

Integrated Programming Support System for MICREX-SX Series

Mitsunori Fukuzumi
Masashi Yamada
Akihide Hamada

1. Introduction

Control software is becoming more complicated and larger-scaled as the functionality of programmable controllers (PLCs) is enhanced. Increasing programming efficiency is an important job the programming support tools. However, the following factors prevent current programming techniques from achieving a higher efficiency.

- (1) Low productivity because of a low-level programming language
- (2) Difficulty in reusing programs
- (3) Difficulty in maintenance due to lack of transparency in the program structure

Consequently, users are unnecessarily burdened by both the programming and program maintenance. Limiting these factors to achieve significantly increased efficiency is strongly desired.

Fuji Electric has developed an integrated programming support system (SES) for the MICREX-SX series (hereafter referred to as the SX-series). Fuji adopted the following three basic concepts for the development of the SES.

- (1) Utilization of an international standard language
- (2) Achieving a breakthrough in the software development procedure
- (3) Unification of various support functions

This paper will introduce features of the SES and present an overview of its functions.

2. Features of the SES

An overview of the entire SES is shown in Fig. 1. SES is a software product used to realize a highly effective programming support system environment for various modules of the SX-series. Features of the SES are described below.

2.1 Realization of a comfortable support environment

One of the most important aspects of a programming support system is its ease of use. The SES uses general purpose computers as its hardware and utilizes Windows^{*1} 95 or Windows NT^{*2}, current international de facto standards, as its operating system. A user

interface with consistent operations, intuitive friendliness, multi-windows, and that can be learned in a short time, has been realized by conforming to the Windows style guide (the standard specifications for Windows display operations), comprised mainly of mouse operations. To comply with environments such as onsite field locations where no mouse can be used, each operation can also be performed from the keyboard.

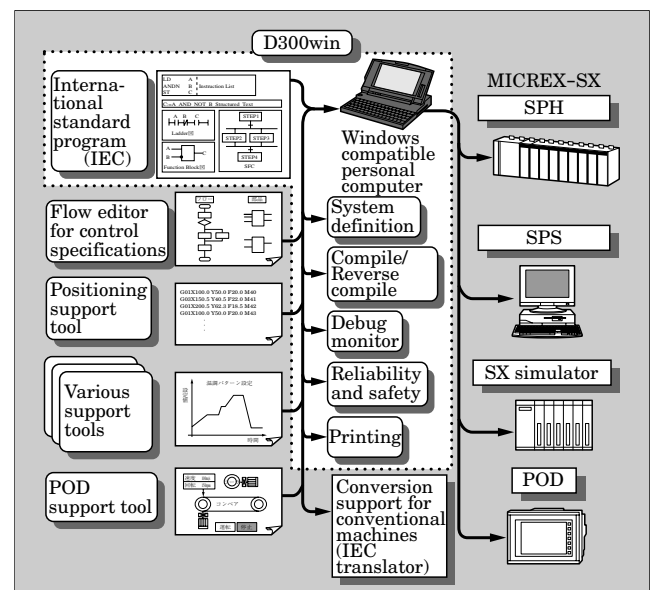
2.2 Utilization of the international standard language [IEC61131-3 (revision of IEC1131-3)1993]

The traditional ladder language (ladder diagram) is mainly used as today's PLC language. The ladder diagram has the advantage of graphical expression that is intuitive and easy to understand. However, due to many other problems, it is generally difficult for the ladder diagram to describe and maintain large-scale

*1 Windows: A registered trademark of Microsoft Corp., USA

*2 Windows NT: A registered trademark of Microsoft Corp., USA

Fig.1 Overview of the SES



programs or complicated numerical operations. Main problems of the ladder diagram are listed below.

- (1) Symbols and functions not common among various PLC products
- (2) Poor structuring and hierarchical function
- (3) Restricted reusability of software
- (4) Poor addressing and data structure

Each manufacturer is making various efforts to rectify these problems. However, this increases the differences among manufacturers. The introduction of a de facto standard for the language can hardly be expected. To resolve these problems, Fuji Electric has fully adopted the international standard language (IEC61131-3).

2.3 Improvement of programming efficiency

- (1) Programming using labels

The use of descriptive labels (variables) is a fundamental aspect of programming. The compiler assigns labels to the internal memory automatically, avoiding such mistakes as double assignment. Since input/output addresses can be assigned individual labels that are separate from the program, the program will not need to be modified if the input/output addresses are changed.

- (2) Reuse and combination of programs

Programs have been made reusable by handling them as parts (functions and function blocks). The parallel development of programs is made easier because partial programs written by several programmers can be combined together by using cut and paste functions.

- (3) Simulation

By implementing a simulator in SES, logic simulation of newly developed programs is possible without using actual hardware.

- (4) Improved document quality

User-specific printing formats are made available. Furthermore, print preview and enlarged/reduced scale printing, regardless of the paper size, have also been made available.

2.4 Utilization of existing software resources

Existing programs developed with the conventional MICREX-F and FLEX-PC can be translated into the IEC61131-3 language (ladder diagram).

2.5 Automatic generation of control programs by the flow editor for control specifications

A system has been developed that, based on the control specifications described by message flows and software parts, automatically generates PLC programs for controlling machines. The elimination of programming procedures drastically improves the efficiency of application programming.

