Elements of SFC

A sequential function chart is a directed graph of *steps* and *transitions* interconnected by *directed links*.

Steps

A step is represented with a rectangle as shown in the following figure and can be defined as follows:

- A step is a set of control actions that are not split in a more detailed sequence of steps at the chosen level of abstraction. A step can be active or inactive. When a step is active, all actions associated to that step are executed.
- For implementation as in IEC61131-3: A *step* represents a situation in which the behavior of a program organization unit with respect to its inputs and outputs follows a set of rules defined by the associated *actions* of the step.

The same sequence can be drawn with 5 or 20 steps; depending on how detailed the sequence should be described. When defining steps, it is important to find a suitable level generalization. On one extreme, there could result an unreadable graph with more than 100 very insignificant steps, on the other extreme there could be 3 very general steps, combining too many action into a step. As a result, important properties are not shown and can therefore not be analyzed. When the SFC should be programmed, the level of generalization is determined by the available action or conversely.

A special step is the initial step. It is drawn with a double line as shown in the next figures. At SFC-initialization, the initial steps are active. Many editors are limited to one initial step.

Transitions

Transitions are represented with a bar and are a symbol for the go-on logic. According to the switching rules, which are explained in the next section, they determine if the following steps will be activated and the preceding steps deactivated. The transition condition is a Boolean function, indicating the process condition for a step change is fulfilled. Its arguments are local variables or external Boolean inputs, e.g. created by a switching device. In a SFC no actions can be associated with transitions.

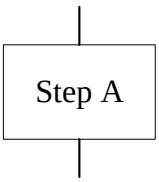

Figure 1: Representation of a step

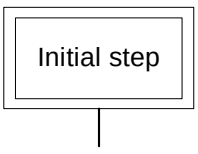

Figure 2: Initial Step

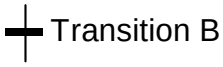

Figure 3: Transition

SFC-drawing rules

Drawing an SFC is very simple and only a few rules have to be followed. If an SFC is drawn according to these rules, the control logic is completely defined and it is possible to use several convenient analysis tolls. SFC-analysis is the topic of the part 2.

Rules:

- Two steps cannot be connecting directly. In-between two steps there is always one transition.
- Two transitions cannot be connected directly. In-between two transitions there is always a step.
- An SFC has at least one initialization step

These rules do not limit the creativity to draw an SFC. With empty steps or ‘true’-transitions any problem can be solved.

All connections between steps and transitions, called arcs, are directed connections. This means that the flow of token is only in one direction. The default direction is top down and is not specially indicated. For other directions like upwards an arrow is drawn to improve readability and minimize misunderstanding.

Simple SFC-Example

A relay controlled motor with temperature control will be the example for the first SFC. As shown in **Fehler! Verweisquelle konnte nicht gefunden werden.** the motor is manually turned on with an ON-button and is turned off either with an OFF-button or through a 'temperature high'-switch. The motor is on, when the coil is under current.

At start, the initial step 'Motor Off', drawn with a double line, is active. The SFC changes to the next step 'Motor ON', when the transition condition is true. In this case the motor temperature must be OK and the ON-button must be pressed. The 'Motor ON'-step is active until the OFF-button is pressed or the motor temperature is too high.

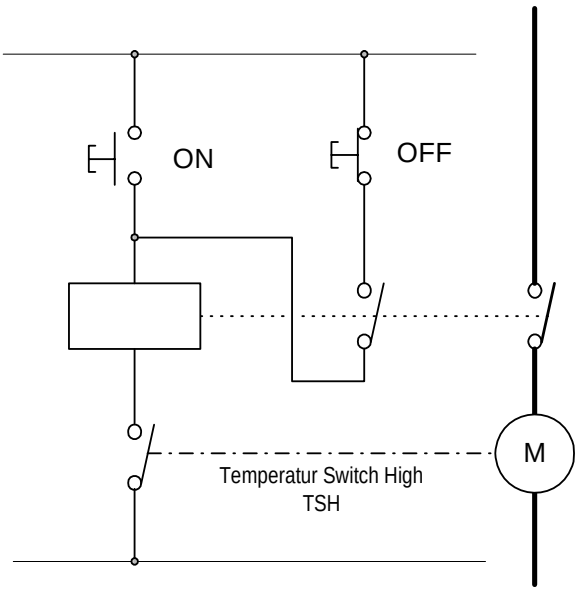

Figure 4: Motor control

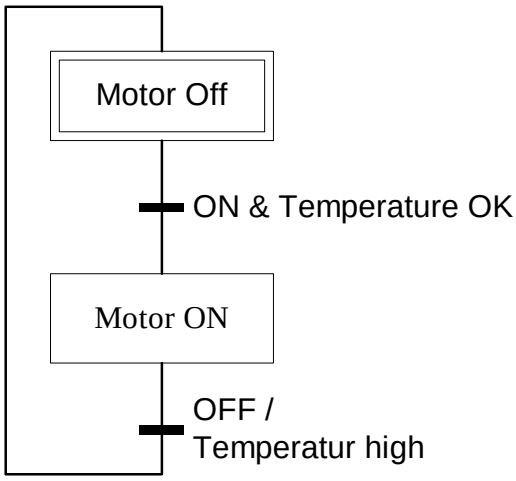

Figure 5: SFC Motor control

SFC-Simulation

SFC-simulation can be visualized, when the active steps are marked with a token. The token can be a black circle or alternatively a specially colored step rectangle. The pattern of token or marked steps is called the marking of the SFC.

