

Low cost open data acquisition system for biomedical applications

Wojciech M. Zabolotny^a and Przemyslaw Laniewski-Wollk^b and Wojciech Zaworski^a

^aInstitute of Electronic Systems, Nowowiejska 15/19, 00-665 Warszawa, Poland;

^bCentrum Zdrowia Dziecka, Al. Dzieci Polskich 20, 04-730 Warszawa, Poland

ABSTRACT

In the biomedical applications it is often necessary to collect measurement data from different devices. It is relatively easy, if the devices are equipped with a MIB or Ethernet interface, however often they feature only the asynchronous serial link, and sometimes the measured values are available only as the analog signals. The system presented in the paper is a low cost alternative to commercially available data acquisition systems. The hardware and software architecture of the system is fully open, so it is possible to customize it for particular needs. The presented system offers various possibilities to connect it to the computer based data processing unit - e.g. using the USB or Ethernet ports. Both interfaces allow also to use many such systems in parallel to increase amount of serial and analog inputs. The open source software used in the system makes possible to process the acquired data with standard tools like MATLAB,¹ Scilab² or Octave^{3,4} or with a dedicated, user supplied application.

Keywords: Data acquisition system, open source solutions, open hardware, biomedical signal processing, RS232, analog signals, biomedical monitoring

1. INTRODUCTION

Acquisition of biomedical data is important subject in biomedical research, but also in diagnosis and therapy. The IEEE 1073 standard⁵ has been proposed and accepted a few years ago, and theoretically data acquisition from devices interconnected into the MIB network should be an easy task. However in biomedical research there are still situations where we need to acquire biomedical data from devices equipped only with RS232 serial ports and/or analog outputs. Theoretically a PC computer with multiport serial adapter and ADC data acquisition card could be used to acquire such data, however such solution is neither cheap, nor convenient.

The standard PC computer is big and noisy, and with separate components like keyboard, mouse and monitor connected with numerous cables just doesn't fit in medical environment. The portable computer is much more convenient, but again supplementing it with multiple serial ports and ADC card may be difficult. There are multiport RS232 USB adapters^{6,7} and USB connected A/D converters^{8,9} commercially available, but such system consisting of many independent devices is still big, expensive and inconvenient.

Our goal was to create a relatively small and cheap device, collecting analog and RS232 data, able to connect to the data analyzing computer or to the computer network through the single cable with reasonable length limit, or even via a wireless link.

2. TECHNICAL REQUIREMENTS AND HARDWARE SOLUTION

Acquisition of biomedical signals typically requires resolution of 12 bits. The required sampling frequency depends on the measured signal, but generally values below 1 kHz are acceptable. The required amount of analog signals depends on application, however 8 channels is usually sufficient.

There are many currently available ADC equipped microcontrollers offering 8 and more analog inputs with 12-bit resolution, and maximum total sampling rates above 100 ksps (kilosamples per second). Therefore meeting the above requirements is relatively easy, and even higher than required sampling rates may be achieved. However such ADC, when

Further author information: (Send correspondence to W.M.Zabolotny)
W.M.Zabolotny: E-mail: wzab@ise.pw.edu.pl, Telephone: +48 22 6607717

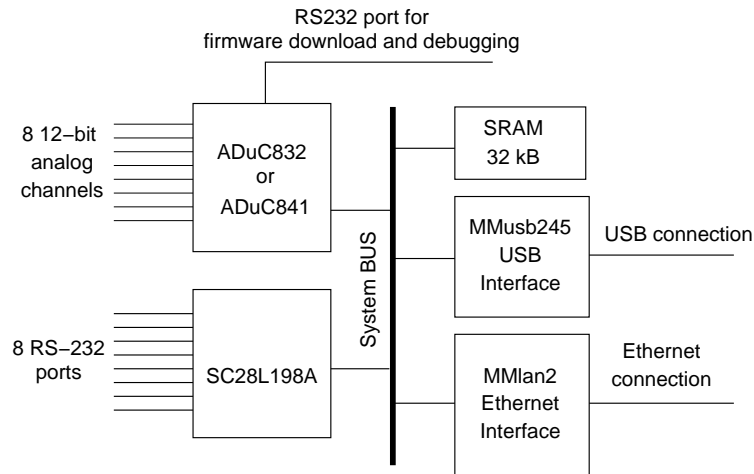


Figure 1. Block diagram of the PL-DAQ acquisition system.

fully utilized, generates the data stream of rate above 1 Mb/s, so a connection to the data analyzing computer or network must offer sufficient throughput to transmit these data.

