

Software interface with ICS

Hinko Kocevar
Software Engineer

www.europeanspallationsource.se

23 May 2017

Outline

- Status at PDR
- Work after PDR
- Status at CDR
- Work after CDR

Crew and tasks

- Hardware Rafael A. Baron
- Firmware Fredrik Kristensen (LTH), porting to SIS8300-KU Maurizio Donna
- Software Hinko Kocevar
- Cabling Rafael A. Baron
- Testing / debugging / data evaluation Rafael A. Baron
- Coordination made through JIRA issues
- Small ICS involvement due to lack of resources

Status at PDR

- LLRF EPICS support and IOC ported to work with BPM firmware/hardware
- Demonstrated that chosen hardware and developed firmware are adequate for BPM application
- Presented path forward
 - Stop using NDS based EPICS support
 - Final BPM EPICS support shall be areaDetector based
 - Decision was in sync with ICS

Work after PDR

- Soon after PDR work on final BPM EPICS support started
- First task was to develop generic Struck SIS8300 digitizer support
 - Previously based on NDS
 - Now based on areaDetector (asynNDArrayDriver C++ class)
 - Allows control and status reporting of the digitizer and RTM
 - Delivers 10 channels of digitized analog input data as waveform PVs to clients
- Integration with MTCA based EVR and mrfioc2

Work after PDR

- Once generic SIS8300 support was in place, BPM application development was started
- BPM EPICS support derives C++ class of generic SIS8300
- Along the way generic support was modularized and improved
 - Fairly straight forward since BPM is derived from generic
- All aspects of control and data transfer were completed in 3 months
 - Also with CSS OPI

Work after PDR

- Two instances of BPM are supported
- User controls:
 - Clock source, trigger source, near IQ parameters, BPM FIR filter parameters, select to monitor desired BPM channels, amount of samples to request, monitoring of BPM core status, setup EVR for trigger generation
- Data streams
 - 10 raw digitized data streams or 24 BPM specific data stream (12 per BPM instance)
- AD plugins utilization (ROI, stats, proc, HDF5, ..)

CSS OPI

- Top level CSS OPI
 - General info
 - Acquisition control
 - Access to other OPI
 - RTM selection
 - BPM instance OPI access

BPM on Struck SIS8300 - BPM:

General Info		Collect	
Device type	0x8301	Data type	Float64
Firmware version	0x2808	# AI Samples	300000
Serial number	115	Array counter	0
Memory size	2048 MB	Acquire	Start Stop Done

Plugins		Parameters	
All	File		
Stats	ROI		
Time series	Other		
FFTs			
Update & Refresh			

Channel control		Acquisition control	
Enable / Disable	<input checked="" type="checkbox"/>	Clock source	RTM01
Decim. Factor	0	Clock divider	1
Decim. Offset	0	Trigger source	Software
Conv. Factor	0	Trigger line	FrontPanel0
Conv. Offset	0	Trigger delay	0
Channels	Channels	Trigger repeat	0

RTM control		Debugging	
RTM type	DWC8300-LF	Message [INF] SIS8300::sisTask: No error	
RTM temp. 1	26.7 C		
RTM temp. 2	31.5 C		
Temperature refresh			

