






**OMC System Software  
High-performanceHMI  
SFCEX Program  
User Manual  
IM41S93-E**

| Notices |  |
|---------|--|
|         | <ul style="list-style-type: none"> <li>● The reproduction, transmission or use of this document or its contents is not permitted without express written authority.</li> <li>● Information and specifications in this document are subject to change without notice.</li> <li>● While information in this document is well edited and checked, mistake or omission may exist. Please don't hesitate to contact SUPCON if you have any question about this document.</li> <li>● Please contact SUPCON via email "SMS@supcon.com" if you have any question.</li> </ul> |

| Trademarks |  |
|------------|--|
|            | <p>Trademarks or marks SUPCON, SPlant, Webfield, ESP-iSYS, MultiF, InScan, SupField are all registered, registering or using by Zhejiang SUPCON Technology Co., Ltd., which owns the properties of all trademarks or marks above. Without the written authority from Zhejiang SUPCON Technology Co., Ltd, no individual or company shall use any trademarks or marks above. We reserve the right to take legal action for any individual or company using trademarks or marks above illegally.</p> |

| Symbol Definition   |   |
|---|---|
|  | <b>WARNING:</b> Indicates information that a potentially hazardous situation which, if not avoided, could result in serious injury or death.  |
|  | <b>RISK OF ELECTRICAL SHOCK:</b> Indicates information that Potential shock hazard where HAZARDOUS LIVE voltages greater than 30V RMS, 42.4V peak, or 60V DC may be accessible.           |
|  | <b>ESD HAZARD:</b> Indicates information that Danger of an electro-static discharge to which equipment may be sensitive. Observe precautions for handling electrostatic sensitive devices |
|  | <b>ATTENTION:</b> Identifies information that requires special consideration.   |
|  | <b>TIP:</b> Identifies advice or hints for the user.  |

# Table of Contents

|   |           |
|---|-----------|
| <b>SFCEX Program User Manual.....</b>                     | <b>1</b>  |
| <b>Section 1 Introduction.....</b>                        | <b>1</b>  |
| 1.1 Program Status Description.....                       | 1         |
| 1.1.1 Status Switching Rule.....                          | 1         |
| 1.1.2 Status Transition Figure.....                       | 2         |
| 1.2 Technical Specifications.....                         | 2         |
| 1.3 Program Composition.....                              | 3         |
| 1.3.1 Basic Elements.....                                 | 3         |
| 1.3.2 Program Structure .....                             | 5         |
| 1.3.3 Program Running Rule.....                           | 6         |
| <b>Section 2 Process Diagram .....</b>                    | <b>7</b>  |
| <b>Section 3 Create and configure SFCEX Program .....</b> | <b>8</b>  |
| <b>Section 4 Configure Logic of Program .....</b>         | <b>11</b> |
| 4.1 Configure FailMonitor .....                           | 11        |
| 4.1.1 Configuration Steps .....                           | 11        |
| 4.1.2 Configuration Result.....                           | 12        |
| 4.1.3 Exception Lifted.....                               | 13        |
| 4.2 Configure Transient Program .....                     | 13        |
| 4.2.1 Add and Configure Step .....                        | 13        |
| 4.2.2 Add and Configure Transition Condition .....        | 16        |
| 4.2.3 Add and Configure Branch .....                      | 20        |
| 4.2.4 Connect Objects.....                                | 21        |
| 4.2.5 Align Graphical Objects .....                       | 22        |
| 4.2.6 Add Annotation .....                                | 22        |
| 4.3 Configure Message .....                               | 22        |
| 4.3.1 Configure Message .....                             | 23        |
| 4.3.2 Apply Message .....                                 | 26        |
| 4.4 Display Detailed Content .....                        | 30        |
| <b>Section 5 Compile and Debug .....</b>                  | <b>31</b> |
| 5.1 Compile .....   | 31        |

|  |           |
|--|-----------|
| 5.2 Debug.....   | 31        |
| 5.2.1 Enter Debugging Interface .....                        | 31        |
| 5.2.2 Debug Break Point .....                                | 33        |
| <b>Section 6 Monitor SFCEX Program.....</b>                  | <b>35</b> |
| 6.1 Monitor Tag Panel of SFCEX.....                          | 35        |
| 6.2 Monitor Fail Alarm.....                                  | 36        |
| 6.3 Monitor SFCEX Message .....                              | 36        |
| 6.3.1 Illustration of Message .....                          | 37        |
| 6.3.2 Operation Instruction of Message.....                  | 38        |
| <b>Section 7 Description of SFCEX Tag Parameter .....</b>    | <b>40</b> |
| <b>Section 8 Appendix Description of SFCEX Language.....</b> | <b>42</b> |
| 8.1 ST Function Key Words Supported .....                    | 42        |
| 8.2 ST Function Operator Supported.....                      | 42        |
| 8.3 Timer .....  | 43        |
| 8.4 Common Array .....                                       | 43        |
| <b>Section 9 Revision.....</b>                               | <b>44</b> |

# SFCEX Program User Manual

## Section 1 Introduction

---

SFCEX program adheres to ISA S88.01 standard, and uses SFCEX to write steps and transition conditions. Steps are used to issue production-related control instructions, such as opening field valves. Transition conditions are used to determine whether next production can be started, and when satisfied, start the next production. Every time SFCEX scans, the system will judge activation step and transition condition. When transition condition is satisfied, steps before transition condition will not be activated, steps after transition condition will become active.

### 1.1 Program Status Description

In High-performanceHMI system software, the SFCEX program supports three statuses: static, transient and FailMonitor, SFCEX can perform different operations in different status.

- Static

Static status refers to the status that program enters after normal execution.

Static status includes Idle, Complete, Aborted, Paused, Held and Stopped.

- Transient

Transient status includes Running, Stopping, Pausing, Holding, Restarting, Aborting, it is used to configure logic related to field production operation, corresponding to Operation in ISA S88.01.

- Exception process status FailMonitor

It is used to monitor program status. When program is in Running, Pausing and Restarting, and after FailMonitor has detected a program operation fault, the program will enter Holding status. Program cannot enter Restarting status until fault is removed from Holding status.

#### 1.1.1 Status Switching Rule

Under normal conditions, status will change automatically by the order: IDLE > RUNNING > COMPLETE. When you need to debug the program, you can switch status manually through the command buttons on the panel of SFCEX program tag. Status switching is controlled by command Start, Stop, Hold, Pause, Restart, Abort, Resume and Reset.

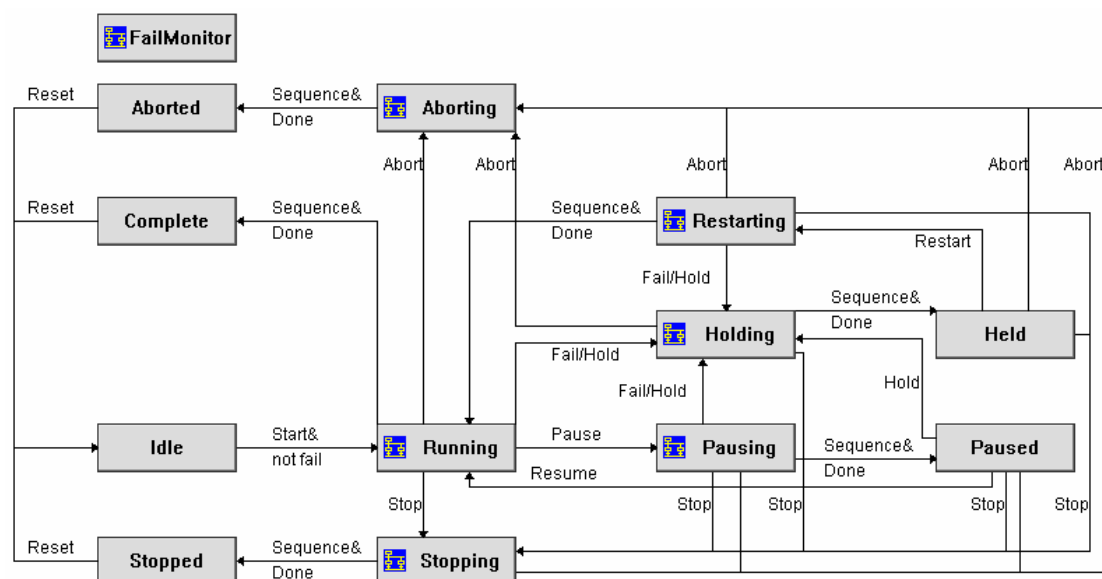
The status switching rule of SFCEX program is listed in the table below.

**Table 1-1 Status switching rule of SFCEX program**

| Command    |          | Start                   | Stop     | Hold    | Restart    | Abort    | Reset | Pause   | Resume  |
|------------|----------|-------------------------|----------|---------|------------|----------|-------|---------|---------|
| IMAN       | End mode | Status switching matrix |          |         |            |          |       |         |         |
| Idle       | —        | Running                 | —        | —       | —          | —        | —     | —       | —       |
| Running    | Complete | —                       | Stopping | Holding | —          | Aborting |       | Pausing |         |
| Complete   | —        | —                       | —        | —       | —          | —        | Idle  | —       | —       |
| Pausing    | Paused   | —                       | Stopping | Holding | —          | Aborting | —     | —       | Running |
| Paused     | —        | —                       | Stopping | Holding | —          | Aborting | —     | —       | —       |
| Holding    | Held     | —                       | Stopping |         | —          | Aborting | —     | —       | —       |
| Held       | —        | —                       | Stopping |         | Restarting | Aborting | —     | —       | —       |
| Restarting | Running  | —                       | Stopping | Holding |            | Aborting | —     | —       | —       |
| Stopping   | Stopped  | —                       | —        | —       | —          | —        | —     | —       | —       |
| Stopped    | —        | —                       | —        | —       | —          | Aborting | Idle  | —       | —       |
| Aborting   | Aborted  | —                       | —        | —       | —          |          |       | —       | —       |
| Aborted    | —        | —                       | —        | —       | —          |          | Idle  | —       | —       |

### 1.1.2 Status Transition Figure

State transition figure as shown below illustrates all the statuses of SFCEX program and related commands of status switching.



**Figure 1-1 Status transition**

## 1.2 Technical Specifications

The technical specifications are listed in Table 1-2.

**Table 1-2 Technical specifications of SFCEX program**

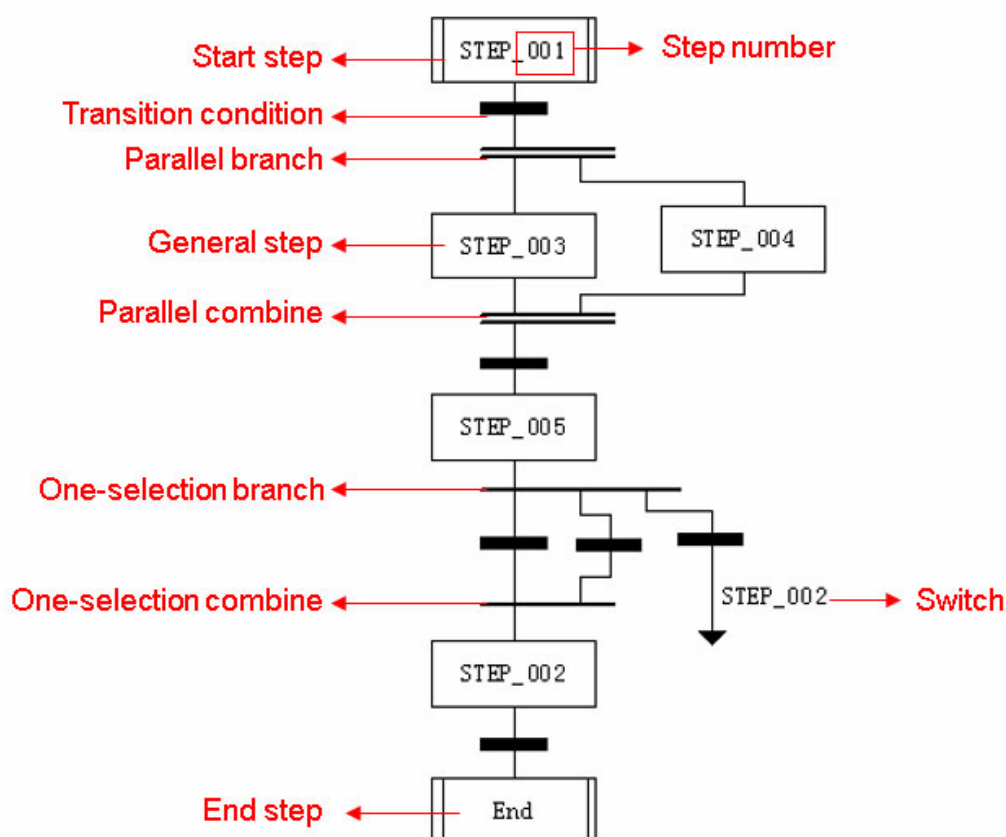
| Description                           | Value |
|---------------------------------------|-------|
| Maximum steps supported by RUNNING    | 256   |
| Maximum steps supported by RUNNING    | 256   |
| Maximum steps supported by HOLDING    | 128   |
| Maximum steps supported by HOLDING    | 128   |
| Maximum steps supported by PAUSING    | 128   |
| Maximum steps supported by PAUSING    | 128   |
| Maximum steps supported by RESTARTING | 128   |
| Maximum steps supported by RESTARTING | 128   |
| Maximum steps supported by ABORTING   | 128   |
| Maximum steps supported by ABORTING   | 128   |
| Maximum steps supported by STOPPING   | 128   |
| Maximum steps supported by STOPPING   | 128   |
| Conditions supported by transition    | 8     |
| Maximum info                          | 64    |
| Maximum bars supported by FailMonitor | 64    |

## 1.3 Program Composition

SFCEX program in High-performanceHMI system software supports sequence diagrams with multiple structures, which consist of step, branch, transition, etc.

### 1.3.1 Basic Elements

The basic elements of SFCEX programs include start step, step number, transition conditions, select branch-select combine, parallel branch-parallel combine, end step and link line, as shown in the figure below. SFCEX program can include the following elements, but not every element is required.



