PLCopen Structuring with SFC do's and don'ts

The PLCopen guideline '*Structuring with SFC – do's and don'ts*' explains the advantages of Sequential Function Chart, SFC. This is a very expressive graphic formalism of the IEC 61131-3 standard. It is not considered a programming language as it needs other languages to express transition conditions and actions.

SFC provides a means for partitioning a programmable controller program organization unit into a set of steps and transitions interconnected by directed links. Associated with each step is a set of actions, and with each transition is associated a transition condition. Since SFC elements require storage of state information, the only POUs which can be structured using these elements are function blocks and programs (not functions).

If any part of a program organization unit is partitioned into SFC elements, the entire program organization unit shall be so partitioned. If no SFC partitioning is given for a program organization unit, the entire program organization unit shall be considered to be a single action which executes under the control of the calling entity.

Whenever a sequential process shall be controlled, SFC shall be considered as most suitable for structuring the internal organization of a POU especially in the Functional Description:

- When the process consists of several steps to be executed sequentially in a time flow, for example an assembly process, SFC can be used to map the different phases of assembly as *steps* and structure the process as sequence of *steps*
- When the process can be modelled as a state machine, these states can be mapped to *steps* and changing from one state to another can be structured by *transitions*.
- SFC structures the internal organization of a program, and helps to decompose a control problem into manageable parts, while maintaining the overview.

This specification is available for downloading on the PLCopen website.

info@PLCopen.org
www.PLCopen.org