2.6 Integration of various support functions

The modules that comprise the PLC include such

various function modules as a positioning module and a programmable operation display (POD). Previously, these function modules required individual custom support tools, but the SES provides a common platform to which each support tool can be added-on. This enables various support tools to be run on the same personal computer that is running the SES, and the sharing of the same labels defined by SES. Various support tools including POD and positioning are introduced in other articles of this special issue.

3. IEC61131-3

To develop internationally standardized programs, IEC61131-3 has the following goals.

- (1) To realize programs that do not depend upon the PLC model type
- (2) To realize easy to understand and maintain programs by means of structured programming
- (3) To improve programming efficiency by reusing program parts (functions and function blocks)
- (4) To decrease program errors by supporting strict syntax checking of data type declarations

3.1 Programming language

For each user application program, IEC61131-3 specifies one common element and four types of language (in some cases this is treated as five types of language, including the common element). Among the common elements, the ladder diagram and function block diagram are defined as the sequential function chart and graphical language, and the list and structure list are defined as the text language. Some examples of languages defined by IEC61131-3 are shown in Fig. 2. The main usage of each language is listed below.

- (1) List: downsizing of applications
- (2) Ladder list: replacement of relay-box
- (3) Function block diagram: applications comprising mainly data processing
- (4) Structured text: applications having complicated flow control
- (5) Sequential function chart: applications driven by time and events

The SES completely supports all of the above elements.

3.2 Structured programming

All programs, functions and function blocks written in the above languages are known as program organization units (POUs). Functions and function blocks can be called from each program or from another function block, making hierarchical programming easier.

A program can function after having been registered as a task in the CPU module (corresponding to the resource in IEC61131-3) of the PLC system (corresponding to the configuration in IEC61131-3).

