New automatic safety programming using safety function blocks of PLCopen

SUGANUMA Hiromu, FUJIMURA Ryosuke, KUBO Hiroko and TAKUMA Nobuyuki

In recent years, manufacturing companies need to respond the request of "multi-variety, variable-volume production" in the era of diversified needs. Along with that, they also need to develop safety systems that can be flexibly modified according to the changeover of production lines. As a solution for that, safety controllers are often introduced.

Since PLCopen, the standardizing association defined the specification of the safety function blocks, introduction of safety controllers has gradually become less challenging. However, it is still not easy for unskilled developers in safety programs to use these function blocks, because the specification defined by PLCopen is highly flexible. Since it is also difficult to create a safety program with high scalability and readability considering future improvements and extensions, it can be said that the barriers to introduce safety controllers are still high.

To offer solutions for those issues, we realized automatic safety programming on the Omron's programming software Sysmac Studio. In this paper, we describe technical achievements and effects of automatic safety programming.

We confirmed that the "automatic programming" function of the Sysmac Studio can shorten development period of safety systems.

1. Introduction

1.1 Necessity for introducing safety controllers

In recent years, manufacturing companies in Japan are facing the significant environmental change of a "serious shortage of human resources." According to the White Paper on Manufacturing Industries (Monodzukuri) 2018[1] published by the Ministry of Economy, Trade and Industry (METI), 94% of businesses answered that they have problems concerning staffing and more than 30% of businesses answered that the shortage also affects their businesses. Among the staffing problems, the most problematic staffing in particular is of "skilled human resources" and the problem is assumed to be caused by changes of factories and the control thereof.

According to the necessity to deal with the change from factories of "high-volume, low-mix production" during the age when goods were insufficient to factories of "multi-variety, variable-volume production" in the age when needs became diversified, the establishment and maintenance of systems where the flexible changeover of production lines is possible are now required.

Although a safety system where the flexible changeover of production lines is possible is also required for safety control, such safety systems for maintaining the production lead time cannot be realized with conventional hard-wring or safety relay units, because they require time for changing wiring or expansion of the system. To deal with this this problem, there is an increased need for introducing safety controllers.

The term "safety controller" means a controller for enabling safety control that has acquired international safety standards certification, wherein a flexible safety system can be realized by wire-saving through realizing the software safety circuit and the safety network.

1.2 Present situation of safety controllers

For more than ten years, safety controllers which conform to international safety standards, a typical example of which is IEC 61508[2], were launched by various manufacturers. Furthermore, PLCopen, which is a standardization organization aiming for the "efficient development of PLC applications," is promoting definition standardization regarding programming, aiming for effects including the reduction of training costs, the improvement of the reusability of programming, and the

Contact : *SUGANUMA Hiromu* hiromu.suganuma@omron.com

reduction of safety standard certification costs.

As a result, the function blocks and the data types used for the safety program were communalized, the bar for program creation was lowered, and the barriers to introducing the safety controllers are becoming lower.

Omron Corporation has also released the NX-SL series safety controllers and the automation software Sysmac Studio that support the safety function blocks of PLCopen, thereby offering a safety system that can deal with diversified needs.

2. Problems and Countermeasures

2.1 Problems with safety function blocks of PLCopen

The safety function block specifications of PLCopen are defined to flexibly deal with even complicated safety systems. Therefore, the designers of safety programs have to choose required parameters out of many parameters defined in such function blocks and set adequate values. For example, the reset signal for interlock has three parameters and setting is required for each parameter.

In addition, the function block for safety input of the PLCopen function blocks does not support the start/restart interlock triggered by the pulse reset signals OFF → ON → OFF that are required by the ISO13849-1[3] international safety standard. Therefore, it is necessary for designers to create a program that conforms to the requirements of the standard by combining multiple function blocks and setting adequate parameters.

As stated above, it can be said that knowledge and experience regarding safety programming that conforms to international safety standards are required for creating a safety program using the safety function blocks of PLCopen.

