THE CIS CONTROL SYSTEM AT IUCF *

J.C. Collins†, Wm. Manwaring
IUCF, Bloomington, IN

Abstract

CIS is the 200 MeV Cooler Injector Synchrotron now being commissioned at IUCF. This paper gives an outline of the entire control system, emphasizing its software aspects. While this is a modest project done with small budget and staff, it includes control of an ion source, an RFQ-Linac pre-injector, a synchrotron and associated beamlines. The hardware platform is based on VMEbus, using modules both purchased commercially and developed in-house, DEC Alpha workstations, X-Terminals and Vsystem software system. This paper discusses our hardware-software interfaces, experience with operator generated control screens, diagnostics displays using BPM and Harp beam sensors, magnet ramp calculation, modification and control, cycle-to-cycle ramp corrections and timing control.

1 INTRODUCTION

The Cooler Injector Synchrotron (CIS)[1] is a 200 MeV, 17m circumference machine now being commissioned at IUCF as an injector to our Cooler Ring. CIS itself is injected by a commercial RFQ-DTL and a simple duoplasmotron source. The duoplasmotron will soon be replaced by the CIS Polarized Ion Source (CIPIOS)[2]. The control system for CIS and CIPIOS evolved from that for the HIPIOS ion source[3], the latter representing a major shift in controls philosophy at IUCF from a totally (hardware and software) in-house built system to a commercially based one. (In fact, the Cooler control computer, its software and the timing system will be replaced later this year by the newer style computer and software to simplify operations. However, full hardware replacement does not seem cost effective at this time.)

One Compaq/DEC Alphastation 200-4/233 serves CIS. The computer communicates with an Allen-Bradley PLC via Ethernet for all on/off, binary status and interlock operations and with VMEbus via a commercial PCI-VME interface for analog, 1553, GPIB and serial control operations. CIS controls uses four 6U and 18 3U VME crates, at present. The computer and three 6U crates are connected in series via copper; the last 6U and all 3U crates are connected to this central group via optical fiber in a star topology. Everything with which the operator is to interact goes through the control computer. While we often use PCs to develop diagnostics, we always design them for eventual integration with the VME system.

Vsystem[4] from Vista Controls provides the framework for building operator displays and control software. Most displays are created using the Vsystem GUI, although a few are windows generated directly by programs using Motif. All programs are written in C using the Vsystem API. The standard operator station is an X-terminal (or PC with suitable host software). To date, only one (of four) CIS operator stations has a pair of knobs for cost reasons. We have recently found a commercial solution, which, if it works, is inexpensive enough that a pair of knobs will become part of the "standard" station.

2 HARDWARE

The most interesting hardware in the CIS control system are the HiRel and sequencer modules, both designed and built at IUCF and described in detail elsewhere in these Proceedings[5]. In summary, the HiRel modules are DAC/ADC combinations which sacrifice speed for accuracy and isolate every signal. They were designed and built because commercial units could not meet the specifications of the CIS designers. The sequencers are limited purpose computers originally designed for ramping, for which we found no commercial alternative.

2.1 Sequencer

The sequencer instruction set was designed to facilitate the generation of output voltages, signals (TTL levels and serial codes) and the reading of analog values at specified time intervals in a repetitive fashion, all for use in magnet ramping. In use, a program on the host control computer generates a sequencer program, downloads it and any necessary data and, perhaps, starts the sequencer program. For ramping, a sequencer is downloaded with ~20,000 DAC values and a program which outputs those values at timed intervals (typically, 30 μsec) during the half-second long ramp. Sequencers also serve in the accelerator timing system and some high speed diagnostics readouts as described below.

2.2 Timing System

The hardware timing system consists of one sequencer and four V102 modules designed at Brookhaven National Laboratory[6]. The sequencer is the master timer, sending trigger events to the V102s (perhaps multiple times in one timing cycle), sending start signals to the ramping sequencers, providing TTL output pulses and accepting TTL inputs for timing program control. The V102

*Work supported by the National Science Foundation grant NSF PHY 96-02872 NUC RES and Indiana University.
†Email: collins@iucf.indiana.edu

modules assert TTL signals of programmable width at programmable delays relative to trigger events. Unlike the sequencer outputs, the V102 outputs are easily grouped together and associated with events. Also, their delay and width values can be changed on-the-fly, allowing users to tune timing in ways which are impractical using only sequencers.

