ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

Comparing USB Data Acquisition Instruments Using Arduino and PIC18F4550 in LabVIEW and Matlab

Elías V. Beauchamp[1], Dr. Elio Lozano[2]

Instructor, Department of Electronics, University of Puerto Rico at Bayamón, San Juan, Puerto Rico [1]

Associate Professor, Department of Computer Science, University of Puerto Rico at Bayamón, San Juan, Puerto Rico[2]

Abstract:This paper designs three acquisition and playback systems for analog and digital data. Using three micro controllers architectures: Arduino (ARM), chipKIT (PIC32) and PIC18F4550 for the transmission of analog data to a PC. We compare differences in architecture and their communication schemes. Program interfaces for data acquisition were implemented in LabVIEW and Matlab. We proceeded to measure the throughput of samples that the computer can receive per unit time. This measure is estimated in several ways.First and second order statistics are used to comparetheir performance.

Keywords: Digital Signal Processing, LabVIEW, Matlab, Virtual Instrument, Arduino, PIC18F4550, USB data acquisition

I. INTRODUCTION

The development of the instrumentation has impacted different areas of science and engineering. Visual instrumentation technology is rising up at a faster rate. National Instrument'sLabVIEW is a platform and development environment that has extensive support for accessing instrumentation hardware. Software controlled electronicsequipment is being developed at a growing rate in today's connected environment. One such system is data acquisition in which physical variables areread into microcontroller based systems andretransmitted into a PC using peripheral communication ports. Data acquisition (DAQ) devices can be developed using microcontrollers with integrated analogue-to-digital converters. One such system is the PIC18F4550 microcontroller which has been used widely as a data acquisition device. In [11] a data acquisition and control system for high-speed gamma-ray tomography based on the USB and Ethernet communication protocols has been designed. This system is based on Microchip's PIC18F4550 and PIC18F4620 whilst the DAQ software is realized using LabVIEW. In [28] the author presents a microcontroller-based DAQ for differential thermal analysis. The DAQ is based on the PIC18F4550 microcontroller and two K-type thermocouples. The user interface was realized using LabVIEW. In [8] different implementations of practical applications using LabVIEW are discussed. In [19], the authors present a module data acquisition device configured as USB RAW for LabVIEW. This module is connected to SCADA systems and other devices that work with Modbus Protocol. In this research, similar experiments were performed using the USB RAW protocol. In [24] a data acquisition system is presented, implemented with a PIC18F4550 microcontroller, which sends data via USB to the PC using software developed in LabVIEW. In [25] an optical spectrometer was designed, using a PIC18F4550 microcontroller in order to send spectral data via the USB port, using a program interface developed in LabVIEW. This research also used the same microcontroller to send data to the PC. The other system is Arduino which is an open source single-board microcontroller that is gaining popularity and acceptance as a quick prototyping tool for DAQ applications. The Arduino Uno development board has been successfully used in many applications such as low-cost platforms [10, 26, 2, 27, 17] for implementing different systems such as biomechanics, LED stimulators, colorimeter, photovoltaic cells, electrochemical etching and others USB DAQ devices available on the market. The application control was implemented in LabVIEW, whereit can controls the system, visualizesthe data obtained from the acquisition system and saves data into a file.

In this paper we describe the implementation of three low-cost acquisition systems intended for control applications with USB [4, 1, 12] interface basing on the PIC18F4550, Arduino Uno, and ChipKit microcontrollers. The content of this paper is organized as follows. In Section I an existing literature review regarding DAQ systems was performed. Identification and selection of the hardware and software required to develop proposed systems was made in section II. Also, in this section a series of preliminary experiments for the characterization of the designed system is presented. In section III we cover the implementation of three different systems, theexperiments used to characterize them and their comparison. Section IV analyzes and discusses the results of the experiments. Finally, in section V the conclusions about the developed systems are presented; additionally the future work to be performed is shown here.

II. HARDWARE AND SOFTWARE DESCRIPTION

In this paper three microcontrollers and a personal computer were used. Both manufacturers, Atmel and Microchip Technology[20] recognized by fans worldwide

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

for their platforms (PIC and AVR) in the area of microcontrollers were chosen. In particular the PIC platform has multiple processors of interest, PIC18 and PIC32 [14, 15]. The AVR platform has become popular among the do it yourself community recently by the implementation of the open architecture Arduino on Atmel AVR [3]. The microcontrollers were programmed using the USB library of the CCS compiler [5] and the open source Arduino software [3]. The interface programs on personal computer were programmed using National Instruments' LabVIEWandits VISA-RAW protocol [22] and Matlab's Instrument Control Toolbox [16]. We used a development board and arbitrary waveform generator (AWG) to feed a signal to the analog input of the microcontroller. After being quantized, the signal was transmitted via the USB port to a PC. Once the data finished collectingit was analyzed using Matlab and each individual system characterized for later comparison.