2.2 Problems with safety programming

In addition, to deal with changes for the evolvable "multi-variety, variable-volume production" factory, it is necessary to realize the establishment of a safety system that can endure future improvements and expansions. For example, for cases where the number of I/O points is increased for expansion, the associating additional number of programs may increase more than originally required, or the program may become complicated owing to the added programs, thereby resulting in deteriorated readability. To avoid such a situation, it can be said that the problem in dealing with the changes is to realize highly-expandable and readable safety programs wherein an increase or decrease in the number of I/O points is taken into consideration in advance.

2.3 Countermeasures

For the purpose of resolving the problems described above, we realized a function that is capable of automatically creating a program only through simple setup without requiring programming on Omron's automation software Sysmac Studio, which is used for executing program design of safety controller.

Conventionally, the skills of experienced designers were required for realizing the start/restart interlock using pulse reset signals or creating programs with high expandability and readability that can deal with increases and decreases in the number of I/O points, and it was hard to automate the creation of safety programs by using programming tools.

This time, the expertise of safety programming was incorporated in the automation software Sysmac Studio, which enabled the minimization of the number of items that are set up by designers, and easily created safety programs with high expandability and readability that are divided into the input part, output part, and logic operating part. It also enabled unexperienced designers to create safety programs equivalent to those created by experienced designers.

3. Outline of Automatic Programing Function

The automatic programing function is a function to automatically create programs by inputting the setup items shown in Table 1.

Table 1 Setup items of automatic programing

Setup Items	Description
Basic setup	Designate variables that are used as reset signals of the safety program.
Input setup	Designate the variables and reset types assigned for safety inputs, and the name of the safety function block used.
Output setup	Designate whether variables assigned for safety outputs and external device monitoring (EDM) are used.
Expected value setup	Set up the expected value (0 or 1) of output variables for the respective input variables in matrix form.

Fig. 1 shows the program created by the automatic programing function.

The program created consists of three blocks including the "input part" which processes inputs from the safety switch or the safety sensor, the "output part" which processes outputs to the safety relay, and the "logic operating part" which connects the aforementioned two parts, and these three blocks are connected via internal variables. Here, the term "internal variables" implies local variables that are used only within the program structural unit.

In the next chapter, the method for realizing the automatic programing function will be sequentially described for each block.

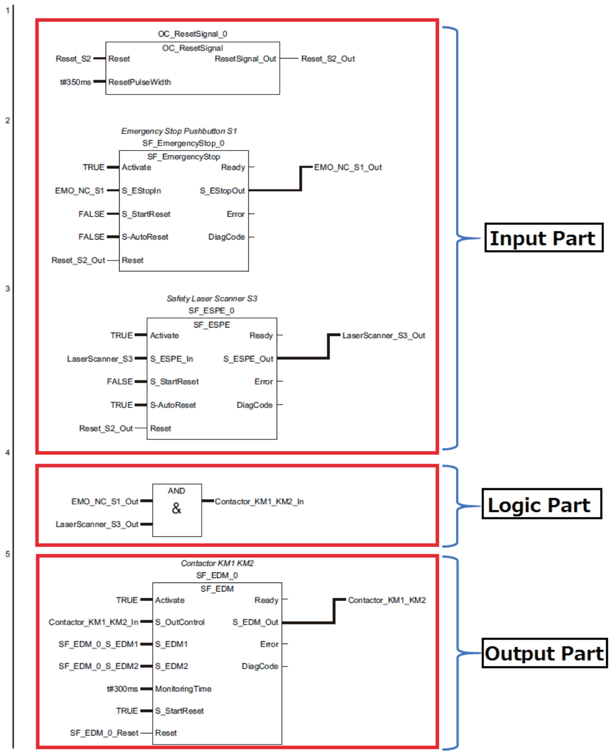


Fig. 1 Program created by automatic programing

4. Method for Realizing Automatic Programing Function

4.1 Generation of input part

In the input part, processing including the loading of variables that are assigned to input signals and start/restart interlock by using the function block for safety input defined by the safety function blocks of PLCopen are executed.

