Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

**DANGER**

indicates that death or severe personal injury **will** result if proper precautions are not taken.

**WARNING**

indicates that death or severe personal injury **may** result if proper precautions are not taken.

**CAUTION**

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

**CAUTION**

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

**NOTICE**

indicates that an unintended result or situation can occur if the relevant information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

**WARNING**

Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.
Table of contents

1 Introduction.................................................................................................................................................. 7

2 Product overview ....................................................................................................................................... 9
  2.1 Functional scope...................................................................................................................................... 9
  2.2 Differences from a "real" PLC ............................................................................................................ 10
  2.3 Simulation view window ...................................................................................................................... 13
  2.4 Memory areas ........................................................................................................................................ 14
  2.5 Blocks .................................................................................................................................................. 15
  2.5.1 Organization blocks (OBs) ............................................................................................................ 15
  2.5.2 System function blocks (SFBs) ....................................................................................................... 16
  2.5.3 System Functions (SFCs) ............................................................................................................... 17

3 Getting Started ........................................................................................................................................ 19
  3.1 Start simulation ...................................................................................................................................... 19
  3.2 Setting the PG/PC interface ................................................................................................................ 21
  3.3 Multi-instance with S7-PLCSIM ........................................................................................................... 23
  3.3.1 Selecting the type of connection .................................................................................................... 24
  3.3.2 Supported communication blocks .................................................................................................. 24
  3.4 Downloading a STEP 7 project ........................................................................................................... 25
  3.5 Simulation and monitoring ................................................................................................................... 26
  3.6 Monitoring program simulation in STEP 7 ......................................................................................... 27
  3.7 Using Help............................................................................................................................................ 28

4 Simulation Tasks ..................................................................................................................................... 29
  4.1 Attach Symbols ...................................................................................................................................... 29
  4.2 What Is the Difference between a .PLC File and a .LAY File? ............................................................ 30
  4.3 Save a simulated PLC ......................................................................................................................... 31
  4.4 Save Layout Command ....................................................................................................................... 32
  4.5 Open the simulated PLC ..................................................................................................................... 33
  4.6 Open Layout Command ...................................................................................................................... 34
  4.7 Select scan mode .................................................................................................................................. 35
  4.8 Changing the CPU operating mode .................................................................................................... 36
  4.9 Simulating a STEP 7 user program ..................................................................................................... 37
  4.10 Slider Control ..................................................................................................................................... 38
  4.11 Debugging a program ......................................................................................................................... 39
  4.12 Using error OBs in your program ...................................................................................................... 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13</td>
<td>Resetting the CPU memory</td>
<td>41</td>
</tr>
<tr>
<td>4.14</td>
<td>Reset Timers</td>
<td>42</td>
</tr>
<tr>
<td>4.15</td>
<td>Turning power on/off for a simulated CPU</td>
<td>42</td>
</tr>
<tr>
<td>4.16</td>
<td>Using Symbolic Addressing</td>
<td>43</td>
</tr>
<tr>
<td>4.17</td>
<td>Record/Playback</td>
<td>44</td>
</tr>
<tr>
<td>4.18</td>
<td>Monitor cycle time</td>
<td>47</td>
</tr>
<tr>
<td>4.19</td>
<td>Close simulated PLC</td>
<td>48</td>
</tr>
<tr>
<td>4.20</td>
<td>Close Layout</td>
<td>48</td>
</tr>
<tr>
<td>4.21</td>
<td>End a simulation</td>
<td>49</td>
</tr>
<tr>
<td>4.22</td>
<td>Simulating T-CPU</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>View objects</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>CPU View Object</td>
<td>52</td>
</tr>
<tr>
<td>5.1.1</td>
<td>CPU Operating Mode Switch Positions</td>
<td>52</td>
</tr>
<tr>
<td>5.1.2</td>
<td>CPU Indicators</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>ACCUs &amp; Status Word View Object</td>
<td>53</td>
</tr>
<tr>
<td>5.3</td>
<td>Block Regs View Object</td>
<td>54</td>
</tr>
<tr>
<td>5.4</td>
<td>Nesting Stacks View Object</td>
<td>54</td>
</tr>
<tr>
<td>5.5</td>
<td>Input Variable View Object</td>
<td>55</td>
</tr>
<tr>
<td>5.6</td>
<td>Output Variable View Object</td>
<td>56</td>
</tr>
<tr>
<td>5.7</td>
<td>Bit Memory View Object</td>
<td>57</td>
</tr>
<tr>
<td>5.8</td>
<td>Timer View Object</td>
<td>57</td>
</tr>
<tr>
<td>5.9</td>
<td>Counter View Object</td>
<td>58</td>
</tr>
<tr>
<td>5.10</td>
<td>Generic View Object</td>
<td>58</td>
</tr>
<tr>
<td>5.11</td>
<td>Vertical Bits Variable View Object</td>
<td>59</td>
</tr>
<tr>
<td>6</td>
<td>Error and Interrupt OBs</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>Logical base addresses</td>
<td>62</td>
</tr>
<tr>
<td>6.2</td>
<td>Hardware Interrupt (OB40-OB47)</td>
<td>63</td>
</tr>
<tr>
<td>6.3</td>
<td>I/O Redundancy Error (OB70)</td>
<td>64</td>
</tr>
<tr>
<td>6.4</td>
<td>CPU redundancy error (OB72)</td>
<td>65</td>
</tr>
<tr>
<td>6.5</td>
<td>Communication Redundancy Error (OB73)</td>
<td>67</td>
</tr>
<tr>
<td>6.6</td>
<td>Time Error (OB80)</td>
<td>67</td>
</tr>
<tr>
<td>6.7</td>
<td>Diagnostic interrupt (OB82)</td>
<td>68</td>
</tr>
<tr>
<td>6.8</td>
<td>Insert/Remove Module Interrupt (OB83)</td>
<td>69</td>
</tr>
<tr>
<td>6.9</td>
<td>Priority Class Error (OB85)</td>
<td>70</td>
</tr>
<tr>
<td>6.10</td>
<td>Rack Failure (OB86)</td>
<td>71</td>
</tr>
</tbody>
</table>
Table of contents

7 Reference information ................................................................. 73
  7.1 Icons and menu commands ..................................................... 73
  7.2 S7-PLCSIM Numeric Data Formats ......................................... 76
  7.3 Troubleshooting Tips ............................................................... 77
Index .............................................................................................. 79
Introduction

Purpose of this document
The information in this documentation enables you to simulate the operation of an S7 programmable logic controller. You can test control programs without connecting to S7 hardware.

Audience
This documentation is aimed at developers, programmers and maintenance staff with knowledge and experience of S7 programmable logic controllers and STEP 7 programming.

Required background
To understand this documentation, you require a general knowledge of automation engineering. Basic knowledge of the following is also necessary:

- STEP 7 basic software, particularly:
  - Working with the SIMATIC Manager
  - Hardware configuration with HW Config

Validity of the documentation
This manual applies to S7-PLCSIM simulation software, V5.4 SP 5 or higher.

Changes since the previous version
The S7-PLCSIM now has the following new properties:

- Optimization of the download scenario
- Support of an additional PG/PC interface: PCinternal (local)
- Simplified means of access
- Optimization of the communication with WinCC and WinCC flexible
- Display of all CPU access addresses in the status bar
Introduction

Documentation classification
This document describes the functions and the operation of S7-PLCSIM.
For further information, refer to the STEP 7 Online Help and the following manuals:

<table>
<thead>
<tr>
<th>Title</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getting Started</td>
<td></td>
</tr>
<tr>
<td>Getting started and exercises with STEP 7</td>
<td>This manual explains how to use the STEP 7 automation software. This manual provides you with an overview of the procedures used to configure a PLC and to develop control programs.</td>
</tr>
<tr>
<td>Reference Manual</td>
<td></td>
</tr>
<tr>
<td>System Software for S7-300/400 System and Standard Functions</td>
<td>This manual provides you with descriptions of the system functions, organization blocks, and standard functions that you use when developing a control program.</td>
</tr>
<tr>
<td>Manual</td>
<td></td>
</tr>
<tr>
<td>Programming with STEP 7</td>
<td>This manual provides basic information on designing and programming control programs. Use this manual when creating a control program with the STEP 7 automation software.</td>
</tr>
</tbody>
</table>

This and other manuals can be found by selecting the menu command Start > SIMATIC > Documentation in the Windows Start menu on the computer on which STEP 7 is installed.

Guide
The present document describes the handling of the S7-PLCSIM simulation software. It consists of instructive sections and a reference section. The documentation includes the following subject areas:

- Product overviews
- Getting started
- Simulation tasks
- Definition of view objects
- Definition of error and interrupt OBs
- Reference information such as tips on troubleshooting

Service & Support on the Internet
A guide to the technical documentation for the various SIMATIC products and systems is available on the Internet (http://www.automation.siemens.com/simatic/portal/html_76/techdoku.htm).

In addition to our documentation pool, we offer our complete online knowledge base on the Internet (http://www.siemens.com/automation/service&support). Here you can find:

- our newsletter, providing the latest information on your products.
- the correct documents for your product via our Service & Support pages.
- the bulletin board, a worldwide knowledge exchange for users and experts.
- Your local contact for Automation & Drives in our contact database.
- Information about on-site services, repairs, spare parts, and lots more.
Product overview

2.1 Functional scope

Introduction

In S7-PLCSIM you can execute and test your STEP 7 user program in a simulated programmable logic controller (PLC). Simulation is executed on your PC or programming device, such as a Field PG. Since the simulation is implemented completely in the STEP 7 software, you do not require any S7 hardware (CPU or signal modules). You can use S7-PLCSIM to simulate STEP 7 user programs that were developed for S7-300, S7-400 and WinAC controllers.

S7-PLCSIM provides a simple interface to the STEP 7 user program for monitoring and modifying different objects such as input and output variables. You can also use the various applications of the STEP 7 software while you are running your program on the simulated CPU. This allows you, for example, to use such tools as the variable table (VAT) to control and monitor variables. S7-PLCSIM provides a graphical user interface for viewing and modifying control program variables, running the program of the simulated PLC in single or continuous scan mode, and changing the operating mode of the simulated controller.

S7-PLCSIM also includes a COM object called S7ProSim that provides programmatic access to a simulated PLC. With S7ProSim, you can write software to perform such tasks as changing the key switch position of the simulated PLC, running the control program in single scan mode, reading or writing controller values, and many more. The documentation on S7ProSim (http://support.automation.siemens.com/WW/view/en/1139855/0/en) is available in the Internet.

Functionalities

S7-PLCSIM offers the following range of functions:

- Open an existing simulation (Page 33) on startup
- Run programs intended for S7-300, S7-400, T-CPUs (Page 50) and WinAC PLCs on a simulated PLC
- Create view objects (Page 51) that allow you to access the input and output memory areas, accumulators, and registers of the simulated PLC.
- Access memory through symbolic addressing (Page 43).
- Automatically run timers
- Set timers manually or reset all timers or one timer (Page 42)
- Change the CPU operating mode (Page 52) (STOP, RUN and RUN-P)
- Halt the simulation using the Pause menu command without affecting the state of the program
- Test the behavior of your program using error and interrupt OBs (Page 61)
- Record (Page 44) a series of events (modify input and output memory areas, bit memories, timers and counters)
- Play back your program recording in order to automate tests
Integration in STEP 7

You can use all of the STEP 7 tools with the simulated PLC. Although the simulated PLC exists entirely in software, STEP 7 works as if the simulated PLC were a real S7 PLC, with few differences (Page 10).

