UPGRADES TO CONTROL ROOM KNOBS AT SLAC NATIONAL ACCELERATOR LABORATORY *

S. Hoobler#, S. Alverson, M. Cyterski, R. Sass, SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, U.S.A

Abstract

For years, accelerator operators at the SLAC National Accelerator Laboratory (SLAC) have favored hardware knobs in the control room for accelerator tuning. Hardware knobs provide a tactile, intuitive, and efficient means of adjusting devices. The evolution of separate control systems for different accelerator facilities at SLAC has resulted in multiple flavors of knob hardware and software. To improve efficiency, space usage, and ease of use, the knob systems have been upgraded and integrated.

BACKGROUND

SLC Knobs

In the 1990s a knob system was developed for the SLAC Linear Collider (SLC), using hardware knob chassis designed by Miranova Systems. SLAC developed software and a DOS knob controller to interface the knob chassis to the SLC VMS control system [1].

EPICS Knobs

The Linac Coherent Light Source control system is based on EPICS (Experimental Physics and Industrial Control System). EPICS software was written to interface the existing knob hardware to the LCLS control system. At that time, it was presumed that the VMS control system would soon be decommissioned and no effort was made to integrate the two systems.

Knob Use at SLAC

To date, the VMS control system and its knobs are still used by the Facility for Advanced Accelerator Experimental Tests (FACET) and the Next Linear Collider Test Accelerator (NLCTA). EPICS knobs are used at LCLS, the SPEAR storage ring, and NLCTA.

Until recently, the VMS and EPICS knob systems were still not integrated, making an inefficient use of space and knob hardware. To improve this, the knob software and hardware have been updated to allow individual boxes to be used by multiple control systems simultaneously.

DESIGN CONSIDERATIONS

Backward Compatibility

To stay compatible with the legacy knob hardware and software, the new system supports the existing communication protocols.

System Performance

The system should maintain the performance standard of the previous systems.
- Good knob resolution.
- Quick response.
- Allow same device to be assigned to multiple boxes simultaneously without conflict.
- Ability to restore device to original value.

Enhancements

To support operation of multiple facilities from the same location, the system was enhanced to allow a single box to be used with multiple control systems simultaneously.

Maintainability

The system will initially use the legacy protocols to remain backward compatible. As older systems become decommissioned, it may be desirable to migrate to other protocols. The software should be modular to facilitate this.

Flexibility

Unlike the legacy SLC control system, the LCLS EPICS control system is distributed. There is no pre-defined list of devices that may be attached to knobs. Any EPICS process variable may be attached. The software should be flexible to handle generic cases and should also provide support for device-specific settings and facility-specific conventions.

SYSTEM IMPLEMENTATION

A knob system consists of the elements shown in Figure 1. The Knob Controller handles all communication with the hardware knobs. The hardware knob boxes and the Knob Controller together act as a server to the Knob Users. The Knob Users are the accelerator control systems which wish to attach devices to knobs. The Controller is the interface between the Users and the hardware knobs and will accept requests from any User in the system.

In this implementation, the knob hardware uses a serial protocol while all software entities use Ethernet protocols. A terminal server performs the conversion between serial and Ethernet packets. An individual knob box has four knobs, each of which can be independently controlled by a different control system. A Knob Controller and terminal server can support up to 16 knob boxes.
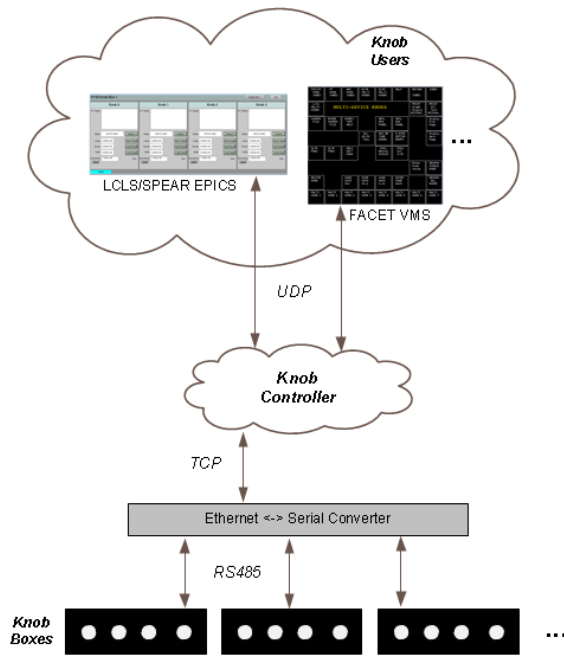
Figure 1: Knob system overview.

HARDWARE

The new knob box hardware is a small desktop chassis. It has four knobs, each with a LCD display, a disable button, and a button used to restore an attached device to its original value. Figure 2 shows a photograph of the new knob box hardware.

Figure 2: Photograph of the Knob Box Pro.

The knob box tasks are separated into two main processes, each of which is executed by a microcontroller, the 16 MHz PIC16F1937. One controller is dedicated to counting rotational ticks and determining direction from the knob optical encoders. The second handles communication with the various devices: the LCD displays, the knob encoder microcontroller, and the software Knob Controller, the latter of which uses the RS-485 serial protocol. This separation of responsibility helps prevent missed knob turn counts and improves the response time to the Controller.

In addition to the optical encoder, each knob also has two associated buttons. The first is a latching switch that toggles the knob between disabled and enabled (counts for that knob are 'zeroed' while it is disabled). The second button is a momentary switch which is coupled to a 555 timer integrated circuit. When depressed by the

user for 5 seconds, the microcontroller will set the 'restore' flag in the message to the Knob Controller, causing the software to set the device back to its original value.