Fig.2 IEC61131-3 language

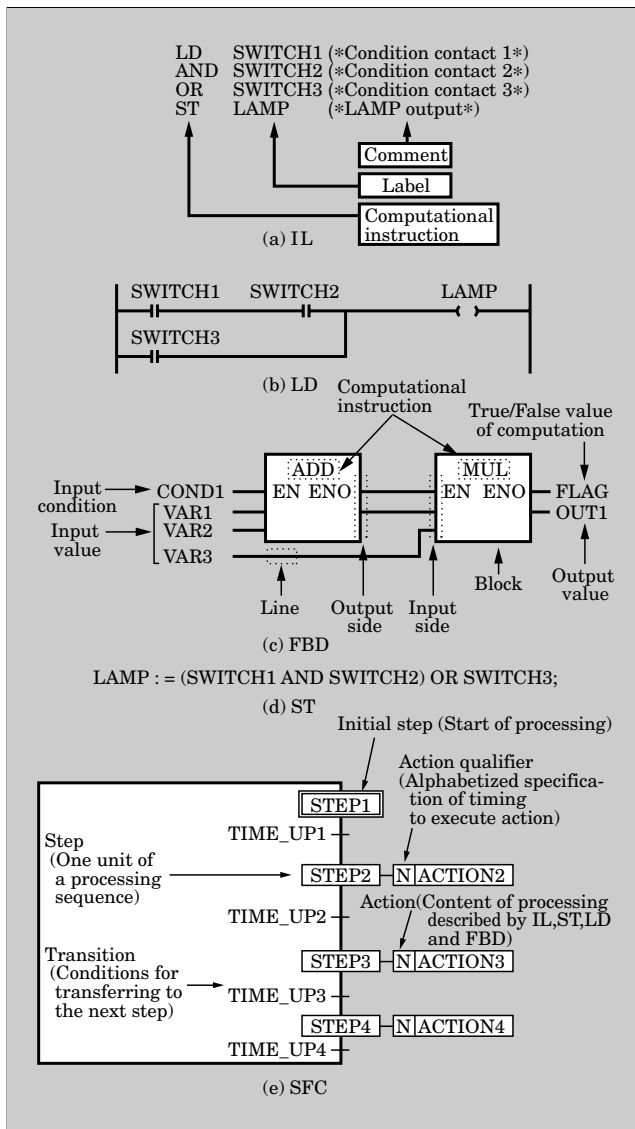


Fig.3 IEC61131-3 software model

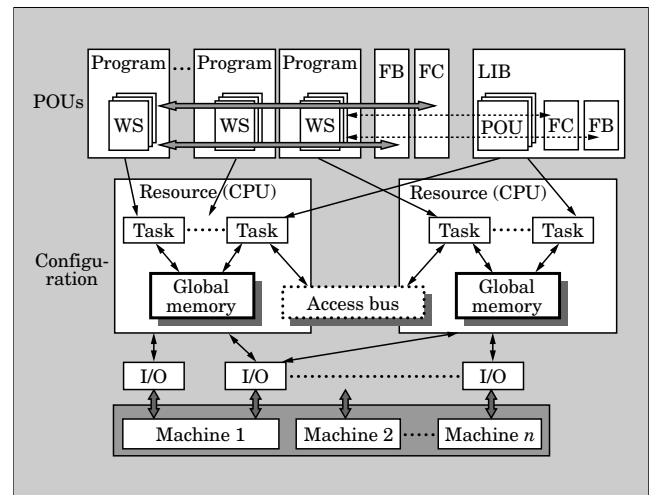
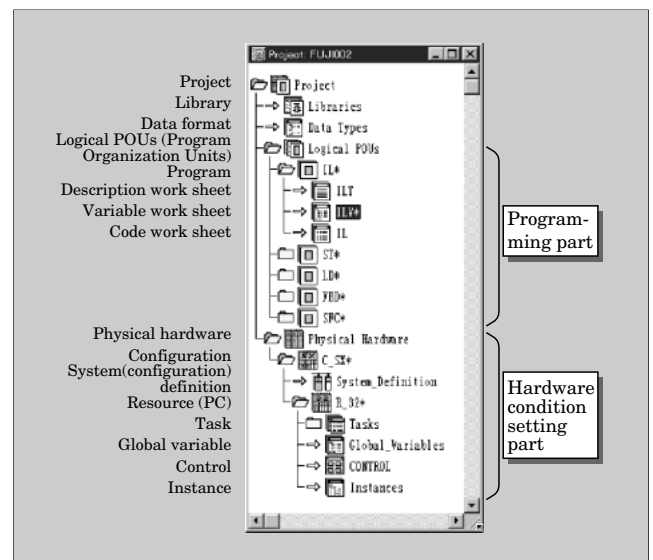


Fig.4 D300win basic user interface



Tasks are a means to activate and control programs. There are different types of tasks, such as: cyclic, fixed-cycle triggered, and event triggered. Several programs can be registered within the same task, and are processed in the registered sequence as if they were a single program. Thus, large-scale and complicated processing can be divided into smaller program blocks (POU)s, improving the programming efficiency largely. The IEC61131-3 software model is shown in Fig. 3.

For further details regarding the IEC61131-3, please refer to the “IEC1131-3 Handbook” (PLCopen JAPAN) or other reference.

4. Programming Design Support

4.1 Basic user interface

The basic user interface of the programming support part (referred to as D300win hereafter) in the SES is known as the project tree. (See Fig. 4.) A project is the management unit for all the information

of programs in the D300win, and is assigned an arbitrary name. All programs developed by the D300win are managed by this project name.

In Fig. 4, the programming part is the logical POUs node and the hardware condition setting part is the physical hardware node, including subordinated nodes. In other words, the hardware condition setting part consists of the parts for managing the configuration definition of the SX series, registering developed programs as tasks, download the PLC, testing and debugging, etc.

In addition, library nodes are provided to register existing developed projects for their reuse, and a data type notebook is provided to generate and register customer-specific data type templates utilizing various data types supplied by the D300win.

4.2 Programming

Programming is performed under the hierarchy of a logical POU in program, function or function block

Fig.5 Worksheet

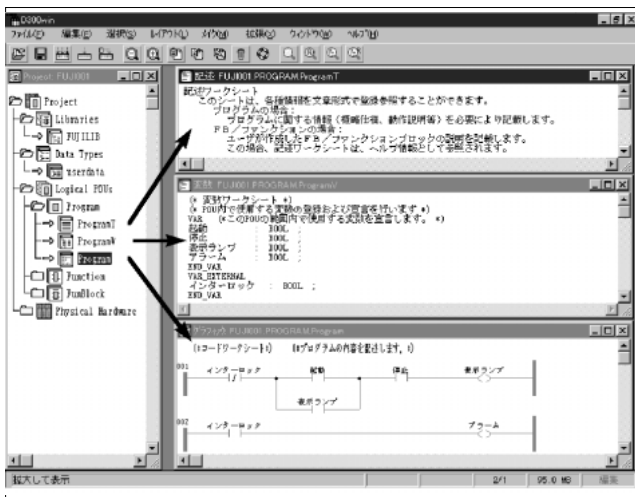


Fig.6 Sample programs by D300win

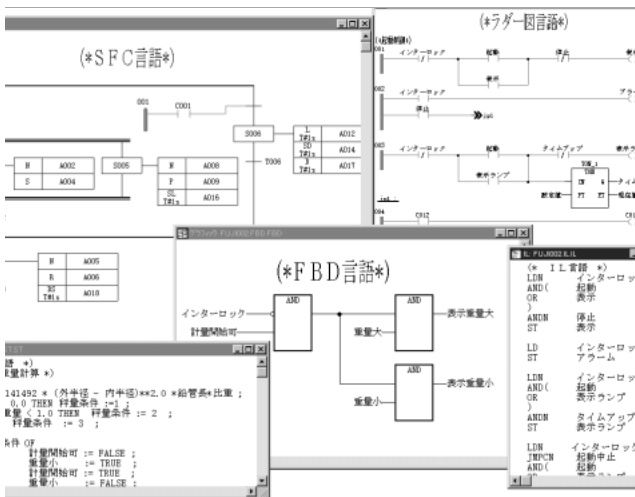


Fig.7 Variable registration



units. A descriptive language can be specified for each POU. Each POU consists of three types of worksheets (description, variable and code). An example is shown in Fig. 5.