A. Firmware Implementation

In an integrated system, a real-time operating system (RTOS) [13] whose works react to external stimuli with the desired response is used. For this work analog data was transmitted via the USB link to test the proposed platform. Due to variations between Arduino and PIC architectures, the proposed systems were implemented in two different ways.

B. Communication scheme

In order to standardize communication between different microcontrollers, a data packet to encapsulate the information in the microcontroller was designed. In order to maximize the performance of the communication channel data transmission is limited to a package of five (5) bytes described in Table 1. This standardized package minimizes the reading error after transmission over the USB.

TABLE I
DATA PACKET FOR TRANSMISSION OF VOLTAGE AND TIME

Voltage Delimiter	Voltage Byte	Time delimiter	Time Byte	End of Record
0x76	0x00 – 0xFF	0x74	0x00 – 0xFF	0x0B

This packet is transmitted as frequently as the microcontroller can send it, as long as a new sample for each package is obtained. To analyze the transmitted data, the time required from capturing voltage of the previous packet to the next is carefully measured.

1) Arduino Implementation.

We used two variations of the Arduino platform, the Arduino UNO [3] and Digilent ChipKIT [7]. These do not have direct access to the USB port. According to the architecturedocumentation, the microcontroller serial ports are used for communication. The data is transferred through the USB channel [3], sometimes via a separate circuit. This restriction creates a limit to the maximum bandwidth that can be used for data transmission. Given the previouslymentioned limitations, the libraries for serial transmission of the Arduino platform are used. A speed of 500kbps for serial communication is selected, after validating the maximum stable velocity of transmission that can be read by the PC in both platforms. The flowchart in Fig. 1 shows the algorithm used for the transmission of analog data.

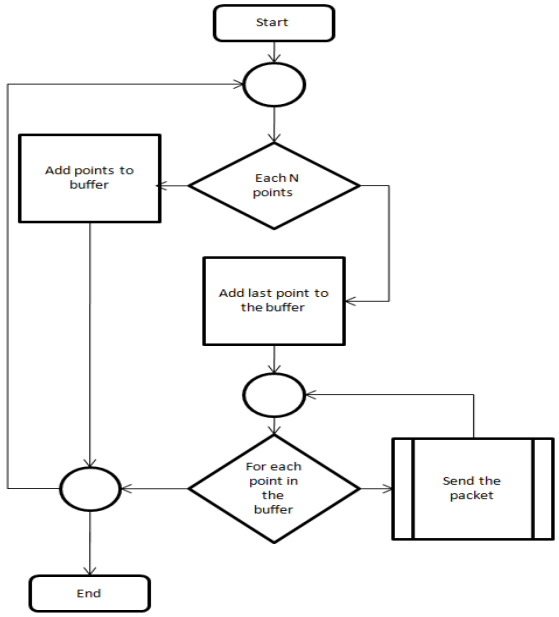

Fig.1 Flowchart of communication for the Arduino platform.

2) PIC Implementation.

The platform used for the PIC architecture was the PIC18F4550 Development Board from Futurtec [9]. The CCS compiler from Custom Computer Services [5] was used for their implementation of the USB drivers. The code was optimized in the following ways:

1. Interrupts are used to minimize the effect of any process in the background. Two interrupts in the code were implemented: TIMER2 and AD. TIMER2 is a high priority interrupt used to ensure that the acquisition time is as precise as possible. The high priority interrupt for TIMER2 is used as a clock with a resolution of 550ns. This is calculated using the frequency of oscillation of the system and the registers used for counting. The following equation calculates the interrupt time for TIMER2

$$t_{TIMER2} = \frac{f_{osc}}{Divider} * RegIntVal$$

The AD interrupt is used to facilitate thetransferofdata, from the analog acquisition circuits, whenever the ADC finishes acquiring the sample. During the interrupt, the transmission packet with the time information of the TIMER2 interrupt is prepared.

2. By utilizing the integrated circuits within the microcontroller we canmaximizethe execution of code while the acquisition processes of analog samples are completed. The AD interrupt brings the added benefit of allowing the microcontroller to execute other instructions while performing the data acquisition.

3. Data is transmitted and received via the USB port in order to maximize the performance. By using the USB port we gainthe benefit of interruptions implemented in CCS libraries and only consume processing cycles when placing data ontothe USB output buffer.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

C. Interface implementation to receive data measurements via USB port

1) Preliminary Implementations.

