# A poloidal field control system based on VxWorks in HT-7 Tokamak

WANG Feng, LUO Jia-Rong, WANG Hua-Zhong

*(Institute of Plasma Physics, the Chinese Academy of Sciences, Hefei 230031)*

**Abstract** The PF (Poloidal Field) control system is one of the most important control systems in HT-7 Tokamak. Most of parameters such as plasma current, plasma horizontal position and plasma vertical position will be monitored. For the purpose of long-pulse discharge and the more precise control to plasma, the real-time operation system VxWorks is applied, instead of the behindhand and unbefitting DOS operation system. This paper describes the development process of HT-7 PF control system based on VxWorks on Intel X86 platforms. The method of using hardware cards in VxWorks, and the network communication with other operation systems are discussed especially. Results of the comparison between VxWorks and DOS operation systems are given too.

**Keywords** Tokamak, Poloidal Field (PF) control system, VxWorks

**CLC numbers** TP316, TL631.2+4

## 1 Introduction

HT-7 is the first superconducting Tokamak device for fusion research in China. Many experiments have been completed in it since 1994, and a lot of satisfactory results have been obtained in the fusion research field on HT-7 Tokamak.[1] The control system of HT7 is designed as a distributed control system (HT-7DCS), including many subsystems. The poloidal field control system (PFCS) is one of the key subsystems for the HT7-DCS. Because the constraint conditions required by nuclear fusion are very rigorous, most of current Tokamak devices, such as HT-7, can only realize the pulse type, but not the steady state, plasma discharges. Generally plasma discharges may merely last dozens of milliseconds to several seconds each time, and they are too fast and vertiginous to control, so a real-time and precise control is highly expected. The PFCS achieves the plasma discharges mainly by changing the trigger-phase of six phase thyristor bridges and converting the voltages from AC to DC. The maximal time delay of changing trigger-phase is not more than one millisecond, so each control cycle must be limited in one millisecond. There are many steps in one control cycle, including signal input, data acquisition and pretreatment, complicated compute, signal output, etc, therefore it is very difficult to design the control system. Even if the error in each cycle is several nanoseconds, the accumulative error in one plasma discharge will be considerable, and the discharge will be killed quickly by the control system while it should be running.

The structure of PFCS is shown in Fig.1. It mainly consists of seven parts:[2] PF Server computer, programmable logic controller (PLC), $\alpha$ angle trigger controller, multivariable feedback controller, plasma vertical feedback controller, engineering data acquisition system and ohmic field voltage setup system single chip computer. Its control objects are the thyristor converter trigger and more then 200 switches of five fields, i.e. vertical field, ohmic heating field, horizontal field, biased field and vertical correcting field, which interlock in time order. The PF Server sends out control commands and presets experimental parameters. The time order signals from PLC are used to control the switches of PF main circle and the time process of feedback loop. The PS trigger sends out trigger signals of the thyristor converters in order to control current over magnetic field coils.[3] PFCS controls the procedure of plasma discharge and current of poloidal

Received date: 2003-07-08

field coils by modified PS output voltage. The main functions of these controllers are as follows:

(1) Multi-variable feedback controller acquires the magnetic data and software decouples the algorithm and calculates the real time value of $\alpha$ angle every 1ms for the $\alpha$ angle trigger controller.[4]

(2) $\alpha$ angle trigger controller gets the value of $\alpha$ angle through the parallel interface from the multi-variable feedback controller; detects frequency of the AC power; transforms the data into the $\alpha$ angle and sets timer trigger to the phase-controlled thyristor rectifiers which drive current through the OH and EF coils.

(3) Plasma vertical feedback controller acquires the magnetic data and transforms the calculation results into analog quantity and sets to the phase-controlled thyristor rectifiers that drive current through the HF coils.

Industrial computers of X86 platforms with MS-DOS OS (operation system) were used before the end of 1999. All the programs were developed with C/C++ Languages. Though it worked well in the past few years, the old PFCS was not fit for the development of HT-7. It has reached considerable capacity and is close upon its limit for expandability, so an upgrade must be done to PFCS.

