SYNCHRONIZATION OF MOTION AND DETECTORS AND CONTINUOUS SCANS AS THE STANDARD DATA ACQUISITION TECHNIQUE

D. Fernández-Carreiras, F. Becheri, T. Coutinho, G. Cuní, R. Homs, G. Jover-Mañas, J. Klora [on leave], O. Matilla, J. Moldes, C. Pascual-Izarra, Z. Reszela, D. Roldan, S. Rubio-Manrique, X. Serra, ALBA-CELLS, Barcelona, Spain

Abstract

Alba [1] is a third generation synchrotron located near Barcelona in Spain. This paper describes the model, objectives and implementation of a generic data acquisition structure for an experimental station, which integrates the hardware and software synchronization of motors, detectors, shutters and in general any experimental channel or events related with the experiment. The implementation involves the management of hardware triggers, which can be derived from time, position of encoders or other sources such as events from the particle accelerator, combined with timestamps for guaranteeing the correct integration of fast triggered or slow software channels. The infrastructure requires a complex management of buffers of different sources, centralized and distributed, including interpolation procedures. ALBA uses Sardana [2][3] built on TANGO[4] as the generic control system for the accelerators and beamlines, which provides the abstraction and communication with the hardware, human-machine interfaces and a complete macro edition and execution environment.

THE CONTROL SYSTEM OF A BEAMLINE

A beamline of a synchrotron light source or a neutron source has specific requirements in terms of control and data acquisition. Since optics used in X-ray or neutron setups needs movable elements, the data acquisition is often coupled to a motion of an axis or a number of axes. During the motion, the detectors take data synchronized at a given number of points or intervals. Traditionally the detectors take data with the motors stopped and wait for the motors to finish the subsequent motion to start again. This is changing to the so-called continuous scans.

Once new experiments are approved, they are scheduled on beamlines typically for a week, although they can be as short as one day or less. Preparing the beamline for a new experiment, requires typically to include new hardware, sample environment, motion, detectors and synchronization. Dismounting one experiment and setting up the new one is frequently a complex task, which often has to be completed few hours –the so called "machine day"- and usually requires to reconfigure the control software and write new sequences. Besides, recurrent tasks in cases particularly complex such as the alignment of the optics, frequently entail to be automated. According to these requirements, a Beamline needs a modular control and data acquisition system, with wide-ranging building blocks for Graphical Interfaces, hardware configuration and data acquisition. And above all, a flexible and customizable macro execution environment is a critical success factor for the experiment control and data acquisition system. A simplified block diagram of the Sardana core is shown in Fig. 1. Much more details are given in the reference [2] and in the official web site [3].

The macro execution and edition environment is one of the strongest points of Sardana and where the efforts converge. Macros are python classes organized in a comprehensive way to optimize the development time and diminish the learning curve for scientists without an IT background. However, the extensive standard macro library covers a broad number of setups and configurations for which no extra code is required.

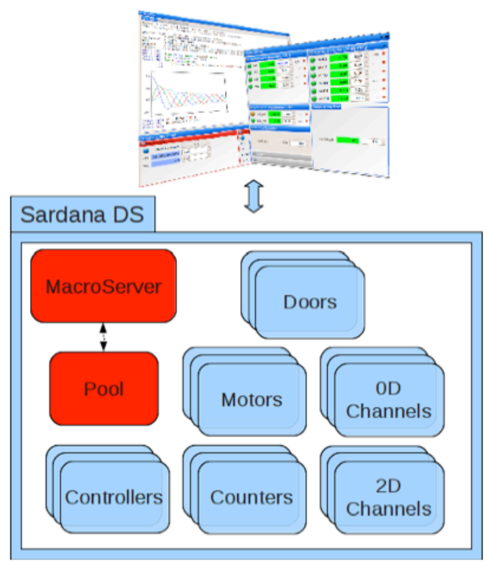


Figure 1: Block diagram of the Sardana Server and its components.

