Fast Data Acquisition Systems

Subjects: Engineering, Electrical & Electronic

Contributor: Haris Turkmanović , Mihajlo Karličić , Vladimir Rajović , Ivan Popović

Data acquisition is the process by which physical phenomena from the real world are transformed into electrical signals that are measured and converted into a digital format for processing, analysis, and storage by a computer. In a large majority of applications, the data acquisition (DAQ) system is designed not only to acquire data but to act on it as well. In defining DAQ systems, it is, therefore, useful to extend this definition to include the control aspects of the total system. A DAQ system is comprised of a computer running specialized software and several data acquisition cards and/or instruments (from now on jointly referred to as DAQ devices). Data acquisition (DAQ) systems are intensively used in laboratory research, industrial facilities, aviation, automotive, etc. The applications include test and measurement, automation, etc.

data acquisition system network embedded software

1. Introduction

A data acquisition (DAQ) system can have either a relatively slow or a relatively fast data transmission speed; it can store and display collected data locally on the main DAQ board, or it can send data to be stored and observed remotely; it can have its own ADC or just collect sampled data from an external ADC; etc. The flexible definition of a DAQ system allows many different types of systems to be classified as DAQ systems. This can make researching and comparing different systems a challenging task, and it can make research results hard to interpret.

2. FPGA-Based Fast DAQ Systems

The main reasons FPGAs are used for designing DAQ solutions is their ability to operate at very high clock frequencies with the possibility of parallel processing, in turn, enabling fast processing of acquired data. Power can be conserved on FPGAs by instantiating only hardware components that are necessary for a specific application. The potential for an increase in performance comes at the cost of a higher price, lower portability, and more complicated reprogramming when compared to microcontroller-based solutions. It is for these reasons that some FPGA-based DAQ systems are used for very specialized applications.

A DAQ system with fast processing and transfer speeds in mind has been designed in ^[1]. The design of a highspeed FPGA-based DAQ system consists of an ADC with a 16-bit resolution for data conversion, two smaller-sized buffers used to store the temporary data samples, and a DMA that was programmed to achieve data transfers without loss of a single data sample when transferring sampled data from the temporary buffers assigned to the ADC to the DD3 SDRAM which is used as permanent storage. On the FPGA, a soft processor core was instantiated and tasked with initiating new DMA transfers. By overcoming the problems caused by the mismatch of operating frequencies between DMA, ADC, and SDRAM, the authors achieved a maximum acquisition rate of 10 MSps. The downside of such a design is the use of on-board memory as permanent storage, which makes it harder to utilize a PC to visualize and further analyze the acquired data, thus without the possibility to analyze the data in real-time.

Designing a highly specialized FPGA-based DAQ system was the approach taken in ^[2]. This design presents a multichannel, high-speed DAQ system that supports acquisition for long periods of time, all of which are requirements of the Experimental Advanced Superconducting Tokamak (EAST) device ^[3]. In the design, four ADCs are connected to an FPGA, which is then connected to a data server through a Peripheral Component Interconnect (PCI) Express interface. This allows acquisition from multiple channels synchronously with the acquisition rate of 80 MSps (20 MSps per channel) and the precision of 12 bits, with the acquisition lasting more than 1250 s of continuous sampling. Synchronous sampling of the four ADCs was achieved by instantiating a clock configuration circuit on the FPGA, which provides four phase-shifted clock signals. To extend the maximum continuous acquisition time, the authors have implemented the Lempel-Ziv-Oberhumer (LZO) algorithm ^[4] on the FPGA, which allows for fast compression and decompression. By employing FPGA parallel processing mechanisms, data acquisition efficiency was not affected by compression. Even though this DAQ system was developed for a specific use case, data visualization could be improved because the paper features only a digital waveform view of the captured data in an unmentioned software tool.