Fig.8 Error check and jump

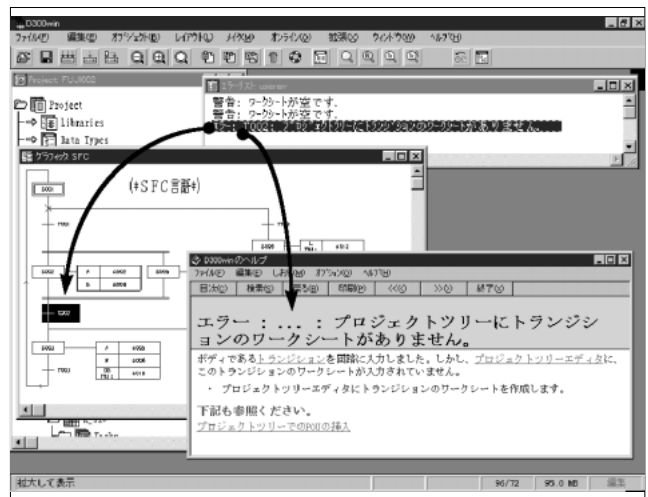
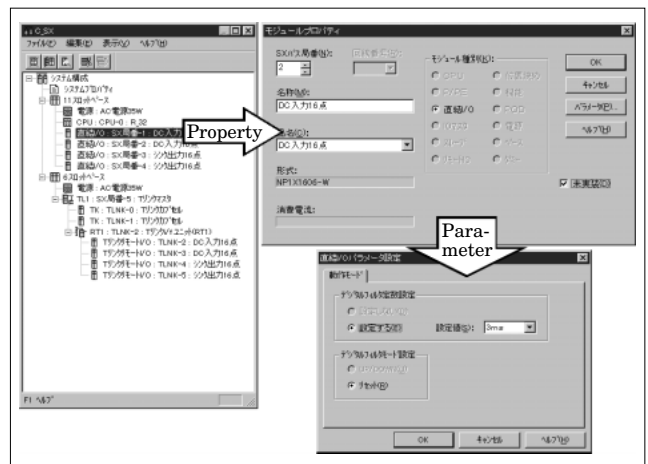


Fig.9 System configuration registration



4.2.1 Programming with various languages

A text editor and a graphic editor are provided for each text type language, such as the instruction list and structured text, and each graphic type language, such as the ladder diagram, function block diagram and sequential function chart. Each editor is automatically activated by the POU programming language. Examples of description by each language are shown in Fig. 6. Comments can be described freely in each language.

4.2.2 Programming assistance function

Various functions are provided for easier and quicker programming, several of which are introduced below.

(1) Variable registration function

Variable declarations can be registered easily via a special dialog, or quickly via a text editor. (See Fig. 7.)

(2) Error check and jump function

Errors in the developed program are detected through compiling and displayed in the user error window. By selecting an error message, display will jump to the error location in the program. By calling the provided help function, the cause of the error and

Fig.10 Task registration

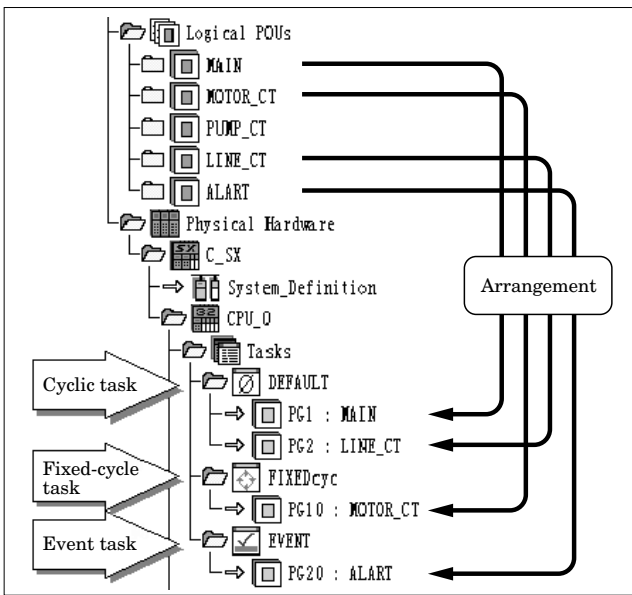
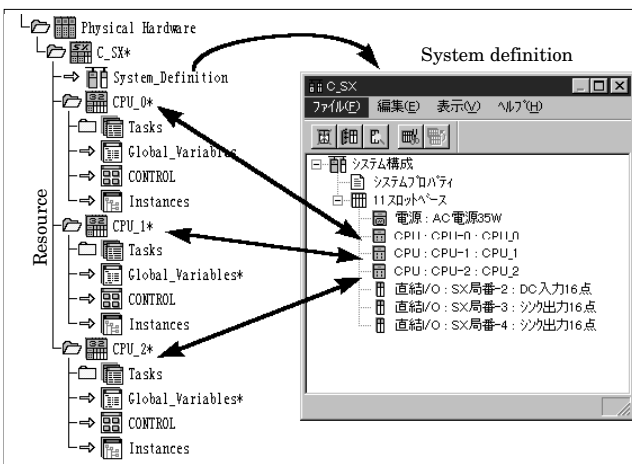


Fig.11 Multi-CPU registration



countermeasures can be displayed. Figure 8 shows an example.

