Block Engineering to Efficiently Improve Control Software Development

Mitsunori Fukuzumi
Yoshiaki Shimada

1. Introduction

Innovations in the design of the D300win program support tool has greatly improved the efficiency with which one may create control programs in the MICR EX-SX series. In particular, the implementation of function blocks (FBs), which allow existing control programs to be reused, may greatly enhance control program productivity. Historically, similar programs for individual controls had to be made from scratch each time. Through the use of FBs, however, one can apply existing FBs to each new control element to construct the desired control program. Like a jigsaw puzzle, the existing FBs need only to be pieced in order to form the desired control program. This paper first introduces the concept of block engineering and then describes in detail the efficacy of FBs in the construction of control programs.

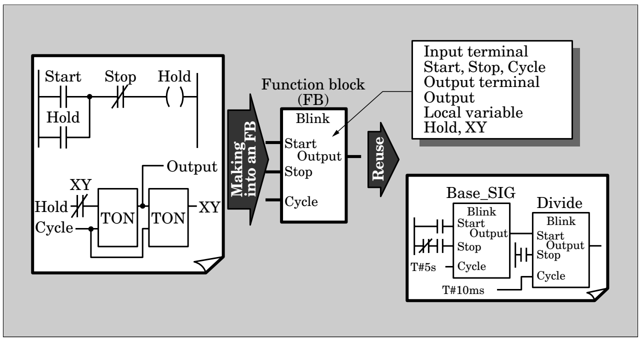
2. What is a Function Block (FB)?

A function block (FB) as prescribed in IEC61131-3 involves modularizing a program into component parts. Figure 1 shows a simple example of an FB. To the left of Fig. 1 a simple ON/OFF circuit is described with a ladder diagram. When a "Start" signal turns the circuit ON, an "Output" signal repeats the ON/OFF with a cycle twice that of the specified cycle.

In the center of Fig. 1, the circuit is made into a function block (FB). The D300win easily converts an arbitrary circuit into an FB (modularization). It is only necessary to specify signal names of an input terminal and an output terminal. This approach is analogous to hardware construction, in that a combination of discrete parts is coordinated into an integrated circuit (IC). In this scenario, signal names are only required for designation of input and output terminals. So, the input-output interface can be clearly expressed.

The right side of Fig. 1 shows an example of how FBs may be used in a ladder circuit. The same FB may be reused as another circuit by simply changing the instance name (corresponding to the part number of the hardware component).

Fig.1 General concept of the function block (FB)



3. Structured Programming in MICREX-SX

Structured programming is realized in MICREX-SX series. In this method, a whole application program is divided into blocks, each corresponding to a specific control object (elevator machine, conveyance machine, etc.) and a specific control purpose (carrying out action, coiling action, etc.). This block-based structure creates an efficient environment in which programming, debugging and administrating services may be performed. Furthermore, while still adhering to the hierarchical framework, each program block can be further modularized and functions simplified. Figure 2 shows this general idea.

In this way, creating hierarchical layers for block structures results in programs that are easy to read and maintain. In addition, the entire program or a component part can easily be divided and made into FBs. This approach allows for the program to be reused and increases the standardization of the control functions.

4. Block Engineering of MICREX-SX

If control functions are prepared in advance as well-proven FBs, most application programs will select the necessary FBs, and compile them together. This technique greatly improves programming efficiency. This is the main objective of the block engineering of MICREX-SX.

