

Sequential Function Chart (SFC) Application Guide

Mar 2015



These instructions do not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met during installation, operation, and maintenance. The information is supplied for informational purposes only, and GE makes no warranty as to the accuracy of the information included herein. Changes, modifications, and/or improvements to equipment and specifications are made periodically and these changes may or may not be reflected herein. It is understood that GE may make changes, modifications, or improvements to the equipment referenced herein or to the document itself at any time. This document is intended for trained personnel familiar with the GE products referenced herein.

GE may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not provide any license whatsoever to any of these patents.

Public Information – *This document contains non-sensitive information approved for public disclosure.*

GE provides the following document and the information included therein as is and without warranty of any kind, expressed or implied, including but not limited to any implied statutory warranty of merchantability or fitness for particular purpose.

For further assistance or technical information, contact the nearest GE Sales or Service Office, or an authorized GE Sales Representative.

Revised: Mar 2015
Issued: Jan 2009

© 2009 - 2015 General Electric Company.

*** Indicates a trademark of General Electric Company and/or its subsidiaries.
All other trademarks are the property of their respective owners.**

**We would appreciate your feedback about our documentation.
Please send comments or suggestions to controls.doc@ge.com**

Document Updates

Location	Description
Throughout the document	For Public Disclosure footer updated for GE Rebranding effort
Cover Illustration	Removed for GE Rebranding effort

Safety Symbol Legend



Warning

Indicates a procedure or condition that, if not strictly observed, could result in personal injury or death.



Caution

Indicates a procedure or condition that, if not strictly observed, could result in damage to or destruction of equipment.



Attention

Indicates a procedure or condition that should be strictly followed to improve these applications.

Control System Warnings



Warning

To prevent personal injury or damage to equipment, follow all equipment safety procedures, Lockout Tagout (LOTO), and site safety procedures as indicated by Employee Health and Safety (EHS) guidelines.



Warning

This equipment contains a potential hazard of electric shock, burn, or death. Only personnel who are adequately trained and thoroughly familiar with the equipment and the instructions should install, operate, or maintain this equipment.



Warning

Isolation of test equipment from the equipment under test presents potential electrical hazards. If the test equipment cannot be grounded to the equipment under test, the test equipment's case must be shielded to prevent contact by personnel.

To minimize hazard of electrical shock or burn, approved grounding practices and procedures must be strictly followed.



Warning

To prevent personal injury or equipment damage caused by equipment malfunction, only adequately trained personnel should modify any programmable machine.



Warning

Always ensure that applicable standards and regulations are followed and only properly certified equipment is used as a critical component of a safety system. Never assume that the Human-machine Interface (HMI) or the operator will close a safety critical control loop.

Contents

- 1 Overview 7**
- 1.1 SFC Structure 8
 - 1.1.1 Steps 8
 - 1.1.2 Transitions 8
 - 1.1.3 End Transition 9
 - 1.1.4 Actions 9
 - 1.1.5 SFC Evolution and Progression 10
 - 1.1.6 Size Limitations 14
- 1.2 Advanced Concepts and Techniques 15
- 1.3 View and Control an SFC 16
 - 1.3.1 Human Machine Interface (HMI) 16
 - 1.3.2 Naming Guidelines 16
 - 1.3.3 EGD Coherency 16
 - 1.3.4 Time Resolution 17
 - 1.3.5 Frame Multipliers 17
- 1.4 Download an SFC 17
 - 1.4.1 Online Download Procedure 17
 - 1.4.2 Controller-Parallel Online Download 17
- 2 Advanced Concepts 19**
- 2.1 Control an SFC from the ToolboxST Application or from CIMPLICITY 19
 - 2.1.1 Monitor interface 19
 - 2.1.2 Control interface 21
- 2.2 SFC and ToolboxST Application Code Libraries 22
- 2.3 Transition Condition Permit Block Option 22
- Glossary of Terms 23**

Notes

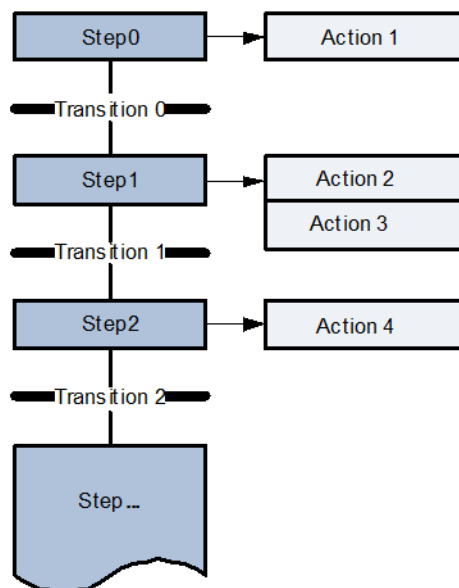
1 Overview

A Sequential Function Chart (SFC) is a graphical programming model typically used in Programming Logic Controllers (PLCs) to define the operational flow for a given process. SFCs allow the user to construct a flow chart consisting of three types of components: Steps, Actions, and Transitions.

Note Mark* VIe V03.05 product features an IEC 61131-3 compliant SFC implementation. The International Electrotechnical Commission (IEC) is an international standards organization which works in conjunction with the Institute of Electrical and Electronics Engineers (IEEE®), the International Organization for Standardization (ISO), and the International Telecommunications Union (ITU) to develop standards for electronic and electrical technologies.