3 SOFTWARE

Vsystem provides tools for display creation, data logging, status alarms and script-driven processes, all of which features we use but about which we shall say nothing here. The smallest unit of hardware access is the channel, which can have other uses as well. Channels are organized into databases. By adopting conventions for defining databases, channel names and reader/writer programs, operations people can name devices and generate displays in parallel with software development, speeding installation. This also helps give operations people a better understanding of the control system and allows them to make some changes themselves, helping keep the developers' blood pressure low. Although at one time there was a lot of talk at IUCF about personalizing operator displays, there has been no movement to actually do that after five years of use (for HIPIOS).

Problems of space and selection persuaded us to make this paper all text. Examples of all the display screens discussed herein may be found on the IUCF Web site[7].

3.1 Knobs

Because IUCF operators used knobs in a particular fashion for 20 years, a Motif program was required to emulate that mode of operation. The program allows each of two knobs to be attached to DAC channels, the channels being organized into display pages defined in an ASCII file editable by operators. The program remembers and restores attachments as pages are changed, it can save and restore DAC values for all channels on a page and it supports "presets", predefined attachment selections to make standard tuning operations faster.

3.2 Save/Compare/Restore

The operational data archival function is performed by a Motif program which performs the three functions in its name, using both disk files and the present machine state as data sources or sinks as appropriate. The data files are created with fixed length data items and no delimiters just so we are not tempted to edit DAC changes into archives.

3.3 Timing

Access to timing hardware and generation of the timing sequencer program is done through a timing server program and timing channels. All event delays and timing parameters have their own channels in the timing database. For operational purposes, we find it convenient to define two timing modes: one wherein beam is only injected into CIS and one which adds acceleration and extraction. When a request to start timing is made, the server program generates and downloads the sequencer program appropriate to the active mode and starts the timing sequencer. If timing mode or values are changed, the server stops the sequencer, creates and loads a new program and restarts the sequencer. The timing sequencer program can cause the sequencer to generate a VME interrupt at a defined point in the timing cycle; this is the end-of-ramp event referred to below. The program also uses bits set in a sequencer register by the host and sequencer TTL inputs to conditionally suppress source, ramp and/or extraction trigger events. This allows the operator to turn off CIS beam when it is not needed to fill the Cooler.

A Motif program provides the normal user interface to the server, although any program can access the server through the timing channels. For all time delays implemented in V102 modules, the user interface program supports direct manipulation of values via either keyboard or mouse buttons (for "no-look" tuning). Groups of outputs (e.g., ion source, RFQ/DTL) can be defined such that the delays of all group members can be moved with respect to all other delays while retaining relative delays within the group. Since updating time delays implemented in the sequencer is slow (create and download a new program), only keyboard input is supported for them. The program also supports temporary and archival storage of timing data.

3.4 Ramp Calculation

This Motif program takes as input starting and ending beam energies (MeV), desired ramp duration τ(msec) and ramp shape parameters P. Its output is seventeen 100-element ramp arrays of points equally spaced in time, one array for each ramping device. The 100-element length, a compromise between ramp accuracy and ease of modification of points by operators, came from experience on the Cooler and works quite nicely. Most ramps are functions of $B\rho(t)$ and are based upon a master $B\rho(t)$ curve, which can take various functional forms; at present we use $B\rho(t) \propto (\sin t\pi/2\tau)^{**}P$, where P=2. The dipole power supply requires both a current ramp, derived from $B\rho(t)$ and mapping data, and a feed-forward voltage ramp, based on $V = iR + L(di/dt)$, with empirical corrections for measured changes in L as a function of current. Ramps for quadrupoles, steerers and RF frequency are all derived from $B\rho(t)$. The ramps for the RF cavity ferrite bias magnets are based on empirical data tables, with the user able to set the slope of the linear RF voltage ramp. Each ramp is stored in a Vista database array-channel, from where it can be graphed or accessed by other programs.

