CONTROL FUNCTIONS OF CENTUM CS 1000

ODA Shinji [*1] *NISHIDA Jun* [*1]

We have developed the basic control functions for the field control stations (FCSs) of the CENTUM CS 1000, a distributed control system for medium- and small-scale plants. The acclaimed control functions of the CENTUM CS as well as enhanced functions for communicating with various subsystems such as programmable logic controllers are incorporated into one compact unit, achieving a system that is both highly functional and maintainable. The virtual test function, which was also newly developed for the CENTUM CS 1000, allows virtual field control stations to be created in a personal computer and control actions to be fully simulated and tested without actual FCS hardware.

INTRODUCTION

It goes without saying that the reduction of the total cost of ownership (TCO) for various facilities and equipment is an important objective for most enterprises, and distributed control systems (DCSs) are no exception. When discussing the TCO of a DCS, the primary concern seems to be the hardware maintenance costs; however, the cost of developing and maintaining the control applications running on the DCS cannot be overlooked either.

Secondly, it is more important than ever before to be able to network with ever-diversified field control equipment when carrying out the plant control. This report introduces the control functions of the CENTUM CS 1000 while focusing on the following:

- How the engineering required for system creation and system modification is made easy
- How communication with the various field control equipment is made flexible

*1 Industrial Automation Systems Business Division

FUNCTION CONFIGURATION

The CENTUM CS 1000 incorporates all of the extensive and flexible control functions that are so highly regarded in the CENTUM CS, in order to achieve a system for medium- and small-scale plants. Figure 1 shows the configuration. These functions are built into each field control station (FCS). Basic control applications are built by assembling software components called function blocks which constitute the minimum unit of control algorithms. These include such components as PID controllers and selectors. By grouping the sets of function blocks for each process unit and using the unit supervision function, the control and monitoring operation of each process unit can be drastically simplified.

As for the inputs and outputs, the process I/O functions for inputting and outputting signals from/to general sensors and actuators, as well as the communication I/O functions for inputting and outputting the signals via general-purpose communication such as RS-232C, are included.

All these control functions can be run on a personal computer using an engineering tool provided in the FCS simulator functions. The user can verify the actions of control applications on a personal computer using the FCS simulator and then install them as is to an actual FCS.

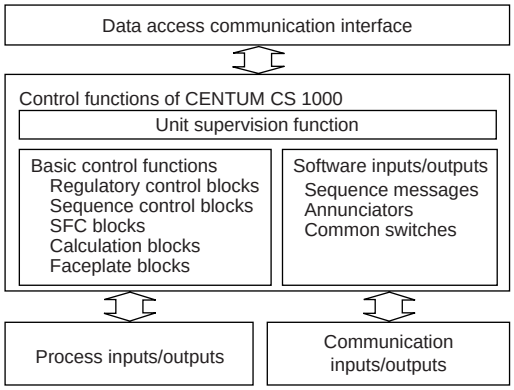

Figure 1 Control Functions of CENTUM CS 1000

Reliability

The synchronous hot stand-by system[*2] employed in the operating system of the CENTUM CS's field-proven FCSs is also employed in the FCSs of the CENTUM CS 1000. Hence, in the event that the CPU fails during operation due to an unexpected error, control is switched over to the stand-by CPU without interrupting the control actions. This is a great merit for sequence control-oriented batch processes because control is always maintained even when switching between duplexed control processors.

Improved Efficiency in Engineering by FCS Templates

Several measures were taken to improve the efficiency of CENTUM CS 1000 system engineering. Such measures include the preparation of a template for each type of application.

Various function block models are combined to build up the control applications in an FCS. These function block models require different internal resources and CPU load; even though the memory available in an FCS is limited. If the combination of function block modules that can be used in an FCS is not restricted, the function blocks and other resources can be arranged so as to achieve optimum effectiveness within this limited memory in an FCS. This requires a great deal of skill and experience however, as the resources to be used and the total CPU load of the arrangement must first be estimated.