SFCs use combinations of Steps, Actions, and Transitions to define the operational flow. Steps are connected to Transitions, and Transitions are connected to Steps. It is not possible to progress from one Step to another without going through a Transition. Similarly, it is also not possible to progress through multiple Transitions without being inside a Step. Also, zero or more Actions can be associated with each Step in the SFC.

SFCs enables the user to develop Mark VIe control logic. They are used to define and manage plant startup and shutdown sequences in which several steps must be performed in a specific order for a system to function properly. SFCs also facilitate automate procedures that run based on alarms received from inside a controller. SFCs are typically used in batch processing applications and can also be used to change modes of control within a given system.



Valid SFC Layout

1.1 SFC Structure

1.1.1 Steps

Steps are fundamental building blocks of an SFC. Steps can be thought of as states in which actions are performed. Each Step has a binary status condition, either active or inactive and multiple Steps can be active at the same time within an SFC. When an SFC is started or is otherwise reset, only the **Initial Step** is active at first. There is only one Initial Step per SFC. The Initial Step is very important in that it is responsible for initializing any state variables associated with the SFC. Steps are activated and deactivated as Transition Conditions are satisfied. Each Step has a *Name* field as well as a *Step Time* field associated with it that indicates the elapsed time the Step has been active since the initiation of the Step. The current implementation of the Mark VIe control product can only allow a maximum of 256 Steps contained within an SFC. The Step Block figure and the Initial Step Block figure display the symbols for a Step block and an Initial Step block respectively, from within the SFC diagram editor.



Initial Step Block



1.1.2 Transitions

Transitions are another fundamental concept critical to the flow of operations within an SFC. Transitions are the conditions that enable control to be passed from one or more preceding Steps to one or more successive Steps along a directed link. Transitions are represented as a horizontal line across the vertically directed link between two Steps. Each Transition has a Name field as well as a *Transition Condition* associated with it that is essentially the result of the evaluation of a singular Boolean expression. The current implementation of the Mark VIe Control product can only allow a maximum of 256 Transitions contained within an SFC. The Transition Block figure displays the symbol for a Transition block from within the SFC diagram editor.



Transition Block

1.1.3 End Transition

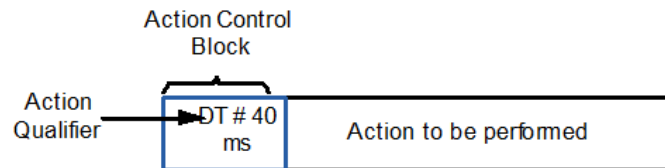
An End Transition is a special Transition used to complete an SFC. When the Transition Condition of an End Transition is set to True, the SFC is placed into the complete state. When an SFC is in the complete state, each Step, Transition, and Action is disabled. Each SFC can have one End Transition, but is not required to have an End Transition. The End Transition figure displays the symbol for the End Transition from within the SFC diagram editor.



End Transition Block

1.1.4 Actions

Actions are the third major concept in SFCs. Each action is associated with a Step and is governed by an *Action Control Block*. When a Step becomes active, the actions associated with that Step are performed according to the behaviors specified by the Action Control Block for each particular Action. These Action Control Blocks are configured using *Action Qualifiers*. The number of serviceable Actions per Step in an SFC depends on the types of operations performed in each Action as well as the overall frame rate of the application. It is recommended that the user limits the number of Actions per Step to 16. The Action Block figure displays the symbol for an Action block from within the SFC diagram editor.



Action Block

1.1.4.1 Action Qualifiers

Each Action can be characterized through the use of an **Action Qualifier**. The Action Qualifier is used to configure the Action Control Block which governs the operation of each Action.

Action Qualifiers

Number	Qualifier	Description
1	None	Non-Stored (null qualifier)
2	N	Non-Stored
3	R	Overriding Reset
4	S	Set (Stored)
5	L	Time Limited
6	D	Time Delayed
7	P	Pulse
8	SD	Stored and Time Delayed
9	DS	Delayed and Stored
10	SL	Stored and Time Limited
11	P1	Pulse (Rising Edge)
12	P0	Pulse (Falling Edge)