The most popular external extension interface available in modern computers is USB. It's raw throughput is equal to 12 Mb/s (full-speed) or 480 Mb/s (high-speed). So even in the full-speed version, and even considering the protocol overhead, the bandwidth should be sufficient to transfer all the data. The USB cable length is limited to 5 m (it can be extended to 15 m by using 2 intermediate hubs). If the distance between the data acquisition system and the computer must be longer, or if network connectivity is required - the Ethernet interface must be used.

The system should be able to service up to 8 RS232 capable devices. It should be able to work with standard RS232 baud rates from 1200 Baud to 57600 Baud, however other higher and non-standard rates would be also appreciated.

2.1. Selection of components

The ideal solution would be a single microcontroller featuring all - 12-bit ADC, USB interface and Ethernet interface. Unfortunately, even though there are Ethernet capable microcontrollers¹⁰ and the full-speed USB capable microcontrollers with 10-bit ADC,¹¹ it was impossible to find a microcontroller combining 12-bit ADC with USB and Ethernet interface. Therefore it was necessary to develop a solution where the USB and Ethernet connectivity is provided by external chips, connected via system bus to the microcontroller.

The external system bus is necessary also to connect the multiport RS232 interface, which must be implemented as an external chip as well. The SC28L198 chip¹² was chosen as 8-port asynchronous interface, and MAX232 compatible chips as RS232 level converters.

Finally the ADuC832 microcontroller¹³ was chosen as the microcontroller for the system. It provides 8-channel 12-bit ADC with 247 ksps total sampling rate, and with internal reference. It is a FLASH based microcontroller, so it is possible to modify its firmware, and it can execute program from external SRAM memory, which makes debugging of firmware easier. In the prototype version the USB and Ethernet interfaces have been implemented using ready to use mezzanine boards. The USB connection is assured by the MMusb245 module,¹⁴ while Ethernet connectivity by the MMLan2 module¹⁵ based on the RTL8019AS chip.¹⁶

Block diagram of the final system is shown in the Fig. 1. The assembled prototype is shown in the Fig. 2. The obtained technical specifications regarding the data transfer are summarized below:

- Data acquisition interfaces
 - 8 analog inputs (0-2.5V, 12-bit, 247 ksps total)
 - 8 RS232 interfaces (up to 115.2 or 230.4 kbaud, depending on the speed of RS232 level converter)

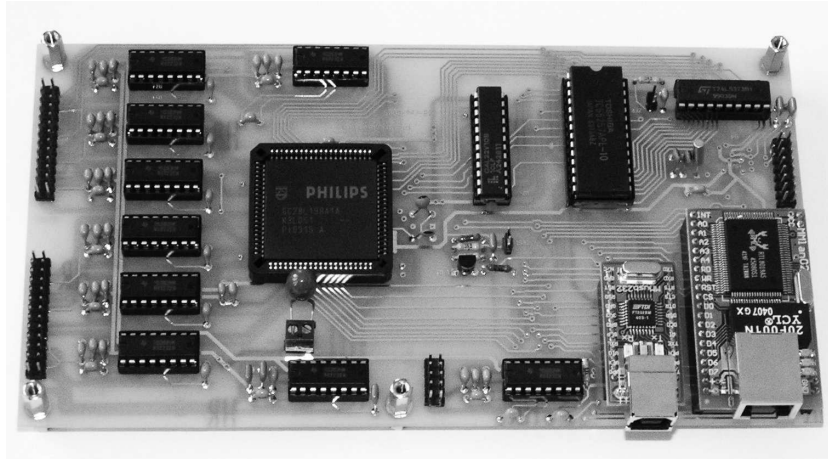


Figure 2. Assembled prototype of the PL-DAQ acquisition system.

