

**OMC System Software**

**High-performanceHMI**






**SFC Program**

**User Manual**

**IM41S09-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>Section 1 Introduction.....</b>	<b>1</b>
1.1 Working Process .....	1
1.1.1 Supported Status.....	1
1.1.2 Status-Switching Rules.....	2
1.2 Program Composition.....	2
1.2.1 Basic Elements.....	2
1.2.2 Program Structures .....	4
1.2.3 Program Running Rules .....	5
<b>Section 2 Process Description .....</b>	<b>7</b>
<b>Section 3 Create and Configure SFC .....</b>	<b>8</b>
3.1 Configure SFC Program .....	8
3.2 Configure the Tag Property of SFC.....	9
3.3 Configure the Logic of SFC .....	11
3.3.1 Add and Configure Sequential Step.....	11
3.4 Add and Configure Transition Conditions .....	13
3.5 Add and Configure Branches.....	16
<b>Section 4 Compile and Debug .....</b>	<b>18</b>
4.1 Compile .....	18
4.2 Debug.....	18
4.2.1 Debugging.....	18
4.2.2 Breakpoint Debugging.....	22
<b>Section 5 Monitor SFC Program.....</b>	<b>24</b>
5.1 Configure Monitoring Authority .....	24
5.2 Start Monitoring .....	24
<b>Section 6 Parameter Description of SFC Program Tags .....</b>	<b>26</b>
<b>Section 7 Appendix—SFC Language.....</b>	<b>28</b>
7.1 Supported ST Function Keywords .....	28
7.2 Supported ST Function Operator.....	28
7.3 Timer .....	29
7.4 Global Array.....	29
<b>Section 8 Revision.....</b>	<b>30</b>

## Table of Contents

---

# SFC Program User Manual

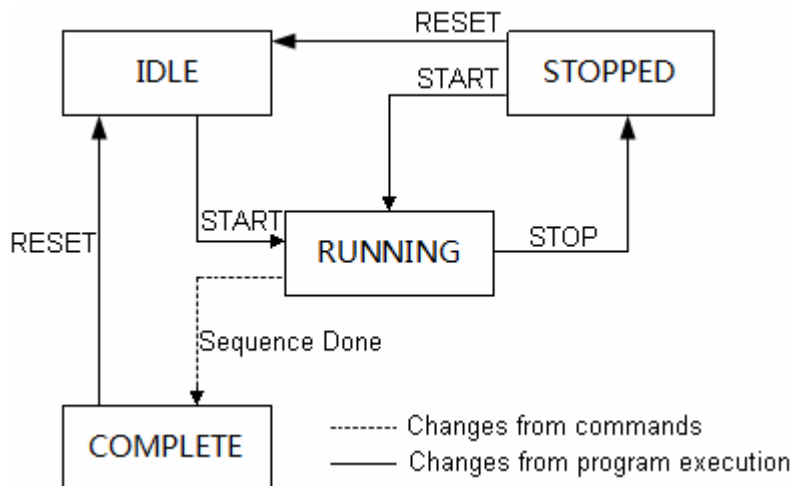
## Section 1 Introduction

SFC (Sequence Flow Chart) is a type of module operation that controls the sequence of time events, such as start, operation, stop, and so on.

SFC consists of steps and transition conditions. The step consists of a series of actions. When the transition condition is satisfied, the order is from the current step to the next step. For each SFC scan, the system determines the activated step and transition conditions. When the transition condition is satisfied, the step before the transition condition is no longer activated, and the step after the transition condition becomes activated.

### 1.1 Working Process

All status of the SFC program and commands related to status switching are illustrated in the status transition diagram shown below.



**Figure 1-1 SFC Status Transition Diagram**

#### 1.1.1 Supported Status

SFC program contains two kinds of status, namely:

- The RUNNING status is transient and supports custom programming.

- The IDLE, COMPLETE, and STOPPED are final status and do not support programming.

### 1.1.2 Status-Switching Rules

The status of the program will automatically change state in the order of IDLE > RUNNING > COMPLETE under normal conditions. During the monitoring or debugging process, the program status can be manually switched via the command button on the SFC program panel. The status switching is controlled by the START, STOP and RESET commands.

The switching rules for program status are listed in the following table.

**Table 1-1 SFC Program Status Switching Rules**

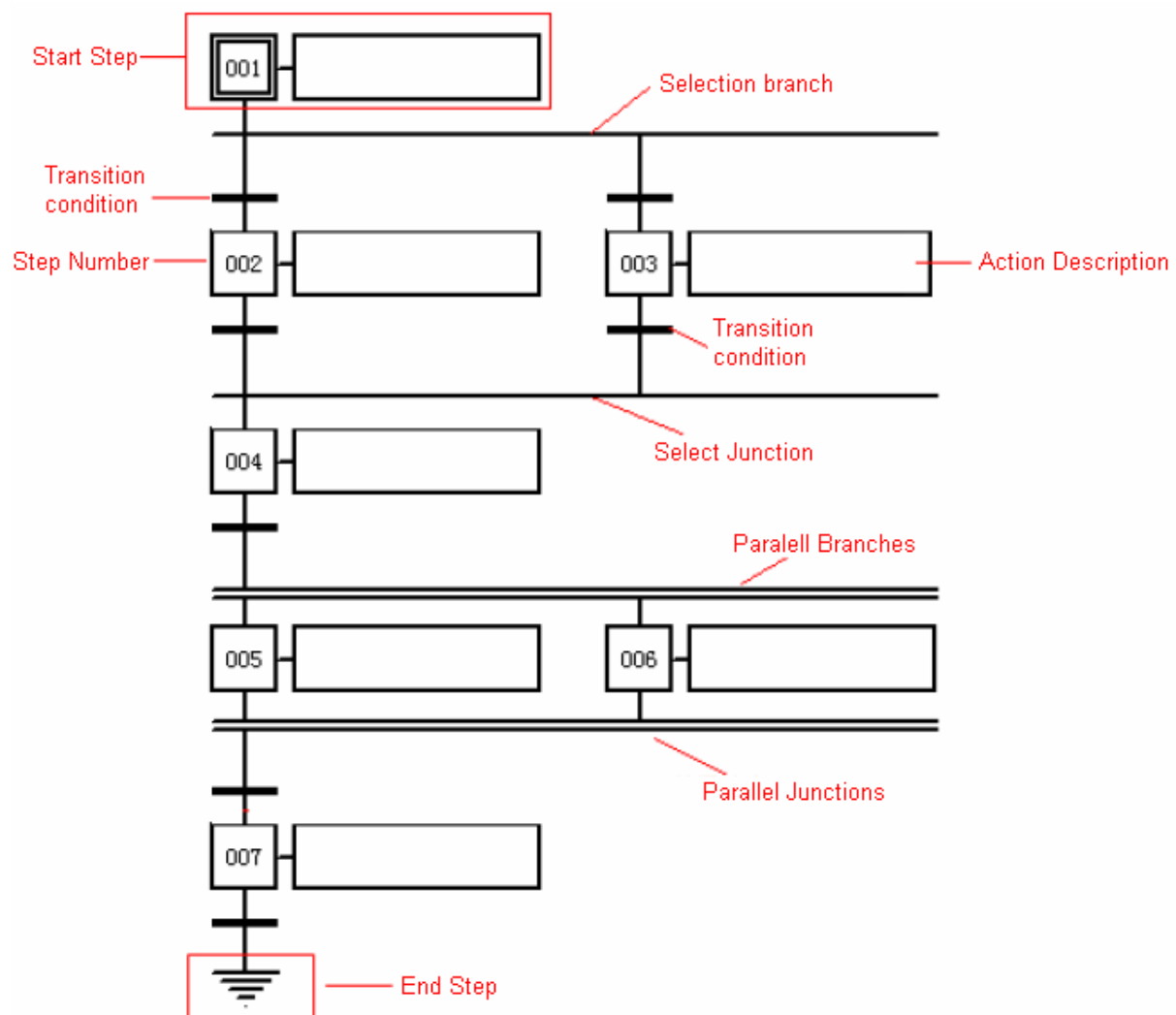
Current Status	Command	Estimated Status	Remark
IDLE	START	RUNNING	IDLE is the initial status of the program. After it enters IDLE status, data of all programs will be initialized.
RUNNING	-	COMPLETE	Execution in order
	STOP	STOPPED	After it receives STOP command in RUNNING status, the program will switch to the STOPPED status.
STOPPED	START	RUNNING	After it receives START command in the STOPPED status, program will continue to execute the activated step in RUNNING status.
	RESET	IDLE	After it receives RESET command in the STOPPED status, program will restore to the initial status that is IDLE status.
COMPLETE	RESET	IDLE	After it receives RESET command in the COMPLETE status, program will restore to the initial status that is IDLE status.

