TIMING SYSTEM SOLUTION FOR MEDAUSTRON; REAL-TIME EVENT AND DATA DISTRIBUTION NETWORK

R. Štefanič, R. Tavčar, J. Dedič, Cosylab, Ljubljana, Slovenia
J. Gutleber, CERN, Geneva, Switzerland
R. Moser, EBG MedAustron, Wr. Neustadt, Austri

Abstract

MedAustron is an ion beam research and therapy centre under construction in Wiener Neustadt, Austria. The facility features a synchrotron particle accelerator for light ions. The timing system for this class of accelerators has been developed in close collaboration between MedAustron and Cosylab.

Mitigating economical and technological risks, we have chosen a proven, widely used Micro Research Finland (MRF) timing equipment and redesigned its FPGA firmware, extending its high-logic services above transport layer, as required by machine specifics. We obtained a generic real-time broadcast network for coordinating actions of a compact, pulse-to-pulse modulation based particle accelerator. High-level services include support for virtual accelerators and a rich selection of event response mechanisms. The system uses a combination of a real-time link for downstream events and a non-real-time link for upstream messaging and non time-critical communication. It comes with National Instruments LabVIEW-based software support, ready to be integrated into PXIe based front-end controllers.

This article explains the high level logic services provided by the real-time link, describes the non-real-time interfaces and presents the software configuration mechanisms.

ABOUT MEDAUSTRON

MedAustron [1, 2] is intended for research and clinical therapy applications. Its synchrotron-based accelerator will have 3-5 ion sources (protons, carbon ions and possibly other light ions) and 5 beamlines, one of which is a rotating gantry. It will provide ion beams with energies up to 800MeV to several irradiation stations used for different purposes. Development follows defined processes and quality management standards. Among the development goals is to use commercially available components where possible to minimize the technological and economical risks. The machine will serve as a blueprint for further applications. Accelerator layout is shown in Figure 1.

TECHNOLOGY CHOICE FOR MEDAUSTRON TIMING SYSTEM HW

No out-of-the box timing system solution exists yet for MedAustron's type of accelerators. The key requirements for the timing system are a deterministic network protocol for real-time operation, reliable event/data distribution and fast response times from the timing system master to all (about 300) controlled devices. It must provide 1μs real-time control loop resolution, 100 ns timestamp resolution and support for 250.000 different accelerator cycles.

When approaching timing system design, we examined the possibilities of reusing existing technology and Micro Research Finland (MRF) [3] proved the most suitable. It is designed for more demanding light-source accelerators and satisfies MedAustron synchronization requirements well. What it lacks is the required higher-level logic and a generic accelerator timing system application. We therefore decided to keep MRF's widely used and proven timing transport layer and build the high level logic application on top in its FPGA firmware. As MedAustron Control System (MACS) software will run on National Instruments' PXIe controllers, we will develop MRF cards in PXIe form factor.

An important part of the timing system is the software support which is fully integrated into MedAustron Control System (MACS). The software provides flexible mechanisms for configuration, control and supervision of the timing system. It is implemented within MedAustron's Front End Control System (FECOS) software framework, developed in LabVIEW, also in close collaboration between MedAustron and Cosylab. The framework is compatible with LabVIEW Real-Time operating system, allowing implementation of both non-real-time and real-time device support applications (i.e. FECOS components).

Reusing the MRF timing transport layer for timing system design enabled us to deal with machine specifics and focus more on integration aspects of the design, which is often an underestimated and overlooked part of timing system development.

TIMING SYSTEM OVERVIEW

MedAustron Timing System, called the Real-time Event Distribution Network (REDNET) comprises a real-time and a non-real-time link. The real-time link is based on MRF transport layer and custom logic in FPGA, whereas the non-real-time link is based on Gigabit Ethernet.

