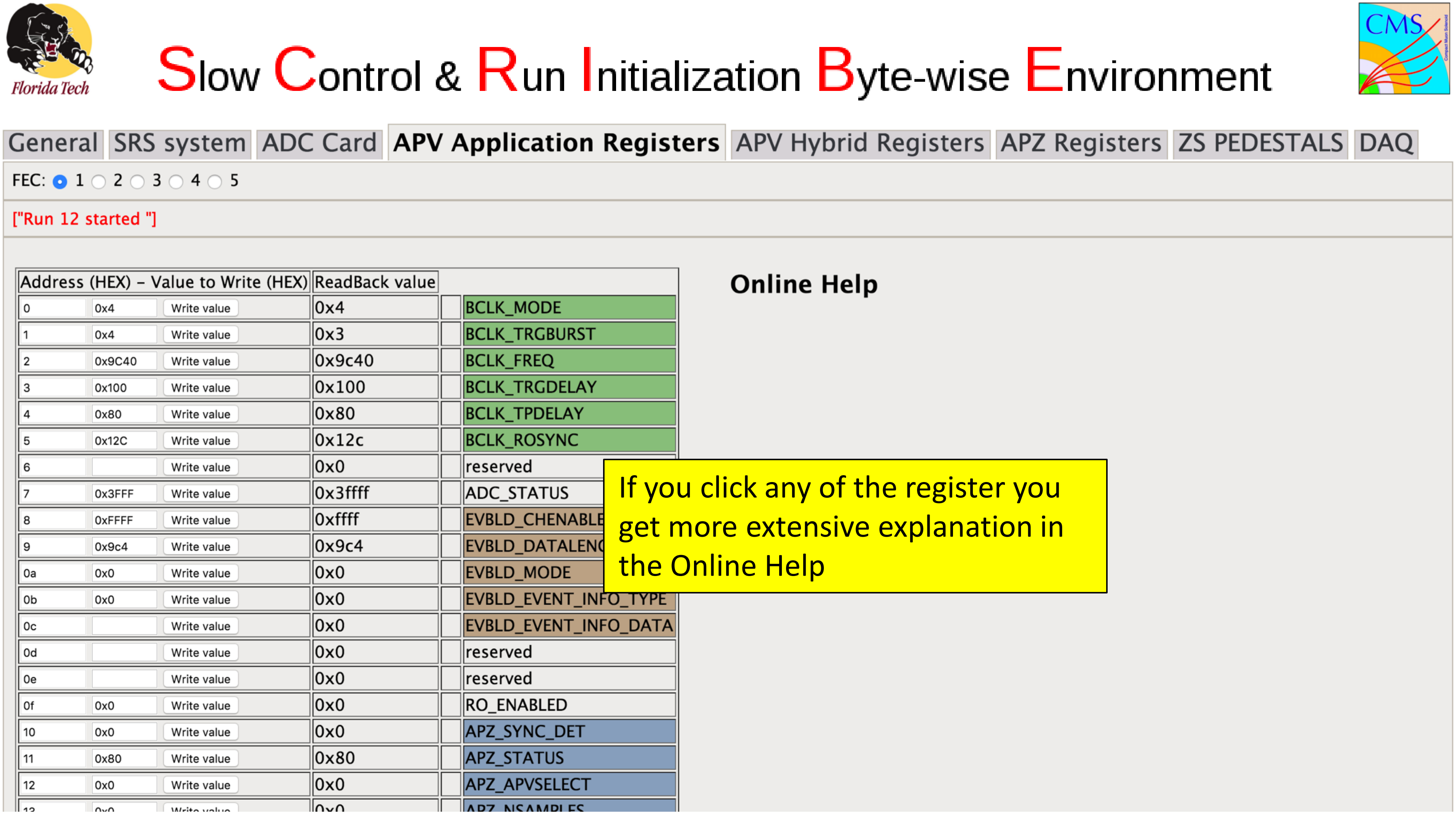
["Run 12 started "]