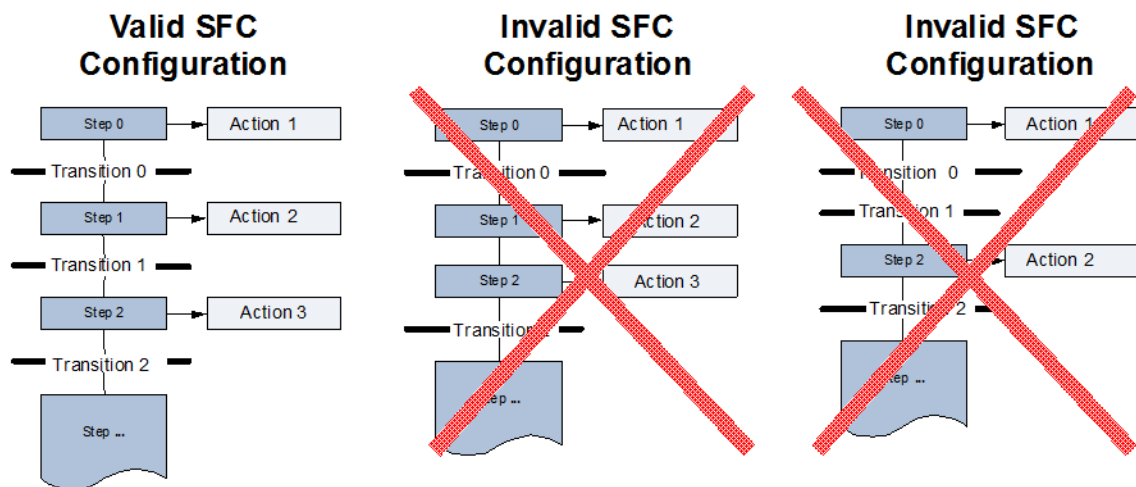
1.1.4.2 Action Control Blocks

Before an Action associated with an active Step runs, the Action Control Block must enable the Action. The Action Control Block is configured by the *Action Qualifier* when using one of the options provided in the Action Qualifiers table above. The Action Control Block allows for trigger conditions and limits to be applied to an Action such as trigger from pulses, time delay, time limits, and many more depending on the *Action Qualifier* designated. *Action Control Blocks* are in milliseconds.

1.1.5 SFC Evolution and Progression

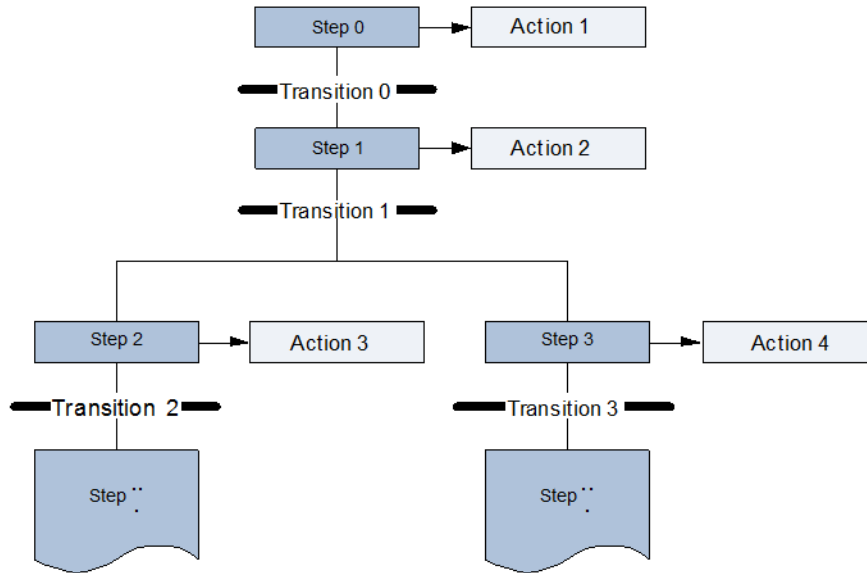
Rules of Evolution define the activation sequence in an SFC. Entry into the SFC begins at the Initial Step. There is only one Initial Step per SFC, and it is activated when starting or resetting the SFC. The state variables for the SFC are also initialized in the Initial Step. Traversals of active Steps, also called *Evolutions*, follow vertically directed links as Transitions are cleared. A Transition becomes enabled when all Steps directly preceding the Transition become active. When a Transition is disabled, or cleared, all of the directly preceding Steps are automatically deactivated. After the preceding Steps are all inactive, the Steps immediately following the directly linked Transition are activated.

An important concept to understand is the Step and Transition link relationship. Two Steps cannot be directly connected. There is always a Transition separating two Steps, however it is possible to achieve the same desired effect by using a Transition with a Transition Condition that is always true. Also, no two transitions can be directly connected. The ordering is always Step-Transition-Step-Transition in alternating pattern. Examples of valid and invalid SFC configurations are provided in the Various SFC Configurations figure.

