

LOW COST FFT SCOPE USING LABVIEW, CRIO AND FPGA

O. Ø. Andreassen, A. Rijllart, I. Mätäsaho, L. Arnaudon, CERN, Geneva, Switzerland

Abstract

At CERN, many digitizers and scopes are starting to age and should be replaced. Much of the equipment is custom made or not available on the market anymore. Replacing this equipment with the equivalent of today would either be time consuming or expensive.

In this paper we look at the pros and cons of using Commercial Off The Shelf (COTS) systems like National Instruments (NI) Compact-RIO (cRIO) and NI PCI eXtensions for Instrumentation (NI-PXIe) and their Field Programmable Gate Array (FPGA) capabilities as flexible instruments, replacing costly spectrum analysers and older scopes.

We add some insight on what had to be done to integrate and deploy the equipment in the unique CERN infrastructure and the added value of having a fully customizable platform, that makes it possible to stream, store and align the data without any additional equipment.

INTRODUCTION

Oscilloscopes, Spectrum and network analysers are all invaluable instruments when studying and understanding everything from electrical networks to physical phenomenon.

A spectrum analyser measures the magnitude of an input signal versus frequency within the full frequency range of the instrument. The primary use is to measure the power of the spectrum of known and unknown signals. The input signal a spectrum analyser measures is electrical, however, spectral compositions of other signals, such as acoustic pressure waves and optical light waves, can be considered through the use of appropriate transducers [1]. For more general-purpose measurements you have oscilloscopes, which are commonly used to observe the exact wave shape of an electrical signal. Oscilloscopes are usually tuned so that one can read voltage and time within a pre-defined bandwidth. This allows the measurement of peak-to-peak voltage of a waveform, the frequency of periodic signals, the time between pulses, the time taken for a signal to rise to full amplitude (rise time), and relative timing of several related signals [2].

Stand-alone traditional instruments such as oscilloscopes and spectrum analysers are very powerful, but they all have their limitations: You have to consider bandwidth, sampling rate, sampling mode, dynamic range, triggering and memory when selecting the instrument, making it difficult or too expensive to have a “all in one” solution that could be used for any type of measurement. As a result, the instruments tend to become integrated parts of system installations, bringing other issues such as scalability, availability, data export and security into the mix. A second challenge is lifecycle

management. Once an instrument starts coming of age, or if the system requirements change, it is difficult to replace or upgrade the unit. Often you have to replace the whole instrument, giving you new interfaces and different software to deal with, making it difficult or expensive to upgrade the system.

We wanted to address these challenges by making use of National Instruments COTS [3] systems, which through the cRIO or PXI platform offer the instrumentation capability needed to meet these trials, providing a high level of flexibility and modularity to develop targeted test and measurement systems that can easily be reconfigured for various measurements needs.

In addition, the software had to be easy to use and offer the ability to create operator, program and data interfaces to existing tools to ease the process of integration into CERN’s production environment and add features which were not existing in the traditional portable scopes. NI LabVIEW software with its wide range of instrument drivers, already widely used in our section, met these challenges and was chosen in preference to ANSI C or C++.

The flexibility of the cRIO and PXI system combined with the ease of use of LabVIEW enabled us to make rapid progress in system design, prototyping and deployment of measurement systems.

ARCHITECTURE

Through the hardware platforms of cRIO and PXI, resources are easily duplicated by plugging in additional modules without changing any of the software, allowing the measurement equipment to be scaled as a factor of throughput. Traditional instruments use similar components, but the most obvious difference from a purely hardware standpoint is how the components are packaged.

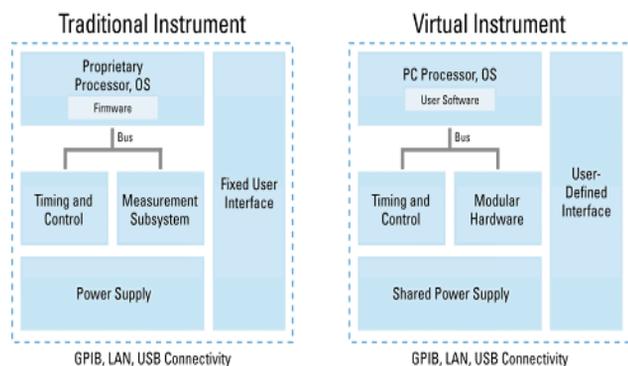


Figure 1: Traditional instrument versus virtual instrument.

