TECHNICAL TRANSACTIONS | CZASOPISMO TECHNICZNE
MECHANICS | MECHANIKA

2-M/2015

GRZEGORZ FILO[*]

ASYNCHRONOUS BUFFER READ METHOD IN DEVELOPMENT OF DAQ APPLICATION FOR SUPPORTING RESEARCH OF HYDRAULIC SYSTEMS

METODA ASYNCHRONICZNEGO ODCZYTU BUFORA W BUDOWIE APLIKACJI DAQ WSPOMAGAJĄCEJ BADANIA UKŁADÓW HYDRAULICZNYCH

Abstract

The paper presents a method of data acquisition from a DAQ card using a direct memory access (DMA) and an asynchronous read of data buffer with the FIFO queue. The method can be used to acquire measurement data in Windows™ system when the required read frequency is higher than the system's clock frequency. The method was used in practice to measure high-speed pressure and position changes in a two cylinder hydraulic system with two-state fast switching valves.

Keywords: *DAQ system, asynchronous data buffer, software development, object oriented programming*

Streszczenie

W artykule przedstawiono metodę odczytu danych z karty pomiarowej z wykorzystaniem bezpośredniego dostępu do pamięci (DMA) oraz asynchronicznego odczytu bufora danych i kolejki FIFO. Metoda ta ma zastosowanie do akwizycji danych pomiarowych w systemie Windows, jeżeli wymagana częstotliwość odczytu jest większa od częstotliwości zegara systemowego. Opracowaną metodę zastosowano do pomiaru szybkich zmian ciśnienia i położenia w układzie z dwoma siłownikami i dwustanowymi szybkimi zaworami przełączającymi.

Słowa kluczowe: *system akwizycji danych pomiarowych, asynchroniczny bufor danych, budowa oprogramowania, programowanie obiektowe*

[*] Ph.D. Grzegorz Filo, Institute of Applied Informatics, Cracow University of Technology.

1. Introduction

Most activities related to science and research require the use of data acquisition systems (DAQ) in order to measure many kinds of physical parameters. The fundamental information on the techniques, methods and devices used in the DAQ systems can be found in [1]. Usually, the systems consist of A/D and D/A transducers and their operation is strictly connected with data conversions, which is presented in [2].

Presently, several different types of DAQ systems are used, as stated in [3]. The first option is to use a computer-based system with a DAQ card installed. In this case, the card is responsible for performing the A/D and D/A conversions, acquiring data and storing it in the memory. The computer software communicates with the card via the dedicated driver and performs processing, storage and visualization of the data. The next possibility is to use a system based on autonomous devices, which, however, may be connected to the computer directly using a serial or a parallel communication port. Systems of this kind are usually more expensive, although they are able to achieve higher data reading frequency. On the contrary, the modular data acquisition system consists of functional units connected by a dedicated system bus. The system may be individually assembled and configured to perform a particular task. Usually, the modular DAQ systems are fully self-contained and do not require additional hardware.

In this work a computer-based system is considered, because it is a low cost, universal solution which can be used in various experiments without the need to replace hardware elements. A typical DAQ card for a computer-based system (Fig. 1) contains analog input (AI) and digital input (DI) ports. Some models also have analog output (AO) and digital output (DO) ports. The sampling rate of A/D and D/A converters installed in low-cost cards is usually in the range of 100 kHz to 1 MHz. [4].

