Proceedings of ICRC 2001: 825 © Copernicus Gesellschaft 2001

ICRC 2001

Absolute GPS time event generation and capture for remote locations

J. D. Smith, S. B. Thomas, and L. R. Wiencke for the HiRes Collaboration

Dept. of Physics, University of Utah

Abstract. The HiRes experiment operates fixed location and portable lasers at remote desert locations to generate calibration events. One physics goal of HiRes is to search for unusual showers. These may appear similar to upward or horizontally pointing laser tracks used for atmospheric calibration. It is therefore necessary to remove all of these calibration events from the HiRes detector data stream in a physics blind manner. A robust and convenient "tagging" method is to generate the calibration events at precisely known times. To facilitate this tagging method we have developed the GPSY (Global Positioning System - YAG) module. It uses a GPS receiver, an embedded processor and additional timing logic to generate laser triggers at arbitrary programmed times and frequencies with better than 100nS accuracy. The GPSY module has two trigger outputs (one microsecond resolution) to trigger the laser flash-lamp and Q-switch and one event capture input (25nS resolution). The GPSY module can be programmed either by a front panel menu based interface or by a host computer via an RS232 serial interface. The latter also allows for computer logging of generated and captured event times. Details of the design and the implementation of these devices will be presented.

1 Motivation

Air Showers represent a small fraction, much less than a percent, of the total High Resolution Fly's Eye data sample. The bulk of the sample is calibration data. Most of this calibration data is generated by two types of systems that use lasers. One type sends light directly to the detectors via optical fibers to monitor detector gains (Girard 2001). The other sends a beam of light into the sky and the scattered light that reaches the detectors is used to monitor atmospheric effects (Wiencke 1998). It is important that these calibration events be cleanly separated from the rest of the sample both to provide a complete set of monitoring information, and more importantly to avoid contaminating events samples used for physics measurements.

Correspondence to: J. Smith (jeremy@cosmic.utah.edu)

One approach is to identify some set of pattern based criteria that distinguishes calibration events from other events. For example, light from an diffused optical fiber in front of a detector illuminates all photo-multiplier tubes (PMT's) at essentially the same time. Furthermore if the entire detector is illuminated in this manner it is probably safe to assume that such events are calibration events. This approach does not work in other cases. For example, horizontal laser tracks are quite useful for monitoring the presence of atmospheric aerosols. Several thousand are generated and recorded on a typical night of operation. However, under some models, a horizontal air shower track is the expected signature of a neutrino induced shower. One could argue that all laser tracks should point back to the laser and can be identified that way. While theoretically true, clouds and aerosols, dead or noisy detector elements are the practical reality. As a result a small number of laser tracks fail to reconstruct as expected and consequently can make their way into other data samples. For physics analysis, and searches in particular, the effect of a few misidentified laser events can be very unfortunate.

Another approach to isolate calibration data from physics data might be to dedicate special times for writing special calibration files. While this approach works well for detector calibration before and after normal detector operation it would not be an optimal choice during night sky observation. The resulting loss of observation time would force a choice between less detector and atmospheric monitoring or less exposure to air showers.

2 Implementation at HiRes

A better solution is to generate calibration data at precisely known times. Then a physics blind identification can be made based only on event times. An even better solution is to use specific times with respect to the second and assign a "schedule" of offsets to each calibration system. Provided

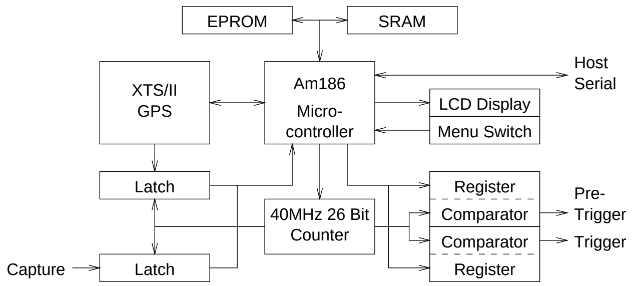

Fig. 1. Block diagram of GPSY module

Device	rate (Hz)	offset (ms)			
hr1sls	4	111	361	611	861
hr2sls	4	222	472	722	972
hr1cal	2	51	551		
hr2cal	2	52	552		
Mobile laser	4	0	250	500	750

Table 1. Millisecond offsets for HiRes calibration devices

the method of generating times has sufficient precision, one can simply separate all events recorded at these offsets as calibration data without loosing a significant amount of exposure. A side benefit is that the calibration events can be used to check the time measurements of the detectors and offline analysis.