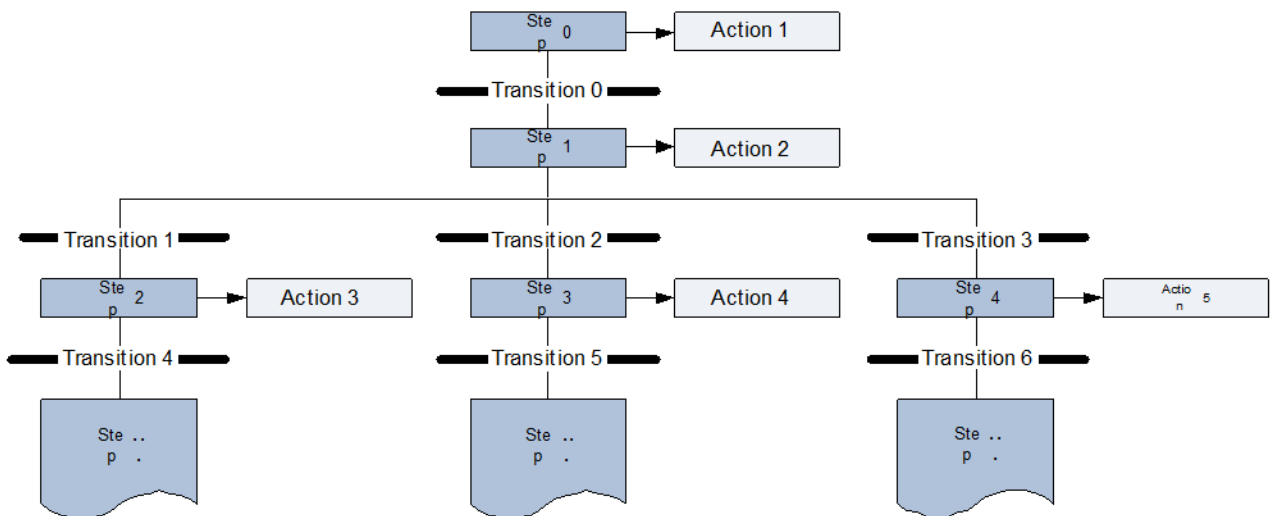


Various SFC Configurations

Divergence allows multiple Steps in an SFC to become active simultaneously after a Transition is cleared. Similarly, while a Step is active, multiple Transitions can be enabled and evaluated concurrently. Divergence support in the SFC feature implementation provides application developers with the flexibility to create sophisticated processes through a simplified graphical means. Example SFC layouts demonstrating divergent Transition paths are provided in the Divergent Steps figure and the Divergent Transitions figure. When multiple Transitions are enabled, and one of the Transition Conditions is met, the remaining parallel divergent Transitions are disabled automatically. There is a maximum width of 32 divergences within an SFC. The Transition Conditions are evaluated from left to right as defined in the SFC diagram.