However, the function block for safety input of the PLCopen safety function blocks does not support the start/restart interlock triggered by the pulse reset signals. For realization by the program, it is necessary to consider signal timing and build the logic to combine multiple function blocks.

Taking the above into consideration, we developed the user-defined function block "OC_ResetSignal" that can deal with pulse reset signals. The term "user-defined function block" implies the function block wherein a combination of multiple function blocks is formed into a component for use with users' programs. Fig. 2 shows the external appearance of the user-defined function block "OC_ResetSignal."

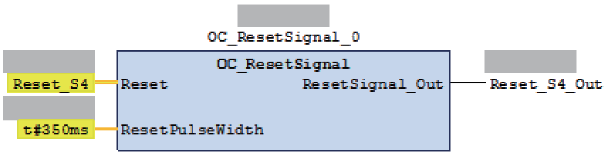


Fig. 2 User-defined function block "OC_ResetSignal"

The internal logic of the "OC_ResetSignal" is not disclosed. However, with the automatic programming function of Sysmac Studio, the start/restart interlock is realized by automatically combining the user-defined function block and the safety function blocks of PLCopen.

In addition, because the function block for safety input has many variations of parameters and setup items, designers are requested to choose the parameters and input adequate values.

Therefore, to make parameter setup of the function block easier, we executed the parameter analysis which is to be described later and narrowed down the parameters for which setting up is required. The parameters that are required for the function block for safety input could be narrowed down to three parameters as shown in Table 2.

Table 2 Parameters required for function block for safety input

Parameter	Description
Safety Input	Input signals of safety switch/safety sensor, etc.
Auto Reset	Parameter to switch Enabled/Disabled of restart interlock.
Reset	Reset signal when restart interlock is Enabled.

We analyzed the output parameters and the input parameters from the viewpoints shown below and narrowed down the required parameters.

・Viewpoints

– Setup by designer is required.

– Setup is required, but setup by designer is not required by enabling automatic setup.

– No setup is required.

First, Table 3 shows the analysis results of output parameters.

Table 3 Analysis results of output parameters

Parameter	Description of Analysis	Analysis Result
Parameters that are not related to safety control	Output parameters for monitoring, including error flags and error codes. The parameters are not set up since they are not related to the operation of the function block or safety control.	No setting up is required.
Safety Output	The output is connected to the logic operating part as the safety control signal. The internal variables are automatically assigned.	Automatic setup

Next, Table 4 shows the analysis results of the input parameters.

Table 4 Analysis results of input parameters

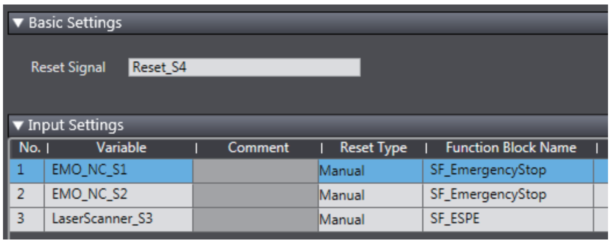
Parameter	Description of Analysis	Analysis Result
Input parameters that are not used for general use applications	Parameters including the operation start condition flag of the function block. The constants are assigned automatically.	Automatic setup
Safety Input	Input signal of safety switch/safety sensor, etc. The variables are assigned by the designer.	Subject to setup
Auto Reset	Parameter to switch Enabled/Disabled of restart interlock. The parameter is switched by the designer according to the setup.	Subject to setup
Reset	Reset signal when restart interlock is Enabled. The variables are assigned by the designer.	Subject to setup

The three parameters that are narrowed down by the aforementioned analysis and the types of function blocks for safety input are defined as shown in Table 5 as the setup items of the input part of the automatic programing function of Sysmac Studio.