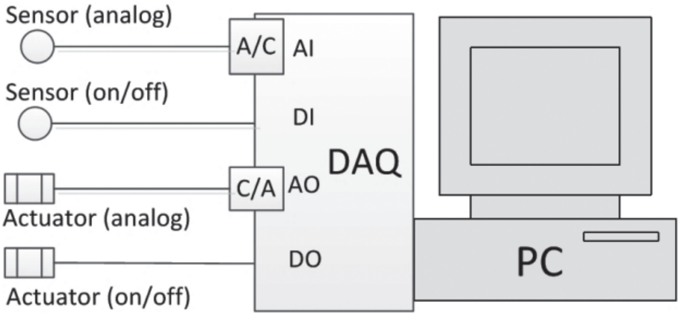
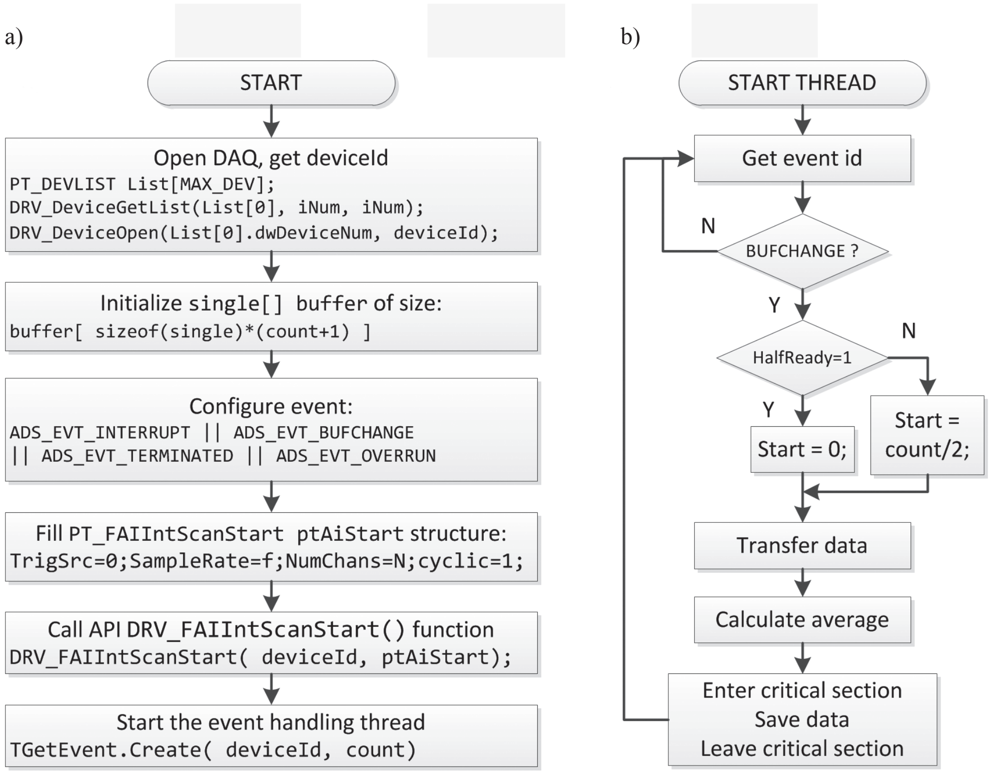


Fig. 1. Typical computer-based DAQ system

2. Algorithm and implementation

Application programming interface provided by a DAQ device driver allows a programmer to use two modes of analog input data read by a software-triggered function or an interrupt function [5, 6]. The software-triggered mode is based on a system timer [6]. Therefore, it does not allow the driver to obtain a high read frequency. Depending on the hardware, the maximum sampling rate in this case is from 20 Hz to 50 Hz. In contrast, data transfer based on interrupts is performed in the background. It, thus, uses less CPU time and speeds the transfer rate. Therefore, this method is recommended for larger amounts of data transfer at higher rates. In general, there are two kinds of interrupt operations for analog input. One generates a hardware interrupt for each conversion, the other one keeps the conversion data in a first-in-first-out (FIFO) queue, then generates a hardware interrupt for half-full of FIFO, or full of FIFO.

The method presented in this work is based on the half-full FIFO hardware interrupt generation. The applied algorithm is presented in Fig. 2.



```
PT_DEVLIST List[MAX_DEV];
DRV_DeviceGetList(List[0], iNum, iNum);
DRV_DeviceOpen(List[0].dwDeviceNum, deviceId);
```

```
buffer[ sizeof(single)*(count+1) ]
```

```
ADS_EVT_INTERRUPT || ADS_EVT_BUFCHANGE
|| ADS_EVT_TERMINATED || ADS_EVT_OVERRUN
```

```
TrigSrc=0;SampleRate=f;NumChans=N;cyclic=1;
```

```
DRV_FAIIntScanStart( deviceId, ptAiStart);
```

```
TGetEvent.Create( deviceId, count)
```

Fig. 2. Algorithm of data acquisition based on half-full FIFO hardware interrupt generation: a) initializing code, b) event handling thread

As can be seen from Fig. 2, the code is divided into two main parts. The first one contains initializing instructions: opening the DAQ device, initializing memory buffer, configuring event id's, filling the proper data structure initializing interrupt data transfer and starting the event handling thread. The second part consists of the thread main loop. At the beginning the event id is read. If the id is `ADS_EVT_BUFCHANGE`, the `HalfReady` flag is checked. On the basis of the value, the first or the second half of the data buffer is transferred into the data array. Then, the average value is calculated and stored in the global array using the `TRTLCriticalSection` structure.

