

Logix5000

Sequential Function Charts

Source – Rockwell Publication 1756-PM006B-EN-P

Sequential Function Charts (SFC)

What is it?

A Sequential Function Chart (SFC) uses **steps** and **transitions** to perform specific operations or actions.

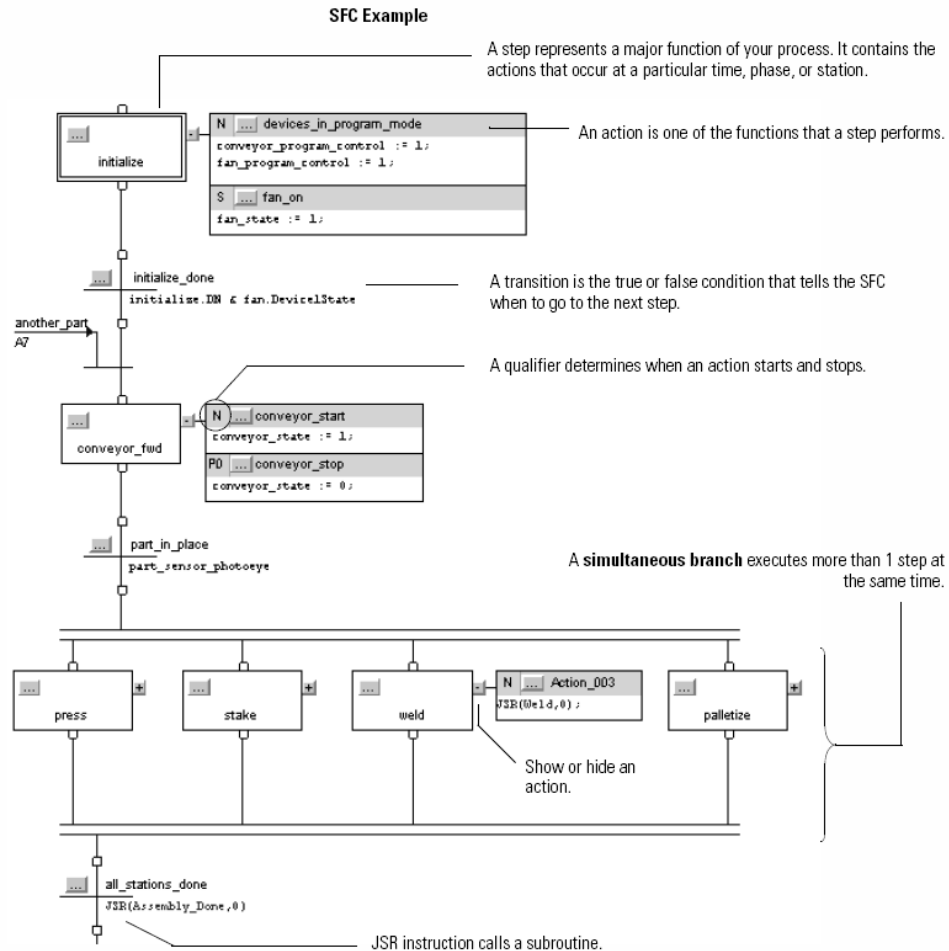
Features

- One of the 4 programming languages outlined in IEC-16131-3
- Similar to a flow chart
- Built-in state engine

Advantages

- Graphical division of processes into its major logic pieces
- Faster repeated execution of individual pieces of logic
- Simpler screen display
- Easier to troubleshoot

Sequential Function Charts (SFC)