4.3 Setting the program execution condition

4.3.1 System configuration registration

The configuration of hardware modules that comprise the SX system is registered and necessary parameters for each module are set. The actual configuration can also be uploaded from the SX system. A configuration tree resembles the physical image of the system. (See Fig. 9.)

4.3.2 Task registration

The CPU has three types of tasks: default, cyclic and event. Developed programs may be registered in any of these tasks freely. (See Fig. 10.) Programs of other projects can also be registered, increasing their reusability.

4.3.3 Multi-CPU

In the case of multiple CPUs, the task registration for each CPU is the same as when only a single CPU is

Fig.12 Resource control

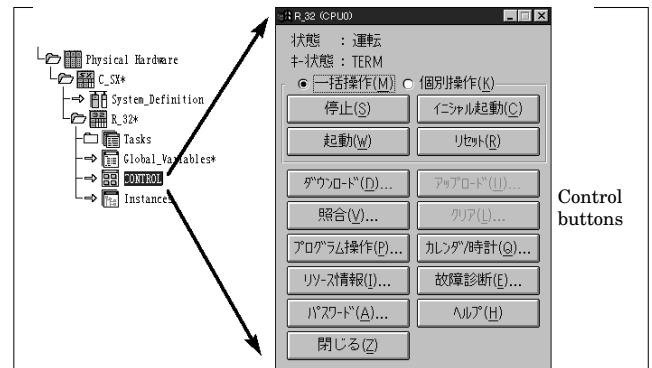


Fig.13 Program monitoring

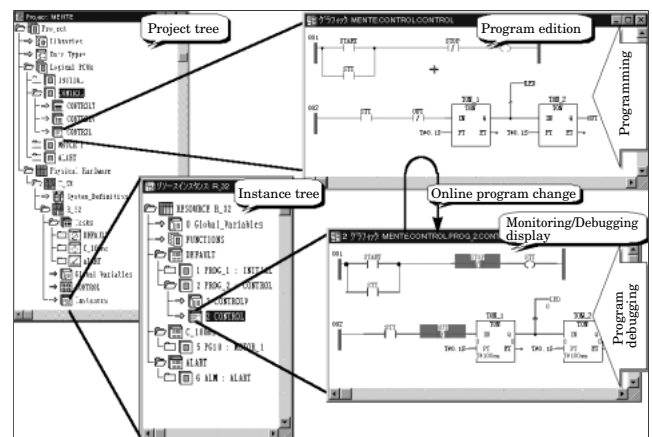


Fig.14 Watch list

| Online value | Variable name | Instance name |
|--------------|-----------------------------|----------------|
| 00058 | 1 Current_value_level | podeau |
| 00000 | 1 Current_value_Temperature | podeau |
| TRUE | 1 Inlet | podeau |
| FALSE | 1 Outlet | podeau |
| 00000 | 3 Current_value | FB_Temperature |
| 00060 | 3 Minimum_level | FB_Temperature |
| 00080 | 3 Maximum_level | FB_Temperature |

used. (See Fig. 11.) Since each program can be registered in any CPU, load dispersing and function distribution can be realized quite naturally.

5. Test and Debugging

5.1 Starting and stopping the controller

The start/stop of the SX system and download of programs is operated by control buttons in the resource control dialog. (See Fig. 12.)

5.2 Monitoring

Four types of monitoring functions are provided.

5.2.1 Monitoring program

The monitoring program function monitors the label values, contact status and coil status in the program. The format is the same as that generated by

Fig.15 Logic analyzer

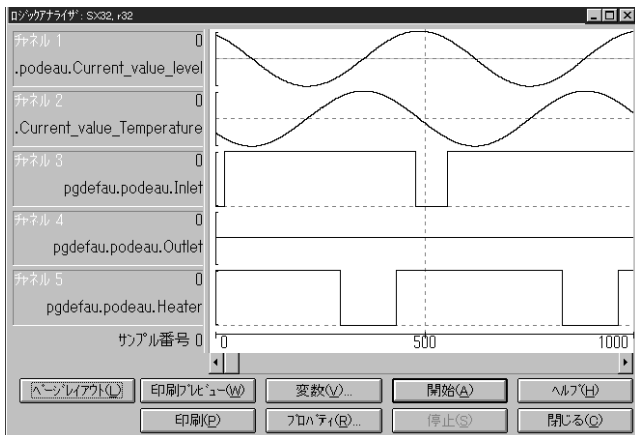
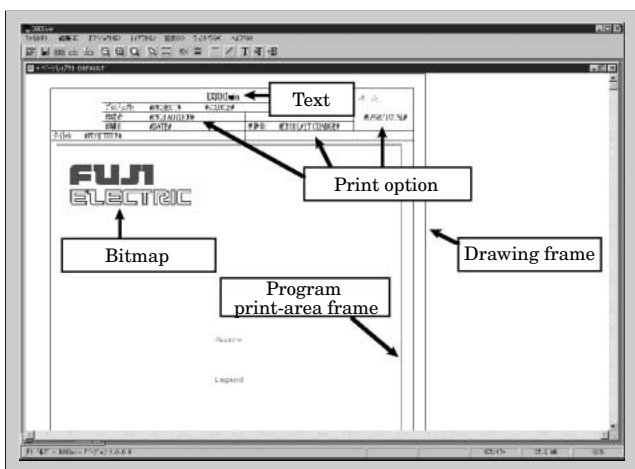


Fig.16 Page layout



the editor of each language. Visualization is increased by assigning meaning to each display color. (See Fig. 13.)