2.2 Differences from a "real" PLC

Features of the simulated PLC

The simulated PLC provides the following functionalities that are not available in a "real" PLC:

- The "Pause" command (Page 39) halts the simulated CPU and allows you to resume the execution of the program at the instruction where the program was halted.

- When you put the simulated CPU in STOP mode, S7-PLCSIM does not change the status of the outputs. When you select the RUN mode selector (Page 52) position you cannot download a STEP 7 user program or use the STEP 7 tools to change any parameters. A real S7 PLC allows program download and parameter changes when the RUN mode selector is set.

- S7-PLCSIM supports four accumulators (like an S7-400 CPU). In certain special cases, a program in S7-PLCSIM (with four accumulators) can behave differently from the same program running on an S7-300 CPU, (with only two accumulators).

- Any change that you make with a view object immediately updates the contents of the memory location. The simulated CPU does not wait until the beginning or end of the scan in order to update any changed data.

- Scan mode options allow you to select how the CPU runs the program:
  - Single scan (Page 35)
  - Continuous scan (Page 35)

- Timers can be automatically processed or values can be entered manually. You can also reset (Page 42) timers globally or individually.
• You can manually trigger the error and interrupt OBs:
  – OB40 to OB47 (Page 63) (hardware interrupt)
  – OB70 (Page 64) (I/O redundancy error)
  – OB72 (Page 65) (CPU redundancy error)
  – OB73 (Page 67) (communication redundancy error)
  – OB80 (Page 67) (time error)
  – OB82 (Page 68) (diagnostic interrupt)
  – OB83 (Page 69) (insert/remove interrupt)
  – OB85 (Page 70) (program sequence error)
  – OB86 (Page 71) (rack failure).

• Process image and peripheral memory: When you change a value in the process input image, S7-PLCSIM copies it immediately to I/O area of the inputs. This way, when the I/O area input value is written to the process input image at the beginning of the next scan, the desired change is not lost. Correspondingly, when you make a change to an I/O area output value, it is copied immediately to process output image. The following diagram illustrates the order of activities in the scan cycle:

![Scan Cycle Diagram]

When modifying variables in a simulated CPU from a STEP 7 variable table, you must ensure that process image updates do not overwrite or overlay your intended modification. Set the trigger points for modifying variables as follows:
• For inputs, select "Beginning of scan cycle" as the "Trigger Point for Modifying".
• For outputs, select "End of scan cycle" as the "Trigger Point for Modifying".
Further differences

The simulated PLC does not provide the following functionalities that are available in a “real” PLC:

- Diagnostic buffer: S7-PLCSIM does not support all of the error messages written to the diagnostic buffer. For instance, messages relating to bad batteries in the CPU or EPROM errors cannot be simulated. However, most I/O and program errors can be simulated.
- A change of operating mode (Page 52) does not change the I/O to a "safe" state.
- Function modules (FMs) are not supported.
- Point-to-point communication (such as between two S7-400 CPUs in the same rack) is not supported.
- S7-PLCSIM does not support forcing variables.
- S7-PLCSIM executes some SFBs (Page 16) and SFCs (Page 17) in the same way as a real S7 PLC; For others, S7-PLCSIM validates input parameters and returns output that is valid, but not necessarily what a real S7 PLC with a physical I/O would return; and otherwise S7-PLCSIM treats the remainder as NOPs.
- The size of the local data for S7-PLCSIM is defined at 32 kB per priority class, regardless of the CPU variant to be simulated. Since the local data can be configured differently in a real hardware than in S7-PLCSIM, it may happen that the downloading to this hardware is rejected.
- Multicomputing is not supported by S7-PLCSIM: S7-PLCSIM cannot simulate SIMATIC stations with several CPUs (multicomputing).
- H systems are not supported by S7-PLCSIM.
- PROFINET I/O is not supported by PLCSIM

Differences with I/O

Most of the CPUs in the S7-300 family autoconfigure I/O: Once a module has been inserted into a physical controller, it is automatically recognized by the CPU. It is not possible to replicate the autoconfiguration feature with a simulated PLC. If you download a program to S7-PLCSIM from an S7-300 CPU that autoconfigures I/O, the system data does not include an I/O configuration. Therefore, you must first of all download a hardware configuration with configured I/O modules to the system data in order to define which modules the CPU should make available.

To do this, create a project and configure a S7-300 CPU in which the I/O are not automatically configured, e.g. the CPU 315-2DP, CPU 316-2DP or the CPU 318-2. Download this hardware configuration to the S7-PLCSIM. Then you can download the program blocks from any S7 projects. The I/Os are applied error-free.
2.3 Simulation view window

User Interface

The simulation view window of S7-PLCSIM includes the workspace, title bar, status bar, and S7-PLCSIM menus and toolbars (Page 73). The S7-PLCSIM layout is where you display view objects (Page 51).

1. Title bar
2. Menu bar
3. Toolbars
4. Workspace
5. View objects
6. Status bar command
### 2.4 Memory areas

#### Memory areas with different functions

You access data in the S7 PLC by addressing specific areas of memory, which perform specific functions:

<table>
<thead>
<tr>
<th>Memory area</th>
<th>Description</th>
<th>Addressing</th>
<th>S7-PLCSIM limits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timers</td>
<td>Storage for timers</td>
<td>T</td>
<td>T 0 to T 2047</td>
</tr>
<tr>
<td>Counters</td>
<td>Storage for counters</td>
<td>C</td>
<td>C 0 to C 2047</td>
</tr>
<tr>
<td>Bit memory</td>
<td>Storage for data used within the STEP 7 user program</td>
<td>M</td>
<td>131,072 bits (16 Kbytes) of M memory</td>
</tr>
<tr>
<td>Addressable I/O</td>
<td>Direct access to input and output modules</td>
<td>PI: peripheral input</td>
<td>262,136 bits (32 Kbytes) of I/O memory</td>
</tr>
<tr>
<td></td>
<td>Note: The CPU updates the peripheral outputs at the end of every CPU scan cycle.</td>
<td>PQ: peripheral output</td>
<td></td>
</tr>
<tr>
<td>Process image</td>
<td>Process image storage for inputs and outputs</td>
<td>I: Input</td>
<td>Maximum: 131,072 bits (16 Kbytes)</td>
</tr>
<tr>
<td>(configurable; updated every scan)</td>
<td>Note: The CPU updates the inputs at the beginning of every CPU scan cycle</td>
<td>Q: Output</td>
<td>Default setting: 131,072 bits (16 Kbytes)</td>
</tr>
<tr>
<td>Local data</td>
<td>Storage used by logic blocks, including temporary variables</td>
<td>-/-</td>
<td>Maximum: 32 Kbytes</td>
</tr>
<tr>
<td>(configurable)</td>
<td></td>
<td></td>
<td>Default setting: 32 Kbytes</td>
</tr>
<tr>
<td>Data blocks</td>
<td>Memory for data blocks</td>
<td>DB: Data block</td>
<td>Maximum number&gt; 65534 Max. length: 65570</td>
</tr>
</tbody>
</table>
2.5 Blocks

2.5.1 Organization blocks (OBs)

Supported OBs

S7-PLCSIM supports the OBs listed below:

<table>
<thead>
<tr>
<th>OB</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OB1</td>
<td>Free cycle</td>
</tr>
<tr>
<td>OB10 to OB17</td>
<td>Time-of-day interrupt</td>
</tr>
<tr>
<td>OB20 to OB23</td>
<td>Time-delay interrupt</td>
</tr>
<tr>
<td>OB30 to OB38</td>
<td>Cyclic interrupt</td>
</tr>
<tr>
<td>OB40 to OB47</td>
<td>Hardware interrupts</td>
</tr>
<tr>
<td>OB55*</td>
<td>Status interrupt</td>
</tr>
<tr>
<td>OB56*</td>
<td>Update interrupt</td>
</tr>
<tr>
<td>OB57*</td>
<td>Manufacturer-specific interrupt</td>
</tr>
<tr>
<td>OB60*</td>
<td>Multiprocessor interrupt</td>
</tr>
<tr>
<td>OB61* to OB64*</td>
<td>Synchronous cycle interrupt</td>
</tr>
<tr>
<td>OB65*</td>
<td>Technology synchronization interrupt</td>
</tr>
<tr>
<td>OB70</td>
<td>I/O redundancy error</td>
</tr>
<tr>
<td>OB72</td>
<td>CPU redundancy error</td>
</tr>
<tr>
<td>OB73</td>
<td>Communication error</td>
</tr>
<tr>
<td>OB80</td>
<td>Timeout error</td>
</tr>
<tr>
<td>OB81*</td>
<td>Power supply error</td>
</tr>
<tr>
<td>OB82</td>
<td>Diagnostic interrupt</td>
</tr>
<tr>
<td>OB83</td>
<td>Insert/remove module interrupt</td>
</tr>
<tr>
<td>OB84*</td>
<td>CPU hardware fault</td>
</tr>
<tr>
<td>OB85</td>
<td>Priority class error</td>
</tr>
<tr>
<td>OB86</td>
<td>Rack failure</td>
</tr>
<tr>
<td>OB87*</td>
<td>Communication error</td>
</tr>
<tr>
<td>OB88*</td>
<td>Processing interrupt</td>
</tr>
<tr>
<td>OB90*</td>
<td>Background OB</td>
</tr>
<tr>
<td>OB100</td>
<td>Warm restart</td>
</tr>
<tr>
<td>OB101</td>
<td>Hot start</td>
</tr>
<tr>
<td>OB102</td>
<td>Cold restart</td>
</tr>
<tr>
<td>OB 121</td>
<td>Programming error</td>
</tr>
<tr>
<td>OB122</td>
<td>I/O access error</td>
</tr>
</tbody>
</table>

* OBs marked with an asterisk (*) are not called.
### System function blocks (SFBs)

**Supported SFBs**

S7-PLCSIM supports the SFBs listed below:

<table>
<thead>
<tr>
<th>SFB No.</th>
<th>Short name</th>
<th>SFB No.</th>
<th>Short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFB0</td>
<td>CTU</td>
<td>SFB20</td>
<td>STOP</td>
</tr>
<tr>
<td>SFB1</td>
<td>CTD</td>
<td>SFB22</td>
<td>STATUS</td>
</tr>
<tr>
<td>SFB2</td>
<td>CTUD</td>
<td>SFB23</td>
<td>USTATUS</td>
</tr>
<tr>
<td>SFB3</td>
<td>TP</td>
<td>SFB31</td>
<td>NOTIFY_8P</td>
</tr>
<tr>
<td>SFB4</td>
<td>TON</td>
<td>SFB32</td>
<td>DRUM</td>
</tr>
<tr>
<td>SFB5</td>
<td>TOF</td>
<td>SFB33</td>
<td>ALARM</td>
</tr>
<tr>
<td>SFB8</td>
<td>USEND</td>
<td>SFB34</td>
<td>ALARM_8</td>
</tr>
<tr>
<td>SFB9</td>
<td>URCV</td>
<td>SFB35</td>
<td>ALARM_8P</td>
</tr>
<tr>
<td>SFB12</td>
<td>BSEND</td>
<td>SFB36</td>
<td>NOTIFY</td>
</tr>
<tr>
<td>SFB13</td>
<td>BRCV</td>
<td>SFB37</td>
<td>AR_SEND</td>
</tr>
<tr>
<td>SFB14</td>
<td>GET</td>
<td>SFB52</td>
<td>RDREC</td>
</tr>
<tr>
<td>SFB15</td>
<td>PUT</td>
<td>SFB53</td>
<td>WRREC</td>
</tr>
<tr>
<td>SFB19</td>
<td>START</td>
<td>SFB54</td>
<td>RALRM</td>
</tr>
</tbody>
</table>
2.5.3 System Functions (SFCs)