Address (HEX) – Value to Write (HEX)			ReadBack value		
0	0xff	Write value	0x41ff	HYBRID_RST_N	Reset pin for each HDMI channel. Valid low for the APV hybrid.
1	0x0	Write value	0x43ff	PWRDOWN_CH0	Power-down control of the analog circuitry for the master path for each HDMI channel
2	0x0	Write value	0x45ff	PWRDOWN_CH1	Power-down control of the analog circuitry for the slave path for each HDMI channel
3	0x0	Write value	0x47ff	EQ_LEVEL_0	Equalization control (bit 0) for each HDMI channel
4	0x0	Write value	0x49ff	EQ_LEVEL_1	Equalization control (bit 1) for each HDMI channel
5	0x0	Write value	0x4bff	TRGOUT_ENABLE	Enables TRGOUT buffer for each HDMI channel
6	0xff	Write value	0x4dff	BCLK_ENABLE	Enables BCLK buffer for each HDMI channel

Online help

Register Bit	7	6	5	4	3	2	1	0
Corresponding HDMI cable	4	5	6	7	0	1	2	3

Configuring the SRS



Slow Control & Run Initialization Byte-wise Environment

General | SRS system | ADC Card | **APV Application Registers** | APV Hybrid Registers | APZ Registers | ZS PEDESTALS | DAQ

FEC: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

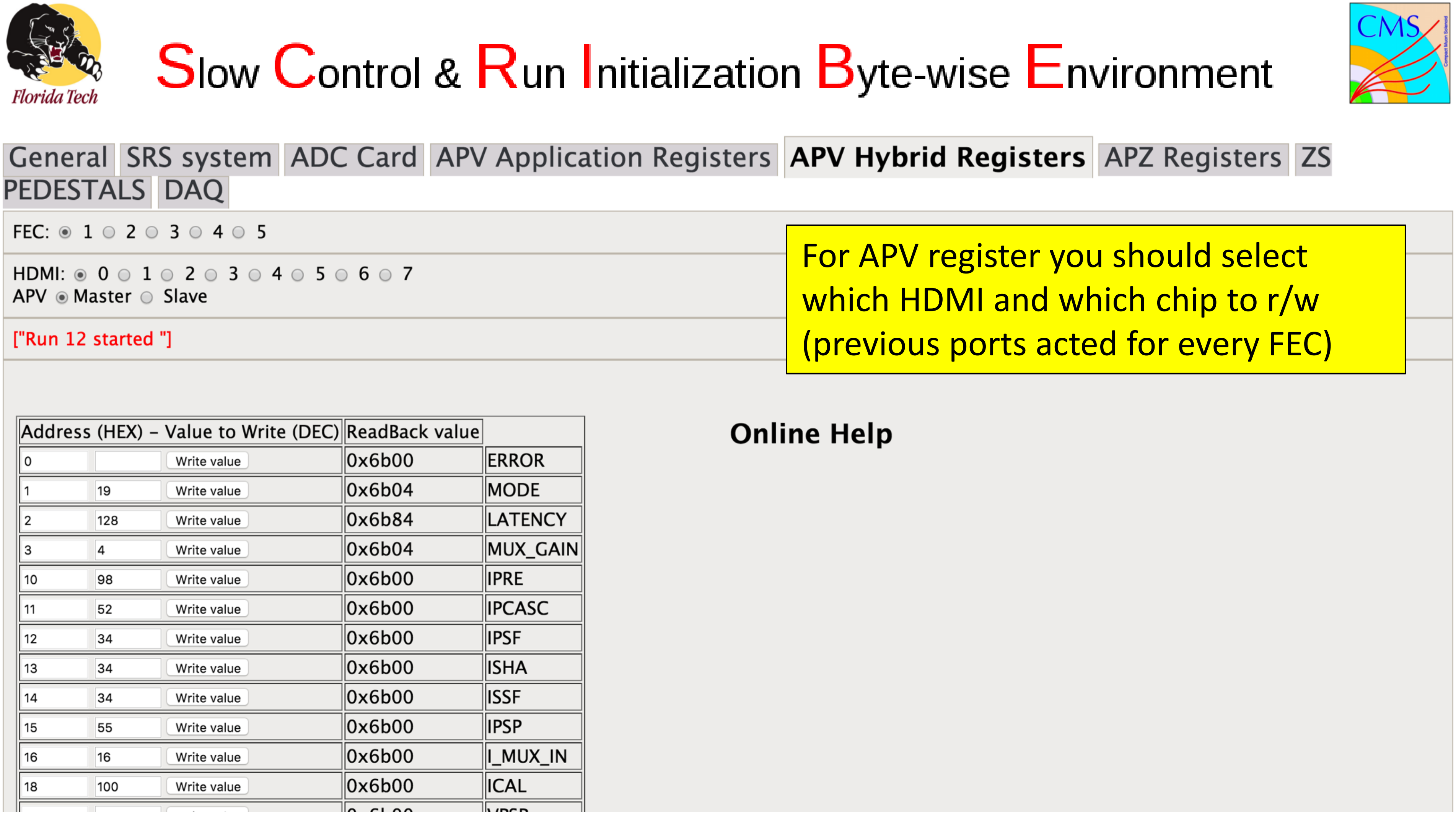
["Run 12 started "]

