

GA-A24514

**SYSTEM DESIGN AND PERFORMANCE FOR THE
RECENT DIII-D NEUTRAL BEAM COMPUTER
UPGRADE**

by

J.C. PHILLIPS, B.G. PENAFLO, N.Q. PHAM,
and D.A. PIGLOWSKI

MARCH 2004

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

SYSTEM DESIGN AND PERFORMANCE FOR THE RECENT DIII-D NEUTRAL BEAM COMPUTER UPGRADE

by

J.C. PHILLIPS, B.G. PENAFLO, N.Q. PHAM,
and D.A. PIGLOWSKI

This is a preprint of a paper presented at the 20th IEEE/NPSS Symposium on Fusion Engineering, San Diego, California, October 14-17, 2003 and to be published in *Fusion Science and Technology*.

Work supported by
the U.S. Department of Energy
under Contract No. DE-AC03-99ER54463

GENERAL ATOMICS PROJECT 30033
MARCH 2004

System Design and Performance for the Recent DIII-D Neutral Beam Computer Upgrade

J.C. Phillips, B.G. Penaflor, N.Q. Pham, and D.A. Piglowski

General Atomics, P.O. Box 85608, San Diego, California 92186-5608

Abstract. This operating year marks an upgrade to the computer system charged with control and data acquisition for neutral beam injection system's heating at the DIII-D National Fusion Facility, funded by the U.S. Department of Energy and operated by General Atomics (GA). This upgrade represents the third and latest major revision to a system which has been in service over twenty years. The first control and data acquisition computers were four 16 bit "mini" computers running a proprietary operating system. Each of the four controlled two ion source over dedicated CAMAC highway. In a 1995 upgrade, the system evolved to be two 32 bit Motorola mini-computers running a version of UNIX. Each computer controlled four ion sources with two CAMAC highways per CPU. This latest upgrade builds on this same logical organization, but makes significant advances in cost, maintainability, and the degree to which the system is open to future modification. The new control and data acquisition system is formed of two 2 GHz Intel Pentium 4 based PC's, running the LINUX operating system. Each PC drives two CAMAC serial highways using a combination of Kinetic Systems PCI standard CAMAC Hardware Drivers and a low-level software driver written in-house expressly for this device. This paper discusses the overall system design and implementation detail, describing actual operating experience for the initial six months of operation.

I. INTRODUCTION

This latest upgrade of the neutral beam (NB) computers was crafted on time and on budget, using an optimized combination of in-house written applications and commercial packages. The systems are generic PC's, using the Intel Pentium 4 processor, clocked at 2.0 GHz. The OS is LINUX (Red Hat 7.2). Hardware is absolutely standard, with the exception of two specialized PCI bus cards, manufactured by Kinetic Systems, used to drive two serial data highways conforming to the CAMAC Standard. There are two identical NB computers, managing a total of four CAMAC Highways, eight neutral beam sources organized into four neutral beamlines.

Generating the graphical user interface (GUI) or screens for the new system was done using a commercial package, Kylix by Borland [1]. These screens are intimately bound to the run-time database for cyclical update of values displayed, and in an on-demand mode for asserting new set-points or the invocation of various processes or tasks. Fig. 1 shows operator top-level control for one of eight neutral beam systems. Each screen is compiled from source code.

II. HARDWARE

Connectivity for neutral beam computer system hardware is now the simplest it has ever been. As shown in Fig. 2, besides the essentials — power, keyboard, mouse, and monitor



Fig. 1. An example of a Kylix generated GUI — the beam operator's console.

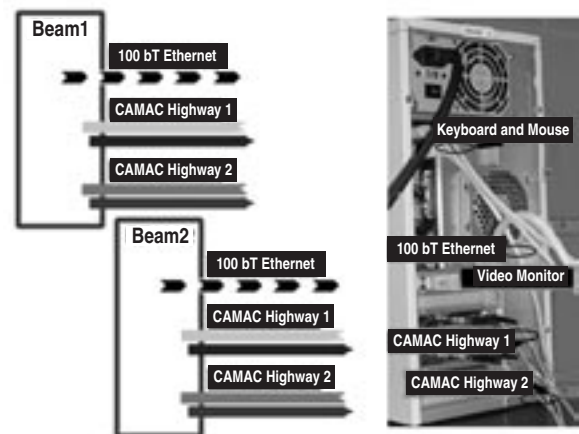


Fig. 2. Neutral beam computer system hardware — connectivity.

