

# Introduction to the ATLAS MiniDAQ

**R. Ball**  
**University of Michigan**  
**7/26/99**

The ATLAS MiniDAQ program in development at the University of Michigan is a LabWindows/CVI-based program set. It is designed as a general purpose, test bench, GUI wrapper for data acquisition, primarily in a VME hardware environment. Extensions for CAMAC hardware are also under development.

The software is under development on a Windows NT 4.0 Pentium II computer. Access to VME is accomplished using the VME-PCI8000 kit from National Instruments. This kit includes the PCI-MXI-2 and VME-MXI-2 modules for connecting the computer PCI bus to the VME bus, and NI-VXI/VISA software for driving the interface. At the lowest call level primitives from this kit are used in accessing the hardware.

Our test system implements a 9U VME-64 crate with 9U to 6U adapter cards for smaller modules. The VME-MXI-2 is in a 6U form factor as is the CSM0 VME module which we are building. Modules with a 9U form factor in use at Michigan are the CDF TDC board and CDF 48 channel pulser card.

The standard GUI advocated by National Instruments for LabWindows/CVI utilizes event interrupts associated with buttons added to the interface. In this CVI-Callback scheme the only priority assigned is the order in which buttons are “clicked”. We chose an alternate method which gave us better control over the order in which events were processed. This alternate method involves an “Event loop” in which data acquisition hardware is given top priority, followed by the buttons on the GUI panels. Finally the counters added on the GUI panel are updated before the loop begins again. For purposes of development a user can build their initial GUI using the standard callback mechanism, then adapt the resultant C-language code to the event loop scheme. This process is described in Reference 1.

The core set of code distributed with the MiniDAQ has no dependency on the external hardware. Instead the user must fill skeleton routines with the code necessary to check for a hardware trigger in a polling fashion, and then read out the data acquired in hardware to an event buffer. This is the code portion which is hardware dependent. We have written drivers for communication with various hardware, including the CSM0 VME module, a VME2510 parallel I/O register, CAMAC crates attached via the Kinetic Systems 2917 VME module, and a CDF pulser card. The underlying assumption in these modules is that the hardware and software listed above for VME access were used, but all such primitives were isolated at the lowest level of the software call tree, simplifying its adaptation to other VME hardware should that be desirable.

The subroutines described in Reference 1 include *setup\_user\_hook*. This routine allows the user to write a routine to analyze an acquired event buffer during the course of

the “Event loop” process, calling the routine after other event processing is completed. However, for more complex analysis than simple content checking, it would be best to do the work separately. To assist in this it is possible to ship the data to a second process for distribution elsewhere via tcp/ip. For example, another PC or a Unix machine can receive event samples and analyze them at a more leisurely rate. This process is described in Reference 2. Of course, event data recorded in a disk file is also available for “off line” analysis.

Development work is in progress. We are investigating the best way to package the MiniDAQ for use elsewhere. We are also developing driver software for the CSM-0 VME module. This module contains a TTC emulator, a JTAG controller, and version zero of the Chamber Service Module hardware for readout of the MDT chamber TDC chips. Currently only the JTAG driver<sup>3</sup> has been prepared, but with the arrival of the VME module in mid-August work on all portions of this software will begin in earnest.

All documentation for the ATLAS MiniDAQ is maintained on the Michigan Docushare<sup>4,5</sup> server at the University of Michigan.

## References

1. ATLAS MiniDAQ Interface Library Software, B. Ball, J. Kuah, J. Hollar, E. Munson, University of Michigan, July, 1999.
2. DDE Client/TCP Server System Software, J. Kuah and J. Hollar, University of Michigan, July, 1999.
3. SCANPSC100F JTAG Controller Software, J. Hollar, University of Michigan, July, 1999.
4. Docushare is document sharing software from Xerox Corporation. For more information see their web page at <http://docushare.xerox.com:8001>
5. The University of Michigan ATLAS electronics Docushare pages can be found on the web at <http://hepbdc.physics.lsa.umich.edu/docushare/default.htm> and then follow the ATLAS Electronics link, to ATLAS Trigger Software.