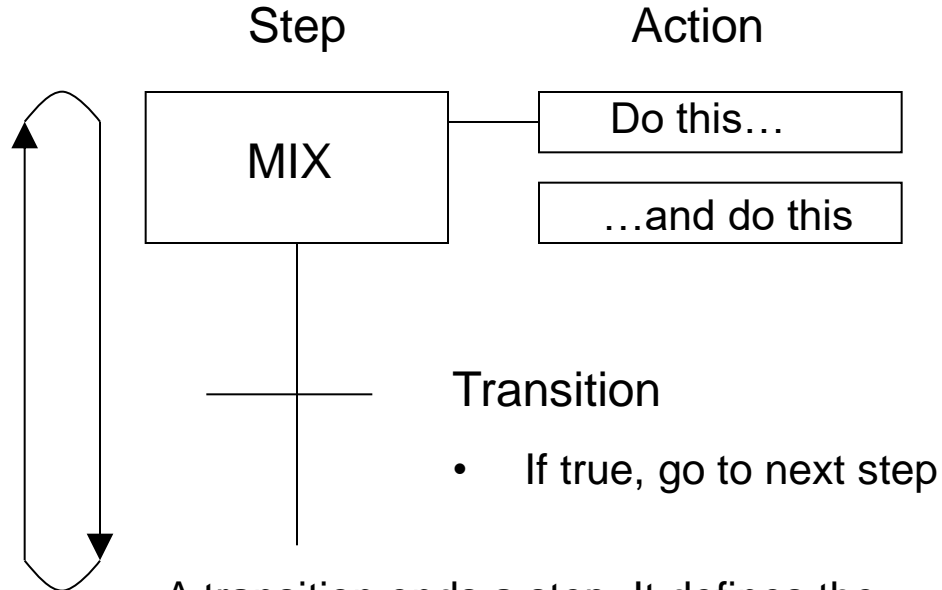


Sequential Function Charts (SFC)

Steps, Actions & Transitions

A step represents a major function of your process. It contains actions that occur at a particular time, phase, or station.

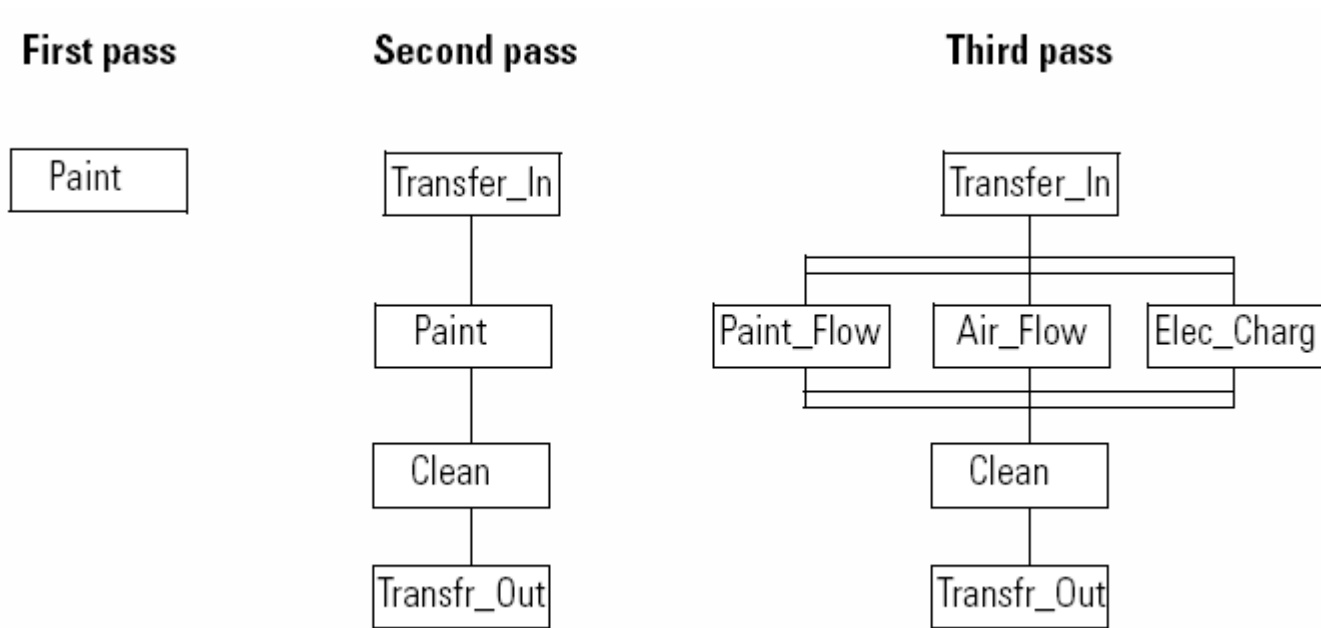
A step executes continuously until a transition tells the SFC to go to the next step.



A transition ends a step. It defines the physical conditions that must occur or change in order to go to the next step.

