CONF-831203--152

UCRL--89282

DE84 012588

A Stand Alone Computer System to Aid the Development of Mirror Fusion Test Facility RF Heating Systems

R. A. Thomas

Proceedings of the 10th Symposium
on Fusion Engineering
Philadelphia, PA
December 5-9, 1983

December 1, 1983

Lawrence Livermore National Laboratory

DISCLAIMER

A STAND ALONE COMPUTER SYSTEM TO AID THE DEVELOPMENT OF MIRROR FUSION TEST FACILITY RF HEATING SYSTEMS

R. A. Thomas
Lawrence Livermore National Laboratory
P.O. Box 5511, L-634
Livermore, CA 94550

Abstract

The Mirror Fusion Test Facility (MFTF-B) control system architecture requires the Supervisory Control and Diagnostic System (SCDS) to communicate with a LSI-11 Local Control Computer (LCC) that in turn communicates via a fiber optic link to CAMAC based control hardware located near the machine. In many cases, the control hardware is very complex and requires a sizable development effort prior to being integrated into the overall MFTF-B system. One such effort was the development of the Electron Cyclotron Resonance Heating (ECRH) system. It became clear that a stand alone computer system was needed to simulate the functions of SCDS.

This paper describes the hardware and software necessary to implement the SCDS Simulation Computer (SSC). It consists of a Digital Equipment Corporation (DEC) LSI-11 computer and a Winchester/Floppy disk operating under the DEC RT-11 operating system. All application software for MFTF-B is programmed in PASCAL, which allowed us to adapt procedures originally written for SCDS to the SSC. This nearly identical software interface means that software written during the equipment development will be useful to the SCDS programmers in the integration phase.

"Work performed under the auspices of the U. S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48."

Introduction

The ECRH system (1,2) an essential part of the MFTF-B, is configured into eight independent channels as shown in Figures 1 and 2. Each channel consists of a 200 kW gyrotron oscillator, an rf transmission system and a regulated source of gyrotron cathode, gun anode and filament power. Common to all channels are a transient compensating capacitor bank, an ignitron crowbar, an unregulated dc power supply, a 4000 gpm Low Conductivity Water-cooling (LCW) system, a 1500 kVA regulated 480 V substation, and computerized local or remote control. Each of the eight gyrotrons generates 200 kW of rf power at a fixed frequency of 28 GHz, 35 GHz, or 56 GHz for a maximum pulse duration of 30 seconds repeated at five-minute intervals. The system also supports high rep-rate short pulse operation for gyrotron conditioning.

Control Architecture

The overall control system as shown in Figure 3 indicates that ECRH uses nine LCC's connected by fiber optics to CAMAC crates located in the associated hardware in the MFTF-B Building. The SSC would normally be located in the same building as the SCDS and the LCC's, but due to not having the fiber optic runs installed yet, we are operating the prototype system completely by local control in the MFTF-B Building. This means that both the LCC and SSC are located in the immediate vicinity of the ECRH hardware. This is satisfactory now, but when the large MFTF-B magnets are turned on we will have approximately a 50-gauss field in the area which precludes operating a disk or a conventional CRT.

Due to the complexities of the overall system, it was decided that the initial SSC system would just plug in (in place of SCDS) to a particular LCC that requires local control, rather than develop initially some sort of multi-port approach. This decision made it possible to develop a stand-alone system quickly enough to be useful in the ECRH development. This system, however, was not available when the contract was let for the construction of the ECRH power supply system. It can only be controlled by a computer, and was therefore built using a single LSI-11 configuration which is not compatible with our SCDS/LCC architecture. The software that was developed will only be useful for acceptance testing and temporary use with the prototype gyrotron socket.

The stand-alone system hardware is shown in Figure 4. The 9600 baud line between processors is exactly compatible with the SCDS system, therefore the LCC is unaware of what is controlling it. The SSC is equipped with a disk and printer to make it useful as a complete development system. DEC RT-11SJ operating system and Oregon Software Pascal compiler provide the software tools.

The LCC

Some background will be given on the LCC in order to understand the function of the SSC. A relatively complete treatment of the LCC system and software is given in Reference 3. Here, I will describe some of the functions of PLEX, the software that operates the LCC.