Initially, for the first experiment the PicUSBVISA software (USB Communication with LabWindows / CVI and NI-VISA, 2010) [6] was used. This program had a PC interface programmed in LabWindows / CVI (Fig. 2) and the firmware was scheduled for a PIC18F2550 microcontroller [14]. For this reason the firmware was modified to be used with microcontroller PCI18F4550. The result was that data was sent to turn on and off light bulbs in the development board running this microcontroller.

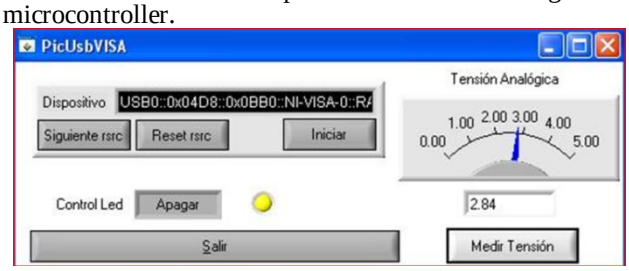

Fig. 2 PicUsbVISA Interface implemented in LABWINDOWS/CVI (source [6]).

For the second test experiment, using a prototypingboard was used. This card includes the controller prototype micro LEDsand potentiometers to indicate the operation of the electrical signals. This device is connected via USB to the PC. The microcontroller that was used for this project is the PIC 18F4550. This microcontroller was programmed using CCS compiler with USB libraries. In [24] an implementation of communication found a microcontroller and PC using LABVIEW language and CCS compiler. The schematic block LABVIEW and firmware source code written in C for PIC 18F4550 microcontroller is in the aforementioned article. For this reason the micro controller is programmed using the CCS compiler. C program which is implemented in [24] and modified for our particular case was used. Initially the program had problems in message passing, because the original program was programmed for communication using interrupts.

National Instruments'driver was installed to recognize the microcontroller as a VISA-RAW device. In [22], the steps to create this driver using the NI-VISA Driver Wizard utility were described. In LABVIEW, a program for the PC to communicate with the microcontroller through the USB port using the VISA protocol was developed. A program was made for switching LEDsin small time intervals in order to verify that the correct driver was being used for the microcontroller. Furthermore, this program helped verify that the electronic equipment was configured correctly. We used a RAW-VISA communication scheme. We chose to use the RAW BULK data transfer, as it's the same as used in [24]. Moreover, the Measurement and Automation Explorer [22] tool that recognize USB devices that communicate with VISA protocol was used. This tool contains an interface called VISA Interactive Control [22], which allows us to send commands to the microcontroller and receive the results of what was read. This experiment helped us to verify that the device was recognized as a NI-VISA USB RAW device.

2) LabVIEW.

We designed a LabVIEW VIto read data from the micro controller. This interface program allows us to turn on and turn off a breadboard's LEDs that are connected to the micro controller. Then another program to read from the microcontroller using a cycle was implemented(Fig. 3, 4). In this way, the program could read data of a potentiometer while its voltage waschanged. The breadboard that was being used has several potentiometers. One of them was connected to the micro controller to be read by the microcontroller's AD converter. Using an oscilloscope we were able to validate the acquisition and communication scheme. The values from the AWG were sent to the VI in LABVIEW for storing the data in a text file. These results were then plotted in Matlab and the data was analyzed in order to compare with the graphs from the oscilloscope.

