GA–A22455

REAL TIME SOFTWARE FOR THE CONTROL AND MONITORING OF DIII–D SYSTEM INTERLOCKS

by
J.D. BROESCH, B.G. PENAFLOR, R.M. COON, J.J. HARRIS,
and J.T. SCOVILLE

This is a preprint of a paper to be presented at the 19th Symposium on Fusion Technology September 16–20, 1996, Lisbon, Portugal and to be published in *The Proceedings.*

Work supported by
the U.S. Department of Energy
under Contract No. DE-AC03-89ER51114

[illegible]

GA PROJECT 3466
OCTOBER 1996

GENERAL ATOMICS

Real Time Software for the Control and Monitoring of DIII–D System Interlocks*

J.D. Broesch, B.G. Penaflor, R.M. Coon, J.J. Harris, J.T. Scoville

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

This paper describes the real time, multi-tasking, multi-user software and communications of the E-Power Supply System Integrated Controller (EPSSIC) for the DIII–D tokamak. EPSSIC performs the DIII–D system wide go/no-go determination for the plasma sequencing. This paper discusses the data module handling, task work load balancing, and communications requirements. Operational experience with the new EPSSIC and recent improvements to this system are also described.

1. BACKGROUND

Diverse control and instrumentation requirements of tokamaks must include robust, reliable, easily maintained, and expandable interlock and protection capability. Such systems must possess efficient and flexible communications capability. These systems are typically distributed through a variety of high EMF environments. They must operate in both cooperative and independent modes.

A variety of subsystems are used to meet these requirements for DIII–D: Programmable Logic Controllers (PLCs); high speed embedded controllers; high performance VME based systems using both real time and non-real time operating systems, and main-frame computers.

EPSSIC's function is to provide shot sequencing, control of the main power switches, and to act as a system wide safety interlock. EPSSIC monitors all key DIII–D sub-systems. If a subsystem abort is detected, EPSSIC generates a system abort and safely shuts down power to the tokamak.

The EPSSIC portion of the control system consists of a VME based multi-processor system employing both a PLC and a 68030 processor. The high EMF environments and vibration found in the E-Power Supply building dictated the use of non-disk based operating systems. Additional requirements included the need to communicate with a variety of network based applications, implement redundant hardware and software interlocks to provide a very high degree of reliability, and to be easily operated remotely by control room personnel.

2. OVERVIEW

Figure 1 shows a high level view of where EPSSIC fits into the overall DIII–D organization. One important category of inputs is the analog voltage commands received from the Plasma Control System (PCS) [1–3]. The PCS commands plasma position and configuration via these analog voltage commands. EPSSIC routes these commands to the appropriate controllers.

Figure 2 shows the details of the EPSSIC block shown in Fig. 1.

There are three main functional areas of EPSSIC: Logic Evaluation/Control; Communications; and hardware interlocking. The logic evaluation and control functions are principally handled by the PLC. Communication, which includes the user interface to the control room, is handled by the CPU. Commands to the high power switches are generated by the PLC based upon inputs from the rest of the DIII–D sub-systems. These commands are validated by the hardware interlock module before being passed on to the switches. Should an error in the program occur that would command the switches to fire at a potentially dangerous point in the cycle or to assume invalid states, the switches will instead be automatically placed in a safe condition and a system abort is generated.

The real time control software is organized around two key concepts. The first is an absolute

*Work supported by U.S. Department of Energy under Contract No. DE-AC03-89ER51114.