Supported SFCs

S7-PLCSIM supports the SFCs listed below:

<table>
<thead>
<tr>
<th>SFC No.</th>
<th>Short name</th>
<th>SFC No.</th>
<th>Short name</th>
<th>SFC No.</th>
<th>Short name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFC0</td>
<td>SET_CLK</td>
<td>SFC27</td>
<td>UPDAT_PO</td>
<td>SFC54</td>
<td>RD_DPARM</td>
</tr>
<tr>
<td>SFC1</td>
<td>READ_CLK</td>
<td>SFC28</td>
<td>SET_TINT</td>
<td>SFC55</td>
<td>WR_PARM</td>
</tr>
<tr>
<td>SFC2</td>
<td>SET_RTM</td>
<td>SFC29</td>
<td>CAN_TINT</td>
<td>SFC56</td>
<td>WR_DPARM</td>
</tr>
<tr>
<td>SFC3</td>
<td>CTRL_RTM</td>
<td>SFC30</td>
<td>ACT_TINT</td>
<td>SFC57</td>
<td>PARM_MOD</td>
</tr>
<tr>
<td>SFC4</td>
<td>READ_RTM</td>
<td>SFC31</td>
<td>ORY_TINT</td>
<td>SFC58</td>
<td>WR_REC</td>
</tr>
<tr>
<td>SFC5</td>
<td>GADR_LGC</td>
<td>SFC32</td>
<td>SRT_DINT</td>
<td>SFC59</td>
<td>RD_REC</td>
</tr>
<tr>
<td>SFC6</td>
<td>RD_SINFO</td>
<td>SFC33</td>
<td>CAN_DINT</td>
<td>SFC62</td>
<td>CONTROL</td>
</tr>
<tr>
<td>SFC9</td>
<td>EN_MSG</td>
<td>SFC34</td>
<td>ORY_DINT</td>
<td>SFC64</td>
<td>TIME_TCK</td>
</tr>
<tr>
<td>SFC10</td>
<td>DIS_MSG</td>
<td>SFC36</td>
<td>MSK_FLT</td>
<td>SFC78</td>
<td>OB_RT</td>
</tr>
<tr>
<td>SFC11</td>
<td>DPSYC_FR</td>
<td>SFC37</td>
<td>DMSK_FLT</td>
<td>SFC79</td>
<td>SET</td>
</tr>
<tr>
<td>SFC12</td>
<td>D_ACT_DP</td>
<td>SFC38</td>
<td>READ_ERR</td>
<td>SFC80</td>
<td>RSET</td>
</tr>
<tr>
<td>SFC13</td>
<td>DPNRM_DG</td>
<td>SFC39</td>
<td>DIS_IRT</td>
<td>SFC82</td>
<td>CREA_DBL</td>
</tr>
<tr>
<td>SFC14</td>
<td>DPRD_DAT</td>
<td>SFC40</td>
<td>EN_IRT</td>
<td>SFC83</td>
<td>READ_DBL</td>
</tr>
<tr>
<td>SFC15</td>
<td>DPWR_DAT</td>
<td>SFC41</td>
<td>DIS_AIRT</td>
<td>SFC84</td>
<td>WRIT_DBL</td>
</tr>
<tr>
<td>SFC17</td>
<td>ALARM_SQ</td>
<td>SFC42</td>
<td>EN_AIRT</td>
<td>SFC85</td>
<td>CREA_DB</td>
</tr>
<tr>
<td>SFC18</td>
<td>ALARM_S</td>
<td>SFC43</td>
<td>RE_TRIGR</td>
<td>SFC87</td>
<td>C_DIAG</td>
</tr>
<tr>
<td>SFC19</td>
<td>ALARM_SC</td>
<td>SFC44</td>
<td>REPL_VAL</td>
<td>SFC90</td>
<td>H_CTRL</td>
</tr>
<tr>
<td>SFC20</td>
<td>BLKMOV</td>
<td>SFC46</td>
<td>STP</td>
<td>SFC105</td>
<td>READ_SI</td>
</tr>
<tr>
<td>SFC21</td>
<td>FILL</td>
<td>SFC47</td>
<td>WAIT</td>
<td>SFC106</td>
<td>DEL_SI</td>
</tr>
<tr>
<td>SFC22</td>
<td>CREAT_DB</td>
<td>SFC49</td>
<td>LGC_GADR</td>
<td>SFC107</td>
<td>ALARM_DQ</td>
</tr>
<tr>
<td>SFC23</td>
<td>DEL_DB</td>
<td>SFC50</td>
<td>RD_LGADR</td>
<td>SFC108</td>
<td>ALARM_D</td>
</tr>
<tr>
<td>SFC24</td>
<td>TEST_DB</td>
<td>SFC51</td>
<td>RDSYSST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SFC26</td>
<td>UPDAT_PI</td>
<td>SFC52</td>
<td>WR_USMSG</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1 Start simulation

Requirement

- No other simulated PLC is open
- There are no connections to real PLCs

Introduction

The following procedures help you to get started. The simulation can be called from the SIMATIC Manager.

Procedure

To start a simulation, proceed as follows:

1. You can use one of the following methods to start S7-PLCSIM:
   - Open the SIMATIC Manager and click the icon or select the menu command Options > Simulate Modules.
     S7-PLCSIM is opened. The user interface language and the mnemonic settings are the same as the STEP 7 settings.
   - From the Windows Start menu, select the menu command SIMATIC > STEP 7 > S7-PLCSIM Simulating Modules.
     S7-PLCSIM is opened. The user interface language is not the same as the STEP 7 settings. When you first launch S7/PLCSIM, the interface language is English. With subsequent starts, the S7-PLCSIM opens using the language that was used last. This setting is user-specific.
Simulation is started. The view object "CPU" is opened.
The PLC must be in the original state. It has the following properties and standard settings:

- Supports any connection
- Supports any address
- Standard address
- Interface configuration on the basis of the interface last used
- Immediately downloadable

Any new connection is automatically established with the simulated PLC. Any program that you download goes to the simulated PLC. If you click the "Accessible Nodes" button in the SIMATIC Manager, the node address for the simulated PLC is shown.

Note
S7-PLCSIM automatically changes the S7ONLINE Access Point to a simulation subnet. During simulation, do not change the access point to an access point that is unknown to S7-PLCSIM with "Set PG/PC interface". S7-PLCSIM will change the access point back to the original setting when you end the simulation.
3.2 Setting the PG/PC interface

Types of connection

In previous releases of S7-PLCSIM you could only simulate a PLC through an MPI connection. With S7-PLCSIM you can connect through any of the following interface configuration types:

- PLCSIM (ISO)
- PLCSIM (Local)
- PLCSIM (MPI)
- PLCSIM (PROFIBUS)
- PLCSIM (TCP/IP)
- ...

<table>
<thead>
<tr>
<th>Interface configuration</th>
<th>Type of connection</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLCSIM (ISO)</td>
<td>Via the MAC address</td>
</tr>
<tr>
<td>PLCSIM (Local)</td>
<td>Via the virtual backplane bus/Softbus</td>
</tr>
<tr>
<td>PLCSIM (MPI)</td>
<td>Via the MPI interface</td>
</tr>
<tr>
<td>PLCSIM (PROFIBUS)</td>
<td>Via the PROFIBUS interface</td>
</tr>
<tr>
<td>PLCSIM (TCP/IP)</td>
<td>via the IP address</td>
</tr>
<tr>
<td>...</td>
<td>unknown connection class</td>
</tr>
</tbody>
</table>

Note

The connection via the MPI interface is the default setting for the simulated PLC in S7-PLCSIM. Subsequently the simulated PLC will start with the last used connection class.
3.2 Setting the PG/PC interface

Procedure

To set a PG/PC interface, proceed as follows:
1. Configure your hardware configuration in STEP 7.
2. Start the S7-PLCSIM.
3. In the dropdown list of the "Standard" toolbar, select one of the configured connection classes for the virtual PLC.

Result

The PG/PC interface is created.

Note

Changes that are made in the dropdown list of the "Standard" toolbar will effect the function of the menu command Tools > Setting the PG/PC interface in the SIMATIC Manager. Changes will also be effected in the reverse case.

Color meanings in the dropdown entries

- **Black** (example: PLC|SIM(MPI))
  This color means that the CPU supports this PG/PC interface. The CPU is clearly accessible via this interface.

- **Grey** (example: PLC|SIM(MPI))
  This color means that the CPU does not support this PG/PC interface. The CPU is not accessible via this interface. The interface can be selected. The CPU is however not accessible.

- **Black-gray** (example: PLC|SIM(MPI))
  This coloring occurs when you work with several CPUs that have the same address. It indicates that although the CPU supports the PG/PC interface, it is not currently accessible via this interface.
3.3 Multi-instance with S7-PLCSIM

Overview

Several CPUs can be simulated at the same time with the new function.

Note

Adjustments to the "Desktopheap Memory" operating system community may possibly be necessary for the multi-instance mode of S7-PLCSIM, if no additional programs can be started or no additional windows can be opened when a large number of Windows programs are running.

For more information on this operating system behavior, refer to the Microsoft Help and Support Internet page, article ID: KB126962.

Requirement

- S7-PLCSIM is opened with at least one instance
- Configured network address in STEP 7 corresponds with that in S7-PLCSIM or the PLC is in original state

Procedure - Simulating

To simulate several CPUs simultaneously, follow these steps:
1. Start a new instance.
2. Select the menu command Simulation > New PLC.
A new instance of the simulation is started in the original state. The view object "CPU" is opened.

Specific downloading when using several instances

All default CPUs have the same default addresses and these are not taken into account during downloading. When several instances are opened with a default CPU, the STEP 7 project is downloaded to the instance with the lowest number in the title bar (example: S7-PLCSIM2). This is the instance number.

Memory behavior when using several instances

If you have opened several instances, you have to save
- every workplace of every instance
- every simulation of every instance individually to archive them.

Closing multi-instances

If you have opened several instances, you must note the following when you close these:
- To close all instances, click 🗝 in the SIMATIC Manager
- To close the instances individually, select the Simulation > Close menu command.
3.3 Multi-instance with S7-PLCSIM

3.3.1 Selecting the type of connection

Color significance of the entry from the drop-down list

To change the PG/PC interface in S7-PLCSIM, use the drop-down list in the "Standard" toolbar. The entries in the drop-down list have different colorings depending on the number of simulated CPUs and the availability via the set interface. The following coloring occurs only if several CPUs are simulated simultaneously:

- **Black-gray** (example: \texttt{PLCSIM(MP1)})

This coloring occurs when you work with several CPUs that have the same address. It indicates that although CPU "1" supports the PG/PC interface, it is not currently accessible via this interface. The reason is that CPU "2" is selected for the communication under the same address. To select CPU "1" for the communication, you have to select this PG/PC interface again. The coloring of the CPU "1" then changes to "Black-black". The coloring of CPU "2" changes to "Black-gray".

3.3.2 Supported communication blocks

Multiple start capability of S7-PLCSIM V.5.4 with communication support between the CPUs

Several CPUs can be simulated at the same time with the new function. These different CPUs can communicate with each other if an appropriate hardware configuration has been loaded beforehand. This requires unique addresses of the CPUs in the same subnet.