PLEX can perform a limited number of functions (called *primitives*) in the hardware. There are presently 23 primitives. Primitives such as OPEN a valve, CLOSE a valve, and READ a valve directly control hardware. CONFIGURE a device, IGNORE a monitor build/maintain internal data tables. WHEN and START control command execution.

The basic transaction with PLEX is a command that consists of a string of primitives. Commands begin execution in the order they are received and most commands execute to completion immediately. There are a number of special command constructs that will allow for delays and looping. A message is always returned from PLEX to acknowledge the message and report any requested information.

By means of a process called configuration, PLEX is informed of the location and type of physical hardware under control of the LCC. The unique identifier of a device is a 32-bit DeviceID. It is possible to configure a device to a single bit of a register, thus allowing a single closure to be a device. A device can also create a vector of data such as would be obtained from a memory device. Internal procedures in PLEX called sequence codes provide the hardware specific instructions that are specified in a particular device configuration.

One of the primary functions of PLEX is to provide monitoring of configured devices. Two types of monitoring are possible: Discrete--Status which is state like such as valve positions and Analog--A quantization of an analog value such as an ADC reading. PLEX constantly polls the devices it was instructed to monitor and compares their present value with the most recent value that it has reported. Whenever a "significant" change occurs, a report

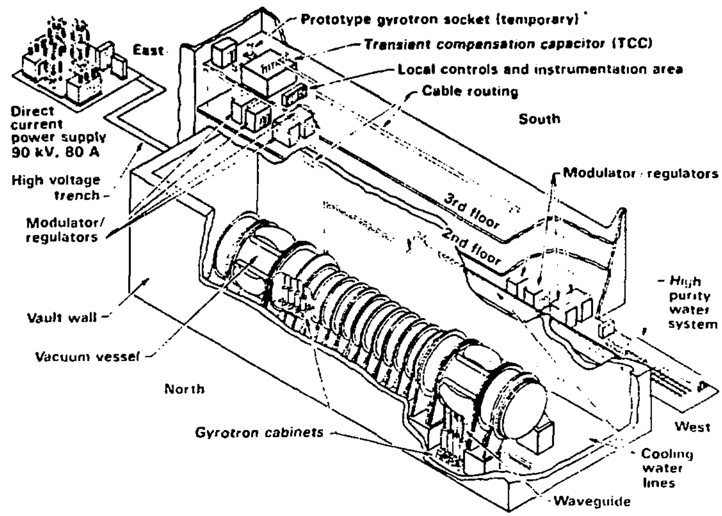

Figure 1.

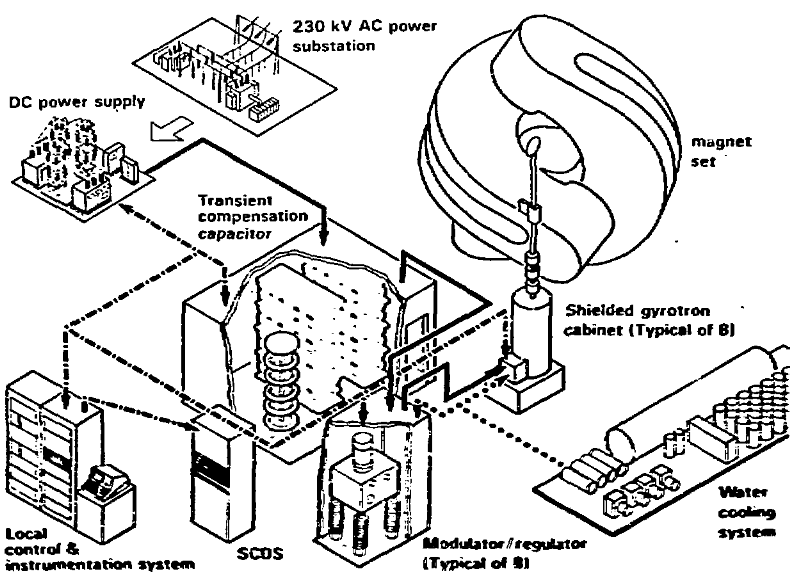

Figure 2.