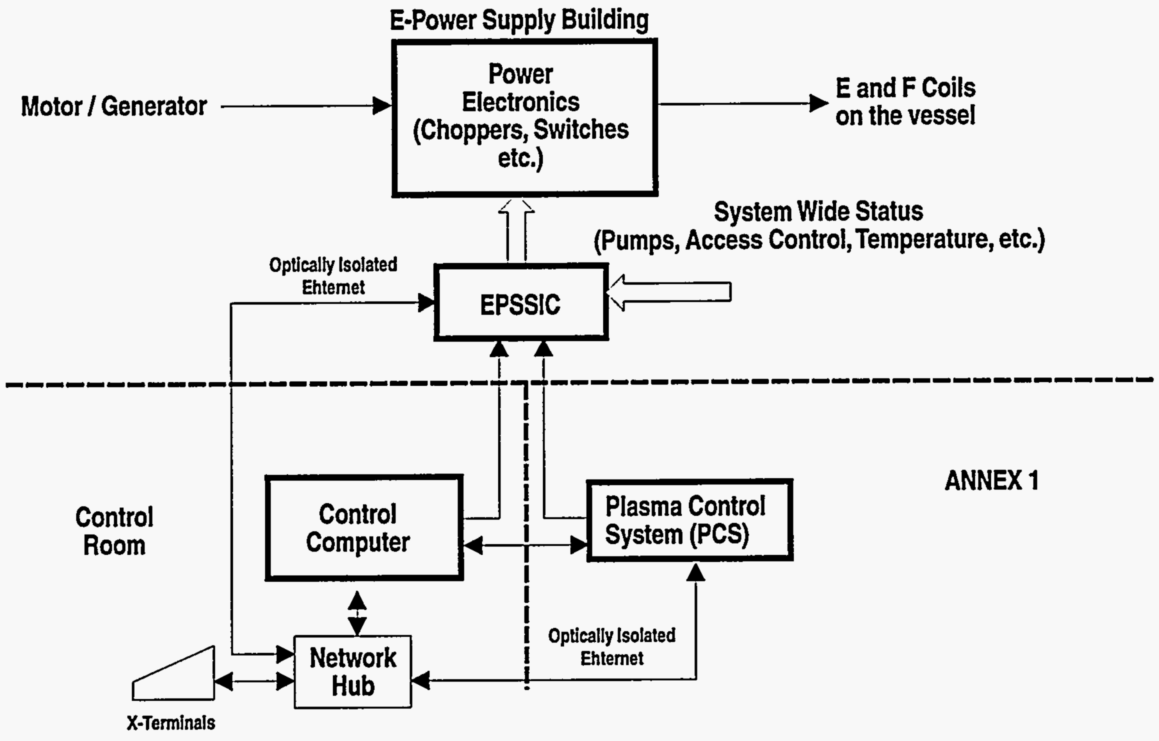
E-Power Supply Building
Motor / Generator
Power Electronics (Choppers, Switches etc.)
E and F Coils on the vessel
System Wide Status (Pumps, Access Control, Temperature, etc.)
Optically Isolated Ehternet
EPSSIC
ANNEX 1
Control Room
Control Computer
Plasma Control System (PCS)
Network Hub
Optically Isolated Ehternet
X-Terminals

Fig. 1. High level view showing EPSSIC.

addressed memory module. This module is shared by the PLC and the 68030, and is the key communications path for inter-processor communications. The second is a relative memory addressed data module that resides in the 68030 memory space. This second data module provides inter-process communications for the OS-9 based real time control, monitoring, and data communications tasks running on the 68030.

The communications implementation is based on a standard Berkeley Software Distribution (BSD) Sockets interface, with appropriate modifications to support the disk-less operation.

3. SOFTWARE ARCHITECTURE

During operations, the data in the real time module is monitored in the control room and from other diagnostics sites. Further, our plan calls for both user monitoring of the data via simple telnet sessions and for automated monitoring of the real time data via the control computer. These considerations clearly dictated a multi-tasking environment.

OS-9 possesses the standard IPC resources such as signals, pipes, etc. However, it also possesses a more powerful resource for real time data exchange: the data module.

Physically, a data module is a contiguous block of memory that is allocated by the operating system. Logically, it consists of a header, a data area, and a cyclical redundancy check (CRC) based checksum. In most practical applications the CRC is ignored. The header consists of information used by the operating system. The data area is user defined. In their use data modules are similar to memory allocations using malloc. Data modules posses a user defined name that is available system wide. Once a data module has been created, a pointer to its data area can be obtained by making a system call using the data module name as a parameter. In effect, this area of random access memory (RAM) becomes a global memory resource, the pointer to which is globally available to all tasks. While the ability to pass memory objects is seen in other operating systems (for example, the Microsoft Windows 95 ability to lock a memory object and pass a handle), the modular structure of OS-9 makes the use