5.2.2 Monitoring variables

The monitoring variable function monitors the variables (labels) declared in the program.

5.2.3 Watch list

The watch list selectively monitors necessary variables (labels). (See Fig. 14.) Necessary variables can be easily registered from the monitoring program and the monitoring variable screens.

5.2.4 Logic analyzer

The change over time of specified variables is displayed as waveforms. Various trigger conditions can be set. (See Fig. 15.)

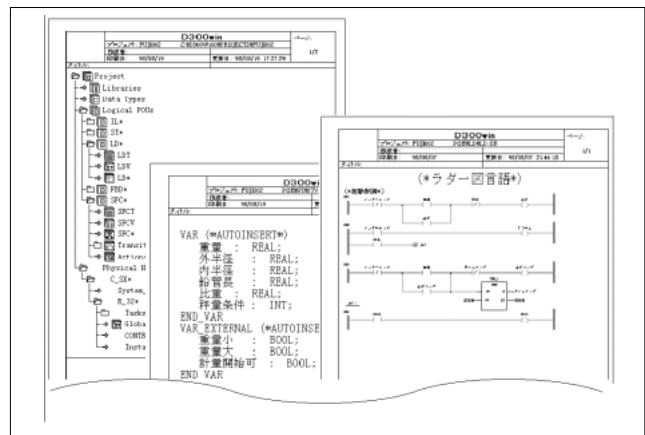
5.3 Online testing

The following online debugging functions are provided.

- (1) Forced set/reset
- (2) Breakpoint
- (3) Step execution
- (4) Monitor stop by condition
- (5) Program control

By means of the forced set/reset, the input can be

Fig.17 Print project



set forcibly as a fixed value, independent from external input, or the output can be set forcibly as fixed value, independent from the results of computation. The program control can enable or disable execution of each program, which is very effective for debugging a structured program.

5.4 Online program change

The path of each POU can be changed while the monitoring program is running. (See Fig. 13.) Changed programs are downloaded automatically without stopping the CPU module.

6. Maintenance Support

6.1 Zip/Unzip of projects

Developed programs and large quantities of auxiliary information for each project can be compressed and zipped into a file as well as decompressed and unzipped.

6.2 Printing

The following main functions are provided to simplify and reduce user work for printing various software documents.

6.2.1 Page layout

Standard drawing frames and other layouts can be generated freely. Figure 16 shows an example of the page layout.

(1) Bitmap

The bitmap selects such logos as a company name in the BMP format and specifies their print area.

(2) Drawing frame, line and text frame

The frame and lines (style and width) for drawings and characters (type of font and size) for notes or other text are set by this function.

(3) Print option

For each layout, print information is specified corresponding to specific key words (worker, time of printing, time of storage, project name to be printed, etc.).

Fig.18 Reliability and safety

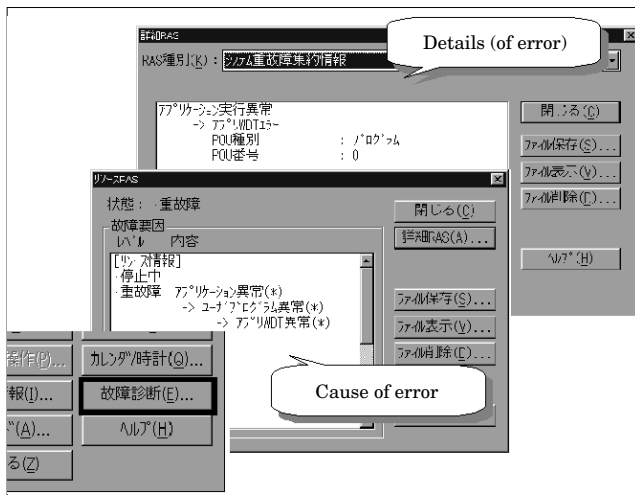
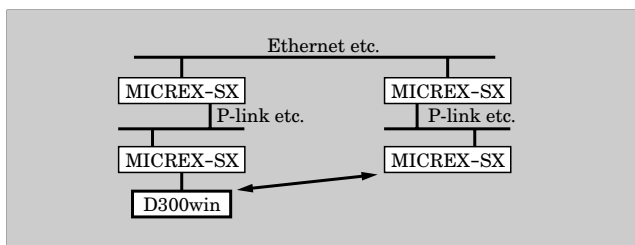


Fig.19 Loader network



- (4) Program print-area frame
Program print areas are specified.

6.2.2 Print preview

The print preview function makes it possible to confirm, without printing, the print conditions of such items as project configurations or various worksheets.

6.2.3 Print project

When printing different types of information such as programs and auxiliary information, to minimize the number of printing sheets, page feed can be omitted to minimize open spaces. A suitable page layout for printing can be selected from pre-prepared layouts. Figure 17 shows an example.