Table 5 Setup items of automatic programing function (input part)

Setup Item	Related Parameter	Description
Reset Signal	Reset	Variable that is assigned to the reset signal when the restart interlock is enabled.
Variable	Safety Input	Variable that is assigned to input signals of safety switch/safety sensor, etc.
Reset Type	Auto Reset	Selection of Manual Reset (restart interlock function is enabled) or Auto Reset
Function Block Name	—	Type of function block for safety input

Fig. 3 shows the parameter setup window of the input part of Sysmac Studio.



▼ Basic Settings

Reset Signal: Reset_S4

▼ Input Settings

No.	Variable	Comment	Reset Type	Function Block Name
1	EMO_NC_S1		Manual	SF_EmergencyStop
2	EMO_NC_S2		Manual	SF_EmergencyStop
3	LaserScanner_S3		Manual	SF_ESPE

Fig. 3 Parameter setup window of input part

The program of the input part that is generated from the aforementioned setup items is shown in Fig. 4.

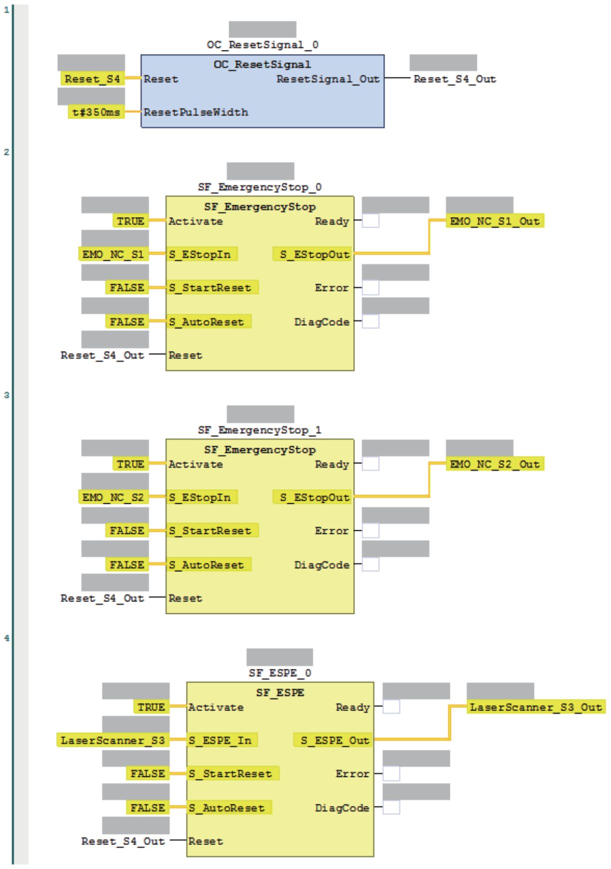


Fig. 4 Generated program of input part

An internal variable is set for the safety signal output from each function block for safety input and is used for connection with the logic operating part which is to be described later. In addition, the OC_ResetSignal that deals with the pulse reset signal is arranged at the head, and an internal variable is set for the output parameters thereof. Setting up of the internal variable to the Reset input parameter of each function block for safety input realizes the start/restart interlock by using the pulse reset signal.

4.2 Generation of output part

In the output part, the safety control signal is output for the variable that is assigned to the output signal, thereby implementing external device monitoring (EDM).

Likewise for the input part, to make the parameter setup of the function block easier, we implemented the parameter analysis which is to be described later and narrowed down the parameters for which setting up is necessary. The parameters that are required for the function block for safety output could be narrowed down to one parameter as shown in Table 6.

Table 6 Parameter required for function block for safety output

Parameter	Description
Safety Output	Output signal of the safety relay that controls the power supply

Also in the output part, necessary parameters were narrowed down by implementing the analysis from the same point of view as the input part.

First, Table 7 shows the analysis results of the output parameters.

Table 7 Analysis results of output parameters

Parameter	Description of Analysis	Analysis Result
Parameters that are not related to safety control	Output parameters for use with monitoring, including error flags and error codes. The parameters are not set up since they are not related to the operation of the function block or safety control.	Setup not required
Safety Output	Output signals of the safety relay that controls the power supply. The variables are assigned by the designer.	Subject to setup

Next, Table 8 shows the analysis results of the input parameters.