The real-time network topology is the same as original MRF: one Event Generator (EVG) and multiple Event Receivers (EVR), connected in tree topology with multiple fanout layers. Each card is located in a PXI crate

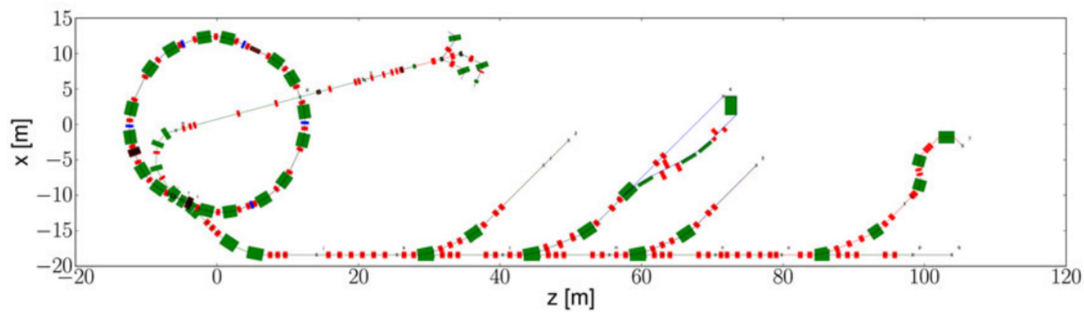

Figure 1: Accelerator layout with injection chain, extraction line and transport lines.

with the PXI controller running support software. The combination of an EVG (or EVR) card with its support software is called the Main Timing Generator (MTG); or Main Timing Receiver (MTR), respectively. The timing system will control about 300 devices, but EVR's sophisticated I/O mechanisms lower the number of required EVRs. Support software provides access to configuration and control to the supervisory control system and triggers transmission of traffic over both the real-time and the non-real-time interface.

The configuration of MTG and MTRs is done by the supervisory control system via a non real-time supervisory interface.

HIGH LEVEL SERVICES OF TIMING SYSTEM REAL-TIME INTERFACE

The following subsections describe the timing system's main services. Richness of logic in communication and event response mechanisms also illustrates specific requirements of MedAustron accelerator class.

Event Response Mechanisms

The timing receiver offers multiple possibilities of controlling its neighbouring cards via numerous hardware connections and via software IRQs. Through LabVIEW API, the user can easily configure the Event Receiver to:

- trigger external devices (via TTL, fibre optic, etc.)
- trigger PXI/PXIe neighbouring cards (via star trigger bus)
- provide neighbouring cards with real-time data (via trigger bus)
- provide SW applications with IRQs and real-time data
- provide delayed, inverted or otherwise "post-shaped" response to timing events (e.g. for fine tuning of synchrotron injection synchronization down to 5 ns resolution)

All types of subscriptions to different events can be used by the same application.

Virtual Accelerator (VAcc) Support

Timing events are scheduled in 5 separate, concurrently usable execution slots (ES), providing concurrently running virtual accelerator functionality. The concept enables users to program individual accelerators as if they were separate, each of them having its own execution configuration. However, timing events of all VAccs are piped through the same serial link (fibre optic), where prioritization takes place in case events are scheduled for emission at the very same moment. Receivers can be linked to any combination of these VAccs and event responses for different ES can be configured separately.

This feature will enable commissioning of different parts of accelerator (e.g. ion sources) at the same time independently, reducing commissioning time immensely. During machine use, it will provide a safe, controllable way of experimenting with new accelerator settings.

Commands

Commands are an extended concept of timing events, also providing data payload, which makes them very useful for distributing near-real-time information. They are implemented using the concept of pipes, additionally abstracting the communication channel to make it even easier to use. Commands have the lowest priority.

Heartbeat Event and Time Service

Along with emission of timing events defined in a sequence, the MTG also provides a unique heartbeat event emitted independently of any other traffic. Its emission range is configurable, ranging from 0.5Hz to 50Hz. Receiver response to the heartbeat event is fully configurable. This functionality is used to operate the injector in temperature stabilized mode.

The time service mechanism provides a transparent way of distributing current time to all receivers.

Real-Time Traffic Prioritization Scheme

Figure 2 explains the prioritization of different real-time traffic and offers an insight into the mechanisms implemented inside EVG FPGA firmware. All emission mechanisms are conveniently configurable through LabVIEW API. The figure shows that among the 5 ES, one has top priority over all other traffic. Next is the heartbeat event, then the other 4 ES, followed by asynchronous user events and commands.