## 1.2 Program Composition

The SFC program in the High-performanceHMI system software supports sequence diagrams of multiple structures, and the sequence diagrams of multiple structures are composed of steps, branches, transitions, and etc.

### 1.2.1 Basic Elements

The basic elements of the SFC program include start step, step number, selection branch, transition condition, action description, selecting branch-selecting junction, parallel branches-parallel junctions, and end step, as shown in the following figure. SFC programs include the following elements, but not every element is required.



**Figure 1-2 SFC Basic Elements**

During the execution of the typical SFC program shown in the above figure, the following rules are followed:

- The program starts from step  
Step supports three states, namely initialization state (not activated, not executed), activated state and complete state.  
The initial state means that the step is not activated and has not been executed, the active state means that the instruction or code is being executed, and the completion state means that it has been executed and is no longer executed.
- Transition condition judgment after step

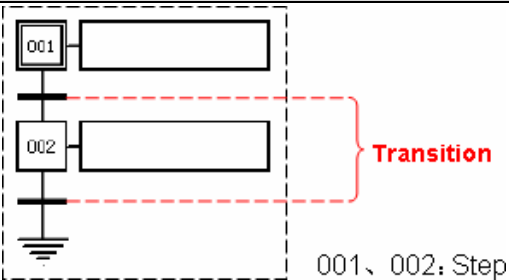
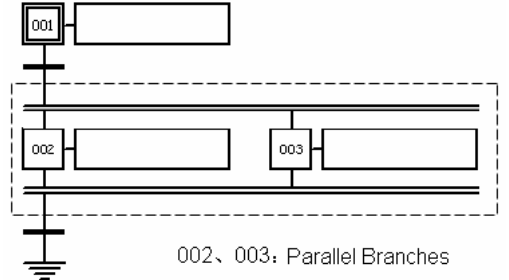
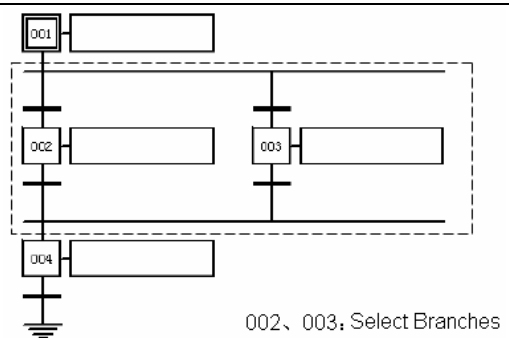
The judgment starts when the previous step of the transition condition is activated. When the entire transition condition passes, the establishment status of each expression in the transition condition is recorded, and the update of the satisfaction status of the transition condition is stopped, that is, the satisfaction condition of the transition condition is a historical data.

- Step jump

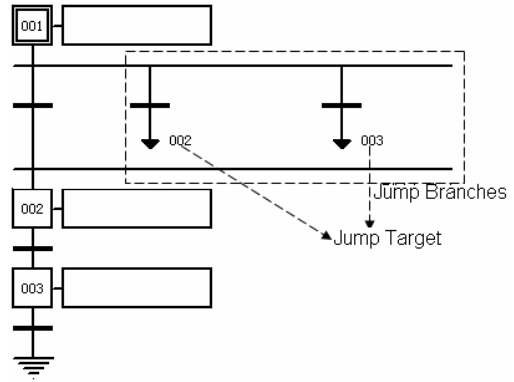
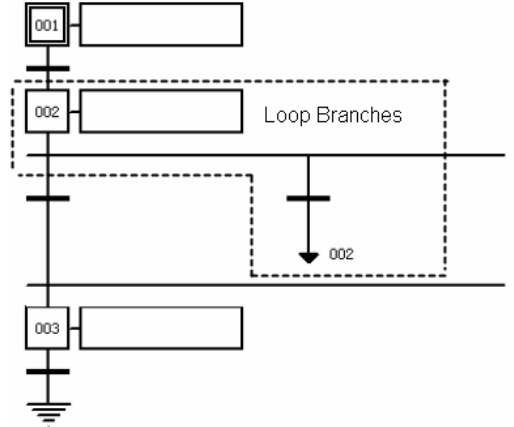
During program execution, you can jump through the steps on the panel or according to the jumps configured in the transition conditions. The jumped step should be the parallel branch of the currently activated step.

### 1.2.2 Program Structures

Program structures include sequential structure, parallel branch, selection branch, jump branch and loop branch as shown in the table below.

Structure	Structure Name
 <p>001、002: Step</p>	<p>The sequential structure consists of steps and transitions. Step is used to perform production-related operations. Transition is used to configure the transition for entering the next step.</p>
 <p>002、003: Parallel Branches</p>	<p>Parallel branches can operate simultaneously in the same cycle, and their execution is independent of each other. The transition condition under the parallel branch can operate only if all the steps at the end of the parallel branches are activated and their transition conditions are all satisfied.</p>
 <p>002、003: Select Branches</p>	<p>Selection branches are usually added after the step. The execution order of the selected branches is starting from the branch where the condition is satisfied first and going into next in turn. If multiple transition conditions are met at the same time, the leftmost transition in the SFC diagram is activated, as 002 shown in the figure on the left side.</p>



Structure	Structure Name
	<p>A jump branch consists of an initial transition condition and an arrow symbol specifying a jump target. If the transition condition of the jump is satisfied, the jump will be performed. If there are multiple jumps at the jump source location (002 and 003), the jump where the transition condition is firstly satisfied operates firstly. If multiple transition conditions are satisfied at the same time, the leftmost transition condition is activated. In the jump structure, the jump only operates at the end of the selection branch and cannot jump into the parallel branches.</p>
	<p>Loop branch is a special condition of the jump branches and its jump objective is the step prior to it. As shown in the figure on the left, the jump objective in the loop branches is the step “002”, which is the previous step of the jump branch.</p>

### 1.2.3 Program Running Rules

- SFC program supports two modes, automatic and manual modes.

In automatic mode, the step starts immediately after activation.

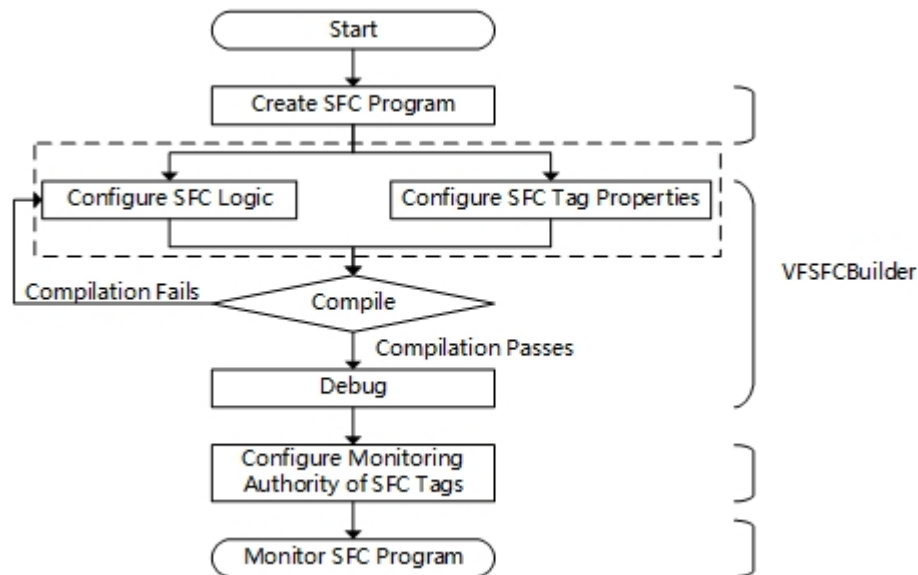
In manual mode, the step starts after it is activated and waits for “Continue” command for next transition’s result. After the step execution is completed, if the next transition condition of the step is satisfied, the transitioned step will be activated and wait for the “Continue” command. If the next transition condition of the step is not satisfied, continue to keep the current step active and wait for the transition condition to be satisfied.

