

# Implementation of local area network extension for instrumentation standard trigger capabilities in advanced data acquisition platforms<sup>a)</sup>

J. M. López,<sup>1</sup> M. Ruiz,<sup>1</sup> E. Barrera,<sup>1</sup> G. de Arcas,<sup>1</sup> and J. Vega<sup>2</sup>

<sup>1</sup>*Grupo de Investigación en Instrumentación y Acústica Aplicada, Universidad Politécnica de Madrid, Ctra. Valencia Km-7, Madrid 28031, Spain*

<sup>2</sup>*Asociación EURATOM/CIEMAT para Fusión, Madrid 28040, Spain*

(Presented 14 May 2008; received 11 May 2008; accepted 14 July 2008; published online 31 October 2008)

Synchronization mechanisms are an essential part of the real-time distributed data acquisition systems (DASs) used in fusion experiments. Traditionally, they have been based on the use of digital signals. The approach known as local area network extension for instrumentation (LXI) provides a set of very powerful synchronization and trigger mechanisms. The Intelligent Test Measurement System (ITMS) is a new platform designed to implement distributed data acquisition and fast data processing for fusion experiments. It is based on COMPATPCI technology and its extension to instrumentation (PXI). Hardware and software elements have been developed to include LXI trigger and synchronization mechanisms in this platform in order to obtain a class A LXI instrument. This paper describes the implementation of such a system, involving the following components: commercial hardware running a Linux operating system; a real-time extension to an operating system and network (RTAI and RTNET), which implements a software precision time protocol (PTP) using IEEE1588; an *ad hoc* PXI module to support hardware implementation of PTP-IEEE 1588; and the multipoint, low-voltage differential signaling hardware LXI trigger bus. © 2008 American Institute of Physics.

[DOI: [10.1063/1.2968694](https://doi.org/10.1063/1.2968694)]

## I. INTRODUCTION

Most advanced data acquisition systems (DASs) currently used in scientific experiments, such as fusion devices, are distributed systems made of a set of instruments housed in a chassis. This requires powerful triggering mechanisms that serve to synchronize the start of data acquisition (DAQ) by the different units in the system. Such systems allow simultaneous detection of events using multiple instruments over the course of the experiment. This is of increasing relevance as the shot duration in fusion devices increases, and it is essential in long-pulse devices. In addition, when the distributed DAQ units also possess processing capability, specific experimental events can be detected by one chassis and transmitted to the others in the system.

The hardware architectures used most extensively for DAQ in nuclear fusion experiments are based on modular instrumentation that is housed in a chassis, as in the case of COMPACTPCI and PXI. All of these technologies include powerful local trigger and synchronization mechanisms to control the operation of the different instruments in the chassis. These architectures generally do include a system controller CPU that has a network adapter in order to receive the setup parameters prior to each acquisition and in order to send the raw or processed data to the central storage system once DAQ is complete. However, the architectures usually lack

specific mechanisms to synchronize DAQ between different chassis. Thus, the present paper presents a powerful trigger and synchronization mechanism for event detection (ED) across distributed DAS. It uses the trigger mechanisms defined in the local area network (LAN) extension for instrumentation<sup>1</sup> (LXI) standard, and it is based on a versatile DAQ and processing platform named ITMS, which has been jointly developed by Universidad Politécnica de Madrid and CIEMAT.<sup>2</sup> This platform is available in both 3U PXI and 6U COMPACTPCI format.

## II. ITMS GENERAL SYSTEM ARCHITECTURE

Figure 1 shows the standard hardware and software architectures of the ITMS platform. The hardware consists of the following elements:

- One standard 3U PXI or 6U COMPACTPCI chassis, with a standard embedded system CPU (SCPU). SCPU is used for the following purposes: to configure the DAQ cards, to acquire data from the DAQ channels, to distribute the acquired data among the system CPUs, and to process data from desired channels.
- Several DAQ cards.
- Several peripheral CPU (PCPU) cards allocated in peripheral slots (Inova ICP-PM-4 or Concurrent Technologies PP300/022). PCPUs are used to process the data acquired from desired channels. PCPUs increase the computing capacity of the PXI platform beyond that of typical PXI

<sup>a)</sup>Contributed paper, published as part of the Proceedings of the 17th Topical Conference on High-Temperature Plasma Diagnostics, Albuquerque, New Mexico, May 2008.