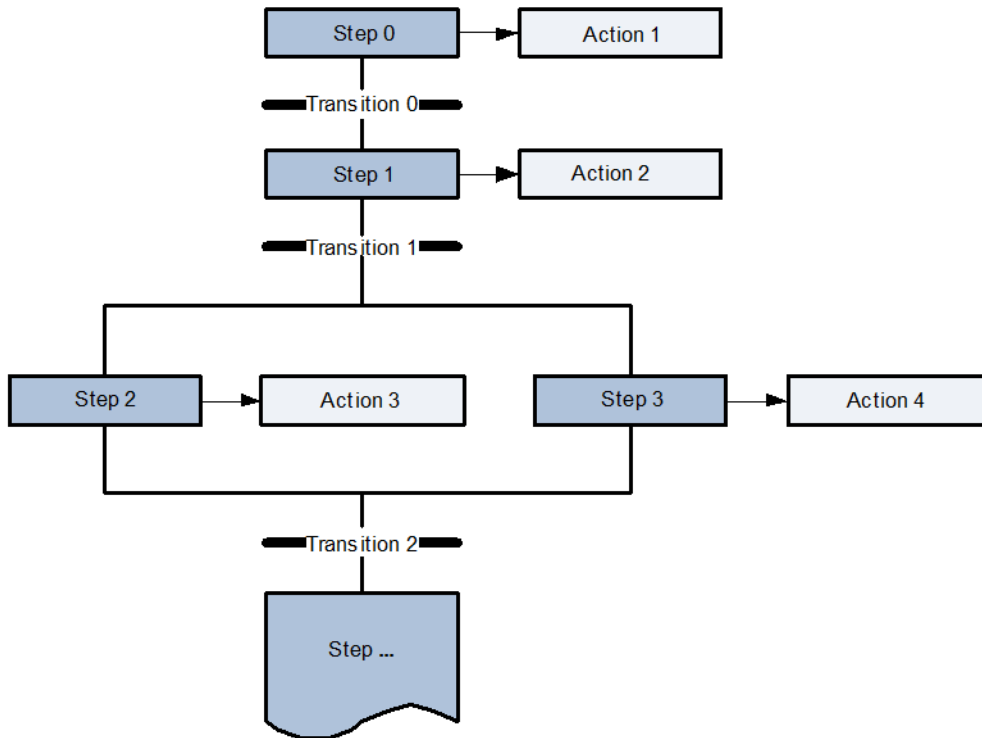


Divergent Steps

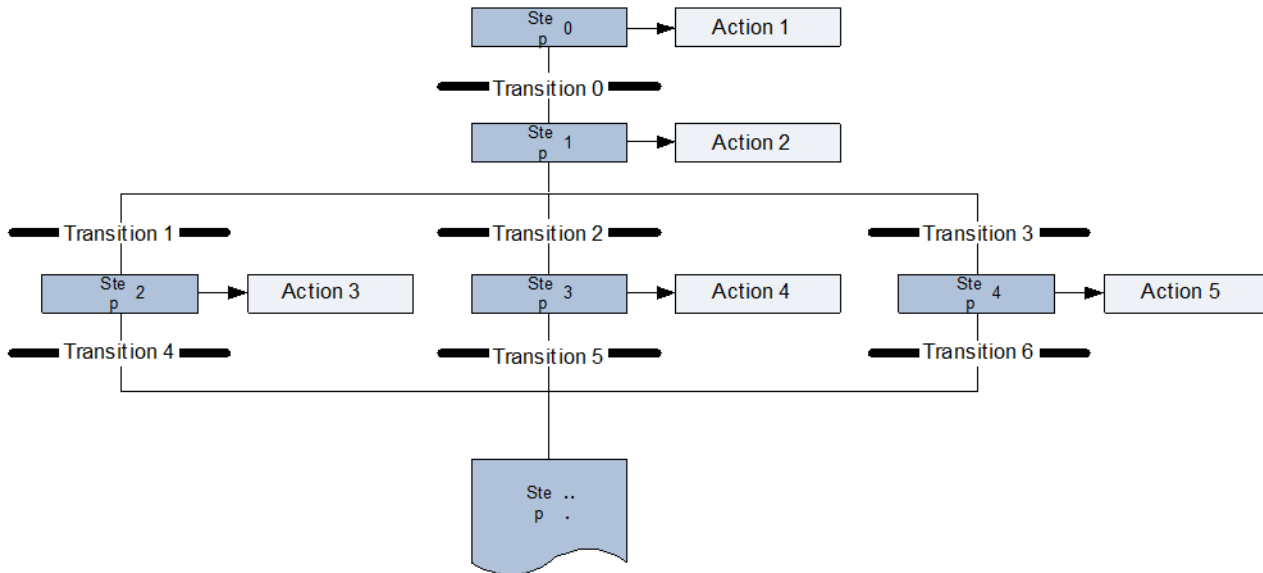


Divergent Transitions

All diverging paths eventually must converge back to a single link. Failure to provide complete convergence results in an invalid SFC. As presented in the Convergent Step Paths figure, the concurrently active Step 2 and Step 3 continue to perform Actions 3 and 4 respectively until Transition Condition 2 has been met. When that condition is met, both Step 2 and Step 3 are disabled, Transition 2 is deactivated, and any Steps directly linked after Transition 2 become active. During Transitions, such as in the Convergent Transitions figure, only one of the three vertical paths is active since the divergence rules disable all parallel Transitions after any one of the Transition Conditions have been met. This provides convergence by essentially closing the paths parallel to the active sequence.

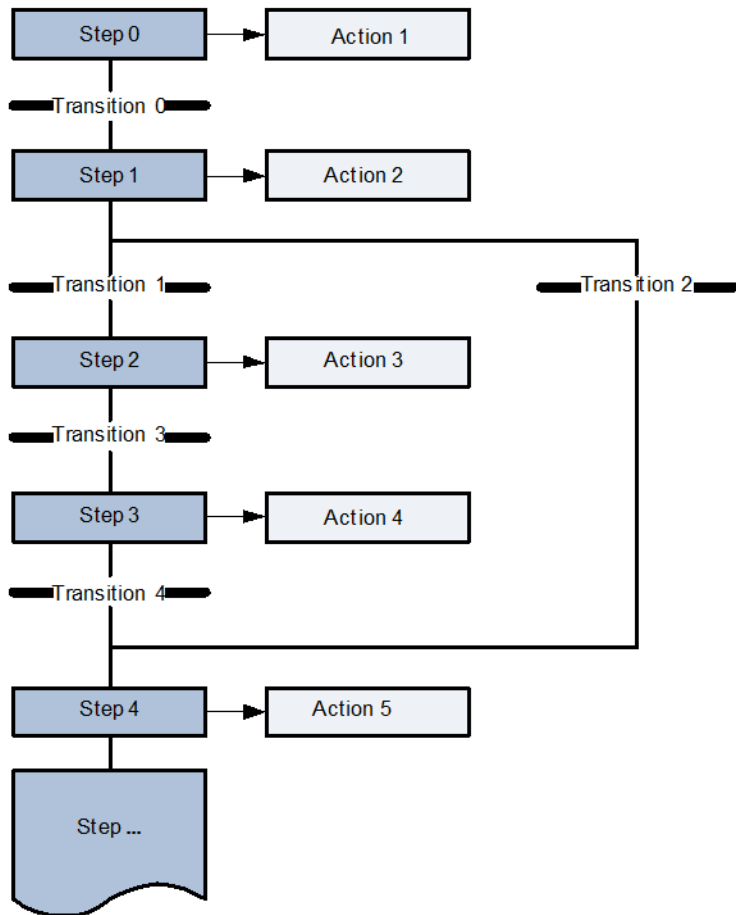


Convergent Step Paths



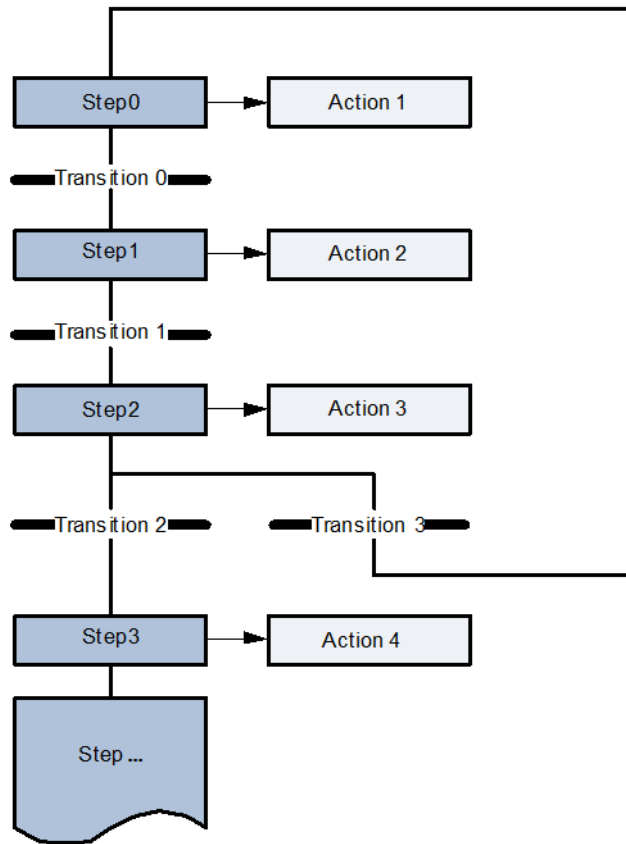
Convergent Transitions

Sequence Skip is a specialized case of selecting a divergent parallel Transition branch. This parallel branch contains no Steps and effectively omits a sequence of one or more Steps that are parallel to it by directly linking to a Step further along in the SFC. The *Sequence Skip* concept is demonstrated the *Sequence Skip Example SFC* figure. A *Sequence Skip* maintains the Step and Transition link relationship.