The communication between the CPUs supports the following communication blocks:

- SFB8 "USEND",
- SFB9 "URCV",
- SFB12 "BSEND"
- SFB13 "BRCV"
- SFB15 "PUT"
- SFB14 "GET"
- SFB19 "START"
- SFB 20 "STOP"
- SFB 22 "STATUS"
- SFB 23 "USTATUS"

\textbf{Note}

S7-PLCSIM is not real-time capable. Time response constraints may occur during the communication.
3.4 Downloading a STEP 7 project

Requirement

- Simulation was started from the STEP 7 SIMATIC Manager
- The appropriate connection type is configured
- Address in the STEP 7 corresponds with that in the S7-PLCSIM or the PLC is in original state

Procedure

To download the STEP 7 project, proceed as follows:

1. Navigate to the station in the SIMATIC Manager.
2. Click or select the menu command PLC > Download.

Result

The blocks and hardware configuration are downloaded to the simulated PLC. The simulation system adopts the identity of the loaded CPU and all configured connection data.

The status bar gives an overview of the network addresses configured in the hardware configuration.

The simulation system resets to its original state with the "MRES" function.

Note

Independently configured CPs

It is not possible to simulate independently configured CPs.
3.5 Simulation and monitoring

Procedure

To simulate the application and to monitor and control the application, proceed as follows:

1. Open the SIMATIC Manager
2. Open the STEP 7 example project "ZEn01_09_STEP7__Zebra".
3. Click on the symbol to apply the S7-PLCSIM.
4. Download (Page 25) the example project
5. Create additional "view objects" (Page 51) in S7-PLCSIM.

The data in the simulated PLC can be monitored.

   - Click or select the menu command Insert > Input Variable. The view object displays IB0 (Input Byte 0). Set the data format to "Bits."
   - Click or select the menu command Insert > Output Variable to insert a second view object QB0 (Output Byte 0).
   - Click or select the menu command Insert > Timer three times to insert three "Timer" view objects. Type 2, 3 and 4 (for timers T2, T3 and T4) in their respective text boxes, pressing the Enter key after each entry. (S7-PLCSIM will fill in the symbolic name for each of these three timers.)

6. Choose the menu command PLC > Power On
7. Choose the menu command Execute > Scan Mode > Continuous Scan
8. Select the menu command Execute > Key Switch Position > RUN or RUN-P.
   The simulated CPU is put into RUN mode.
9. Click bit 0 of IB0, to simulate turning on input 0.0, and
10. Monitor the effects on the timers.
11. Click or select the menu command File > Save PLC As to save the current state of the simulated PLC as a new file (Page 31).
3.6 Monitoring program simulation in STEP 7

Requirement

- View objects (Page 51) have been created
- Example project “Zebra” is opened and the station downloaded to the S7-PLCSIM

Procedure

To monitor the simulation of your program in STEP 7, proceed as follows:

1. Click or select the menu command View > Online.
   The Online mode is activated.
2. Navigate to the “Blocks” object in the ZEBRA example project.
3. Open the function FC1.
   The "LAD/STL/FBD" application is called.
4. Put the simulated CPU in RUN mode.
5. Turn on Bit 0 of IB0.
6. Select the menu command Debug > Monitor in the LAD/STL/FBD Editor.
   The effects on your program can be monitored.
3.7 Using Help

Introduction

You can access the S7-PLCSIM online help through the Help menu or in any of the following ways:

- To get help about an object in the S7-PLCSIM window, click the Help button on the toolbar, and then click the object.
- To get help about any dialog or error message, click the Help button in the dialog or message box, or press F1.

The Help window provides the following buttons, menu commands and tabs:

Help Buttons

- **Hide Button / Show Button**: Toggles the display of the navigation area (Table of Contents, Index, and Search tabs). To reduce the overall size of the help window, you can hide the navigation area; When you are ready to view new topics, click the Show button to restore the navigation area.
- **Back Button**: If you have examined more than one topic, this button allows you to move back to the previous topic(s).
- **Forward button**: If you have examined more than one topic, this button allows you to move to the next topic(s).
- **Home page**: Opens the web page that is defined as the home page for the Online Help of S7-PLCSIM.
- **Print button**: Allows you to send a selected topic, or an entire book, to any printer that you have installed.

Help Browser Tabs

- **Contents Tab**: Choose this tab to view the table of contents for the help system. Double-click any book icon to expand it and view the topics that it contains.
- **Index tab**: Choose this tab to view an alphabetical list of index keywords for the help system.
- **Search tab**: Choose this tab and type in a term that you wish to find. Double-click a topic from the list to view the topic. By default, the term is highlighted every place that it appears in the topic to make it easy to locate the term. You can toggle highlighting off or on before displaying a topic. Use the Options button to do this.
4.1 Attach Symbols

Predecessor method***

Up to now the symbols could be attached from the "Select CPU Access Node" dialog. To use the symbols from this STEP 7 project, you have to select the "Attach Symbols" check box.

Procedure

To use the symbols of a project, proceed as follows:

1. Start a simulation.
2. Select the menu command Tools > Options > Attach Symbols or click .
   The "Open" dialog is opened.

3. Navigate to the appropriate project or library entry.
4. Select the symbols.
5. Confirm with "OK".
   The symbols are attached.

See also

Using Symbolic Addressing (Page 43)
4.2 What Is the Difference between a .PLC File and a .LAY File?

PLC file
A file *.PLC is used to save the simulated PLC. The following information is saved:
- Program
- Hardware configuration
- Operating mode
- current status of the I/O

LAY file
A *.LAY file is used to save the current window arrangement of your workspace in S7-PLCSIM. If you arrange your view objects in a certain order and want to preserve that order for future work sessions, save the layout before you close S7-PLCSIM.

Sequence for opening
When you work in S7-PLCSIM, you can open both a .PLC file and a .LAY file.
1. Open the simulated PLC (.PLC file).
2. Then open the layout (.LAY file).
4.3 Save a simulated PLC

Saved data

The following data are saved when you save the PLC:

- Program
- Hardware configuration
- Simulated subnet and nodes
- The check box selection for the operating mode (key switch position) of the CPU: namely, RUN-P, RUN, or STOP
- Scan mode option (continuous scan, single scan)
- The status of the I/O
- Timer values (T memory)
- Symbolic addresses
- Power on/off setting

Procedure

To save the current state of the simulated PLC under the current file name, proceed as follows:

1. Select the menu command **File > Save PLC**.
   
   Use the menu command **File > Save PLC As** to archive the configuration of the PLC to a new file.

2. To display symbolic addresses, use the **Tools > Options > Show Symbols** command.

Result

S7-PLCSIM saves your files in the associated project directory of STEP 7.

If no project data is available for the simulation (e.g.: default CPU), the PLC file is saved under the following path: [Installation directory]\Siemens\PLCSIM\S7WSI\Archive
4.4 Save Layout Command

Introduction

A layout is simply an arrangement of view objects (Page 51). The .LAY file archives only the position and selected data format of the view objects in your simulation. The data values that are displayed in the view objects are not saved as part of the layout.

Procedure

To save the current position of the view objects in S7-PLCSIM, proceed as follows:

1. Select the menu command File > Save Layout As.
   
   Select the menu command File > Save Layout to save the layout in the current file.
   
   The "Save Layout As" dialog is opened.

2. Confirm with "Save".

Result

S7-PLCSIM saves your files in the associated project directory of STEP 7.

If no project data is available for the simulation (e.g.: default CPU), the PLC file is saved under the following path: [Installation directory]\Siemens\PLCSIM\S7WSI\Archive
4.5 Open the simulated PLC

Requirement

- The simulated PLC was previously saved in a PLC file.
- The file is not read-only.
- The file is not opened in another application.

Procedure

To open an existing simulation of a PLC, proceed as follows:

1. Select one of the following options:
   - Select the Simulation > Recent Simulation menu command and choose the entry of a saved PLC.
   - Select the File > Open PLC menu command and then browse to and select an existing .PLC file.
2. Confirm with "OK".

Note

If you see a dialog stating that the file is read-only and cannot be opened, you must use Windows Explorer to remove the read-only designation for the file.

Files from older versions

To open a file from S7-PLCSIM V5.3 or earlier, S7-PLCSIM must convert the file to the current file format. During the conversion, a back-up copy of the original file is created with the extension *.BAK. The original PLC file is then converted to the current format. The conversion will fail if a read-only backup file already exists. If so, use Windows Explorer to delete the pre-existing backup file and then open the .PLC file in S7-PLCSIM.

In some cases, S7-PLCSIM will not be able to open an old .PLC file. Simulation files that included multiple DP networks, hot backup systems, or fault tolerant CPUs are potentially incompatible with S7-PLCSIM V5.4.

Note

When you open a new or archived PLC simulation, any view objects that were displayed in S7-PLCSIM are automatically closed. If you intend to open an archived layout as well as a new or archived PLC simulation, open the simulated PLC before opening the layout.

Operating mode after opening

When opening an archived simulated PLC, the saved operating mode is restored.
4.6 Open Layout Command

Introduction

A layout is an arrangement of view objects. The *.LAY file archives only the position and selected data format of the view objects in your simulation. The data values that are displayed in the view objects are not saved as part of the layout.

Requirement

- The layout was previously saved in a LAY file.

Procedure

To open a layout, proceed as follows:

1. Select one of the following options:
   - Select the menu command Simulation > Recent Layout and select a layout.
   - Select the Simulation > Open Layout menu command and then browse to and select an existing *.LAY file.
2. Select the Simulation > Open Layout menu command and the "Open" dialog opens.
3. Navigate to the storage location of the file.
4. Select the file.
5. Confirm with "Open".
   The layout is opened.
4.7 Select scan mode

Introduction

There are 2 options for running the simulated program:

- **Single scan**: The CPU executes one scan and then waits for you to initiate another scan. Each scan consists of the CPU reading the peripheral inputs (PI), executing the program, and then writing the results to the peripheral outputs (PQ). The CPU then waits for the command to run the next scan. Executing a program one scan at a time allows you to see the changes in each scan. As a "real" CPU can execute faster than the editor can display data, the Single Scan option allows you to "freeze" the state of the program from scan to scan.

- **Continuous Scan**: The CPU executes one complete scan and then starts another scan. Each scan consists of the CPU reading the peripheral inputs (PI), executing the program, and then writing the results to the peripheral outputs (PQ).

The default setting is Continuous Scan.

Procedure

To set the "Single Scan" mode, proceed as follows:

1. Click \[ \text{ } \] or select the menu command **Execute > Scan Mode > Single Scan**.
2. To run the next scan, select the menu command **Execute > Next Scan** or \[ \text{ } \].

To set the Continuous Scan Mode, proceed as follows:

1. Click \[ \text{ } \] or select the menu command **Execute > Scan Mode > Continuous Scan**.
4.8 Changing the CPU operating mode

Introduction
The simulated PLC responds to changes in the operating mode like a "real" PLC. The LEDs in the "CPU" view object of the simulated PLC show the current operating mode.

Procedure
To change the CPU operating mode, proceed as follows:

1. Click the appropriate check box or select the menu command **Execute > Key Switch Position > [mode]**.
   The CPU changes the operating mode.

Comparison with a real CPU
The CPU operating modes on the simulated CPU function like the key switch on a real CPU: if you use the STEP 7 tools to change the operating mode, or if the CPU automatically changes mode (for example, encounters an error condition that causes the CPU to change from RUN to STOP), the check boxes on the simulated CPU view object do not change. The LED indicator changes, but not the key switch. This alerts you that the CPU changed operating mode, possibly because of some error in the program.
4.9 Simulating a STEP 7 user program

View object

You can display different types of view objects that allow you to monitor and modify the STEP 7 user program running in the simulated PLC.