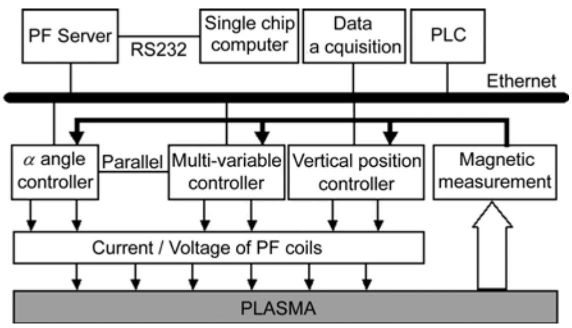


**Fig.1** Structure of the poloidal field control system

## 2 Motivations and goals of the upgrade

As a very old and laggard operation system, the MS-DOS OS has many innate limitations. It is very difficult for developer to make programs, in network, multi-task, debugging and so on. Especially, the memory management of MS-DOS is extremely poor, the conventional memory is only 640K and the array size cannot be larger than data segment size 64K usually, it is not easy and convenient to use a large number of memories. With the expansion of the pulse length of HT-7, the array size is growing fast, and more memories are needed.

Learning from the pitfalls of the previous control system, a key step is to improve the memory management to provide a very long time control of plasma discharge. Large memories can be got through the method of extending memory or some other third party tools in MS-DOS, but the time limit of access memory cannot be ensured. So a new OS must replace the old one.

There are many RTOS (Real-Time Operation Systems), such as VxWorks, QNX, VRTX pSOSystem, and so on. VxWorks is the most widely adopted RTOS. The VxWorks RTOS has a high-performance wind microkernel, that supports a full range of real-time features including fast multitasking; interrupt support, and both preemptive and round-robin scheduling. The microkernel design minimizes system overhead and enables fast, deterministic response to external events. VxWorks includes some other important features: high scalability, advanced networking support, powerful files and I/O system, 32 bit flat memory management etc. Furthermore, the cross development IDE called Tornado II along with VxWorks is very powerful and efficient but easy to use. In a word, the VxWorks RTOS could be applied in PFCS instead of MS-DOS OS, so the upgrade of PFCS has focused on building an efficient and reliable real-time control system using VxWorks.

## 3 Software implementation

Since the control theory of PFCS is mature, the working mode of all the controllers needn't be changed. The main task is to transform old OS of the three feedback controllers into VxWorks, and to transplant the programs to new version of VxWorks. There are many differences between VxWorks and DOS, some problems should be solved, including building up the cross development environment and knowing how to make it working, the real-time analysis, arrangement of multiple tasks, writing drivers of the hardware that are not supported by VxWorks default, and network communication with other computers. The primary module diagram is shown in Fig.2 and the central flow chart is shown in Fig.3.