Sequential Function Charts (SFC)

Steps Guidelines



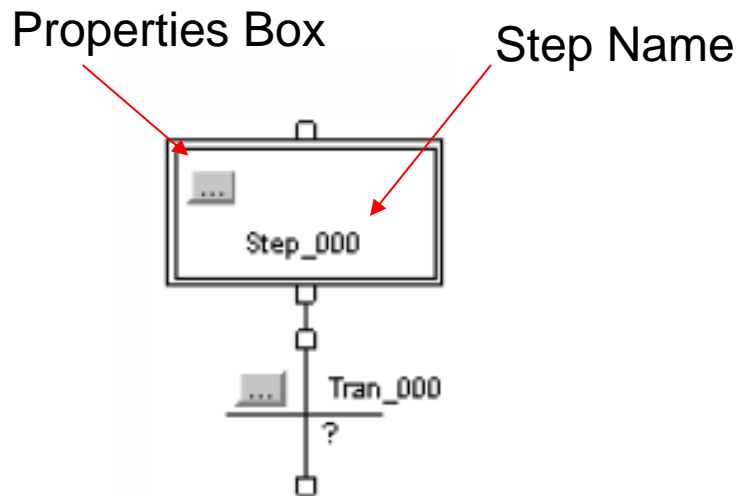
Define the steps of your process.

- Start with large steps and refine in several passes
- Important not to make steps too granular or too high level

Sequential Function Charts (SFC)

Step Structure

Each step uses a tag to provide information about the step.



Step Properties:

- Timer with alarms
- Counter
- Enable (FS, LS & Running)
- Various status bits

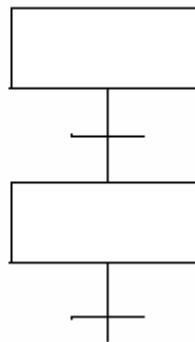
Refer to User Manual for complete description.

Sequential Function Charts (SFC)

Organize Steps

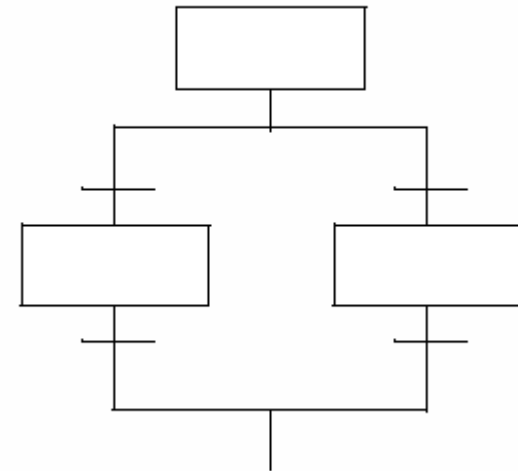
Once the steps are defined organize them into sequences, simultaneous branches, selection branches, or loops.

Sequence



- Execute 1 or more steps in sequence
- When transition is true next step is executed

Selection Branch



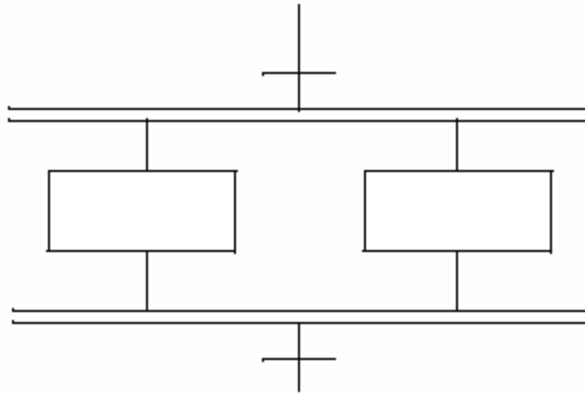
- Choose between two or more steps depending on logic conditions
- SFC checks the transitions from left to right and takes the first true path (can be changed)

Sequential Function Charts (SFC)

Organize Steps

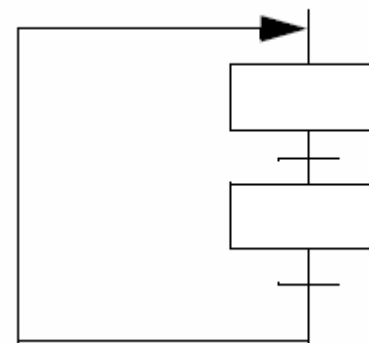
Once the steps are defined organize them into sequences, simultaneous branches, selection branches, or loops.

