

Flexible PC Measurement System Based in Extensible Mark-Up Language Files

Nuno B. Brás¹, Pedro M. Ramos¹ and A. Cruz Serra¹

¹ *Instituto de Telecomunicações and Department of Electrical and Computer Engineering, Instituto Superior Técnico, Technical University of Lisbon, IST, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal
Phone: +351-218418485 Fax: +351-218418472 e-mail: nuno.bras@Lx.it.pt*

Abstract- For the development of advanced measurement algorithms, it is common to handle large organized data resulting from multiple acquisitions for post processing. Most of the times the saved measured data does not include relevant information like the context in which the data was acquired, if there is other related data to the acquisitions and not even the hardware involved. Some systems solve this problem by creating a batch of files containing the information but reducing the test data portability, compression and consistency.

This paper describes a PC based acquisition, archive and post-processing system that works as a generic mainframe for measurement tests performed with data acquisition devices. This system allows structuring several acquisition tasks, transmitting the resulting files through web, save memory space and is scalable for computer languages that process Extensible Mark-up Language (XML) files.

I. Introduction

Modern measurement systems are based on efficient numerical algorithms to extract relevant parameters from data acquisitions. In order to develop new algorithms, large amounts of real data are needed. However, in some applications it is not practical to develop the algorithms as the acquisitions are performed either because the measuring phenomenon occurs rarely or because the measuring location is inhospitable. In these situations the best solution is to acquire a large set of acquisitions, maybe varying some system parameter, store the complete set of data in a hard drive or transmit it through the web and then develop the algorithms using as inputs the stored data.

In this paper, a mainframe that allows the acquisition and processing of data for algorithm development in many instrumentation fields is described. The acquisition system generates a XML file that is software and hardware independent and that includes all the acquisition related information. It is also important that the resulting data has an optimal storage size and allows easy web transmission. The XML allows organizing structured data and is a flexible file type to send over the internet. It is an independent language standard with broad acceptance in internet technologies.

In addition to the described use, the system may be employ to test developed algorithms by the introduction of simulated data acquisition exported as a XML file.

II. System Architecture

The system is composed of two basic program modules: an acquisition module and a processing module. Each module has two implementations: one in LabVIEW and one in Matlab. The XML storage file is compatible with all the modules enabling its usage by developers in either programming languages. If the developer wishes to use a different programming language, the modules can be easily programmed to read/write the common XML file format. In Figure 1 a basic system flowchart is shown.

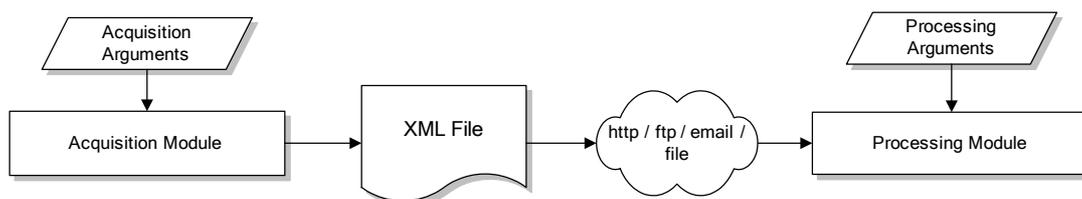


Figure 1 – Basic system flowchart including the acquisition and processing modules and the XML file.

The acquisition modules receive the input arguments by the user or imbedded in a XML input file, that define the acquisition process. These parameters include the acquiring device (GPIB instrument, PCI Card, USB, Bluetooth, etc.), and how each acquisition is processed (including external commands to other devices).

The developed process modules can be used embedded in other programs or as stand alone programs. It is possible to define what specific acquisitions, channels and acquired points will be processed and to specify how many bits to use. This allows simulating how an algorithm will work with the ADC acquisition resolution changed. Each run works as a new acquisition where even the bit rate can be altered within certain limits. As it was said, it can be also used to simulate data if specified correctly as a XML data file. The output is an array of data and parameter variables that can be manipulated by other programs.

