EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN - PS DIVISION

CERN/PS 99-067 (BD)

SOFTWARE FOR BEAM DIAGNOSTICS FRONT-END SYSTEMS: SYNCHRONIZATION AND IMPLEMENTATION ISSUES

V. Chohan, M. Ludwig

Abstract

Front-end software systems used for beam diagnostics at CERN's PS accelerator complex perform control and data acquisition of local hardware components in synchronization with specific accelerator events. The principal part of the software is generally hosted in a VME create, which drives all system components, provides interactivity with the general controls environment through networking and decouples the networking layer from the machine layer. Using three real-world examples of operational instrumentation systems, namely the beam intensity measurement between the PS-Booster and the PS, the AD Coherent Oscillations measurement and the PS Closed-Orbit Synchronization, the paper describes their synchronization to accelerator events and states. Sometimes these instrumentation systems are subject to complex real-time constraints and external conditions. The strategies to meet these requirements in the real-time software are discussed in the context of the general design and implementation in the PS control system environment.

11th IEEE NPSS Real Time Conference, June 14-18, 1999 Santa Fe, New Mexico, USA

Geneva, Switzerland
15 December 1999

Software for Beam Diagnostics Front-End Systems: Synchronization and Implementation Issues

M. Ludwig and V. Chohan
CERN, 1211 Geneva 23, Switzerland

Abstract

Front-end software systems used for beam diagnostics at CERN's PS[†] accelerator complex perform control and data acquisition of local hardware components in synchronization with specific accelerator events. The principal part of the software is generally hosted in a VME create, which drives all system components, provides interactivity with the general controls environment through networking and de-couples the networking layer from the machine layer. Using three real-world examples of operational instrumentation systems, namely the beam intensity measurement between the PS-Booster[†] and the PS, the AD[†] Coherent Oscillations measurement and the PS Closed-Orbit Synchronization, the paper describes their synchronization to accelerator events and states. Sometimes these instrumentation systems are subject to complex real-time constraints and external conditions. The strategies to meet these requirements in the real-time software are discussed in the context of the general design and implementation in the PS control system environment.

I. INTRODUCTION

The PS control system has two layers, which are linked by a network: a user-interface layer and a synchronized front-end layer [1], [2]. The user interface layer consists of workstations, which generally run graphical applications to operate the PS accelerator complex. The front-end layer consists of many VME crates, which host system-specific hardware and is connected via Ethernet to dedicated front-end servers for every accelerator. The VME crates run highly specialized control and acquisition tasks, which need to be synchronized to specific events during the accelerator cycle and have to be always operational and running. An overall distributed software telegram called Program Line Sequencer[†](PLS) permits the PS, which was originally designed for 28 GeV protons, to act like 24 different virtual accelerators for protons, electrons, positrons, and heavy ions at various injection and extraction particle energies, together with the PS Booster (PSB). These 'virtual' accelerators are 'copied' into the real accelerator every multiple of a basic period, i.e. 1.2s or 2.4s, before the beam is produced during a user-cycle. This sequencing of the PS and PSB accelerators of up to 24 different functionalities is known at CERN as the Pulse-to-Pulse Modulation (PPM) of the various sub-systems, which constitute these accelerators.

The Real-Time Tasks (RTT) in the VME crates control the hardware associated with the corresponding accelerator, in relation to a PS user-cycle. The interface between the synchronized front-end layer and the non-synchronized user-interface is generally implemented by accessing structured shared memory segments inside the front-end computers from both sides, generally referred to as 'data-tables'. The structure of the data-tables for all VME crates is managed by a central database [3], which generates object-orientated Equipment-Modules (EM) [4], [5]. A given EM, when installed in a front-end computer, implements a specific data-table structure and an accessing mechanism from the application-layer, using Remote Procedure Calls (RPC). The three beam diagnostics systems discussed in this paper are embedded in the general PS complex control environment as VME-based front-end systems. They are but a subset of the many beam diagnostics systems in the PS complex and have all been subject to the design and implementation criteria described here.