— each CPU has one 100bT Ethernet connection, and two fiber optic pairs, one for each of two CAMAC Highways.

The primary interface for each NB computer (Fig. 3), is a "3 Headed" X Terminal — The X Terminal is comprised of three 21" screens which form a continuous virtual desktop. In other words, the mouse may be moved continuously from the left edge of the leftmost monitor through the center screen, to the right edge of the rightmost monitor, as one virtual screen.

III. SYSTEM DESIGN

The basic design of the neutral beam computer systems is in common with the DIII-D tokamak operations computer, but with certain extended functionality. The NB computers each controls two CAMAC Highways, rather than one. The NB computer systems, in addition to controlling power supplies and shot sequencing, do data acquisition, (some 40 waveforms



Fig. 3. The operator interface — A three headed X-terminal

per system, 160 per computer). The tokamak operations computer does not acquire waveform data.

Much of the specialization for the NB systems is found in what's known as the POPS procedures. These are some 75 individual procedures written in "C," organized by function, ranging from simple (SCOPECLEAR) to complex (GETREADY). POPS procedures may call other POPS procedures, and the overall effect is a very modular and maintainable system. This is especially important for the NB systems. Consider for example one computer system preparing for NB injection in support of a DIII-D plasma discharge. At a given time there will be four instances of the POPS procedure "GETREADY" differentiated by the beam identity each represents. Likewise, four instances of codes to do data acquisition, four to set up timing, etc. Each instance acts independently on behalf of the beam identity originally invoking it.

An important part of the operational experience with a computer system is the ease or difficulty of maintenance. Maintaining these systems would truly be a nightmare if not for a few strictly enforced fundamentals. Starting with the GUI, there is only one "generic" control screen which when instantiated, adopts properties specific to each beam. These properties include screen location, beam identity (30L, 33R, 21R etc), as well as pointers to the run time database consistent with the beam identity. When a GUI's GETREADY button is pressed by the NB operator, the beam identity is passed with that event to the GETREADY POPS procedure, and to all procedures it may call. POPS procedures use their present beam identity to construct point-names for access of the shared memory/run-time database. This allows all POPS procedures to be generic — only one source code and one executable exists for GETREADY. While slightly more difficult to write, the result is far better than having to maintain four specialized copies for each NB Computer.

Advancing one step further, (remember there are two independent NB Computers) only one copy of source code exists for each POPS, and only one copy of source code for each operator screen or GUI. When compiled, a copy of the executable or screen is also moved to the second NB computer, where the same executable then stands ready to adopt a completely different set of beam identities. Key to this design is that all differentiation and specialization is achieved at run-time not at compile time. This ranks as one of the most important factors in the NB systems' ease of maintenance.

Key building blocks have been designed to make the overall system as efficient as possible, as outlined in Fig. 4. It's

easy to imagine a system becoming more and more loaded given dozens of procedures or codes making frequent and repetitive queries of the database (polling) while waiting for a specific set of conditions to occur. A key feature of the NB computer systems (as well as tokamak operations) is known as RULES, and serve to eliminate most instances of polling. RULES are designed to work efficiently with the cyclical scanning of the CAMAC I/O and shared memory database. The CAMAC/Database scanner operates on roughly a once per second refresh rate, though longer intervals may be specified for certain point-names. To do this, RULES declares in advance, database points which are "of interest" by setting a bit which is observed by the scanner. The work for the scanner at the margin is trivial. Whenever there is a change of state observed for a database point with a RULES bit set, the appropriate RULE code is invoked. The overall effect is a set of actions, efficiently event driven, implemented without the widespread use of polling by user codes. Importantly, these become a set of event-driven actions which are not tethered to the NB shot cycle or its procedures. For example a RULE named TIMEEDGE fires whenever the "trailing edge" for a given beam's pulse length is changed. The rule takes the new pulse duration and adjusts all supporting timing for filament, gas, arc, suppressor, etc., accordingly. Contained within the TIMEDGES rule are the methods giving the relationship between all timing signals. Just as with the GUI's and POPS, RULES are written to be generic, there is only one copy of a RULE's source code to maintain, beam identity is determined at run-time.