Table 8 Analysis results of input parameters

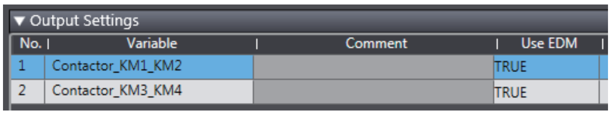
Parameter	Description of Analysis	Analysis Result
Input parameters that are not used for general use applications	Parameters including the operation start condition flag of the function block. The constants are assigned automatically.	Automatic setup
Safety Input	The input is connected to the logic operating part as the safety control signal. The internal variables are automatically assigned.	Automatic setup
EDM Input	External device monitoring signal. To be changed by the designer later. The internal variables are automatically assigned.	Automatic setup
EDM Reset	Parameter to delete EDM error. To be changed by the designer later. The internal variables are automatically assigned.	Automatic setup

The one parameter narrowed down through the aforementioned analysis and the use of the function block SF_EDM for safety output was defined in Table 9 as the setup of the automatic programming output part of Sysmac Studio.

Table 9 Setup items of automatic programming function (output part)

Setup Item	Related Parameter	Description
Variable	Safety Output	Variable that is assigned to the output signal of the safety relay which controls the power supply.
Use EDM	—	Designation of use of the function block "SF_EDM" for monitoring of external device.

Fig. 5 shows the parameter setup window of the output part of Sysmac Studio.



No.	Variable	Comment	Use EDM
1	Contactor_KM1_KM2		TRUE
2	Contactor_KM3_KM4		TRUE

Fig. 5 Parameter setup window of output part

The program of the output part generated from the aforementioned setup items will be like the one shown in Fig. 6.

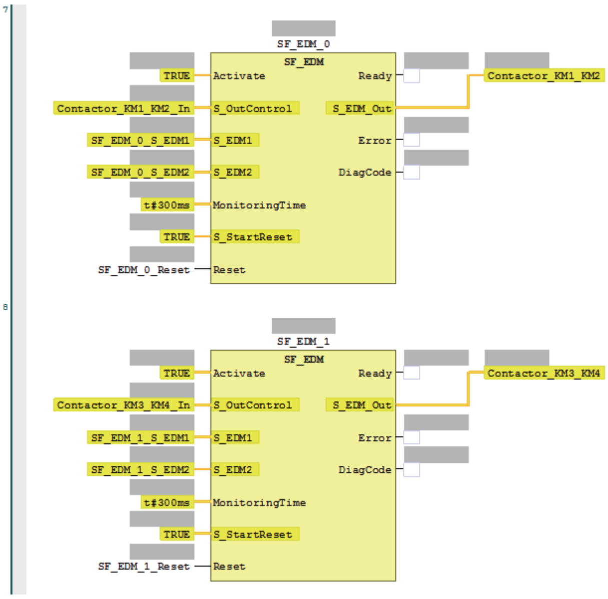


Fig. 6 Generated program of output part

Internal variables are set to the input parameter S_OutControl of the SF_EDM function block which is the safety control signal and are used for connection with the logic operating part which is to be described later. When external device monitoring (EDM) is not used, output is made to the variable that is directly designated in the logic operating part not via the SF_EDM function block.

4.3 Generation of logic operating part

The logic operating part is the logic circuit that connects the input part and the output part. This time, we developed the technology to generate the logic circuit based on the table in which expected values of the safety output when the safety input is turned off are stated.

The logical operation of the safety control signal consists of the logical AND (AND function) of the input part and the output part. Then, by utilizing the design document which the safety designer owns, the logic operating part is automatically generated.

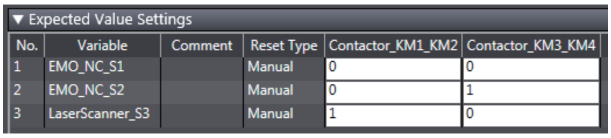
Table 10 shows an example of the table, which is the design document owned by the safety designer, showing the relationship between the safety input and the safety output.