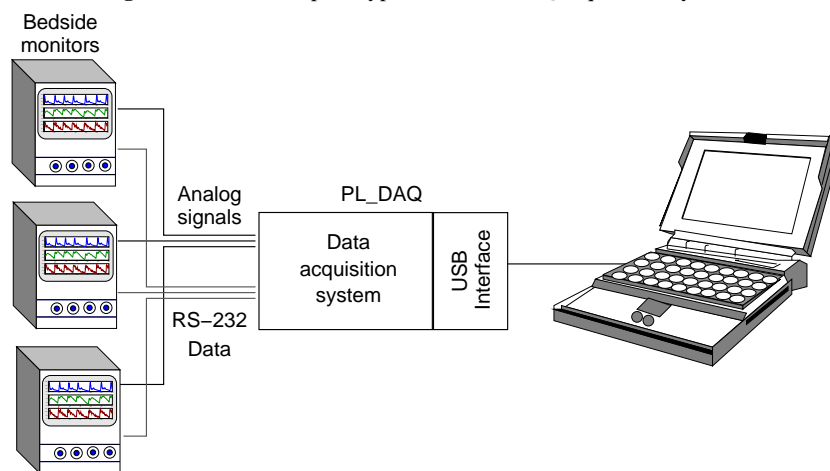


Figure 3. PL-DAQ connected via USB interface

- Upstream connections

- USB 1.1 and 2.0 compatible, effective full-speed transfer rate up to 1MB/s
- Ethernet 10Base-T (10 MB/s raw throughput, limited by protocol and software overhead)

2.2. Possible uses of the system

The developed hardware may be used in different configurations. It is possible to connect it via the USB cable to the portable computer sufficiently distant from the patient's bed, as shown in the Fig. 3.

If the distance between the patient and the analyzing computer must be longer, or if the computer may not be located in the same room as patient, or a few computers must access simultaneously the data provided by the system, it is possible to connect the system to the Ethernet network, as shown in the Fig. 4.

Sometimes the network socket may not be available in the patient's room. In this case it is still possible to use the system, connected to simple wireless bridge (or access point working in the client mode), as shown in the Fig. 5.

In all configurations the user must be aware that the PL-DAQ system relies on the insulation barrier built into the biomedical devices connected to the analog inputs and RS232 interfaces. No additional biological barrier between the