A stand-alone instrument puts all of the components in the same box for every instrument. These instruments are designed as discrete portable entities and not designed for system use. While there are a large number of traditional instruments, the software processing and user interface are fixed in the instrument itself and can be updated only when and how the vendor chooses (for example, through a firmware update.) Thus, the user has to follow the standards and interfaces provided by the instrument vendor.

By contrast, a software-defined modular instrument makes the raw data from the hardware available to the user, and the user can design their own interface.

With this approach, users can make custom measurements, perform measurements for emerging standards, or adapt the system if requirements change (for example, to add instruments, channels, or measurements). While user-defined software can be applied to stand-alone, application-specific hardware, it is ideally paired with general-purpose, modular hardware where the full flexibility and performance of the measurement software can be made of good use. This combination gives a flexible, scalable and integration friendly addition to the traditional instruments.

MODULAR HARDWARE

Modular instrumentation can take several forms. In a well-designed system, many of the components are shared across instrument modules instead of duplicating these components for every instrument function. These modules include different types of hardware, including oscilloscopes, function generators, digital, and RF. In some cases the measurement hardware can simply be a peripheral device that is connected to one of the host computer ports or peripheral slots. In this case, the host CPU provides the processor for performing the measurements in software as well as the chassis for the power supply and I/O.

In other cases, on the cRIO platform, the measurement hardware is housed in an industrial chassis (Fig. 2).

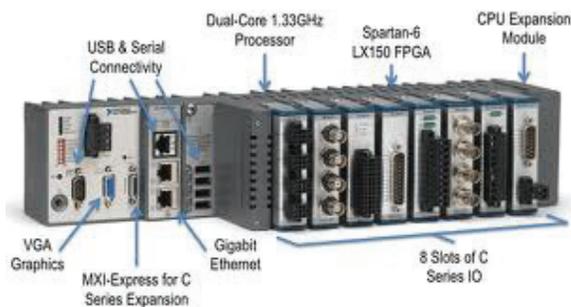


Figure 2. cRIO platform.

In a PXI system (Fig. 3), the host computer can be embedded in the chassis or it can be a separate laptop, desktop, or server that controls the measurement hardware through a network interface. Because a PXI system uses the same buses internal to a PC (PCI and PCI Express), the same modular instrumentation concepts

apply equally using a PXI system or a PC [4]. Regardless of what modular equipment used, this sharing of the chassis and controller greatly reduces cost [4], and also enables the user to control the measurement and analysis software.



Figure 3: PXI system.

As a result, the modular system can easily be scaled in every direction. If you need more channels, you can easily add digitizers, keeping the software and interfaces intact. If you need higher sampling speeds or processing power, you can add scope cards or FPGA cards. It is also possible to reduce the measurement system and channel count, when for instance going from lab prototypes to installed systems, custom tailoring the scope based on need and cost.

SOFTWARE

A traditional instrument is designed to perform one or more specific tasks defined by the vendor. The user generally cannot extend or customize them. The knobs and buttons on the instrument, the built-in circuitry, custom software, and the functions available to the user, are all specific to the nature of the instrument. If you need to export data or integrate your instrument with existing infrastructure, you have to follow the vendor's development plans and adapt your interfaces accordingly.

One of the main advantages when considering virtualized instruments (VI) is the possibility of getting all the functionality of a regular scope or spectrum analyser through the hardware, but at the same time adding any type of interface or analysis needed.

Software Architecture

The LabVIEW real time software architecture is quite modular, where each module has a specific task such as I/O operations, analysis, monitoring, network communication and logging. A typical application architecture (Fig.4) where each task has been divided into specific modules, completely separate from each other, communicates through dedicated message handlers and queues.

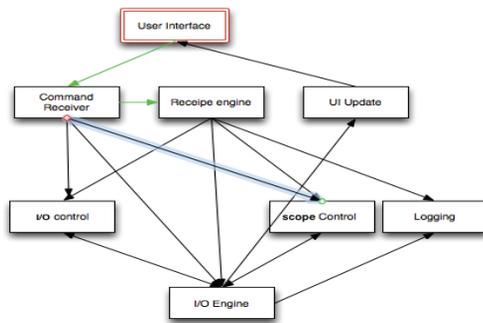


Figure 4. Module messaging and architecture.

