

A SOFTWARE UPGRADE FOR THE SNS WIRE-SCANNER*

W. Blokland, C. Long, ORNL, Oak Ridge, TN 37831, U.S.A.

Abstract

The SNS Beam Instrumentation Group has designed new software for all wire scanners at SNS. The original wire-scanner software was written by one of the Spallation Neutron Source (SNS) partners, Los Alamos National Laboratory. This software was designed for the types of wire-scanners initially planned and their usage at that time. New variations in the wire-scanner hardware added gearing, different position read-back methods, and a timing card. The new software handles these hardware variations in a flexible manner through configuration control files. The software upgrade allows the user to synchronize the stepping of the fork with external applications, such as with loss monitors to calibrate energy dependence of the beam losses caused by the wire. Another new functionality allows physicists to select what part of the beam pulse is used to determine the transverse profile after the data has been taken. This avoids time-consuming rescans in case the timing was not initially correct. The new software, also a LabVIEW program, is structured around a state-machine with sequence capabilities to manage the complexities of stepping through a scan and interacting with the user. This paper discusses the new software features, the implementation, and the obtained results of field tests.

INTRODUCTION

Wire-scanners are used in the warm linac and the transfer lines to determine the transverse profiles of the accelerated particle beam. During a scan a fork holding three wires, one horizontal, one vertical, and one diagonal wire is stepped through the beam. As the beam hits each wire, it deposits on or strips charge from the wire. The resulting signal is low-pass filtered (30kHz) to reduce noise and sampled by a digitizer (up to 1Mhz). The sum of the trace at each step is plotted versus the wire position to create the transverse profiles.

The SNS accelerator was built in phases with different labs designing and supplying the wire-scanner hardware.

The software for these scanners was implemented by LANL and tested with the first wire-scanners at Berkeley, [1]. This software has now been successfully used for over 5 years. However, due to several factors it was decided that the software needed to be upgraded.

One factor was that new wire-scanner actuators, see Figure 1, has been developed that uses different motors, gearing, and position read-back methods. The old software has trouble handling these new variations leading to incomplete scans. Its code would have to be modified for each new style resulting in different versions, or a significant rewrite of the old code would have to be done.

Another factor is that the old software uses the ActiveX interface to Channel Access. This was the only available LabVIEW to Channel Access interface available to us at the time. Now most SNS diagnostic instruments use the Shared Memory IOC, which has faster processing and has a quicker response than the ActiveX interface.

Another motivation is the need for additional features. One important feature is the synchronization of the scan, the stepping through the beam, with an external application such as the Loss Monitor. By giving the loss monitor program control over the stepping of the wire-scanner, we can determine the energy dependency of the loss monitors.

Taking a scan from single wire-scanner can take several minutes. Often a scan has to be redone because the timing was not quite right. During studies the beam parameters are often adjusted, for example from a chopped beam to an unchopped beam. This change in the beam also requires a change in the setup parameters of the wire scan. To get these scan parameters just right can take several scans. At several minutes per scan this can become quite time-consuming.

All these factors combined lead us to the decision to rewrite the program.



Figure 1. Photo of a new CCL wire-scanner actuator.