**Figure 1-2 Basic elements of SFCEX**

The typical operation process of SFCEX program as shown in the figure above follow these rules:

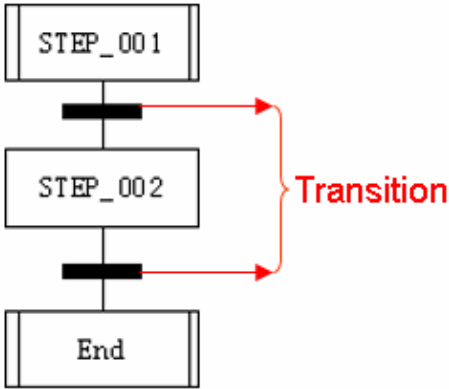
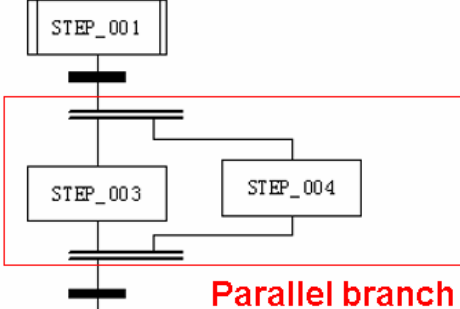
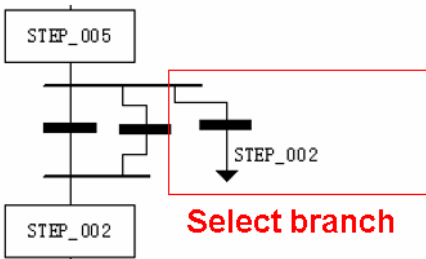
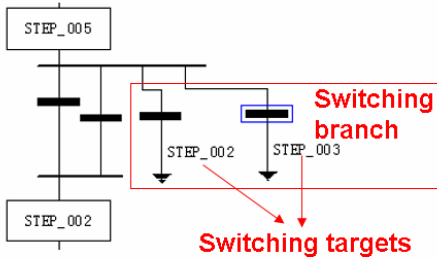
- Program starts from step  
Step supports three statuses, initialization (inactive, unexecuted), active and completed.  
Initialization means that step is not active and has not been executed, active refers to the status of command or code being executed, completed means that step is already executed and no longer needs executing.
- Judgment of transition condition after step  
Judgment starts when the previous step of transition condition is activated, and when the whole transition condition passes, it will record the establishment status of each expression in transition condition and stop updating the satisfaction status of the transition condition, which means the satisfaction status of transition condition would become historical data.
- Switch of step

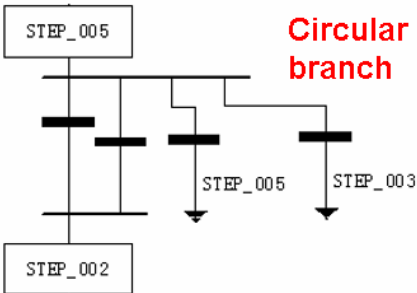


During program execution, you can switch step through the panel or the switching step configured in transition configuration. The switching step should be the parallel branch of the currently active step.

### 1.3.2 Program Structure

Program structure is divided into sequential structure, parallel branch, select branch and circular branch, as shown in the table below.

| Structure Diagram   | Structure Name  |
|---|---|
|   | <p>Sequential structure consists of step and transition.</p> <p>Step is used for production-related operation, such as STEP_001 and STEP_002 as shown in the left figure.</p> <p>Transition is used to configure transition condition to enter next step.</p>   |
|  | <p>Parallel branch can be executed simultaneously in the same period, and their execution is independent of each other.</p> <p>Only when all steps at the end of the parallel branch path are active, and its transition condition is satisfied, the transition condition after the parallel branch can be executed.</p>  |
|  | <p>Select branch, normally added after step.</p> <p>The execution order of select branch is to execute the branch whose transition condition is satisfied first.</p> <p>If more than one transition condition are satisfied at the same time, the leftmost transition in SFCEX graphics will be activated.</p>  |
|  | <p>Switching branch consists of an initial transition condition and an arrow sign specifying the switching target.</p> <p>If the transition condition of switch is satisfied, this switch will be executed. If there are more than one switch at the source location of switch, the switch whose transition condition is satisfied first will be executed. If more than one transition condition is satisfied at the same time, the leftmost transition will be activated.</p> <p>In switching structure, switch is only supported at the end of select branch, and never into a parallel branch.</p> |

| Structure Diagram   | Structure Name  |
|---|---|
|  | <p>Circular branch is a special case of switching branch, its switching target is the previous step. As shown in the left figure, the switching target in circular branch is the previous step of the branch "005".</p> |

### 1.3.3 Program Running Rule

- SFCEX program supports auto mode and manual mode.  
In auto mode, step will be immediately executed after activation.  
In manual mode, step will be executed after activation, and wait for "Continue" command to judge the next transition condition. When the step execution is completed, if the next transition condition of step is satisfied, the step converted will be activated and wait for "Continue" command. If the next transition condition of step is not satisfied, it will keep the current step active and wait for satisfaction of transition condition.
- When transition condition is not satisfied, the currently active step will be executed in every scanning period, until transition condition is satisfied and then current active step will fail, and the next step activated. If you select "Executed only once" (set in step properties), the step will only be scanned in the first running period in which program is called, and not thereafter.
- In select branch, it will execute along the branch whose transition condition is satisfied first and other branches will not be executed. When there are more than two select branches whose transition conditions are satisfied, the leftmost branch will be executed first.
- In parallel branch, when the transition condition before the branch is satisfied, all the branches under this parallel branch will start running at the same time, and when all the branches finish running and transition conditions are satisfied, parallel branch will finish running.
- When transition condition before switch is satisfied, the program logic will switch directly to target step, and continue to execute from this step. When transition condition is satisfied, program turns to the next step. When transition condition is not satisfied, every control cycle needs to scan this transition.

## Section 2 Process Diagram

When the controller in OMC system is FCU713-S, you can create SFCEX program through VFExplorer, and edit SFCEX program through VFSFCEXBuilder, the detailed configuration flow is as follows.

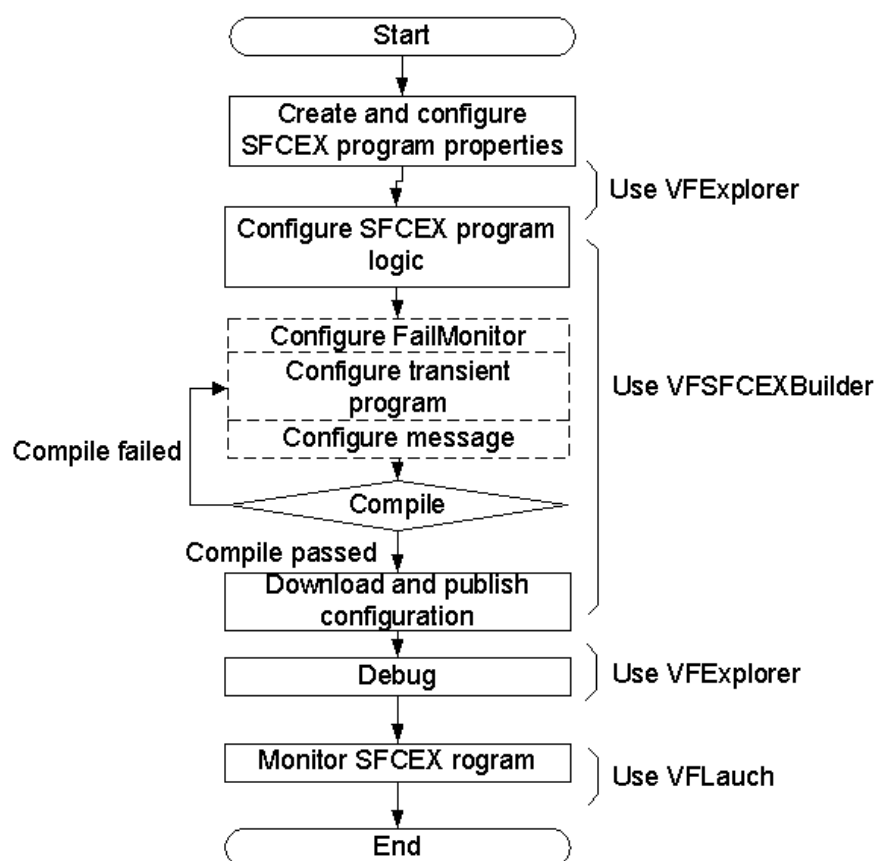


Figure 2-1 SFCEX program configuration diagram of High-performanceHMI software

## Section 3 Create and configure SFCEX Program

Before starting SFCEX programming, you need to create SFEX program.

- 1) Open VFExplorer.
- 2) Select "Control Station> user program" in project tree, and select "New" in its right-click menu.
- 3) In pop-up dialog box, select "SFCEX" in drop-down list "Type", and configure "Name" and "Description" for SFCEX program.
  - Program name supports up to 64 characters, cannot include "`"\\"V,;:<>|?*\%\\n`".
  - Program description supports up to 128 characters.
  - Cycle supports fast cycle, one time, two times, five times and ten times cycle multiple, the default is one time.
  - Start Phase: 0~ 9, the default is 0.
- 4) Double-click the newly-created SFCEX program to open VF\_SFCEXBuilder program, and program configuration interface is shown as the figure below.

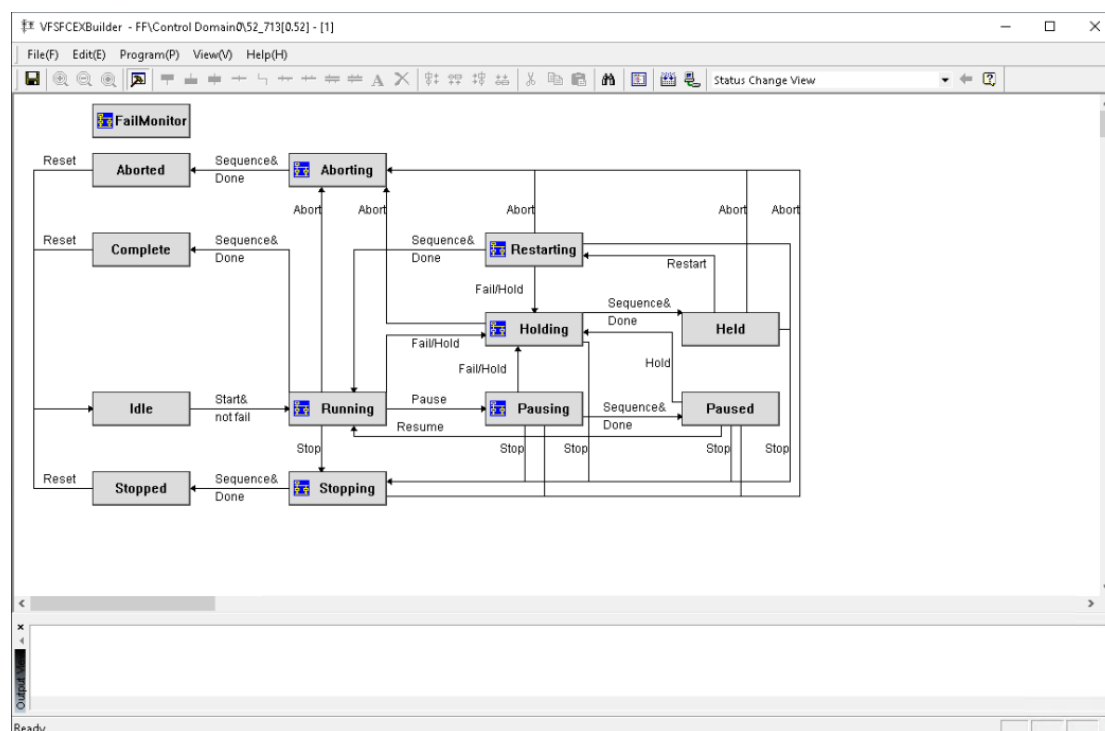
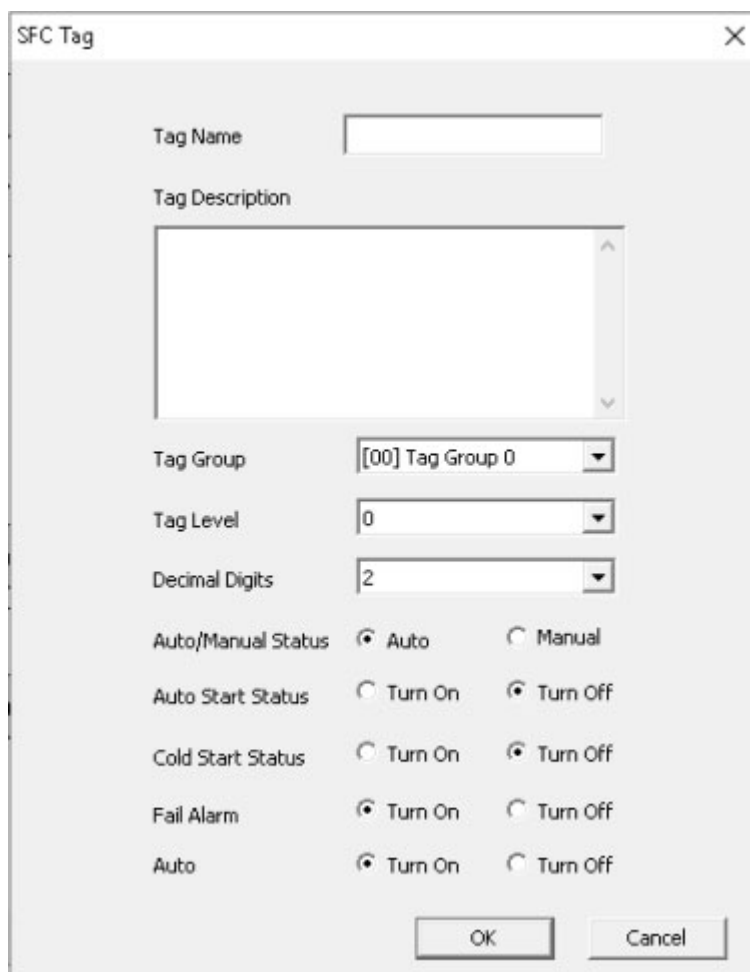


Figure 3-1 VF\_SFCEXBuilder initial interface

- 5) Configure tag name of SFCEX

- 6) Select “Edit > Property” in menu to pop up “SFC tag” configuration dialog box, as shown in the figure below.