Analysis Modules

LabVIEW comes with hundreds of built-in mathematical and analytical modules. For the virtual scope application we focused on functions such as Fast Fourier Transforms (FFT), power spectrum, peak power and frequency, in-band power, adjacent-channel power, and occupied bandwidth. We also added functions for performing modulation-domain operations such as pass band (IF) to baseband (I-Q) conversion, I-Q to IF conversion, and generation/analysis of analogue modulated signals.

As an example on LabVIEW’s built-in analytical functions, take the standard FFT (Eq. 1)

$$X(f) = F\{x(t)\} = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (1)$$

This function can in LabVIEW be easily added as a simple block (Fig. 5).

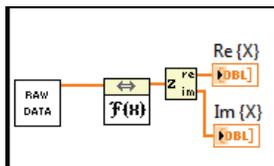


Figure 5: LabVIEW FFT function.

FPGA

For instruments with high throughput or high channel count, we use FPGA technology. With FPGA, you can perform high-speed signal processing, I/O synchronization and custom triggering. For control systems, you can also run advanced control algorithms directly in the FPGA fabric to minimize latency and maximize loop rates.

The NI LabVIEW FPGA Module lets you graphically implement digital circuits within an FPGA chip, re-using the same code you would when implementing the real time application. In Figure 6 you see a the same FFT function as in Figure 5, but now the FFT runs on a 100MHz FPGA clock (each FFT takes about 30.000 cycles to perform for 8192 spectral lines), producing data every ~100 us/channel. This is a very good performance.

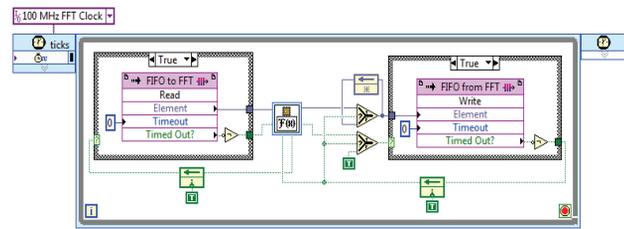


Figure 6. LabVIEW FPGA FFT with DMA FIFO.

As a proof of concept we made a low cost spectrum analyser for the Antiproton Decelerator (AD), doing real time FFTs, visualizing the “cooling” effect [5] (passing antiprotons through clouds of electrons) of the accelerator. For this instrument, we used a standard cRIO-9076 and a NI 9214, 4 channels, 100 kS/s Analog input module. We made an intensity vs time plot (Fig. 7) which runs on the operators’ console using one of the standard LabVIEW plots.

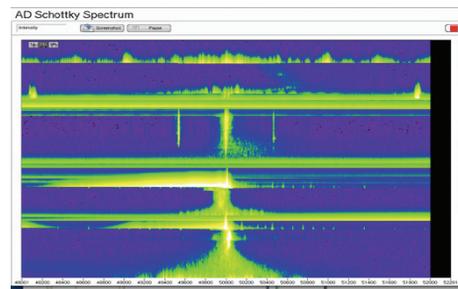


Figure 7: AD cooling frequency intensity plot.

In the case of the AD instrument, the source signal was already available from a down converter, so we didn’t have to buy an expensive RF module to digitize the data. What is noteworthy is the real time processing done on a 100 ms refresh interval, giving you instant feedback, without any specific client libraries needed except for rendering.

NI-SCOPE

In many cases following the Nyquist theorem is not enough, and to accurately digitize the incoming signal, the digitizer’s real-time sample rate has to be at least three to four times the digitizer’s bandwidth [4]. The reason can easily be understood from a plot (Fig. 8).

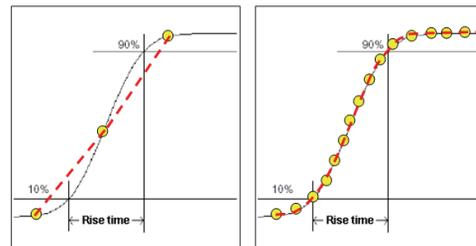


Figure 8. Accurately reconstructing signals with 3-4 times more samples than the Nyquist criteria.

For high speed digitizers (typically from 200 MS/s up to 12.5 GS/s) LabVIEW has a dedicated development API

called NI-SCOPE. We have used this API for performing RF measurements on the CLIC Test Facility and the Proton Synchrotron (PS). The basic communication scheme is a client-server (Fig. 9), which is used to transmit the data from the instrument to the operators.