II. DESIGN CONDITIONS AND SPECIFICATION

A. General

The rejuvenation and evolution, with reduced manpower, of the PS Controls System from the Norsk-Data mini-computers to a 'standard model' architecture [2] also implied a conceptual redesign of how the specialized accelerator systems were used and interfaced to the control system. The new controls architecture permitted much flexibility and decoupling of specialized systems from the overall controls hardware and software architecture, leading to considerable improvement in versatility of use of the these systems, e.g., systems used for beam measurement and diagnostics. However, this versatility also implied a shifting of responsibilities in design and development of sophisticated front-end hardware and software systems to the specialized systems experts, as opposed to the overall control infrastructure providers. While the ultimate hardware interfacing to the controls infrastructure is done using a certain standard set of VME modules where possible, the overall specialized system design also required substantial software effort in the front-end VME crates for these systems. This so-called specific software has a clear separation from the standard controls infrastructure software, allowing beam diagnostics experts to solve the problem at hand in the most suitable manner and facilitate maintenance. The specific software design and methodology used for implementation depends very much on the software expertise available. This varies from the software template to facilitate development by non-software experts to the methodologies developed for experienced real-time software developers; the latter profits from using the full capabilities of the controls infrastructure and support of its providers.

B. Beam Transfer Transformer system

[†] See Appendix in Section VII

The PSB/PS Beam Transfer Transformer system (BTT) measures the intensity of an extracted beam with a variable time-structure. It consists of six beam transformers along the beam transfer lines between the PSB and PS and inside an experimental area, their associated amplifiers with selectable sensitivity ranges and an array of 28 gated integrators [6]. The PS Booster's four beams are ejected sequentially either to the PS or directly to an adjacent physics experimental area, ISOLDE[†]. Two fundamentally different extraction modes, *'normal'* and *'staggered'*, are possible. The BTT-system acquires the beam current in the transfer lines between the PSB, PS and the ISOLDE area by reading out six beam current transformers. A VME-based multi-channel ADC samples 28 gated integrators and the data is synchronously processed and made available following a range of external conditions from the control environment. The extraction always starts with PSB ring 3.

Table 1
Real time constraints for the BTT-system

N	Synchronization requirement	Precision	Real Time constraint
1	First bunch	120ns	Hard
2	User-cycle	Better than one basic period	Hard

In normal mode, all four beams are ejected sequentially without delay; leading to a bunch train with bunches equally spaced. In staggered mode, a delay ΔT of up to 100μs is introduced between the ejection from the rings, leading to a more complex beam time-structure: i.e., the bunches from ring 3 are followed ΔT later by the bunches from ring 4, then after ΔT by ring 2 and 1. Additionally, the processed data has to be ready at the end of every user-cycle. Table 1 and 2 summarize the Real-Time (RT) constraints and external conditions, where a hard RT constraint means that if the constraint is not met the result is useless.

Table 2
External conditions for the BTT-system, all with precision requirement better than one basic period.

N	External Condition	Range	Real Time constraint
1	User-cycle	[1, 24]	Hard
2	Beam ejection mode	Normal or staggered	Hard

C. AD Coherent Oscillations

The measurement of the Coherent Oscillations of a bunch of particles in the AD after injection is achieved by fast-sampling the signal of a horizontal or a vertical electrostatic pickup over many turns, as already reported by [7]. It consists of two transient receivers, two wide-band multiplexers and two adjustable attenuators hosted in a CAMAC crate, which is controlled and read out over a serial link by a VME crate. The data can then be used to calculate the optimum injection parameters and therefore to minimize the injection coherent oscillations.

This system can be compared to a 100 MHz two-channel storage oscilloscope with additional synchronization, storage and trigger functionality, which is controlled and read out through the PS-control system. The acquisition of the coherent oscillations after injection into the AD is based on high-speed transient sampling of a horizontal and a vertical pick-up. Initially, the sampling is running continuously, and is stopped when the injected beam has circulated for typically fifty turns. Since the AD does not run in PPM, in contrast to the PS and the PSB, there is no synchronization to a user-cycle, and the data is simply overwritten with every new measurement. Nevertheless the slow serial link connection between VME and CAMAC introduces a severe performance bottleneck which limits the repetition rate of the measurement. Table 3 summarizes these constraints, where a soft RT constraint means that if the constraint is not met the result is nevertheless valid.

Table 3
Real time constraints for the AD coherent oscillations measurement

N	Synchronization requirement	Precision	Real Time constraint
1	Injection + 50 turns	Approx. 1 turn	Hard
2	Serial Link	Transmission time depends on data amount	Soft

D. Synchronization for the CODD

The Synchronization for the Closed Orbit Data Display, (CODD-Synchro), is part of a larger system, CODD [8]. The CODD-system measures the particle trajectories over two consecutive turns for a single particle bunch, allowing ten trajectories during one user-cycle. It consists of 40 pick-ups distributed around the PS ring, which are sampled by gated integrators. In order to measure the position of a passing bunch these gates have to be synchronized to the passing bunches during all stages of beam acceleration.