The image shows a dialog box titled "SFC Tag" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Tag Name:** A text input field.
- Tag Description:** A large text area with a vertical scrollbar.
- Tag Group:** A dropdown menu showing "[00] Tag Group 0".
- Tag Level:** A dropdown menu showing "0".
- Decimal Digits:** A dropdown menu showing "2".
- Auto/Manual Status:** Two radio buttons, "Auto" (selected) and "Manual".
- Auto Start Status:** Two radio buttons, "Turn On" and "Turn Off" (selected).
- Cold Start Status:** Two radio buttons, "Turn On" and "Turn Off" (selected).
- Fail Alarm:** Two radio buttons, "Turn On" (selected) and "Turn Off".
- Auto:** Two radio buttons, "Turn On" (selected) and "Turn Off".

At the bottom right, there are "OK" and "Cancel" buttons.

**Figure 3-2 SFC Tag configuration interface**

- 7) Configure SFCEX program properties according to the following table.

| Item            | Function Description   |
|-----------------|--|
| Tag Name        | <ul style="list-style-type: none"> <li>Used to specify the current program tag Name. In real-time monitoring, you can control SFCEX program through the tag monitor panel of SFCEX program.</li> <li>Maximum of 24 characters are supported, and can only be composed of number, letter, "-", and "_", cannot be repeated with the function block tag name in station.</li> </ul>  |
| Tag Description | Used to configure the description of current program tag, supporting 64 characters maximum.  |
| Tag Group       | Used to specify the tag group of the program tag.  |
| Tag Level       | Tag level is divided into 10 levels, 0~9 (0 lowest, 9 highest), tag level can be selected through drop-down box. Tag level is used to determine whether secondary confirmation is required when changing value in tag panel. You can set a batch of important tag to a certain level, and select that level to require secondary confirmation when changing value in the panel in "security settings" of System Builder. |

| Item               | Function Description   |
|--------------------|--|
| Decimal Digits     | You can select decimal digits, ranging from 0~7, through drop-down box.  |
| Manual/Auto Status | Used to specify the initial manual/auto status of the program <ul style="list-style-type: none"> <li>● If you select "Auto", the program will run automatically after start.</li> <li>● If you select "Manual", you need to manually trigger it.</li> </ul>  |
| Auto Start Status  | Used to specify whether the program rerun after running to complete status, on valid in manual status. <ul style="list-style-type: none"> <li>● If you select "Turn on", the program will automatically enter RUNNING status and rerun after running to COMPLETE.</li> <li>● If you select "Turn off", the program will remain as COMPLETE after running to COMPLETE.</li> </ul>   |
| Cold Start Status  | Used to specify whether the program continues to run after controller cold reset. <ul style="list-style-type: none"> <li>● If you select "Turn on", the program will enter IDLE status after controller cold reset.</li> <li>● If you select "Turn off", the program will remain as current status after controller cold reset.</li> </ul>   |
| Fail Alarm         | Used to specify whether the program generates the process alarm of the program tag after entering FailMonitor status. <ul style="list-style-type: none"> <li>● If you select "Turn on", the program will generate the process alarm of the program tag after entering FailMonitor status.</li> <li>● If you select "Turn off", the program will not generate the process alarm of the program tag after entering FailMonitor status.</li> </ul>                    |
| Auto Recovery      | Used to specify whether the program recover the execution of program after entering Held status. <ul style="list-style-type: none"> <li>● If you select "Turn on", the program will automatically enter Restarting status after entering Held status, and will start executing from first step.</li> <li>● If you select "Turn off", you need to manually trigger Restart command after the program enter Held status, and the program will re-execute.</li> </ul> |

8) Click "OK" to complete the creation and configuration of SFC EX program.

## Section 4 Configure Logic of Program

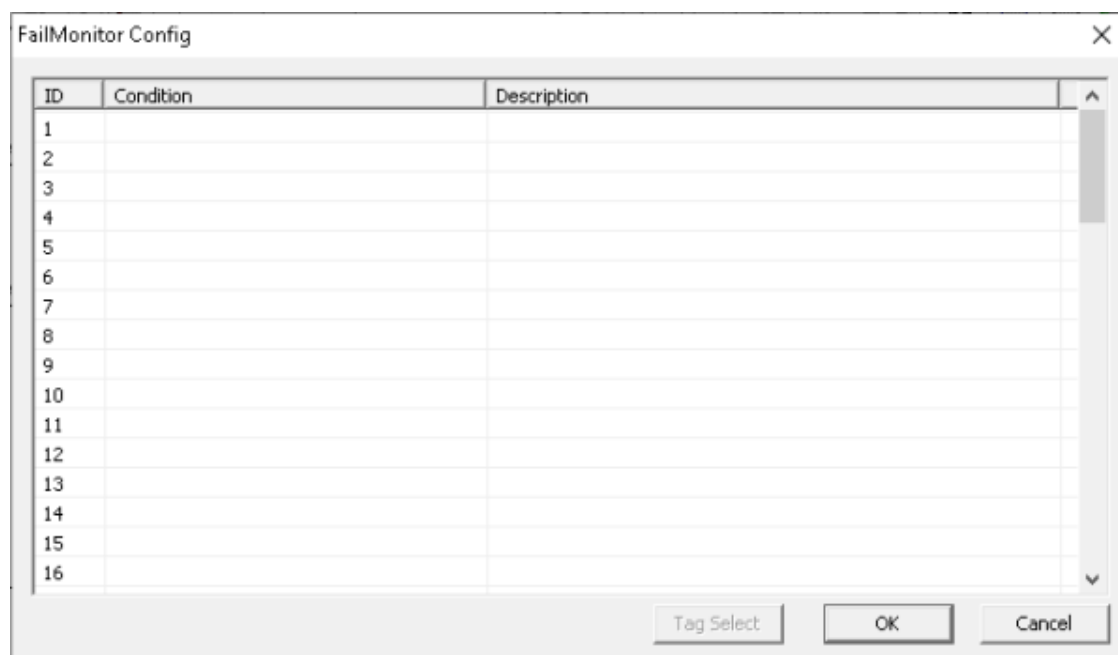
When configuring logic of SFCEX program, you need to specify the judgment condition for FailMonitor status of the program, step and transition condition of transient program, message of the program and etc.

### 4.1 Configure FailMonitor

FailMonitor is used to configure the judgment condition of exception process status, High-performanceHMI system supports a maximum of 64 FailMonitor.

#### 4.1.1 Configuration Steps

Double-click “FailMonitor” in the configuration interface of SFCEX program to pop up the configuration interface as shown in the figure below, and you can configure the condition of exception monitoring in it.



**Figure 4-1 Configuration interface of FailMonitor**

Configure message properties of exception monitoring according to following table.

| Item        | Configuration Description   |
|-------------|---|
| ID          | Number of the FailMonitor   |
| Condition   | Used to specify the judgment condition of the FailMonitor, you can enter judgment condition through “tag selection” after specifying tag. |
| Fail Value  | Used to specify the custom abnormal value of the FailMonitor, ranging from 1~255.   |
| Description | Used to specify the other description of FailMonitor.   |

Click “Exception Process” to configure exception condition for this message. When configuring exception condition, you should relate condition to message ID, the format of relation is “FAIL\_IDX:=n”, n is Fail value.

#### 4.1.2 Configuration Result

As shown in the figure below, create two exception message, ID is 1 and 2 respectively. Define the judgment conditions for abnormal message ID of 1 and 2 respectively in “exception process”. Exception message 1 is generated under the condition “AI04020000.PV > 50.0”, exception message 2 is generated under the condition “AI04020001.PV < 20.0”.

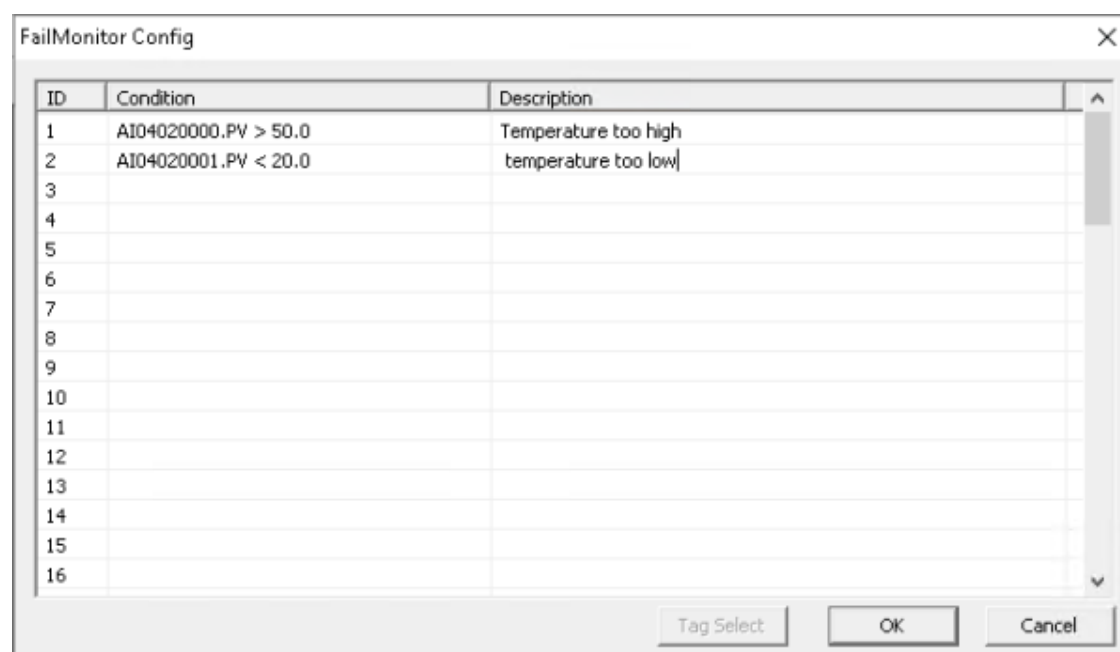
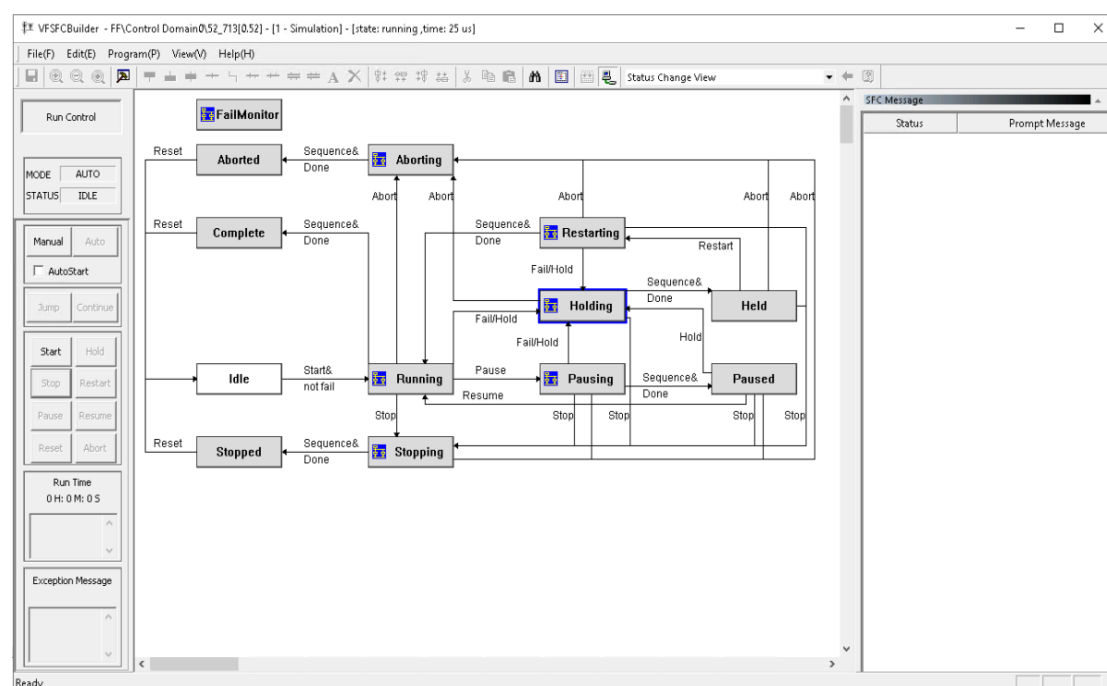


Figure 4-2 Example of “FailMonitor” configuration

During program operation, when an exception item configured in FailMonitor occurs:

- The program will enter Holding status, which means that the program is in “Holding” status in monitoring panel.
- The error code and description of exception message are displayed in the lower left corner, as shown in the figure below.





**Figure 4-3 Monitoring effect of FailMonitor (in the debugging status)**

### 4.1.3 Exception Lifted

When FailMonitor detects that the exception is lifted, you need to reset FAIL\_IDX parameters, to send RESTART command, and make SFCEX program enter Restarting status. If FAIL\_IDX parameters is not reset, SFCEX program will reenter Holding status.

You can reset FAIL\_IDX parameters by using the following methods:

- Through the rest button in SFCEX program tag panel.
- In Holding logic, after exception handling, use ST code to reset.

## 4.2 Configure Transient Program

Transients includes Aborting, Running, Pausing, Stopping, Restarting and Holding, it is used to configure the program related to field production operation.

Transient program consists of step, transition condition, branch, etc.

### 4.2.1 Add and Configure Step

This section mainly introduces how to configure start step, end step and general step.

#### Types and Constraints of Step

In SFCEX program, there are three kinds of step: start step, end step and general step.

- Start step: each programming status only allows to contain and has only one start step. In the program, start step is the first step, programmable, after start

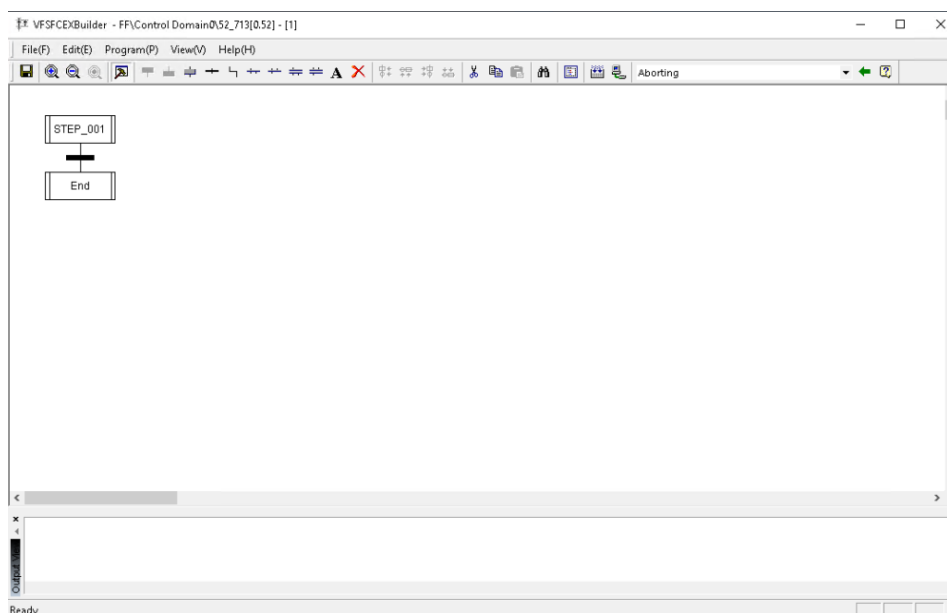
step only connection transition or one-selection branch is supported.