To simplify the engineering work, FCS templates are provided for the user's expected application — whether that be regulatory control, sequence control, or monitoring. In these database-type templates, the database of an FCS is appropriately allocated to each block type and the other resources required for the user's application without exceeding the CPU load limitation.

Field Communication

Not only is a variety of control equipment scattered throughout the field, but the types of data and communication interfaces for field control equipment also vary a great deal. If a control application needs to be changed according to the type of data and communication interface of the subsystem, the engineering for system configuration and modification would be extremely difficult. In the CENTUM CS 1000, this issue is solved by the following measures:

(1) Separation of Communication Function from Control Application
In the control function of the CENTUM CS 1000, data fetched via communication are stored in an area referred to as the communication I/O area. Control blocks and other resources in the control application access this communication I/O area in order to exchange data with field control equipment. When accessing each data item in the communication I/O, the relative address inside the area or the user-defined label is specified. The data type (integer, floating-point value, etc.) of each data item in the area is automatically identified so that the data type does not need to be considered when configuring a control application. Since the communication function is functionally separate from the control application, the control application need not be modified when there is a change in the communication protocol.

(2) Use of C Language for Coding Communication Programs
Field control equipment can be connected to an FCS of the CENTUM CS 1000 using various general-purpose communication interfaces; however, it may require different communication protocol. Communication packages have been developed for popular protocols; while other protocols can be coded in C language. Refer to Figure 2 for the flow of communication data.

FCS SIMULATOR FUNCTION

The reduction in engineering work required for a DCS and cost of preparing the engineering environment is of great merit for users. An FCS simulator function that runs under the Windows NT[*3] operating system has been developed to enable DCS engineering, including the testing and inspection of control applications, to be accomplished using just a personal computer. Figure 3 illustrates the relation between the FCS simulator and other functions in a personal computer.