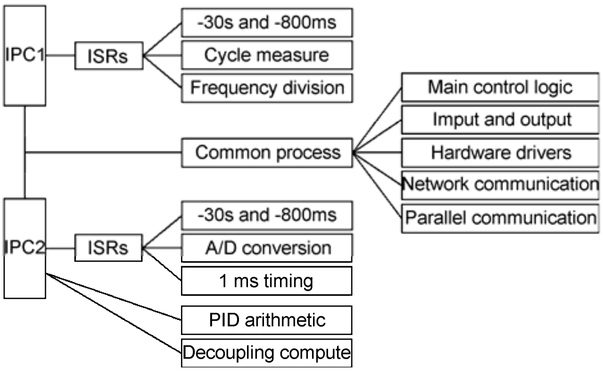


**Fig.2** The primary module diagram

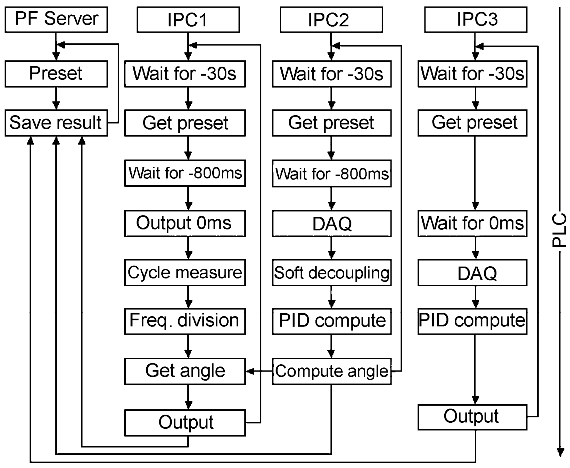


**Fig.3** The central flow chart

### 3.1 Cross-development

Tornado is an integrated environment for software cross-development. It provides an efficient way to develop real-time and embedded applications with minimal intrusion on the target system. The Tornado environment is designed to provide this full range of features no matter whether the target is resource-rich or resource-constrained. Tornado facilities execute primarily on a host system, with shared access to a host-based dynamic linker and symbol table for a remote target system. The relationship between the principal interactive host components of Tornado and the target system[5] is shown in Fig.4. Communication between the host tools and VxWorks is mediated by the target server and target agent. With Tornado, the cycle between developing an idea and the opportunity to observe its implementation is extremely short. Fast incremental downloads of application code are linked dynamically with the VxWorks OS and are available for symbolic interaction with minimal delay.

The cross-development environment should be set up before the application of VxWorks. The primary steps are as follows: making boot disk; scaling VxWorks; target setup; host setup; booting target and attaching to target. The PF Server computer is the host computer, where Tornado is installed. The three feedback controllers are targets. Networking connects the host and the three targets. In respect that target agent and target server are one-to-one, three target servers can be running in PF Server computer when the application is being cross-developed. Developing and debugging on the host, downloading dynamic to the target, and running on the target are the main idea of cross-development.

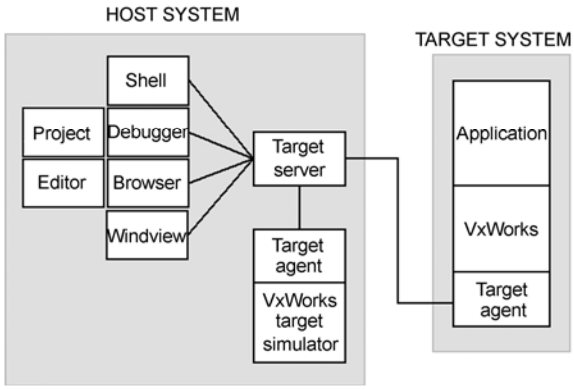


**Fig.4** Tornado development environment

### 3.2 Real-time analysis

The real-time capacity must be considered when developing a real-time control system. It is also very important to the PFCS that controls the plasma parameters. When the high-performance microkernel wind is running, a context switch only spends 3.8 μs, and the latency time of interrupt is less than 3 μs. In PFCS, the interrupt method is adopted to response to mostly external events, so the performance of real-time ought to be ensured and obtained.

There are different kinds of way to measure and detect the real-time capability in details, including entire hardware, entire software, part software and hardware. As software ways are much convenient and fast, they are generally utilized. Two kinds of software means have been adopted in the developing of PFCS. Firstly, VxWorks provides two functions to test the execution time. The function timex() times a single execution of a function or functions, and timexN() times repeated executions of a function or group of functions. This way is static and not very accurate. Secondly, a powerful tool, WindView acts as a logic analyzer for real-time system, which can provide a dynamic situation of a system. It allows developers to study dynamic interactions of all elements of the complex system.

### 3.3 Hardware drivers

VxWorks is supported by many hardware, it is easy for developers to make drivers for such hardware. However, there are more than ten custom cards in PFCS, of which the drivers should be developed by the users, and which are divided into three kinds, A/D, I/O and Timer/Counter. All these cards based on ISA bus, but not PCI bus, could be used for the following reasons. First, it is enough for the need of PFCS with ISA cards. Next, development of ISA cards is much easier than that of PCI. Last, it is too expensive to buy so many PCI cards.

Developing drivers for ISA cards is similar to that in DOS, which mainly includes three steps. First of all, the resources used by these cards should be adapted for I/O mapped address and IRQ number, and generally they are static and fixed with the jumpers on card. Secondly, with some low level functions, the I/O address should be operated. VxWorks has provided a number of functions to access the I/O address, such as sysInByte(), sysInWord(), sysOutByte(), sysOutWord() and so on. At last, it is very important to write effective interrupt service routines for interrupt which response to external events. Hardware interrupt handling has a key significance in real-time system, because it is usually through interrupts that the system is informed of external events. For the fastest possible response to interrupts, interrupt service routines (ISRs) in VxWorks run in a special context outside of any task's context. Thus, interrupt handling involves no task context switch. The Interrupt Descriptor Table (IDT) occupies the address range 0x0 to 0x800. Vector numbers 0x0 to 0x1f are handled by the default exception handler. Vector numbers 0x20 to 0xff are handled by the default interrupt handler. By default, vector numbers 0x20 to 0x2f are mapped to IRQ levels 0 to 15, which are normally used in Intel X86 Architecture computers. To redefine the base address, edit sysVectorIRQ0 in sysLib.c.[5]