- When the transition condition prior to the branch is not satisfied, the currently activated step will be executed in each scan cycle until the transition condition is satisfied. Then, the currently activated step will fail, and the next step is activated. If you select “Execute once” (set in step property settings), it will be scanned during the first cycle during which the custom function block is called, and will not be scanned any more.
- In the selection branch, the branch that first satisfies the transition condition operates first, the other branches will not operate. When there are more than two selection branches satisfying the condition at the same time, the leftmost branch operates first.

- In the parallel branch, when the transition condition prior to the branch is satisfied, all branches under the parallel branch will start running at the same time. When all the branches finish running and the transition condition is met, the parallel branches end operation.
- When the transition condition prior to the jump is satisfied, the program logic jumps directly to the target step and continues operating from that step. When the transition condition is satisfied, the program goes to the next step. If it is not satisfied, each control period needs to scan this transition.

## Section 2 Process Description

The creation, configuration and monitoring of the SFC program in High-performanceHMI system software is shown in the following figure.



**Figure 2-1 SFC Program Configuration Flowchart**

## Section 3 Create and Configure SFC

---

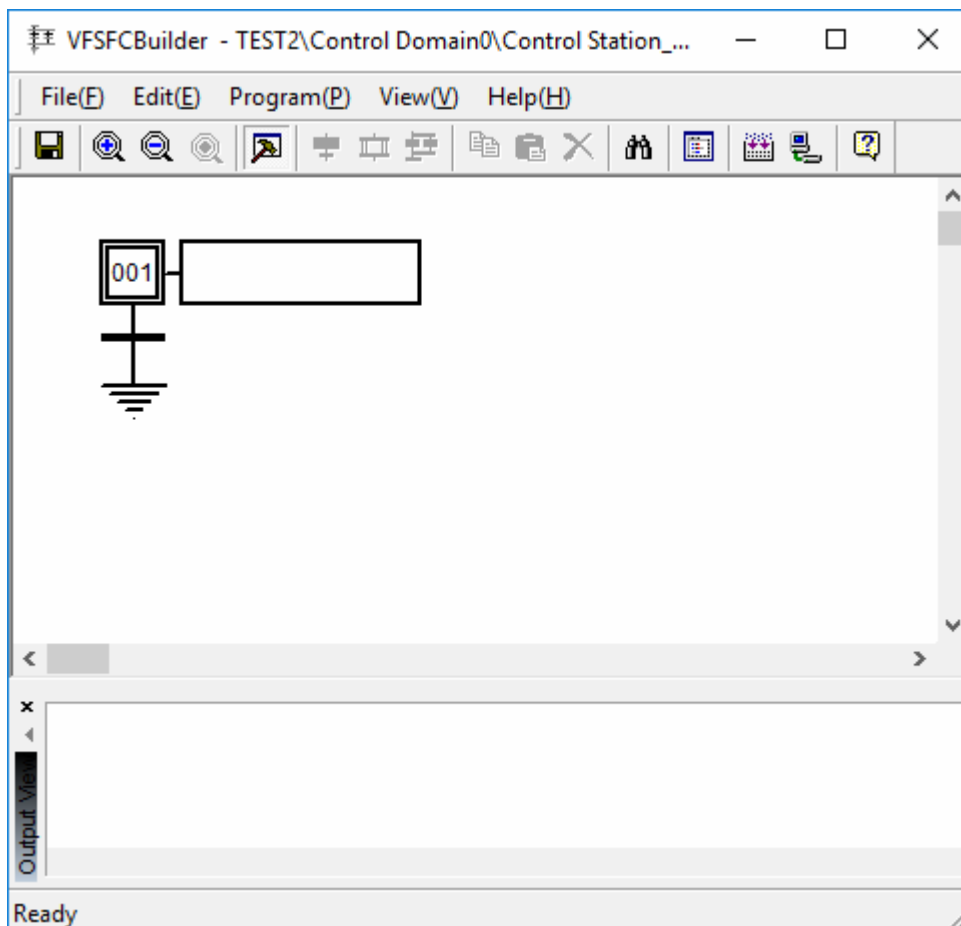
When the controller type of the control station in the High-performanceHMI project is FCU712-S, or FCU713-S, the SFC can be created.

### 3.1 Configure SFC Program

Before starting SFC programming, create a SFC program at first. The specific steps are as follows.

1. Open the VFExplorer.
2. Select “Control Station > Custom Programs” in the project navigation tree and select “New” from its right-click menu.
3. In the pop-up dialog box, select “SFC” in the “Type” drop-down list and configure “Name” and “Description” for the SFC program.
  - The program name supports 64 characters and cannot contain the characters “\”\”\V,;:<>|?\*%\r\n”.
  - The program description supports 128 characters.
  - The cycle supports fast cycle, 1 time, 2 times, 5 times, 10 times cycle multiples, the default is 1 time.
  - Start phase 0~9, default 0.

After creating the SFC program, double-click the SFC program to enter the VFSFCBuilder program configuration interface shown in the figure below. In this interface, you can operate the tag property configuration, logic configuration, compile, debugging of the SFC program.



**Figure 3-1 Initial Interface of VFSFCBuilder Software**

## 3.2 Configure the Tag Property of SFC

In the High-performanceHMI system software, SFC program is monitored by SFC program tag, so you should specify the tag name for each SFC program.

1. Configure the tag name of the SFC program.  
Choose "Edit > Properties" in the menu. The "SFC Tag" dialog box is displayed as shown in the following figure.

**Figure 3-2 Configuration Interface of SFC Tags**

2. Configure SFC program property as per the table below.

Configuration Item	Function Description
Tag Name	Used to specify the tag name of the current program. In real-time monitoring, the SFC program can be controlled through the tag monitoring panel of the SFC program. It supports up to 24 characters and can only be composed of numbers, letters, "-", "_", and cannot be duplicated with the name of the function block in the station.
Tag Description	It is used to configure the description information of the current program and supports up to 64 characters.
Tag Group	The tag group used to specify the group of the program tags. Please ensure that the monitoring user has the corresponding monitoring authority for it.
Tag Level	The tag levels are divided from 0 to 9 (0 is the lowest, 9 is the highest). Tag levels are selected from the drop-down menu. Tag level is used to set whether or not it needs to double check before values of tag panel are changed.
Decimal digitals	Decimal digital range is from 0 to 7. It is selected from the drop-down menu.
Auto/Manual Status	It is used to specify the initial manual automatic state of the program. Select "Auto", the program will start to run automatically after the program starts. Select "Manual". After the program starts, it needs to be triggered manually to start running.

Configuration Item	Function Description
Auto Start Status	It is used to specify whether the program will automatically restart after running to the COMPLETE state. It is only valid in the manual mode. Select "Open", which means that the program will automatically restart after running to the COMPLETE state. Select "Close", which means that the controller will automatically enter RUNNING status to restart after the program runs to the COMPLETE.
Cold Start Status	It is used to specify whether the program continues to run after a cold start of the controller. Select "Open", the program will enter IDLE status after the controller is cold started. Select "Close", the controller will keep the current running status after the cold start before the program.