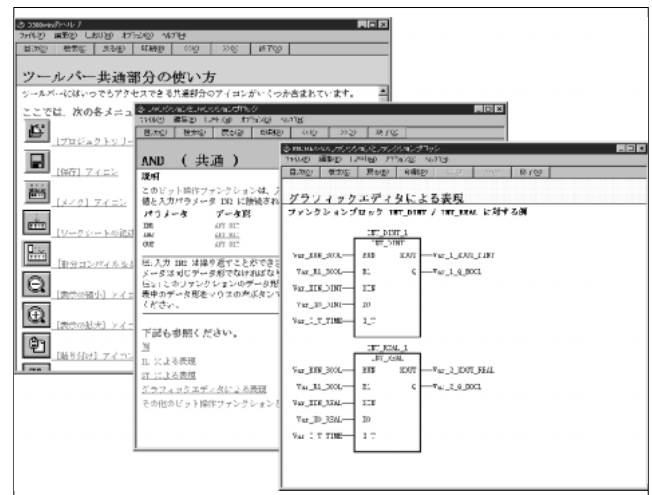
6.3 Password

D300win permits setting a password in the controller. By setting the password, illegal program calls and system configuration changes are forbidden. Thus, the system is protected from negligent change or operation by a third party.

7. Reliability and Safety

When a failure occurs, identification of the failed point is urgently required. This identification is made easier by displaying messages hierarchically in the following order: failure level → failed module → specific error cause within the module. (See Fig. 18.)

Fig.20 Various types of help



8. Assistance Functionality

8.1 Loader network

In the past, Fuji Electric has provided connection to a controller via a single network with P/PE links or other means. The D300win system, however, links the connected SX system to a controller of another SX series via the relayed connection of a maximum of two systems by means of various networks (Ethernet^{*3}, P/PE link etc.). Figure 19 shows an example connection. This function enables program maintenance, debugging and monitoring in a high-level, sophisticated network system.

8.2 Reverse compile and cross compile

The reverse compiler uploads programs from the controller into D300win and redisplay them in the instruction list format.

Variable information is downloaded in advance together with programs. Thereafter, if no project data is available, the variable information is uploaded together with programs again so the original variable labels used during programming are displayed.

The cross compiler translates programs written in instruction language into the ladder diagram, and then displays them.

8.3 IEC translator

In conformance with IEC61131-3, the IEC translator translates existing programs (including statements and comments such as device names) for Fuji Electric's MICREX-F series and FLEX-PC series into instruction list expressions. The IEC translator also translates some system definitions and parameter settings into the system definition of the SX system. This makes it easy to transition from an existing system to an SX system.

*3 Ethernet: A registered trademark of Xerox Corp., USA

Fig.21 Overview of the flow editor

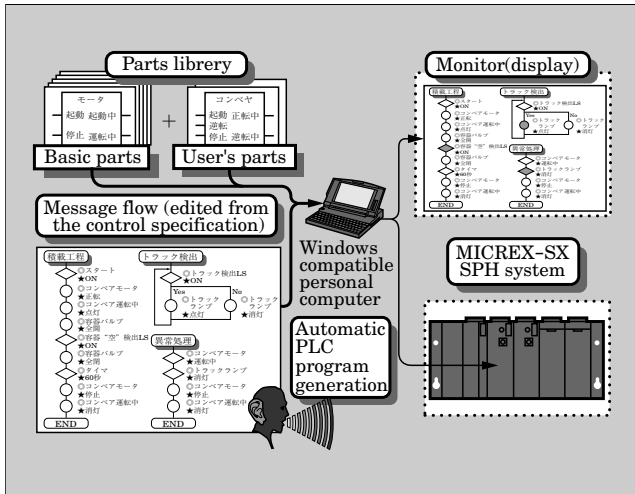
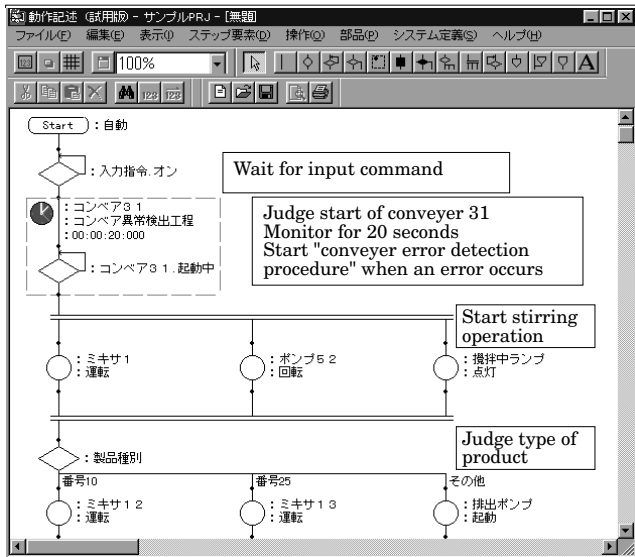