Address (HEX) – Value to Write (HEX)			ReadBack value		
0	0x4	Write value	0x4	□	BCLK_MODE
1	0x4	Write value	0x3	□	BCLK_TRGBURST
2	0x9C40	Write value	0x9c40	□	BCLK_FREQ
3	0x100	Write value	0x100	□	BCLK_TRGDELAY
4	0x80	Write value	0x80	□	BCLK_TPDELAY
5	0x12C	Write value	0x12c	□	BCLK_ROSYNC
6		Write value	0x0	□	reserved
7	0x3FFF	Write value	0x3ffff	□	ADC_STATUS
8	0xFFFF	Write value	0xffff	□	EVBLD_CHENABLE
9	0x9c4	Write value	0x9c4	□	EVBLD_DATALENG
0a	0x0	Write value	0x0	□	EVBLD_MODE
0b	0x0	Write value	0x0	□	EVBLD_EVENT_INFO_TYPE
0c		Write value	0x0	□	EVBLD_EVENT_INFO_DATA
0d		Write value	0x0	□	reserved
0e		Write value	0x0	□	reserved
0f	0x0	Write value	0x0	□	RO_ENABLED
10	0x0	Write value	0x0	□	APZ_SYNC_DET
11	0x80	Write value	0x80	□	APZ_STATUS
12	0x0	Write value	0x0	□	APZ_APVSELECT
13	0x0	Write value	0x0	□	APZ_NSAMPLES

Online Help

If you click any of the register you get more extensive explanation in the Online Help

Configuring the SRS



Slow Control & Run Initialization Byte-wise Environment

FEC: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

HDMI: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7
APV ● Master ○ Slave

["Run 12 started "]