Bpm

BPM common BPM1 BPM2

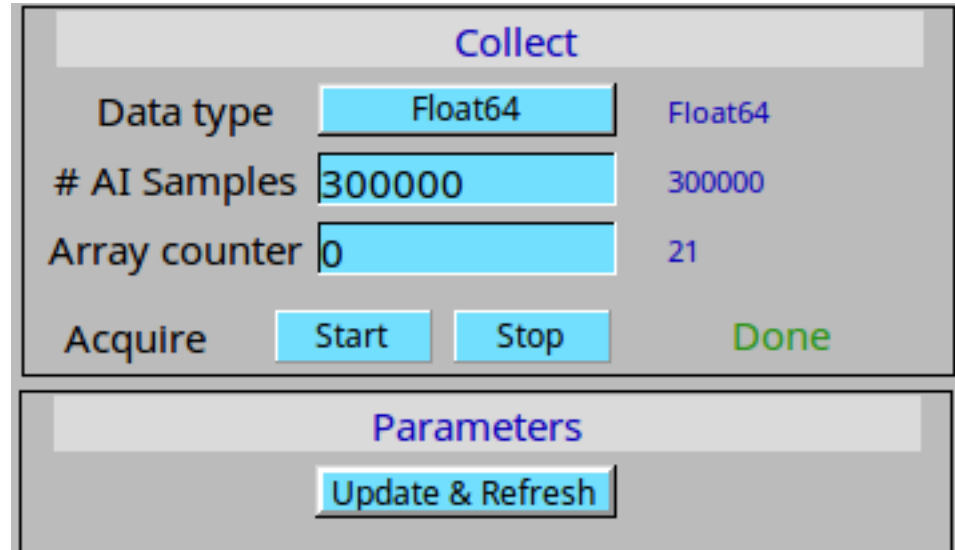
CSS OPI general info

- Some generic SIS8300-x digitizer information

General info		
Device type	0x8303	SIS8300-KU
Firmware version	0x28E0	
Serial number	6	
Memory size	2048 MB	

CSS OPI acquisition control

- Selectable data type
- Desired number of raw samples
- Number of arrays (pulses) collected so far
- Force parameter update
 - FPGA parameters are latched and automatically applied between the pulses



The screenshot displays a software interface for data acquisition control, divided into two main sections: 'Collect' and 'Parameters'.

Collect Section:





- Data type:** A dropdown menu is set to 'Float64', with the current selection 'Float64' displayed to the right.
- # AI Samples:** A text input field contains '300000', with the value '300000' also displayed to the right.
- Array counter:** A text input field contains '0', with the value '21' displayed to the right.
- Acquire:** A label followed by two buttons: 'Start' and 'Stop'.
- Done:** A green button located to the right of the 'Start' and 'Stop' buttons.

Parameters Section:

- Update & Refresh:** A button located below the 'Collect' section.


CSS OPI analog channels

- Channel enable/disable
- Software scaling and offset control
- Attenuation control (RTM)

Channel	Status	Decimation		Conversion		Attenuation	
Ch0		Factor	<input type="text" value="1"/> 1	<input type="text" value="1.00000000"/> 1.00000000	<input type="text" value="0.0"/> 0.0	<input type="text" value="0.0"/> 0.0	<input type="button" value="Plots"/>
		Offset	<input type="text" value="0"/> 0	<input type="text" value="0.000"/> 0.000			
Ch1		Factor	<input type="text" value="1"/> 1	<input type="text" value="1.00000000"/> 1.00000000	<input type="text" value="0.0"/> 0.0	<input type="text" value="0.0"/> 0.0	<input type="button" value="Plots"/>
		Offset	<input type="text" value="0"/> 0	<input type="text" value="0.000"/> 0.000			
Ch2		Factor	<input type="text" value="1"/> 1	<input type="text" value="1.00000000"/> 1.00000000	<input type="text" value="0.0"/> 0.0	<input type="text" value="0.0"/> 0.0	<input type="button" value="Plots"/>
		Offset	<input type="text" value="0"/> 0	<input type="text" value="0.000"/> 0.000			
Ch3		Factor	<input type="text" value="1"/> 1	<input type="text" value="1.00000000"/> 1.00000000	<input type="text" value="0.0"/> 0.0	<input type="text" value="0.0"/> 0.0	<input type="button" value="Plots"/>
		Offset	<input type="text" value="0"/> 0	<input type="text" value="0.000"/> 0.000			

Channel control

Enable / Disable



Decim. Factor

Decim. Offset

Conv. Factor

Conv. Offset

Channels

CSS OPI acquisition control

- Selectable clock source
 - backplane, RTM, front panel, internal
- Clock divider
 - integer divisor
- Trigger repeat
 - how many pulses to acquire
- Others
 - not applicable to BPM at this stage