Table 10 Example of table showing relationship between safety input and safety output

		Safety Output		
		Contactor KM1_KM2	Contactor KM3_KM4	…
Safety Input	Emergency stop switch S1	OFF	OFF	…
	Emergency stop switch S2	OFF	ON	…
	Laser scanner S3	ON	OFF	…
	:	:	:	

OFF: Disconnection; ON: No Disconnection

Fig. 7 shows the setup window of the expectation value of the Sysmac Studio automatic programming function that is set up based on the information from this table.


▼ Expected Value Settings

No.	Variable	Comment	Reset Type	Contactor_KM1_KM2	Contactor_KM3_KM4
1	EMO_NC_S1		Manual	0	0
2	EMO_NC_S2		Manual	0	1
3	LaserScanner_S3		Manual	1	0

Fig. 7 Expected value setting window of automatic programming

From the expected values, the safety inputs corresponding to the respective safety outputs are extracted and are then connected with the AND function, thereby realizing the logic operating part as shown in the following Fig. 8.

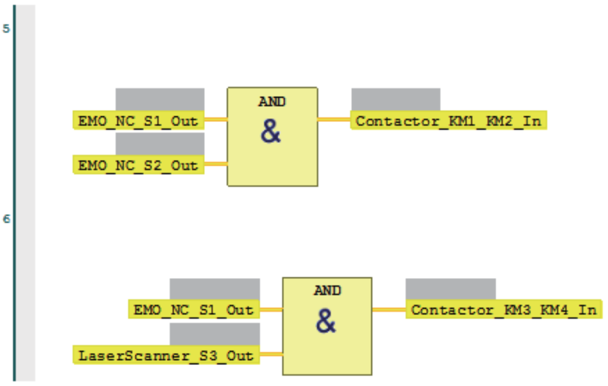


Fig. 8 Generated program of logic operating part

Referring to this simple configuration, designers will be able to create the logical operation circuit without mistakes. For more complicated configurations, however, it can be said that such automatization will work effectively because a lot of simple tasks may cause careless mistakes.

4.4 Advantage of separating logic operating part from input and output parts

The method for automatically generating the program by dividing it into the “input part,” “output part” and “logic operating part” is described above. It can be said that division of the program into those three blocks can enhance the expandability and readability of the program.

For example, referring to the example of a general safety system wherein multiple safety inputs affect one safety output, the program for cases where the logic operating part is separated is shown in Fig. 9, and the program for cases where the logic operating part is not separated is shown in Fig. 10.