Sequence Skip Example SFC

A *Sequence Loop* is a special application of a Sequence Skip that links backward to a previous Step in the SFC. This Sequence Loop structure is a very powerful and flexible feature that provides the ability to define control loops in which a particular sequence of Steps can be repeated based on a certain set of conditions. An example SFC containing a Sequence Loop is provided in the sequence Loop Example SFC figure. In this particular example, the loopback path allows for progression back to the Initial Step, however, Sequence Loops can converge back to any point in the SFC provided that the Step and Transition link relationship is maintained.



Sequence Loop Example SFC

1.1.6 Size Limitations

There are several size restrictions and limitations defined for an SFC. A maximum of 256 Steps can be contained in an SFC. Typical SFCs do not exceed 25 Steps. Similarly a maximum of 256 Transitions are allowable. Typical SFCs do not exceed 25 Transitions. Divergent paths as well as their converging counterparts are limited to a total number, or width, of 32. Typical SFC divergence and convergence paths are two to five Transitions wide. Since sophisticated applications can exceed these size limitations, the Mark VIe product provides a feature which enables an SFC to be started from another SFC. This feature allows an application to be sectioned into multiple SFCs to meet these sizing restrictions.

1.2 Advanced Concepts and Techniques

A *Hold Condition* is associated with Transitions and is essentially a means of indicating that a Transition does not allow an SFC to progress even if the Transition's Transition Condition evaluates to True unless there is additional input from the user. The user can place an *Operator Hold* on any Transition in an SFC. As the SFC progresses and gets held up on at a Transition, the user can acknowledge the hold and allow the SFC to continue to evolve. The user can also remove Operator Holds from the SFC. *Implicit Hold*, which is a Hold Condition, cannot be removed from a Transition by the user, similar to an Operator Hold. An Implicit Hold must be acknowledged for an SFC to progress.

In contrast to Hold Conditions is the concept of a *Force Condition*. A Force Condition can be placed on a Transition to cause a progression to the next Step regardless of the Transition Condition. When an SFC has advanced through a Transition with a Force Condition applied to it, the Force Condition is cleared.

If a Force is placed on a currently Held Transition, the Force overrides the Hold, to advance the state of the SFC. It does not matter if the Transition is Held due to an Operator Hold or an Implicit Hold.

There are two modes of operation provided for SFCs in the Mark VIe product, *Single Step Mode* and *Free Running Mode*. The user can select the desired operation mode through the controls featured in the ToolboxST* application.

In Single Step Mode, an SFC does not progress to the next Step through one or more Transitions that evaluate to True without additional input from the user. This means that in Single Step Mode, Implicit Hold conditions are placed on all Transitions in the SFC, and the user cannot remove these holds. You cannot add or remove Operator Hold Conditions from an SFC in Single Step Mode.

In Free Running Mode, an SFC is able to progress from Step to Step through Transitions that evaluate to True without any additional input from the user. Initially when an SFC is started, no Operator Hold conditions are active in the SFC. The user can add or remove Operator Hold conditions as needed. No Implicit Hold conditions exist in Free Running Mode. Also, if user switches back from Single Step Mode to Free Running Mode, all previously set Operator Hold conditions are restored.

Another advanced concept regarding Sequential Function Charts is the confirmation of *Actions of Interest*. Each Step in an SFC can contain multiple actions, and the Mark VIe SFC implementation provides a means to select certain Actions for confirmation of completion. These selected Actions are called Actions of Interest. Each Step with Actions of Interest provides a variable that subsequent Transactions can use in their Transition Condition logic, ensuring that the SFC does not progress past that particular Transition until all Actions of Interest criteria has been satisfied. After all of the Actions of Interest have been performed and the Transition Conditions have evaluated to True, the SFC progresses as normal to the next Step.

1.3 View and Control an SFC

A user can pause or continue an SFC from the ToolboxST application. The ToolboxST application interfaces with the controller running the SFC and can invoke a pause or continue operation by modifying the SFC's exposed Control Variables. When an SFC is paused, all actions for all Steps are stopped. No progression of the SFC occurs while the SFC is paused. When paused, the user can also continue the SFC which re-enables all of the Steps in the SFC and continues progression as normal. Pausing an SFC is different from applying Operator Hold conditions on all Steps. Operator Hold conditions are intended to stop progression of the SFC through an extension of the Transitions between Steps. All of the Actions associated with the active Steps are still performed in that case. In a paused SFC however, no actions are performed and no Transitions are evaluated.

An SFC can also be reset by the user. Resetting an SFC stops the current progress of the SFC and returns the application to the Initial Step. The Initial Step is also responsible for initializing the state variables for the SFC so after a reset no previous state information is kept. Any state variables such as flags or counters set by the user is reset with the initial value specified by the user. For safety, a user can only reset an SFC when it is paused or it is complete. This is done to ensure the integrity of state information for an actively running SFC as well as to prevent an accidental reset of an SFC.

1.3.1 Human Machine Interface (HMI)

A drag-and-drop feature allows you to drag certain application blocks, as well as SFC Step and Transition objects from the ToolboxST application and drop them onto an open CIMPLICITY project. When dropped, CIMPLICITY instances an object with a predetermined graphical view as well as data connections tied to the ToolboxST application.