- End step: each programming status only allows to contain and has only one end step. In the program, end step is the last step, non-programmable, before end step only connection transition or one-selection combine is supported.
- General step: programmable, former order can be connection transition, parallel branch and one-selection combine, after order can be connection transition, one-selection branch and parallel combine.




## Configuration of Step

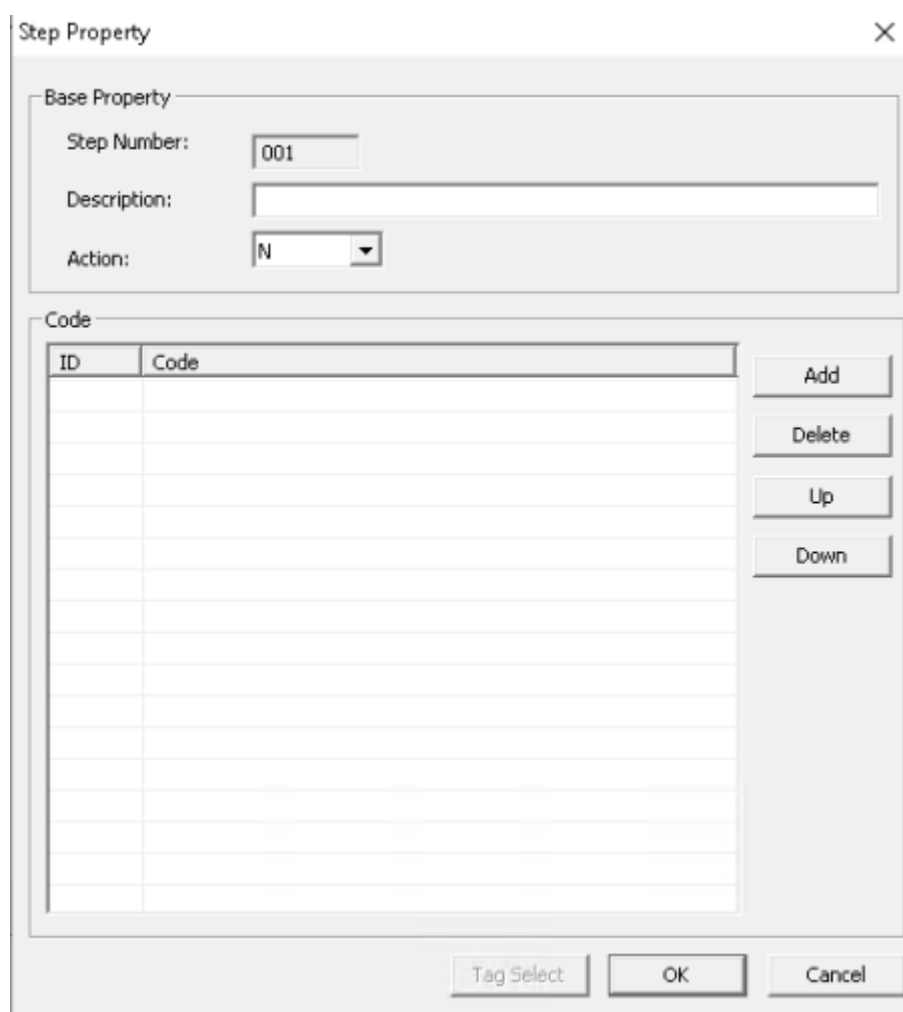
You can add and configure steps in transient program through operations as follows.

- 1) Double-click transient to enter programming interface, as shown in the figure below.



**Figure 4-4 Example of transient programming page**

- 2) Add step  
Click  button in toolbar to add start step in the program, click  button to add general step in the program, click  button to add end step in the program.
- 3) Configure step property.  
Double click the start step to pop up “Step Property” dialog box as shown in the figure below.



The image shows a 'Step Property' dialog box with a close button (X) in the top right corner. It is divided into two main sections: 'Base Property' and 'Code'.

**Base Property Section:**

- Step Number:** A text input field containing '001'.
- Description:** A large empty text input field.
- Action:** A dropdown menu currently showing 'N'.

**Code Section:**

- A table with two columns: 'ID' and 'Code'. The table is currently empty.
- Four buttons are located to the right of the table: 'Add', 'Delete', 'Up', and 'Down'.

**Bottom Section:**

- Three buttons: 'Tag Select', 'OK', and 'Cancel'.

**Figure 4-5 “Step Property” Dialog Box**

- 4) Configure properties of step according to the following table.

| Item        | Configuration Description |
|-------------|---------------------------|
| Description | Description of the step   |

| Item        | Configuration Description   |
|-------------|---|
| Instruction | <p>Used to specify the method of the step execution, you can select in drop-down list:</p> <p>N: execute repeatedly.</p> <p>P: execute only once (rise along the pulse).</p> <p>L: execute repeatedly in a limited time. After selecting L, you need to configure time limit after this configuration item.</p> <p>D: execute repeatedly after a time delay. After selecting D, you need to configure time delay after this configuration item.</p> <p>S type command/R: S type command and R command appear in pairs, S type commands include S, SL, SD and DS. After the step configured as S type commands is activated, this step will continue to be executed until a stop command from R step is received. When configuring R step, you should specify the stopped S type command step.</p> <p>If 001 step is configured as S step, 003 step is configured as R step, the execution task in 001 step will not be stopped until 003 step is executed.</p> <p>SL: hold execution in a limited time. After selecting SL, you need to configure time limit after this configuration item., the unit is second.</p> <p>DS: hold execution after activating step and extending the time limit; in the process of D (delay), if current step is no longer activated, then D action stops execution, and S action no longer execute; DS action can be stopped by R command. The time limit of delay can be configured in text box "time" after DS command, the unit is second.</p> <p>SD: hold execution after activating step and extending the time limit; in the process of D (delay), current step is no longer activated, then D action stops execution, and S action no longer execute; SD action can be stopped by R command. The time limit of delay can be configured in text box "time" after SD command, the unit is second.</p> |
| Code        | <p>Use ST function to configure the code that this step need to execute, and you can add tag of the code in it by clicking "tag selection". Notes: write operation is only supported with writable tag in "code".</p> <p>When editing code, general tag and array tag.</p> <p>General tag takes the format "tag name.fields", such as AO01020000.IN.</p> <p>Array tag applies the format "array parameter name.field name[dimension 1, dimension 2]", such as G_IMF.OUT1[2,1]. G_IMF is the common function block name, OUT1 is the array output pin of G_IMF, [2,1] is subscript of array elements.</p>  |

- 5) Click "OK" to save configuration of current step properties.

After configuring the step, select the step and select "Show Detail" in its right-click menu, configuration of the step will be displayed according to the number configured in the step, as shown in the figure below.


|          |   |                      |
|----------|---|----------------------|
| STEP_001 | N | Open valve           |
|          | 1 | A001020000.IN = 5.0; |
|          | 2 | D001020000.IN = ON;  |

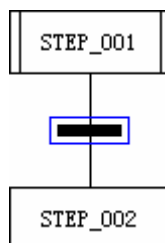
**Figure 4-6 Configuration result for step**

#### 4.2.2 Add and Configure Transition Condition

Transition condition is normally used to judge whether you can activate subsequent steps. You can and configure transition condition though operation as follows.

## Configuration Steps

- 1) Click  in toolbar to add a new transition condition in programming area.
- 2) Double-click the transition condition to be configured, as shown in the blue box in the figure below.



**Figure 4-7 Transition condition in SFCEX program**

- 3) “Transition Property” dialog box will pop up as shown in the figure below.



**Transition Property**

Base Property

Name:

Description:

☐ Jump

Condition

Gate  Gate1  Gate2  Gate3

| ID | Condition | Gate |
|----|-----------|------|
| 1  |           |      |
| 2  |           |      |
| 3  |           |      |
| 4  |           |      |
| 5  |           |      |
| 6  |           |      |
| 7  |           |      |
| 8  |           |      |

Tag Select OK Cancel

**Figure 4-8 “Transition Property” dialog box**

- 4) Configure transition properties according to the following table.

| Item        | Configuration Description  |   |
|-------------|--|---|
| Name        | Program assigns automatically, and takes Tn format, n is 1~256 ..  |   |
| Description | Used to configure the description of the transition condition, supporting 64 characters at most.   |   |
| Jump        | Check box  | Used to configure whether the transition needs to jump to the designated step. Noted: this item can only be configured when transition condition is branch structure, and transition condition is not on the far left.<br>Check, means requiring jump. Otherwise, automatically execute to next step. |
|             | Text Box   | When check box is checked, this item takes effect.<br>Used to specify the target step to jump to when this transition condition holds. After configuring this item, a jump arrow will be inserted below the transition condition, and the target step will be displayed on the right of the arrow.    |
| Condition   | Used to configure the judgment statement of transition condition, it supports two levels of logic.<br>Gate, used to specify the logic relation between gate 1, gate 2, and gate 3, logic relations including AND, OR and XOR are supported.<br>Gate 1, 2, and 3, are used to specify the direct logic relation of the condition. If gate 1 is related to three conditions: condition 1, condition 2 and condition 3, and the logic relation between these three conditions is AND, then configure "gate 1" as "AND" and configure condition 1, condition 2 and condition 3, specify the "gate" of these tree conditions is "gate 1". |   |

### Configuration Result

If the execution condition of 001 is:

(NA01020000.VALUE > 10.0 AND NA01020000.VALUE < 50.0)

AND

(ND01020000.VALUE OR ND01020001.VALUE)

AND

(ND01020002.VALUE XOR ND01020003.VALUE XOR ND01020004.VALUE)

Then you can configure transition condition T1 as shown in the figure below.

Transition Property

Base Property

Name: T1

Description: Heat up

☐ Jump STEP\_001

Condition

Gate AND

Gate1 AND

Gate2 OR

Gate3 XOR

| ID | Condition                | Gate  |
|----|--------------------------|-------|
| 1  | NA01020000. VALUE > 10.0 | Gate1 |
| 2  | NA01020000. VALUE < 50.0 | Gate1 |
| 3  | NA01020000. VALUE        | Gate2 |
| 4  | NA01020001. VALUE        | Gate2 |
| 5  | NA01020002. VALUE        | Gate3 |
| 6  | NA01020003. VALUE        | Gate3 |
| 7  | NA01020004. VALUE        | Gate3 |
| 8  |                          |       |

Tag Select OK Cancel

Figure 4-9 Example of transition property

After configuration transition, select the transition and select “Show Detail” in its right-click menu, configuration converted will be displayed according to the condition configured in transition, as shown in the figure below.

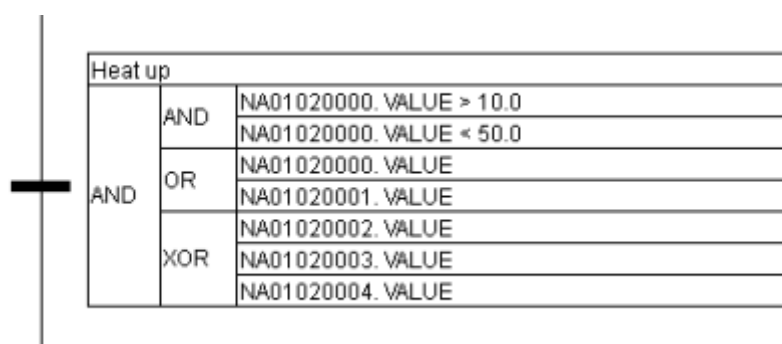


Figure 4-10 Example of details of transition

In online status, when transition is activated, the status of each condition in the transition

will be displayed, red means not valid, green means valid.

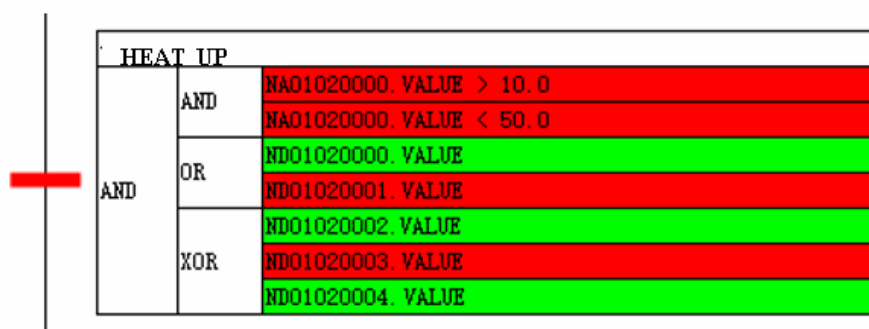


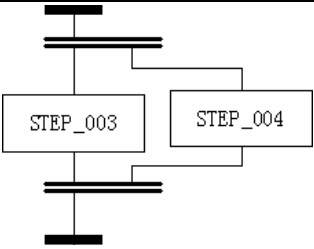


Figure 4-11 Debugging result of transition

### 4.2.3 Add and Configure Branch

Branch in SFCEX includes one-selection branch, one-selection combine, parallel branch and parallel combine. Methods of adding branches are listed in the table below.

| Operation Target      | Operation Graphics | Instruction   |
|-----------------------|--------------------|---|
| One-Selection Branch  |                    | <p>Used to realize the selection structure, when one of the multiple judgment conditions holds after the step is completed.</p> <p>Method of adding: select  in menu, and click program editing area.</p> <p>Connection limits: former order of one-selection branch can connect to start step and end step, after order can connect to transition, as shown in the figure below.</p> <p>Quantity of branch: in the default state, quantity of one-selection branch is 2, you can double-click to set quantity of branches, a maximum of 16 is supported. One-selection branch with three branches is shown in the left figure.</p>         |
| One-Selection Combine |                    | <p>One-selection combine and one-selection branch are used in conjunction to realize entering next step after condition in one-selection branch is satisfied.</p> <p>Method of adding: select  in menu, and click program editing area.</p> <p>Connection limits: former order can connect to transition; after order can connect to end step and general step.</p> <p>Quantity of branch: in the default state, quantity of one-selection branch is 2, you can double-click to set quantity of branches connected to former order, a maximum of 16 is supported. One-selection branch with three branches is shown in the left figure.</p> |




| Operation Target | Operation Graphics  | Instruction  |
|------------------|---|--|
| Parallel Branch  |  | Used to specify the multiple steps that can be executed after transition condition is satisfied.<br>Method of adding: select  in menu, and click program editing domain.<br>Connection limits: former order can connect to transition; after order can connect to general step.<br>Quantity of branch: in the default state, quantity of one selection branch is 2, you can double-click to set quantity of branches, a maximum of 16 is supported. Parallel branch with two branches is shown in the left figure.  |
| Parallel Combine |   | Parallel combine and parallel branch are used in conjunction to realize entering subsequent transition condition after multiple step execution.<br>Method of adding: select  in menu, and click program editing area.<br>Connection limits: former order of can connect to start step and end step, after order can connect to transition.<br>Quantity of branch: in the default state, quantity of one selection branch is 2, you can double-click to set quantity of branches, a maximum of 16 is supported. Parallel branch with two branches is shown in the left figure. |

You can create order step, branch and etc., in branch, the methods are similar to branch operation of basic elements.