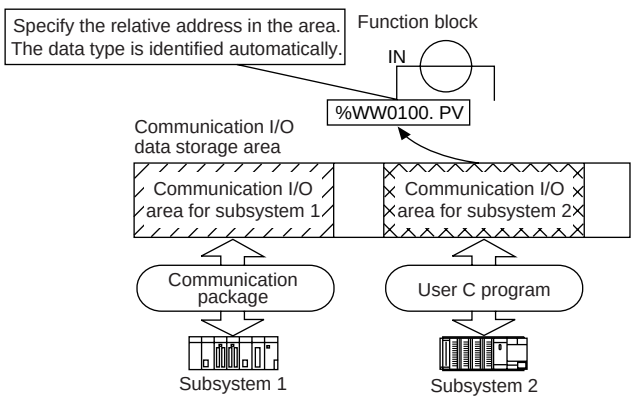

Figure 2 Flow of Communication Data

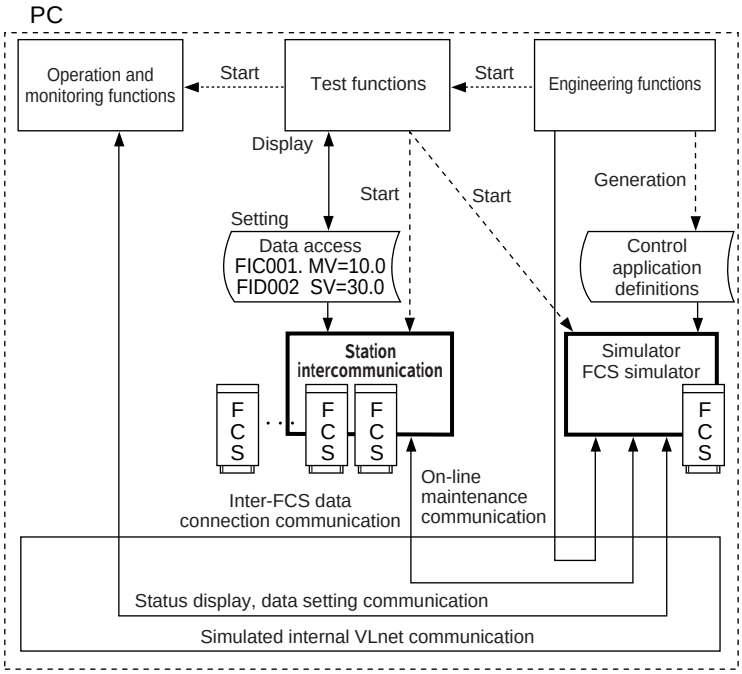

Figure 3 FCS Simulator and Other Functions

FCS Simulator

The FCS simulator runs as a process under Windows NT. The FCS simulator reads the control application definition generated by the engineering functions, and then carries out the specified control actions in the exact same way as done by an actual FCS. Communication between the operation and monitoring functions and the FCS simulator is performed by the simulated internal VLnet communication, which enables the VLnet (control bus) communication protocol to be used for communication between processes inside a personal computer. The use of simulated internal VLnet communication means that the operation and monitoring functions (HIS functions) and test functions are carried out the same way regardless of whether an actual or virtual FCS is used.

Station Intercommunication Simulator

If the applications in each FCS could be isolated from those of other FCSs when engineering a control application that requires data to be exchanged between FCSs, multiple users would be able to carry out system engineering concurrently, greatly improving engineering efficiency. To do this, a function that can simulate the communication with other stations was developed. This function is called a station intercommunication simulator and also runs under Windows NT. The station intercommunication simulator receives all communication packets sent from the FCS simulator to other FCSs; and then (1) saves the value requested to be set (by a data setting command) and (2) returns the preset value (upon a data reading command). The data values can be viewed using the graphic user interface provided with the test functions. This same interface can also be used to set the data values to be returned in response to data reading commands.

Thus, with just an FCS simulator and station intercommunication simulator, a control application relating to other FCSs can be tested without requiring any increase in CPU load.

Internal Operation of FCS Simulator

Figure 4 shows the internal operation of the FCS simulator. As previously mentioned, the FCS simulator runs under Windows NT. Each task that performs an FCS function is modularized as a dynamic link library (DLL) and behave as a thread in the FCS simulator when run.

Just like in an actual FCS, each thread calls up an FCS-kernel simulation function that is modularized as a DLL inside the FCS simulator, to carry out the specified action. The functions of the FCS-kernel simulation functions are equivalent to the kernels in an actual FCS. They are called upon when exchanging data with other tasks or with links outside the FCS such as VLnet communication. The interface of the FCS-kernel simulation functions is completely the same as the interface of the kernel in an actual FCS. Hence, the same source code of each task can be used in both, thereby guaranteeing functional equivalency between an actual FCS and the FCS simulator.

The scheduling of each thread is performed immediately before the thread returns to the task after calling up the FCS-kernel simulation function. The large black dots in Figure 4 indicate the points at which scheduling is performed (referred to as dispatch points). Whether the task should be switched over is checked at each dispatch point. If it is determined that the task be switched, a thread that is currently suspended is then resumed. This switchover is performed using the suspend and resume commands in the Win32API. The suspend command suspends a thread, and the resume command releases a thread from the suspended state. Thus at each dispatch point, the current thread performs a switchover by first resuming another thread and then suspending itself.

The interrupt process is also built as an independent thread. Window messages are used as interrupt triggers. The interrupt thread waits for a window message and, upon reception of a message, suspends the current task and carries out the interrupt process corresponding to that message. When the interrupt thread completes the interrupt process, dispatching is carried out and the suspended task to be carried out next is resumed. The interrupt thread then returns to the state of waiting for a window message.