Fig.2 Structured programming of the MICREX-SX

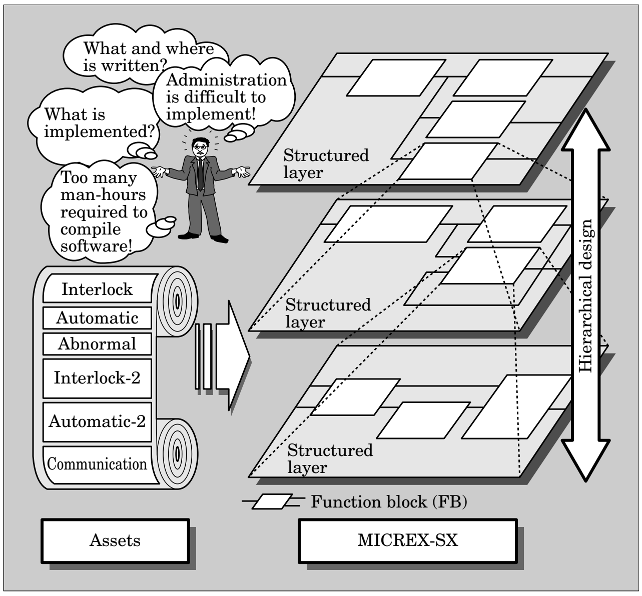
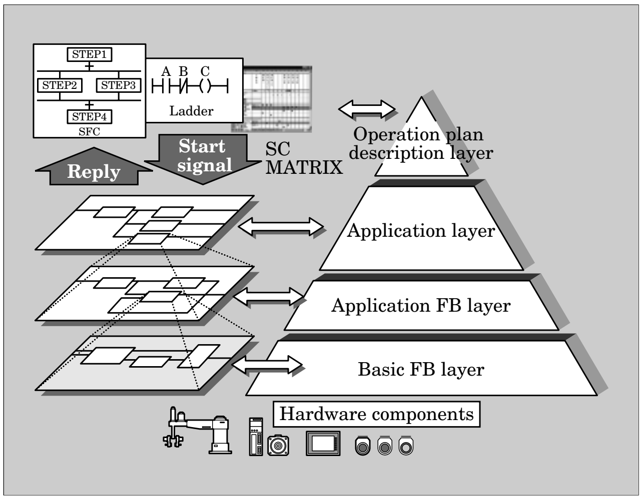


Fig.3 Hierarchical structure of block engineering



In this respect, it is important to know in what type of units should FBs be classified and provided. The block engineering of MICREX-SX builds an application program with hierarchical structure, and then classifies and provides FBs to match each hierarchy as shown in Fig. 3.

4.1 Basic FB layer

When modularizing a real control system into blocks (structured) with the intent to match up hardware and control purpose, the technique ultimately decomposes the control system into function blocks (FBs) for a small control object and its control function. Independent of the overall control purpose of the entire system, the FB functions that directly control real hardware components (sensor, actuator, etc.), that is, the functions to issue command pulses to a servo-amplifier or to receive data from a bar-code reader, depend only on the object component in the whole

Table 1 Control functions classification (partial)

Motion control	Positioning control	Point-to-point		
		Interpolation		
		Calculation of rotations		
		Synchronization	Synchronous operation	
			Traveling cutting	Rotary shear
				Flying cutter
	Speed control	Ratio control		
		Dancer control		
		Loop control		
	Torque control	Tension control		
		Torque balancing		
		Torque limiting		
	Coiling control	Dancer control		
		Tension control (tension detection type)		
		Tension control (coil diameter operation type)		
	Control operations			
Measurement control				
POD interface				
Information control	Web connection			
etc.				

Table 2 Machinery control functions for each type of business

Products suppliers	Rubber and chemical manufacturing industry Car manufacturing industry Others
Machinery system suppliers	Transportation machinery Print machinery Metalworking machinery Paper, film or plastic manufacturing machinery Semiconductor fabricating machinery Food processing machinery Wrapping machinery Others

hardware. In light of this, a set of FBs administrating the interface with hardware components is referred to as the basic FB layer.

4.2 Application FB layer

Table 1 shows a few of the classifications for machinery control. These classifications are for control algorithms, and do not theoretically depend on the control object, which is a type of sensor or actuator. As an example of the positioning control by a servo, consider the scenario in which a servo-amplifier is either related to the pulse train interface, remote I/O combination or the D-I/O combination. In this instance, the positioning control operation algorithm would be common. In this way, a set of FBs that are independent of the hardware components is referred to as an application FB layer.

4.3 Application layer

To implement the machinery control shown in

Table 2, the basic FBs and the application FBs are combined to create the desired control for each customer and for each machinery. The layer that forms this control is called an application layer. Generally, the application layer puts application FBs and basic FBs together and generates a program. The entire program or a component part can be made into an FB and registered as an application FB.

4.4 Operation plan description layer

In block engineering, a program that specifically describes the operation plan is extracted and separated from the application layer. This program is called an operation plan description layer. Between the operation plan description layer and application layer, event signals (bit signals) are exchanged.

The operation plan description layer is an administrative component that is designed to control step or state transitions in the system. In other words, it describes a set of steps in the control procedure. For example, the operation plan description layer records when (work arrives), what it is (a work axis) and how (to go forward) in order to issue event signals (“forward command to work axis”) to the application layer.