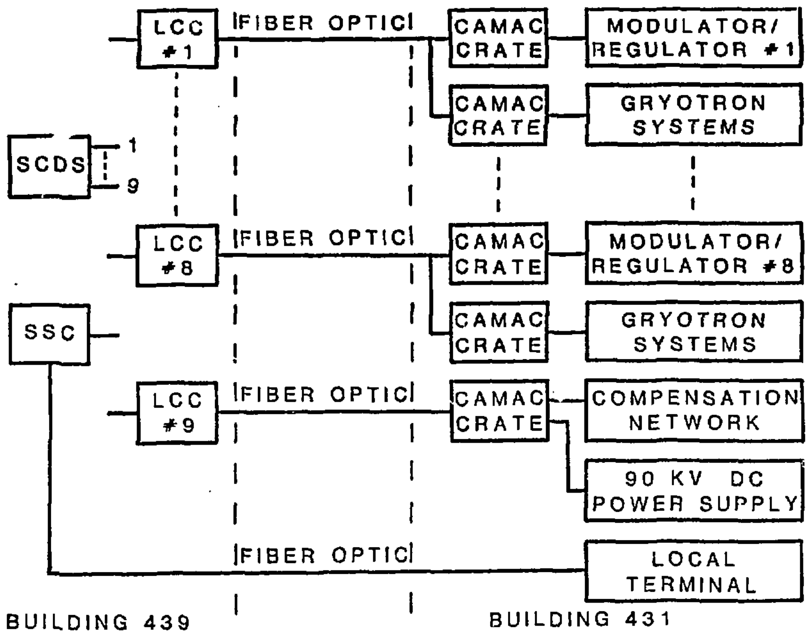

Figure 3.

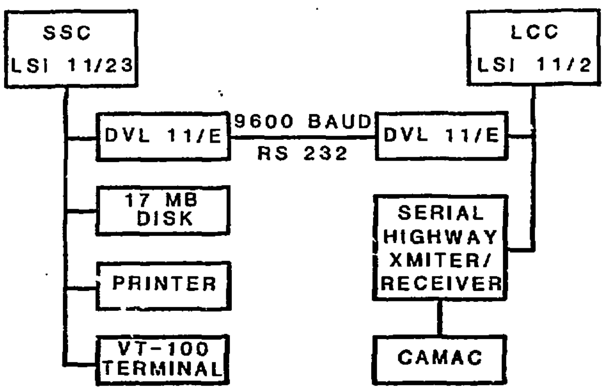

Figure 4.

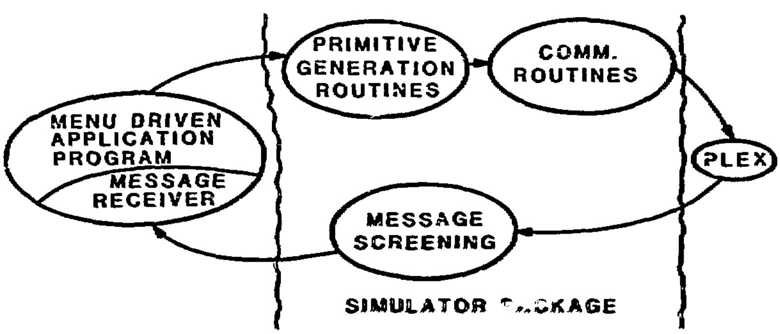

Figure 5.

is sent to the host computer. This monitor message is the primary means of updating the data base in the host.

The SSC

The primary goal in the development of the software for the SSC was to provide a software interface with PLEX that was as identical as possible to that provided in SCDS. With Pascal being the common language, it was possible to use all the primitive generation routines written for SCDS only making changes necessary to accommodate the differences between a 32-bit and 16-bit computer; i.e., the 32-bit device ID type was changed to a packed array of four 8-bit bytes, etc. Communication routines, obtained from a previously written simulator (4), were interfaced by a message sending procedure that serves a function similar to a call to the LCCManager routine in SCDS.

Figure 5 shows the main software components in the system. The simulator package is compiled separately and stored as an object module that is linked with the compiled application program. A single global type and external file is used with all software and provides the software interface with the simulator. The application program must provide a procedure to receive the messages from PLEX. These messages serve to update the data base and notify the receiving program that a change has taken place.