#### 4.2.4 Connect Objects

After adding objects like step, transition, branch and etc., you need to connect related objects with sequential relationship.

- 1) Click  in toolbar.
- 2) Click the upstream object first, and click the downstream object.

After linking the objects, link line will be displayed as one of the following three forms. Direct line does not support adjustment, the lines allowed for adjustment and adjustment position of Z line and N line are shown in the figure below.

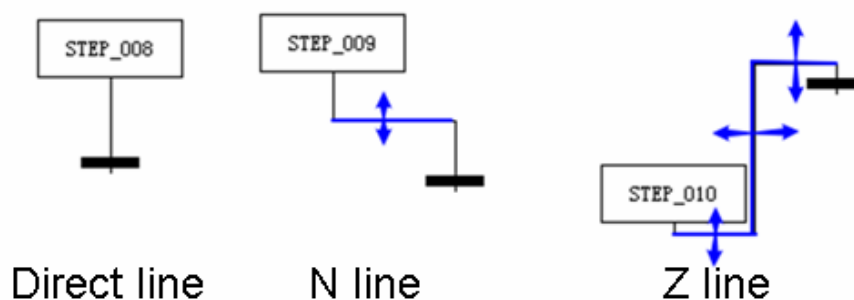







Figure 4-12 Three forms of link line

### 4.2.5 Align Graphical Objects


During programming, after adding multiple graphical objects, if you need to adjust the alignment form of graphical objects, you can realize that by following methods:

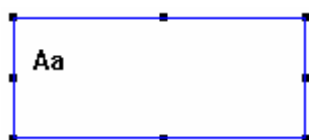
- 1) Multi-select the graphical objects to be aligned.
- 2) According to the needs, click align button  in toolbar.

| Legend  | Description  |
|---|--|
|  | Left alignment, align the selected object left along the center.   |
|  | Right alignment, align the selected object right along the center. |
|  | Up alignment, align the selected object up along the center.       |
|  | Down alignment, align the selected object down along the center.   |

### 4.2.6 Add Annotation

You can add annotation as needed in programming area.

- 1) Click  in toolbar, and click programming area, the text box of annotation as shown in the figure below will be displayed in programming area.



*Figure 4-13 Example of text box of comment*

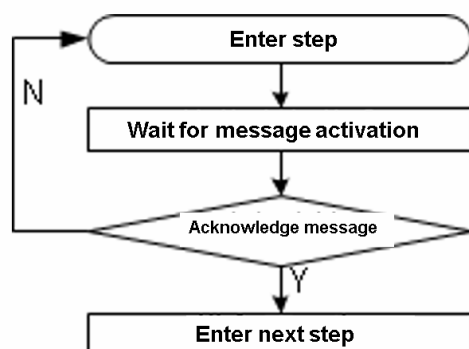
- 2) Double-click the text box and enter annotation into it.
- 3) After entering annotation, you can right-click the annotation to set font style in pop-up “Font” dialog box.

## 4.3 Configure Message

Contents in SFCEX program, including step, transition condition, etc., all support message. Message is used to specify the condition to be satisfied after step and transition condition are activated, next step can only be executed when the message is satisfied.

As shown in the figure below, the responding process of message in SFCEX program is

illustrated by example of step.

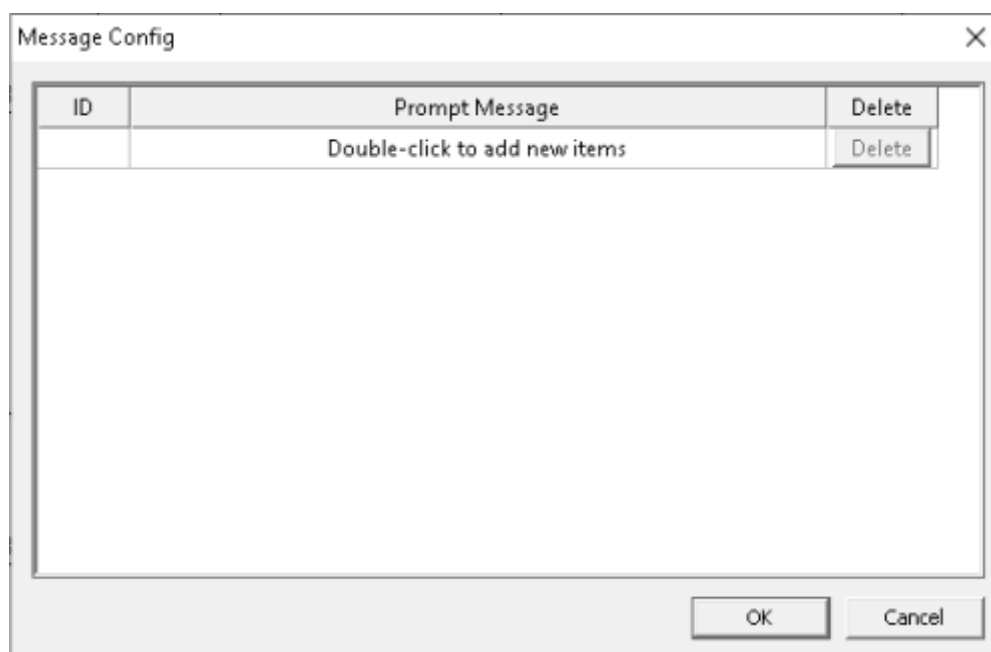


**Figure 4-14** Responding process of message in SFCEX program

### 4.3.1 Configure Message

You can specify message for SFCEX program by doing as follows:

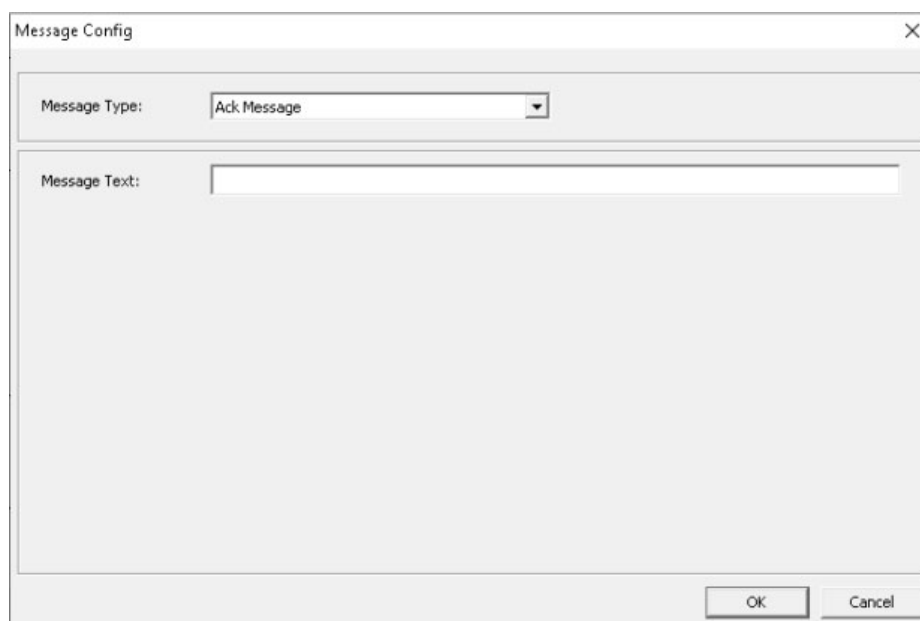
- 1) Select “Edit > Prompt Message Config” in menu to pop up “Message Config” dialog box.



**Figure 4-15** “Message Config” dialog box

When applying message, you can distinguish message by ID, for detailed application instruction please refer to “Apply Message”.

- 2) Double-click to add a new item, and a dialog box as shown in the figure below will pop up.

The image shows a 'Message Config' dialog box. At the top, there's a title bar with 'Message Config' and a close button. Below the title bar, there's a section for 'Message Type' with a dropdown menu currently showing 'Ack Message'. Underneath that is a large text area labeled 'Message Text:'. At the bottom right of the dialog, there are two buttons: 'OK' and 'Cancel'.

**Figure 4-16 Interface of message content configuration**

- 3) Select the type of message in “Message Type” drop-down list, and configure message content according to the message type.
    - Ack Message: displayed as acknowledge prompt, the message is satisfied after acknowledgement.
    - When “message type” is “Ack Message”, you need to specify the message text, as shown in the figure below.
    - Select message: displayed as selection dialog box, the message is satisfied after selection.
    - When “Message Type” is “Select Message”, you need to specify the message text and text to be displayed, as shown in the figure below.
      - Specify the content of the message to be displayed in “Message Text”.
      - You can double-click the blank line in “selection list” to configure the selection of select message. For each selection you need to specify the related parameter and reset value, after selecting the specified message, the related parameter will initialize according to the except value.
- Notes: a maximum of 8 selections can be added.

Message Config

Message Type: Select Message

Message Text:

| Display Text | Tag Parameter     | Data Type | Except Value | Delete |
|--------------|-------------------|-----------|--------------|--------|
| 90           | -click to add new |           |              | Delete |

OK Cancel

**Figure 4-17 Interface of Select Message configuration**

- Input message: displayed as configuration dialog box of the specified value, the message is satisfied after configuring values.
- When “Message Type” is “Input Message”, you need to specify the parameter related to message and the value to be input, as shown in the figure below.
  - Specify the content of the message to be displayed in “Message Text”.
  - You can click the blank line in “Input list” to add new items and configure the related parameter and its properties. Notes: a maximum of 8 input items can be added.

Message Config

Message Type: Input Message

Message Text:

| Display Text              | Tag Parameter | Data Type | Minimum | Maximum | Unit | Delete |
|---------------------------|---------------|-----------|---------|---------|------|--------|
| able-click to add new ite |               |           |         |         |      | Delete |

OK Cancel

**Figure 4-18 Interface of input message configuration**

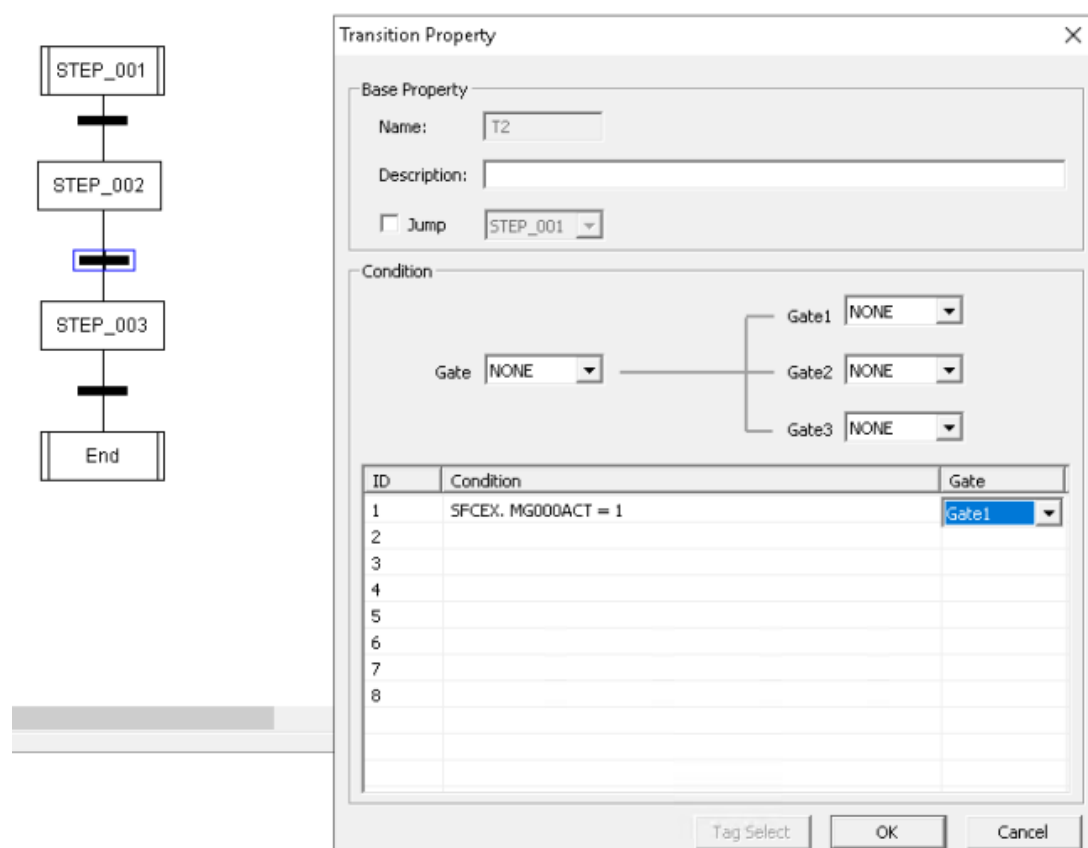
### 4.3.2 Apply Message

After configuring message, you can apply message in logic. When applying the message, the message is generally activated in the step and acknowledged in the transition, meaning using MG\*\*\*ACT in step, and using MG\*\*\*ACK in transition condition.

Application methods of the message are illustrated as followed by the example of step 001 and the transition after it

- 1) Add the activation command of the message in step 001, as shown in the figure below.