3. Click "OK" to complete the creation and configuration of SFC program.

## 3.3 Configure the Logic of SFC

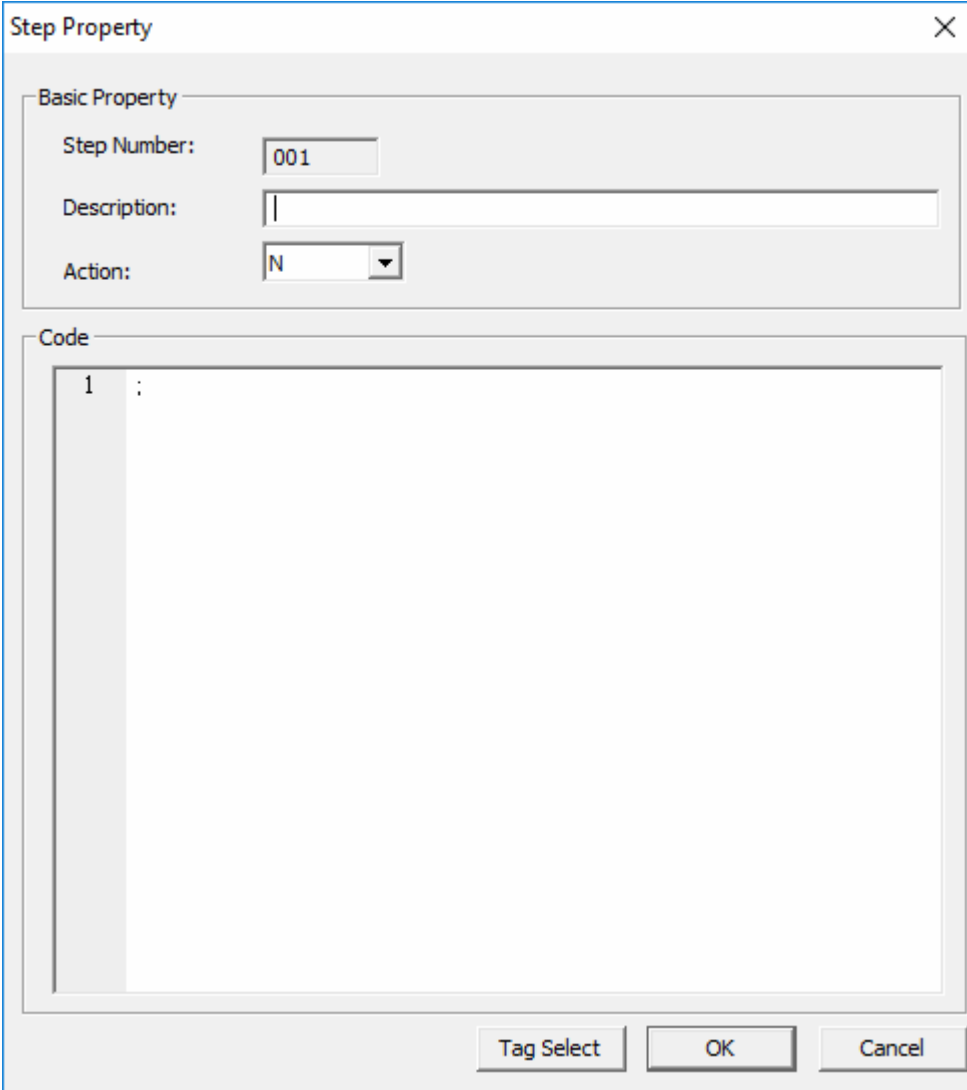
### 3.3.1 Add and Configure Sequential Step

The steps in the SFC program can be configured by the following operations.

1. Enter the SFC program configuration interface shown in Figure 3-1.

In the initial status, SFC program includes one step001. If new steps need to be added, please firstly select one step and then select "Edit > Add sequential step". The newly added step will display after the selected step.

2. Double click the sequential step you need to configure and the "Step property" dialog box will pop up as shown 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: 'Basic Property' and 'Code'. The 'Basic Property' section contains three fields: 'Step Number' with a text box containing '001', 'Description' with an empty text box, and 'Action' with a dropdown menu showing 'N'. The 'Code' section contains a large text area with a vertical line on the left and a single line of code '1 ;'. At the bottom of the dialog box are three buttons: 'Tag Select', 'OK', and 'Cancel'.

**Figure 3-3 “Step Property” Dialog Box**

3. Configure the property values for the step according to the table below.

Configuration Items	Description
Describe	The description for the step



Configuration Items	Description
Command	<p>To appoint how the step is performed, selecting from the drop-down list:</p> <ul style="list-style-type: none"> <li>● N: Execution repetitively.</li> <li>● P: Execute only once (rising edge of pulse).</li> <li>● L: Repeated execution within a limited time. After L is selected, you need to configure a limited time next to the configuration item.</li> <li>● D: Repeat after a delay. After selecting D, you need to configure the delay time next to this configuration item.</li> <li>● S class command/R: S class command occurs with R command in pairs. S class command includes S, SL, SD and DS. After the step configured with S class command is activated, this step action will continue to execute until it receives the command that the signal R step sent. When configuring R step, you need to specify the step with S class command.</li> </ul> <p>For example, if 001 step is set to S step, 003 to R step, then only after 003 step is executed, can the action of 001 step stop.</p> <ul style="list-style-type: none"> <li>● SL: Keep executing within a limited time. After you select SL, you need to configure a limited time next to the configuration item. The unit of the limited time is second.</li> </ul> <p>DS: The step holds running after it is activated and experiences the limited time delay; During the D (delay) process, if the current step is no longer activated, the D action will stop performing and no longer perform S action. DS action can be stopped by R command. The limited time of delay is set in "time" text box next to DS command and the unit is second.</p> <p>SD: The step holds running after it is activated and experiences the limited time delay; During the D (delay) process, even if the current step is no longer activated, the D action will still perform and hold S action; SD action can be stopped by R command. The limited time of delay is set in "time" text box next to SD command and the unit is second.</p>
Code	<p>Use the ST function to configure the action you need to perform in this step. Click "Tag Select" to add the tags of which codes need to be executed. It is noted that "Code" only supports writing operation on the writable parameters. When editing codes, only normal tags and array tags are supported.</p> <ul style="list-style-type: none"> <li>● Normal tags adopt "tag name, segment" such as AO11020000.IN.</li> <li>● Array tags adopt "array parameter name. segment name[dimension1. dimension2]", such as G_IMF.OUT1[2,1]. G_IMF is a global function block name, OUT1 is the output pin of G_IMF array, [2,1] is the array's subscript of OUT1.</li> </ul>

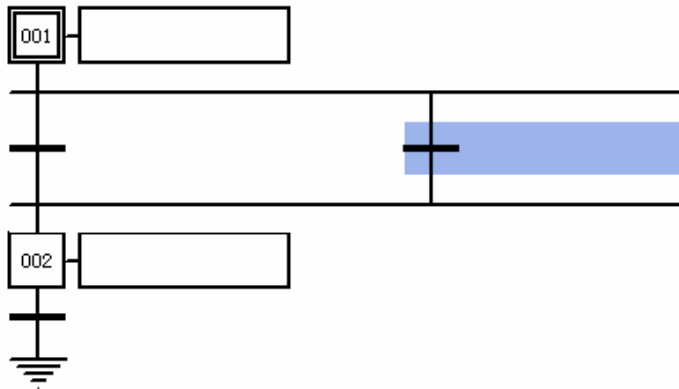
4. Click **OK** to save the property configuration of the current step.

### 3.4 Add and Configure Transition Conditions

The transition condition is generally used to determine if a subsequent step can be activated. You can add and configure transition conditions by the follows.

#### Configuration Step

1. On the SFC program configuration interface, double-click the transition condition you want to configure, as shown in the blue area of the figure below.



**Figure 3-4 Transition Conditions in SFC Program**

2. The “Transition Properties” dialog box shown in the figure below pops up.

The image shows a "Transition Property" dialog box. It has a title bar with a close button (X). The dialog is divided into two main sections: "Basic Property" and "Condition".

**Basic Property:**

- Name:** T1
- Description:** (empty text field)
- Jump:** ☐ 1

**Condition:**

- A list box on the left contains the number "1".
- A large empty text area for entering the condition logic.

