Controlling CAMAC Instrumentation through the USB Port

R.V. Ribas

Laboratório Aberto de Física Nuclear.
Universidade de São Paulo, Instituto de Física – CP 66318 São Paulo, Brazil

Abstract. A programmable device to interface CAMAC instrumentation to the USB port of computers, without the need of heavy, noisy and expensive CAMAC crates is described in this article. Up to four single-width modules can be used. Also, all software necessary for a multi-parametric data acquisition system was developed. A standard crate-controller based on the same project is being designed.

Keywords: Data Acquisition. CAMAC instrumentation. USB port.
PACS: 29.50.+v; 29.85.Ca

INTRODUCTION

In many laboratories, CAMAC instrumentation such as ADCs, TDCs, scalers, etc., even if still very useful for data acquisition in low energy nuclear physics, are no longer in use. This is because most of the common interfaces of CAMAC crate-controllers to micro computers are not available in modern computers (such as ISA slots, GPIB boards, etc.). Also, when still in use, even in applications where only a few modules are necessary, a noisy and heavy CAMAC crate must be employed. In order to have a portable and simple interface to personal computers, we developed two interface boards that can accommodate a small number of CAMAC modules connected to a standard USB port (both 1.1 and 2.0). These interfaces include a microcontroller to handle the trigger signals, the related CAMAC operations, event formatting, buffering and data transfer to the computer. The first version of the interface is limited to two specific modules available in our Laboratory (an ORTEC AD811 8x2048 channels ADC and a LeCroy 2228A 8x2048 channels TDC). A second version, capable of handling up to four single-width CAMAC modules was then developed. Typically 4 μs is spent to process each CAMAC read operation and about half this time to dataless operations. A standard CAMAC crate-controller based on these boards is now under development.

THE HARDWARE

The FX2 Interface

The main goal of this project was to develop a small hardware board capable of interfacing a few (up to 4) CAMAC modules to a USB port. The interface should be programmable in order to handle the CAMAC modules, gathering of the data and formatting the events, besides buffering several events before sending them to the computer through the USB port. Also, this dedicated interface avoids the use of expensive, heavy, and noisy CAMAC crates, when using only a small number of modules. For this, the EZ-USB FX2LP device from *Cypress*, that embodies a 8051 microcontroller and a USB interface, has proven to be perfectly adequate. For this device, there exists free software available on Linux to load a program, perform all necessary IO, control, etc., such as the cycfx2prog [1]. The 8051 microcontroller is also c-programmable using the sdcc compiler, that is available in most Linux distributions. Besides all USB 2.0 specifications (Full and High speed), endpoint FIFO buffers, the FX2 microcontroller includes 5 IO ports (in the 100 or 128 pin versions), that can be bit-assigned as input or output and 16 kB of RAM memory for both program code and data storage. Maximum clock rate is 48 MHz, with four clock per instruction cycle. Figure 1 shows the logic block diagram [2].

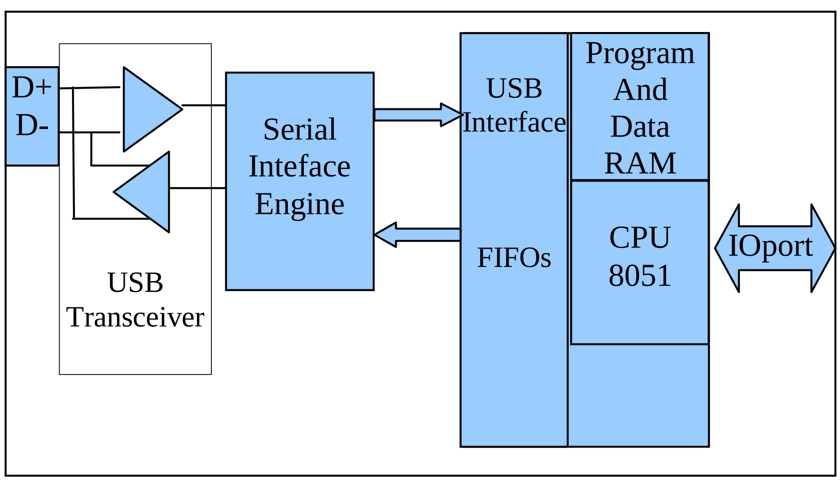

FIGURE 1. Block diagram showing the USB interface, FIFOs and the microcontroler in the Cypress EZ-USB FX2 chip.