THE CONTINUOUS SCAN AS A STANDARD

A continuous scan, known also by other names as "fly scan" or "quick scan", consists on moving a motor or set of motors and acquiring with the different detectors a number of times for a certain acquisition time during the motion. The overall idea of a continuous scan is to be as flexible and adaptable as a step scan, where the user can directly select an arbitrary number of motors and

ISBN 978-3-95450-139-7

detectors with a number of intervals and exposure time for every scan.

In order to make it standard and generic, a continuous scan shall be as close as possible to the step scan, in both configuration and execution aspects. Though, the complexity of the continuous scan is much higher due to synchronization, speed and data buffering.

The first obstacle to overcome is to allow any combination of “movable elements”, typically motors and pseudomotors to move synchronously on a given trajectory. Pseudomotors and pseudocounters are logical elements (movable or detectors) created from a combination of physical elements (e.g. motors, counters, or detectors) and arithmetic operations. Then, during the motion, the detectors shall take data synchronized with the motion. In this context, detectors mean any arbitrary combination of scalars such as counters, pseudocounters, and zero-dimensional (0D), one-dimensional (1D) such as Multi Channel Analyzers (MCA), Position Sensitive Detectors (PSD), and two-dimensional (2D), i.e. CCD cameras or pixel detectors.

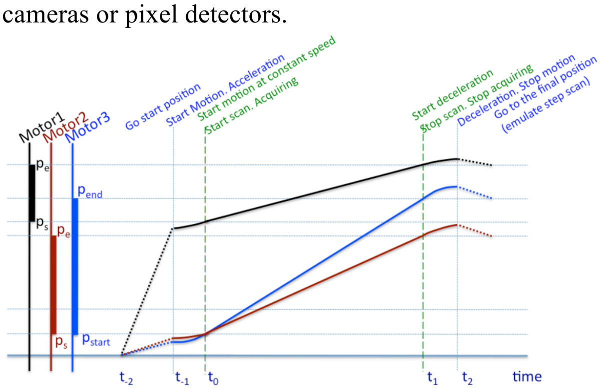


Figure 2: Position represented versus time of a set of synchronized motors during a continuous scan.

When considering linear motions, the motors shall start moving before the start of the scan, in order to allow beginning the acquisitions of the scan at the desired point at constant speed. Motors and pseudomotors shall have a configurable acceleration and speed. In the case of the pseudomotors, setting both acceleration and speed in the final units is tricky, needing a cascade to the implicated motors. The process is shown in Fig. 2 where t_{-2} represents the initial state, t_{-1} is the initial position, foreseeing the time to accelerate. t_0 is the actual starting point of the scan, where all motors are synchronized and at constant speed whereas t_1 is the final point, where the last acquisition is done and the motors begin to decelerate. In a traditional step scan, t_0 and t_1 would correspond to the start and end point of the scan. However, in the continuous scan the motion needs to start before and finish after the final acquisition with the consequent overshooting. In order to keep compatibility with the step scan, the motors return to the end position and restore the preset velocity and acceleration.