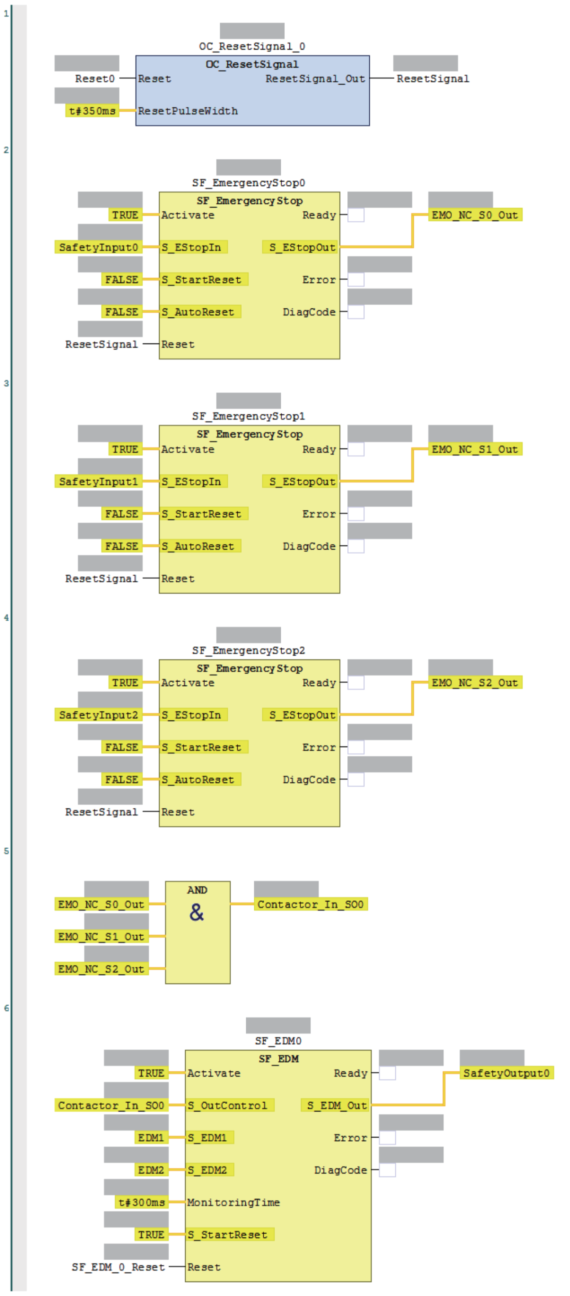


Fig. 9 Example of program where logic operating part is separated

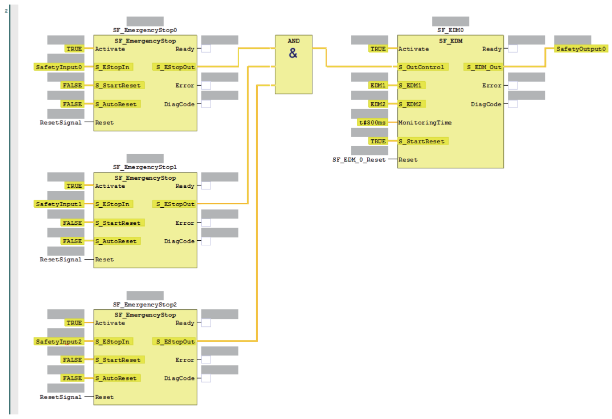


Fig. 10 Example of program where logic operating part is not separated

The programs that must be added when one safety output is added to the respective programs are shown in Fig. 11 and Fig. 12.

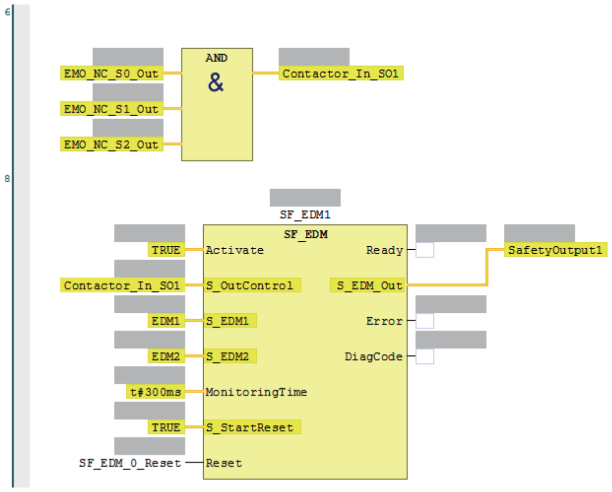


Fig. 11 Program to be added when logic operating part is separated

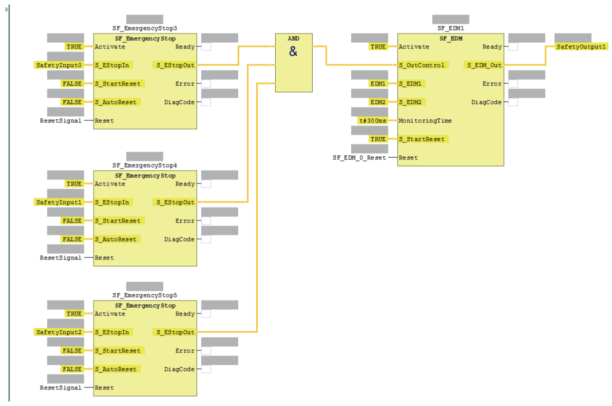


Fig. 12 Program to be added when logic operating part is not separated

For cases where the logic operating part is separated, only the addition of the logic operating part and the function block for safety output are required as shown in Fig. 11.