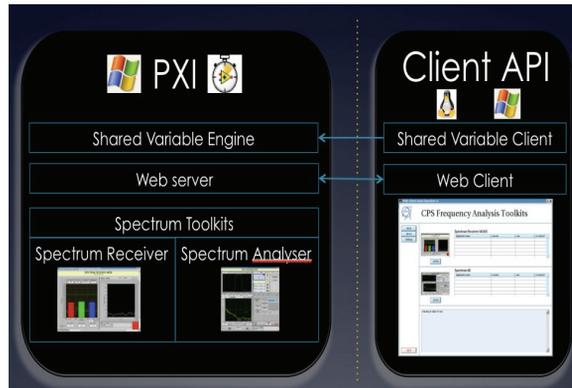


Figure 9: Spectrum analyser concept.

INTEGRATION

An effort has been done to integrate the virtual instruments at CERN. A lot of the work before this initiative was already done through the LabVIEW Rapid Application Development Environment [6][7], facilitating integration with the CERN Front End Software Architecture (FESA) and general infrastructure, but we wanted to integrate the instruments as plugin modules to the existing Open Analogue Signal Information System (OASIS)[8] as well.

To do so we made a binding using the built-in shared variables [4] in LabVIEW and a tool from the BE-CO group at CERN called IE-PLC [9].

The IE-PLC tool generates communication bindings and bridges the CERN control middleware with NI equipment by creating an instrument specific class design, making it possible to control, stream and store data from any instrument, through all the main used APIs at CERN.

CHALLENGES

Some of the challenges we still need to solve are:

- Timing integration
- Mass configuration and deployment
- Remote management and monitoring

The CERN General Machine Timing (GMT) timing is a custom built timing system, used to orchestrate all the different hardware around the accelerator. It will be a crucial part of the instrument integration and is needed if we want to reach nanosecond synchronization on distributed systems. Also deployment, monitoring and management tools are still immature in the NI world, but it is being worked on.

CONCLUSION

By using NI hardware, LabVIEW, FESA and IE-PLC we managed to create a fully scalable, modular and flexible measurement framework, capable of adapting to

almost any measurement need using COTS components. The framework is fully integrated in the CERN accelerator control domain and can be easily extended to include any type of cRIO and PXI equipment, which is not only available from NI, but also from 70 other hardware manufacturers.

FUTURE PLANS

In the future we plan to integrate the GMT timing, further empowering the flexible measurement framework, making the virtual instruments inseparable from any other measurement equipment at CERN. We are also working on different designs on incorporating remote management and maintenance tools in the LabVIEW and NI equipment and we are planning on embedding the EN-ICE's Monitoring Operation of cOntrols Networks (MOON) [10] interface on both the PXI and cRIO platform.

REFERENCES

- [1] M. T. Hunter, A. G. Kourtellis "Fundamentals of Modern Spectral Analysis", USENIX ATC, Boston MA, (2010).
- [2] N. Kularatna, "Fundamentals of Oscilloscopes", Digital and Analogue Instrumentation: Testing and Measurement, Institution of Engineering and Technology, pp. 165–208 (2003).
- [3] D. McKinney, "Impact of Commercial Off-The-Shelf (COTS) Software and Technology on Systems Engineering", Presentation to INCOSE Chapters, (2001).
- [4] <http://www.ni.com/white-paper/4426/en/>
- [5] P. Belochitskii, "Status of the Antiproton Decelerator and of the ELENA project at CERN", CERN, Geneva, Switzerland.
- [6] O. Ø. Andreassen, et al., "Turn-Key applications for accelerators with LabVIEW-RADE", ICALEPCS (2011), Grenoble, France.
- [7] O. Ø. Andreassen "The LabVIEW RADE Framework" ICALEPCS (2011).
- [8] S. Deghaye et al., "OASIS: a new system to acquire and display the analog signals for LHC", ICALEPCS'03, Gyeongju, Korea, October 2003.
- [9] F. Locci, "IEPLC Framework: Automated Communication in a Heterogeneous Control System Environment" ICALEPCS (2013).
- [10] F. Valera et al. "Monitoring Control Applications at CERN", ICALEPCS (2013).