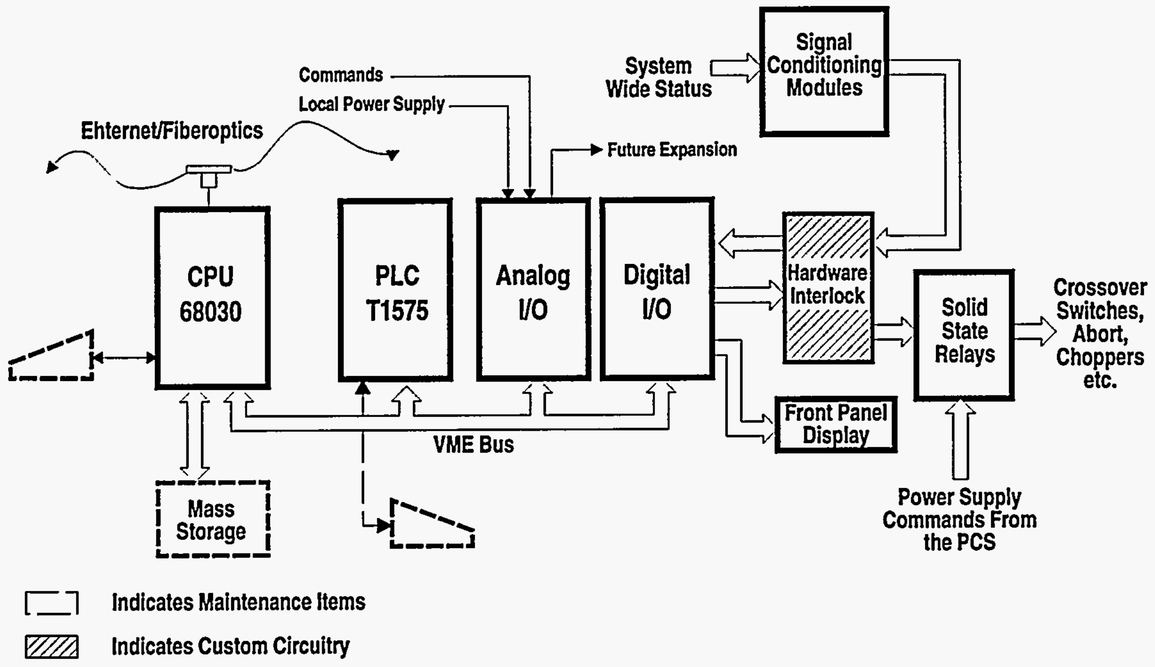

Fig. 2. Conceptual architecture for the updated EPPSIC.

and management of data modules particularly easy to use for real time data interchange.

The one disadvantage of this approach is due to the fact that pointers are used to access memory. No mechanism exists to strongly type check the variables among the various modules. It is the responsibility of the programmer, via the header file, to ensure variable coherency.

Figure 3 shows the high level software architecture. Note that daemons (in this paper, a daemon is defined as a console-less process) are used to perform the data updating between the data modules, I/O ports, and the PLC memory.Since this data is principally viewed by humans during and shortly after a shot, it was determined that a relatively long latency in updating the data module from the I/O and the PLC memory was acceptable. As general rule, 100 ms latency is nearly undetectable to a human operator. Therefore, the daemons are invoked periodically at a 50 ms rate, and the user interface routine is updated at a 50 ms rate. The worst case latency is therefore 100 ms and is sufficiently fast that it presents no significant delay in operator actions. This period is also sufficiently long enough that the system is not heavily loaded. Daemon execution time is approximately 2% of the overall CPU loading. The operator interface accounts for approximately another 10% of the processor bandwidth. This scheduling will be reviewed when the control computer interfaces are implemented; it is not anticipated, however, that a significant increase in the periodic execution rate will be required.

The actual scheduling of the daemons is accomplished by using the signaling capabilities of OS-9. Each daemon is set to respond to a signal. The system is then programmed to generate the signals periodically. At the end of each invocation each daemon voluntarily sleeps.