3. Object of study

A diagram of a hydraulic system used to test the developed DAQ algorithm is presented in Fig. 3.

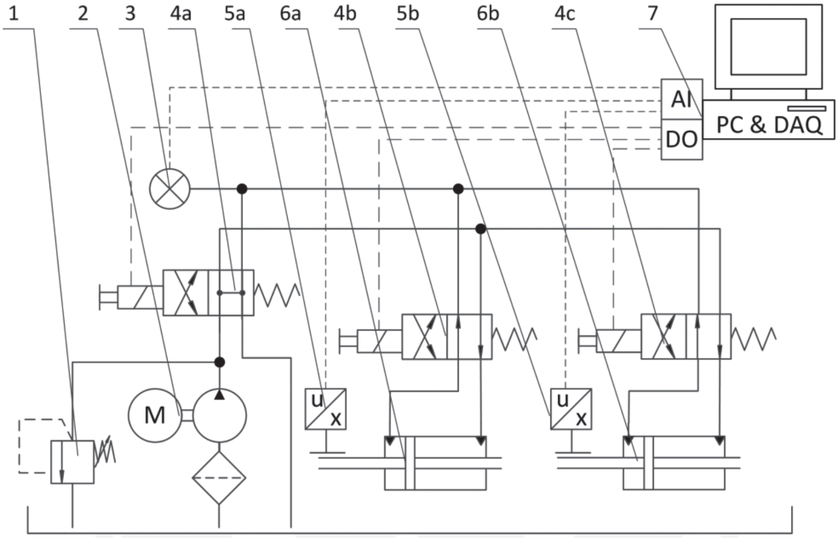


Fig. 3. Hydraulic system: 1 – relief valve, 2 – supply unit, 3 – pressure transducer, 4a, 4b, 4c – control valves, 5a, 5b – displacement transducers, 6a, 6b – hydraulic cylinders, 7 – DAQ system

The hydraulic system contains a supply unit 2 with a proportional relief valve 1, the main control valve 4a and two subsystems (4b, 6a and 4c, 6b), each consisting of a control valve and a hydraulic cylinder. The DAQ system acquires data from a pressure transducer 3 and two displacement transducers 5a, 5b through the AI ports. It also generates binary control signals for control valves using the DO ports. The created test bench is shown in Fig. 4.

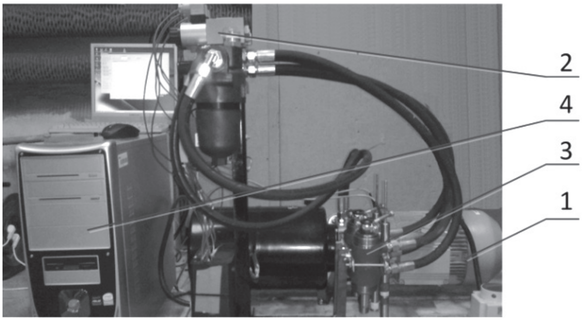


Fig. 4. Test bench: 1 – supply system, 2 – control valves, 3 – hydraulic cylinders, 4 – DAQ system

4. Measurement results

The following values of the DAQ program parameters were set in order to investigate the hydraulic system: the base sampling frequency f_{base} = 32 kHz, number of AI channels to read n_{c} = 4, number of samples used to calculate the average value in each step n_{s} = 8. Therefore, the resulting sampling frequency f_{res} was:

$$f_{res} = \frac{f_{base}}{n_c \cdot n_s} = 1\ \text{kHz} \tag{1}$$

The experiments consisted in setting the input signals for each control valve and investigating the system response. The input signal was in the form of a set of rectangular pulses. The response included signals from the displacement transducers and from the pressure transducer. Sample result plots of the pressure and the cylinder displacements are presented in Fig. 5, while in Fig. 6 are shown selected time charts of cylinder 2 displacement in a narrow time interval.