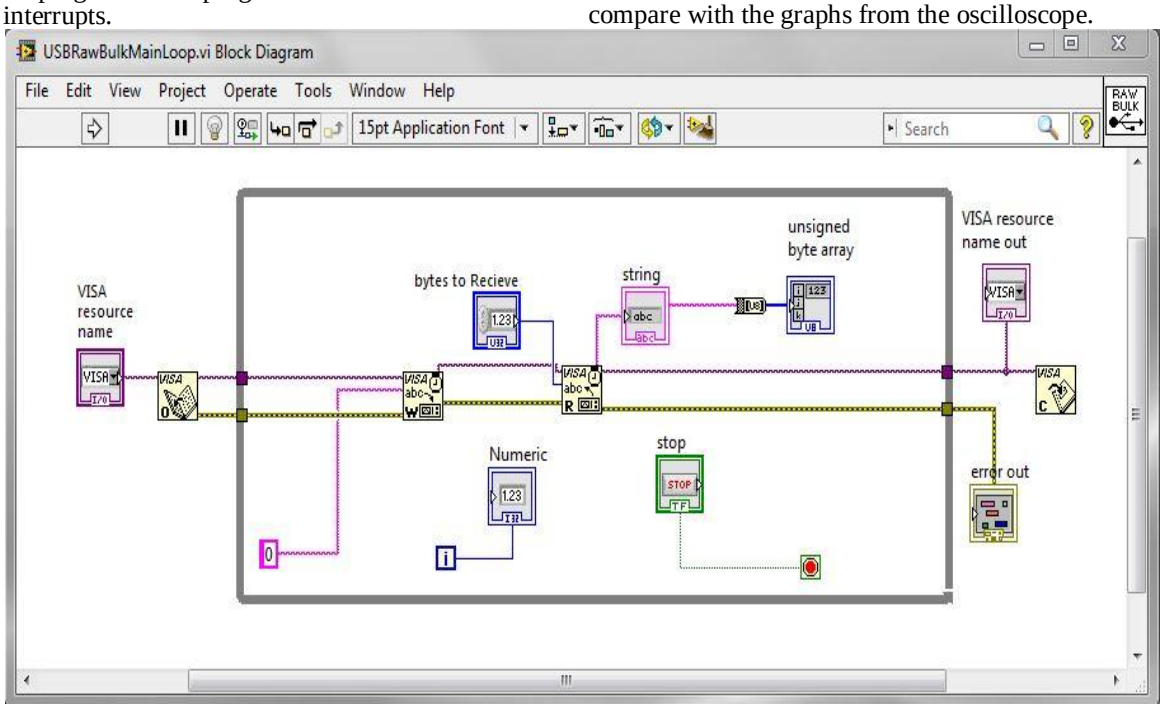

Fig. 3 Block diagram of a LabVIEW program used to read data from a USB-RAW device using loops.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

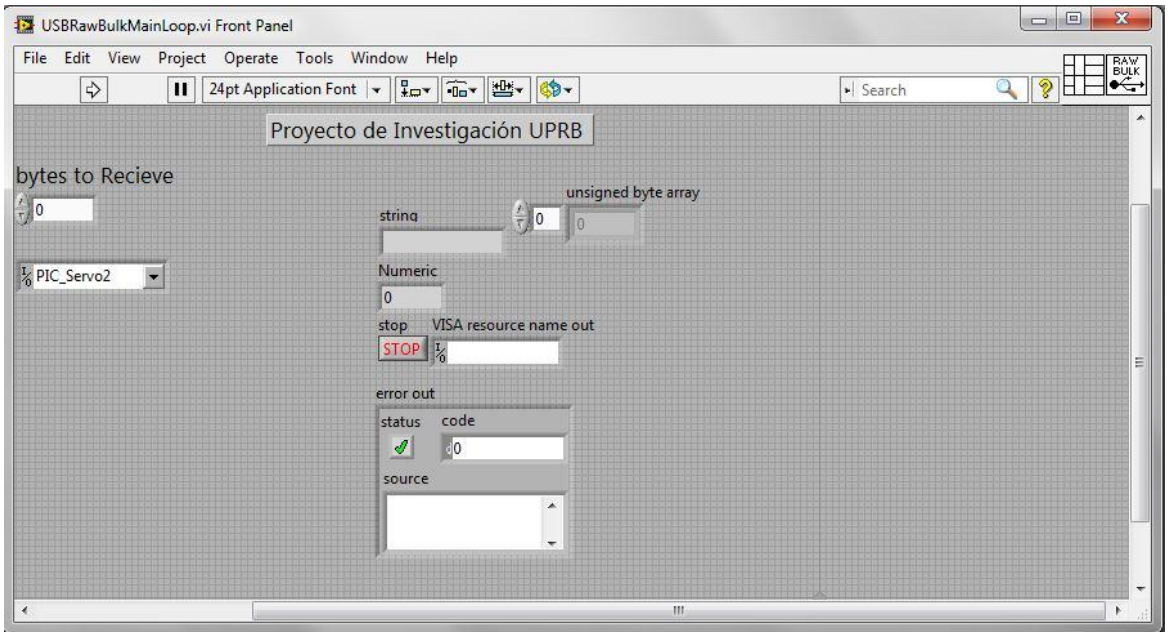

Fig. 4Interface program in LabVIEW of a program used to read data from a USB-RAW device using loops.

3) Matlab.

Another tool available in industry to communicate with the VISA is the Mathworks' Instrument Control Toolbox [16] for Matlab. This language was used for data acquisition, processing and analysis of results. This tool contains a basic implementation of communication tools, both VISA-USB as a Serial (real or emulated) port. This provides us the opportunity to create unified communication tools for all deployments. In addition to the implementations for reading a series of devices support functions for automation of experiments were set up by an arbitrary waveform generator via VISA-USB.