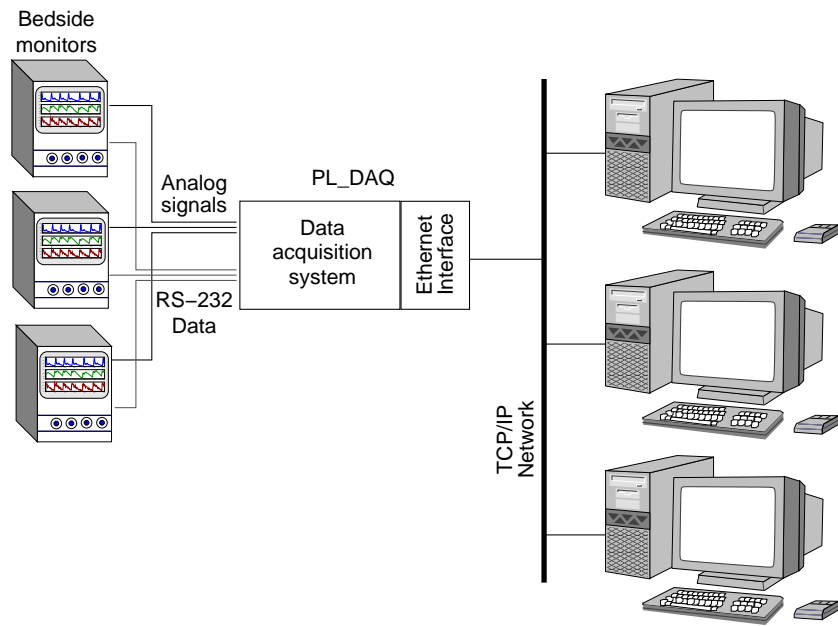


Figure 4. PL-DAQ connected via standard (wired) Ethernet

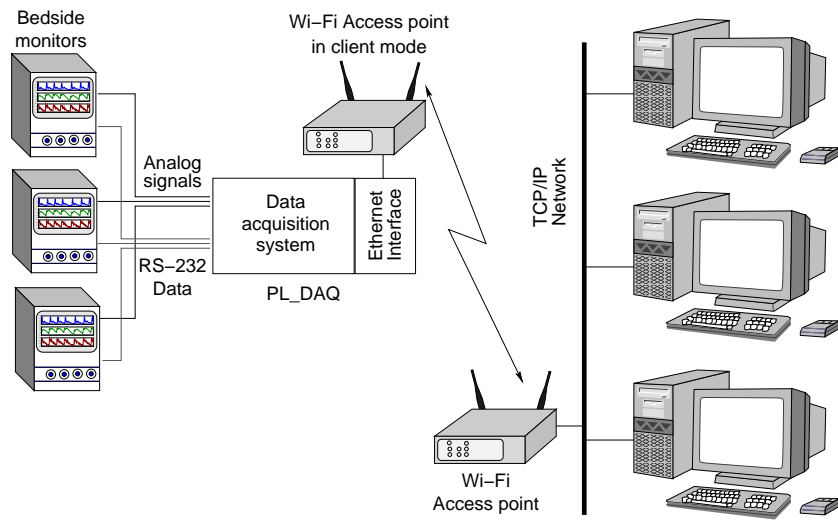


Figure 5. PL-DAQ connected via wireless Ethernet

connected devices and between the system and data analyzing computer or network is provided. The configuration shown in the Fig. 5 may be used to provide insulation barrier between the system and the network, when both PL_Daq and access point are powered from the insulated, secure power supply.

3. SYSTEM SOFTWARE

The software part of the system is still being developed. The system software consists of two main components. First of them is the firmware programmed into the ADuC832 microcontroller, and the second one is the communication software running on the data analyzing computer.

3.1. Firmware

The firmware is being developed in C language, using the free SDCC¹⁷ compiler.

The firmware implements automatic sampling of the analog channels with programmable sampling rates. The order of channels may be programmed, and sampled channels may be arbitrarily chosen. The sampled data are encapsulated in records with time markers, and passed to the communication interface.

Because of relatively low computational power of the ADuC832 microcontroller, when communicating with RS232 devices, the system works as a transparent bridge between the device and the data analyzing computer. Data received from different ports are encapsulated in packets together with the port number, port status, data length and time marker and sent to the computer. Information about special events (like change of state of the CTS line, break condition or frame error) is additionally encoded into the data stream using special character sequences. The data to be sent to different RS232 ports are also transmitted from the data analyzing computer as similar packets, containing the port number, data length and the data themselves.

The interpretation of the RS232 data stream according to the protocol used by the particular device will be performed by the data analyzing computer. However it is possible to implement a special communication mode in the firmware, if communication protocol used by the particular device requires some real time handshake.

Because resources offered by the ADuC832 chip are not sufficient to implement the full TCP/IP stack, therefore a simplified, UDP based protocol is being implemented for sending the data over the Ethernet network.

3.2. Communication software

The communication software is being written for the Linux OS, using the Python language, with some C extensions. The Twisted¹⁸ library has been used to implement the network protocols. The Numerical Python¹⁹ library and its extensions has been used for efficient handling of numerical data.

Such solution allows for fast prototyping and convenient debugging. It is also possible to port easily the communication software to other platforms running Python interpreter, e.g. to MS Windows. The communication software uses the separate modules for decoding of data transmitted from the system. One module is responsible for decoding of the analog signals' samples. Additional modules must be provided to implement RS232 protocols used by particular devices.

The decoded data may be either processed by the user application, written in Python and integrated with the communication software, or stored into a file for further processing with signal processing packages available in MATLAB,¹ Scilab² or Octave^{3,4}

A dedicated Python library for biomedical signals processing in Python is being developed, as a BSc thesis, by one of our students. This library will support creating of user applications integrated with the system.