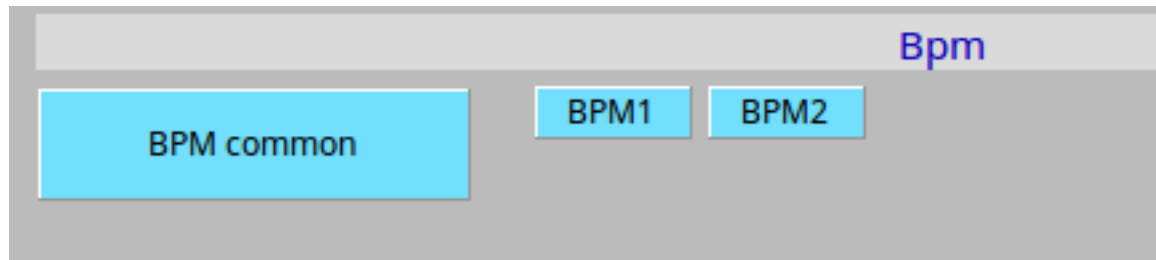
Acquisition control		
Clock source	<input type="text" value="RTM01"/>	RTM01
Clock divider	<input type="text" value="1"/>	1
Trigger source	<input type="text" value="Software"/>	<input type="text" value="Software"/>
Trigger line	<input type="text" value="FrontPanel0"/>	<input type="text" value="FrontPanel0"/>
Trigger delay	<input type="text" value="0"/>	<input type="text" value="0"/>
Trigger repeat	<input type="text" value="-1"/>	-1

CSS OPI RTM controls

- One type of RTM
 - At least for BPM
- RTM temperature sensors readout
 - Over PCIe
 - Slow conversion (~ 150 ms)
 - Better to use IPMI which should cover other sensors, too

RTM control	
RTM type	DWC8300-LF
RTM temp. 1	30.7 C
RTM temp. 2	35.1 C
Temperature refresh	

- Two instances of BPM DSP on single AMC
 - Instances are pretty much independent, but..



- Some controls are common for both instances
 - Trigger source, clock source, decimation,..





CSS OPI common BPM

- Collection of common BPM controls and statuses
 - These apply to both BPM instances
 - Snag: how to name them in control room?

Register read error	<input type="checkbox"/>	
Register write error	<input type="checkbox"/>	
Trigger setup	<input type="text" value="MLVDS 1,2"/>	<input type="text" value="MLVDS 1,2"/>
Pulse done	<input type="checkbox"/>	
Pulse count	<input type="text" value="562866"/>	
Pulse missed	<input type="text" value="0"/>	
Num samples	<input type="text" value="125841"/>	
Num IQ samples	<input type="text" value="8389"/>	
Num BPM samples	<input type="text" value="8384"/>	
Near IQ M	<input type="text" value="4"/>	<input type="text" value="4"/>
Near IQ N	<input type="text" value="15"/>	<input type="text" value="15"/>
Mem Mux	<input type="text" value="BPM1 AI, BPM INT"/>	<input type="text" value="BPM1 AI, BPM INT"/>
Mem Mux 10	<input type="text" value="POSITIONS"/>	<input type="text" value="POSITIONS"/>
Filter Control	<input type="text" value="Disabled"/>	<input type="text" value="Disabled"/>
Filter Coeff 0	<input type="text" value="0.00720997664"/>	<input type="text" value="0.007209976643"/>
Filter Coeff 1	<input type="text" value="-0.0019530947"/>	<input type="text" value="-0.001953094774"/>
Filter Coeff 2	<input type="text" value="-0.0530600372"/>	<input type="text" value="-0.053060037296"/>
Filter Coeff 3	<input type="text" value="0.00564759007"/>	<input type="text" value="0.005647590077"/>
Filter Coeff 4	<input type="text" value="0.29573114815"/>	<input type="text" value="0.295731148153"/>
Filter Coeff 5	<input type="text" value="0.49227318439"/>	<input type="text" value="0.492273184395"/>
Filter Gain		<input type="text" value="0.000000000000"/>
Filter Apply	<input type="button" value="Apply"/>	
Self Triggering	<input type="text" value="Disabled"/>	<input type="text" value="Disabled"/>
Self Triggering Threshold	<input type="text" value="0"/>	<input type="text" value="0"/>
Self Triggering Num sampels	<input type="text" value="0"/>	<input type="text" value="0"/>
Self Triggering Reference Ch.	<input type="text" value=""/>	<input type="text" value=""/>

