

DESIGN AND IMPLEMENTATION OF A FINITE STATE MACHINE QUEUING TOOL FOR EPICS

D. Beltran, Synchrotron Light Laboratory - LLS-IFAE; J.A. Perlas, Synchrotron Light Laboratory - LLS-IFAE; J. Rosich, Synchrotron Light Laboratory - LLS-IFAE

In the process of making a detailed design for a synchrotron accelerator in Barcelona, we have set up a magnet test facility to evaluate the performance of a dipole prototype. We have designed the control system of this measurement bench as an evaluation for the accelerator control system and in this sense we have used the software and tools that we would use for the accelerator complex. Our software toolkit is EPICS, which has been an excellent choice since it provides the scalability, flexibility and robustness necessary for both the magnet test facility and the whole synchrotron accelerator. Nevertheless, we have found severe limitations in some EPICS tools. One of these tools is called the Sequencer, which is provided for applications requiring the Finite State Machine (FSM) approach in their design and which we use extensively in our software. In this article we present a new design (called Finite State Queuing Tool or fsqt for short), we discuss its advantages and we compare it with the old design using a new implementation for our test bench applications. The Finite State Queuing Tool provides several layered libraries to access Channel Access easily both synchronously and asynchronously, using the C-language directly without any intermediate pre-processing. It provides an event queue that avoids losing any transition due to an external or internal stimulus (user intervention or auto-change state) while guaranteeing the proper time sequence. In addition, an easy-to-use system library is provided containing facilities for watchdog timers, events, signals, semaphores and others. And last but not least, fsqt contains a graphical editor that allows to easily design the FSM and even generate the corresponding C-code in the host computer. In this way, the programmer can very easily change the program logic with the guarantee that he will get consistent code.