III. XML file structure

The XML file includes information organized hierarchically as shown in Figure 2. The only information common to all acquisitions is the acquisition device and its number of bits. Also defined in the XML root section is the number of acquisitions, a location string, comments and a project name. In the AQS section, there is one sub-section for each acquisition which includes one sub-section for each acquired channel. For each channel a amplitude increment and offset are included together with the number of points, reference channel delay, the acquired values and the ADC range.

```

<?xml version="1.0" encoding="ASCII" ?>
- <root>
  <init_timestamp>09-06-2005 10:23:45</init_timestamp>
  <local>IST-LAB.IM</local>
  <instr_code>DAQ6013</instr_code>
  <bit_num>16</bit_num>
  <comments />
  <project>IMEKO Tests</project>
  <n_aq>5</n_aq>
- <aqs>
  - <aq1>
    <timestamp>09-06-2005 10:23:45</timestamp>
    <rate>10000</rate>
    <n_ch>2</n_ch>
    - <chs>
      - <ch1>
        <ch_name>0</ch_name>
        <yinc>1.525879E-4</yinc>
        <offset>0.000000E+0</offset>
        <aqpoints>2000</aqpoints>
        <delay>0.000000E+0</delay>
        <values>51v5XCJeaF/dYI9ic2RiZkBo+4dYlc(.....)WgAA</values>
        <range_max>5.000000E+0</range_max>
        <range_min>-5.000000E+0</range_min>
      </ch1>
      - <ch2>
        <ch_name>1</ch_name>
        <yinc>1.525879E-5</yinc>
        <offset>0.000000E+0</offset>
        <aqpoints>2000</aqpoints>
        <delay>5.000000E-5</delay>
        <values>i2k3avZq0GvEb0ltGW9KcGtxcHFhNn(.....)jaAAA</values>
        <range_max>5.000000E-1</range_max>
        <range_min>-5.000000E-1</range_min>
      </ch2>
    </chs>
  </aq1>
  + <aq2>
  + <aq3>
  + <aq4>
  + <aq5>
</aqs>
</root>

```

Figure 2 – XML file hierarchy for five acquisitions <aq1> up to <aq5> with a 16 bit DAQ. In the first acquisition, the samples are acquired from channel 0 and channel 1 with 2000 points acquired at 10000S/s. The full acquisition data in the <values> field was cut from this figure to improve readability. Details from the other four acquisitions are collapsed much like a directory view in a computer. This is the typical view if the XML file is opened in a web browser.

The acquired data is encoded using base64 [1] and the resulting string is stored in the XML file. Base64 encoding is widely used to convert binary data into a string of printable characters at the expense of an increased number of bytes. Each byte of binary data is composed of eight bits. However, most programs and transmission systems are only ready to support 7-bit ASCII. To prevent bit loss, base64 is used. In the encoding process, the binary data to be transmitted is divided into 3 byte groups (24 bits). These bits are divided into 4 groups of 6-bit sets (each set has values from 0 to 63, hence the name base64 coding). To avoid the use of special characters included in the 0 to 63 range of ASCII, mapping is used from the original 64 possible codes to printable characters (26 uppercase letters, 26 lowercase letters, 10 digits, “/” and “+”). One of the biggest applications of base64 is the coding of binary files for transmission by email clients [2]. This coding is used at a lower level and most users don’t even know that they are using it.

In the processing module, the channel acquisition binary data is transformed into the original acquired voltages by applying the increment and offset data after the base64 decoding. The delay information at the channel level describes the typical acquisition card delay between channels when only ADC is used for multiple channel acquisition systems (each channel is acquired at consecutive instants but not simultaneously).

Since XML is straightforwardly usable over the Internet [3], the file can be sent to other computers using any protocol that supports XML documents (http, ftp, e-mail) without the need for binary coding. A comparison between several compressing data processes and its usability capabilities is presented in the Table 1. The file sizes result from 50k acquisition values with 16 bits resolution.

	File size (Kbytes)	Single File?	Web Portability?	Multi-software platform?
XML file with Base64 encoding	130,2	Yes	High	Yes
Binary file	97,7	No	Low	Yes
Text file	732,0	Yes	Low	Yes
Matlab data file	98,0	Yes	Low	No
LabVIEW Waveform Binary File	391,3	No	Low	No

Table 1 – Comparison between several file types for acquisition data logging corresponding to a total of 50k samples acquired in a 16 bit acquisition system.

Low web portability means that the file is portable by e-mail and ftp, and high web portability means that it is also possible to use http. It can be seen that, despite the 33% increased file size when compared to a binary file, the proposed XML file is the only that can join the data from several acquisitions, that is completely web portable and that can be processed by several software platforms. The 33% ratio is derived from the 4/3 base64 coding process, where each 3-byte set is encoded by 4 ASCII bytes.

IV. Application Example

To demonstrate the system capabilities, it was used to acquire and store in a XML file the output voltages of two Hall sensors. The sensors are monitoring the current and voltage supplied to a load by the electrical distribution grid. This setup is used to monitor power quality and will be the backbone for the development of new algorithms to efficiently detect, classify and quantify power quality disturbances.

The voltage sensor outputs voltages in the –2 to 2 V range for the 230 V nominal grid voltages. The current sensor has an output voltage in the –0.2 to 0.2 V range for a nominal current of 0.4 A. A National Instruments DAQ is used to acquire the sensor voltages. The device used (DAQ-6013) has 16-bit resolution and ranges ± 50 mV, ± 0.5 V, ± 5 V and ± 10 V. The sensor voltage channel is set for the ± 5 V while the current sensor channel is used in the ± 0.5 V range.

The load used is a RC series with $R = 300 \Omega$ and $C = 5.1 \mu\text{F}$. For the system test, 500 acquisitions were performed at a sampling rate of 10 kS/s and with 20000 samples per channel – the total number of samples is 20M. Since each sample has 16 bits, it requires two 8-bit characters before base64 coding. For this situation, the pure binary file would have a size of 40 000 000 characters while the XML file has a length of 53 608 044 bytes (includes the 33% ratio and the characters necessary for acquisition

parameters definition).