### 3.4 Network communication

All the controllers are not isolated and they have been connected to the control network of HT-7. So the network communication with each other is absolutely necessary. The standard VxWorks network stack supports the following protocols and utilities: SLIP, PPP, TCP, UDP, IP, FTP, RSH, RPC, telnet and Sockets etc. VxWorks provides standard BSD socket calls, which allow real-time VxWorks tasks and other processes to communicate in any combination with each other over the network. There are three types of network communication about VxWorks among those computers in PFCS, which include:

(1) Communication with Windows. The three controllers receive preset data from PF Server 30 ms before discharge, and on the other hand, they transfer the result data back to PF Server after the discharge finished.

(2) Communication with Linux. The three controllers have to request the ShotNo of every discharge from ShotNo Server which is running redhat7.3 Linux OS.

(3) Communication with VxWorks. The three controllers are compact and they are directly related by parallel interface, and besides, there are some handshakes and communications with each other which make use of network.

In order to realize these various network communications, socket must be used which is the standard API of TCP/IP. The primary method-using socket in VxWorks is very similar to that in Linux.

## 4 Summary

The new PFCS based on VxWorks RTOS had been successfully applied to the 2002~2003 physical experiment of HT-7. It is more effective for real-time control than the old one during the plasma discharge. Excellent performance has been seen in HT-7. The longest shot is shown in Fig.5. More than 60 second discharge's control has been implemented by this real-time control system, yet this effect has never been assumed and achieved with the old control system.

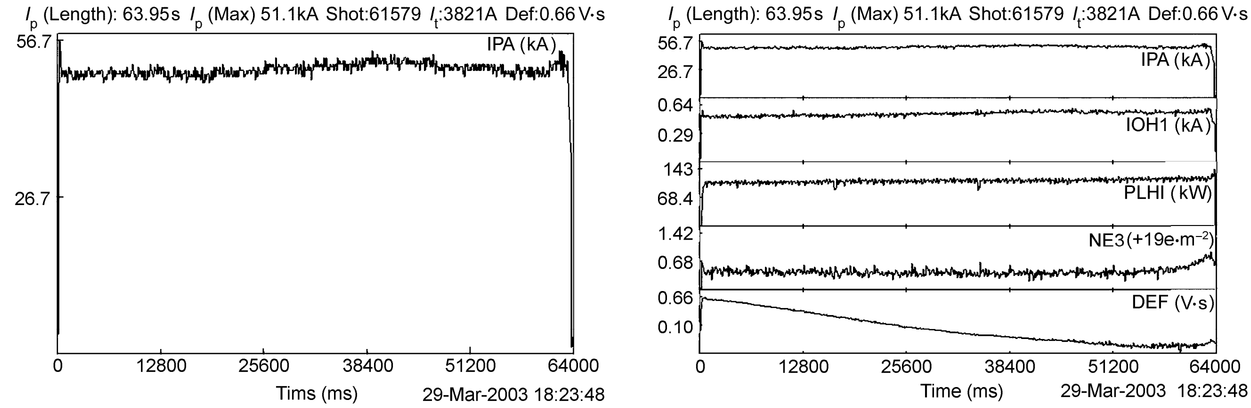


**Fig.5** Typical waveform of the longest shot

## Reference

1 XIE J K, HT-7 group. in Plasma physics and controlled nuclear fusion research 1996 (Proc. $16^{th}$ Int. Conf. Montreal, 1996), IAEA, Vienna, 1997, **1**: 685

2 WANG L, Luo J R, YING F X, *et al*. Plasma Science & Technology, 2000, **2** (3): 311

3 LUO J R, HT-7 Poloidal Field Control Group. Plasma Science & Technology, 2000, **2** (1): 111

4 FU P, LIU Z Z, LUO J R, *et al*. Acta Physica Sinica (in Chinese), 1999, **48** (4): 685

5 WindRiver, Tornado Online Manuals