**Buttons:** Tag Select, OK, Cancel

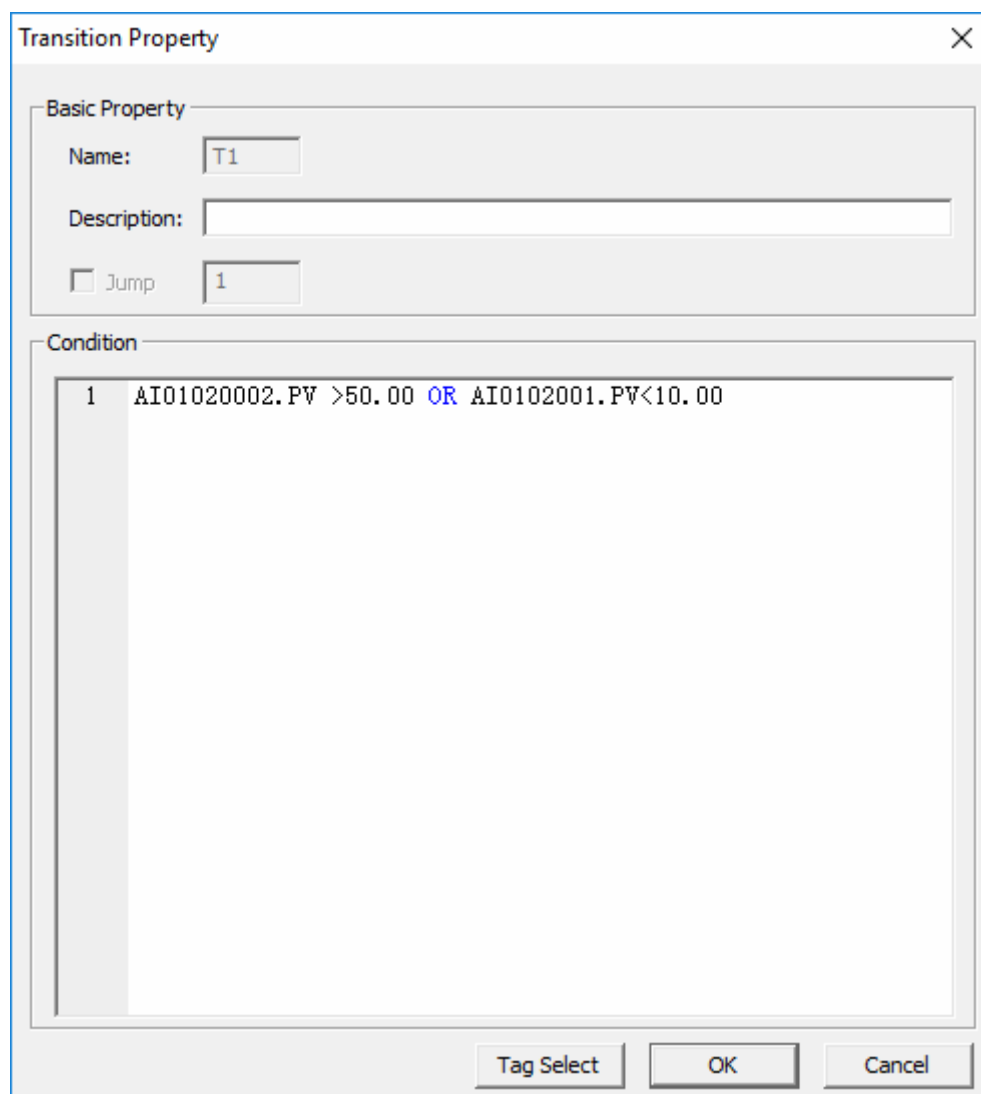
**Figure 3-5 “Transition Properties” Dialog Box**

3. Configure the “Transition properties” according to the table below.

Configuration Items	Description	
Name	Name will be automatically designated in the format of Tn, and n represents from 1 to 128.	
Description	It is used to configure the description of the transition condition and supports up to 64 characters.	
Jump	Check box	It is used to configure whether the transition condition needs to jump to the particular step. It is noted that only when the transition condition is branch structure and the transition condition is not at the leftist side, the jump can be configured. If it is checked, it indicates that you need to jump. Otherwise, it will automatically proceed to the next step.
	Text box	When the check box is checked, the configuration item takes effect. It is used to specify the target step to jump to when the transition condition is met. When this item is configured, a jump arrow will be inserted below the transition condition, and the target step of the jump is displayed on the right side of the arrow.
Transition	It is a statement used to configure the transition condition. The judgment statement should be written in ST language.	

### Configuration Effects

As shown in the figure below, when the transition condition “AI01020002.PV >50.00 OR AI01020001.PV <10.00” is established, it will jump to step 001 to start execution.



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

**Basic Property:**

- Name:** A text box containing 'T1'.
- Description:** An empty text box.
- Jump:** A checkbox labeled 'Jump' is unchecked, followed by a text box containing the number '1'.

**Condition:**

A list box containing a single item with index '1' and the condition text: `AI01020002.PV >50.00 OR AI0102001.PV<10.00`. The word 'OR' is highlighted in blue.

At the bottom of the dialog are three buttons: 'Tag Select', 'OK', and 'Cancel'.

**Figure 3-6 Transition Property Configuration Instance**

### 3.5 Add and Configure Branches

Branches in an SFC program include selection branches, parallel branches, jump branches, and loop branches. The methods for adding various branches are listed in the table below.

Operation Target	Operation Instruction
------------------	-----------------------

Operation Target	Operation Instruction
Add Parallel Branch	<p>After selecting the transition condition, you can add parallel branches by selecting “Edit &gt; Add Branch”</p>
Add Selection Branch	<p>After selecting the step, select “Edit &gt; Add Branch” to add selection branches.</p>
Extend Branch	<p>After selecting the branch start position or the parallel branch start position, select “Edit &gt; Expand Branch” to extend branches. The maximum branches are 16.</p>
Jump Branch	

You can also create sequential steps, branches, and other structures in the branch. The method is similar to that of the branch operation of basic elements. Therefore, it is not described here.

## Section 4 Compile and Debug

---

After editing the SFC program, you can compile and debug the program in VFSFCBuilder.

### 4.1 Compile

VFSFCBuilder supports the compilation of logic in SFC. The specific steps are as follows.

Select “Program > Compile” in the menu bar, VFSFCBuilder starts the syntax check of the SFC program, and displays the check result in the “output window”:

- If the syntax is correct, the compilation will be passed.
- If the syntax is wrong, the cause of the error will be prompted. When you double-click the error item, you can jump to the target object of the error syntax and select the object.



---

**Tips:**

If the SFC program referenced a tag, and the tag's name was modified after being referenced, the control station's configuration should be recompiled and downloaded.