Simultaneous Branch



- Execute 2 or more steps at the same time
- All paths must finish before continuing the SFC
- A single transition ends the branch

Wire to a Previous Step

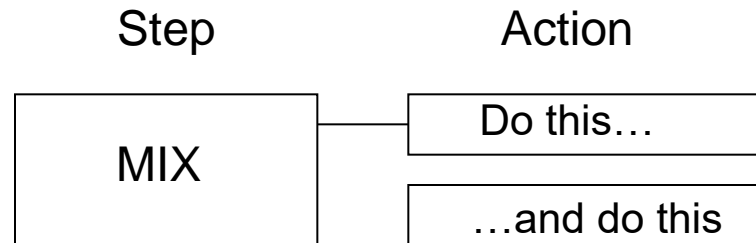


- Loop back to a previous step
- Do not wire into, out of, or between a simultaneous branch

Sequential Function Charts (SFC)

Actions

Actions are used to divide a step into different functions that the step performs such as commanding a motor, setting the state of a valve or placing a group of devices in a specific mode.



Sequential Function Charts (SFC)

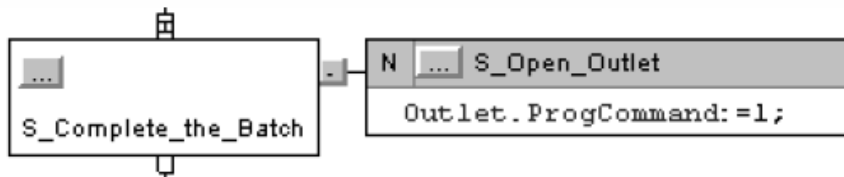
Actions

2 Types of Actions:

1. Non-Boolean
2. Boolean

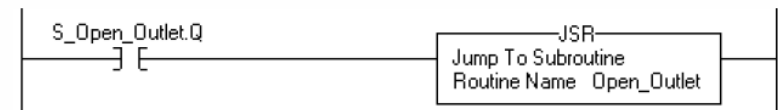
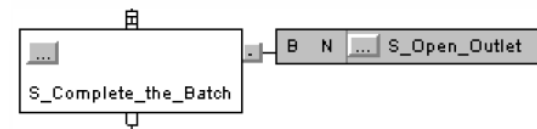
Non-Boolean

- Execute structured text directly in the SFC
- Call a sub-routine
- Use automatic reset option to reset data upon leaving a step



Boolean

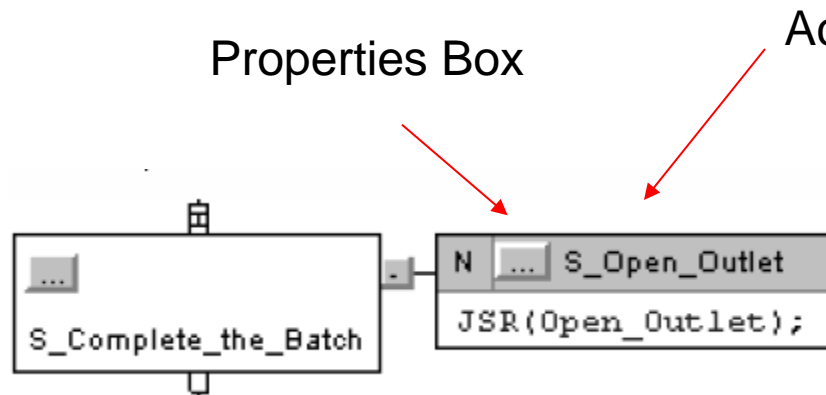
- Only sets a bit
- Other logic monitors bit and executes code



Sequential Function Charts (SFC)

Action Structure

Each action uses a tag to provide information about the action.



Action Properties:

- Timers & Counters
- Enable bits
- Various status bits
- Regular and Time based qualifiers

Sequential Function Charts (SFC)