HiRes has two steerable lasers that fire at 4Hz. These are seen by both detectors. Each detector also sees its local fiber-optic based laser calibration system that fires at 2Hz. Thus a detector could see a maximum of 8 calibration events a second. The effective time window dedicated to these events between 1 and 2ms. This represents a loss of observation time of 8 to 16ms each second, or 0.8 to 1.6 percent. Table 1 shows the "schedule" used at HiRes.

3 The GPSY module

We have designed and built several GPSY (Global Positioning System - YAG) modules to implement these scheduled calibration events. The GPSY modules were designed specifically to trigger a YAG laser and therefor has two trigger outputs: A pretrigger output "flash" to fire the YAG laser flash-lamp and a main trigger output "fire" to trigger the YAG laser Q-Switch a fixed number of microseconds after the flash lamp is fired. The GPSY module is not limited to firing lasers; any device that that can be triggered by a TTL logic level signal can be triggered by the GPSY module. The GPSY module can also time-stamp externally generated events with a 25 nanosecond resolution with it's "capture" input.

A block diagram of the GPSY module is shown in figure 1 The GPSY module uses a Motorola GPS receiver that is optimized for timing applications. The GPS receiver's one-pulse-per-second out latches and clears the value of a 40MHz, 26 bit counter. This latched value is read by a micro-controller to accurately measure the exact frequency of the 40MHz system clock. A second latch is provided for the external event capture input. The micro-controller initializes the GPS engine and reads the GPS status information through an RS232 serial port. To generate pretrigger and trigger output pulses, the micro-controller computes the predicted counter value for the desired time of the trigger pulse and loads the value into a trigger register. When the counter value matches the value loaded into the trigger register, an output pulse is generated.

The output pulse period, delay (pretrigger to trigger), pulse width, number of triggers in a trigger sequence and a list of start time for trigger sequences are all software programmable. Also, the pretrigger output can be fired without the trigger output to "warm" the laser and the trigger output can be programmed to fire after every N^{th} pretrigger pulse to allow for slow firing of lasers that have been optimized for a fast flash lamp fire rate.

The GPSY time capture resolution is 25ns and the trigger output time resolution is 1μs. The accuracy of the capture and trigger times is limited by the accuracy of the one-pulse-per-second output of the GPS receiver and is generally better than 100ns.

The GPSY module can used manually by a front panel LCD display and switches to select and modify operational parameters through a simple to use menu interface or by a RS232 serial interface to a host computer. The manual mode is useful for mobile operation of calibration equipment such a mobile steerable laser. In manual mode, a two line LCD displays a selectable configuration or status item on the top line and the items value on the bottom line. Two momentary-up/momentary-down switches are used to control and enter data; one switch selects which item to examine or modify and the other switch selects the value of the item. The host computer mode is useful for automated trigger sequences from permanently installed calibration equipment. When the GPSY module is used with a host computer, the time and GPS location of all triggered and captured events can be logged to a file on the host computer. Table 2 lists the commands that can be sent to the GPSY module through the host serial port. The GPSY module echos back commands with the current set values.

4 GPSY hardware and software

The GPSY module is controlled by a AMD Am186EM micro-controller. All of the event capture and generation logic is implemented in CPLDs (Complex Programmable Logic Devices) and the VHDL hardware description language. All of the GPSY micro-controller software was written in C and uses Jean J. Labrosse's MicroC/OS-II real-time kernel (Labrosse 1999). The GPSY software comprises eight separate real-time threads: two threads to compute the counter compare values for each of the two trigger outputs, a thread to log captured event times, a thread to capture the counter value on the GPS one-pulse-per-second and compute the counter frequency, a thread to read status messages from