CSS OPI BPM counters

- FPGA access status
- Trigger line select
 - Always backplane
- Sample / pulse counters
 - # pulses
 - Raw
 - IQ
 - BPM

Register read error		
Register write error		
Trigger setup		MLVDS 1,2
Pulse done		
Pulse count		562866
Pulse missed		0
Num samples		125841
Num IQ samples		8389
Num BPM samples		8384

CSS OPI BPM near IQ

- Near IQ sampling control

Near IQ M	<input type="text" value="4"/>	<input type="text" value="4"/>
Near IQ N	<input type="text" value="15"/>	<input type="text" value="15"/>

CSS OPI BPM stream muxes

- Using suboptimal BPM data streams at the moment
 - Limited number of data stream channels
 - Need to specify which data stream we want to have from FPGA
 - Muxes allow both BPM data to be interleaved in data streams
 - If becomes software bottleneck then firmware needs to be improved



CSS OPI BPM filter

- Filter control enable/disable
- FIR coefficients
- Resulting gain
- Manual coefficient application

Filter Control	<input type="button" value="Disabled"/>	<input type="button" value="Disabled"/>
Filter Coeff 0	<input type="text" value="0.00720997664"/>	<input type="text" value="0.007209976643"/>
Filter Coeff 1	<input type="text" value="-0.0019530947"/>	<input type="text" value="-0.001953094774"/>
Filter Coeff 2	<input type="text" value="-0.05306003729"/>	<input type="text" value="-0.053060037296"/>
Filter Coeff 3	<input type="text" value="0.00564759007"/>	<input type="text" value="0.005647590077"/>
Filter Coeff 4	<input type="text" value="0.29573114815"/>	<input type="text" value="0.295731148153"/>
Filter Coeff 5	<input type="text" value="0.49227318439"/>	<input type="text" value="0.492273184395"/>
Filter Gain		<input type="text" value="0.000000000000"/>
Filter Apply	<input type="button" value="Apply"/>	

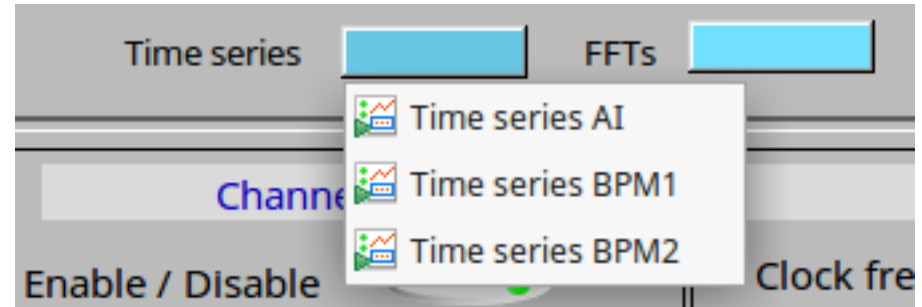
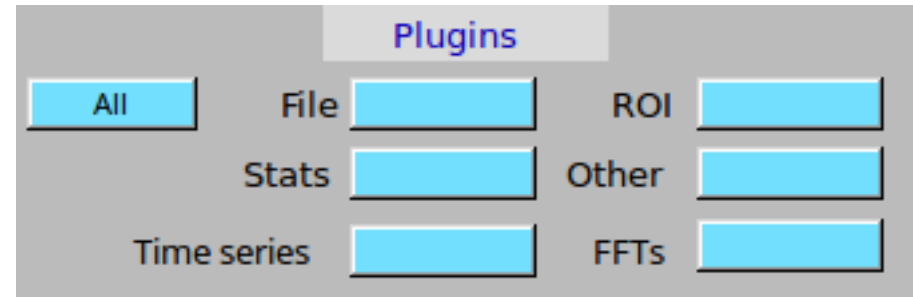
CSS OPI self-triggering

- Self triggering instead of backplane trigger
 - When fishing for pulse
 - Debugging timing (triggers)
 - No timing, no triggers