A basic application program was written to provide a direct contact with PLEX. The following items were included in the main menu:

1. Set an analog monitor
2. Set a discrete monitor
3. Ignore a monitor
4. Direct CNAFV to CAMAC
5. Configure a device
6. Configure devices from an external file
7. Write a 24-bit value to a configured device
8. Read a 24-bit value from a configured device

More menus may be added to control the operation of complex components such as the ECRH power supply system, gyrotron systems, etc.

The following segment of Pascal code illustrates the creation of a message to PLEX that will open two valves, close one valve, set a 24-bit value in a register, and send the message to PLEX:

```
procedure Operate_On_Camac;

type
 Bit8 = 0 . . 255;
 Plex_Primitive_String_Type = record
    Text_Length : 0 . . 250;
    Next_Free_Byte : 1 . . 250;
    Overflow : boolean;
    Text : packed array [1 . . 250] of Bit8;
 end;
 Plex_Dev_ID_Type = packed array [1 . . 4] of Bit8;

var
 Com_Text : Plex_Primitive_String_Type;
 Valve_1, Valve_2, Valve_3, Register_1 : Plex_Dev_ID_Type;
 Message_ID : integer;

begin
 INIT(Com_Text);
 OPEN(Valve_1, Com_Text);
 OPEN(Valve_2, Com_Text);
 CLOSE(Valve_3, Com_Text);
 SET(Register_1, Com_Text);
 SENDMsg(Message_ID, Com_Text);
end;
```

The function of the primitive generation routines is to put the correct token and other required information in the text string. When the text string is complete, the message is sent to PLEX. A completion message is always returned from PLEX. If no error conditions are set, the message is discarded in he simulator. If any error conditions exist, a report is printed on he screen from a routine in the simulator. Only monitor messages and messages containing requested data are passed back to the application program.

A separate program is available that down-loads the PLEX program to the LCC. After PLEX is down-loaded, the application program is run. Configuration files may be loaded with one of the main menu selections, and equipment operation initiated by entry to other menus.

ECRH Application

After the simulator package was completed, the first application was the control of the 90 kV, 90 A power supply used by ECRH. This was a relatively simple application and readily proved the utility of the system by allowing special tests to be quickly written and tried. The control consisted of a real-time status display and the ability to turn the power supply on and off and to change the tap-settings to selected values.

The next control application is the gyrotron systems. This consists of the set points for nine magnet supplies and a status display of all the associated systems such as LCW and FC75. Due to the critical nature of the gyrotron settings, hardware bounds checking is employed to minimize the change of improper settings. All of the status bits are inputted to in-house designed interlock modules to provide an overall permissive to turn high voltage on to the gyrotron.

The prototype ECRH modulator/regulator and capacitor bank is now installed and operating. Local control software will be written using the simulator to allow a SCDS compatible configuration and aid in the integration of this system with SCDS. Figure 6 shows the local control area of the prototype gyrotron system. Note that two terminals are involved, one operating the modulator/regulator and one operating the HV power supply and the gyrotron systems.

The simulator has proven valuable for the generation and test of the device configurations. The configuration files provide a good interface point with the SCDS programmers by providing the information needed to create a device tree. Another invaluable service provided by the simulator is in the development and testing of sequence codes for PLEX.

Other Applications

Large subsystems developed at other locations, away from Lawrence Livermore National Laboratory (LLNL), can use the simulator to develop software that will both be useful for local control at LLNL and also provide easy communication of the operation to the SCDS programmers. Figure 7 shows a LCC/SSC system assembled for TRW to develop the controls for Ion Cyclotron Resonance Heating (ICRH). Similar systems are being supplied for development of other MFTF-B subsystems at TRW.

Conclusion

A stand-alone development system is desirable for developing hardware in complex systems that can only be controlled by computers. The fact that the various pieces of the stand-alone system can be located nearly anywhere allows us the flexibility to coordinate our efforts with the MFTF-B construction schedule.