<table>
<tr><th>Command</th><th>Comment</th></tr>
<tr><td>Help</td><td>GPSY replies with list of valid commands</td></tr>
<tr><td>Version</td><td>Returns current version of GPSY software</td></tr>
<tr><td>SecondOffset [<value>]</td><td>Set offset into GPS second of first trigger in sequence</td></tr>
<tr><td>FlashPeriod [<value>]</td><td>Set period of ‘flash’ trigger output in milliseconds</td></tr>
<tr><td>FlashWidth [<value>]</td><td>Set pulse width of ‘flash’ trigger output</td></tr>
<tr><td>FlashCount [continuous| <value>]</td><td>Set number of ‘flash’ triggers before first ‘fire’ trigger</td></tr>
<tr><td>TriggerDelay [<value>]</td><td>Set delay between ‘flash’ and ‘fire’ output pulses</td></tr>
<tr><td>TriggerPeriod [<value>]</td><td>Set period of ‘fire’ trigger output in ‘flashes’</td></tr>
<tr><td>TriggerWidth [<value>]</td><td>Set pulse width of ‘fire’ trigger output</td></tr>
<tr><td>TriggerCount [continuous| <value>]</td><td>Set number of ‘fire’ triggers in sequence</td></tr>
<tr><td>TriggerList [<hour>:<minute> [<count> |/]]</td><td>Add/Remove trigger sequence start time from list</td></tr>
<tr><td>Trigger [$Stop|Flash|Fire$]</td><td>Set trigger mode</td></tr>
<tr><td>Capture [$Disable|Enable$]</td><td>Set capture mode</td></tr>
<tr><td>ClockSource [$GPS|CPU$]</td><td>Set source of one-pulse-per-second</td></tr>
<tr><td>GpsBaud [<value>]</td><td>Set serial baud rate to GPS serial port</td></tr>
<tr><td>HostBaud [<value>]</td><td>Set serial baud rate to host serial port</td></tr>
<tr><td>HostEcho [$Disable|enable$]</td><td>Set host serial port character echo mode</td></tr>
<tr><td>Date</td><td>Return current GPS date</td></tr>
<tr><td>Time</td><td>Return current GPS time</td></tr>
<tr><td>Latitude</td><td>Return current GPS latitude</td></tr>
<tr><td>Longitude</td><td>Return current GPS longitude</td></tr>
<tr><td>Altitude</td><td>Return current GPS altitude</td></tr>
<tr><td>GPSInit</td><td>Initialize GPS engine</td></tr>
<tr><td>GPSLog [$Disable|Enable$]</td><td>Set mode to log GPS messages to host serial port</td></tr>
<tr><td>GPS <hexafied message></td><td>Send message to GPS engine</td></tr>
</table>

Table 2. GPSY Host Serial Commands

the GPS receiver and update GPS related time and location data, a thread to write queued text messages (generated by other threads) to the host serial port, a thread to read and execute commands from the host serial port, and last, a thread to poll the front panel switches and update the LCD display.

5 Conclusion

Ten GPSY modules have been constructed so far. Two units have been working reliably in regular use at the HiRes2 site since August 2000 and one unit at the HiRes1 site since October 2000 (The HiRes1 fiber optic calibration system has not yet been retrofitted with a GPSY trigger). Additionally, the GPSY triggered calibration events have proven to provide a useful check of detector timing.

6 Acknowledgments

This work is supported in part by the National Science Foundation (grants PHY-93-22298, PHY-99-74537, & PHY-99-04048), the U.S. Department of Energy (grant FG03-92ER40732), and by the Australian Research Council. We gratefully acknowledge the contributions from the technical staffs of our home institutions. We would like to thank Colonels John Como and Edward Fisher as well as the staff of Dugway Proving Ground for their continued cooperation and assistance. We would also like to thank the University of Utah's Center for High Performance Computing for their support.

References

J.H.V. Girard, et al., A fiber-optic-based calibration system for the High Resolution Fly's Eye cosmic ray observatory, Nuclear Instruments and Methods A, 460, 278-288, 2001.

L.R. Wiencke, et al., Steerable Laser System for UV Atmospheric Monitoring at the High Resolution Fly's Eye, in Ultraviolet Atmospheric and Space Remote Sensing: Methods and Instrumentation II, G.R. Carruthers, K.F. Dymond, Editors, Proc. of SPIE Vol 3818, 46-55, 1999.

J.J. Labrosse, MicroC/OS-II The Real-Time Kernel, R&D Books, ISBN: 0-87930-543-6