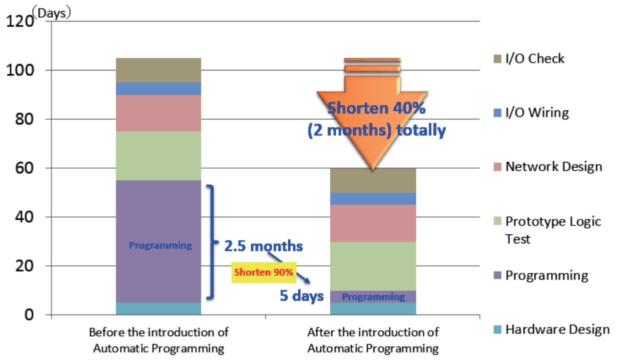
On the other hand, for cases where the logic operating part is not separated, it is necessary to duplicate the entire program, set up the function block for safety input and add the function block for safety output as shown in Fig. 12.

Therefore, it can be said that the case of separating the logic operating part by using an internal variable enhances the expandability and readability because the program to be added is simple and the amount of change is little.

4.5 Validation of effect

This time, we realized the automatic generation of programs with high expandability and readability through minimum necessary simple setup by using the automatic programing function added to Sysmac Studio.

Furthermore, we implemented validation within the company regarding the introduction effect for cases where the automatic programing function of Sysmac Studio is used. The results revealed that, as shown in Fig. 13, the time required for creating a program for the model case of a certain safety system could be shortened by 90%, thereby ensuring the effect to shorten the entire development period of device design by 40%.



Note: Programming part contains the desk debug and fixing.

Fig. 13 Introduction effect of automatic programing function

5. Conclusion

In this paper, we described the technical achievements regarding the automatization of safety programming by utilizing the safety function blocks of PLCopen.

We confirmed that the automatic generation of safety program that is of an equivalent level to that of experienced designers from the minimum necessary setup items by utilizing the technology to narrow down necessary parameters and the logical circuit generation technology offers the effect of shortening the development period of a device. Furthermore, it was made possible to establish a flexible safety system by defining a safety program with high expandability and readability that can deal with increased/decreased I/O points by separating the logic operating part and incorporating the safety program in the automatic programing function.

In the future, we will further advance the automatic programing function through the reception of feedback from the market and continue to work on technical innovations that can deal with increasingly diversified customer needs.

Finally, regarding the realization of the automatic programing function of Sysmac Studio, we would like to express our deep appreciation for those persons from the business planning, consulting and marketing divisions, as well as those who were involved in development, for their considerable cooperation.

References

1) Ministry of Economy, Trade and Industry. White Paper on Manufacturing Industries (Monozukuri) 2018 -Part 1 Current Status and Challenges of Manufacturing Infrastructure Technology. https://www.meti.go.jp/report/whitepaper/mono/2018/honbun_pdf/pdf/honbun01_01_02.pdf, (accessed 2018-12-6).

2) IEC 61508:2010 Functional safety of electrical/electronic/programmable electronic safety-related systems.

3) ISO 13849-1:2015 Safety of machinery -- Safety-related parts of control systems -- Part 1: General principles for design.

About the Authors

SUGANUMA Hiromu

Software Development Dept. Controller Div.
Product Business Division HQ.
Industrial Automation Company
Speciality: Software Engineering

FUJIMURA Ryosuke

Software Development Dept. Controller Div.
Product Business Division HQ.
Industrial Automation Company
Speciality: Software Engineering

KUBO Hiroko

Software Development Dept. Controller Div.
Product Business Division HQ.
Industrial Automation Company
Speciality: Software Engineering

TAKUMA Nobuyuki

Controller Div.
Product Business Division HQ.
Industrial Automation Company
Speciality: Software Engineering

The names of products in the text may be trademarks of each company.