Table 4
Real time constraints for the synchronization subsystem of CODD

N	Synchronization requirement	Precision	Real Time constraint
1	Bunch counting	Better than one bunch length	Hard
2	Harmonic change notification	Better than duration of the change	Hard
3	User-cycle	Better than one basic period	Hard
4	User-cycle time slot	1 ms	Hard
5	Bunch resynchronization	Better than one bunch length	Hard

The CODD-Synchro sub-system [9], [10], which supplies these gates to the gated integrators, consists mainly of VME based gate-generator modules, which are clocked with the bunch-frequency f_b. Additionally several highly specialized modules are used to generate the frequency f_b which has a stable phase-relationship to the bunches.

In order to produce a gate which is synchronized to a given bunch, turn and time during the PS accelerating cycle (user-cycle) of a beam, the system has to have a counting mechanism which permits synchronization to the bunches (Table 4, *N*=1).

Table 5
External parameters for the Synchronization subsystem of CODD, all with precision requirement better than one basic period.

N	**External Condition**	**Range**	**Real Time constraint**
1	User-cycle	[1, 24]	Hard
2	Harmonic number sequence	1 out of 16 possible sequences	Hard
3	Particle type	1 out of 6 possible types	Hard

Furthermore, since the harmonic number of the RF System† typically changes several times during a user-cycle, and the sequence of harmonic numbers (Table 5, *N*=2) and their switching times is different for different user-cycles, CODD-Synchro needs to be notified immediately when the harmonic number changes (Table 4, *N*=2). The system has to be set up correctly to allow for up to 10 measurements during a user-cycle, with a precision of 1 ms (Table 4, *N*=4). Since the PS runs in PPM, a set of external parameters and conditions, which describe a user-cycle, has to be acquired periodically in order to set up the measurement correctly for the specific beam (Table 4, *N*=3 and Table 5, *N*=1, 3).

III. IMPLEMENTATION

A. General

The PS Controls System renewal to the new architecture was spread over 7 years (1991-98). This was necessary because the PS Complex runs continually to satisfy all the physics demands at CERN, except for ~2 month annual shutdowns. Thus, the implementation of all the specific systems had to follow this scenario, i.e., depending on the accelerator, the shutdown year chosen for the conversion to the new infrastructure and given manpower and budget profiles. Furthermore, for each specific system, careful analyses had to be carried out to find a solution which satisfied the requirements with due considerations of (a) existing specific beam diagnostics hardware, (b) existing specific interfacing hardware, (c) use of new VME based hardware using standard commercial modules where possible, (d) software implications, e. g., system drivers etc., for novel VME modules and finally (e) specific software effort required. Often, additional constraints of how the specialized system was seen and used by the accelerator operator came into play; this then led to the need (or not) of new application layer software and its implications for the front-end specific software design and implementation.

B. Beam Transfer Transformer system

The BTT system performs data acquisition and treatment tasks exclusively, where the information flows from the hardware to the front-end software (and from there ultimately to the application program) only. The principal and most severe RT constraint from Table 1 requires a precision of better than 120 ns, which would be very difficult to meet in a distributed system, based on software only. As a consequence, this condition is met by the hardware part of the system: a separate beam synchronization unit produces the gates required for the sample-and-hold gated integrators and a separate pulse train generator triggers the following conversions in the ADC. Both devices are triggered by the first ejection from the PSB, which is usually the third ring[1].

The hardware does not distinguish dynamically between the bunch-trains which are separated by ΔT (‘staggered ejection’) and the normal ejection where the bunches are not separated: the generation of the gates for a given PSB ring is always triggered by the ejection of this ring. The gate lengths are not in PPM and do not make the distinction between the normal and the staggered ejection mode: in normal mode all bunches of all four rings fall within the first gate only. This leads to an external condition, where the beam ejection mode has to be acquired in addition to the basic synchronization for PPM (Table 2, *N*=2), to allow for different software conditions for both of the ejection modes. The precision required for the beam ejection mode and the user-cycle can be fully met by the standard control environment timing and sequencer distribution; the software needs to acquire only the conditions from Table 2 from the sequencer and to synchronize to the user-cycles within one basic period.

C. AD Coherent Oscillations

The acquisition of the AD coherent oscillations at injection makes it necessary to modify the system settings like the sampling frequency, sampling time, offsets, attenuators and the multiplexing channel according to the beam conditions. The system must perform control actions on the hardware as well as the data acquisition, which makes it possible to use the system in a flexible manner, for potentially other beam diagnostic tasks in other accelerators. The VME crate, via a serial link, controls the CAMAC crate; this hosts the transient receivers, the programmable attenuators and the multiplexers. On the VME side there is only a standard VME serial controller running at 5 MHz, which is accessed by the RTT via a standard driver. Both control actions, which are passing only a few values over the serial link and the data acquisition, which transmits up to 32K samples, have to use the serial link and cannot be interleaved. Nevertheless the VME system can process the data while the serial link is busy. In order to get the maximum performance from the system, the RTT was designed with two working POSIX-threads [11] in the program. During one measurement cycle, the higher priority thread handles the data acquisition over the link. During the dead-times in the link communication and when nothing else is going on, the lower priority thread treats and writes the data to the data-table and performs control actions over the serial link.