The application layer and the other underlying layers receive an event signal, perform the corresponding action and then report the status (“action completed”, etc.) of the action back to the operation plan layer. Thus, they play a role in the action execution component.

By separating the event administrating component from the action execution component, the functions of the latter become extremely simplified. Therefore, it becomes easier to implement structuring (modularization into blocks) and increases the standardization for programming.

Fig.4 Application of block engineering (system configuration)

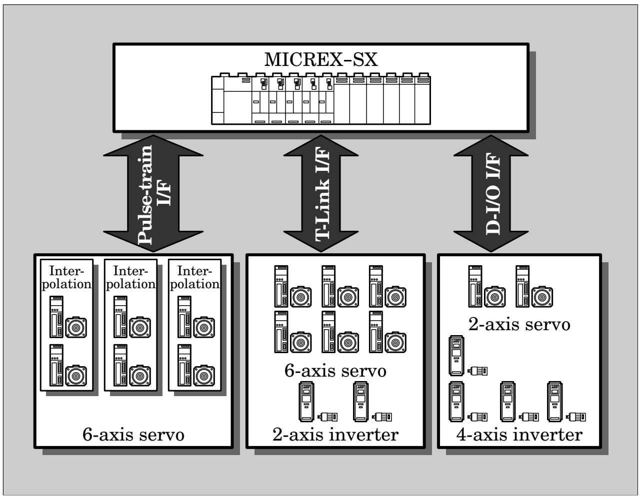
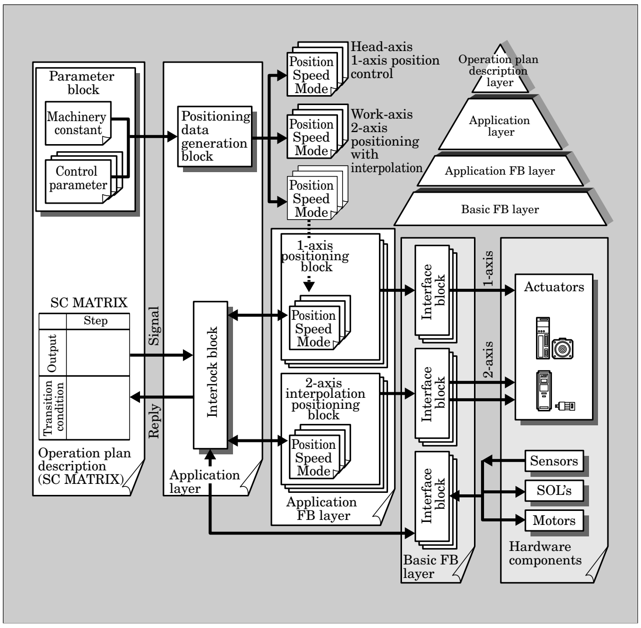


Fig.5 Application of block engineering (hardware configuration)

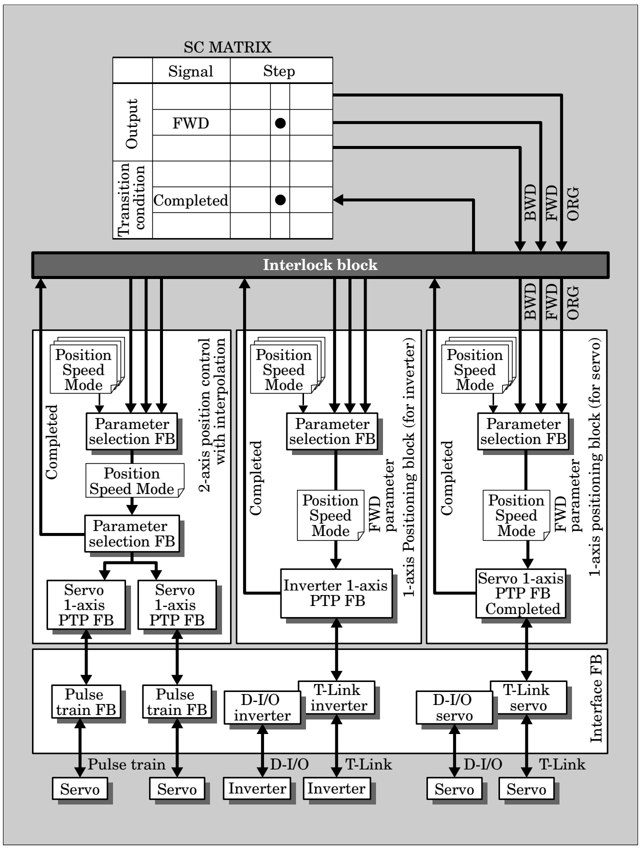


5. Application of Block Engineering

Figure 4 to Fig. 6 illustrate the application of block engineering to positioning control. It should be remembered that this example extracts only a general idea for explanation.