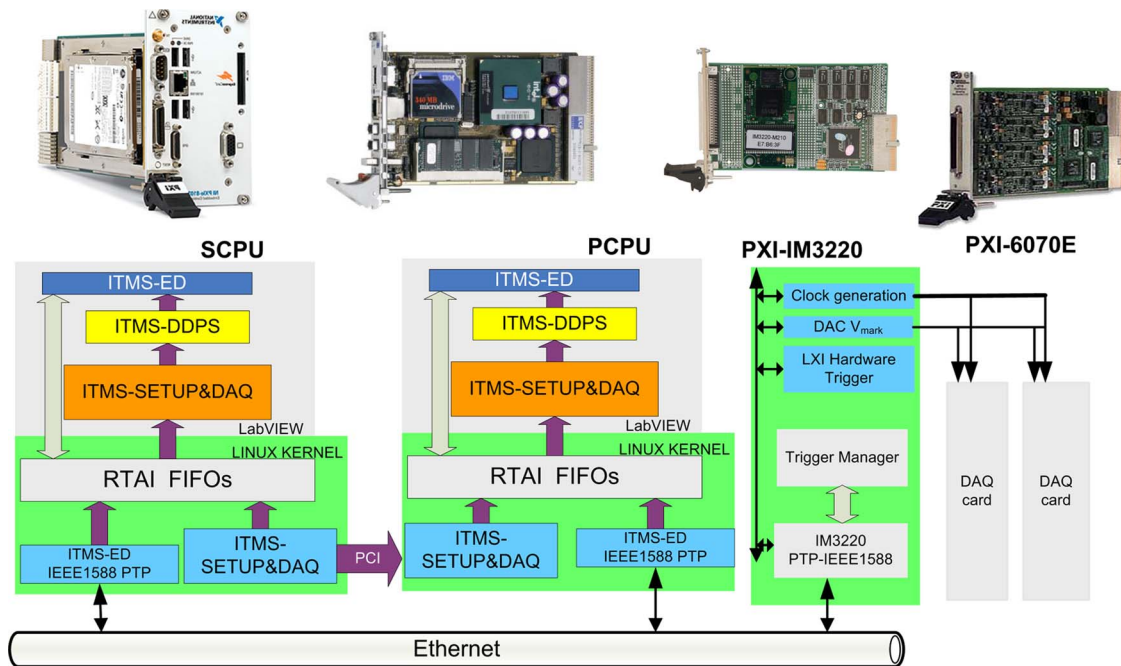


FIG. 1. (Color online) Architecture of the basic ITMS system.

systems, which are based only on the system's CPU controller.<sup>3</sup>

ITMS software architecture is based on three software modules that run both in the SCPU and in the PCPUs (Fig. 1): Setup&DAQ, dynamic data processing system (DDPS), and ED. These modules were developed using Linux kernel modules (Fedora Core 5) based on RTAI (version 3.5), the COMEDI DAQ driver (version 0.7.76), and LABVIEW software (version 8.2.1).

The Setup&DAQ module controls the DAQ and data distribution configuration linking different processing elements of the system (PCPUs and SCPU). The user interface was developed in LABVIEW, while the DAQ and distribution aspects were coded using RTAI-Linux kernel modules. The software controlling continuous data collection in the final user applications was developed in C language and integrated in LABVIEW code using code interface nodes.

The DDPS module runs the user-defined data processing routines, which the user must develop in LABVIEW. The ED module is a client-server application that runs in each CPU (SCPU or PCPUs). The client, which usually runs in the PCPUs, detects events generated by the DDPS module and sends them to the RTAI module. This module then generates a message that will be sent to another PCPU/SCPU or DAS in the system.