The box contains two rotary potentiometers to control brightness and contrast of the LCD screens. These are read back through the analog-to-digital converter on the communications microcontroller. Each box also contains an internal set of eight DIP switches which can be used for assigning an 8-bit hardware address for the RS-485 communication protocol. Figure 3 shows a block diagram of the knob box system.
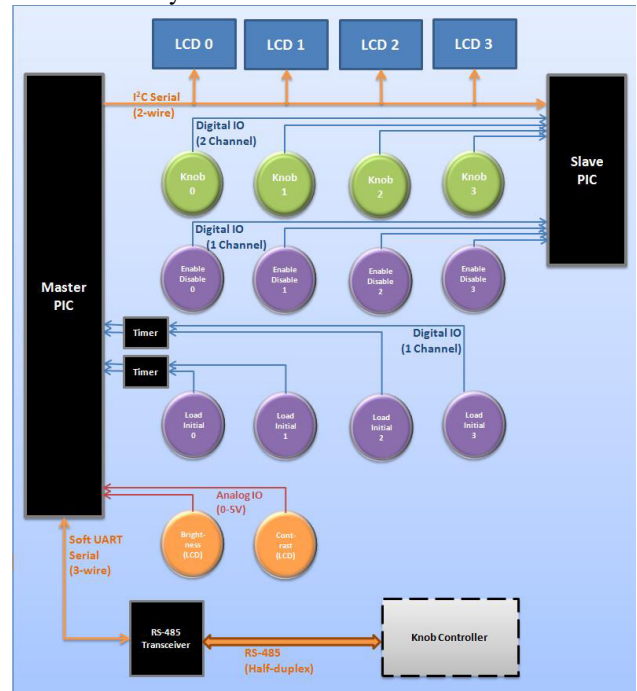

Figure 3: Knob box hardware block diagram.

SOFTWARE

The Knob Controller and EPICS Knob User are software processes running on Linux servers. They are comprised of 'C' code running in the context of EPICS IOC (Input Output Controller) processes. The legacy VMS Knob User is a combination of 'C' and Fortran software running on an Alpha.

Knob Controller

The Knob Controller handles communication with the hardware knobs. While devices are assigned to knobs, it polls the knobs for changes and sends the information to the Knob User. The Controller writes updated device information to the knob hardware display when requested by the User.

The Controller uses multiple threads to optimize performance, primarily by performing slow serial operations in parallel.

A single thread processes incoming messages from Users. It listens on a dedicated UDP port, awaiting requests. When a new attach request is received, the Controller assigns the requesting User to be the new knob

owner. It saves the User's information for future messages and identification. Other than attach requests, any message that does not come from a knob's owner is ignored. When a User takes ownership of a knob, the Controller notifies the previous owner.

Additionally, each knob box has a dedicated thread. It polls the attached knobs for changes, sending the results to the User. It receives and handles additional information from the box, including notification of a restore request, a recent reboot, and the box's revision information. These last features are only supported by the Knob Box Pro.

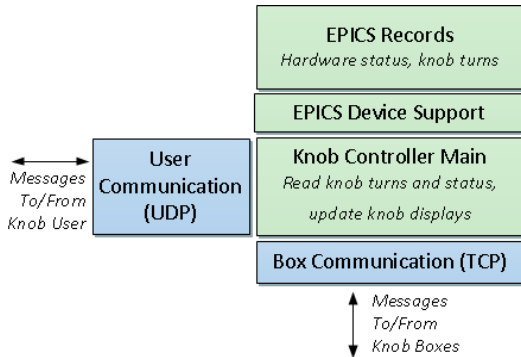Figure 4 shows the architecture of the Knob Controller software.

Figure 4: Knob Controller architecture. Blue sections are implemented in callbacks.

EPICS Knob User

The Knob User is specific to a control system. It provides the graphical user interface (GUI) for attaching devices to knobs. When a User receives knob turn information from the Controller, it converts knob turns to engineering units and sets the attached device to its new setting. In addition to attaching and detaching devices, the GUI provides the ability to modify and save knob sensitivity values and to save and restore device settings. Figure 5 shows the architecture of the Knob User software.
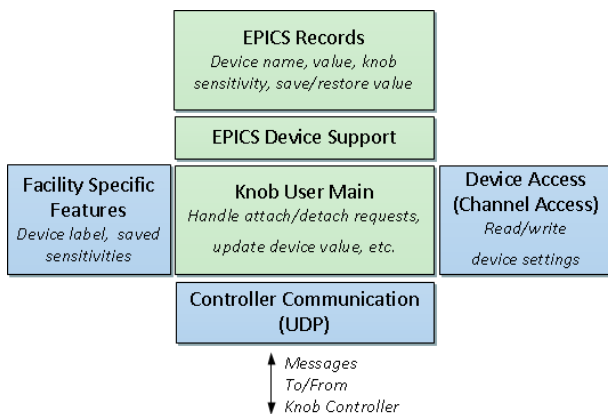

Figure 5: Knob User architecture. Blue sections are implemented in callbacks.

FUTURE IMPROVEMENTS

- Indicate on knob hardware which system owns each knob.
- Allow users to save device label to be used on knob display next time (this is already done for sensitivity).
- Allow users to save device precision to be used on knob display next time.

REFERENCES

[1] R. Johnson, R. Sass, "PC Knobs Software Design," December 1995, SLAC