Note Application engineers need to provide a SFC.cim CIMPLICITY library inside of a CIMPLICITY project. The CIMPLICITY SFC.cim library needs to contain definitions and instance scripts for a STEP and TRANSITION object for the drag and drop feature to be functional.

1.3.2 Naming Guidelines

It is recommended that every SFC in an application have a unique name regardless of Program placement. This guideline also applies to instancing SFCs from a Library into an application. All Steps and Transitions need unique names. A Step and a Transition cannot have the same name. Finally, all Actions defined in a Step need unique names.

1.3.3 EGD Coherency

Information regarding the state of an SFC is produced by a controller over EGD for other components such as a server running the ToolboxST application or an HMI. Although, state information on EGD is guaranteed to be produced coherently, there are no guarantees placed on the coherency of acquired EGD. Therefore it is advised to not trigger control logic based on EGD state information gathered from another controller. This means that the user cannot rely on EGD for developing applications with tight inter-controller control loop structures.

1.3.4 Time Resolution

The time resolution provided within the SFC implementation is on the order of milliseconds precision. The upper limit for the elapsed time spent inside an active Step is 4294967295 milliseconds or 49.71 days. The *Step Time* remains clamped at 4294967295 milliseconds to prevent a rollover or overflow condition if the elapsed Step time exceeds the upper limit.

The time values for L and D Action Qualifiers are defined in milliseconds. The upper limit for these time values is 4294967295 milliseconds or 49.71 days.

1.3.5 Frame Multipliers

Frame Multipliers allow you to specify the frequency at which the application is run on the controller. A multiplier as well as an offset can be configured to run the application at the desired frequency of frames. The maximum number of frames that can be skipped by a Frame Multiplier is 512.

1.4 Download an SFC

1.4.1 Online Download Procedure

The online downloading procedure for applications using SFCs differs from previous versions of the product for TMR configurations. In a TMR configuration, the state information for an SFC is voted by all three controllers each frame. Therefore, to maintain consistency throughout an online load, the designated controller must be healthy and be the initial controller for the first download. Also, the designated controller must not switch while an online load is in progress. The new downloading procedure affects only TMR configurations as state information is not voted in Simplex and Dual configurations.

1.4.2 Controller-Parallel Online Download

Controller-parallel online downloads are highly discouraged when downloading a redundant controller with an application which contains SFCs. In Controller-parallel online downloads, all redundant controllers are downloaded immediately instead of using the standard download procedure. There is no automatic mechanism which guarantees the state information of the SFCs remain consistent through a Controller-parallel online download. If one still wishes to perform a parallel online download of an application which contains SFCs to a redundant controller, the user must manually pause each SFC in the application, perform the Controller-parallel online download, and finally manually un-pause each SFC in the application. This is the only way to preserve state information through a Controller-parallel online download.

Notes

2 Advanced Concepts

2.1 Control an SFC from the ToolboxST Application or from CIMPLICITY

There can be applications that require automated control of an SFC from application code. Also, there can be HMIs that require control over SFCs.

Note To control an SFC from a CIMPLICITY screen, the SFC information must be published on EGD. The HMI running the CIMPLICITY screen must also consume the EGD page.

2.1.1 Monitor interface

An SFC has two interfaces (monitor and control) that are implemented by variables. The SFC monitor interface allows you to display helpful information about an SFC. The following table lists all available monitor interface items for an SFC named MySFC with an Initial Step, containing a single Action named Action1, and a single Transition named Trans1.

Monitor Interface Item	Sample Variable Name	Value Description	Data Type	External Access
SFC Status	MySFC.STATUS	0 = SFC is paused 1 = SFC is running 2 = SFC is complete	Enumerated Unsigned Integer	Read -Only
SFC Reset Status	MySFC.RESET_STATUS	False = SFC has not been Reset True = SFC has been Reset	Boolean	Read -Only
SFC Major Version	MySFC.Major_Version	Time Stamp of the last Major Structural Change	Unsigned Double Integer	Read -Only
Step Active Flag	MySFC.Initial_Active_Flag	False = Initial Step is not Active True = Initial Step is Active	Boolean	Read -Only
Step Active Time	MySFC.Initial_Step_Time	Time the Initial Step has been Active in ms	Unsigned Double Integer	Read -Only
Step Actions of Interest Status	MySFC.Initial_Actions_Of_Interest_Status	False = Actions of Interest for the Initial Step have not completed True = Actions of Interest for the Initial Step have completed	Boolean	Read -Only
Step Action List	MySFC.Initial_Action_List	False = Step Containing Actions is not active True = Step Containing Actions is active Note: This variable's description field lists all of the Action, by Name, in its Description Field. This enables Action Description 1st and 2nd language support from an HMI.	Boolean	Read -Only