### III. IMPLEMENTATION OF LXI TRIGGER MECHANISMS ON THE ITMS PLATFORM

Trigger mechanisms defined in the LXI standard were chosen in order to provide the ITMS platform with advanced and efficient trigger mechanisms, which in turn, would allow development of distributed acquisition and processing systems. The standard specifies the following trigger protocols:

*Trigger over the LAN.* The instrument receives a mes-

sage in a user datagram protocol (UDP) or internet protocol (IP) packet containing trigger information. This mechanism features a longer latency time than traditional hardware trigger lines. Nevertheless, it is very useful for distributed applications in which it may be difficult to connect a physical trigger signal among all subsystems due to costs or physical limitations. All LXI class B and C instruments must comply with these types of trigger mechanisms.

*Precision time protocol IEEE1588 trigger.* This type of trigger uses a universal time coordinated clock (UTC), as required for LXI class A instruments. The instrument is triggered when a predefined absolute time is reached. As long as the clock is very precise, this type of trigger is useful because it does not rely on a hardware line to connect all the systems. The standard supports both hardware and software implementations of this mechanism. The ITMS platform features both options: the SCPU and PCPUS implement the precision time protocol (PTP)-IEEE1588 using software based on RTAI/RTNET, and it also includes a hardware PXI hardware module designed with commercial modules using the IM3220 microcontroller. This microcontroller implements the PTP-IEEE1588.

#### A. Implementation of the trigger mechanism on SCPU and PCPUS

The SCPU and PCPUS in the ITMS platform are standard 3U or 6U boards running Linux with an RTAI real-time module and the real-time network module RTNET.<sup>4,5</sup> A software application, ITMS-ED, was developed as an RTAI/RTNET kernel module that implements the PTP protocol and manages the board's hardware clock.<sup>6</sup> In addition, the application includes a LABVIEW module that can receive events or messages arising during data processing and that can send them to the RTAI module, which then generates LXI compliant, module-to-module messages that include a timestamp. LXI Event iden-

tifiers can be defined in this software module. The kernel module keeps the SCPU and PCPU clocks synchronized with the master clock. Neither the SCPU nor the PCPU supports the generation of hardware signals for trigger purposes.

## B. Implementation of hardware trigger and sampling rate clock signal for DAQ cards.

A PXI hardware module was developed to include the generation of hardware trigger signals based on the LXI standard and sampling rate clocks for DAQs modules in the ITMS platform. The PXI module has been implemented using two commercial Imsys Technologies modules (M20 and S20) and a custom-made one. The M20 module includes an IM3220 microcontroller and dynamic random access memory resources. The S20 module includes additional hardware to support the Ethernet physical interface, RS232 and USB interface, and digital input-output. The custom-made module includes hardware to implement the digital trigger and clock signals for DAQ ITMS modules, and the digital hardware to support the multipoint, low-voltage differential signaling hardware triggering.

The hardware functionalities provided by this PXI module are the following:

- Support of the precision time protocol (IEEE1588). The IM3220 chip is based on reduced instruction set computer (RISC) architecture and a field programmable field array (FPGA) devoted to support network time stamping for IEEE1588. This microcontroller includes a PTP protocol engine implemented in assembler language and microcode, and a software API to manage the engine.
- Implementation of the hardware trigger signaling mechanisms required for class A LXI instruments and generation of two sampling clock signals that can be used as the sampling clock for ITMS DAQ cards. The module also generates the analog voltage proportional (or related) to the sampling rate that is applied to channel 0 in DAQ cards.

A software application named trigger manager (TM) was developed under the RUBUS operating system included with IM3230 development tools. TM waits for UDP connections and manages the PTP engine. Using a UDP socket, the TM waits for LXI module-to-module message bearing information about when a DAQ trigger signal is generated (timestamp). The TM processes the message and programs the PTP engine to generate the DAQ trigger signal. The uncertainty in the value of the timestamp in the trigger signal is reported with our results.