S7-PLCSIM provides view objects that you use to monitor and modify program objects. You can also use symbolic addressing to address these view objects. The following view objects are activated from the Insert menu:

- Input Variable (Page 55)
- Output Variable (Page 56)
- Bit Memory (Page 57)
- Timer (Page 57)
- Counter (Page 58)
- Generic (Page 58)
- Vertical Bits (Page 59)

The following three view objects are activated from the View menu:

- Accumulators (Page 53)
- Block Registers (Page 54)
- Stacks (Page 54)

Procedure

To also simultaneously monitor the program in the STEP 7 "LAD/STL/FBD" application, proceed as follows:

1. In the SIMATIC Manager, click or select View > Online to switch to online mode.
2. Navigate to the "Blocks" object of the STEP 7 project and open a logic block.
   (Example: In the S7_ZEBRA example project open FC1).
   The "LAD/STL/FBD" application displays the program that the simulated PLC is executing. Use the menu commands to view the status of the instructions.
### Introduction

The Slider Control is a tool which is available to different numeric formats. Values are not entered using the numeric keypad, but using the slider control.

<table>
<thead>
<tr>
<th>Without Slider Control</th>
<th>With Slider Control</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Slider Control Image" /></td>
<td><img src="image2" alt="Slider Control Image" /></td>
</tr>
</tbody>
</table>

### Simulating Values

The slider control allows you to simulate values that change gradually or have a specific range, such as analog values.

### Procedure

To work with slider control, proceed as follows:

1. Select one of the following view objects for which slider control is available.
   - Input Variable
   - Output Variable
   - Bit Memory
2. You can access the memory area with either a memory address or a symbolic address.
3. To specify a slider control for one of the view objects, you select a slider control representation from the "Select Numeric Format" drop-down list.
4. Select whether to represent the values as decimal (positive integers), integer (positive and negative integers), or real numbers.
   - Byte (B): Decimal
   - Word (W): Decimal and integer
   - Double word (D): Decimal, integer, and real
5. You can use either the mouse or the arrow keys to change the position of the slider control indicator or enter a specific value in the "Value" field.
   - The value of the variable stored in the memory location changes.
Configuring a minimum and maximum value

To select a minimum value for the slider, select "Min" from the "Display Value, Min, or Max" drop-down list of the view object. Then, enter the numeric value for the minimum in the Min field.

To select a maximum value for the slider, select "Max" from the "Display Value, Min, or Max" drop-down list of the view object. Then, enter the numeric value for the maximum in the Max field.

Benefits of a range of values

Selecting a range of values provides the following benefits:

- A range of values does not affect the values that can be stored in the variable. The minimum and maximum values affect only the values that can be entered or displayed by the slider control.
- You can simulate a specific range of values, for example, the range of values that would be generated by a specific analog module.
- By limiting the range of values between the minimum and maximum, you can provide better resolution for entering data with the slider control. You can always enter an exact value in the Value field of the view object.

4.11 Debugging a program

Features for testing

S7-PLCSIM provides the following features to help you debug your program:

- The "Pause" command interrupts the simulated CPU immediately and allows you to resume the execution of the program at the instruction where the program was halted.
- Any change that you make with a view object (Page 51) immediately updates the contents of the memory location. The CPU does not wait until the beginning or end of the scan to update any changed data.
- **Scan mode** (Page 35) options allow you to select how the CPU runs the program:
  - Single scan: executes the program for one scan and then waits for you to initiate the next scan.
  - Continuous scan: executes the program like a "real" PLC: It starts a new scan immediately after the previous one finishes.
4.12 Using error OBs in your program

Introduction

You can use S7-PLCSIM to test how your program handles different interrupt OBs.

Requirement

To be able to simulate error OBs, you must have downloaded an appropriate hardware configuration in the S7-PLCSIM.

Procedure

To trigger error OBs, proceed as follows:

1. Choose the menu command **Execute > Trigger Error OB**
2. Select a specific OB.
   The dialog for the OB is opened.
3. Select the corresponding options.
4. Confirm with "OK" or "Trigger".

Result

The simulated PLC generates the appropriate event and runs the program in the associated OB.

Note

The OBs that are available from the "Trigger Error OB" menu depend on the hardware configuration loaded in the simulation.

Supported OBs

S7-PLCSIM supports the following error and interrupt OBs:

- OB40 to OB47 (Page 63) (hardware interrupt)
- OB70 (Page 64) (I/O redundancy error) {417-H systems only}
- OB72 (Page 65) (CPU redundancy error) {417-H systems only}
- OB73 (Page 67) (communication redundancy error) {417-H systems only}
- OB80 (Page 67) (time error)
- OB82 (Page 68) (diagnostic interrupt)
- OB83 (Page 69) (insert/remove module)
- OB85 (Page 70) (priority class error)
- OB86 (Page 71) (rack failure)
4.13 Resetting the CPU memory

Introduction

Resetting has the following effect:

- the memory areas are reset
- the program blocks are deleted
- the hardware configuration of the simulated PLC is deleted

Procedure

To reset the memory of the simulated CPU, proceed as follows:

1. Select the menu command PLC > Clear/Reset, or click the MRES button on the CPU view object.
   The "Clear/Reset" dialog is opened.
2. Confirm with "Yes".
   The CPU automatically goes to STOP mode and all existing connections are disconnected.
4.14 Reset Timers

**Procedure**

To reset the timers in your program, proceed as follows:

1. Choose the menu command **Execute > Reset Timers**
   
   The "Reset Timers" dialog is opened.

2. Select one of the following options:
   
   - Select "All Timers" to reset all of the timers in the program
   - Select "Specific Timer" to specify a specific timer to reset.

3. Confirm with "OK".

   The timers are reset

**Alternative procedure**

The CPU Mode toolbar also provides a button for resetting timers.

- To reset all of the timers in your program, use the Reset Timers toolbar button.
- To reset an individual timer, use the Reset Timer button on the view object for that timer.

**Manual Timers Command (Execute Menu)**

Select the menu command **Execute > Manual Timers** to set up the CPU to allow you to enter a specific value or to reset the timers in the program. Select the menu command **Execute > Automatic Timers** to return to running the timers automatically

---

4.15 Turning power on/off for a simulated CPU

**Procedure**

To turn power on/off for a simulated CPU, proceed as follows:

1. Select the **PLC > Power On** and **Power Off** menu commands to simulate turning power on and off to the CPU.
4.16 Using Symbolic Addressing

Introduction

By default, the S7-PLCSIM uses the symbols of the loaded STEP 7 program. If you did not attach symbols when you started the simulation or you want to use symbols from another STEP 7 program, you can specify a symbol table to attach.

Procedure

To use symbolic addressing in your simulated program, proceed as follows:

1. Select the menu command **Tools > Options > Attach Symbols**.
   The "Open" dialog is opened.
2. Browse to the storage location of the STEP 7 symbol table to be referenced.
3. Confirm with "OK".
4. Create view objects for variables that you want to address symbolically.
5. To turn on symbols for all view objects, select the menu command **Tools > Options > Show Symbols**.
   To hide the symbols, select the command again.

Tooltips for symbols

When you use symbolic addressing to monitor your program, tooltips are available for all view object fields that have symbolic addresses assigned to them. Point to a field with the mouse to see its symbolic address and comment (separated by a colon) in a tooltip box.

See also

Vertical Bits Variable View Object (Page 59)
4.17 Record/Playback

Introduction

The Record/Playback dialog box allows you to record or play back a series of data changes.

![Record/Playback Dialogue Box]

Requirement

- CPU is in RUN or RUN-P mode

Procedure

To call the "Record/Playback" dialog, proceed as follows:

1. Select the menu command Tools > Record/Playback or click the Record/Playback symbol.

   The dialog is opened.

2. To record a series of events, click the "Record" button.

3. To finish recording, remember to save it by using the Save Event File button before you close S7-PLCSIM.

Note

The key sequence Alt + F5 toggles the display of the Record/Playback toolbar button.
How to Record or Play Back an Event File

- Click the New Event File button to create a new event file.
- Click the Open Event File button to locate and open an existing event file.
- Click the Save Event File button to save the events that you have just recorded.
- Click the Play button to play back an existing recording of events.
- Click the Record button to begin recording a series of events. Use the view objects in your simulation to turn bits on and off or assign data values as desired. The recorder captures every change you make to memory areas.
- Click the Pause button to temporarily suspend recording or playback. The Pause function is convenient because it allows you to pause the recording of events and resume later. If you need to perform some other activity (for instance, add new view objects or answer the telephone) before you are finished generating events, you can click Pause and thus avoid a long delay in your recording. Pause allows you to minimize the time lag between events as you record, in contrast to the Delta button, which affects the overall rate at which the recording is played back.
- Click the Stop button to stop recording or playing back events.
- The Delta button allows you to select a rate of speed before you play back a recording. Your selection affects the overall playback duration. However, if some events were recorded closer together, or further apart than others, the relative time intervals are preserved even as the overall playback time is reduced or increased by your Delta selection.

Check

There are two ways for you to confirm that you are successfully recording or playing back events:

- Check the status bar of the Record/Playback dialog to see whether it is in Recording, Playing, or Idle mode.
- Watch the title bar of the Record/Playback dialog. It should display a numeric value that increments each time you record or play back an event.
## Troubleshooting Tips

<table>
<thead>
<tr>
<th>Problem</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Play button is de-activated and I cannot play back a recording.</td>
<td>You must have an open event file before you can play back the recording of the events. Use the Open Event File button to select and open an event file.</td>
</tr>
<tr>
<td>I recorded a series of events and then closed S7-PLCSIM. The next time I opened S7-PLCSIM, I could not find the events.</td>
<td>If you close S7-PLCSIM without saving the recorded events in an event file, your work is lost. Use the Save Event File button to save your work before you close S7-PLCSIM.</td>
</tr>
<tr>
<td>I recorded a series of events, but when I tried to play them back, nothing happened.</td>
<td>Check the status bar of the Record/Playback dialog to see what mode it is in. If it says Playing, watch the title bar to see when a numeric value appears. When events are played back, a counter in the title bar keeps track of how many have been played back. Note that if you start a recording but do not promptly begin to trigger events, the recorder captures the time lag. When you play back the recording, the first event will take an equally long time to occur. You can examine the event file to verify that in fact, your events have been properly recorded. You can adjust the playback speed of the recording by using the Delta button.</td>
</tr>
<tr>
<td>I cannot remember which event file contains the sequence of events that I want to play back.</td>
<td>You can use long, descriptive file names to help differentiate your event files. If necessary, you can use a text editor to examine your files and locate the one with the correct sequence. The default storage location for event files is [Program Folder]\Siemens\PLCSIM\S7wsi\events.</td>
</tr>
<tr>
<td>I changed a single bit, but when I played back my recording, the entire byte changed.</td>
<td>If an Input Variable, Output Variable, Bit Memory, Generic Variable, or Vertical Bits View Object shows only one bit (e.g., Q0.0, Bits), a bit change is recorded correctly as only a change in that particular bit. However, if the view object displays all eight bits (e.g., QB0, Bits), a change to a single bit is recorded as a change in byte value instead of a change in that bit only. Consequently, it is possible that during playback of the recording, other bits within the byte could be represented as changing (for instance, process flags or Boolean inputs), when in fact they would not be affected during operation of a real PLC.</td>
</tr>
<tr>
<td>When trying to record events in Single Scan mode, the recorder does not function as expected.</td>
<td>When recording events with the CPU view object in Single Scan mode, note the following: You cannot start a recording in Single Scan mode unless you click the &quot;Next Scan&quot; button to increase the scan count. The first event in your recording must have a Delta value of 1 or higher. The Delta value is based on the number of scan cycles that have occurred since the previous event (which in this case was when you clicked the Record button). However, if you are beginning the recording in Single Scan mode, no scan cycles have elapsed yet. You must increment the scan. When you record events in Single Scan mode, they have a Delta value of zero (because they are all occurring within the same scan). Therefore, when you play back the recording, all the events that you recorded during a single scan are displayed in such rapid sequence that they appear to occur simultaneously. To provide a discernable time lag between events, you would have to click the &quot;Next Scan&quot; button or switch between Continuous Scan mode and Single Scan mode for each event.</td>
</tr>
<tr>
<td>My event file contains German mnemonics even though that is not my selection in STEP 7.</td>
<td>Events are recorded with German mnemonics in S7-PLCSIM regardless of your STEP 7 selection. You can disregard this phenomenon.</td>
</tr>
</tbody>
</table>
4.18 Monitor cycle time

Introduction

Program execution may be significantly slower in S7-PLCSIM than with an actual CPU (especially when other applications are running at higher priority). You may experience annoying timeouts because of this. This dialog makes it possible for you to disable or extend the scan cycle monitoring without modifying the program for the target PLC.