Action Structure

Choose a Qualifier for an Action

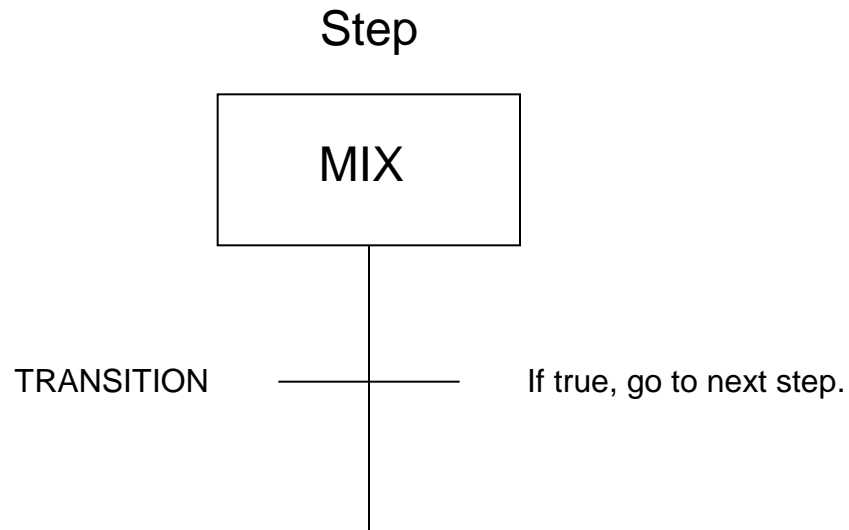
If you want the action to	And	Then assign this qualifier	Which means
Start when the step is activated	Stop when the step is deactivated	N	Non-Stored
	Execute only once	P1	Pulse (Rising Edge)
	Stop before the step is deactivated or when the step is deactivated	L	Time Limited
	Stay active until a Reset action turns off this action	S	Stored
	Stay active until a Reset action turns off this action Or a specific time expires, even if the step is deactivated	SL	Stored and Time Limited
Start a specific time after the step is activated and the step is still active	Stop when the step is deactivated	D	Time Delayed
	Stay active until a Reset action turns off this action	DS	Delayed and Stored
Start a specific time after the step is activated, even if the step is deactivated before this time	Stay active until a Reset action turns off this action	SD	Stored and Time Delayed
Execute once when the step is activated	Execute once when the step is deactivated	P	Pulse
Start when the step is deactivated	Execute only once	P0	Pulse (Falling Edge)
Turn off (reset) a stored action	—————▶	R	Reset

- S Stored
- SL Stored and Time Limited
- DS Delayed and Stored
- SD Stored and Time Delayed

Sequential Function Charts (SFC)

Transitions

The ***Transition*** is the physical conditions that must occur or change in order to go to the next step.



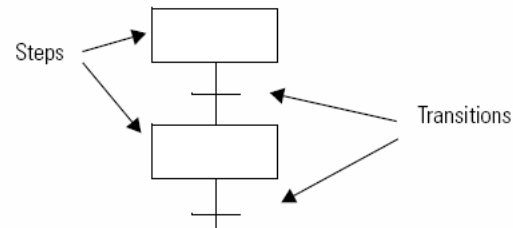
The transition tells the SFC when to go to the next step.

Sequential Function Charts (SFC)

Transitions

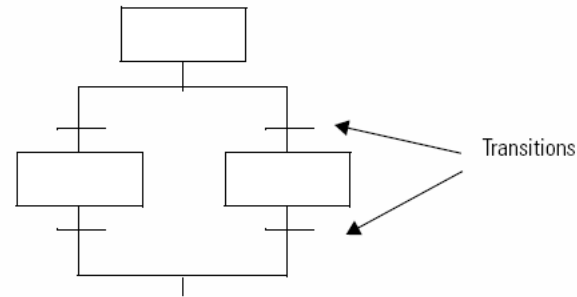
Sequence

A transition is between each step.



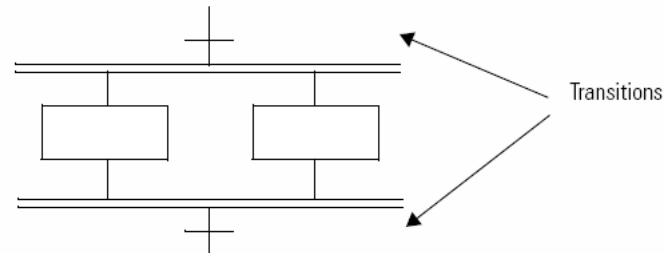
Selection branch

Transitions are inside the horizontal lines.



Simultaneous branch

Transitions are outside the horizontal lines.