III. EXPERIMENTS

The experiments consist in stimulating each implementationwithan AWG and converting its signal for transmission over USB to the PC. This allows us to characterize our analog acquisition; in particular we focus on the frequency characteristics. The most important feature is the system'sresponse to frequency changes as suchit's linked directly to the performance of the system. Performance is measured asthe amount of samplesmeasured per unit time [18]. This measure should not be confused with the bandwidth used in the communications channel asthesample packetcontains multiple bytes. You can convert the output bandwidth of the channel by simply multiplying the number of bits in the data packet sent [18].

$$BW = 8\ bits * \frac{Bytes}{Packet} * \frac{Packets}{s}$$

To validate the results of the experiment, we will be looking for the maximum range of frequencies where the number of samples acquired is kept constant and were comparableto the estimated sampling time [21]. Nyquist's theorem [23] states that in a band limited signal sampling must occur at a frequency that is at least twice the highest frequency component found in the signal. This tells us that our maximum acquisition frequency will be limited by our sampling time. Because we don't know whatis the sampling time we measure frequencies from 1Hz to 100 kHz. In order to estimate the Nyquist frequency in the experiment we will look for aliasing byincreasingtheexcitation signal frequency until it the number of acquired samples starts fluctuating significantly. Thiswill present itself as a periodical anomaly that is repeated approximately every multiplicative integer sampling frequency of the system under test. The flowchart of the algorithm for measuring the experiment is shown in Figure 5.

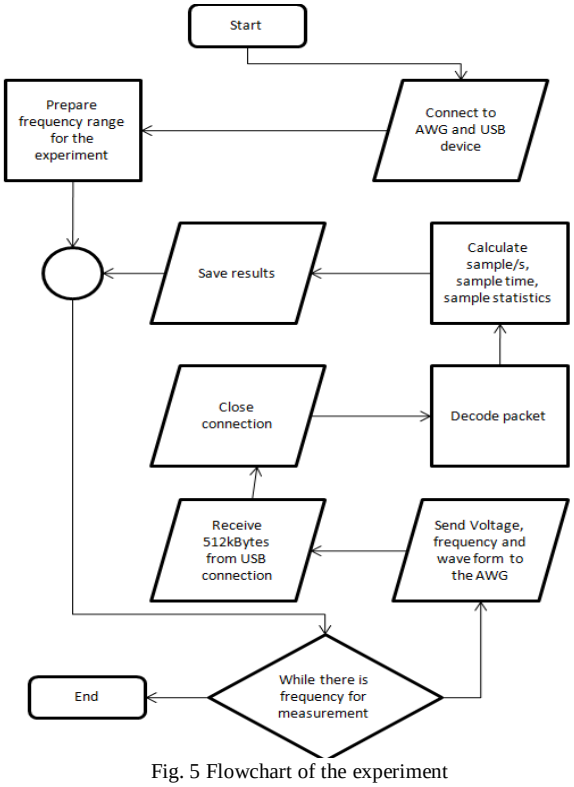

Fig. 5 Flowchart of the experiment

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

IV. DISCUSSION OF THE RESULTS

The most important parameter for the characterization of implementations is the sampling frequency. This parameter is obtained directly from the implementation and depends largely on the speed of the ADC converter in the micro controller. However, the transmission rate affects how often you can send a sample thus both affect the effective sampling time. This effect can be observed when measuring the average acquisition time per platform (see Table 2)

TABLE II
AVERAGE SAMPLING TIME AND SAMPLING FREQUENCY FOR EACH PLATFORM.

Platform	Sampling Average Time	Average Sampling Frequency
PIC18F4550	48.5282 µs	20.6066 kHz
ArduinoUNO (AVR)	210.3423 µs	4.7542 kHz
ArduinoChipkit (PIC32)	94.5218µs	10.5729 kHz

During the experiment, the acquisition time was calculated on the platform and the value was transmitted via USB to the PC. An interesting artifact of this formulation is that by having monotonically increasing frequencies we expect the acquisition times remain constant. The data show otherwise (see Fig.6). In order to define the point at which the sampling time shows signs of instability, the -3dB point for the limit was selected and the acquisition time averaged to that frequency. The lobes have repeated by each integer multiplied by the read frequency sample of the system. The lobes are the product of the lack of an anti-aliasing filter within the microcontrollers. Without this filter the conversion from analog to digital has no idea that it's sampling beyond its limit and begins to fail due to sampling rate being considerably slower than the one needed.