Monitor Interface Item	Sample Variable Name	Value Description	Data Type	External Access
Step Action Active	MySFC.Initial_Action1_Active	False = Configured Action Qualifier has not enabled the User's Action Logic True = Configured Action Qualifier has enabled the User's Action Logic Note: This variable's description field contains the description of the Action. This enables Action Description 1st and 2nd language support from an HMI.	Boolean	Read -Only
Transition Active Flag	MySFC.Trans1_Active_Flag	False = Trans1 Transition is not Active True = Trans1 Transition is Active	Boolean	Read- Only
Transition Condition Values	MySFC. Trans1_Transition_Condition_Value	False = Trans1 Transition Condition is False True = Trans1 Transition Condition is True	Boolean	Read -Only
Transition Progression Status	MySFC. Trans1_Transition_Condition_Value	0 = No_Progression, Transition has not been progressed through 1 = Transition_Conditon, Transition progressed through due to Transition Condition Value evaluating True 2 = Hold_And_Acknowledge, Transition progressed through due to Transition Condition Value evaluating True, while a Hold applied to the Transition was Acknowledged 3 = Force, Transition progressed through due to a Force applied to the Transition	Enumerated Unsigned Integer	Read -Only

2.1.2 Control interface

The SFC Control Interface provides the controls to enable running and pausing an SFC, resetting a paused or complete SFC, setting an SFC's Mode to Free Running or Single Step, applying and removing Operator Holds on Transitions, applying and removing Acknowledges to Holds applied to Transitions, and applying and removing Forces to Transitions. The following table lists all available control interface items for an SFC named MySFC with an Initial Step and a single Transition named Trans1.

Control Interface Item	Sample Variable Name	Value Description	Data Type	External Access
SFC Run/Pause	MySFC.RUN	False = SFC is Paused True = SFC is Running	Boolean	Read / Write
SFC Reset	MySFC.RESET (Note: Reset can only be performed on an SFC that is paused or complete. The Status variable can be used to determine if the SFC is paused or complete.)	False = Do Not Reset SFC True = Reset SFC	Boolean	Read / Write
SFC Mode	MySFC.MODE	0 = SFC in Free Running Mode 1 = SFC in Single Step Mode	Enumerated Unsigned Integer	Read / Write
Transition Operator Hold	MySFC.Trans1_Hold	False = No Hold placed on Transition Trans1 True = Hold placed on Transition Trans1	Boolean	Read / Write
Transition Operator Hold Acknowledge	MySFC.Trans1_Hold_Acknowledge	False = No Hold Acknowledge placed on Transition Trans1 True = Hold Acknowledge placed on Transition Trans1	Boolean	Read / Write
Transition Force	MySFC.Trans1_Force	False = No Force placed on Transition Trans1 True = Force placed on Transition Trans1	Boolean	Read / Write

2.2 SFC and ToolboxST Application Code Libraries

Application Code Libraries are used to template Application Programs, Tasks, and User Blocks which can be referenced by and instanced into a controller's application code. SFCs can be defined into and be instanced from the ToolboxST application code libraries. When instancing an SFC into application code, use a unique task and SFC name.

2.3 Transition Condition Permit Block Option

Each SFC defined in the ToolboxST application can have the DCS PERMIT block tied to the Transition_Condition_Value automatically. This allows an application engineer to create CIMPLICITY screens that display why a transition condition is evaluating False instead of True. This option is located in the Property Editor for each SFC. This option is called AutoPermitBlock under the new Transition condition options property.

Glossary of Terms

Action A scripted behavior in an SFC that performs predetermined operations.

Action Control Block A software subroutine that governs the Actions contained within a Step. Action Control Blocks are configured by Action Qualifiers, T (Time Input), and P (Pulse Input) items available when configuring an Action.

Action Qualifiers A set of preconditions that manages each Action.

Actions of Interest User-designated Actions that do not allow the SFC to progress past a particular Transition until the criteria for the designated Actions have been satisfied.

End Transition A special Transition used to complete or finish an SFC.

Force Condition A user-induced condition, placed upon a Transition, that is used to cause a progression to the next Step regardless of the Transition's Transition Condition value.

Frame Multipliers Allows the user to specify the frequency at which the application is run on the controller.

Free Running Mode A mode in an SFC in which the SFC advances from Step to Step without any additional input from the user.

Hold Condition A condition, applied to a Transition, that does not allow an SFC to progress through a Transition even if the Transition's Transition Condition evaluates to True unless there is additional input from the user.

Implicit Hold A Hold Condition that must be always explicitly acknowledged by the user in order for an SFC to progress. Implicit Holds cannot be removed from the SFC and are applied to an SFC in Single Step Mode.

Initial Step A Step that is responsible for initializing any state variables associated with the SFC. An Initial Step is always active when an SFC is started or reset.

Operator Hold A Hold Condition that must be always explicitly acknowledged by the user in order for an SFC to progress. Operator Holds can be removed from the SFC.

Rules of Evolution Rules of Evolution

Sequence Loop A software subroutine that allows the SFC to link backward to a previous Step in the SFC. Sequence Loops can converge back to any point in an SFC provided that the Step and Transition link relationship is maintained.

Sequence Skip An SFC design technique which uses a Transition in a sequence selection branch to skip over Steps in an SFC.

Sequential Function Chart (SFC) A graphical programming model to define the operational flow for a given process.

Single Step Mode A mode in an SFC in which the progression is halted at each Step until there is additional input from the user. This SFC mode of operation uses Implicit Holds to promote the behavior described above.

Step A state in an SFC in which actions are performed.

Step Name A name field that can be uniquely applied to each Step in an SFC.

Step Time A time field that indicates the elapsed time the Step has been active since the initiation of the Step.

Transition A condition that enables control to be passed from one or more preceding Steps to one or more successive Steps along a directed link.

Transition Condition The evaluation of a singular Boolean expression.

Transition Name A name field that can be uniquely applied to each Transition in an SFC.



Public Information