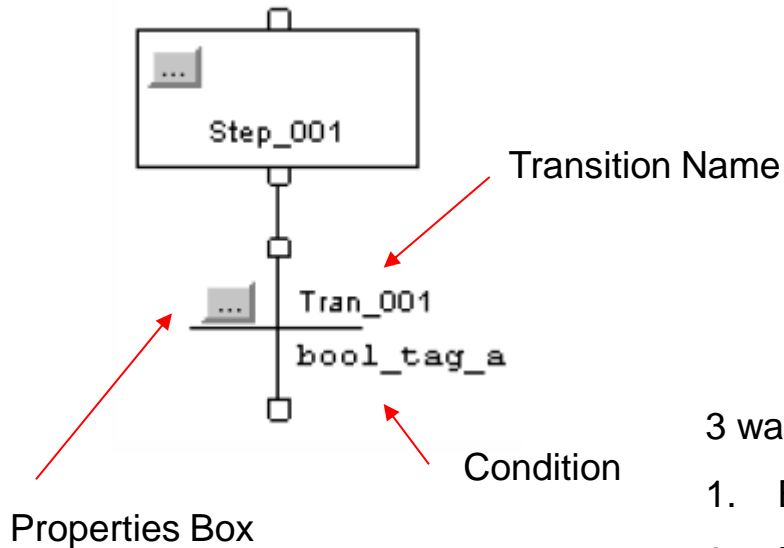


Sequential Function Charts (SFC)

Transitions

Transition Tag

- Each transition uses a BOOL tag to represent the true or false state of the transition.
- If true the SFC goes to the next step.
- If false the SFC continues to execute the current step.



Programming Transitions

3 ways to program a transition:

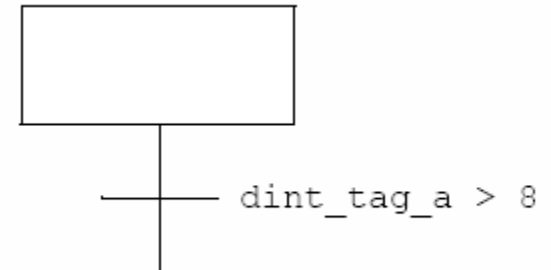
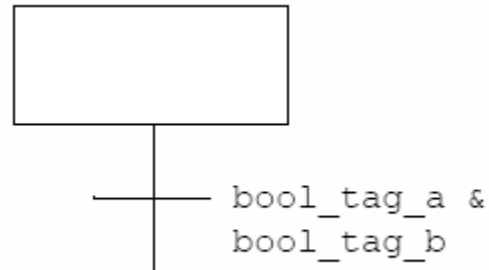
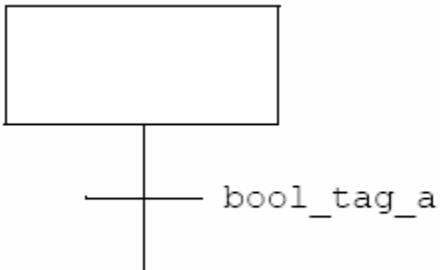
1. Enter condition as Structured Text → Use a BOOL expression
2. Enter condition as instructions from another routine → JSR
3. Use same logic for multiple transitions → JSR

Sequential Function Charts (SFC)

Transitions

Examples

Use a BOOL expression:



Simplest way to transition

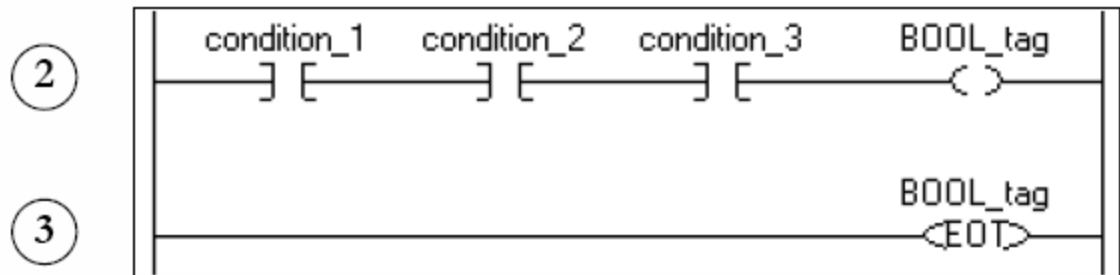
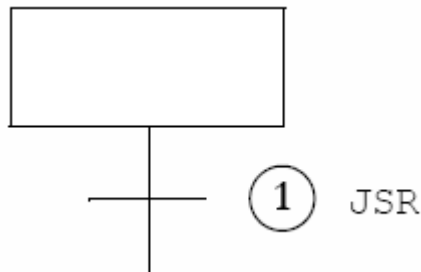
Consult RSLogix5000 Help File for ST syntax.