3.5 Ramp Modification

This is a Motif program that allows the operator to fine-tune ramps by modifying any of the 100 points of the ramp arrays generated by the calculation program. Modifications can be applied point-by-point or to each point in a specified interval and are entered as a percentage of the unaltered value in either case. The program also permits easy addition of a predefined ramp change to any of the four dipole trimcoils to position the beam properly near the end of the ramp for the extraction kick. A single command stops the ramp (via the timing server), loads all modified ramps and restarts the ramp. The RF Frequency ramp can be modified automatically via the BPM program described below. Those modifications are displayed by this program and are updated as the BPM program changes them.

3.6 Ramp Control Interface

This program acts as a server controlling the ramping sequencers. Its functions include

- Download the sequencer ramping programs; start and stop ramp execution.
- Read the 100-point ramp arrays, convert them, via interpolation, into ~20,000 DAC values, then download these values to sequencer RAM.
- Create the linear reset ramp and download it.
- Perform single write operations to a sequencer DAC when the corresponding channel value is changed, since a sequencer acts as the DAC and ADC for the power supply to which it is connected. Also, poll all sequencer ADCs, updating the corresponding channels (displays).

3.7 Beam Position and Ramp Correction

Pictures of the beam position during a ramp are obtained by attaching a 48-channel ADC multiplexer to a single sequencer and running a sequencer program that steps through the mux channels, reading and storing ADC values in RAM. During a ramp, each channel is sampled every two milliseconds. Upon receipt of an end-of-ramp event, a host program scans the sequencer RAM and extracts 200-point position (mm) and 200-point beam current (mA) arrays for each of four BPMs, writing these arrays to database channels for graphing and/or calculation. A single sample of BPM values during Fill or Flattop can be triggered by the timing system. This program can read these single values and update database channels for bar-graph display.

The host program can also correct the RF Frequency ramp in such a way as to flatten a chosen BPM position vs. time graph. The operator can choose to "flatten BPM-2 from 200 msec to 600 msec at -3mm offset from center", for example. In this case, to each RF frequency ramp value in the interval 200-600 msec after ramp start, the program applies a correction proportional to the difference between the corresponding BPM-2 position and -3 mm and informs the Ramp Modification Program (see above) that a change was made. The modified ramp is then loaded. The present correction algorithm typically requires five or six iterations to achieve satisfactory results; more sophisticated algorithms are under test to achieve faster convergence and smoother RF ramps.

3.8 Harp Display

Three 48-wire (24 vertical, 24 horizontal) harps[8] are multiplexed to a commercial VME ADC module which includes FIFO RAM. A timing system trigger initiates sequential conversion and storage in the FIFO of the wire currents. Upon receipt of an end-of-ramp event, a host program reads the FIFO values into six 24-element database arrays. For each array, the program subtracts an array of background values, converts each resulting element into proper units and calculates a beam profile centroid and width (each array should contain one peak) and an equivalent Gaussian curve for visual comparison. A Vsystem tool displays three horizontal or vertical arrays in different colors on the same graph. The operator can use these displays to tune beam position and focus. Six additional harps will be included in the extraction beamline.

4 ACKNOWLEDGEMENTS

The authors would particularly like to acknowledge the efforts of all IUCF electronics technicians and operator Pete Goodwin for his work in generating control screens and maintaining databases.

5 REFERENCES

[1] D.L. Friesel, "Performance of the IUCF Cooler Injector Synchrotron", these proceedings.
[2] V.P.Derenchuk, et al., "Polarized Beam for the IUCF Cooler Injector Synchrotron", 7th Inter. Workshop on Polarized Gas Targets and Polarized Beams, Urbana, IL (1997).
[3] J.C. Collins, et al., "The IUCF High Intensity Polarized Ion Source Control System", 13th Inter. Conf. On Cyclotrons and Their Applications, Vancouver (1992).
[4] P.N. Clout, et al., "A Comparison of Vsystem and EPICS", ICALEPCS'97, Beijing, China (1997).
[5] W. Hunt, "CIS Control Hardware at IUCF", these proceedings.
[6] RHIC Report AD/RHIC/RD-16 (1993).
[7] http://www.iucf.indiana.edu/Facilities/Controls.
[8] M.Ball, et al., "Beam Diagnostics in the Indiana University Cooler Injector Synchrotron", 7th Workshop on Beam Instrumentation, Argonne, IL (1996).