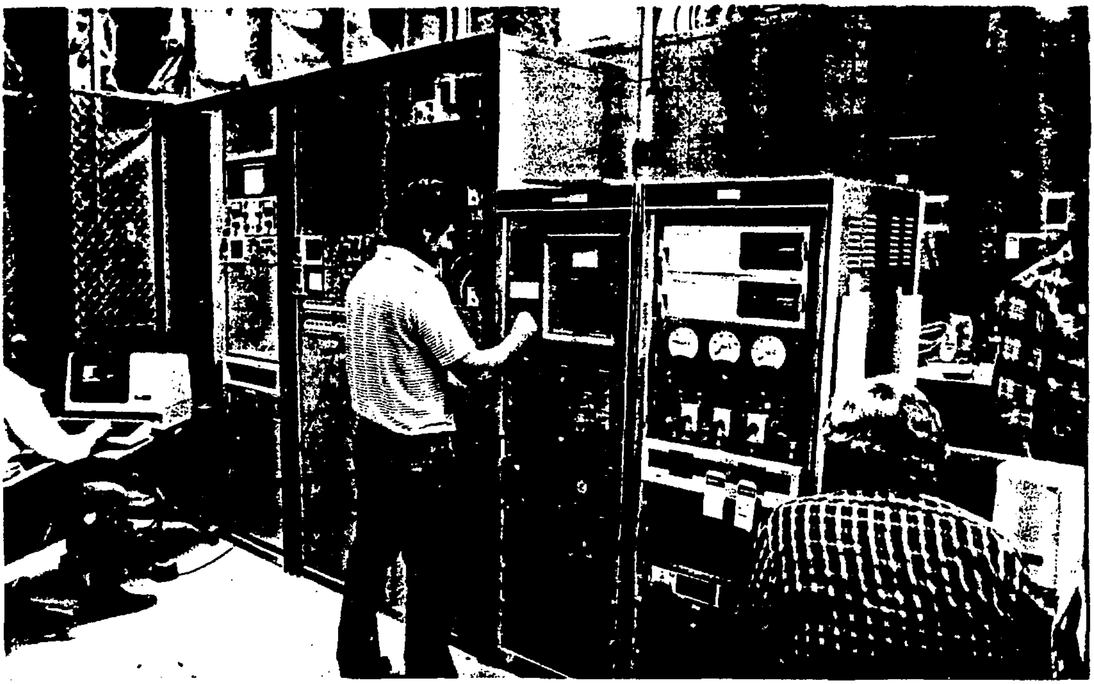

Figure 6

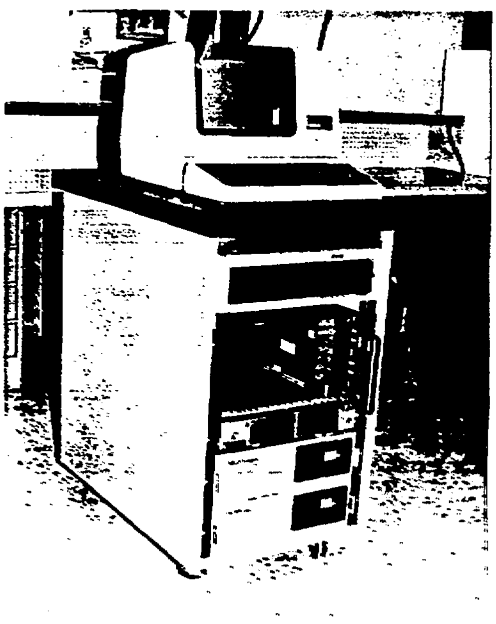

Figure 7

References

1. Krause, K. H., Pollock, G. G., Yugo, J. J. "The MFTF-B Electron Cyclotron Resonance Heating System", Proc. 9th Symp. on Eng. Prob. of Fusion Research, IEEE Publ., (1981).

2. Krause, K. H., "MFTF-B ECRH System Construction Progress and Preliminary Test Results", Proc. 10th Symp. on Eng. Prob. of Fusion Research, IEEE Publ., (1983).

3. Labiak, W. G., "Software for the Local Control and Instrumentation System for MFTF", Proc. 8th Symp. on Eng. Prob. of Fusion Research, IEEE Publ., (1979)

4. Minor, E. G., Labiak, W. G., "Tools and Methods for Implementing the Control Systems on the Mirror Fusion Test Facility", Proc. 9th Symp. on Eng. Prob. of Fusion Research, IEEE Publ., (1981).