Self Triggering	<input type="button" value="Disabled"/>	<input type="button" value="Disabled"/>
Self Triggering Threshold	<input type="text" value="0"/>	<input type="text" value="0"/>
Self Triggering Num sampels	<input type="text" value="0"/>	<input type="text" value="0"/>
Self Triggering Reference Ch.	<input type="radio"/>	

CSS OPI AD plugins

- AreaDetector provides plugins
 - Chain data stream/array
 - Source can vary, so can destination
- Many provided out of the box
 - Statistics, processing, time series, ROI, saving to file,...
- Can develop own plugins
 - Already done for other systems



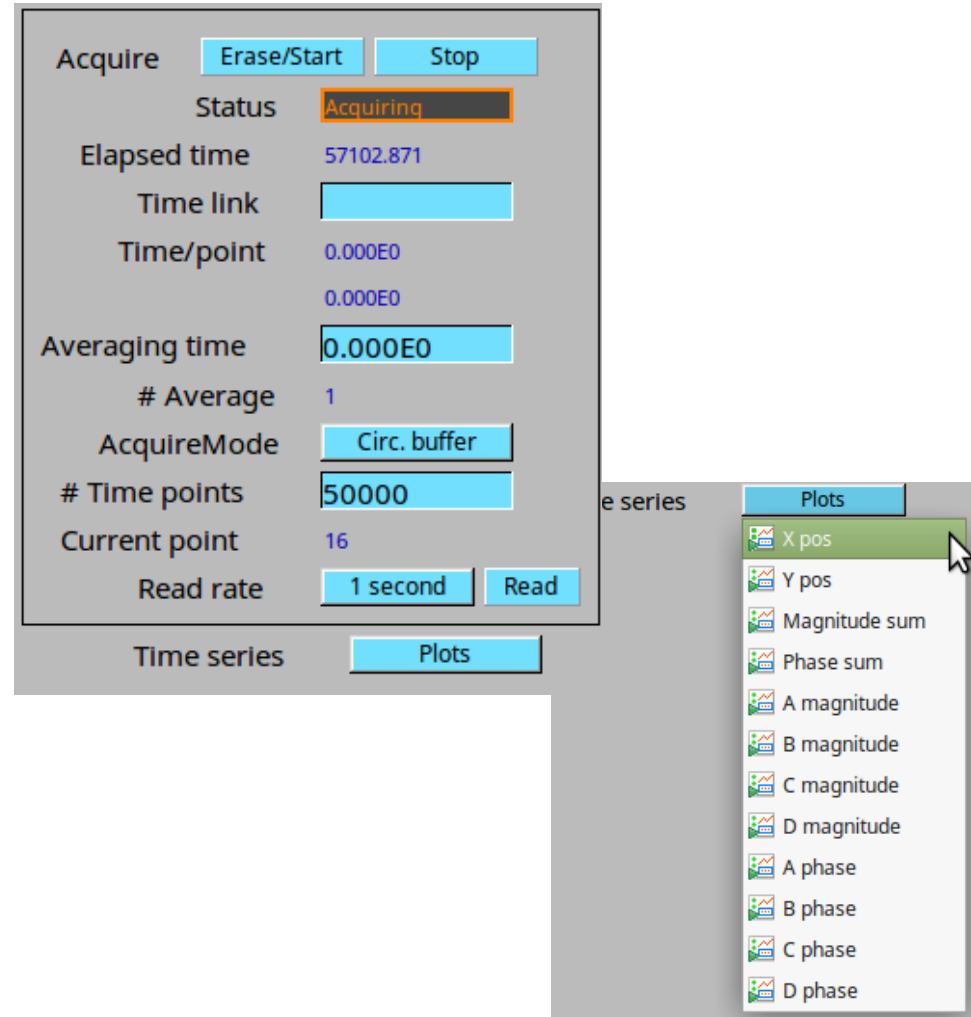
CSS OPI AD plugin setup

- Generic for all AD plugins
- Define data source
- Monitor array arrival to the plugin
- Inspect array properties
- Control publishing resulting array to other plugins

asyn port	TS1		
Plugin type	NDPluginTimeSeries		
Array port	BPM	BPM	
Array address	1	1	
Enable	Enable	1	
Min. time	0.000	0.000	
Callbacks block	No	0	
Queue size/free	20	20	
Array counter	0	583224	
Array rate	14.00		
Dropped arrays	0	0	
# dimensions	2		
Array Size	14	8384	0
Data type	Float64		
Color mode	Mono		
Bayer pattern	RGGB		
Unique ID	1749670		
Time stamp	864372827.458		
Attributes file			
Array callbacks	Enable	1	
asyn record			