Figure 4 shows the system configuration. This example is of a positioning control with a total of 20 axes of servos and inverters. Of the 20 total axes, 14 axes control 1-axis PTP (point-to-point) positioning,

Fig.6 Details of function blocks (FBs)



while the remaining 6 axes control 2-axis interpolation of PTP positioning. Figure 5 shows the software configuration by block engineering. Each block of Fig. 5 is explained below.

5.1 Operation plan description layer

The operation plan description layer includes an operational plan (state transition description) describing the positioning, the control parameters and the constants of the machinery for each product to be manufactured. This example depicts the case in which the SC MATRIX is used. Any user, even those without programming knowledge can compile and revise contents of the operation plan at will by writing the operation plan description layer with the SC MATRIX. Detailed information about the SC MATRIX is discussed in another paper.

5.2 Application layer

The application layer consists of the interlock blocks and the positioning control parameter generation block program for each axis. The interlock block is a program designed to take a necessary interlock, based on command signals (head-axis DOWN, work-axis FORWARD, etc.) from the SC MATRIX, and transfer it to the application FB layer and the other underlying FB layers. The positioning control parameter generation block is a program that analyzes data trains (machinery constants and control parameters) downloaded from the SC MATRIX and then generates positioning control parameters (position, speed and mode) for each axis.

5.3 Application FB layer

The application FB layer consists of 1-axis positioning blocks and 2-axis interpolation positioning blocks. The 1-axis positioning block consists of parameter selection FBs and 1-axis positioning control FBs (Fig. 6). The parameter selection FBs select positioning control parameters (position, speed and mode) which correspond to the bit command (forward, backward and return to origin) from the SC MATRIX, and then pass it to the 1-axis positioning control FB. The 1-axis positioning control FB executes positioning control operation using given positioning control parameters, and completes by returning the transition condition of the SC MATRIX.

The 2-axis interpolation positioning block adds 2-axis interpolation control FBs to the above blocks (Fig. 6). In other words, the positioning control parameters given by the 2-axis interpolation control FB are resolved into those corresponding to each of two axes, and shared with each 1-axis positioning control FB. The processing is the same as that of the 1-axis positioning block.

5.4 Basic FB layer

The basic FB layer consists of interface control FBs for hardware components. Corresponding to the hardware configuration of Fig. 4, the basic FB layer is configured from the following five types of FBs (Fig. 6).

(1) Inverter control FB with T-Link interface (I/F)
(2) Servo control FB with T-Link I/F
(3) Inverter control FB with D-I/O I/F
(4) Servo control FB with D-I/O I/F
(5) Servo control FB with pulse train I/F

Connecting the appropriate basic FBs listed above to the 1-axis positioning FB makes the control system compatible with all types of actuators (inverters or servos) and interfaces (T-Link, D-I/O or pulse train).

6. Effects of Block Engineering

Construction of a control system using the configuration listed above produces the following favorable effects.

(1) Because FBs are divided into basic and application FB layers, independent of the interface, common use of the control component (application component) is possible. In Fig. 6, the interface is varied among T-Link, D-I/O, pulse train, etc., however, the positioning control can only commonly use the “1-axis positioning FB”.
(2) Combining application FBs together can build various functions. In Fig. 6, combining two of the 1-axis positioning FBs together with one of the 2-axis positioning FBs can create 2-axis interpolation positioning.
(3) An application can be administrated and standardized on an FB unit basis. In addition, software created in this manner is durable and can be used in the future because FBs are described in the international standard language IEC61131-3.
(4) By separating the action component from the operation plan component (event component), the functionality of the action component becomes more simplified, making it easier to implement structuring (modularization into blocks) and achieve standardization. Furthermore, automatic adjustment of operation can be easily made by simply revising the event component.

7. Conclusion

The authors propose that the block engineering method, in the form of a series of integrated function blocks (FBs) based on IEC61131-3, result in an efficient operative method. As a consequence of this approach, FBs are likely to play an indispensable role on the PLC programs of the future, as the block method allows for the reuse and standardization of FBs. The authors hope that this paper will help apply the block engineering to practical applications.