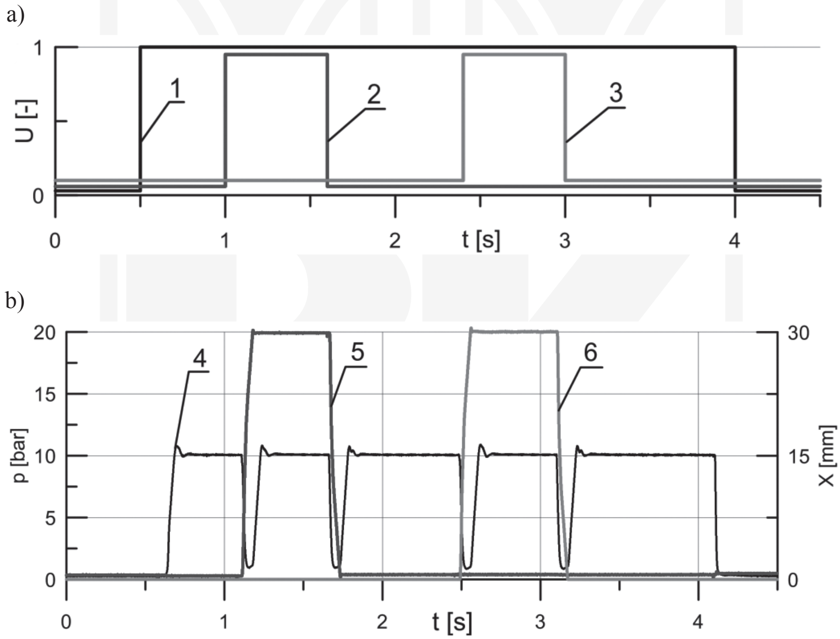


Fig. 5. Time plots: a) input functions, b) time charts; 1 – main control valve, 2, 3 – control valves of cylinders, 4 – pressure, 5 – cylinder 1 displacement, 6 – cylinder 2 displacement

As can be seen from Fig. 5–6, the delay time between setting the input function value and starting the cylinder movement is about 0.09–0.11 s. The movement over a distance of 0.03 m takes about 0.06–0.07 s.

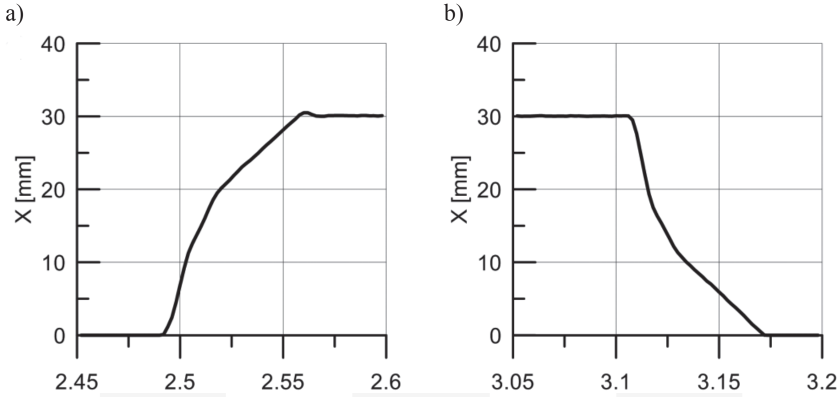


Fig. 6. Section of the cylinder 2 movement chart: a) forward movement, b) backward movement

5. Conclusions

This paper concerns the issues of a software development for computer-based DAQ systems. The software was built using the techniques of the object-oriented programming and multi-thread programming. The presented practical application involves using software for acquiring data in the 2-cylinder hydraulic system. The resulting plots of signals indicate that this method allows for reliable and accurate results to be obtained.

The proposed half-buffer read method is universal. Although the sufficient sampling frequency used in the presented experiments was 1 kHz, the tests up to 10 kHz (4 AI ports) or 64 kHz (1 AI port) were also carried out and were successful. The method is limited only by the DAQ card parameters and the computer system's performance.

References

[1] Maurizio Di Paolo E., *Data Acquisition Systems*, Springer, 2013.

[2] Analog Devices Inc., edited by Kester W., *The Data Conversion Handbook*, Newnes 2005.

[3] Ferrero Martín F.J., Valledor Llopis M. et.al., *Low-cost open-source multifunction data acquisition system for accurate measurements*, Elsevier, Measurement, Vol. 55, 2014, 265-271.

[4] Advantech, *Advantech Device Driver Manual*, Advantech Industrial Automation, http://www.advantech.com/industrial-automation, 2011.

[5] Włodarczyk P., Pustelny S., Budker D., Lipiński M., *Multi-channel data acquisition system with absolute time synchronization*, Elsevier, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Vol. 763, No. 2014, 150-154.

[6] Patel J.J., Reddy N. et.al., *Embedded Linux platform for data acquisition systems*, Elsevier, Fusion Engineering and Design, Vol. 89, Issue 5, 2014, 684-688.