Sequential Function Charts (SFC)

Transitions

Examples

Call a sub-routine:



1. Call a sub-routine
2. When conditions are true set a BOOL Tag
3. Add EOT instruction to set the state of the transition to the state of the BOOL Tag

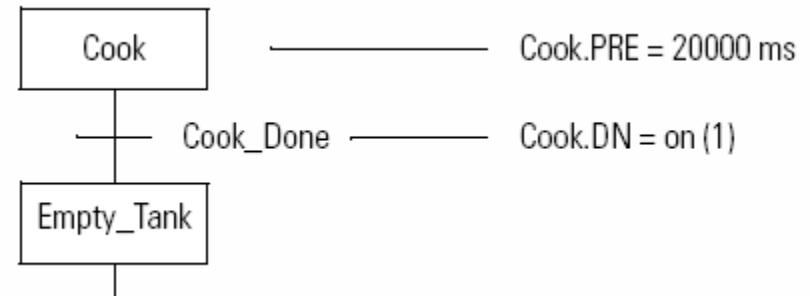
Sequential Function Charts (SFC)

Transitions

Examples

Transitioning after a specified time:

- Enter a preset time in the 'Cook' tag properties
- The transition condition is 'Cook.DN'



Sequential Function Charts (SFC)

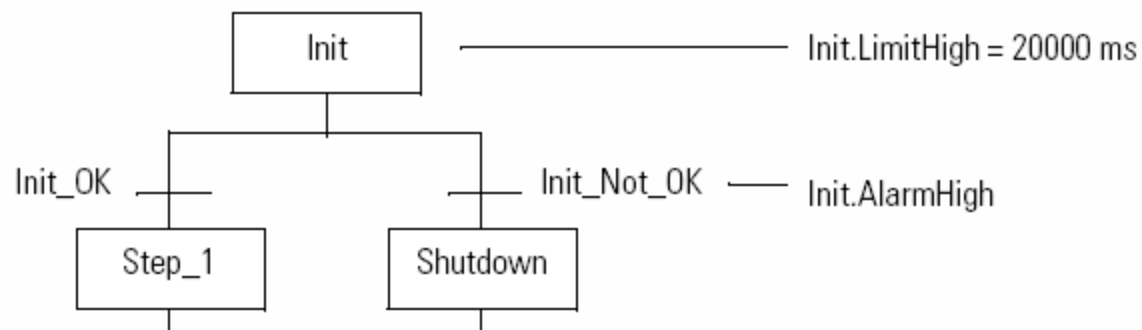
Transitions

Examples

Transitioning after a 'time out' condition:

If a process is taking too long you may wish to run a 'shutdown' step

- Set the 'AlarmEnable' in the Step properties
- Enter a value (ms) in the 'LimitHigh' box.
- Add transition 'Init.AlarmHigh'



Sequential Function Charts (SFC)

Device Control

When the SFC leaves a step there are several options to turn off devices that were turned on during the step.