IV. EXPERIENCE

While there have been specific problems, for the most part the new NB computer systems are highly reliable and stable.

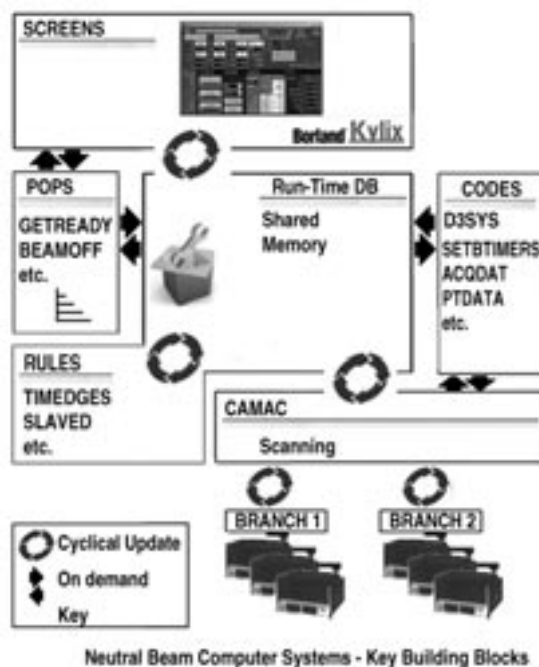


Fig. 4. Neutral beam computer systems—key building blocks.

It's not uncommon for BEAM1 and BEAM2 to have uptimes (time since last re-booted) of 65 days or more. Last revised dates for more than 90% of POPS procedures and RULES are from the year 2002. With the exception of the code which sets neutral beam timing (our timing system has undergone a major upgrade) the last revision dates for major DIII-D codes for data acquisition etc. ranges from six months to a year ago. Because the system is so stable and it is desired to keep it that way, code now runs each night and alerts the NB operators when they arrive in the morning with a list of any and all codes which have been changed since the previous day. Most days this is a null set, but in the event of any problems, inspecting the change list is now one of the first steps to troubleshooting.

The previous NB computers, doing essentially the same work, experienced CPU loading which remained at 50% in between shots (occupied with CAMAC Scanning and updates of the GUI's) then hitting 100% at the peak of a shot cycle and hanging at 100% for a period of minutes — clearly not a desirable situation. The new NB computers have CPU loads on the order of 4%–6% between shots, hitting peaks of 12% or so, given four beams in use for the shot.

The NB computers were the limiting factor, pacing the pulse repetition rate, while running asynchronous to DIII-D. The new NB computer systems are significantly faster, to the extent that now a count-down timer regulates the inter-shot interval to prevent exceeding duty-cycle limits for equipment like filament power supplies.

Neutral beams employ two byte-serial CAMAC Highways per computer, operating at 2.5 MHz. Because of isolation requirements the byte-serial highway is implemented with fiber optics and U-Port adapters for remote CAMAC Crates. While single, or, even repetitive small CAMAC transfers fare well, problems have been experienced problems

with occasional low level CAMAC errors during sustained high volume data transfers — (as when reading digitizer memory). Waveform acquisition performance has been brought into acceptable range with several work-arounds, including data transfer re-tries, and reduction in the size of individual transfers.

VI. CONCLUSION

Overall, the new Neutral Beam computer systems performance has been excellent. Screens remain responsive through the shot cycle, performance is robust and reliable. The systems have extremely high availability, and have proved easy to maintain. This third generation implementation passes the basic tests for having “got it right.” CPU loading is a quiescent 4%, rising to 12% during a shot cycle. This compares to statistics for the previous system of 50% quiescent, 100% during a shot cycle. It is now extremely rare for one of the NB computers to “crash,” and not uncommon to see uptimes of 30–60 days (time since last re-boot) for BEAM1 and BEAM2. For the first time ever the NB computers are not pacing the neutral beam shot cycle time. In fact new timer software has been installed to prevent firing too quickly in succession.

ACKNOWLEDGMENT

Work supported by U.S. Department of Energy Contract No. DE-AC03-99ER54463.

REFERENCES

- [1] N.Q. Pham, “DIII-D neutral beam control system for 2003 and beyond,” Presented at the 4th IAEA Tech. Mtg. on Control, Data Acquisition, and Remote Participation for Fusion Research, San Diego, California, (2003).