Add "MG000ACT" statements as shown in the figure below, "SFCEX" means the current SFCEX program name, "MG" means the message, "000" means message ID and "ACT" means activation message. In default state, MG000ACT=0, which means inactive. Executing "MG000AC: =1" means activating the message.

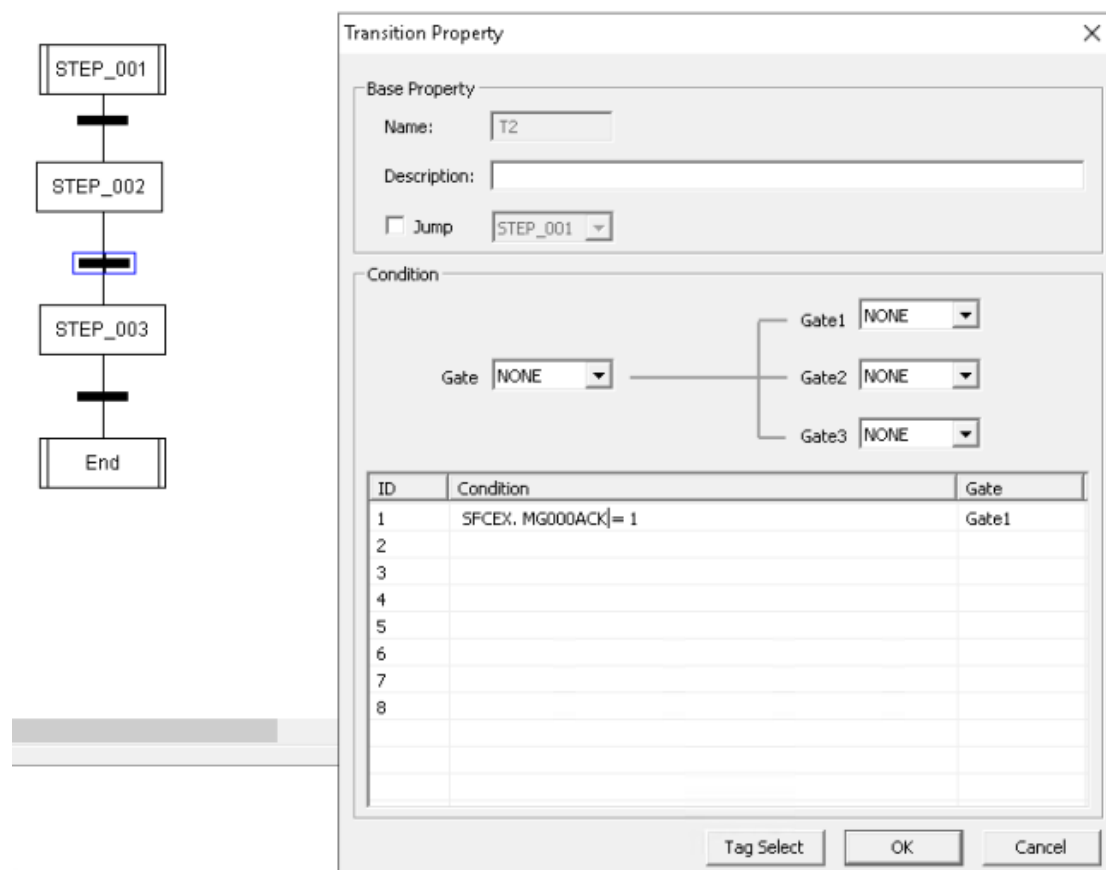


**Figure 4-19 Add message activation statements in the step**

- 2) Add the acknowledge command of the message in the subsequent transition, as shown in the figure below.

Add "MG000ACK" statements in the transition as shown in the figure below, "SFCEX" means the current SFCEX program name, "MG" means the message, "000" means message ID and "ACK" means ack message.

In the default state, MG000ACK=0 means not acknowledged, keep the previous step active and repeat the execution of the code in the step. If "MG000ACK=1", which means the message is acknowledged, end the previous step and activate the next step.



**Figure 4-20 Add message acknowledge statements in the transition**

### 3) Add message in operation instruction

After applying the message in the step and its subsequent transition, you need to add messages in operation instruction to pop up the notice of the message during monitoring. You can add the operation instruction of the message as follows.

Open VFHMICfg and select “operation group > operation instruction” to pop up the “operation instruction configuration” interface as shown in the figure below.



[illegible]

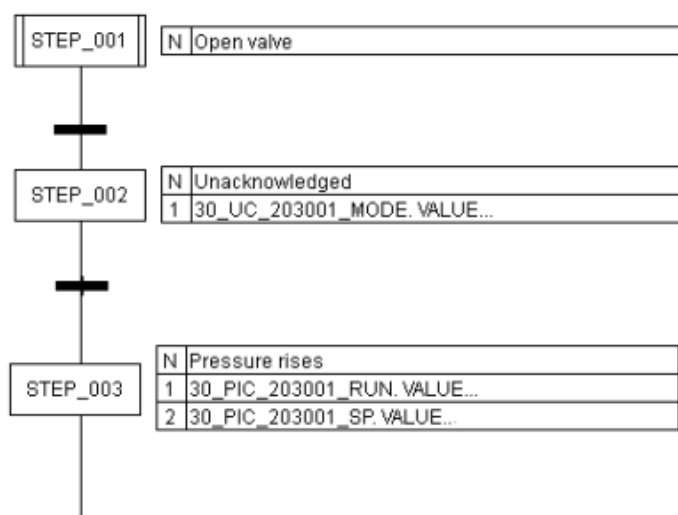
**Figure 4-21 Add SFCEX message in operation instruction**

- 4) Configure SFCEX program message in “Phase/SFCEX Message Setting” group area.  
  
Select the message needed in “Available Phase/SFCEX Message” on the right, and click “←” to add it into the left area.
- 5) Click OK to save the settings.

## 4.4 Display Detailed Content

The steps in SFCEX program include the contents of execution, the transition includes transition condition. In VFSFCEXBuilder, the detailed configuration information of displaying or hiding the steps and transition will be displayed in the program.

- When no details are displayed, right-click the blank part, and select “Show Detail” to display the details of the steps and transition, as shown in the figure below.



- When details are displayed (as shown in the figure above), right-click the blank part, and cancel checking “Show Detail” to hide the details of the steps and transition.

## Section 5 Compile and Debug

---

After editing SFCEX, you can compile and debug the program in VFSFCEXBuilder.

### 5.1 Compile

VFSFCEXBuilder supports compiling the logic in the SFCEX program, the detailed steps are as follows.

Select **Program > Compile** in menu, VFSFCEXBuilder will begin syntax check on SFCEX program, and display the result in the “Output View”:

- If the syntax is right, then prompt that the compiling passes.
- If the syntax is wrong, then prompt reasons for the errors, double-click the wrong item to go to the object of the wrong syntax and select the object



---

**Tips:**

**If the tag referenced in SFCEX user program is modified after reference, the control station configuration needs to be recompiled and downloaded.**

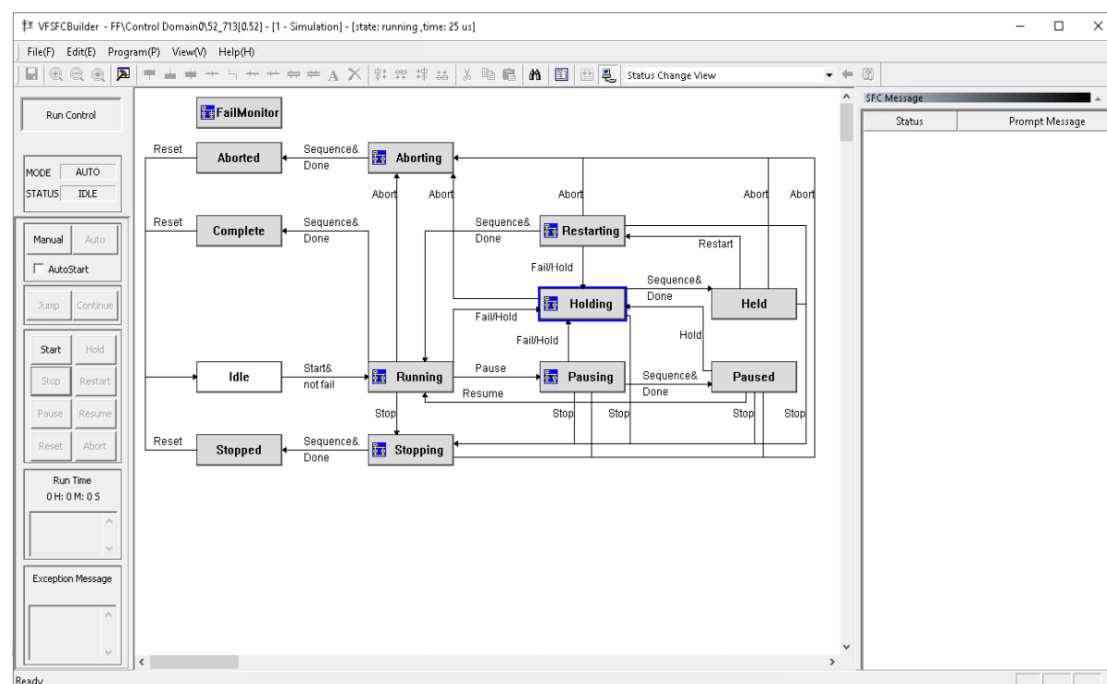
---

### 5.2 Debug

After the compiling passes and downloads, you can debug the SFCEX program online in VFSFCEXBuilder.

#### 5.2.1 Enter Debugging Interface

Select menu command “Program > Online”, the state transition figure will switch to debugging state, as shown in the figure below.



**Figure 5-1 SFCEX program in debugging status**

- 6) The SFCEX program debugging interface shown in the figure above, mainly contains these areas:

| Area Name                    | Description  |
|------------------------------|--|
| Mode/State                   | Current Mode and Status  |
| Auto/Manual Switching Region | <p>Auto: default, when transition condition is satisfied, activate automatically and execute the next step.</p> <p>Manual: after the execution of the current active step, the step will keep executing, only when it receives the command to continue and the transition condition is satisfied, the next step is activated and executed.</p> <p>Auto start: check this item, the program will automatically enter the first step and restart execution after execution in manual mode.</p>   |
| Control Command Region       | <p>SFCEX program supports controlling through these commands:</p> <p>Jump, clear the status parameter of the target step in manual mode and activate the target step.</p> <p>Continue, continue to execute the program in manual mode.</p> <p>Start, the program enters RUNNING from IDLE or STOPPED.</p> <p>Keep, the program enters Holding status from Running, Pausing, Paused and Restarting status.</p> <p>Stop, the program enters STOPPED from RUNNING</p> <p>Restart, the program enters Restarting status from Held status.</p> <p>Pause, the program enters Pausing from Running.</p> <p>Recover, the program enters Running from Paused.</p> <p>Reset, the program enters IDL from COMPLETE or STOPPED.</p> <p>Quit, the program enters Aborting from Running, Holding, Held, Stopping, Restarting, Pausing and Paused, used for quick abort of the program.</p> |
| Run Time                     | <p>Include total execution time of the program and the current active step:</p> <p>The total execution time is the total time from the time the program enters IDLE status from RUNNING status to the time it enters Complete status.</p> <p>The current active step is the tag of the current active step.</p>  |

| Area Name         | Description   |
|-------------------|---|
| Exception Message | Used to display the abnormal information monitored by FailMonitor after the FailMonitor logic is triggered. Error code corresponds to the tag of FailMonitor. As shown in the figure below, the condition of FailMonitor whose ID is 2 is triggered, after the corresponding conditions are cleared, the exception message will disappear too.  |
| Step              | Green means that the step is activated, the execution time will be displayed on “Run” on the right.<br>Grey means completion of execution<br>White means that the execution is not activated  |
| Transition        | When flashing red, the transition is not satisfied. When flashing green, the transition is satisfied, but the break points are triggered.<br>In the mode where the transition displays details, the red condition background color means condition not satisfied; the green background color means condition satisfied. Double-click to open the transition, and you can view the satisfaction status of each transition condition, the red condition background color means not satisfied; the green background color means satisfied. |

In the debugging interface of SFCEX program, double-click “Step” or “Transition” to pop up its detailed configuration interface, which includes the real-time value of the tags, etc.

When the tag referenced is in the form of array, the “view” button will be displayed in the real-time value of the tag, click this button to pop up the real-time value of the array type tag, as shown in the figure below.