A similar approach was taken in ^[5], a design of a DAQ system developed for Superheated Emulsion Detectors (SEDs) ^[6]. The paper presents a 10-channel, 12-bit resolution high-speed DAQ system with acquisition rates of 500 kSps per channel. The design features triggered acquisition and a user-programmable threshold with pre- and post-trigger adjustments. This allows the user to choose how many samples before and how many samples after the automatic or manual trigger are desired to be captured. Because of the specific application of this DAQ system, all 10 channels need to be sampled at once. Each of the 10 analog signals is connected to a signal conditioning board, followed by an ADC board, which is then connected to the main FPGA board, which features the main processor. All the 10 ADCs use the same sampling clock, and all the conversion results are read at the same time, stored in a circular buffer, and the converted values are compared with a threshold value. If the threshold value is exceeded, a configured number of samples before and after the threshold sample are collected and sent to a PC using a Universal Asynchronous Receiver/Transmitter (UART) interface ^[7]. The DAQ system uses a graphical user interface (GUI) designed in LabVIEW ^[8] to provide visualization of the collected data. Although the maximum possible acquisition rate simultaneously over all the channels is 5 MSps, UART communication technology does not allow for a higher baud rate than 921,600 bps, limiting the total system throughput when acquiring signals over longer periods of time. As soon as the internal FPGA buffers get filled, acquisition stops until all the samples are transmitted.

3. Microcontroller-Based Fast DAQ Systems

When a DAQ system is developed for a general-purpose application, microcontroller-based DAQ systems are generally a better choice. If needed, microcontroller-based DAQ systems are more easily adapted to a specific application thanks to easier reprogramming and changes in existing software. If a system is designed with portability in mind, it can be relatively easy to adapt the design from one microcontroller platform to another. The availability of different microcontrollers and the fact that more developers are better acquainted with microcontroller programming than FPGA programming also greatly increases the popularity of microcontroller-based DAQ systems over FPGA-based systems. Microcontrollers have grown into powerful yet complicated systems, but the utilization of an RTOS facilitates programming without wasting too much theoretical efficiency. This way, microcontrollers can even emulate the parallelism of FPGA.

A DAQ system used for high-speed current sensing is described in ^[9]. The main purpose of this system is to detect faults during welding and mark damaged joints by comparing the recorded data to a learned pattern. The system consists of a sensor board that measures welding voltage and current waveforms with additional sensors and evaluation software running on a remote PC. The sensor board consists of a measurement part and a processing part that includes ADCs, a microcontroller, external memory and an Ethernet PHY module. A whole welding sequence can be buffered locally on the board. The board uses two 16-bit ADCs connected via SPI bus to the microcontroller. The sampling rate is 82 kSps, with averaging done in firmware, so the effective acquisition rate is 1 kSps. In order to run an open-source TCP server on the board, a Cortex-M7-based 32-bit microcontroller ^[10] uses the Keil RTX RTOS ^[11]. On the PC side, a MATLAB ^[12] environment (version 9.5 R2018a) is set up to evaluate faults during welding. From the paper it is not clear, but it seems that the firmware is not divided into smaller tasks but uses just a main loop with interrupts.

To detect early failures in machinery equipment, a DAQ system for acquiring vibration signals in the X, Y, and Z axes in rotating machinery has been developed [13]. The goal of the paper was to create a real-time, multiparameter, high-precision, miniaturized, and portable DAQ system. The central part of the system's hardware is a Cortex-M3-based 32-bit microcontroller $\begin{bmatrix} 14 \\ 24 \end{bmatrix}$ with the μ C/OS-II RTOS $\begin{bmatrix} 15 \\ 25 \end{bmatrix}$ used to meet the multitasking requirements. To increase the portability of the system, an on-chip SRAM module is used for data storage, and an LCD touchscreen is used to display the acquisition results in real-time. A 12-bit ADC is used for sampling. The system can sample vibration signals from a selected axis at a time by sampling one of the three vibration acceleration transducers with a sampling frequency of 5 kSps. The firmware is divided into six tasks used for data acquisition, data transmission, data display, touch screen driver, key-press management, and µC/GUI interface ^[16] for the touchscreen LCD. Data acquisition starts when an acquisition request is received and sent by the user. Acquired data are moved using a DMA and stored in the SRAM until it is full, at which point an interrupt signal is sent to the CPU, which stops acquisition, reads the SRAM, and displays the result on the LCD display. It is debatable whether this system, which collects data only on the user's request and only until the storage is full, is really a real-time system. A software architecture diagram in the paper mentions a data transmission block that can communicate through USB, USART, and Ethernet, but it does not mention if, how, and when this block is used. Although increasing portability, storing data only locally and observation only using the LCD display greatly limits visualization and disables automated data analysis.