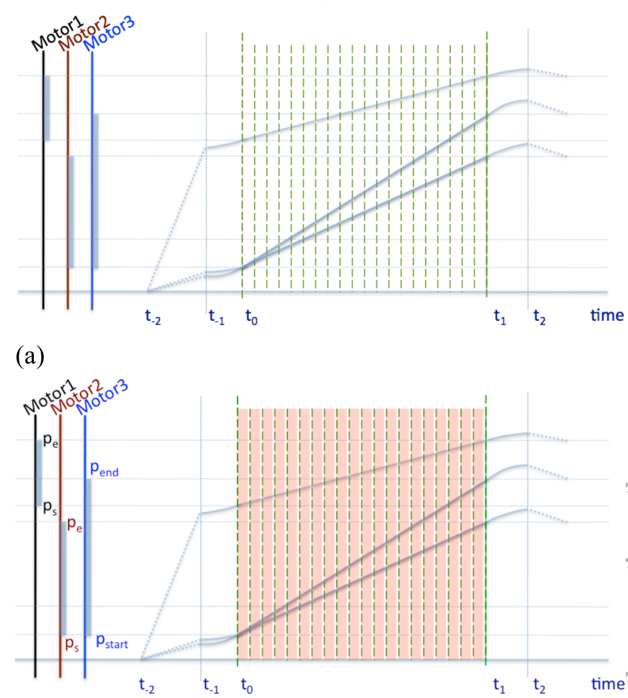
TRIGGERING AND BUFFERING

A continuous scan usually needs a synchronization of different detectors combined with data interpolation from additional slow channels.

The data acquisition takes place during the scan at synchronized intervals. The synchronization can happen at equidistant time intervals, naturally assuming a constant speed in all movable axes, or at arbitrary defined positions of the master axes.

A trigger object manages the triggers signals computed from the source, either from a counter-timer device or from encoder or an indexer of an axis or set of axes. The detectors intervening in this particular data acquisition of a scan shall be configured accordingly. Typically a trigger can be a pulse or a gate. The pulse indicates the start of the acquisition and the detector manages the acquisition time that has been preset. The gate indicates the start and duration of the acquisition time. Slow process variables, such as temperatures, diagnostics etc., not triggered, could cohabit with the synchronized detectors.

This generic setup needs triggering and intelligent buffering capabilities. The buffer handles fast detectors with or without internal memory, slow channels, interpolation of data taken asynchronously or at slower rate, and rolling-buffers for large scans.



(a)

(b)

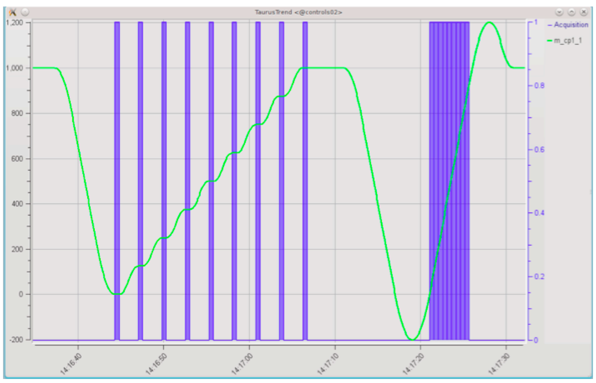
Figure 3: (a) Representation of triggers at equidistant time (and position assuming a constant speed). (b) Acquisition (live) time of channels and detectors intervening in the scan.

In a step scan, the last point coincides with the last acquisition, following the sequence move-count. As shown in Fig. 3.b in the case of a continuous scan, the move-count sequence is executed in parallel, so the acquisition starts from the beginning of the motion at

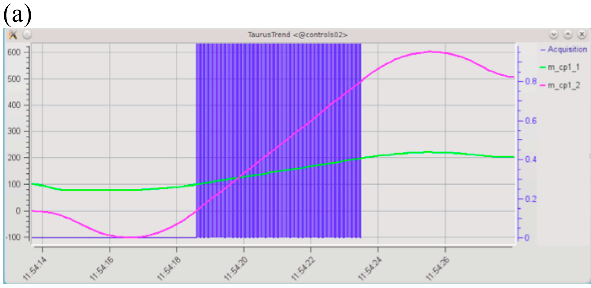
ISBN 978-3-95450-139-7

constant speed, having completed one acquisition when reaching the first point and having completed the last acquisition when reaching the last point.

Figure 4 shows the empirical tests of the continuous scans. The first graph represents the step scan followed by the continuous scan with the noticeable gain in time. The second graph shows a 2-axes continuous scan. Note the overshoots of the motions in the continuous scan for allowing acceleration and deceleration before and after the data acquisition [5].



(a)



(b)

Figure 4: (a) Step scan followed by a continuous scan. (b) Continuous scan with 2 participant axes.