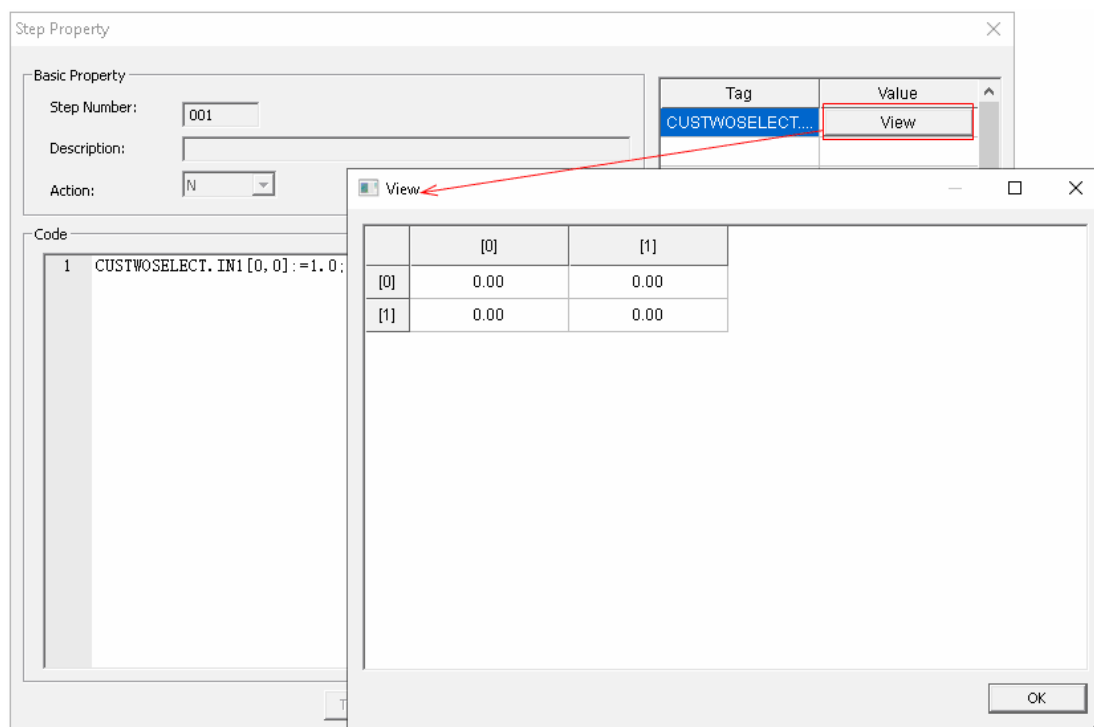


Figure 5-2 Example of the real-time value of the array type tag

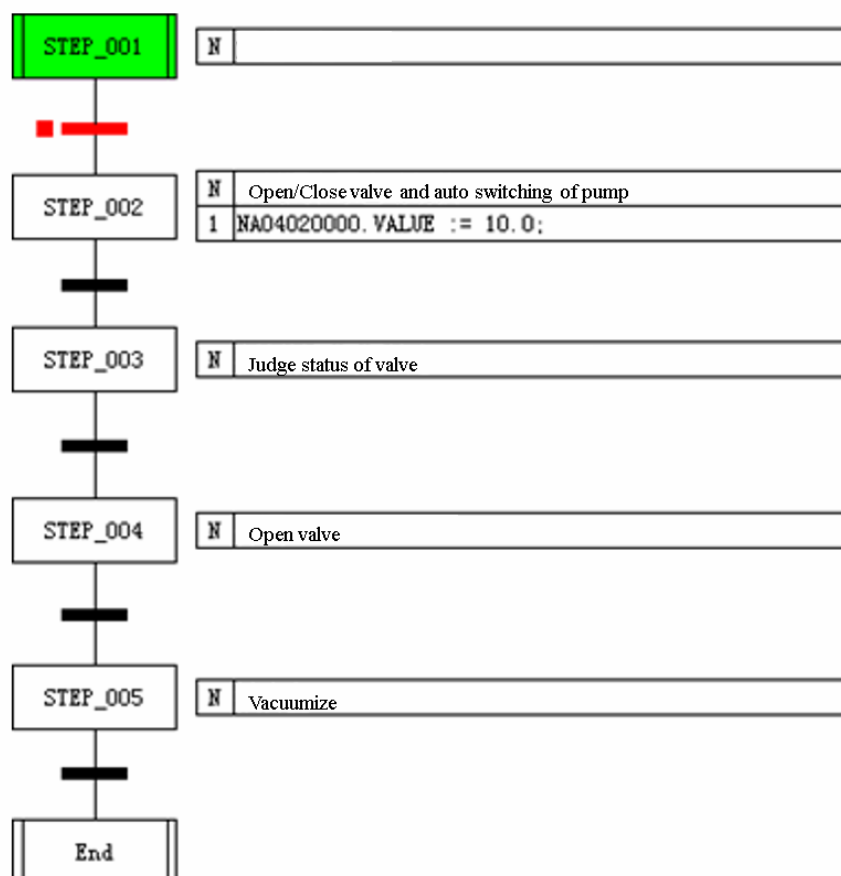
### 5.2.2 Debug Break Point

In the debugging process, you can insert break points in the transition step to know more

clearly the result of every step. After adding break points, program execution will pause at the break points.

You can add break points in the program through operations as follows.

- 1) Click the transition step and select “Add break points” command in the right-click menu, to add break points in the current transition step, as the transition step between “001” and “002” shown in the figure below.



**Figure 5-3 Example of break point debugging**

In the break point debugging interface as shown in the figure above, different status of break point has different meaning:

- Flashing red means the transition condition not satisfied.
- Flashing green means that the transition condition is satisfied, the break points have been inserted in the transition condition, and the break points are triggered.


To delete a break point, click the break point and select “delete break point” command in the right-click menu. After deleting the break points, the status prompt of the break points will disappear, and the break points will no longer be triggered.

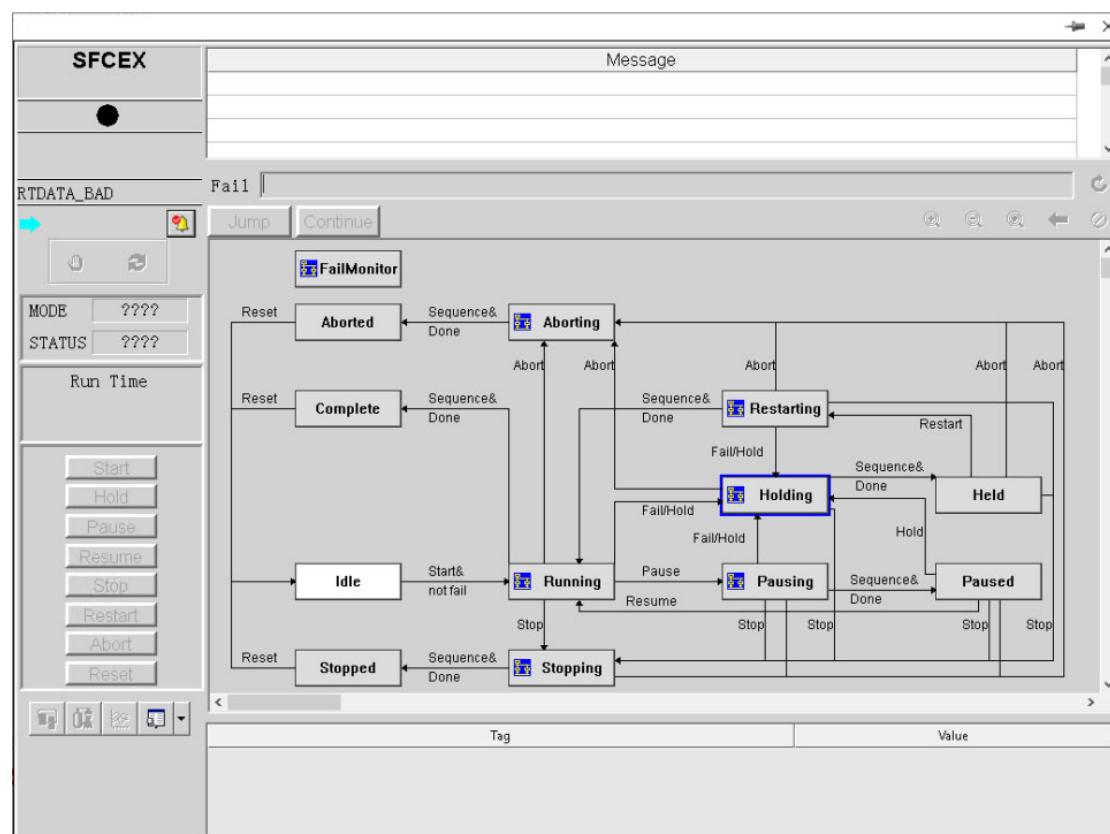
## Section 6 Monitor SFC EX Program

Create SFCEX program, and after the configuration of tag name and logic of SFCEX program, you can monitor SFCEX program through the tag panel.

## 6.1 Monitor Tag Panel of SFC EX

You can monitor SFCEX program through operations as follows:

- 1) Open High-performanceHMI Real-time monitoring software and log in.
- 2) Click  in the upper right corner of toolbar in monitoring header, and enter the tag name of SFCEX in the pop-up tag searching toolbar, to pop up the program panel of SFCEX as shown in the figure below.



**Figure 6-1 Program panel of SFCEX under supervision**

The operations of SFCEX during monitoring and debugging are almost the same:

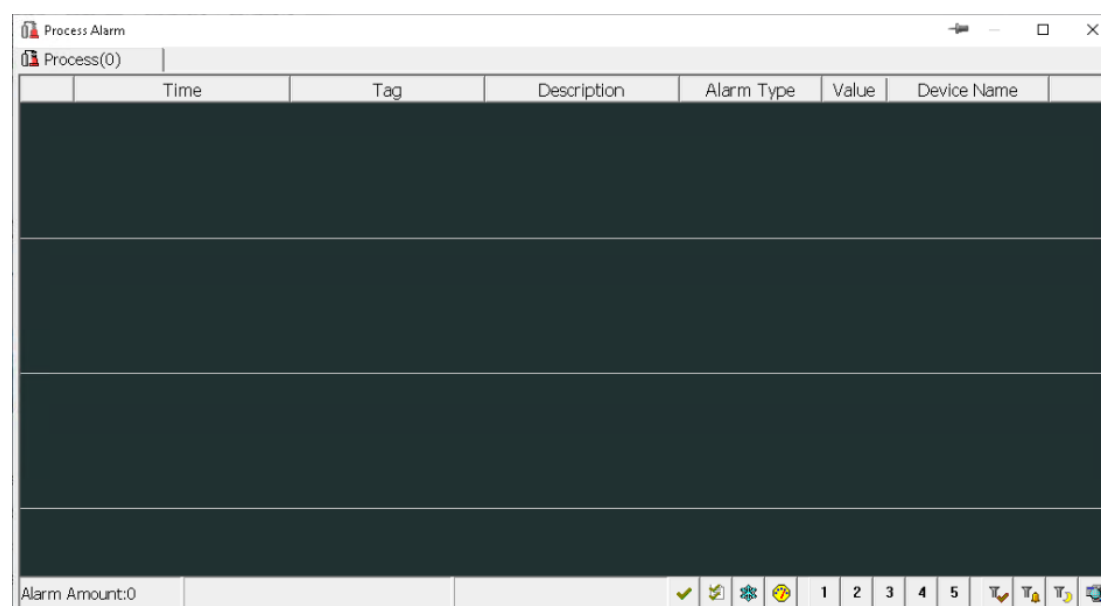
- The program status, operating time and control command of SFC EX are displayed in the left area.
- The execution status of SFC EX is displayed in the right area

Select the step in the program, the configuration of the step and the real-time value of the tag related to the step will be displayed below.

Select the transition in the program, the configuration of the transition and the real-time value of the tag related to the transition will be displayed below.

## 6.2 Monitor Fail Alarm

When SFCEX program enter FailMonitor status and the “Fail” alarm of SFCEX is enabled, the alarm information of SFCEX program tag will be displayed during the process alarm of the real-time monitoring. As shown in the figure below, system detects that the FailMonitor alarm condition 2 of “SFCEX” program is satisfied.




**Figure 6-2 Example of Fail alarm in SFCEX**

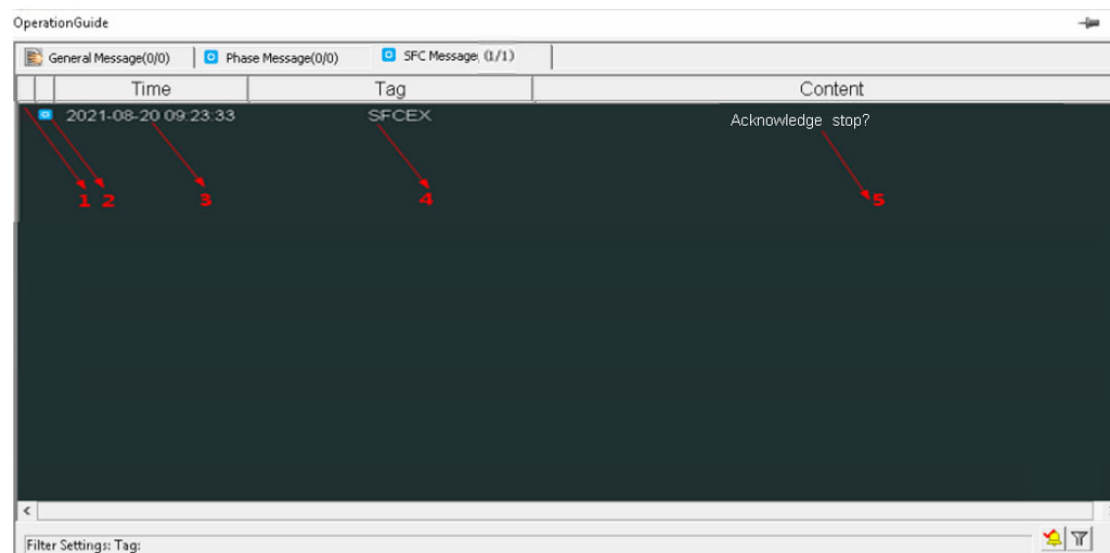
Select Fail alarm of SFCEX program tags in the alarm list to go to the program panel of SFCEX program.

## 6.3 Monitor SFCEX Message

After configuring the operation guide of SFCEX message, you can monitor SFCEX message during the operation through the following methods.

- 1) After logging in High-performanceHMI Real-time monitoring software, operation instruction icon  on the right means message prompt.
- 2) Click the operation instruction icon to pop up the “operation guide” dialog box as shown in the figure below.







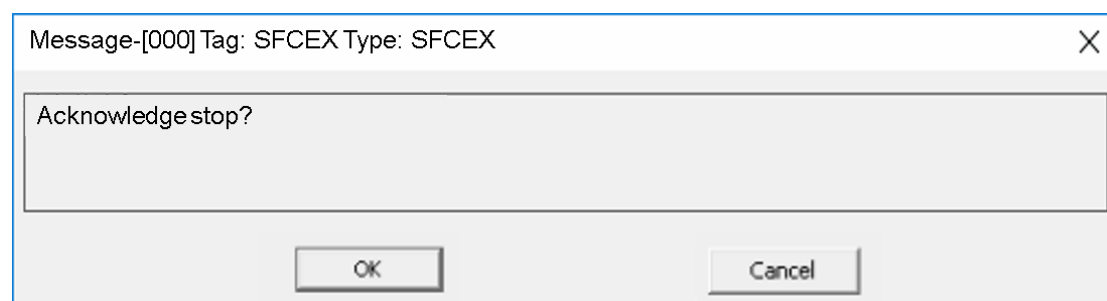
**Figure 6-3 SFCEX message in operation guide**

### 6.3.1 Illustration of Message

The information list shown in the Figure 6-3 includes these information:

| Column | Meaning   |
|--------|---|
| 1      | Used to show whether the message is acknowledged, if acknowledged then display  , if not acknowledged then the column is blank.                                    |
| 2      | Used to show whether the message is essential message. If display  , then the message is essential message. If no sign, then the message is not essential message. |
| 3      | Used to show the generating time of the message.  |
| 4      | Used to show the SFCEX program tag name related to the message.   |
| 5      | Used to show the content of the message   |

In the SFCEX message list shown in Figure 6-3, you can double-click the SFCEX message to display the details of the message, as shown in the figure below.