Xnode, a wireless high-fidelity DAQ system for structural health monitoring, has been presented in ^{[127][18]}, developed with technical standards in China and lessons learned from previously developed similar systems in mind. The hardware of Xnode consists of three printed circuit boards—a processor board, a radio/power board, and a sensor board. The hardware is designed around a Cortex-M4-based 32-bit microcontroller ^[19]. The system uses an external NAND flash memory for temporary and an SD card for permanent local storage but also features a wireless transceiver on the radio/power board for communication with a remote station, compatible with the ZigBee protocol ^[20]. The sensor board features accelerometers and temperature sensors and an 8-channel 24-bit ADC with a sampling rate of up to 16 kSps and programmable antialiasing filters. The software is developed using FreeRTOS and a middleware service-oriented framework that supports network and application scalability. The application, which utilizes the middleware services, is organized into three tasks—application task, sensing task, and a radio task. The papers do not specify any visualization software that is used with Xnode, the lack of which can complicate the usage for users that are not experienced in that field.

OpenDAQ, a data acquisition and control system, was designed to be a cost-effective, multifunction computerbased DAQ system open to users ^[21]. OpenDAQ has up to 8 analog input channels, a 16-bit resolution with a maximum sampling rate of 20 kSps for a single channel at a time, as well as one analog output. The system also has the option for digital sampling and control using the provided digital ports. The OpenDAQ hardware principally consists of an analog input signal conditioning block, analog output conditioning block, ADC, and an 8-bit AVR ATmega microcontroller ^[22]. Microcontroller software does not utilize an operating system, and it is written using libraries, such as Arduino libraries ^[23]. The system uses a USB connector for communication with a PC, which runs visualization and control software. The authors of OpenDAQ have created two options for the final user to choose from. The user can use either Python ^[24] or LabVIEW ^[8] for visualization and control. The wide range of capabilities and options the system offers makes it suitable for both industrial and educational applications. While OpenDAQ features an open design, it is possible that performances can be increased by the use of an RTOS. That way, the maximum acquisition rate could be increased to utilize the maximum 100 kSps sampling rate of the microcontroller's ADC.

An example of a DAQ system used in the automotive industry can be found in ^[25]. Like OpenDAQ, this system has been developed with cost-effectiveness in mind and utilizes low-cost components. The system is Arduino-based ^[23], and the hardware has been developed as an Arduino shield. One of the reasons for this design choice is that the system is designed to be easily programmable and customizable for different uses in-vehicle data acquisition and control. Arduino Due ^[26] is used as the development board because of the higher processing capabilities, the larger number of available inputs and outputs, and a higher resolution (12-bit) ADC when compared to other Arduino development boards. The result is a system that can acquire data from up to 12 analog inputs, an I2C sensor, an SPI sensor, and a GPS module along with multiple digital I/O pins. For applications where analog control is needed, there are two analog outputs. The maximum sampling frequency is 2 kSps if only one channel is used. Data can be stored locally using an SD card, or they can be stored remotely using Arduino's serial communication. The shield uses a Bluetooth module to enable acquisition from a mobile device and an LCD for displaying the DAQ status. No visualization software has been developed for the DAQ system. Although the system offers a user the possibility to connect and sample multiple analog and digital sensors at once, the

sampling rate drops significantly with every additional sensor connected, and it might be too low for some applications.