PRECISE TIMESTAMPING AS THE GENERIC REQUIREMENT

One crucial requirement to handle for the next generation continuous scans is the introduction of a precise timestamp. Depending on the desired precision and on the particular setup, timestamps may be synchronized by hardware or by software. Software synchronized times can have enough precision for a large number of setups. Network Time Protocol (NTP) and Precision Time Protocol (PTP) can reach 1 digit microsecond precision or better, although most old implementations of NTP are often below the millisecond range. In the case of Alba, we use timestamps generated by hardware from the central timing system [6] (MRF) [7] and we use a NTP-based software time synchronization for all computers, back end servers and frontends.

Most experimental stations of a synchrotron or neutron source require synchronization. The accuracy required depends on the particular station or experiment, but typically a single digit microsecond is in most cases sufficient.

For example, Alba's X-ray absorption beamline (BL22) is equipped with a direct drive double crystal monochromator capable of moving at four degrees per second. Considering a silicon 111 Bragg crystal, the equation giving the energy would be:

$$E(eV)=hc/\lambda =12398.419 \text{Å}eV / (2*3.1354161\text{Å}*\sin\boldsymbol{\theta})$$

The energy is not linear with the incident angle θ, but for small energy intervals at certain energies we can assume linearity. For example, if we scan 1000 eV in the range of 8 keV, the Bragg angle moves by about 2 degrees. If the speed is configured to 4 degrees per second, the scan completes in 0.5 s, taking 2000 points (one point per mili-degree or every 0.5 eV, which is suitable for a ΔE≈0.76 of Silicon 111). Acquiring data from an Alba Em (electrometer)[8] digitalized with an ADC (ADLINK2005, four channels 16 bits simultaneous at 500 kHz), we could theoretically have at least 80% live time that makes (250000/2000)*0.8 = 100 values per scan point. Averaging these values, the statistical noise could be reduced by a factor of 10 (√100).

When the trigger generator computes encoder positions, equidistant pulses make constant angle intervals. If the trigger generator computes time, the angle intervals may have a jitter, when the speed of the Bragg angle is not constant, but the final energies correspond to the right measurements because both encoder counters and detectors ADCs received trigger signals. Scans synchronized at equidistant time intervals are suitable only for linear trajectories at constant speeds. Extending accurate timestamps to every value acquired, overcomes this limitation allowing any trajectory and even makes triggers not mandatory in certain conditions. Triggers will be needed to synchronize actions but not to timestamp the measurements.

The scan participants are configured to start and stop at the needed positions as described previously in this paper. Assuming that the timestamps associated with the scan values are precise enough, the data taken is valid and can be presented in the right format to be analyzed, for example interpolated to equally distant intervals in order to make arithmetic operations.

The precision required in terms of synchronism and jitter of timestamps depends on the speed of the scan. In the example previously described, we would need to distinguish between 2 consecutive energies. Considering the given speed of 4 kS/s, the maximum jitter would be 125 µs peak to peak (0.25ms/2), which means the minimum accuracy required of the timestamp system for this application. Assuming a confidence of about 1E-6 (99.999999% BER or Bit Error Rate) it would result in a RMS (Root Mean Square) value for the jitter equal to 125/9.507µs = 13.14 µs. This figure validates the single digit microsecond precision for "software" timestamps.

The current implementation of the timestamps at Alba does not allow this number, but it is feasible and achievable with the current technology.

ISBN 978-3-95450-139-7

TIMESTAMP IN THE CORE OF HARDWARE AND SOFTWARE

The timing system of the accelerators at Alba is implemented using MRF hardware. The time reference is generated from a GPS based receiver with an embedded high quality Oven Controlled Crystal Oscillator (OCXO) that guarantees 1ns accuracy and distributed using MRF timing system to the rest of the facility. The timestamps are distributed to the event receivers with a very high accuracy (25 ps jitter RMS), which is needed for the synchronization, diagnostics, fast-interlocks, etc. of the accelerators. However, not all devices receive this high precision timestamps. The computers, intervening in the control and data acquisition systems are synchronized by NTP on a millisecond range. As discussed in the previous paragraph, this is not enough for some applications. Besides, the control system shall be prepared for getting the timestamp as in the low-level as possible and keep it through the whole data acquisition chain.

