Measurement 55 (2014) 265–271

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Low-cost open-source multifunction data acquisition system for accurate measurements

F.J. Ferrero Martín [a,*], M. Valledor Llopis [a], J.C. Campo Rodríguez [a], J.R. Blanco González [b], J. Menéndez Blanco [b]

[a] *Department of Electrical and Electronic Engineering, University of Oviedo, Campus de Gijón, 33204, Gijón, Spain*
[b] *Ingen10 Ingeniería, Technology Park of Asturias, 33428 Llanera, Asturias, Spain*

A R T I C L E I N F O

Article history:
Received 28 November 2013
Received in revised form 28 April 2014
Accepted 7 May 2014
Available online 24 May 2014

Keywords:
Data acquisition
Signal conditioning
Python
LabVIEW

A B S T R A C T

Engineers and scientists commonly use data acquisition hardware and software to research unknown characteristics through measurement and analysis. Computer-based data acquisition systems are a powerful, flexible, and cost-effective measurement solution. This paper presents the hardware and firmware design of a low-cost computer-based data acquisition system that can be very interesting for conducting laboratory experiments and industrial applications. Its open-source philosophy opens the door for free software users and developers that would be interested in contributing with the project.

1. Introduction

In any activity related to science and technology there is usually the need to measure many kinds of physical and electrical parameters. In this context, the term "data acquisition" is often applied to a variety of measures, ranging from analog to digital signals of several types, from the acquisition from a simple motion sensor, to the acquisition from a complex pulse train. The fundaments of data acquisition systems (DAQ) can be found in excellent books [1–3]. Papers about DAQs are usually devoted to design DAQs to measure and monitoring specific parameters [4–7]. This work presents the design of a multifunction DAQ that provides analog input and output, digital input and output, and counter/timer circuitry.

In a broad sense DAQs respond to three general configurations:

(1) Computer-based systems, in which a DAQ is connected to the computer (Fig. 1a) [8–12]. The DAQ is basically responsible of performing the signal conversion between the analog and digital domains. Then, the computer performs the processing, storage and visualization. The DAQ is connected to the PC via a bus (USB, PCI, PCI Express, Ethernet) or wirelessly, being each of these alternatives most appropriate depending on the type of application.
(2) Systems based on autonomous instruments, independent from the computer's bus (Fig 1b), but with the ability to connect to this, through instrumentation buses like GPIB, or through serial or parallel communication ports (RS-232, USB, etc.). They are typically used in demanding and higher cost applications that require capturing large number of variables.
(3) Modular acquisition systems (Fig. 1c) connected to a specialized local bus such as VXI or PXI [8]. This allows configuring more powerful DAQ, at a cost that could be smaller than with stand-alone instruments, especially at systems of a certain size or with the need of future growth.

* Corresponding author. Tel.: +34 985182552; fax: +34 985182138.
E-mail address: ferrero@uniovi.es (F.J. Ferrero Martín).

http://dx.doi.org/10.1016/j.measurement.2014.05.010
0263-2241/

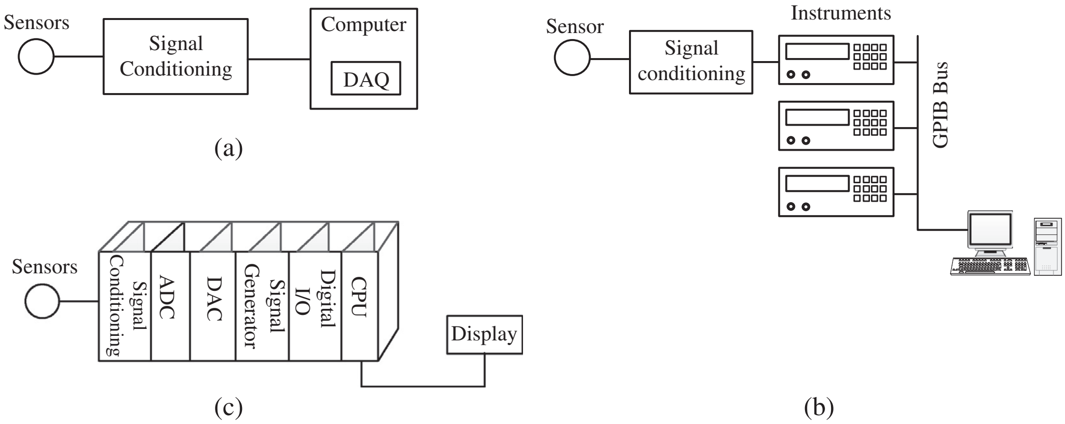

Fig. 1. DAQ configurations: (a) DAQ based on computer; (b) DAQ based on independent acquisition instruments; (c) modular DAQ.

This work focuses on computer-based DAQ, where the instrument is connected to the computer through the USB bus. It presents a low-cost DAQ that is open to the users (openDAQ) [13]. It is a powerful tool that allows configuring, in a short time, many kinds of experiments, related to instrumentation and measurement systems. The open source communications protocol and the availability of low level libraries for handling work environments in Python and LabVIEW increase its scope.

The paper is organized as follows: Section 2 presents the general characteristic of the DAQ developed and it is compared in relationship to other popular device. Section 3 describes the hardware implementation. Section 4 presents the firmware implementation. Section 5 is devoted to explain how to use the DAQ. Section 6 shows an example of application. Finally, Section 7 is for the conclusions.

2. General characteristics

The openDAQ features a mini-USB connector for power and communications, two screw terminal connector plugs for I/O, and an LED indicator, as shown in Fig. 2. In relationship to others DAQs, openDAQ presents excellent features. Table 1 shows a comparison with the popular National Instruments NI USB-6009. In general openDAQ presents better characteristics. Only the sample rate is lower. However for the large majority of signal to be measured the frequency is less than 1 kHz so a sample frequency of 20 kHz may be enough. The firmware has a powerful feature that allows the possibility to generate predefined waveforms through DAC output.

Fig. 2. OpenDAQ top view.

Table 1
Comparison between NI-USB6009 and openDAQ.

	NI USB-6009	OpenDAQ
Analog inputs		
• Number of channels	8	8
• Resolution	14 bits	16 bits
• Maximum voltage range	±10 V	±12 V
• Maximum sample rate	48 KS/s	20 KS/s
Analog outputs		
• Number of channels	2	1
• Resolution	12 bits	14 bits
• Maximum voltage range	0–5 V	±4 V
Signal generator	No	Yes
Digital I/O	12	6
Counters/timers	1	1
Timing source	Software	Hardware
Compatibility	LabVIEW	Python, LabVIEW
Dimensions (mm)	85.1 × 81.8 × 23.1	50 × 50 × 14
Prize (approx.)	$370	$270

3. Hardware implementation

Fig. 3 shows the block diagram of the openDAQ. It can be divided into two functional blocks:

3.1. Analog block

The analog signal conditioning circuit is implemented around the instrumentation amplifier (IA) INA826, from Texas Instruments [14]. The coupling of the input signals and the reference voltages to this AI is done through two analog multiplexers (CD4051). Another multiplexer selects the resistance which determines the gain of the IA. Finally, other multiplexer allows reducing the input voltage, applying a resistive divider which divides the voltage to automatically access to ±12 V range. The output of the AI is conditioned in order to convert a ±4.096 V span to 0/+4.096 V. This last voltage reaches the ADC, which is TI's ADS8321. It is a 16 bit ADC, SAR type, capable of sampling up to a 100 KS/s rate.