Procedure

To monitor the scan cycle, proceed as follows:

1. Choose the menu command **Execute > Scan Cycle Monitoring**
   - The "Scan Cycle Monitoring" dialog is opened.
2. Activate the option "Enable Scan Cycle Monitoring".
3. Enter a monitoring time to any value between 1 second (1000 ms) and 1 minute (60000 ms), inclusive.
   - The default scan cycle monitoring time is 6000 ms.
4. Confirm with "OK"

Definition - Maximum scan cycle time

The maximum scan cycle time is the maximum time the process is allowed to take for one full scan cycle of the S7 user program in OB1 and for the update of the relevant I/O. If this time is exceeded, the simulated PLC goes into STOP mode.

Note

Note that the "Scan Cycle Monitoring" dialog does not reflect the monitoring time set in the hardware configuration. Changes only affect the simulation.
4.19 Close simulated PLC

Introduction
When a simulated PLC is closed, a new CPU is automatically generated in the original state.

Requirement
- The simulation was saved (Page 31).

Procedure
To close the simulation of a program, proceed as follows:
1. Select the menu command File > Close PLC.

Result
The simulated subnet, nodes, and all opened view objects are closed. A new PLC in original state is automatically opened.

Note
Closing a simulated program can result in errors in applications which are currently connected to the simulator.

4.20 Close Layout

Introduction
Closing the layout does not end the simulation session. The current PLC is still open. S7-PLCSIM is still active. You can open another layout.

Requirement
- The layout was saved (Page 32).

Procedure
To close the layout of a program, proceed as follows:
1. Select the menu command File > Close Layout.
   All view objects except "CPU" are closed.
4.21 End a simulation

Introduction

Ending the simulation ends the simulation session. S7-PLCSIM is closed.

Requirement

- Save a simulated PLC (Page 31)
- Save Layout Command (Page 32)

Procedure

To end a simulation, proceed as follows:

1. Close any STEP 7 applications involved in the monitoring of the simulation.
2. Select the menu command File > Exit.
   
   The simulated subnet, nodes, and all opened view objects are closed.

Note

Exiting S7-PLCSIM, like closing a simulated PLC, can result in errors in applications that are currently connected to the simulation.
4.22 Simulating T-CPU

Introduction

S7-PLCSIM can simulate control programs, which, for example, have been developed for a CPU S7-317T, with limitations.

Special features

The simulation does not access any motion control devices. Calls to motion control function blocks simply return to the calling block with some limited error checking.

Error checking includes:

- Existence of instance DB
- Existence of technology DB
- Range checking of parameters with defined ranges

S7-PLCSIM sets, for some of the MC commands, parameters, provided the input parameters are valid (example: CPU S7-317T):

<table>
<thead>
<tr>
<th>MC Command</th>
<th>Parameter</th>
<th>Value(s) set</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC_Power</td>
<td>Statusword.DriveEnabled</td>
<td>True for enabled, false for disabled</td>
</tr>
<tr>
<td></td>
<td>Statusword.Standstill</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>(Technology DB parameters)</td>
<td></td>
</tr>
<tr>
<td>MC_Stop</td>
<td>Statusword.Stopping</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>Statusword.Standstill</td>
<td>True</td>
</tr>
<tr>
<td></td>
<td>(Technology DB parameters)</td>
<td></td>
</tr>
<tr>
<td>MC_MoveAbsolute</td>
<td>Position</td>
<td>Input parameter position</td>
</tr>
<tr>
<td>(MC_MvAbs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MC_ExternalEncoder</td>
<td>Position</td>
<td>Input parameter position</td>
</tr>
<tr>
<td>(MC_ExEnc)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Introduction

S7-PLCSIM provides several view objects that allow you to monitor and modify various components of the simulated PLC. These view objects are listed below:

- **CPU (Page 52) View Object**
- **ACCUs & Status Word (Page 53) View Object**
- **Block Regs (Page 54) View Object**
- **View Object "Stacks" (Page 54)**
- **Input Variable (Page 55) View Object**
- **Output Variable (Page 56) View Object**
- **Bit Memory (Page 57) View Object**
- **Timer (Page 57) View Object**
- **Counter (Page 58) View Object**
- **Generic (Page 58) View Object**
- **Vertical Bits (Page 59) View Object**

Symbolic addressing in view objects

You can use symbolic addressing (Page 43) with view objects. If you do, tooltips are available for all view object fields that have symbols assigned to them. Point to a field with the mouse to see its symbolic address and comment (separated by a colon) in a tooltip box.

---

**Note**

If you use an address in a view object that corresponds to F-System peripheral I/O, S7-PLCSIM displays a yellow background for that view object.
View objects

5.1 CPU View Object

5.1 CPU View Object

Introduction

This view object is displayed by default when you open a new simulation.

Function

- Display status
- Change operating mode
- Reset memory with MRES
- Delete blocks and the hardware configuration with MRES

Note

The operating modes on the CPU view object function like the key switch on a real CPU: if you use the STEP 7 tools to change the operating mode, or if the CPU automatically changes mode (for example, encounters an error condition that causes the CPU to change from RUN to STOP), the RUN/STOP indicators also change. The key switch does not change. This alerts you that the CPU operating mode changed, possibly because of some error in the program.

5.1.1 CPU Operating Mode Switch Positions

RUN-P

The CPU runs the program, and you can change the program and its parameters. In order to use the STEP 7 tools for modifying any of the parameters of the program while the program is running, you must put the CPU in RUN-P mode. You can use the view objects created within S7-PLCSIM to modify any data used by the program.

When you select RUN-P, the operating mode status indicators on the CPU view object and STEP 7 display RUN.

RUN

The CPU runs the program by reading the inputs, executing the program, and then updating the outputs. By default, you cannot download any program or use the STEP 7 tools to change any parameters (such as input values) when the CPU is in RUN mode. If you have configured CiR (Configuration in Run) elements in your STEP 7 project, however, you can download the CiR objects in RUN mode. With the simulated CPU in RUN mode, you can use the view objects created within S7-PLCSIM to modify any data used by the program.
STOP

The CPU does not run the program. Unlike STOP mode for real CPUs, the outputs are not set to predefined ("safe") values but remain at the state they were in when the CPU changed to STOP mode. You can download programs to the CPU while the CPU is in STOP mode. Changing from STOP mode to RUN mode starts execution of the program from the first instruction.

The CPU operating modes, the CPU indicators (Page 53), and the Memory Clear/Reset (Page 41) button are all displayed on the CPU view object (Page 52). You can set the CPU operating mode with the Key Switch Position command. You can pause execution of the simulated PLC program when the CPU is in RUN or RUN-P mode.

5.2 ACCUs & Status Word View Object

To add this view object to the simulation, proceed as follows:

- Choose the menu command View > Accumulators
- Click the CPU Accumulators button:

Function

This view object allows you to monitor and modify the following data:

- **Accumulators**: Allows you to monitor the contents of the CPU accumulators. The view object displays four accumulator fields to also accommodate programs for the S7-400 CPU; Programs for the S7-300 CPU use only two accumulators.
- **Status word**: Allows you to monitor the bits of the status word.
- **Address registers**: Allows you to monitor the contents of the two address registers (AR1 and AR2). These address registers are used for the indirect addressing of data.
5.3 Block Regs View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command View > Block Registers
- Click the Block Registers button:

**Function**

This view object allows you to monitor the contents of the data block address registers (DB1 and DB2). This view object also displays the number of the current logic block and the previous logic block, along with the number of the instruction (step address counter, or SAC) for each block.

5.4 Nesting Stacks View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command View > Stacks
- Click on the "Stacks" symbol:

**Function**

This view object allows you to monitor information stored in the following stacks of the CPU:

- The nesting stack stores up to seven entries. For each entry, the nesting stack stores the states of the RLO and OR bits of the status word. An entry in the nesting stack is made for each instruction that starts a new logic string. These instructions are: And (A), And Not (AN), Or (O), Or Not (ON), Exclusive Or (X), and Exclusive Or Not (XN).
- The MCR stack stores up to eight levels of nesting for a master control relay (MCR). Each level shows the status of the RLO bit for an MCR instruction, which begins an MCR area.
5.5 Input Variable View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Input Variable
- Click the Insert Input Variable button:

**Note**
The CPU reacts immediately to any changes made with this view object. (Any modifications made with a STEP 7 variable table take effect at the proper time in the CPU scan: Inputs are read at the beginning of the scan, and outputs are written at the end.)

**Function**

This view object allows you to monitor and modify the following data:

- **Peripheral (external) input variables**: You can access the peripheral input (PI) memory areas of the CPU. S7-PLCSIM displays a yellow background for the view object if the variable address corresponds to F-System peripheral I/O.

- **Process input images**: You can access the input (I) memory areas of the CPU. By default, the CPU overwrites the I memory with the PI memory at the beginning of every scan. If you change an I memory value, the simulation immediately copies the changed value to the peripheral area. This way, the desired change is not lost when the peripheral value overwrites the process input value on the next scan.

**Note**
You can choose the numeric data format for the input variable and you can also use symbolic addressing if you have attached symbols. You can also view input variables with a Vertical Bits View Object.

Display of S7-300 PIs and PQs is not supported by S7-PLCSIM if connected via a CP card

S7-PLCSIM cannot display PIs and PQs for a simulated S7-300 if the PIs and PQs are connected via a CP card. The configuration of a CP card of a S7-300 differs from a S7-400. S7-PLCSIM only supports the CP card configuration of a S7-400.
5.6 Output Variable View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Output Variable
- Click the Insert Output Variable button:

Note

The CPU reacts immediately to any changes made with this view object. (Any modifications made to a STEP 7 variable take effect at the same time in the CPU scan: Inputs are read at the beginning of the scan, and outputs are written at the end.)

Function

This view object allows you to monitor and modify the following data:

- **Peripheral (external) output variables**: You can access the peripheral output (PQ) memory areas of the CPU. Any change to a PQ memory value immediately updates the corresponding output (Q) memory value. S7-PLCSIM displays a yellow background for the view object if the variable address corresponds to F-System peripheral I/O.

- **Process output images**: you can access the output (Q) memory areas of the CPU. During the scan cycle, the program calculates output values and places them in the process-image output table. At the end of the scan cycle, the operating system reads the calculated output values from this table and sends them to the process outputs. The process output image table maps the first 512 bytes (CPU-dependent) of the peripheral output memory.

Note

You can choose the numeric data format for the output variable and you can also use symbolic addressing if you have attached symbols. You can also view output variables with a Vertical Bits View Object.