CONCLUDING REMARKS

This report focuses on the improvement of engineering efficiency and communication flexibility among field control

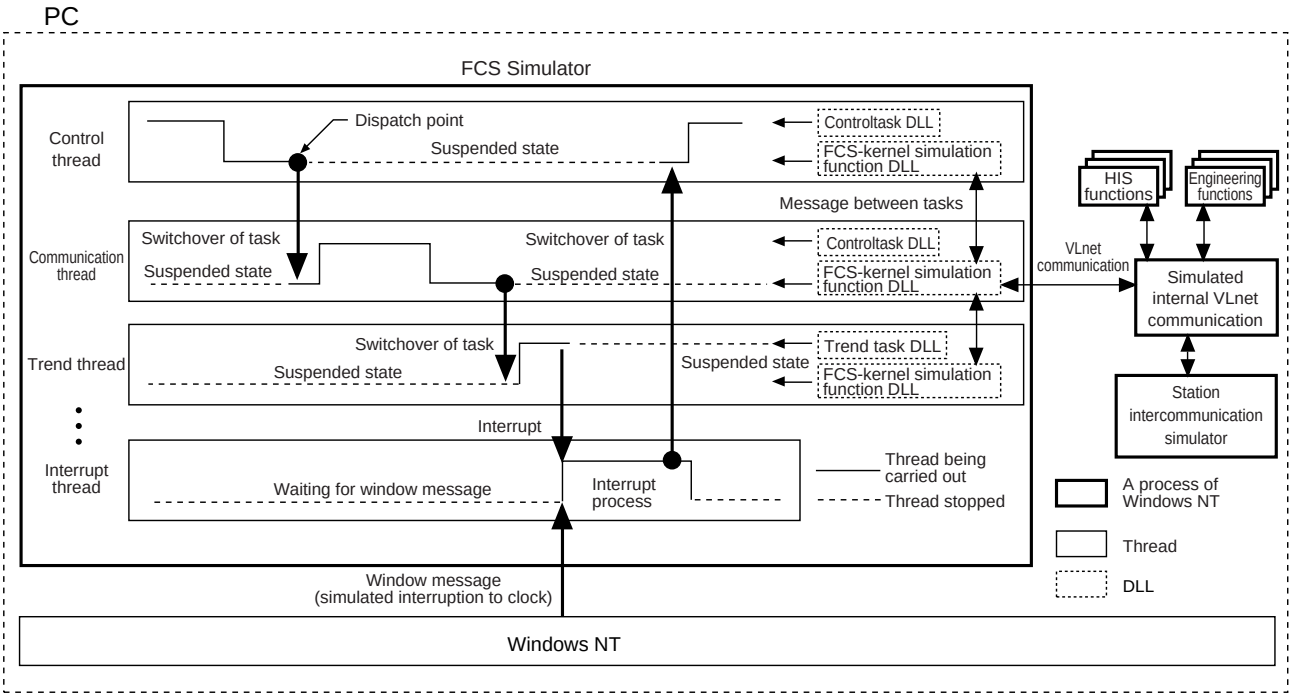

Figure 4 Flow of FCS Simulator

equipment that is achieved by the control functions of the CENTUM CS 1000. The assessment of the control functions also considers their reliability.

It is also expected that control functions will soon be required to have the same open interface and platform-free features as are required in human-machine interfaces and engineering functions. In this respect, we believe that the functions of the newly developed FCS simulator for the CENTUM CS 1000 can serve as a key technology for software development in the future. Such software may involve FCS software that runs on a personal computer and enables processes to be directly controlled from a personal computer, and gateway functions that use the communication interface of a personal computer. Furthermore, the enhanced field-communication functions for an FCS can surely be expanded into a highly reliable gateway. ◆

REFERENCES

*2 MATSUDA T., et. al., "Fault Tolerant Design for Field Control Stations," Yokogawa Technical Report, Vol. 18, pp. 10-13 (1994)

*3 Windows NT is a registered trademark of Microsoft Corporation, U.S.A.