* ORNL/SNS is managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725

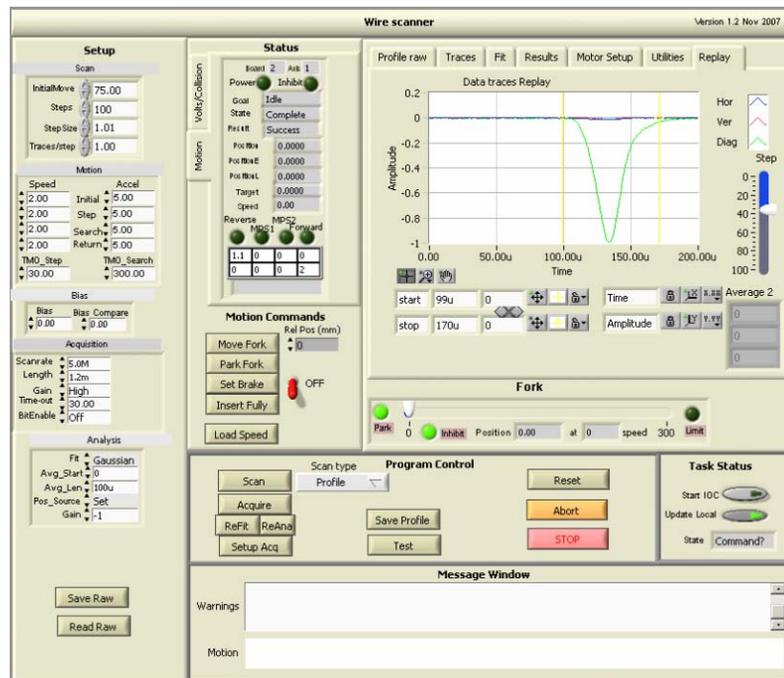


Figure 2. The front panel of the LabVIEW-based wire-scanner application.

IMPLEMENTATION

One of the standard SNS templates for LabVIEW-based instruments is the state machine. The template includes support for the Shared Memory IOC and configuration files that are stored in the Oracle database, [2].

The application's front-panel, see Figure 2, shows the setup for the scan, motion, digitizer, and analysis on the left. The middle of the figure shows the status of the motion and motion commands. The program maintains continuous information about the motion to verify that the fork is not stuck. The top right of the figure shows the results of the scan including the Gaussian fit to estimate the width of the beam.

Program Structure

For this particular application where the application must control a series of steps and acquisitions, we chose to use the state-machine with sequencer. The sequencer allows the programmer to define a scan as a sequence of simple actions, such as move fork 1 mm, acquire trace, and analyze trace that are each mapped into a state of the state machine. Parameters can be passed to each state to describe items such as the length of the move of the fork or what type of analysis must be done. The sequence can be as complex as needed and generated from a routine using specific parameters or read in from a file. During the execution of the sequence, after exiting a state, the state-machine returns to the "Command?" state where it checks if there was user input, e.g to stop the sequence, before continuing to the next step. This allows the program to be responsive during the execution of a sequence using the same state that is called when the program is idle. This does require that each state is

executed relatively quickly and can't linger around waiting for a condition, such as the completion of a move, to become true. The sequence can call the Wait state to check for this condition and the sequence is not allowed to progress unless the condition has been met. The state machine structure is shown in Figure 3.

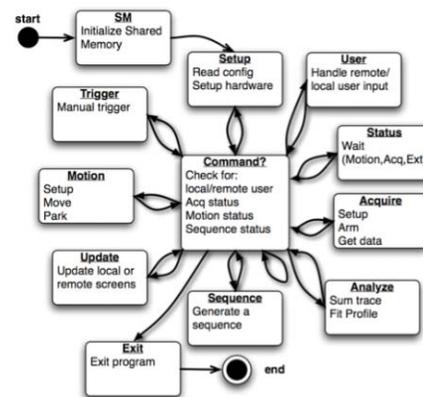


Figure 3. The State-machine structure.

Hardware variation

The new program handles the variation of the hardware from the ground up. From the main program, a technician can start a program to input the specifications of the individual parts such a gear, motor, and position readout device, to create the entries for the configuration file. An additional tool tests the motion of the fork independent of the main program.

Timing Adjustment

To support changing the timing after the data has been taken, more data is stored in memory than is actually used for a single profile. The old program only takes the data it uses for the profile as defined. The new program takes data before and after the data required for the currently defined profile. During a wire scan the SNS beam is limited to at most 100 usecs. The chopped or unchopped beam, will not be further away than 200 usec either direction so that a trace of only about 500 usec will capture the beam under almost any circumstances. Even with oversampling at 1 Mhz this requires only 500 samples per trace. With about 120 steps and perhaps two to four multiple traces per position to do averaging, the program has to store less than 1 million data points. This is not a problem with PCs having now 512 Mbytes to 1Gbytes of RAM. After the scan the user can modify the start and stop locations for the integration and immediately see the effect. This feature also allows us to instantaneously see to some degree (the signal is lowpass filtered) the difference in profiles between the head and the tail of the beam.