Display of S7-300 PIs and PQs is not supported by S7-PLCSIM if connected via a CP card

S7-PLCSIM cannot display PIs and PQs for a simulated S7-300 if the PIs and PQs are connected via a CP card. The configuration of a CP card of a S7-300 differs from a S7-400. S7-PLCSIM only supports the CP card configuration of a S7-400.
5.7 Bit Memory View Object

To access this view object, do one of the following:

- Choose the menu command Insert > Bit Memory
- Click on the Insert Bit Memory button:

Function

This view object allows you to monitor and modify bit memory:

- variables that are stored in the bit memory (M) area of the CPU.
- The bit memory (M) area provides storage for interim results calculated in the program.
- Data format to be used to access the data

Note

You can choose the numeric data format for the bit memory and you can also use symbolic addressing if you have attached symbols. You can also view bit memory with a Vertical Bits View Object.

5.8 Timer View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Timer
- Click the Insert Timer button:

Function

This view object allows you to monitor and modify any timer used by your program. The timer view object displays the name of the timer, the actual timer value, and the time base.

Note

If you change the time base, the actual timer value changes, while the displayed value remains the same. This is because the actual timer value is the product of the displayed value and the time base. For example, if the value of timer T 0 is 600 and the time base is 10 ms, this represents a timer of 6 seconds. If you change the time base to 100 ms, then the actual timer value becomes 60 seconds. (600 * 100 ms = 60 seconds)

You can use symbolic addressing for the timers if you have attached symbols. You can also configure the timers to be under either automatic or manual control. Use the commands on the "Execute" menu for this.
5.9 Counter View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Counter
- Click the Insert Counter button:

Function

This view object allows you to monitor and modify the counters used by your program. This view object opens with a default memory location of C 0.

You can choose the numeric data format for the counter and you can also use symbolic addressing if you have attached symbols.

5.10 Generic View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Generic
- Click the Insert Generic Variable button:

Result: The view object "Variable" is opened.

Function

This view object allows you to monitor and modify the following data:

- Peripheral (external) input and output variables: you can access the peripheral input (PI) and peripheral output (PQ) memory areas of the CPU. S7-PLCSIM displays a yellow background for the view object if the variable address corresponds to F-System peripheral I/O.
- Process-image input and output variables: You can access the input (I) and output (Q) memory areas of the CPU. By default, the CPU overwrites the I memory with the PI memory at the beginning of every scan. If you change an I memory value, the simulation immediately copies the changed value to the peripheral area. This way, the desired change is not lost when the peripheral value overwrites the process input value on the next scan.
- Bit memory: You can access the variables stored in the bit memory (M) area of the CPU.
- Timers and counters: You can access the timers and counters used by the program.
- Data blocks: You can access the data stored in the data blocks of the program, for example, DB1.DBX 0.0 or DB1.DBW 0.

The CPU reacts immediately to any changes made with this view object. Any modifications made with a STEP 7 variable table take effect at the proper time in the CPU scan; inputs are read at the beginning of the scan, and outputs are written at the end.

You can choose the numeric data format for the generic variable and you can also use symbolic addressing if you have attached symbols.
5.11 Vertical Bits Variable View Object

To add this view object to a simulation, do one of the following:

- Choose the menu command Insert > Vertical Bits
- Click the Insert Vertical Bits button:

Result: The view object "Variable" is opened.

Function

You can use the Vertical Bits view object with bit or byte addresses. You can see the symbolic or absolute addresses of all bits in the "Vertical Bits" view object. You can monitor and control the following data:

- Peripheral (external) input and output variables: You can access the peripheral input (PI) and peripheral output (PQ) memory areas of the CPU. S7-PLCSIM displays a yellow background for the view object if the variable address corresponds to F-System peripheral I/O.

- Process-image inputs and outputs: You can access the input (I) and output (Q) memory areas of the CPU. By default, the CPU overwrites the I memory with the PI memory at the beginning of every scan. If you change an I memory value, the simulation immediately copies the changed value to the peripheral area. This way, the desired change is not lost when the peripheral value overwrites the process input value on the next scan.

- Bit memory: you can access the variables stored in the bit memory (M) area of the CPU.

- Data blocks: you can access the data stored in the data blocks of the program.

The CPU reacts immediately to any changes made with this view object. Any modifications made to a STEP 7 variable take effect at the same time in the CPU scan. Inputs are read at the beginning of the scan, and outputs are written at the end. You can use symbolic addressing if you have attached symbols for variables represented with a Vertical Bits view object.
View objects

5.11 Vertical Bits Variable View Object
Introduction

S7-PLCSIM supports the following interrupt and error OBs:

- OB40 to OB47 (Page 63) (hardware interrupt)
- OB70 (Page 64) (I/O redundancy error) {417-H systems only}
- OB72 (Page 65) (CPU redundancy error) {417-H systems only}
- OB73 (Page 67) (communication redundancy error) {417-H systems only}
- OB80 (Page 67) (time error)
- OB82 (Page 68) (diagnostic interrupt)
- OB83 (Page 69) (insert/remove module)
- OB85 (Page 70) (priority class error)
- OB86 (Page 71) (rack failure)

Procedure

To trigger the simulation of one of these OBs, proceed as follows:

1. Choose the menu command Execute > Trigger Error OB.
2. Select the desired OB or OB group. The OBs that are available depend on the hardware configuration loaded in the simulation.

Note

If an OB dialog is already open, changed system data that have been loaded in the simulation will not be taken into consideration. To cause the OB to use the changed data, you must close the OB dialog and reopen it.
6.1 Logical base addresses

Function

Logical addresses will be used as identification addresses of modules. The following OBs require the logical address of an input or output module in order to be able to identify them:

- OB40 - OB 47
- OB82
- OB83
- OB86

Definition

A logical address is the smallest address of an inserted module that can be configured in STEP 7 HW Config. The logical base address is always the smallest integer address.

Logical address for different module types

The logical start address depends on the type of module and the installation location:

<table>
<thead>
<tr>
<th>Module Type</th>
<th>Central rack</th>
<th>Profibus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input module</td>
<td>smallest input address for example, PIB0</td>
<td>smallest input address for example, PIB0</td>
</tr>
<tr>
<td>Output module</td>
<td>smallest output address for example, POB0</td>
<td>smallest output address for example, POB0</td>
</tr>
<tr>
<td>I/O module</td>
<td>smallest input address for example, PIB0</td>
<td>smallest module address (input or output) for example, PIB0 or POB0</td>
</tr>
</tbody>
</table>
6.2 Hardware Interrupt (OB40-OB47)

This dialog allows you to trigger an interrupt OB and test the program that you downloaded in OB40 to OB47.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Address</td>
<td>OB4x_MDL_ADDR</td>
<td>WORD</td>
<td>Logical base address (Page 62) of the module that triggers the interrupt.</td>
</tr>
</tbody>
</table>
| Module status           | OB4x_POINT_ADDR | DWORD (Hex) | In the case of digital modules: bit field with the status of the inputs on the module. (Bit 0 corresponds to the first input.)  
|                         |               |           | For analog modules, CPs or FMs: interrupt status of the module (irrelevant to user). |
| Interrupt OB (read-only)| OB4x_OB_NUMBR | BYTE      | OB number (40 to 47).                                                     |

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
6.3 I/O Redundancy Error (OB70)

This dialog allows you to trigger a loss of redundancy on the PROFIBUS-DP which calls the OB70. The menu command for opening this dialog can only be selected if the configuration of an H-CPU is loaded in the simulation.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter (Input/Output)</th>
<th>Variable</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Event Class</td>
<td>OB70_EV_CLASS</td>
<td>BYTE</td>
<td>Event class and IDs:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* B#16#72: Incoming event</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* B#16#73: Outgoing event</td>
</tr>
<tr>
<td>Error Code</td>
<td>OB70_FLT_ID</td>
<td>BYTE</td>
<td>Error code (possible values):</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* B#16#A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>* B#16#A3</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.

Also enter the following parameters:
- DP Master Base Address
- DP Master System ID
- DP Slave Base Address
- DP Station Number

The field entries must match the values in the STEP 7 hardware configuration of the simulated CPU.

Note

The DP Slave Base Address, DP Station Number, and the Input/Output selection are only available for Error Code 0xA3.
6.4 CPU redundancy error (OB72)

This dialog allows you to trigger a CPU loss of redundancy which calls the OB72. The menu command for opening this dialog can only be selected if the configuration of an H-CPU is loaded in the simulation. The operating system of the H CPU calls OB72 when one of the following events occurs:

- Loss of redundancy on the CPUs
- Reserve-master switchover
- Synchronization error
- Error in a SYNC module
- Data update abort
- Comparison error (e.g. RAM, PAA)

OB72 is executed by all H CPUs that are in the RUN mode following a suitable start event.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Event Class</td>
<td>OB72_EV_CLASS</td>
<td>B#16#73</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B#16#75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B#16#78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B#16#79</td>
</tr>
<tr>
<td>Error Code</td>
<td>OB72_FLT_ID</td>
<td>see fault ID table</td>
</tr>
</tbody>
</table>
### OB72 Fault IDs (Error Codes)

The error code in OB72FLT_ID indicates which event occurred that caused the OB72 call:

<table>
<thead>
<tr>
<th>OB72_FLT_ID</th>
<th>Start Event of OB72</th>
</tr>
</thead>
<tbody>
<tr>
<td>B#16#01</td>
<td>Loss of redundancy (1 of 2) due to a CPU failure</td>
</tr>
<tr>
<td>B#16#02</td>
<td>Loss of redundancy (1 of 2) due to STOP on the reserve triggered by user</td>
</tr>
<tr>
<td>B#16#03</td>
<td>H system (1 of 2) changed to redundant mode</td>
</tr>
<tr>
<td>B#16#20</td>
<td>Error in RAM comparison</td>
</tr>
<tr>
<td>B#16#21</td>
<td>Error comparing process image output value</td>
</tr>
<tr>
<td>B#16#22</td>
<td>Error comparing memory bits, timers, or counters</td>
</tr>
<tr>
<td>B#16#23</td>
<td>Different operating system data recognized</td>
</tr>
<tr>
<td>B#16#31</td>
<td>Standby-master switchover due to master failure</td>
</tr>
<tr>
<td>B#16#33</td>
<td>Standby-master switchover due to operator intervention</td>
</tr>
<tr>
<td>B#16#34</td>
<td>Standby-master switchover due to sync module connection problem</td>
</tr>
<tr>
<td>B#16#35</td>
<td>Standby-master switchover triggered by 90 &quot;H_CTRL&quot;</td>
</tr>
<tr>
<td>B#16#40</td>
<td>Synchronization error in user program due to elapsed wait time</td>
</tr>
<tr>
<td>B#16#41</td>
<td>Synchronization error in user program due to waiting at different synchronization points</td>
</tr>
<tr>
<td>B#16#42</td>
<td>Synchronization error in operating system due to waiting at different synchronization points</td>
</tr>
<tr>
<td>B#16#43</td>
<td>Synchronization error in operating system due to elapsed wait time</td>
</tr>
<tr>
<td>B#16#44</td>
<td>Synchronization error in operating system due to wrong data</td>
</tr>
<tr>
<td>B#16#50</td>
<td>Missing SYNC module</td>
</tr>
<tr>
<td>B#16#51</td>
<td>Modification at SYNC module without Power On</td>
</tr>
<tr>
<td>B#16#52</td>
<td>SYNC module removed/inserted</td>
</tr>
<tr>
<td>B#16#53</td>
<td>Modification at SYNC module without reset</td>
</tr>
<tr>
<td>B#16#54</td>
<td>SYNC module: rack number assigned twice</td>
</tr>
<tr>
<td>B#16#55</td>
<td>SYNC module error/eliminated</td>
</tr>
<tr>
<td>B#16#56</td>
<td>Unauthorized rack number set on SYNC module</td>
</tr>
<tr>
<td>B#16#C1</td>
<td>Data update abort</td>
</tr>
<tr>
<td>B#16#C2</td>
<td>Abort of update attempt because a monitoring time was exceeded during the (n)th attempt ((1 \leq n \leq \text{maximum possible number of update attempts after an abort due to the monitoring time being exceeded}))</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
6.5 Communication Redundancy Error (OB73)

This dialog allows you to trigger a loss of redundancy for a fault-tolerant S7 connection. The menu command for opening this dialog can only be selected if the configuration of an H-CPU is loaded in the simulation.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Event Class</td>
<td>OB73_EV_CLASS</td>
<td>BYTE</td>
<td>Possible value of B#16#73 (loss of redundancy in communication) or B#16#72 (problem eliminated)</td>
</tr>
<tr>
<td>Error Code</td>
<td>OB73_FLT_ID</td>
<td>BYTE</td>
<td>Possible value of B#16#E0</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.