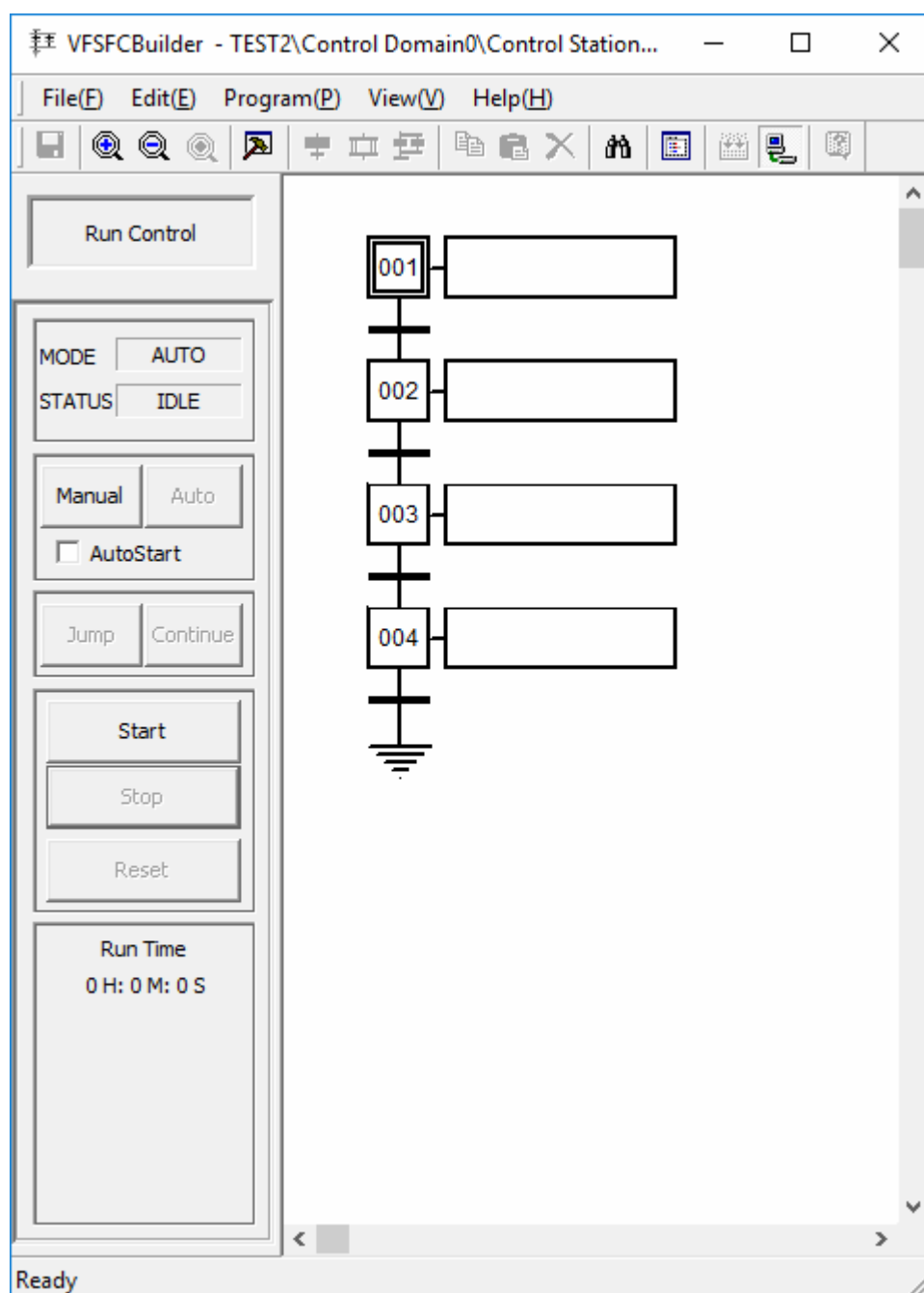
---

### 4.2 Debug

After the SFC program is compiled and downloaded, it can be debugged online in VFSFCBuilder.

#### 4.2.1 Debugging

Select **Program > Online** and the status transition diagram will switch to the debugging status as shown in the figure below.



**Figure 4-1 SFC Program in the Debugging Status**

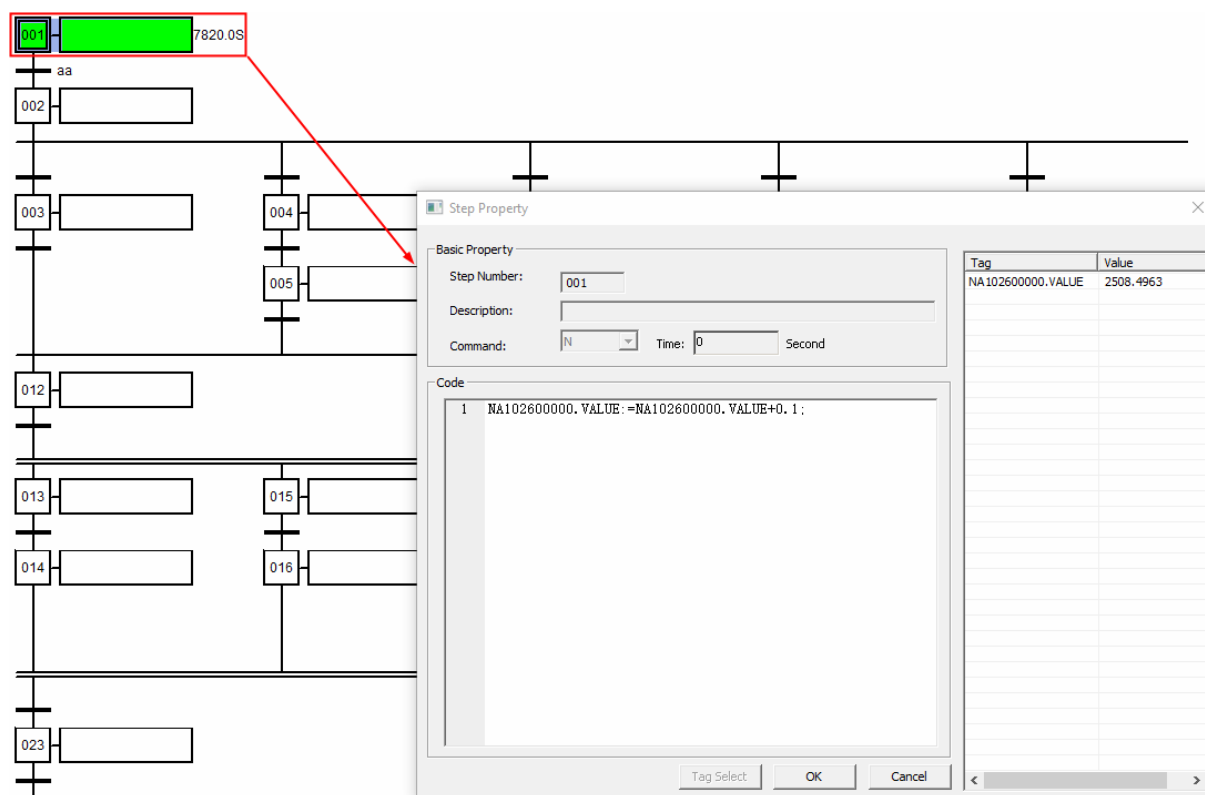
The SFC program debugging interface shown in the figure above mainly includes the following areas:

Area Name	Description
Mode/Status	The mode and the status of the current program.

Area Name	Description
Manual/Auto Area	<ul style="list-style-type: none"> <li>● Auto: By default, when the transition condition is met, the next step is automatically activated and executed.</li> <li>● Manual: After the currently activated step is completed, the step continues to be executed. Only when the continuing command is accepted and the transition conditions are met, the next step is activated and executed.</li> <li>● Auto start: After it is checked, the program will automatically enter the first step to run again after it finishes running manually.</li> </ul>
Control Command Area	<p>The SFC program supports controlling by the following commands:</p> <ul style="list-style-type: none"> <li>● Jump, in the manual mode, the target step will be activated.</li> <li>● Continue, continue executing the program in manual mode.</li> <li>● Start, makes the program enter the RUNNING status from IDLE or enter RUNNING from STOPPED status.</li> <li>● Stop, makes the program enter the STOPPED status from the RUNNING status.</li> <li>● Reset, makes the program enter the IDLE status from the COMPLETE status or the STOPPED status.</li> </ul>
Running Time	<p>This area contains the total execution time of the program and the current activation step: The total time the program is executed, that is, the time that the program starts counting time from the IDLE state to the RUNNING state until it reaches the total time of the COMPLETE state.</p> <p>The current activated step, which is the step number of the currently activated step.</p>
Step	<ul style="list-style-type: none"> <li>● Green indicates that the step has been activated. After the step is activated, the execution time of the step is displayed in the “Run” on the right.</li> <li>● Gray indicates completion of execution</li> <li>● White means execution is not activated</li> </ul>
Transition	<p>Green flashing indicates that the current transition condition is not satisfied; double-click to open the transition to see the satisfaction status of the entire transition and a single condition, “×” indicates that the condition is not satisfied, and “√” indicates that the condition is satisfied.</p>