Increasing the time resolution of timestamps is one of the key issues for high performance generic continuous scans. In the millisecond range, a NTP based timestamp offers a flexible and generic solution. When we push the requirements to the microsecond range or faster, specific timestamps distributed and managed by specific hardware become in most cases mandatory. These “hardware” timestamps are then an enhanced trigger where every measurement add in an accurate time stamped by the hardware. The electrometers and other detectors shall have an input for the timestamp distribution, and have the instant timestamp associated with every measurement at a particular moment. All values acquired are then hardware time stamped. The control software buffers, interpolates, merges, and stores the data from slow and fast channels in the desired format.

The performance in terms of accuracy, achieve the level of two digits picoseconds with the current hardware configurations and single digit microsecond jitter for the software distributed timestamps although various values such temperatures often show much slower variations and for which a precision in the range of a second might be sufficient.

CONCLUSION

Continuous Scans are the core of a modern X-ray experimental station extensible to neutron and other sources or laboratories. Time resolved experiments need a much higher precision, translated into synchronization and data rate, which may be required for a large number of setups. But besides kinetic phenomena studied with time resolved data acquisition, virtually all experiments can benefit from continuous scans. They are much faster, increase the throughput of the beamline, reduce thermal drifts, and other risks associated with the variation of the conditions over the time, like radiation damage in some cases, etc.

Having a generic setup for continuous scans is rather more complex than for step scans, involving the configuration and handling of hardware triggers. A good timestamp system, combining hardware and software time distribution, facilitate means to build generic continuous scans on any combination of axes and channels.

The control system manages the triggers, configuring hardware or software triggers according to the particular requirements and hardware involved. The other key component that manages the complexity is the buffering system. The buffers gather the data of the scan, taking data from fast and slow detector, during the acquisition or at the end, with a timestamp or without a timestamp, and finally presenting the data in the right format to be archived, raw, interpolated, etc.

Finally the Sardana SCADA manages the configuration, the access to the hardware, the sequencer, the programming environment and the human machine interfaces for the generic use in an experimental station.

CONTRIBUTIONS

Many people have worked in this project. This is a work achieved with the effort of the whole computing division with a particular mention to the controls and electronics groups. The authors would also thank the cooperation on Sardana, for their feedback ideas and contributions, in particular T. Nuñez and T. Kracht from DESY, F. Picca and N. Leclerq from Soleil, A. Homs, V. Rey, E. Taurel and G. Berruyer from the ESRF, D. Spruce and A. Milan from MaxIV and P. Goryl and L. Żytniak from Solaris.

REFERENCES

[1] D. Fernández-Carreiras et al. “The design of the Alba Control System. A Cost-Effective Distributed Hardware and Software Architecture”. ICALEPCS 2011. Grenoble. FRBHMUST01.

[2] T. Coutinho et al. “SARDANA: The software for building SCADAS in Scientific Environments”. Proceedings of ICALEPCS 2011, Grenoble, France. WEPMSO23.

[3] http://www.sardana-scada.org. Sardana Official Documentation.

[4] http://www.tango-controls.org Official Tango web page.

[5] Zbigniew Reszela et al. “Implementation of continuous scans used in beamline experiments at Alba synchrotron”. ICALEPCS 2013, San Francisco, USA. TUPPC060.

[6] Oscar Matilla et al. “Alba timing system. A known architecture with a fast interlock system upgrade”. ICALEPCS 2011, Grenoble, France. WEPMS023.

[7] http://www.mrf.fi. Micro-Research Finland Oy.

[8] Xavier Serra et al. “Em# Project. Improvement of Low Current Measurements at Alba Synchrotron”. ICALEPCS 2013, San Francisco, USA. TUPPC094.

ISBN 978-3-95450-139-7