This is because the signal begins to change faster than the converter can acquire the signal and theacquisition algorithm,SAR,thinks that itendedearlier than would normally be expected.Using this information we can estimate a more realistic sampling time of each system using the data in Table 3.

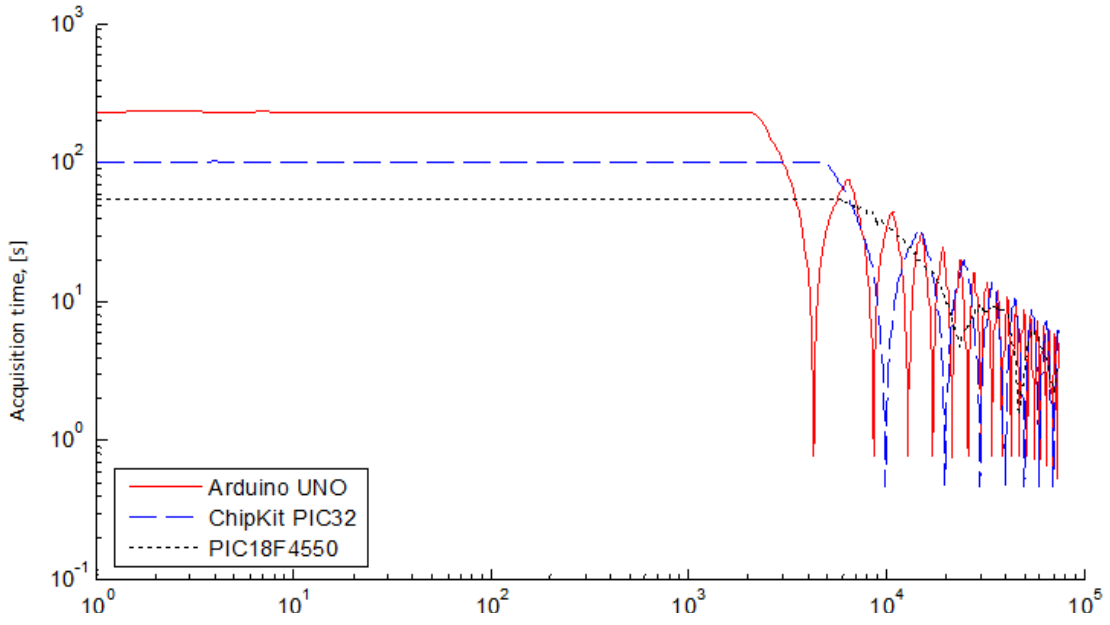

Fig. 6Frequency Vs Time of acquisition

TABLE I
ACQUISITION TIME OF THE DATA IN THE ESTIMATED SYSTEMS.

Platform	Sampling Time	Sampling Frequency
PIC18F4550	55 µs	18.1818 kHz
ArduinoUNO (AVR)	232 µs	4.3101kHz
ArduinoChipkit(PIC32)	100 µs	10 kHz

Once we know the sampling time we can characterize the system and compare it with the estimated sampling frequency. We can see in Fig.7 as the average of acquired samples show periodical patterns identical to those observed acquisition time. As a way to characterize the bandwidth of the communications channel the number of packets received is measured and divided by the time of acquisition of these packets. This measure does not suffer from artifacts caused by aliasing observed in the previous cases. The value is kept largely constant for the frequency sweep and it was averaged in Table 4. One of the things we can observe the characterization is the little variability of the measurements (<3.5%). With this information we can infer that the characterization is stable and independent of the excitation frequency of the system.

TABLE II CHARACTERIZATION OF THE PLATFORMS.

Platform	SamplesperSecond	Bandwidth
PIC18F4550	435,703.9±14,121	17.4±0.564Mbps
ArduinoUNO (AVR)	4,377.3± 3.13	175.1±0.125kbps
ArduinoChipkit(PIC32)	9,986.9 ± 0.675	399.5±0.03kbps

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING
Vol. 2, Issue 9, September 2014