---

[1] a typical PSB extraction sequence is from rings 3, 4, 2, 1

The transient receivers are in the sampling mode at all times: for a measurement, an external trigger, which is independently provided by a standard programmable timing, stops the sampling. This programmable timing has a resolution of 100 ns; in the case of the coherent oscillations at injection, it is synchronized to the general AD injection forewarning timing with a fixed delay to compensate for the cable lengths. This timing also interrupts the RTT high-priority thread to start the data transmission from the CAMAC to the VME, immediately after the sampling is stopped.

The AD cycle lasts much longer than one basic period and it is desirable to use this system for other beam diagnostics as well, once the injection measurement is accomplished. In order to have the possibility to perform several control actions during one AD cycle, the low priority thread for communication with the data-table is interrupted, and allows synchronization to another programmable timing. The minimum delay necessary for the hardware to respond correctly to a control action must be respected before the sampling is stopped; other than that, both timings may be freely adapted for different measurements. In any case, all data from a previous measurement is always overwritten by the current data. Since the AD runs in a non PPM-mode, the user must retrieve all data before any new measurement.

D. Synchronization for the CODD

The implementation of the CODD-Synchronization focuses around three challenges: the generation of a frequency f_b with a stable phase relationship to the beam during the whole accelerating cycle, the changes of the harmonic number h of the RF system during a user-cycle and lastly, the fast control of the gate-generators.

The f_b, which is the frequency of a bunch passing one of the 40 pick-ups in the PS, is generated using a programmable digital lookup-table hosted in an external module, where the table pointer is driven by the gradient of the main magnetic field of the PS, which is represented by a pulse for every step of 0.1 G ('B-train'). One table holds the theoretical f_{bt} as a digital word as a function of B, for a particle type and a harmonic number. This digital frequency word, which is extracted from the lookup-table, synthesizes the theoretical analogue bunch frequency f_{bt} in a DDS[†] module. After beam injection, it is then phase-locked with a signal from the pick-up-units, to generate the real bunch frequency f_b. During an accelerating cycle, the harmonic number and therefore, the frequency-table, can change successively several times; the RTT has to prepare the correct tables in advance and switch over to the next table in the Harmonic Number Sequence (HNS) for the current cycle. Generally, this HNS must be acquired from the PLS before a cycle starts (Table 5, N=2). Different cycles have different HNS, in order to allow sufficiently fast stepping through the HNS at every harmonic change during the cycle (Table 4, N=2).

The harmonic changes are notified to CODD-Synchro by an external impulse train generated by the RF system (Table 4, N=2), which interrupts the RTT at every change. Then, the RTT steps to the next harmonic in the HNS and activates the correct frequency-table for the new harmonic.

For every measurement during one cycle, the gate generators must first be loaded with the corresponding set of controlling parameters. At every harmonic change a different set of parameters must be loaded (Table 4, N=2, 4). To meet both these hard RT constraints, the RTT derives in advance all possible settings of the gate generators for all the combinations of h, user-cycle slot, bunch and turn. These are buffered as virtual settings inside a map-like structure. During the cycle and as a function of the time slots of the measurements to be carried out as well as the harmonic number h, the RTT has to decide which virtual setting must be loaded into the gate generators.

IV. CONCLUDING REMARKS

In general, the design and implementation of specific software in front-end VME crates in the current controls system has long-term usage and maintenance implications. In the context of CERN's PS complex where the specific systems' requirements evolve according to physics needs, experience has shown that specific front-end software activities have to follow a common approach as far as possible, with peer appraisals at the design stage, use of standard software methodologies, book-keeping, repositories and so forth. This then permits sufficient knowledge distribution among the systems integrators and the limited community of specific software developers to assure smooth running, future evolution and maintenance.

Design and constraints of specific front-end software have always to be discussed in the light of a possible implementation; for the designer and the system integrator, this requires a high degree of familiarity with the general system environment, the specific hardware and the beam measurement to be performed. Based on the examples of the three operational beam diagnostics front-end systems, this paper demonstrates how the relatively innocent looking RT constraints and external conditions can lead to rather complex implementations, as soon as the functional requirements exceed a simple controls- or acquisition-only specification.