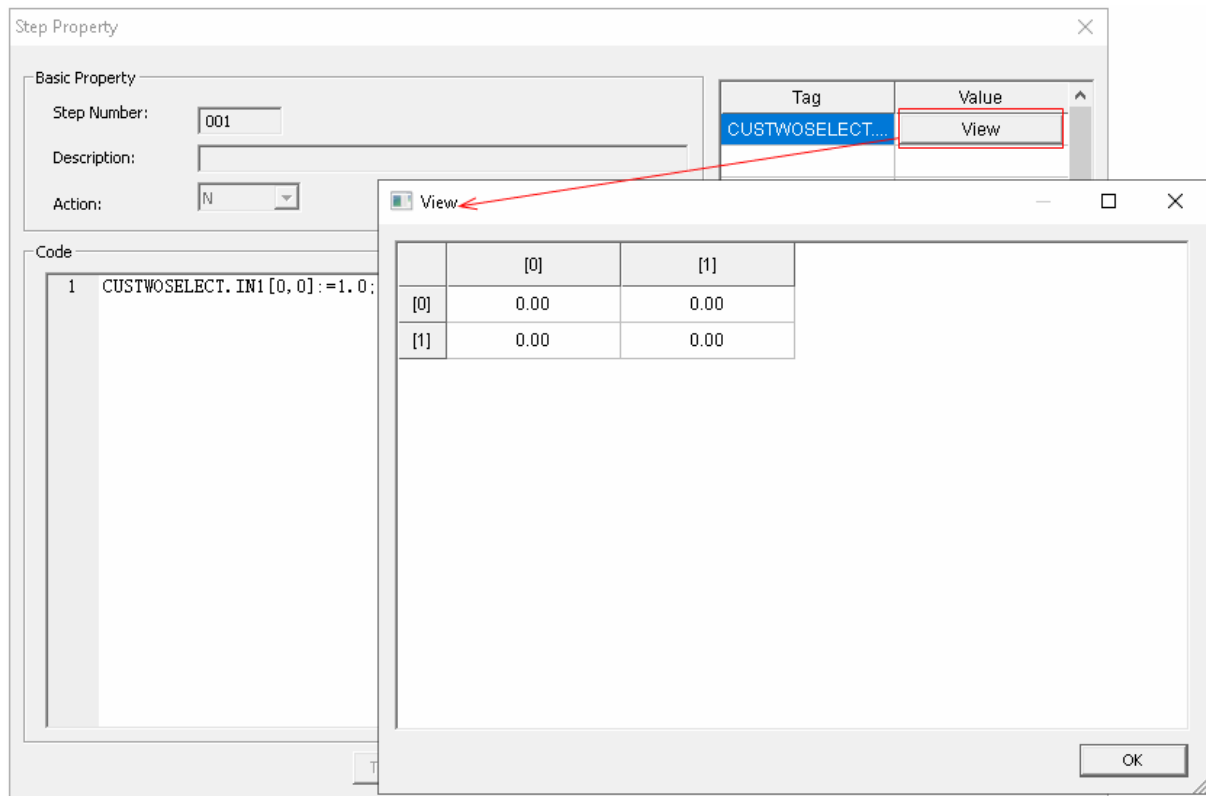
In the SFC debugging interface, double click “step” or “transition” and the detailed configuration interface will pop up, including the real-time data of tags shown in the figure below.





**Figure 4-2 Step property setting in SFC program debugging**

When the referenced tags are of the type of array, you can see “View” button in the real value of tags, click this button and the real-time value of tags of the type of array pops up as shown in the figure below.



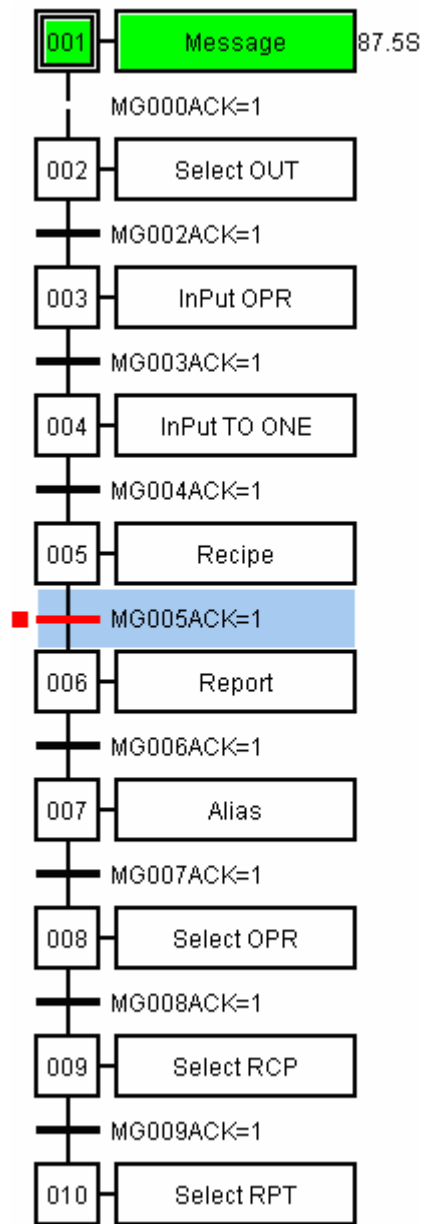
**Figure 4-3 Real-time value example**

#### 4.2.2 Breakpoint Debugging

During the debugging process, you can insert a breakpoint in the transition step, so that you can more clearly understand the execution effect of each step of the program. After adding a breakpoint, the program will pause at the breakpoint during execution.

Through the following operations, you can add breakpoints in the program.

Click the transition step and select the right-click menu command "Add Breakpoint" to add a breakpoint to the currently selected transition step, as shown in the following figure between "005" and "006".



**Figure 4-4 Breakpoint debugging**

In the breakpoint debugging interface as shown in the figure above, different breakpoint statuses mean different definitions:

- Green flashing, represents the transition condition is not satisfied.
- Red flashing, represents the transition condition is satisfied and the breakpoint has been inserted into the transition condition and the breakpoint triggers.

If you need to delete the breakpoint, you can click the breakpoint and select “Delete breakpoint” in the right-click menu. After deleting the breakpoint, the status indication of the breakpoint would disappear and the breakpoint will not be triggered any more.

## Section 5 Monitor SFC Program

---

After creating the SFC program and completing the tag name and logic configuration of the SFC program, the SFC program can be monitored through the tag panel of the SFC program.

### 5.1 Configure Monitoring Authority

Before starting monitoring, confirm that the monitoring user has the monitoring authority for the data grouping where the SFC program tag is.

- 1) Confirm the tag group of the SFC program

When configuring the tag proof SFC program, specify the tag group of SFC program, as in Figure 3-2.

- 2) Open the VFAccess software

Double-click the "User Management" node in the tree on the left.

- 3) Select the monitoring user

In the user list tree, click the user that need to be configured, such as "Admin".

- 4) Configure "Data Group" authority

Click the "Data Group" tab, and check the tag group in the SFC program, as in Figure 3-2.



---


**Tip:**

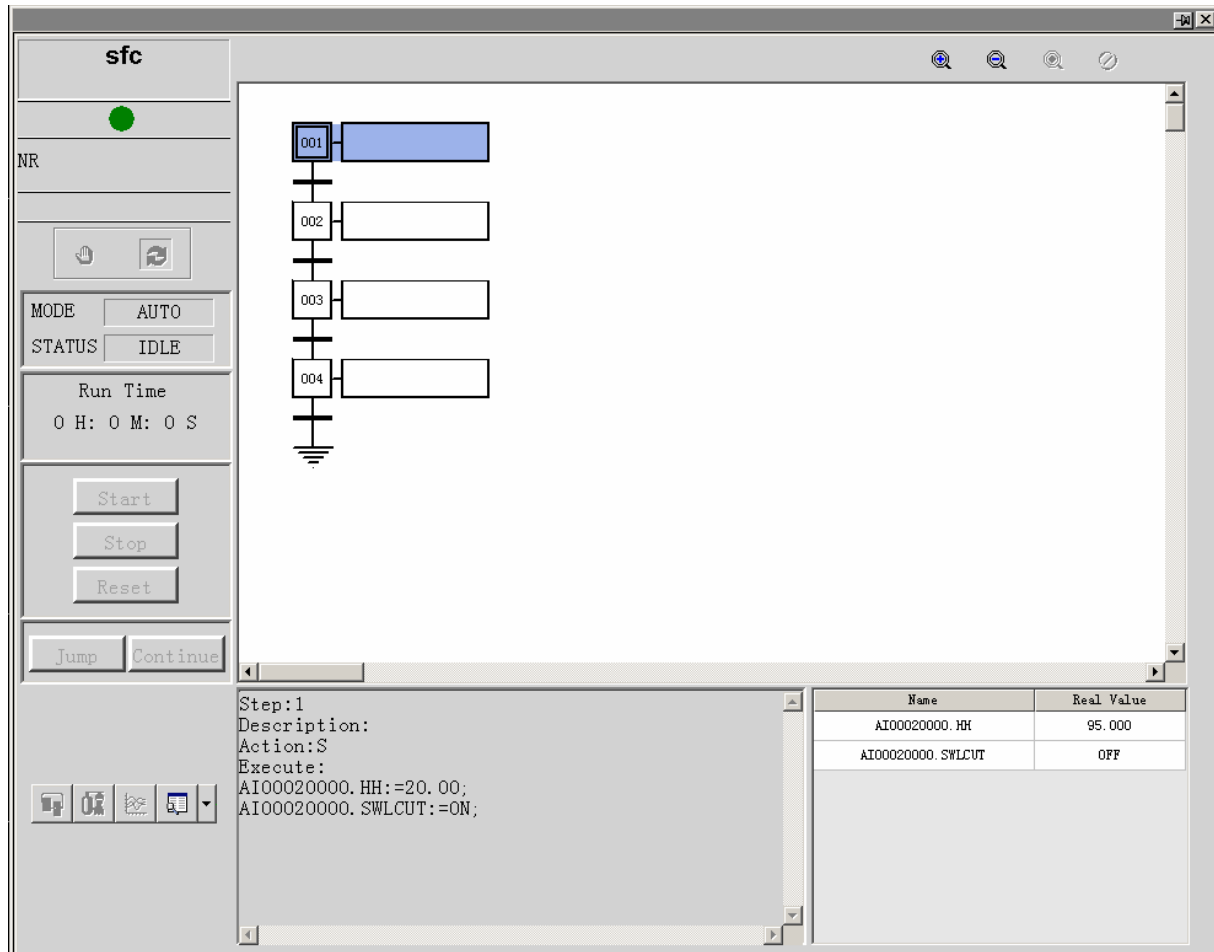
For detailed configuration of the monitoring user's tag monitoring authorities, please refer to *User Access Software User Manual*.