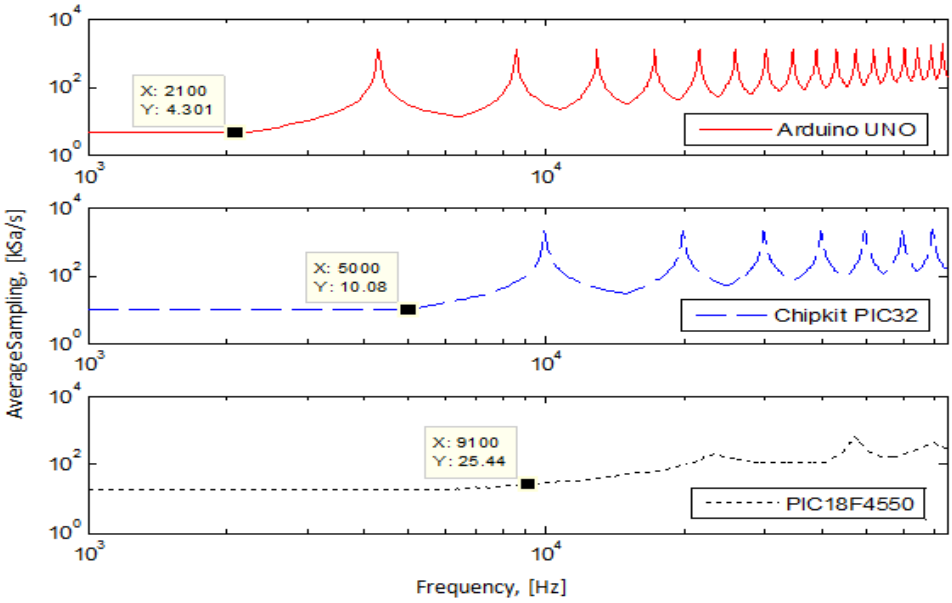

Fig. 7 Frequency Vs Average Sampling.

One of the curious details of these results is observed when comparing clock speeds for different implementations (see Table 5). However when comparing the frequency of acquisition of theoretical implementations (according to data sheet) with the measured values we find that there is room for growth in deployments.

TABLE V
CLOCK FREQUENCY COMPARISON FOR EACH IMPLEMENTATION.

Platform	Clock Frequency	Maximum Frequency found in datasheet
PIC18F4550	20 MHz	800 kSa/s
ArduinoUNO (AVR)	36 MHz	76.9kSa/s
ArduinoChipkit (PIC32)	80 MHz	1,000 kSa/s

Note that in the implementation of PIC18F4550, the difference can be attributed to the preferential interruptions that were implemented for USB communication. While the Arduino implementations had a Serial-USB communications channel that could not exceed 500kbps. With this we can estimate the time that each implementation was waiting whilethe transmission of samples took place. Using the estimated sampling (T_S) in Table 3 and the average time acquisition (T_{acq}) until the frequency begins to lose 3dB in Fig. 6. We can estimate the transmission time (T_{TX}) as T_{acq} and measurement time T_M as the difference T_S and T_{acq} (see Table 6).

TABLEVI
ESTIMATED AVERAGE TIMEOF THE ACQUIRED DATA.

Platform	T_S	$T_{TX} = T_{Acq}$	$T_M = Ts - T_{TX}$
PIC18F4550	55 µs	*D.* 48.5282 µs	*E.* 6.4718 µs
Arduino UNO (Atmel AVR)	232 µs	210.3423 µs	5.4182 µs
Arduino Chipkit (PIC32)	100 µs	94.5818 µs	21.6577 µs

V. CONCLUSIONSAND FUTURE WORK

The acquired data demonstrates the following: Implementing a VISA system is possible with the right resources. The Arduino platform is a viable and inexpensivealternative for the implementation of data acquisition systems. The Arduino boot loader creates a series of delays that could not be explained in the course of this research. The use of interrupts in microcontroller implementations allows us to optimizesystems so that resource utilization is maximized. The communications channel may be limiting applications for data acquisition factor. Between what is learned in the course of this research we can recommend future work: Implement the VISA-RAW on the PIC32 to compare the difference between the Serial-USB drive with raw USB. Use other architectures for implementation. Make a complete deployment VISA. Design a better acquisition system with commercially available ADCs.

ACKNOWLEDGMENT

The authors acknowledge the Electronics and Computer Research Laboratories of the Computer Science and Electronic Engineering Technology Departments of the University of Puerto Rico at Bayamón, for the support in this research. This work was funded by the University of Puerto Rico Bayamon Campus' Research and Development Initiative.

REFERENCES

[1] Anderson, D. (2001). Universal Serial Bus System Architecture. Addison-Wesley Professional.
[2] Anzalone, G.C.; Glover, A.G.; Pearce, J.M. (2013).Open-Source Colorimeter Sensors, 13, 5338–5346.
[3] Arduino.(2006). Arduino - Serial. URL: http://arduino.cc/en/Reference / Serial
[4] Axelson, J. (2009) USB Complete: The Developer's Guide (Complete Guides series).Lakeview Research; 4th edition.
[5] CCS.(2013). CCS Compiler URL http://www.ccsinfo.com/.
[6] Micros Designs.(2010). Comunicación USB con LabWindows/CVI y NI-VISA.URLhttp://web-archive-ar.com/ar/m/micros-designs.com.ar/2013-03-04_1541801_30/Micros_Desings/
[7] Digilent (2013).ChipKit.URL http://www.digilentinc.com/