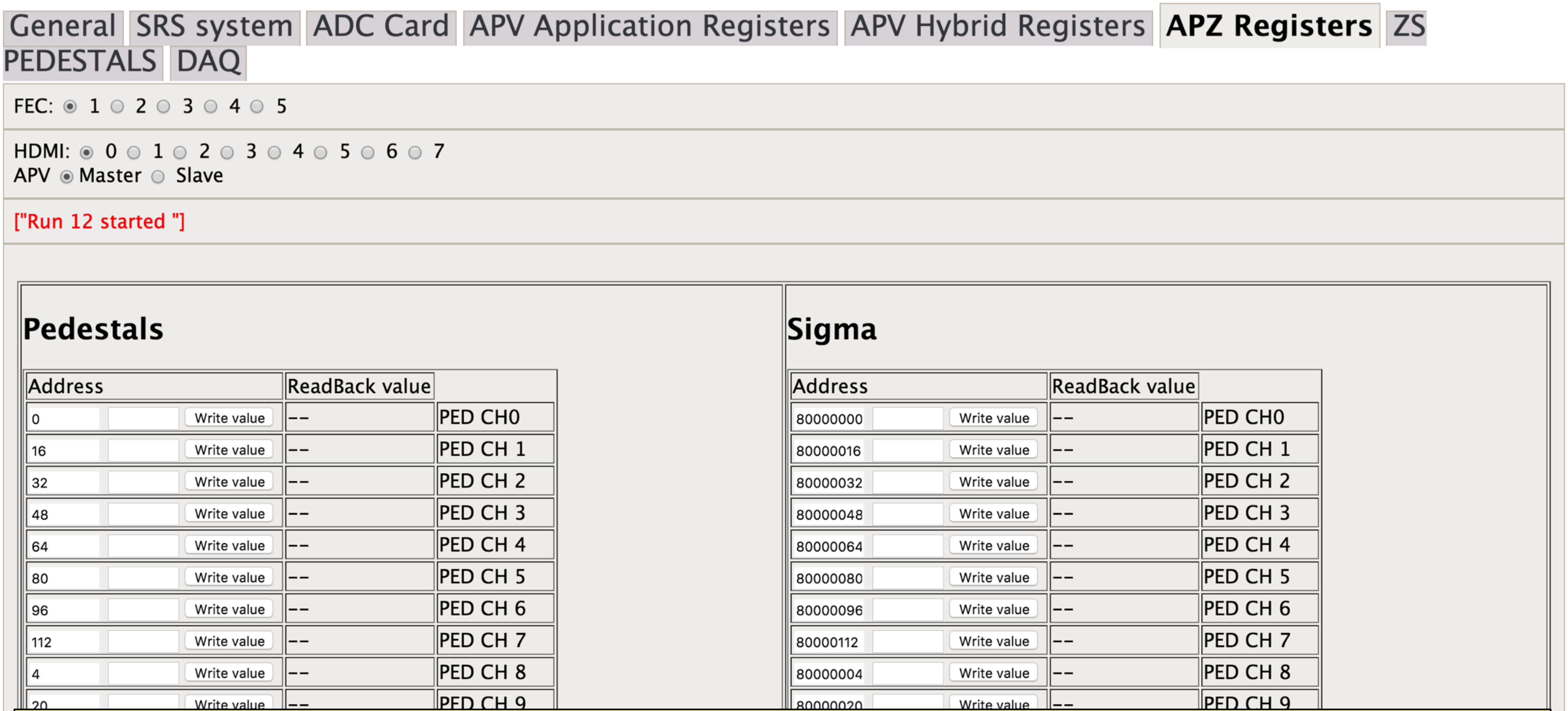
For APV register you should select which HDMI and which chip to r/w (previous ports acted for every FEC)

Address (HEX) – Value to Write (DEC)			ReadBack value	
0		Write value	0x6b00	ERROR
1	19	Write value	0x6b04	MODE
2	128	Write value	0x6b84	LATENCY
3	4	Write value	0x6b04	MUX_GAIN
10	98	Write value	0x6b00	IPRE
11	52	Write value	0x6b00	IPCASC
12	34	Write value	0x6b00	IPSF
13	34	Write value	0x6b00	ISHA
14	34	Write value	0x6b00	ISSF
15	55	Write value	0x6b00	IPSP
16	16	Write value	0x6b00	I_MUX_IN
18	100	Write value	0x6b00	ICAL

Online Help

Configuring the SRS

Slow Control & Run Initialization Byte-wise Environment



FEC: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

HDMI: ● 0 ○ 1 ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7
APV ● Master ○ Slave

["Run 12 started "]