3.3. Future enhancements

Sending of biomedical data over the widely accessible Ethernet network may cause concerns regarding the confidentiality and integrity of the data. Therefore a cryptographic data protection would be needed for secure transmission of data. However encrypting of data with ADuC832 would significantly affect performance of the system because of low computational power of this microcontroller.

In the future versions of the system the ADuC841 or ADuC842 microcontrollers, with significantly increased speed, may be used to implement cryptographic algorithms.

This cryptographic extension would be particularly important when using the system with the wireless link, as shown in the Fig. 5.

4. OPEN ARCHITECTURE

The system is supposed to be fully open, to create a basis for open source and hardware data acquisition and processing system.

The free and open source solutions, including EDA tools, programming tools and libraries, have been used for the development of the system.

Both hardware and software documentation and source codes will be released under the open license as soon, as the system is mature enough. Currently the hardware documentation may be found on the project's site.²⁰

5. CONCLUSIONS

The cheap and simple data acquisition system has been developed for collecting of biomedical data from devices equipped with analog outputs and RS232 interfaces.

The system allows for data acquisition in conditions, where it is impossible to locate a standard PC based data acquisition system near to the patient.

The system may be connected to the data recording or analyzing computer using the USB cable. It is also possible to connect the system to the TCP/IP network using the Ethernet twisted pair cable, or even using the wireless link.

The architecture of the system (both hardware and software) is open, so it is possible to customize it to particular needs and extend it.

ACKNOWLEDGMENTS

In the development of the prototype system, free samples, offered by Analog Devices and Philips, have been used.

REFERENCES

1. "The MathWorks - MATLAB and Simulink for Technical Computing." <<http://www.mathworks.com>>.
2. "Scilab a Free Scientific Software Package." <<http://scilabsoft.inria.fr>>.
3. "GNU Octave Home Page." <<http://www.octave.org>>.
4. "GNU Octave Repository." <<http://octave.sourceforge.net/>>.
5. "IEEE 1073 Medical Device Communications." <<http://www.ieee1073.org/index.html>>.
6. "USB to RS-232 Multiport serial converter." <<http://www.alloy.com.au/products.umc102.htm>>.
7. "2, 4, 8, & 16 Port Serial/RS232 to USB Adapters." <<http://www.wcscnet.com/HdwPX232USB.htm>>.
8. "Multiple Channel Data Acquisition ADC-11/10 and ADC-11/12." <<http://www.picotech.com/data-acquisition.html>>.
9. "DT-9834 Series USB Data Acquisition Units." <http://www.agile.co.za/dt/daq_usb/dt-9834.htm>.
10. "TINI Networked Microcontrollers." <<http://www.maxim-ic.com/products/microcontrollers/tini/>>.
11. "PIC18F2455/2550/4455/4550 Data Sheet." <<http://ww1.microchip.com/downloads/en/DeviceDoc/39632b.pdf>>.
12. "SC28L198 Data Sheet." <http://www.semiconductors.philips.com/acrobat/datasheets/SC28L198_5.pdf>.
13. "ADuC832 Data Sheet." <http://www.analog.com/UploadedFiles/Data_Sheets/466325469ADuC832_0.pdf>.
14. "MMusb245 technical information (in Polish)." <http://www.propox.com/products/t_94.html>.
15. "MMlan2 technical information (in Polish)." <http://www.propox.com/products/t_91.html>.
16. "MMlan2 technical information (in Polish)." <<ftp://152.104.238.194/cn/nic/rtl8019as/spec-8019as.zip>>.
17. "SDCC - Small Device C Compiler." <<http://sdcc.sourceforge.net>>.
18. "Twisted Home Page." <<http://twistedmatrix.com/projects/twisted>>.
19. "Numerical Python." <<http://numeric.scipy.org>>.
20. "PL_Daq - Analog and RS-232 Data Acquisition System." <http://www.ise.pw.edu.pl/~wzab/pl_daq>.