The TM also supports special LXI module-to-module messages that change the sampling rate in the DAQ cards. When the TM receives such a message in a UDP socket, it proceeds to generate a new sampling clock and an analog voltage for the channel 0 of DAQ cards. In addition, the TM generates a new module-to-module message that includes the timestamp of sampling rate change. This idea of an adaptive sampling rate DAQ system has already been described.<sup>3</sup> In this situation, the most important thing is to know when the sample frequency has been changed—i.e.,

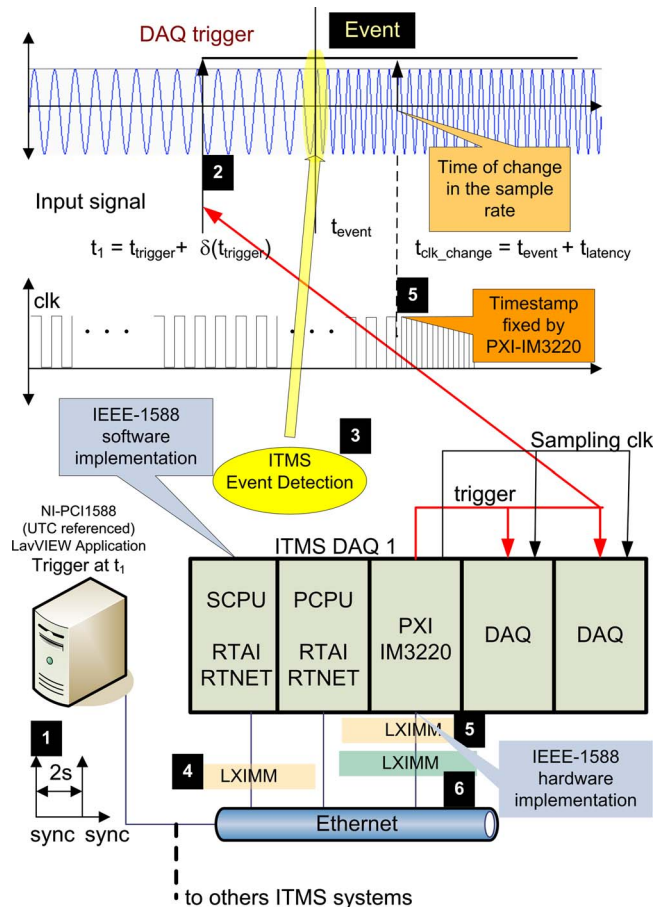


FIG. 2. (Color online) Distributed DAQ system using an ITMS platform and LXI trigger mechanisms.

the exact timestamp—and not the moment when the rest of the modules become aware of this event.

Figure 2 shows how the abovementioned ITMS functionalities can be used to create versatile and powerful distributed DASs. The trigger of the DASs,  $t_{\text{trigger}}$ , is controlled using the PXI module based in IM3220. Its clock is synchronized with a master clock implemented with a PCI IEEE1588 card from National Instruments (Fig. 2, steps 1 and 2). The system achieves a very low  $t_{\text{trigger}}$  uncertainty,  $\delta(t_{\text{trigger}})$ , due to hardware implementation of the PTP protocol as described in the next paragraph. Once DAQ has begun, the acquired data are processed in both the SCPU and PCPU LABVIEW applications. These applications can generate module-to-module LXI messages (LXIMM) depending on the results of the data processing (step 3). For example, SCPU can detect the need to change the sampling rate during DAQ, and it can send a message (step 4) to the TM software module running in IM3220 of the PXI module. As a result, the sampling clock and the analog voltage connected to each board's analog input channel 0 are updated. It is important to point out that the system has a certain latency interval since high-level applications detect each event and signal this event to the rest of the system (step 5). This latency has two terms, the first is due to the time necessary to detect the event in the block of acquired data,<sup>3</sup> and the second is due to propagation of the LXI message throughout the network. The former depends on several experimental parameters, includ-

ing buffer block length, sampling rate, and processing algorithm,<sup>3</sup> whereas the latter depends on network delay.