Papers ^{[22][28]} offer insight into another field where microcontroller-based DAQ systems find their use. In these papers, DAQ systems are used to acquire electrophysiological signals from live laboratory test animal subjects and send those results remotely. By using the UDP internet protocol, these systems achieve high data transfer speeds. In ^[22], 16 analog channels with a 16-bit resolution are sampled using an external ADC connected to an ESP32-based development board ^[29], which has a dual-core LX6 microcontroller ^[30] and WiFi communication capabilities. Utilizing FreeRTOS ^[31], the system divides tasks between the two processor cores to boost the performance of the system. This way, a total system throughput of 6.4 Mbps is achieved when acquiring 16 analog input channels simultaneously, with 25 kSps acquisition rate. The samples are wirelessly sent to the host using WiFi and, as mentioned previously, UDP transfer protocol. In ^[28], the authors explore the advantages of using UDP protocol over TCP protocol to reach high data transfer speeds when using Ethernet communication technology. In the experiment performed in this paper, the maximum system throughput achieved using TCP protocol was 17.8 Mbps. When the protocol was changed to UDP, the maximum system throughput reached 47 Mbps when acquiring data simultaneously from 64 channels, with an acquisition rate of 24 kSps. Even though both systems are tailored for a specific, in-house application, visualization of the data acquired could be improved, which would increase the applicability of both solutions.

From the examples above, it becomes clear that to meet the requirements of high-speed, high-resolution acquisition using a microcontroller-based system, an RTOS is almost always needed. Although OpenDAQ manages to reach the maximum acquisition rate of 20 kSps for 16-bit resolution without an RTOS, any major increase in acquisition rate would require it. As seen in many of the papers mentioned in this section, an RTOS enables the system to divide microcontroller firmware into multiple tasks that can control sensing and communication with the outside world separately and concurrently. In order to increase the system's applicability, it is beneficial to observe the collected data on a host system. For that purpose, it is necessary that the DAQ device has remote communication capabilities, and if high throughput is required, communication technology that supports high data transfer speeds is needed. With that in mind, it is advantageous to use high-speed protocols, for example, UDP, as shown in ^{[22][28]}.

4. Comparison of the Reviewed DAQ Systems

Table 1 gives an overview and comparison of the DAQ systems reviewed here. The systems are compared in multiple categories that are essential when describing and choosing a DAQ system. The categories are as follows:

Table 1. Overview and comparison of the reviewed DAQ systems.

Paper	Num. of Channels	Resolution [bit]	System Acquisition		. <i>1</i>	Type of		Comm.
			Throughput [bps]	Rate [Sps]	Visualization	Storage	Remote	Technology
[1]	1	16	152.6 M	10 M	No	Local SDRAM	No	/
[<u>2</u>]	4	12	915.6 M ¹	20 M	No	Server	No	PCle
[<u>5</u>]	10	12	921.6 k	500 k	Yes	Server	Yes	UART
[<u>9]</u>	2	16	31.25 k ²	1 k	Yes	Server	Yes	Ethernet
[<u>13]</u>	1	12	58.6 k ²	5 k	Yes, on small LCD	Local SRAM	No	/
[<u>17][18]</u>	8	24	1 M	16 k	No	Local and server	Yes	IEEE 802.15.4
[<u>21</u>]	8	16	312.5 k ²	20 k	Yes	Server	Yes	USB
[25]	1	12	23.5 k ²	2 k	No	Local and server	Yes	UART
[<u>27</u>]	16	16	6.4 M	25 k	Yes	Server	Yes	WiFi
[28]	64	32	47 M	24 k	No	Server	Yes	Ethernet
This research	1	16	51.2 M	3.2 M	Yes	Server	Yes	Ethernet

Paper	Num. of Channels	1 Resolution [bit]	System Throughput [bps]	Acquisition Rate [Sps]	2 Visualization	Type of Storage	Comm. Remote Technology	maximum
			,	,				

- System throughput—as defined in Section 1, without overhead,
- Acquisition rate—as defined in Section 1,
- Visualization—whether the authors of the system have provided their own method of data visualization and/or processing,
- Type of storage—The type or types of permanent sampled data storage used by the system,
- Remote—whether the system can be used for remote data acquisition,
- Communication technology (for systems where acquisition and final storage are not situated within the same device).