Due to the efficient inter-process communications, a number of users can be supported while ensuring a highly responsive user interface. The daemons have very little impact on the PLC ladder process accessing the I/O ports. This is important because the low latency allowed between inputs and outputs for certain key control events. Theoretically, the number of users is only limited by the maximum number of processes that the OS-9 system can support. In practice, we find that 68030 CPU can support three user logins executing the user interface routine without suffering any noticeable degradation in performance.

As noted above, approximately 12% of the processor bandwidth is used for user interface and daemons. Current loading analysis indicates that approximately 25% of the available CPU's cycles are being utilized. The other 13% appear to be system overhead, TCP/IP, etc. Empirical testing has indicated that with up to approximately 50% bandwidth . utilization no deterioration in responsiveness is detected by the operators.

4. FUTURE WORK

Much of the remaining CPU capacity is targeted for implementing a shadow control algorithm. This algorithm will monitor the control functions of the PLC, and if it detects a failure in the PLC, step in and perform a controlled abort sequence.

An additional design effort is to interface EPSSIC with the tokamak control computer via the TCP/IP protocol. This will allow the operators in the control room to view EPSSIC status from a common and uniform interface screen on the control computer.

5. CONCLUSIONS

The ability to monitor the various system status in the E-Power Supply from the control room has greatly improved the ability to diagnose system problems. This improved capability has meant less down time and a higher number of shots per operating cycle.

REFERENCES

1. G. Campbell, *et al.*, "New DIII–D Tokamak Plasma Control System," *Proc. 17th Symp. on Fusion Tech.*, (1992).
2. J.J. Harris, *et al.*, "A Combined PLC and CPU Approach to Multiprocessor Control," *Proc. Symp. on Fusion Engineering* (1995).
3. B.G. Penaflor, *et al.*, "A Structured Architecture for Advanced Plasma Control Experiments," this conf.

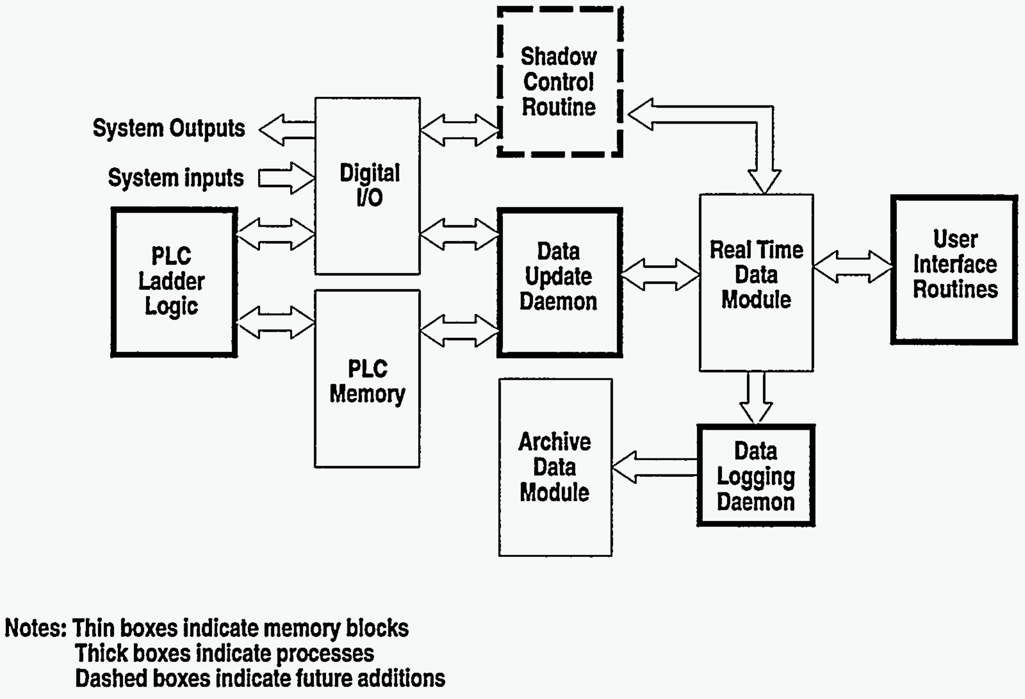

Fig. 3. EPSSIC software architecture.