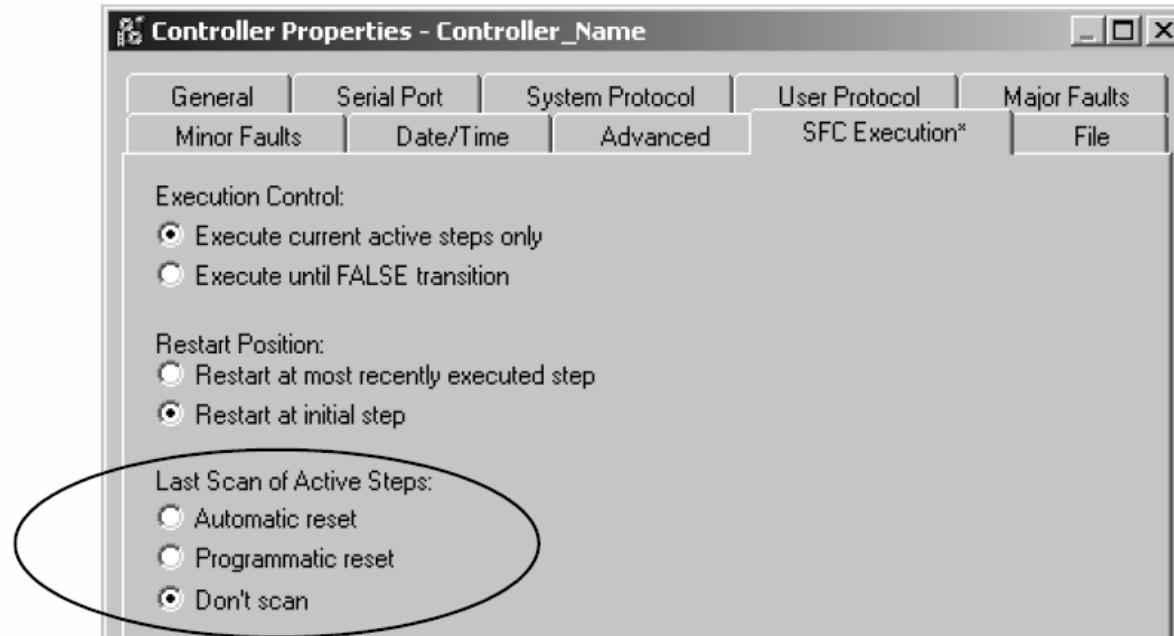
- Programmatic
 - Write logic to clear devices
- Automatic
 - Let the controller automatically clear devices

Each option requires you to make these choices:

- Choose a last scan option
- Based on the last scan option, develop your logic so that the last scan returns data to the desired values

Sequential Function Charts (SFC)

Device Control



Last Scan Options

Sequential Function Charts (SFC)

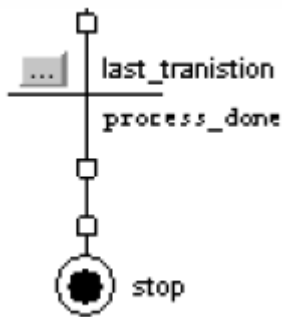
Device Control

Last Scan Options

Characteristic	During the last scan of a step, this option does		
	Don't scan	Programmatic reset	Automatic reset
Execution actions	Only P and PO actions execute. They execute according to their logic.	All actions execute according to their logic.	<ul style="list-style-type: none"> • P and PO actions execute according to their logic. • All other actions execute in Postscan mode. • On the next scan of the routine, the P and PO actions execute in Postscan mode.
Retention of data values	All data keeps its current values.	All data keeps its current values.	<ul style="list-style-type: none"> • Data reverts to its values for postscan. • Tags to the left of [:=] assignments clear to zero.
Method for clearing data	Use P and PO actions.	Use either: <ul style="list-style-type: none"> • Status bits of the step or action to condition logic • P and PO actions 	Use either: <ul style="list-style-type: none"> • [:=] assignment (non-retentive assignment) • instructions that clear their data during postscan
Reset of a nested SFC	A nested SFCs remains at its current step.	A nested SFCs remains at its current step.	For the Restart Position property, if you choose the Restart at initial step option, then: <ul style="list-style-type: none"> • a nested SFC resets to its initial step. • the X bit of a stop element in a nested SFC clears to zero.

Sequential Function Charts (SFC)

Ending (stopping) SFC's



Stop Element

The Stop Element allows you to stop an entire SFC or a path on a simultaneous path from executing and wait for a restart.

- The X bit turns on
- Stored actions remain active

Restart an SFC

Once an SFC is stopped you have several options on how to restart it.

Sequential Function Charts (SFC)

Ending (stopping) SFC's

Restart an SFC

Once an SFC is stopped you have several options on how to restart it.

If the SFC is	And the Last Scan of Active Steps option is	Then
Nested (i.e., another SFC calls this SFC as a subroutine)	Automatic reset	At the end of the step that calls the nested SFC, the nested SFC automatically resets: <ul style="list-style-type: none"> • The nested SFC resets to the initial step. • The X bit of the stop element in the nested SFC clears to zero.
	Programmatic reset	<ol style="list-style-type: none"> 1. Use an SFC Reset (SFR) instruction to restart the SFC at the required step. 2. Use logic to clear the X bit of the stop element.
	Don't scan	
Not nested (i.e., no SFC calls this SFC as a subroutine)	→	<ol style="list-style-type: none"> 1. Use an SFC Reset (SFR) instruction to restart the SFC at the required step. 2. Use logic to clear the X bit of the stop element.