For a better comparison between the reviewed papers and the results achieved with the prototype presented as part of this entry, the characteristics of the prototype are added at the end of the table.

It can be observed in **Table 1** that the reviewed DAQ systems have very different characteristics across the compared categories. To sum up, work ^[2] has the greatest system throughput of 915.6 Mbps due to the utilization of an FPGA and fast PCIe communication interface. On the other hand, it is not a remote DAQ system since the DAQ device is connected to the host communication bus directly. Moreover, the work does not include real-time visualization or analysis but offers logging only. The work ^[28] represents the remote DAQ system with the greatest system throughput of 47 Mbps, but it offers only data logging on the host. Among the works that include visualization, the one with the greatest system throughput of 6.4 Mbps is ^[27]. The system is a remote one, based on WiFi technology, but the system throughput is not very high compared to ^{[2][28]}. Obviously, none of the "best" solutions accomplish the requirements for as high as possible system throughput, remote operation, and at least host visualization at the same time.

Generally, FPGA-based DAQ systems have a higher system throughput and acquisition rate than microcontrollerbased DAQ systems. As previously mentioned, microcontroller-based DAQ systems are more easily assembled due to a number of factors. Thus, they show greater flexibility in terms of visualization, storage, and connectivity. By using the methods described later in this entry, the researchers managed to achieve a respectable system throughput within a remote DAQ system with visualization and logging at the host computer, as can be observed in the last row of **Table 1**.

References

- 1. Khedkar, A.A.; Khade, R.H. High Speed FPGA-Based Data Acquisition System. Microprocess. Microsyst. 2017, 49, 87–94.
- 2. Shu, S.; Wang, L.; Liu, D.; Chen, M.; Zhang, Y.; Luo, J.; Ji, F. A High-Speed Data Acquisition System Based on FPGA for Tokamak. Rev. Sci. Instrum. 2018, 89, 10K120.
- 3. Wan, B. Experimental Advanced Superconducting Tokamak. In Magnetic Fusion Energy; Woodhead Publishing: Sawston, UK, 2016; pp. 409–437. ISBN 978-0-08-100315-2.
- 4. Oberhumer, M.F.X.J. LZO Real-Time Data Compression Library. Available online: http://www.oberhumer.com/opensource/lzo/ (accessed on 15 August 2023).
- Sahoo, S.; Chaddha, N.; Sahoo, S.; Biswas, N.; Roy, A.; Das, M.; Pal, S. FPGA-Based Multi-Channel Data Acquisition System for Superheated Emulsion Detectors. Nucl. Instrum. Methods Phys. Res. Sect. A Accel. Spectrom. Detect. Assoc. Equip. 2021, 1009, 165457.
- 6. Apfel, R.E. The Superheated Drop Detector. Nucl. Instrum. Methods 1979, 162, 603–608.
- Dawoud, D.S.; Dawoud, P. Universal Asynchronous Receiver/Transmitter (UART). In Serial Communication Protocols and Standards; River Publishers: Aalborg, Denmark, 2022; pp. 47–92. ISBN 978-1-00-079691-9.
- 8. National Instruments LabVIEW. Available online: https://www.ni.com/en-rs/shop/labview.html (accessed on 15 August 2023).
- 9. Podešva, P.; Gebauer, J.; Fojtik, D.; Mahdal, M. High Speed Current Sensing System for Welding. IFAC-PapersOnLine 2019, 52, 139–144.
- Arm Limited. Arm Cortex-M7 Processor Datasheet; Arm Limited: Cambridge, UK, 2020; Available online: https://www.arm.com/-/media/Arm%20Developer%20Community/PDF/Processor%20Datasheets/Arm-Cortex-M7-Processor-Datasheet.pdf (accessed on 15 August 2023).
- 11. RTX Real-Time Operating System. Available online: https://www.keil.com/arm/rl-arm/kernel.asp (accessed on 15 August 2023).
- 12. Mathworks Matlab Product Webpage. Available online: https://www.mathworks.com/products/matlab.html (accessed on 15 August 2023).