**Figure 6-4 Example of message**

The following is an example of ack message to show the meaning of the title bar of the

details dialog box:

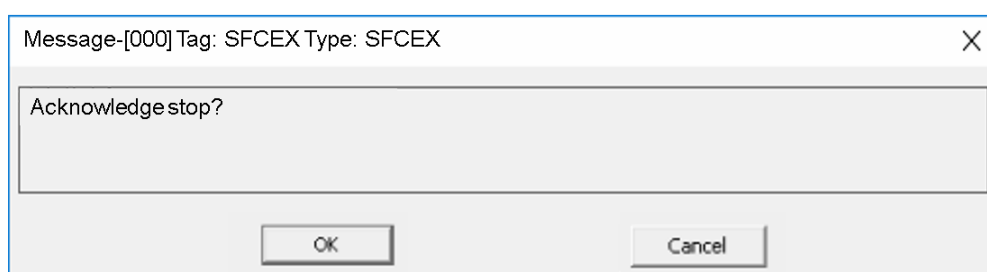
- Message number, such as [000] as shown in the figure above.
- The SFCEX program tag related to the message, such as “SFCEX” as shown in the figure above.
- Message type, such as “SFCEX” as shown in the figure above.

### 6.3.2 Operation Instruction of Message

After adding SFCEX message in the operation guide

Different operations can be executed in monitoring according to message prompt:

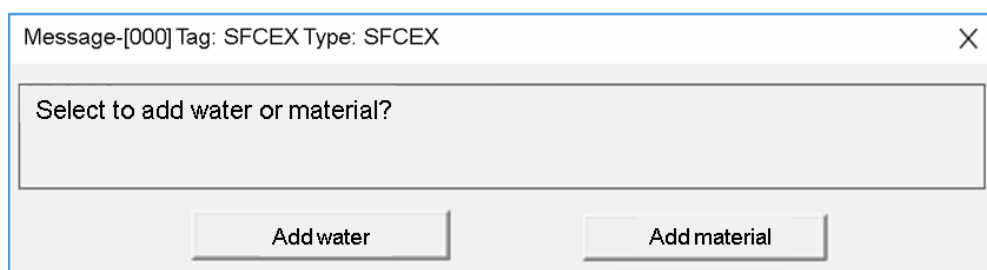
- Ack message, as shown in the figure below.



A screenshot of a dialog box titled "Message-[000] Tag: SFCEX Type: SFCEX". The main text area contains the prompt "Acknowledge stop?". At the bottom, there are two buttons: "OK" and "Cancel".

As shown in the figure above, you need to click the “OK” button to enter the next active step. If “Cancel” is selected, the execution will stop.

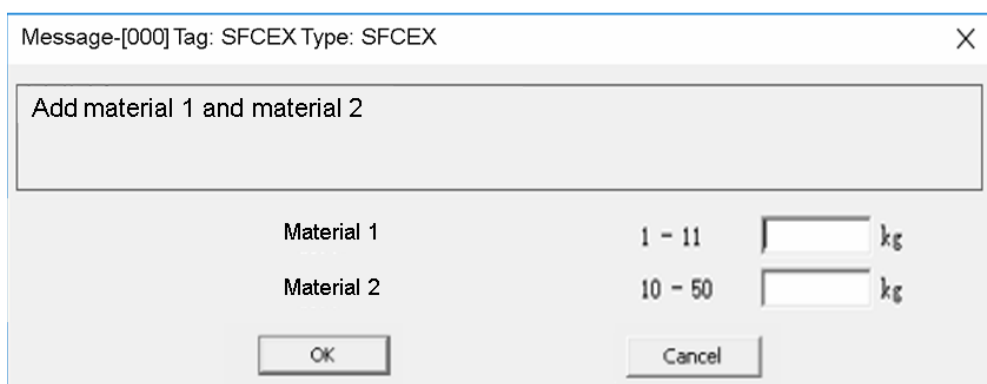
- Select message requires subsequent operations.



A screenshot of a dialog box titled "Message-[000] Tag: SFCEX Type: SFCEX". The main text area contains the prompt "Select to add water or material?". At the bottom, there are two buttons: "Add water" and "Add material".

As shown in the figure above, you need to select the action of the next step, such as “Add water” or “Add material”, to enter the next active step

- Input message, you need to input the production-related parameters



A screenshot of a dialog box titled "Message-[000] Tag: SFCEX Type: SFCEX". The main text area contains the prompt "Add material 1 and material 2". Below the text area, there are two rows of input fields. The first row is labeled "Material 1" and contains the text "1 - 11" followed by a text box and the unit "kg". The second row is labeled "Material 2" and contains the text "10 - 50" followed by a text box and the unit "kg". At the bottom, there are two buttons: "OK" and "Cancel".

As shown in the figure above, you need to input the production-related weight of the next step to execute the production of the next step.

Notes: when the input value message includes more than one value to be input, you need to enter all the values to enter the next step, otherwise the message will not be acknowledged.

## Section 7 Description of SFCEX Tag Parameter

SFCEX tag can be applied in the user program and HMI Config, and the parameters of SFCEX tag that can be referenced are listed in the table below.

| Name     | Data Type | Initial value | Description  |
|----------|-----------|---------------|--|
| CMD      | USINT     | 0             | Command:<br>0=CMD_NONE; 1=CMD_START; 2=CMD_STOP;<br>3=CMD_PAUSE; 4=CMD_ABORT; 5=CMD_RESUME;<br>6=CMD_RESET; 7=CMD_JUMP; 8=CMD_RESTART;<br>10=CMD_HOLD; 11=CMD_NEXTSTEP.  |
| STATUS   | USINT     | 0             | Current Status:<br>0=Spare, 1=run, 2=suspending, 3=recovering, 4=stopping,<br>5=stopped, 6=quit, 7=jump, 8=finished, 9=quitting, 10=stopped,<br>11=holding, 12=held, 13=restarting.  |
| COLDST   | BOOL      | OFF           | Cold start mode: ON=cold start enabled; OFF= cold start disabled.  |
| SWOOS    | BOOL      | OFF           | Function prohibition: ON=prohibited  |
| SWAM     | BOOL      | ON            | Manual/auto switch: OFF=manual; ON=auto.   |
| AU_START | BOOL      | OFF           | Auto start: On=auto restart enabled; OFF=auto restart disabled.  |
| NEXTSTP  | USINT     | 0             | Next Step  |
| JUMP_TO  | USINT     | 0             | Switching object   |
| RUBn     | BOOL      | OFF           | RUNNING switches the break point status of n, n is the transition number, it is dependent on the controller type and the current status.<br>The range of RUBn is: On=valid break point; OFF=invalid break point.   |
| RUETn    | BOOL      | OFF           | RUNNING switches the final judgment result of n, n is the transition number, it is dependent on the controller type and the current status. The range of RUETn is : On=satisfied; OFF=not satisfied.   |
| RUTn     | USINT     | 0             | RUNNING switches the final judgment result of the subexpression of n, n is the transition number, it is depended on the controller type and the current status. RUTn is used by bit from low to high, each bit shows the status of single expression, 1 means satisfied, 0 means not satisfied.  |
| RUSnS    | USINT     | 0             | Status of the RUNNING step n, n is the step number, it is depended on the controller type and the current status. RUSnS displays the status by bit and by decimal:<br>Bit 0, 0=inactivated; 1=activated.<br>Bit 1, 0=unexecuted; 1=executed.<br>Bit 2, 0=unfinished; 1=finished.<br>If this value is 6 (110 in binary), then the status of the step is finished, executed and inactivated. |
| RUAnS    | USINT     | 0             | Instruction status of the RUNNING step n, n is the step number, it is depended the controller type and the current instruction status.<br>RUAnS=0, it means the step is not executed.<br>RUAnS=1, it means the step is being executed.<br>RUAnS=2, it means the step finishes execution.   |
| RUSnT    | UDINT     | 0             | Activating time of RUNNING step n, unit is millisecond. n is the step number, it is depended on the controller type and the current status.  |

| Name     | Data Type | Initial value | Description  |
|----------|-----------|---------------|--|
| RUAnT    | UDINT     | 0             | Execution time of RUNNING step n, unit is millisecond. n is the step number, it is depended on the controller type and the current status.   |
| TIMERSn  | UDINT     | 0             | Timer n count (unit: second), n is 1~4.  |
| TIMERMn  | REAL      | 0             | Timer n counts (unit: second), n is 1~4.   |
| TMRPTNn  | USINT     | 0             | Timer n running mode (0=normal, 1=count down), n is 1~4.   |
| CMDTn    | USINT     | 0             | Timer n operation, n is 1~4. The range of CMDTn is:<br>0=reset; 1=start; 2=pause; 3=synchronization, the data of minute timer is synchronized to second timer;<br>4=synchronization, the date of second timer is synchronized to minute timer.   |
| TOT_TIME | UDINT     | 0             | Total time(unit: second)   |
| MGnACT   | USINT     | 0             | The activation status of message n, n is the message number, the range is 000~063.<br>MGnACT=0 means the message is not activated.<br>MGnACT=1 means the message is activated.   |
| MGnACK   | USINT     | 0             | The acknowledge status of message n, n is the message number, the range is 000~063.<br>MGnACK=0 means the message is not acknowledged.<br>MGnACK=1 means the message is acknowledged.  |
| FAILENA  | BOOL      | ON            | Fail alarm enable switch:<br>ON means enabled, SFCEX will generate Fail alarm after entering FailMonitor.<br>OFF means disabled, SFCEX will not generate Fail alarm after entering FailMonitor.  |
| AUTO_RS  | BOOL      | ON            | Auto restarting mode:<br>ON means enabled, SFCEX automatically enters Restarting status after entering Held status.<br>OFF means disabled, SFCEX holds after entering Held status, wait for Restart command to enter Restarting status.  |
| RS_STEP  | INT       | 0             | Auto activation step, the range is 1~256.<br>When function block enters Running status from Restarting status, start from the step specified by RS_STEP.<br>Assign: assign RS_STEP in the execution codes in Holding or Restarting<br>Reset: After recovering to Running status, start execution from RS_STEP, RS_STEP=0 after execution. If RS_STEP does not exist or cannot be activated, the program enters Holding and makes an alarm. |
| MGACT1   | UDINT     | 0             | The activation status of (0~31) message  |
| MGACT2   | UDINT     | 0             | The activation status of (32~63) message   |
| MGACK1   | UDINT     | 0             | The acknowledge status of (0~31) message   |
| MGACK2   | UDINT     | 0             | The acknowledge status of (32~63) message  |

## Section 8 Appendix Description of SFCEX Language

### 8.1 ST Function Key Words Supported

The ST function key words supported by SFCEX in High-performanceHMI system software are listed in the Table 8-1

**Table 8-1 ST function key word**

| Key word                   | Function  |
|----------------------------|---|
| WHILE; DO; END_WHILE       | Constitute WHILE statement  |
| FOR; TO; BY; DO; END_FOR   | Constitute FOR statement  |
| IF; THEN; ELSEIF; END_IF   | Constitute IF statement   |
| ELSE                       | Constitute IF Statement or CASE statement                         |
| REPEAT; UNTILL; END_REPEAT | Constitute REPEAT statement                                       |
| CASE; OF; END_CASE         | Constitute CASE statement   |
| EXIT                       | Exit statement  |
| AND; NOT; MOD; OR; XOR     | Operator  |
| ON; OFF; TRUE; FALSE       | BOOL type value   |
| RETURN                     | Key words of ST language, normally not used in user function bock |

### 8.2 ST Function Operator Supported

The ST function operators and the priority supported by SFCEX in High-performanceHMI system software are listed in the table below.

**Table 8-1 ST operators supported by SFCEX**

| Serial Number | Operator name                     | Operator | Priority |
|---------------|-----------------------------------|----------|----------|
| 1             | Parenthesis operator              | ()       | 1        |
| 2             | Function reference operator       | SIN()    | 2        |
| 3             | Logic negation operator           | NOT, ~   | 3        |
| 4             | Multiply operator                 | *        | 5        |
| 5             | Mod operator                      | MOD      | 5        |
| 6             | Divide operator                   | /        | 5        |
| 7             | Plus operator                     | +        | 6        |
| 8             | Minus operator                    | -        | 6        |
| 9             | Greater than operator             | >        | 7        |
| 10            | Greater than or equal to operator | >=       | 7        |
| 11            | Less than operator                | <        | 7        |

| Serial Number | Operator name                  | Operator | Priority |
|---------------|--------------------------------|----------|----------|
| 12            | Less than or equal to operator | <=       | 7        |
| 13            | Unequal to operator            | <>       | 8        |
| 14            | equal to operator              | =        | 8        |
| 15            | XOR                            | XOR      | 9        |
| 16            | AND                            | AND      | 10       |
| 17            | OR                             | OR,      | 11       |

### 8.3 Timer

SFCEX software supports reading and writing second timer TIMERS, hundred millisecond timer TIMERMS and minute timer TIMERM.

**Table 8-2 Type of timer and related information**

| Type                      | Mark    | Total | Data Type | Range   |
|---------------------------|---------|-------|-----------|---------|
| Second timer              | TIMERS  | 512   | UINT      | 0~65535 |
| Hundred millisecond timer | TIMERMS | 128   | UINT      | 0~65535 |
| Minute timer              | TIMERM  | 128   | UINT      | 0~65535 |

### 8.4 Common Array

SFCEX supports three types of common array, each type of array can contain up to 32 rows\*128columns numbers.

**Table 8-3 Description of common array supported by High-performanceHMI**

| Mark           | Array type                            | Element type |
|----------------|---------------------------------------|--------------|
| ARRREAL[X,Y]   | 32 rows*128 columns (X=0~31, Y=0~127) | REAL         |
| ARRBOOL[X,Y]   | 32 rows*128 columns (X=0~31, Y=0~127) | BOOL         |
| ARRUINT[X,Y]   | 32 rows*128 columns (X=0~31, Y=0~127) | UINT         |
| ARRBOOLEX[X,Y] | 32 rows*32 columns (X=0~31, Y=0~31)   | BOOL         |
| ARRREALEX[X,Y] | 32 rows*32 columns (X=0~31, Y=0~31)   | REAL         |

## Section 9 Revision

---

*Table 9-1 Revision history*

| Document Version | Applicable Version                       | Remarks              |
|------------------|--|----------------------|
| V1.0 (20230301)  | OMC High-performanceHMI<br>V4.70.00.00   | First release        |
| V1.1 (20230830)  | OMC High-performanceHMI<br>V5.10.00.00-M | Updated screenshots. |