[8] Folea, S. (2011).Practical Applications and Solutions using LabVIEW™ Software.Intech, Croatia.
[9] Futurlec (2013).PIC18F4550 Development board. URL http://www.futurlec.com/PIC18F4550_Board.shtml
[10] Kornuta, J.A.; Nipper, M.E.; Brandon Dixon, J. (2013).Low-cost microcontroller platform for studying lymphatic biomechanics in vitro. J. Biomech. 46, 183–186.
[11] Hjertaker, B.T.; Maad, R.; Schuster, E.; Almas, O.A.; Johansen, G.A. (2008).A data acquisition and control system for high-speed gamma-ray tomography. Meas. Sci. Technol., 19,doi:10.1088/0957-0233/19/9/094012.
[12] Hyde, J. (2001). USB Design by Example: A Practical Guide to Building I/O Devices (2nd Edition). Intel Press.
[13] Ibrahim, D. (2008). Advanced PIC Microcontroller Projects in C: From USB to RTOS with the PIC 18F Series. Newnes.
[14] Inc., Microchip Technology. (2009) PIC18F2455/2550/4455/4550 Data sheet. URL http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf
[15] Inc., Microchip Technology. (2011) PIC32MX3XX/4XX Data sheet. URL http://ww1.microchip.com/downloads/en/DeviceDoc/61143H.pdf
[16] Ingle, V. K., &Proakis, J. G. (2011).Digital Signal Processing Using MATLAB. CL Engineering.
[17] Jobbins, M.M.; Raigoza, A.F.; Kandel, S.A. (2012).Note: Circuit design for direct current and alternating current electrochemical etching of scanning probe microscopy tips. Rev. Sci. Instrum. 83,doi: 10.1063/1.3695001.
[18] Malaric, R. (2011).Instrumentation and Measurement in Electrical Engineering.Brown Walker Press.
[19] Manosalvas, A. &Quinga, J. (2012). Diseño y construcción de un módulo de adquisición de datos para la supervisión y control de una mini planta de procesos con interfaz USB para LabView. Tesis de Ingeniería Eléctrica. Universidad Politécnica Salesiana. Quito
[20] Microchip Technology Inc. (2012). Microchip Technology Inc: http://www.microchip.com/
[21] Mitra, S. (2010).Digital Signal Processing with Student CD ROM. McGraw-Hill.
[22] National Instruments (2014).USB Instrument Control Tutorial. URL www.ni.com/white-paper/4478/en/pdf
[23] National Instruments (2006).Nyquist Theorem Sampling Rate Versus Bandwidth. URL http://www.ni.com/white-paper/4653/en/
[24] Pérez-Moret, Y. (2012). Implementación de Comunicación USB con Microcontrolador PIC18F4550 y LabVIEW. URL www.youblisher.com/pdf/481469
[25] Pérez-Moret, Y. (2012). Diseño y construcción de un espectrómetro óptico por USb. Publisher http://lulu.com
[26] Teikari, P.; Najjar, R.P.; Malkki, H.; Knoblauch, K.; Dumortier, D.; Gronfier, C.; Cooper, H.M.(2012). An inexpensive Arduino-based LED stimulator system for vision research. J. Neurosci. Methods 211, 227–236.
[27] Zachariadou, K.; Yiasemides, K.; Trougkakos, N. (2012).A low-cost computer-controlled Arduino-based educational laboratory system for teaching the fundamentals of photovoltaic cells. Eur. J. Phy. 33, 1599–1610.
[28] Zoric, A.C.; Perisic, D.; Obradovic, S.; Spalevic, P. (2011).PC-Based Virtual DTA Recording System Design.Prz. Elektrotech. 87, 156–160.

BIOGRAPHIES

Dr. Elio Lozano received B.S. in mathematics at National University of San Antonio Abad del Cusco, Perú in 2000. He received M.S. in Scientific Computing and Ph.D. in Computer and Information Science and Engineering at University of University of Puerto Rico at Mayaguez Campus in 2003 and 2006 respectively. Dr. Lozano is with the department of Computer Science, University of Puerto Rico at Bayamón, San Juan, PR, 00959 USA. He has developed several software and firmware for different architectures.

Elias Beauchamp received B.S. and MS in Computer Engineering at University of Puerto Rico at Mayaguez. Beauchamp is with the department of Electronics, University of Puerto Rico at Bayamón, San Juan, PR, 00959 USA. He has developed several software and firmware for different architectures.

.