Pedestals

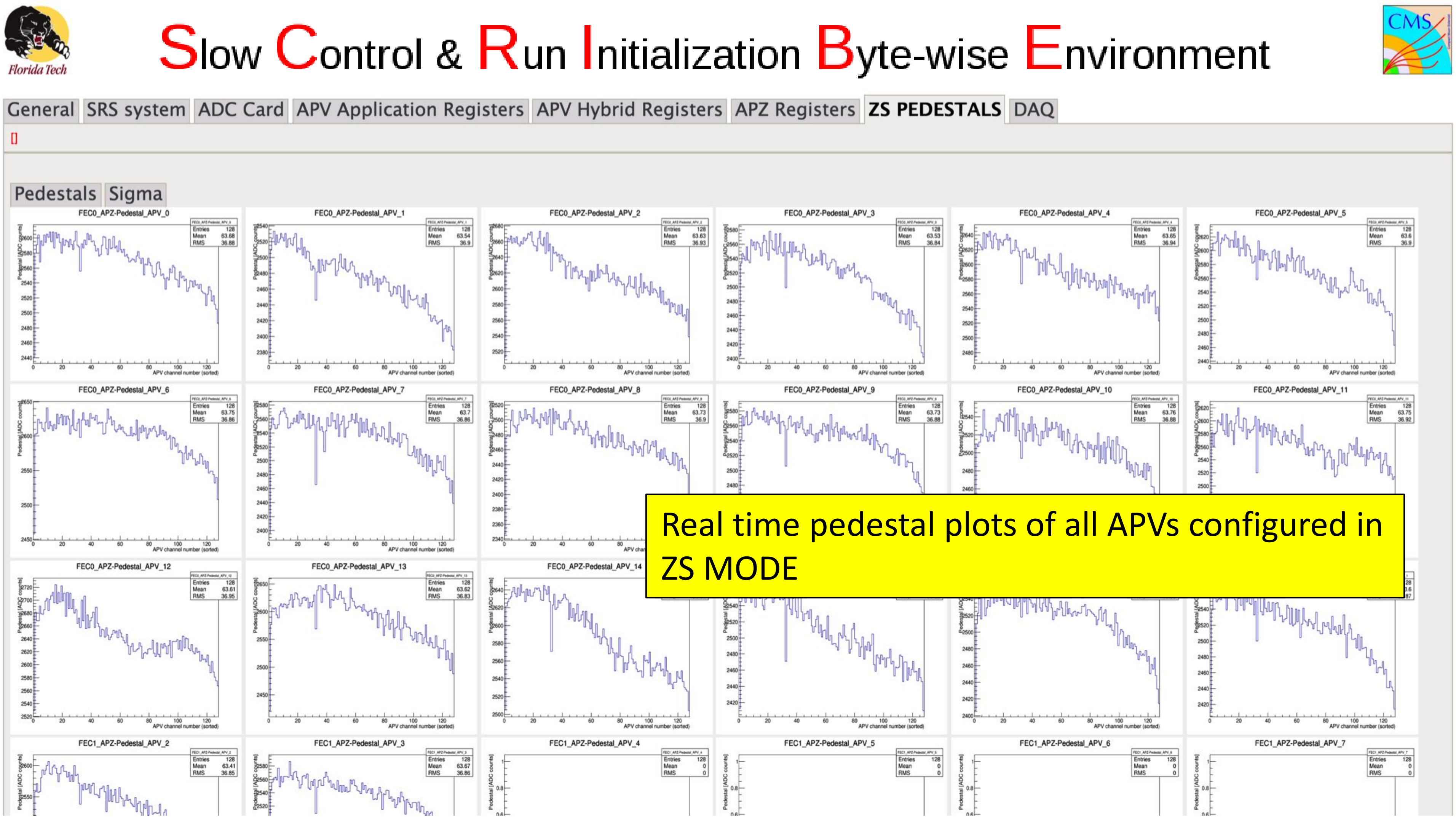
Address			ReadBack value	
0		Write value	--	PED CH0
16		Write value	--	PED CH 1
32		Write value	--	PED CH 2
48		Write value	--	PED CH 3
64		Write value	--	PED CH 4
80		Write value	--	PED CH 5
96		Write value	--	PED CH 6
112		Write value	--	PED CH 7
4		Write value	--	PED CH 8
20		Write value	--	PED CH 9

Sigma

Address			ReadBack value	
80000000		Write value	--	PED CH0
80000016		Write value	--	PED CH 1
80000032		Write value	--	PED CH 2
80000048		Write value	--	PED CH 3
80000064		Write value	--	PED CH 4
80000080		Write value	--	PED CH 5
80000096		Write value	--	PED CH 6
80000112		Write value	--	PED CH 7
80000004		Write value	--	PED CH 8
80000020		Write value	--	PED CH 9