- 13. Zhang, H.; Kang, W. Design of the Data Acquisition System Based on STM32. Procedia Comput. Sci. 2013, 17, 222–228.
- Arm Limited. Arm Cortex-M3 Processor Datasheet; Arm Limited: Cambridge, UK, 2020; Available online: https://documentation-service.arm.com/static/62053c120ca305732a3a5c14?token= (accessed on 15 August 2023).
- 15. UC/OS-II Documentation. Available online: https://micrium.atlassian.net/wiki/spaces/osiidoc/overview (accessed on 15 August 2023).
- Micrium, Inc. MC/GUI Embedded Graphical User Interface; Micrium, Inc.: Weston, FL, USA, 2022; Available online: https://www.mouser.com/datasheet/2/266/gui_dsheet-1326.pdf (accessed on 15 August 2023).
- 17. Spencer, B.F.; Park, J.W.; Mechitov, K.A.; Jo, H.; Agha, G. Next Generation Wireless Smart Sensors Toward Sustainable Civil Infrastructure. Procedia Eng. 2017, 171, 5–13.
- Fu, Y.; Mechitov, K.A.; Hoang, T.; Kim, J.; Lee, D.H.; Spencer, B.F. Development and Full-Scale Validation of High-Fidelity Data Acquisition on a next-Generation Wireless Smart Sensor Platform. Adv. Struct. Eng. 2019, 22, 3512–3533.
- Arm Limited. Arm Cortex-M4 Processor Datasheet; Arm Limited: Cambridge, UK, 2020; Available online: https://documentation-service.arm.com/static/62053f0a0ca305732a3a5c17?token= (accessed on 15 August 2023).
- 20. Farahani, S. ZigBee Basics. In ZigBee Wireless Networks and Transceivers; Newnes: Oxford, UK, 2008; pp. 1–24.
- Ferrero, M.F.J.; Valledor, L.M.; Campo, R.J.C.; Blanco, G.J.R.; Menéndez, B.J. Low-Cost Open-Source Multifunction Data Acquisition System for Accurate Measurements. Measurement 2014, 55, 265–271.
- 22. 8-Bit AVR® MCUs Website. Available online: https://www.microchip.com/enus/products/microcontrollers-and-microprocessors/8-bit-mcus/avr-mcus (accessed on 15 August 2023).
- 23. Arduino Website Home Page. Available online: https://www.arduino.cc/ (accessed on 15 August 2023).
- 24. Python Programming Language Official Website. Available online: https://www.python.org/ (accessed on 15 August 2023).
- 25. González, A.; Olazagoitia, J.L.; Vinolas, J. A Low-Cost Data Acquisition System for Automobile Dynamics Applications. Sensors 2018, 18, 366.
- 26. Arduino Due Documentation. Available online: https://docs.arduino.cc/hardware/due (accessed on 15 August 2023).

- 27. Mukati, M.R.; Kocatürk, S.; Kocatürk, M.; Baykaş, T. A Microcontroller-Based Wireless Multichannel Neural Data Transmis-sion System. In Proceedings of the 21st National Biomedical Engineering Meeting (BIYOMUT), Istanbul, Turkey, 24 November–26 December 2017; pp. 1–4.
- Tibor, S.; Dukán, P.; Odadžíc, B.; Péter, O. Realization of Reliable High-Speed Data Transfer over UDP with Continuous Storage. In Proceedings of the 11th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 18–20 November 2010; pp. 307–310.
- 29. Espressif Systems ESP32 Series Datasheet; Espressif Systems: Shanghai, People's Republic of China, 2023. Available online: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (accessed on 15 August 2023).
- Cadence Design Systems. Xtensa LX6 Customizable DPU; Cadence Design Systems: San Jose, CA, USA, 2014; Available online: https://mirrobo.ru/wpcontent/uploads/2016/11/Cadence_Tensillica_Xtensa_LX6_ds.pdf (accessed on 15 August 2023).
- 31. FreeRTOS Real-Time Operating System for Microcontrollers. Available online: https://freertos.org/ (accessed on 15 August 2023).

Retrieved from https://encyclopedia.pub/entry/history/show/114395