There are several evaluating boards for the EZ-USB FX2 in the market. The FX2-128-TB from *Siphec* [3], which provides all pins of the device in two connectors, was employed. First, a test interface was built, for up to two CAMAC modules, that were, up to now, unused in our laboratory, an ORTEC 811 ADC (8x2K) and a LeCroy 2228 TDC (8x2K). Since the results were satisfactory, a more complex board was constructed, capable of accommodating up to four modules. Even though not all CAMAC capabilities are available in both interfaces, the second one can accommodate most of the modules in use at our lab. A standard CAMAC crate-controller, fully functional, is now being designed.

THE SOFTWARE

The Front-end Software

The front-end software consists of the program to be loaded in the FX2-8051 microcontroller and the communication program running in the Linux environment. In the FX2 program, a library of functions for communication with the computer (start, stop, init acquisition) and accessing the CAMAC modules (NAF, Z, etc.) is linked to two functions that should be written by the user. One, the init_camac(), that is executed at every start of acquisition and the second, the event(), that runs at each occurrence of an event. Events are detected using an OR of all LAM signals programmed in the modules. Each event is put in the event data buffer and then transferred to the FIFO buffer (mostly during the next event converting time, in high rate situation). After a few events are gathered in this FIFO, a flag is raised to request the transfer of this data buffer to the computer. Since this will occur at slow rate, interrupt in not necessary in the USB transfer protocol. For the present application, the structure of the functions available to the users was defined in order to make the event() program as similar as possible to the event description used in the standard data acquisition system at our laboratory [4]. This could be easily modified to another particular situation.

Acquisition and On-line Processing Software

The remaining acquisition and monitoring software, the spm-fx2, consists of three modules running concurrently in the computer. The standard interprocess communication (semaphores, shared memories, etc) are used to exchange data and command to the processes. The first module is just an interface between user and the other two modules. The second one gets data buffer from the front-end, sends to the sorting module and also group them in larger buffers in order to store the data on disk. All these modules have the same functionality as the present data acquisition software in use at our laboratory. The on-line histograms are stored into a 64 MB shared memory block that can be accessed by the display and analysis program. Any user logged into the acquisition account can run the display program and have access to the on-line histograms. The more recent data acquisition software available in our lab, the spmroot [5], mostly based on the CERN *root* system could also easily be fit to work with this new CAMAC front-end device.

Timing and Performance

The timing characteristics presented here were measured with the simpler interface, for two specific modules. These numbers depends also on the software running in the 8051 microcontroller and can be improved. During idle time, the FX2 program is looping, waiting for a LAM signal. Event servicing starts about 10 μs after the LAM. A dedicated NAF function is then called to read data from the module and takes about 4.5 μs to read 16 bit data from a module and copy the data from the 8051 port to the event buffer. After the event processing is finished and the modules are cleared and ready for the next conversions, the event buffer data are copied to the FIFO buffer. Since this buffer is external to the 8051, the copy is much slower than that to the event buffer, located in the RAM area of the microcontroller. When this copy is in progress the system is already waiting for or converting new data, there is a time overlap between them and specially for slower converter, like the 811 ADC (80 μs), the coping time does not affect the dead time of the system. Data transfer from the FIFO buffer to the computer is processed at USB high-speed rate.

CONCLUSIONS

CAMAC instrumentation has a much longer lifetime than present computer systems and related interfaces. A versatile, programmable CAMAC-USB interface was developed, allowing the use of up to four CAMAC module in a small data acquisition system. All related software for a data acquisition system based on this new front-end was also developed, keeping all functionality of the standard data acquisition system presently in use at our laboratory. A crate-controller based on the present project is under development.

REFERENCES

1. W. Wieser – http://www.triplespark.net/elec/periph/USB-FX2/software/ available in most Linux distributions.
2. Cypress - http://www.cypress.com/?id=193
3. Siphec – http://www.siphec.com/
4. R.V. Ribas – Nucl. Instr. Meth. A 483, p. 830 (2002).
5. A.A.P. Suaide - http://www.dfn.if.usp.br/~suaide/pelletron/spmroot.html