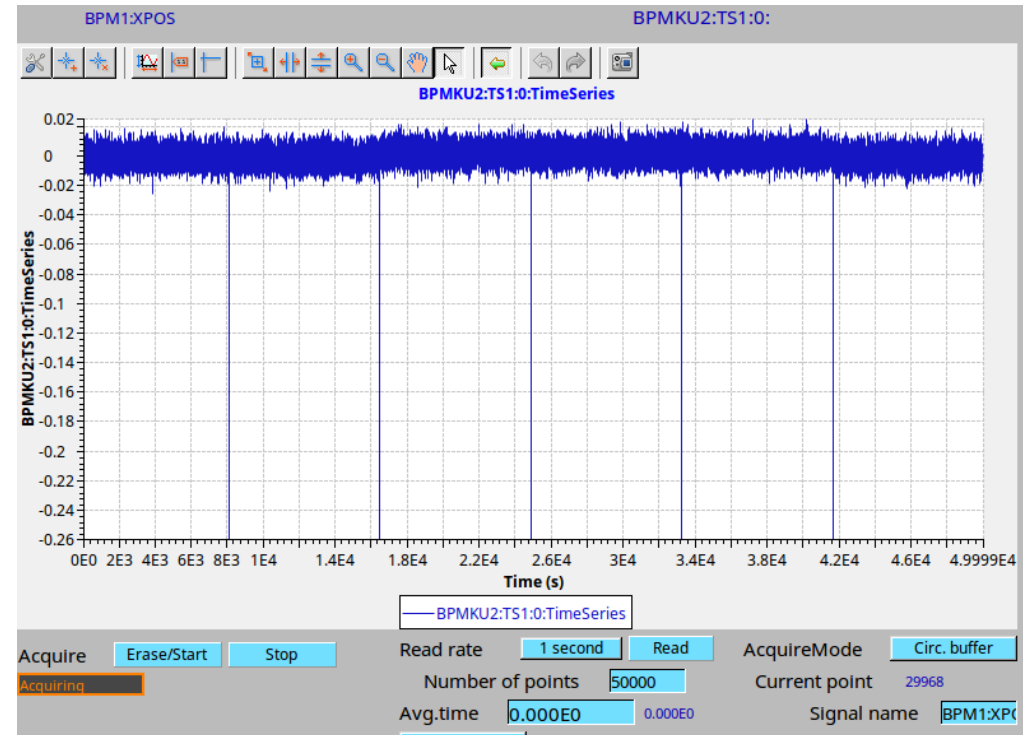
CSS OPI time series plugin control

- Control acquisition
- Averaging
- Mode
 - Circular buffer or one shot
- Number of points in the buffer
- Publishing rate
- Access to individual plots



CSS OPI time series plot

- Plot of BPM1 X position
- Holds ~6 pulses
- Glitches due to firmware



Status at CDR

- Debugged and tested with several different SIS8300 boards and board flavors
- Updated EPICS support with latest firmware features
- Works with final BPM RTM
- Works with final BPM AMC (SIS8300-KU)
- Several over-the-weekend run tests passed
-

Work after CDR

- Handover the EPICS support for SIS8300 and BPM to ICS
- Perform integration with multiple SIS8300 AMCs in the same crate
- Long term and stability tests (IPMI IOC)
- Python scripts for automated tests and reports
- Determine minimum set of PVs and amount of data to be delivered (as per BP requirements)

Work after CDR

- Add recently added FPGA parameters
- Solve some outstanding firmware issues (glitches seen in data)
- Assess performance of the CPU with multiple AMCs / BPMs
- Test, test, test..

Questions ?