Figure 6: marked steps

Actions

The SFC shows the motor control logic, but it is not obvious, what actions are executed in each step. In many cases, it is preferable to explicitly list the actions associated to a step. This can be done in an additional rectangle as shown in the next figure.

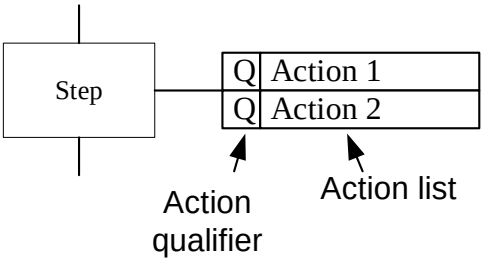

Figure 7: Step with actions

A representation of the Motor-SFC could be:

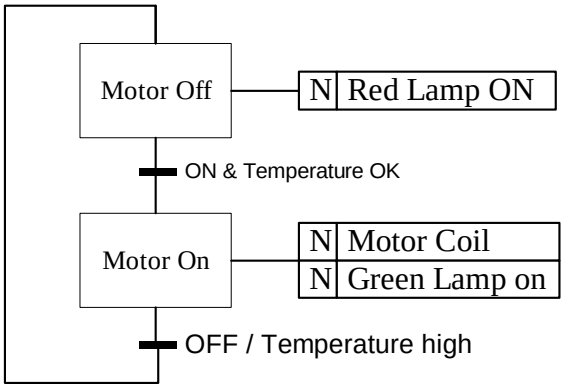

Figure 8: SFC-Representation with steps

In the 'Motor OFF'-step a red lamp is indicating that the motor is off. In step 'Motor ON' the motor relay coil is under current and a green lamp indicates that the motor is on.

When a SFC is implemented, several questions about how to exactly execute the actions arise. For example, there might be a need for actions, that are only executed once when the SFC change to a new step. In order to be able to specify the way an action is executed a set of action qualifiers were defined.

The IEC-61131-3 defines 10 different qualifiers. In the sequel 5 of the most frequently used qualifiers are explained.

Action qualifiers

Qualifier 'N'

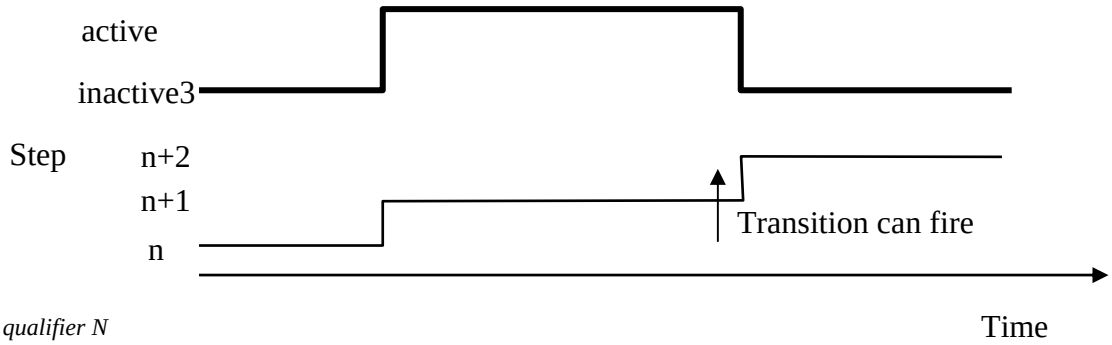

Figure 9: Action qualifier N

An action with qualifier N is executed after a step change and will be executed according to the task settings, e.g. periodically, until the transition condition becomes true and the automata switch to the next step. After the step change, the action is executed once more, so that the programmer has the possibilities to reset whatever is necessary. Otherwise there is no possibility to turn anything off, because an action-program is only executed, when it is in an active step. This last execution of a step is indicated with a 'Last'-flag (usually 'stepname'.Q).

If the 'activate motor relay coil'-action has a qualifier N, the PLC-output has to be set to FALSE, when it is executed the last time.