Fig.22 Example program of the flow editor

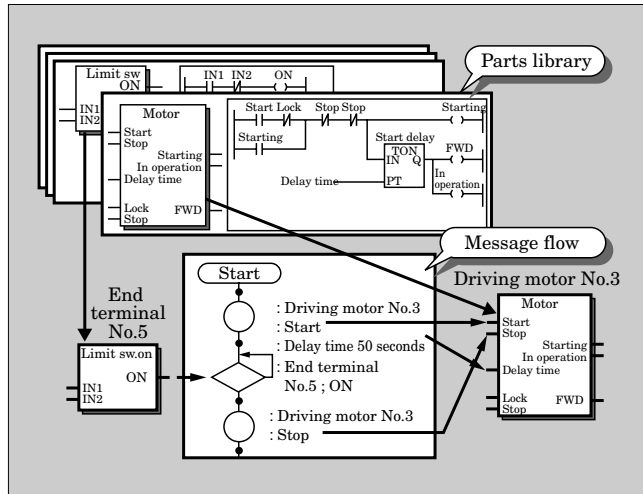


9. Various Types of Help

In the D300win system, necessary help information for the particular work situation can be called with a simple operation. An environment is provided in which work can continue rapidly, with no need to keep an operating manual on hand. The configuration and function of main help topics are introduced below. Figure 20 shows some examples of help information.

- (1) Configuration of the help information
 - (a) How to use and program the D300win
 - (b) Description of functions and function blocks of each PLC that is used, usage of each language, etc.
 - (c) Detailed description of error messages that occur during debugging and examples of countermeasures
 - (d) Description of the IEC61131-3 standard
- (2) Context sensitive help

Fig.23 Software parts and message flow



Suitable help information according to the context of the work can be called by a simple key operation.

10. Flow Editor for Control Specifications

Automatic program generation from control specifications results in a tremendous improvement in the production of control software. The flow editor for control specifications generates programs automatically from a description of the control specifications by flows and software parts. Figure 21 shows an overview and Fig. 22 shows an example description of the flow editor.

10.1 Software parts

These are software packages for controlling individual machines. The contents of these software packages are the same as the function blocks defined by IEC61131-3. A function block consists of an input parameter and an output parameter. Function blocks are used by the flow editor for the control specification, and as the interface to the message flow, have the following meanings.

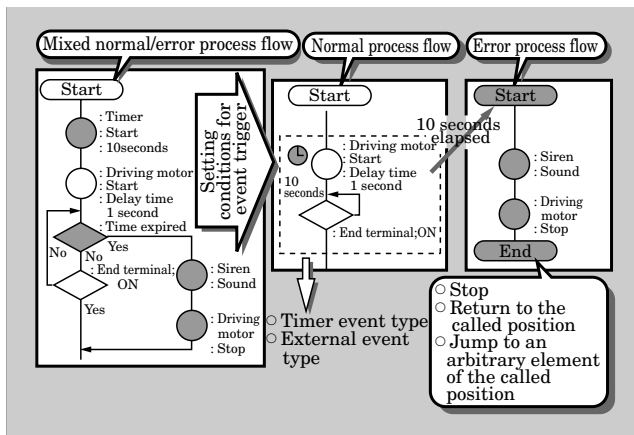
- (1) Command-terminal
Writes commands from the message flow
- (2) Status-terminal
Reads status of the message flow
- (3) Parameter-terminal
Specifies details of the command
- (4) Input terminal and output terminal
Used for external input and for connection between exterior and parts

Software parts are managed as a library of the flow editor for the control specifications. Software parts can be created freely by users.

10.2 Message flow

The message flow describes, in flowchart format, control commands for software parts and status judgments, i.e. the control specifications. The message flow

Fig.24 Setting conditions for event trigger



consists mainly of command elements (○ elements) and judgment elements (◇ elements).

For example, if a part [motor] has a command terminal [start], then [part: motor, command: right rotation] is described using a command element in the message flow. If waiting for input of a limit switch, [waiting condition: limit sw. on] is described using a judgment element. The correlation between software parts and message flows is shown in Fig. 23.

10.3 Monitoring

The message flow can be used as a monitor display. When a message flow being executed is shown on the monitor display, the step element being executed changes color.

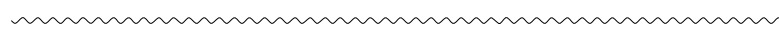
10.4 Setting conditions for the event trigger

Control specifications must usually describe not only how to handle normal operation, but also various errors. This makes it difficult to decipher a message flow that includes many judgement elements for every error case.

The flow editor for control specifications solved this problem by separating normal operation from erroneous operation by setting a condition area for an event trigger. Event and activation processes (another message flow) are set by enclosing a portion of the message flow within a rectangle. When an event occurs during program execution within the rectangle, progress of that message flow is stopped and a specified message flow is activated. Thus, a message flow that describes normal operation can be separate from, and not mixed with diverse processing for various error cases.

11. Conclusion

Features and functions of an integrated programming support system for the MICREX-SX series have been introduced above. The IEC standard language, structured programming and flow editor for the control specification, as proposed by the SES, can provide an innovative environment to develop PLC programming, which is gradually approaching its limitations. In the future, Fuji Electric will enlarge the concept of SES, promote development mainly of the flow editor, and contribute to improving the efficiency of application development for control systems.





* All brand names and product names in this journal might be trademarks or registered trademarks of their respective companies.