In Figure 3, the front panel of a VI which processes the XML file is shown. This VI calls the LabVIEW Processing Module which opens the XML file and outputs an array of waveforms for processing.

In this very simple example, the VI determines the average power spectrum of the current and voltage, the average voltage RMS and current RMS as well as the average phase difference between the current and voltage. These values are used to determine the load impedance at the electrical grid frequency.

It should be noted that while in this case the XML file is located in the processing PC, it is possible to introduce a URL to process a file published in the web.

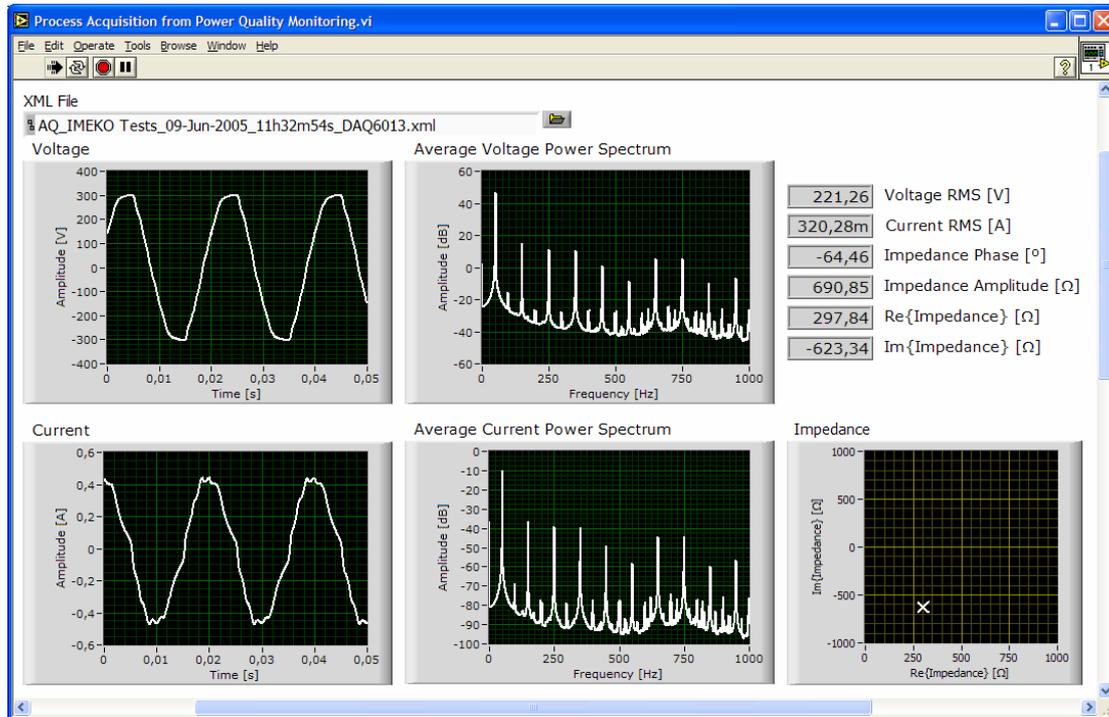


Figure 3 – VI Front panel to process Power Quality Monitoring acquisitions stored in a XML file.

V. Conclusions

Measurement systems that use large amounts of data in several acquisition series with different acquisition conditions may benefit from this versatile system. Examples include ADC characterization [4], impedance measurements [5] and power quality measurements [6]. This platform will be used in several projects for algorithm development strategies.

References

- [1] Paul V. Biron, Kaiser Permanente, Ashok Malhotra, “XML Schema Part 2: Datatypes Second Edition” *W3C Recommendation*, Oct. 2004.
- [2] RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One, Format of Internet Message Bodies*, Section 6.8 - Base64 Content-Transfer-Encoding, available at www.imc.org/rfc2045
- [3] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, “Extensible Markup Language (XML) 1.0 (Third Edition)” *W3C Recommendation*, Feb. 2004.
- [4] A. C. Serra, F. Alegria, R. C. Martins and M. F. Silva, “Analog to digital converters testing – New proposals”, *Journal of Computer Standards & Interfaces*, vol. 26, pp. 3-13, Jan. 2004.
- [5] P. M. Ramos, M. F. Silva and A. C. Serra “Impedance Measurement using a Least-Squares Waveform Fitting Algorithm”, *IEEE Conf. on Precision Electromagnetic Measurements*, London, United Kingdom, vol. 1, pp. 125-126, June 2004.
- [6] G. Bucci, E. Fiorucci, and C. Landi, “Digital Measurement Station for Power Quality Analysis in Distributed Environments” *IEEE Trans. on Instrumen. and Meas.*, vol. 52, pp. 75-84, Feb. 2003.