Qualifier 'P+'

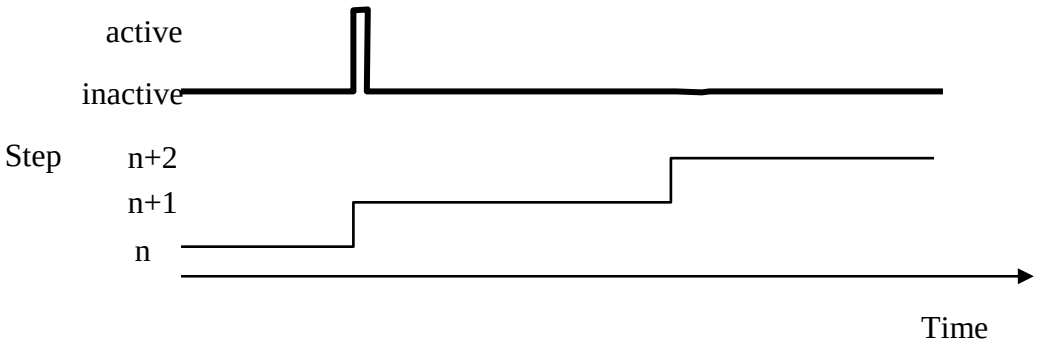

Figure 10: Action qualifier P+

An action in step n+1 with qualifier P+ is executed once at step change from step n to step n+1.

Qualifier 'P-'

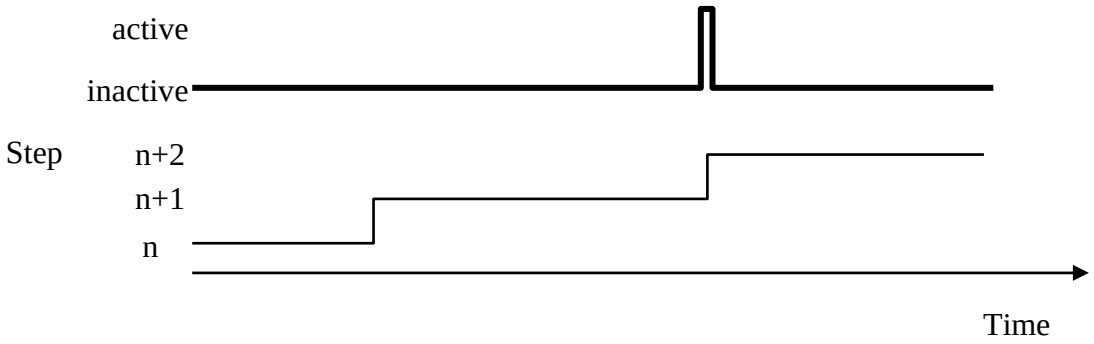

Figure 11: Action qualifier P-

An action in step n+1 with qualifier P- is executed once at step change from step n+1 to step n+2.

Qualifier 'S'

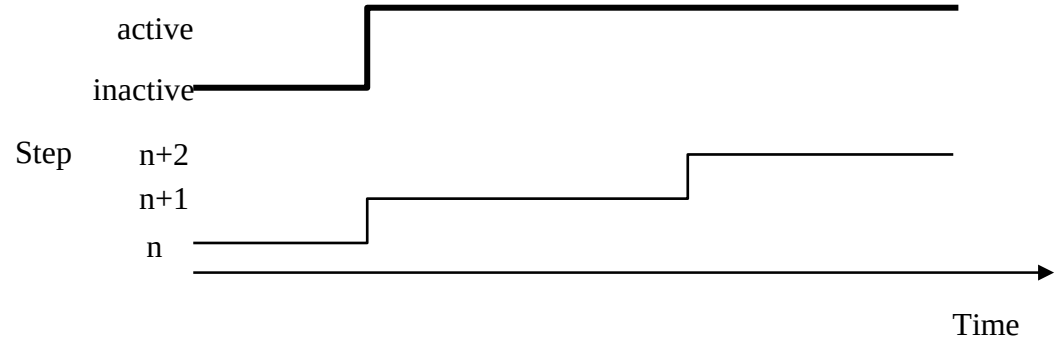

Figure 12: Action qualifier S

An action in step n+1 with qualifier S is executed after a step change from n to n+1 until the action is stopped with a call of the action with qualifier R.

Qualifier 'R'

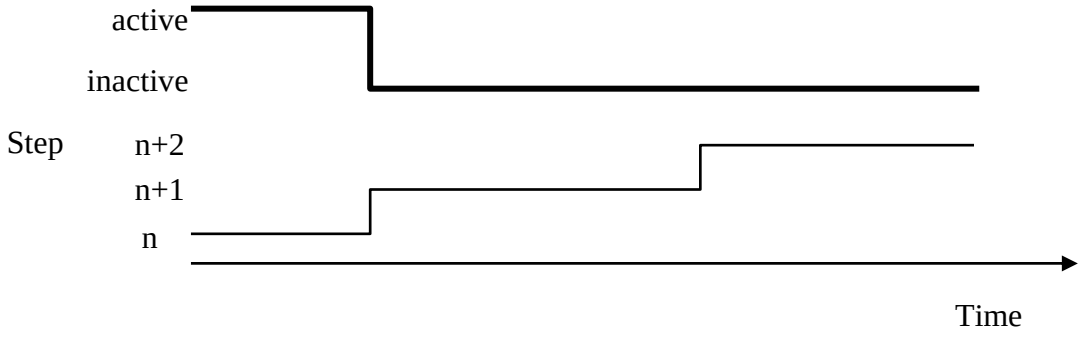

Figure 13: Action qualifier R

Execution of an action is stopped in step n+1 if the action is called with qualifier R.