Synchronization

Synchronization with other applications is easily implemented with the state-machine. It already has a "Status" state to see if the motion or acquisition has completed. The "Status" state can now also be asked to wait on a change to a Channel Access Process Variable (PV). The loss monitor program can use this PV to halt the stepping until it is done with its loss measurements.

RESULTS

First the program was tested without any motion or digitizer hardware. Data was simulated to test the logic of the program and the analysis routines. The next step tested the program in the lab with an actuator but with no signal to the digitizers. Next, the program was been tested in the HEBT section of the SNS accelerator. Figure 4 shows the beam pulse for which the transverse profile was being measured.

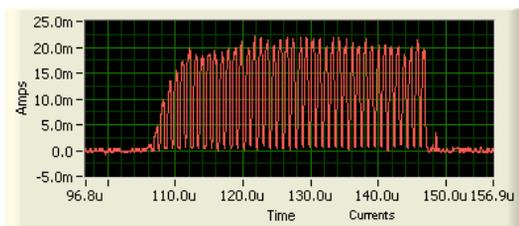


Figure 4. Plot of the beam being measured.

Figure 5 shows the data traces as the beam hits the vertical wire. Figure 6 shows the profiles for all the wires from a different scan. Obtaining these results mean that the program was able to complete a full scan and analyze the data. The interface to the control system has not been finalized.

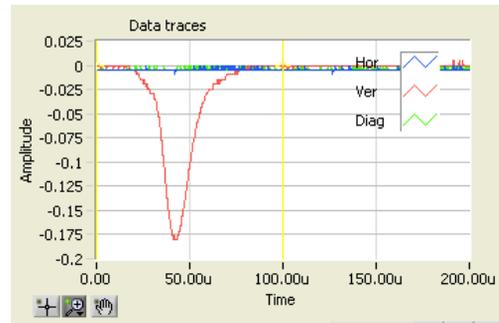


Figure 5. One data trace for the diagonal wire

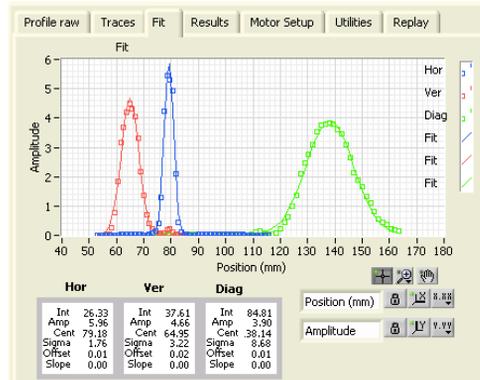


Figure 6. The profiles.

FUTURE

While the program has been tested with two types of wire-scanners, we will have to test it with the other types as well before starting to update all the wire scanners. As many wire-scanner use PCs that are around 5 years old, we will also replace the older PCs at the time of the software upgrade.

Besides being operator controlled through standard EDM screens, the wire-scanners are also accessible through applications of the physics groups using the XAL Java-based framework, [4]. Therefore the switch over has to be carefully coordinated to make sure that all wire-scanner whether running the old software or the new software are accessible.

REFERENCES

- [1] M. Plum et al. "The SNS Linac Wire Scanner System, PAC'03, Portland, May 12-16, p2485-7.
- [2] W. Bloklund et al, "Network Attached Devices at SNS", DIPAC'03, Mainz, Germany May-5-7, 2003, pp 146-8.
- [3] W. Bloklund and C. Long, "An In-Line Emittance Scanner Based On A LabVIEW Style State Machine With Sequencer", ICALEPCS'05. Geneva, 10 - 14 Oct 2005, PO1.057-6.
- [4] J. Galambos et al., "XAL Application Programming Structure," <http://accelconf.web.cern.ch/AccelConf/p05/PAPERS/ROPA001.PDF>, PAC'05, Knoxville, TN