#### IV. EXPERIMENTAL RESULTS

The setup used to achieve the delays in the previously described experiment comprises the following:

- a PC running a LABVIEW application using the NI PCI-1588 card,
- a PXI chassis with a SCPU, one PCPU, the PXI module designed based on the IM3220 (PXI-IM3220), and two 6070E DAQ cards,
- a DGS-1008D switch from D-Link, and
- a Tektronix digital oscilloscope (TDO) TDS 754C

The first measurement is the deviation (in time) of the trigger signal generated by PXI-IM3220 with respect to the master clock (NI PCI-1588). After 1 h, during which the clocks are synchronized every 2 s, the LABVIEW application causes NI PCI-1588 and PXI-IM3220 to generate a pulse (using their *I/O* available pins) every 5 s. The TDO acquires both pulses and the delay between edges is measured based on a reference edge generated by the NI PCI-1588. The mean delay value obtained after an experimental run of 24 h was 97 ns with a standard deviation of 165 ns. In the worst-case scenario, based on these values and a coverage factor of 2 (probability of 95%), the system is predicted to have an error of one sample at 1 MS/s and four samples at 10 MS/s.

The second measurement is the latency time to change the sampling rate. A Linux kernel module running in the PC was developed to send an LXI message to the PXI-IM3220 using the UDP socket. At the same time, a pulse is generated in a pin of the parallel port of the PC. When the TM, which is running in PXI-IM3220, receives the LXI message, it programs the new sampling rate and generates another pulse. In our experiment, the delay measured between the rising edges of the pulses had a maximum value of 10 ms. This delay has two terms, the first of which is the time required by the Linux module to send the UDP packet to the network. This value is on the order of 500  $\mu$ s.<sup>5</sup> The second term is due to the latency of the RUBUS operating system running in

IM3220. The latency has a quasiuniform distribution and was found to vary between 2 and 9 ms in our experiments. Finally, the deviation of SCPU and PCPU clocks are in the same order of magnitude of the values presented in the work of Correl *et al.*<sup>6</sup>

#### V. DISCUSSION

Trigger mechanisms defined by the LXI standard can be very useful for creating distributed DAS for long-pulse fusion devices. The PXI-IM3220 hardware module is an effective solution to control the start of DAQ in these systems, with a very low uncertainty on the order of 200 ns. In addition, the combination of RTAI and RTNET applications allows the user to achieve low-latency access to Ethernet, which is very useful to signal events and allow synchronization in distributed systems. In this implementation it is necessary to improve the network response time for LXI messages of PXI-IM3220 module. So, we are working in the design of a new embedded hardware based in the use of DP83640 chip (hardware support for ethernet and PTP timestamping). Finally, this solution has been tested using two systems like shown in Fig. 1 and one master clock (PC with NI-IEEEE1588) in the same network segment. The results obtained are the same presented in the previous section.

#### ACKNOWLEDGMENTS

The authors wish to express their gratitude to the Spanish Ministry of Education and Science for support (Project No. DPI 2006-06624).

<sup>1</sup>LXI Standard Rev. 1.2.01. Nov 2007, LXI consortium, Inc.

<sup>2</sup>M. Ruiz, E. Barrera, S. López, and D. Machón, *Rev. Sci. Instrum.* **75**, 4261 (2004).

<sup>3</sup>M. Ruiz, J. M. López, G. de Arcas, E. Barrera, R. Melendez, and J. Vega, *Fusion Eng. Des.* **83**, 358 (2008).

<sup>4</sup>RTNET (<http://www.rts.uni-hannover.de/rtnet>).

<sup>5</sup>A. Luchetta, A. Barbalace, G. Manduchi, A. Soppelsa, and C. Talierno, *Fusion Eng. Des.* **83**, 520 (2008).

<sup>6</sup>K. Correll, N. Barendt, and M. Branicky, (<http://ptpd.sourceforge.net>).