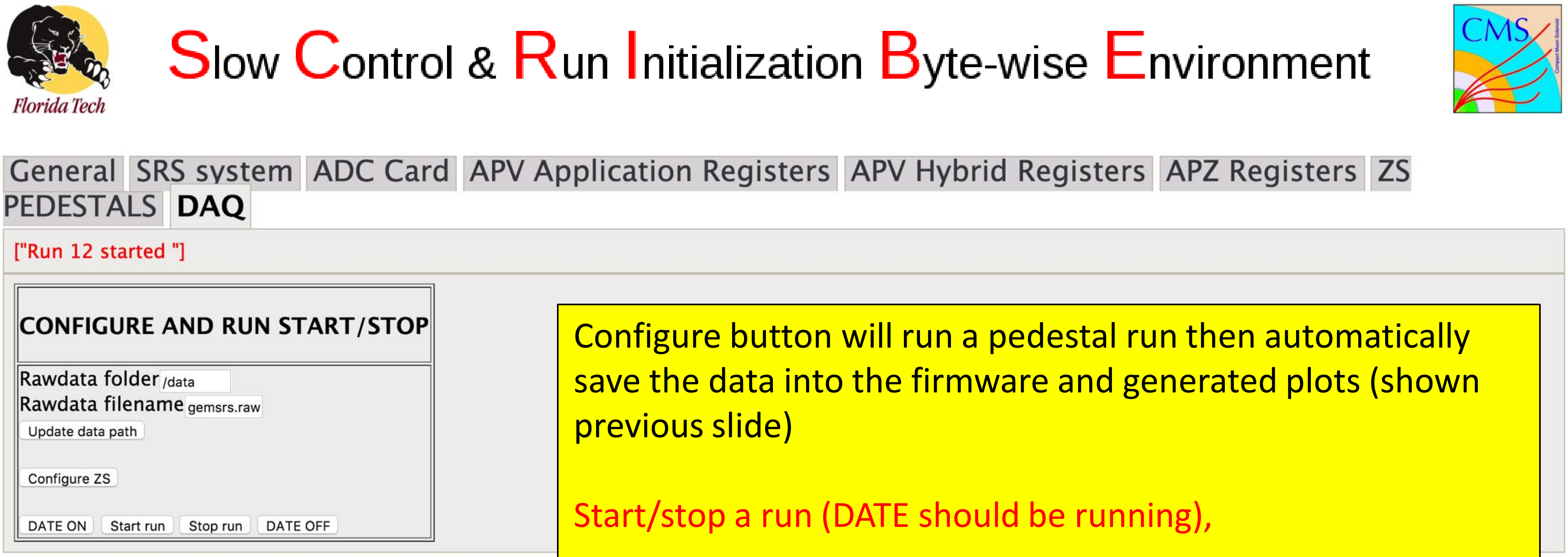
This register holds pedestal data (at firmware level) so output data is zero suppressed,
Here you could mask/unmask any channel on any chip for instance..

Configuring the SRS



Real time pedestal plots of all APVs configured in ZS MODE

Configuring the SRS



Slow Control & Run Initialization Byte-wise Environment

["Run 12 started "]

CONFIGURE AND RUN START/STOP

Rawdata folder /data

Rawdata filename gemsrs.raw

Configure button will run a pedestal run then automatically save the data into the firmware and generated plots (shown previous slide)

Start/stop a run (DATE should be running),

For every run a zip file with register dump of SRS memories and pedestal data saved in the firmware are generated and saved (DB?). Also an elog is generated.

I.E:
gemsrs33.raw
gemsrs33_ped.root
gemsrs_dump.zip

AMORE ZS

AMORE features

AMORE

- Amore framework has been used to perform data analysis of CMS GEM detector (gain uniformity in lab and detector performance at test beams)

  →Already existing knowledge/scripts that we like to continue to use
- We realized that the data unpacking is a small part of the AMORE code, we had the idea of just “rewriting” that part to make AMORE compatible with ZS data
  - Same output files
  - Input files smaller (100x)
  - CPU analysis time faster (>10x?)
  - No pedestal needed
  - DATE acquisition rate 2 FECs 24APVs: 2kHz stable (2,5MB/s room for higher rates..)

An Example for one of the dataset, taken a few days ago

- 100k events (~1min at 2kHz) [no SRU yet]
- 21 time bins
- File dimension: 170MB
- AMORE Analysis rate ~400Hz (~5min CPU time for 100k events)

Conclusions

SCRIBE+AMORE_ZS

- SCRIBE allows:
  - a quick and easy configuration procedure featuring implementation of ZS
  - Monitoring of pedestals uploaded into the FECs, mask/unmask channels
  - Start data-taking and store them sequentially
    - Higher level functionalities easy to add (threshold, HV, latency scans)
- AMORE_ZS allows:
  - Read ZS rawfiles and produce same output files as standard AMORE.
  - Faster analysis time is crucial during detector commissioning

FIT as CMS GEM assembly site is about to start a massive data-taking campaign to fully exploit the developed code, so far very exciting preliminary results!