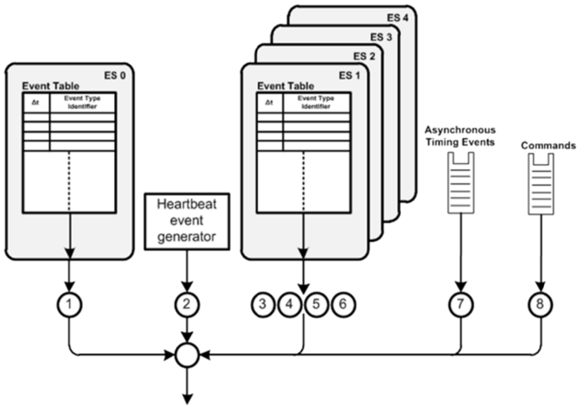

Figure 2: Message prioritization scheme.

Synchronization Parameters

Table 1 shows the timing characteristics of the event emission mechanism.

Timing events can be scheduled with the granularity of 1µs, meaning that every µs, events from all 5 execution slots and the heartbeat event can be scheduled for emission. In such case, fixed prioritization occurs, as shown in Figure 2. At reception, events are time-stamped with 100 ns resolution. These timing characteristics satisfy MedAustron timing system requirements, which strongly guided the system design. The underlying MRF transport layer greatly exceeds this synchronization performance.

Table 1: Timing System Synchronization Parameters

Event Emission Granularity	1µs
Timestamp Resolution	100 ns

NON-REAL-TIME INTERFACE AND SOFTWARE CONFIGURATION

There are two main messaging services implemented for the non-real-time interface. Both are based on National Instruments Simple TCP/IP Messaging (STM), extending its functionality to serve our specific needs.

Simple Messaging Mechanism (SIM)

This is a simple enhancement of STM. It provides the interface to the PVSS-based supervisory control system, which uses it for execution control and configuration of accelerator cycles by issuing requests to the MTG, moving it through the cycle generation process.

Publish- Subscribe Service

Built upon the SIM interface, the Publish-Subscribe Service provides enriched messaging mechanisms for communication between all software support components across the non-real-time network. MTG publishes commands via this interface to all "slow" devices without requirements for deterministic operation. One of the mechanisms based on this service is acknowledgement of commands; for monitoring the system integrity during operation, all commands received by relevant devices via real-time and/or non-real-time interface must be acknowledged via the Publish-Subscribe interface. The MTG denies service if any acknowledgements fail to arrive. The Publish-Subscribe service can freely be used for non-real-time communication between device-specific software components within FECOS.

Configuration of Software Components

Configuration of all FECOS components is provided in XML files, providing startup and operation parameters, process variable locations and any other user-defined information. In fact, all structurally rich information is distributed to and among components in standard XML format. This information includes accelerator sequence definitions used by the MTG and timing event response configurations used by the EVR.

Process variables for online configuration and monitoring of end devices are implemented via LabVIEW's shared variable engine and connected via OLE for Process Control (OPC) to the PVSS-based SCADA server of the supervisory interface.

CONCLUSION

By using the off-the-shelf product from MRF and building the machine specific real-time application on top of its already widely used and stable transmission layer, the overall design time can be greatly reduced. The design effort can thus be strongly focused on specifics of the machine itself and not on the timing distribution layer, which has already been successfully addressed multiple times before. The total time from requirements to the working system was less than 2 years. This design approach is the way to achieve precise, but versatile solutions able to cope with complex use cases of different machines.

The control system software framework provides developers with flexible communication and configuration mechanisms. It also provides a generic component structure which readily integrates into the control system framework and allows the device support developer to focus mainly on the implementation of device-specific application, speeding up the development process.

REFERENCES

[1] M. Benedikt, A. Wrulich, MedAustron—Project overview and status, Eur. Phys. J. Plus (2011) 126: 69

[2] M. Benedikt, A. Fabich, "MedAustron — Austrian hadron therapy centre", Nuclear Science Symposium Conference Record 2008. NSS '08. IEEE, pp.5597-5599, 19-25 Oct. 2008

[3] J. Pietarinen, "MRF Timing System", Timing Workshop CERN, February 2008