UVX CONTROL SYSTEM: AN APPROACH WITH BEAGLEBONE BLACK

S. Lescano, INSA Lyon, Villeurbanne, France
A. R. D. Rodrigues, E. P. Coelho, G. C. Pinton, J. G. R. S. Franco*, P. H. Nallin, Brazilian Synchrotron Light Laboratory (LNLS), Campinas, Brazil
guilherme.franco@lnls.br

Abstract

UVX is a 1.37 GeV synchrotron light source that has been in operation by the Brazilian Synchrotron Light Laboratory (LNLS) since 1997. Its control system, which was completely developed in-house, has received some upgrades lately in order to get around issues from aging, improve performance and reduce maintenance costs. In this way, a new crate controller was designed. It is based on BeagleBone Black single-board computer (SBC) [1], a cheap open hardware and community-supported embedded Linux platform that will be adopted for some control system applications in Sirius [2], the upcoming brazilian light source. In this work, we describe an overview of the design and results obtained.

LOCO (LOcal COntroller)

- Low-level hardware for control applications.
- Developed about 20 years ago.
- 3U crates of Eurocard-sized boards.
- I/O cards with digital and analog inputs and outputs.
- Local controller: special card with an universal software, responsible for managing I/O cards and communications to the upper levels of the control system. There are three generations of this board.
- "Transparent" architecture: only high level applications know what equipments are controlled by each crate.

Motivation for a new controller design

- Provide a substitute for old controllers. All of them are obsolete.
- Evaluate a new computing platform in a real operation environment.

Hardware design

We made a simple “carrier PCB” design for BeagleBone Black, just placing in a Eurocard-sized board digital buffers between BeagleBone Black GPIO pins and the LOCO bus signals on crate backplane. The board also has a counter, used during synchronized operations over power supplies (energy ramp and magnets cycling), a 7-segment display for status indication (just as it was since the first generation of local controller boards) and a reset monitoring circuit.

Embedded software

Software for this new design is an adaptation of third generation controllers software (named PROSAC). Low-level routines for reading and writing I/O cards were modified. Now they use BeagleBone Black GPIO pins. PROSAC runs four threads, described below:
- Main thread: launches the other threads and periodically updates a 7-segment display, which shows local controller status.
- Reader thread: reads continually all inputs from the I/O cards, storing these values in RAM memory.
- Networker thread: deal with client I/O operations through a TCP/IP socket. When the client only wants to read the inputs of the cards, the last values obtained by the reader thread (stored in RAM) are retrieved.
- Interrupter thread: this thread is active when the controller is performing synchronized operations over power supplies. Operation of this thread assumes that controller has in RAM memory a waveform, which is point-by-point traversed each time it receives a trigger pulse from UVX timing system.

Threads tuning

In order to evaluate and improve interrupter thread performance, we considered a series of tests exploring PROSAC configuration parameters and important aspects of the embedded operational system (Linux), such as its kernel configuration and scheduling policies and priorities.

Configurations that didn't present a good performance:
- Linux kernel with PREEMPT_RT patch ([6]), SCHED_FIFO scheduling policy and CPU frequency set to 1 GHz.
- Traditional Linux kernel, SCHED_FIFO scheduling policy and CPU frequency set to 1 GHz.
- Traditional Linux kernel, Completely Fair Scheduler (CFS) with appropriate nice values for each thread and CPU frequency set to 1 GHz.

Chosen configuration:
- Traditional Linux kernel, CFS with appropriate nice values for each thread, reader thread disabled during synchronized operation and CPU frequency set to 1 GHz. Bus readings are performed inside interrupter thread routine when the controller is under synchronized operation.

Conclusion

Tests showed that BeagleBone Black can serve as a hardware platform for UVX local controllers. The new design has an acceptable performance during synchronized operation, comparable to that of the previous one.

References

[1] BeagleBoard.org, http://beagleboard.org/
[2] J. P. S. Martins et al., “Sirius control system: design, implementation strategy and measured performance”, in Proc. ICALEPCS’15, Melbourne, Australia, Oct. 2015, pp. 456-459.
[3] J. G. Franco et al., “LNLS control system”, in Proc. ICALEPCS’99, Trieste, Italy, Oct. 1999, pp. 651-653.
[4] J. G. R. S. Franco et al., “Upgrading the LNLS control system from a proprietary to a commercial communications environment”, in Proc. EPAC’04, Lucerne, Switzerland, Jul. 2004, pp. 530-532.
[5] R. Love, “Process scheduling”, in Linux Kernel Development: Addison-Wesley, 2010, pp. 64-65.
[6] Real-Time Linux, https://wiki.linuxfoundation.org/realtime/start