---

### 5.2 Start Monitoring

The SFC program can be monitored by the following operations:

1. Open the High-performanceHMI software and log in.
2. Click  in the upper right corner of the toolbar of the monitoring header and enter the tag name of the SFC program in the search box of the pop-up box to bring up the SFC program panel as shown in the figure below.



**Figure 5-1 SFC Program Panel in the Monitoring**

The basic operations of SFC program in the monitoring period and the debugging period are mostly the same:

- The program status of SFC, the running time and the control command of SFC program are displayed in the left area.
- The right area shows the current SFC program execution status.

Select the step in the program, the configuration information of the step and the real-time value information of the tag associated with the step will be displayed below.

Select the transition in the program, the configuration information of the transition and the real-time value information of the tag associated with the transition will be displayed below.

## Section 6 Parameter Description of SFC Program

### Tags

The SFC program tag can be applied in the user program and monitoring configuration. The reference parameters of the SFC program tag are listed in the following table.

Name	Data Type	Initial Values	Description
CMD	USINT	0	Commands: 1=start, 2=stop, 6=reset, 12=jump.
STATUS	USINT	0	status: 0=initial state; 1=running state; 8=complete state; 10=stop state.
COLDST	BOOL	OFF	Cold start mode: ON=Cold start on; OFF=Cold start on and off.
SWOOS	BOOL	OFF	Function block disable: ON=disable
SWAM	BOOL	ON	Manual/Auto switch: OFF=MANUAL; ON=AUTO.
AU_START	BOOL	OFF	Self-starting: ON=Automatically restart and turn on; OFF=Automatically restart and turn off.
NEXTSTP	USINT	0	Next step
JUMP_TO	USINT	0	Jump objective
RUBn	BOOL	OFF	The breakpoint status of RUNNING transition n, where n is the transition number, and the value range is from 1 to 128. The values of RUBn is: ON=breakpoint is valid; OFF=breakpoint is invalid.
RUETn	BOOL	OFF	The final judgment results of the RUNNING transition n, and n is from 1 to 128. The values of RUETn is: ON = satisfied; OFF = not satisfied.
RUTn	USINT	0	The judgment results of the sub-expression of the RUNNING transition n, n is the transition number, and the value range is from 1 to 128. Each bit represents the established status of single expression. RUTn is used by bit from high to low. 1 represents being established, while 0 represents not being established.
RUSnS	USINT	0	The status of running STEP n, n is the step number and the value range is from 1 to 128. RUSnS represents different statuses according to different bit and it displays in decimal system: RUSnS values: The 1 <sup>st</sup> bit, 0=inactivated status; 1=activated status The 2 <sup>nd</sup> bit, 0=not executed status, 1=executed status; The 3 <sup>rd</sup> bit, 0=uncompleted status, 1=complete status. If this value is displayed as 6 (binary system: 110), then the step status is completed, executed, not activated.
RUAnS	USINT	0	The status of RUNNING step n command, n is the step number, the value range is from 1 to 128. RUAnS=0, it means this step hasn't been executed. RUAnS=1, it means this step is being executed. RUAnS=2, it means this step has been executed.
RUSnT	UDINT	0	The activated time for RUNNING step n, The unit is millisecond. n is the step number. Its range is from 1 to 128.
RUAnT	UDINT	0	The time for RUNNING step n, executing actions. The unit is millisecond. n is the step number. Its range is from 1 to 128.
TIMERSn	UDINT	0	Timer n counter (unit: second) and n is from 1 to 4.

Name	Data Type	Initial Values	Description
TIMERMn	REAL	0	Timer n counter (unit: minute ) and n is from 1 to 4.
TMRPTNn	USINT	0	Timer n running modes (0=normal, 1=count down) and n is from 1 to 4.
CMDTn	USINT	0	Timer n operates, n is 1~4. The value range of CMDTn is: 0=Reset; 1=Start; 2=Pause; 3=Synchronize, the minute timer data is synchronized to the second timer; 4=Synchronize, the second timer data is synchronized to the minute timer.
TOT_TIME	UDINT	0	Total time (unit: second)

## Section 7 Appendix—SFC Language

### 7.1 Supported ST Function Keywords

The keywords of ST functions supported by the SFC program in the High-performanceHMI system software are listed in Table 7-1.

**Table 7-1 ST function Keywords**

Key Words	Function
WHILE; DO; END_WHILE	Constitute WHILE statement
FOR; TO; BY; END_FOR	Constitute FOR statement
IF; THEN; ELSEIF; END_IF	Constitute IF statement
ELSE	Constitute IF statement or CASE statement
REPEAT; UNTIL; END_REPEAT	Constitute REPEAT statement
CASE; OF; END_CASE	Constitute CASE statement
EXIT	Exit statement
AND; NOT; MOD; OR; XOR	Operators
ON; OFF; TRUE; FALSE	BOOL Type
RETURN	It is the keywords of ST language and cannot be used in the custom function block.

### 7.2 Supported ST Function Operator

The ST function operators supported by the SFC program in the High-performanceHMI system software and their priorities are listed in the following table.

**Table 7-2 ST Operators SFC Supported**

Sequence Number	Operator Name	Operator	Priority
1	Bracket	()	1
2	Function Call	SIN()	2
3	Logic NOT	NOT, ~	3
4	Multiply	*	5
5	Modulo	MOD	5
6	Divide	/	5
7	Add	+	6
8	Subtract	-	6
9	Greater than	>	7
10	Greater than or equal to	>=	7
11	Less than	<	7
12	Less than or equal to	<=	7



Sequence Number	Operator Name	Operator	Priority
13	Not equal to	<>	8
14	Equal to	=	8
15	Logic XOR	XOR	9
16	Logic AND	AND, &	10
17	Logic OR	OR,	11

## 7.3 Timer

The SFC software supports the read/write seconds timer TIMERS, the hundred millisecond timer TIMERMS, and the minute timer TIMERM.

**Table 7-3 Timer Type and its Relevant Information**

Type	Identifier	Number	Data Type	Range
Second timer	TIMERS	512	UINT	0~65535
100-millisecond timer	TIMERMS	128	UINT	0~65535
Minute timer	TIMERM	128	UINT	0~65535

## 7.4 Global Array

The SFC software supports multiple types of global arrays, each of which can contain up to 32 rows x 128 columns.

**Table 7-4 Global Array High-performanceHMI Supports**

Identifier	Array Type (row*column)	Element Type
ARRREAL[X,Y]	32*128(X=0~31, Y=0~127)	REAL
ARRBOOL[X,Y]	32*128 (X=0~31, Y=0~127)	BOOL
ARRUINT[X,Y]	32*128 (X=0~31, Y=0~127)	UINT
ARRBOOLE[X,Y]	32*32 (X=0~31, Y=0~31)	BOOL
ARRREALE[X,Y]	32*32 (X=0~31, Y=0~31)	REAL

## Section 8 Revision

---

*Table 8-1 Retrofit list of the version*

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