To meet hard RT constraints, the implementation choice usually suggested is a hardware-only solution, but which in turn may lack the required functional flexibility. In the case of the BTT-system, adding another external condition to allow for the differences in data treatment for normal and staggered extraction could compensate this. Sometimes software and hardware have to be combined in complex ways to meet both functional and RT requirements, as is shown by the generation of f_b and the switching of the frequency-tables for CODD-Synchro. More advanced strategies such as the use of a map of virtual modules to organize the data, which can be based on object-orientated technologies as well, may be used to meet certain constraints. If the system contains one principal performance bottleneck, such as the serial link in the Coherent Oscillation system for the AD, the overall implementation may be almost fully predetermined by this, but with the advantage of a possibly simpler design.

V. ACKNOWLEDGMENT

The work of system integration is always based and dependent on building blocks that are contributed by many people. The authors would therefore like to express their gratitude in a general way to all their colleagues who could not be referenced or named individually but have contributed nevertheless in many ways to these systems.

VI. REFERENCES

[1] R. Rausch, C. Serre, "Common Control System for the CERN accelerators”, *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems,* KEK, Tsukuba, Japan, Nov. 11-15, 1991, pp.54-58.

[2] F. Perriollat, C. Serre, “The New CERN PS control system overview and status”, *Proceedings of the Third International Conference on Accelerator and Large Experimental Physics Control Systems*, Berlin, Germany, Oct. 18-23, 1993, pp.86-90.

[3] J. H. Cuperus, M. Lelaizant, “Integration of a Relational Database in the CERN PS Control System”, *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Bejing, China, Nov. 3-7, 1997, pp.433-436.

[4] L. Casalegno, J. Cuperus, C. H. Sicard, “Process Equipment Data Organisation in CERN PS Controls”, *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems*, Vancouver, Canada, Oct. 30 – Nov. 3, 1989, pp.412-415.

[5] A. Gagnaire, N. de Metz Noblat, C. Serre, C. H. Sicard, “Replacing PS Controls Front End Mini Computers by VME based 32 bit Processors”, *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems,* KEK, Tsukuba, Japan, Nov. 11-15, 1991, pp.375-377.

[6] C. Carter, V. Chohan, G. Gelato, F. Lenardon, M. Ludwig, J. D. Schnell, “Real Time Control, Acquisition and Data Treatment for Beam Current Transformers in a Transfer Line”, *Proceedings of the* X^{th} *IEEE Real Time Conference,* Beaune, France 1997, pp.349-352.

[7] V. Chohan, C. Johnson, J. P. Potier, "Fast CAMAC-Based Sampling Digitizers and Digital Filters For Beam Diagnostics and Control in the CERN PS Complex”, *Proceedings of the Europhysics Confr. on Computing in Accelerator Design and Operation*, Berlin, Sept. 20-23, 1983, Springer Verlag, pp.405-410 .

[8] Ian Milstead, "Closed Orbit Digital Display (CODD) VME based equipment and related hardware development", CERN PS-BD Note 96-07, July 1996.

[9] Jose-Luis Gonzalez: "Specification for VME CODD Synchronisation", CERN PS/BD Note 98-05 (tech), July 1998.

[10] M. Ludwig, "The Synchronisation for the PS Closed Orbit Measurement System: Real Time Software Development and Implementation", CERN PS/BD Note 99-03 (tech), May 1999.

[11] B. O. Gallmeister, POSIX.4: Programming for the Real World, O’Reilly & Associates Inc., Sebastopol, CA, 1995.

VII. APPENDIX: ACRONYMS

PSB: CERN’s Proton Synchrotron Booster, consisting of four similar stacked rings, from which the beam is ejected sequentially to the PS at 1.4GeV. The PSB receives the beam at 50MeV from the proton Linac.

PS: CERN’s Proton Synchrotron, which accelerates the beam injected from the PSB at 1.4GeV to nominally 26GeV.

AD: CERN’s Antiproton Decelerator.

RF system: *R*adio *F*requency system, used to accelerate a bunched beam. The harmonic number of the RF system is the number of “RF-buckets” in the accelerator; a “RF-bucket” can either contain a bunch or be empty.

PLS: *P*rogram *L*ine *S*equencer, a distributed telegram-like timing structure, which dynamically describes a user-cycle related operating parameters for the accelerators in the PS complex.

ISOLDE: *I*sotope *S*eparator *O*n-*L*ine *De*vice; this experimental physics facility uses the PSB beam directly.

DDS: Direct Digital Synthesizer, generates analogue frequencies from digital words.