6.6 Time Error (OB80)

This dialog allows you to trigger a time error which calls the OB80.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle time exceeded</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#01</td>
</tr>
<tr>
<td>Requested OB is still being processed</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#02</td>
</tr>
<tr>
<td>OB call buffer overflow for the current priority class</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#07</td>
</tr>
<tr>
<td>Expired time of day interrupt:</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#05</td>
</tr>
<tr>
<td>* Due to a time jump</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#05</td>
</tr>
<tr>
<td>* on return to RUN after STOP</td>
<td>OB80_FLT_ID</td>
<td>BYTE</td>
<td>Error code: B#16#06</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
6.7 Diagnostic interrupt (OB82)

This dialog allows you to trigger a diagnostic interrupt which calls the OB82.

Error conditions

Default Tests: (Optional) In this drop-down box, choose from the list to set the fault conditions for the desired test automatically.

For all fault conditions the event class (OB82_EV_CLASS) is set to B#16#39 (incoming event) and OB82_FLT_ID uses the error code B#16#42. If no fault condition is selected (Module OK), the event class is set to B#16#38 (outgoing event).

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

Module Address

In the chapter Logical base addresses (Page 62), you can find information on logical start addresses for various module types.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module Address</td>
<td>OB82_MDL_ADDR</td>
<td>Int</td>
</tr>
</tbody>
</table>

Error conditions

Check the appropriate check boxes to enable the following fault conditions:

<table>
<thead>
<tr>
<th>Parameter Check box</th>
<th>Variable</th>
<th>Data type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module defect (read-only)</td>
<td>OB82_MDL_DEFECT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Internal error</td>
<td>OB82_INT_FAULT</td>
<td>BOOL</td>
</tr>
<tr>
<td>External error</td>
<td>OB80_EXT_FAULT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Channel error</td>
<td>OB82_PNT_INFO</td>
<td>BOOL</td>
</tr>
<tr>
<td>External auxiliary supply missing</td>
<td>OB82_EXT_VOLTAGE</td>
<td>BOOL</td>
</tr>
<tr>
<td>Front panel connector not plugged in</td>
<td>OB82_FLD_CONNCTR</td>
<td>BOOL</td>
</tr>
<tr>
<td>Module not configured</td>
<td>OB82_NO_CONFIG</td>
<td>BOOL</td>
</tr>
<tr>
<td>Incorrect parameters on module</td>
<td>OB82_CONFIG_ERR</td>
<td>BOOL</td>
</tr>
<tr>
<td>Channel information available</td>
<td>OB82_MDL_TYPE</td>
<td>BYTE (Bit 4)</td>
</tr>
<tr>
<td>User information available</td>
<td>OB82_MDL_TYPE</td>
<td>BYTE (Bit 5)</td>
</tr>
<tr>
<td>Diagnostic interrupt from substitute</td>
<td>OB82_MDL_TYPE</td>
<td>BYTE (Bit 6)</td>
</tr>
<tr>
<td>User module is missing or has an error</td>
<td>OB82_SUB_MDL_FAULT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Communication problem</td>
<td>OB82_COMM_FAULT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Operating mode is STOP</td>
<td>OB82_MDL_STOP</td>
<td>BOOL</td>
</tr>
<tr>
<td>Watchdog timer responded</td>
<td>OB82_WTCH_DOG_FLT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Internal power supply failed</td>
<td>OB82_INT_PS_FLT</td>
<td>BOOL</td>
</tr>
<tr>
<td>Battery exhausted</td>
<td>OB82_PRIM_BATT_FLT</td>
<td>BOOL</td>
</tr>
</tbody>
</table>
6.8 Insert/Remove Module Interrupt (OB83)

This dialog allows you to trigger an insert/remove interrupt which calls the OB83.

Error conditions

For all fault conditions, the event class (OB83_EV_CLASS) is set to B#16#39 (incoming event). If no fault condition is selected (Module OK), the event class is set to B#16#38 (outgoing event).

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Value / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module removed or not responding</td>
<td>OB83_FLT_ID</td>
<td>BYTE</td>
<td>B#16#61</td>
</tr>
<tr>
<td>Module inserted: module type OK</td>
<td>OB83_FLT_ID</td>
<td>BYTE</td>
<td>B#16#61</td>
</tr>
<tr>
<td>Module inserted: incorrect module type</td>
<td>OB83_FLT_ID</td>
<td>BYTE</td>
<td>B#16#63</td>
</tr>
<tr>
<td>Module inserted: type ID cannot be read</td>
<td>OB83_FLT_ID</td>
<td>BYTE</td>
<td>B#16#64</td>
</tr>
<tr>
<td>Module inserted: error in module parameter assignment</td>
<td>OB83_FLT_ID</td>
<td>BYTE</td>
<td>B#16#65</td>
</tr>
<tr>
<td>Module Address</td>
<td>OB83_MDL_ADDR</td>
<td>WORD</td>
<td>In the chapter Logical base addresses (Page 62), you can find information on logical start addresses for various module types.</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
6.9 Priority Class Error (OB85)

This dialog allows you to trigger a program sequence error which calls the OB85. The OB85 is automatically triggered if an error occurs while the operating system is accessing a block (error code B#16#A3).

Note that events B#16#A1, B#16#A2, B#16#B1, and B#16#B2 must be generated by other means, for example, by deleting one of the required blocks from your program.

Programmable parameters

The parameters that you select with this dialog are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated function:</td>
<td>OB85_Z1</td>
<td>WORD</td>
<td>W#16#0100 W#16#0101 W#16#0102 W#16#0103</td>
</tr>
<tr>
<td>No error resolution</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block not loaded</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Range length error</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Write protect error</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| IEC timer:              | OB85_Z1  | WORD      | W#16#0200 W#16#0201 W#16#0202 W#16#0203 |
| No error resolution     |          |           |           |
| Block not loaded        |          |           |           |
| Range length error      |          |           |           |
| Write protect error     |          |           |           |

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
6.10 Rack Failure (OB86)

This dialog allows you to trigger a rack failure which calls the OB86.

Error conditions

For all fault conditions the event class (OB86_EV_CLASS) is set to B#16#39 (incoming event). If no fault condition is selected (Rack Return, Rack Return with Discrepancy, DP Station Return and DP Station OK), the event class is set to B#16#38 (outgoing event).

The two tabs of the dialog provide the following options:

- Expansion Rack Failure Tab
- DP Failure Tab

Expansion Rack Failure Tab

The parameters that you select in this tab are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IM address</td>
<td>OB86_MDL_ADDR</td>
<td>WORD</td>
<td>Address of the interface module to which expansion racks are connected.</td>
</tr>
<tr>
<td>Failure</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C1</td>
</tr>
<tr>
<td>Return</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C1</td>
</tr>
<tr>
<td>Return with discrepancy</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C2</td>
</tr>
<tr>
<td>Expansion Rack operational again but error in module parameter assignment</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C6</td>
</tr>
<tr>
<td>Rack Status</td>
<td>OB86_RACKS_FLTD</td>
<td>ARRAY OF BOOL</td>
<td>Shows the status of up to twenty-one expansion racks connected to the interface module (IM). In the array, you must select the rack with the fault conditions. Green = Good Red = Fault Gray = Not configured</td>
</tr>
</tbody>
</table>
### DP Failure Tab

This tab allows you to trigger faults in the DP system and view the status of the DP under various error conditions. The parameters that you select in this tab are passed to the following variables when the OB is called:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Variable</th>
<th>Data type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet</td>
<td>OB86_MDL_ADDR</td>
<td>WORD</td>
<td>Logical base address of the DP master system.</td>
</tr>
<tr>
<td>Failure of DP master system</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C3</td>
</tr>
<tr>
<td>Station failure</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C4</td>
</tr>
<tr>
<td>Station return</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C4</td>
</tr>
<tr>
<td>All station return</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C4</td>
</tr>
<tr>
<td>Triggers a &quot;station return&quot; for all faulty DP slaves. No message is issued.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Station operational again but error</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C7</td>
</tr>
<tr>
<td>Station return with discrepancy</td>
<td>OB86_FLT_ID</td>
<td>BYTE</td>
<td>B#16#C8</td>
</tr>
<tr>
<td>DP Status</td>
<td>OB86_RACKS_FLTD</td>
<td>ARRAY OF BOOL</td>
<td>Shows the status of up to 126 DP stations. In the array, you must select the rack with the fault conditions. Green = Good Red = Fault Gray = Not configured</td>
</tr>
</tbody>
</table>

For further information, refer to the STEP 7 Help on Organization Blocks help system or to the SIMATIC System Software for S7-300/400 System and Standard Functions manual.
# Reference information

## 7.1 Icons and menu commands

S7-PLCSIM Menu Commands

<table>
<thead>
<tr>
<th>Icon</th>
<th>Toolbar</th>
<th>Menu command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; New PLC</td>
<td>Generates a new instance with a new CPU in original state.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Open PLC</td>
<td>Closes the current simulation and generates a new CPU from the saved data in the same instance.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Close PLC</td>
<td>Closes the current simulation and generates a new CPU in the original state in the same instance.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Save PLC</td>
<td>Saves the current simulation.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Save PLC As</td>
<td>Saves the current simulation under a new name.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Open Layout</td>
<td>Opens a saved layout.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Close Layout</td>
<td>Closes the current layout.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Save Layout</td>
<td>Saves the current arrangement as a layout.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Save Layout As</td>
<td>Saves the current layout under a new name.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Recent Simulation</td>
<td>Opens a recent simulation.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Recent Layout</td>
<td>Opens a recent layout.</td>
</tr>
<tr>
<td><img src="image" alt="Simulation" /></td>
<td>Standard</td>
<td>File &gt; Exit</td>
<td>Closes all the windows of the application and exits the application.</td>
</tr>
<tr>
<td><img src="image" alt="Edit" /></td>
<td>Standard</td>
<td>Edit &gt; Undo</td>
<td>Undoes the last action.</td>
</tr>
<tr>
<td><img src="image" alt="Edit" /></td>
<td>Standard</td>
<td>Edit &gt; Cut</td>
<td>Deletes the selected objects and saves them to the clipboard.</td>
</tr>
<tr>
<td><img src="image" alt="Edit" /></td>
<td>Standard</td>
<td>Edit &gt; Copy</td>
<td>Copies the selected objects and saves them to the clipboard.</td>
</tr>
<tr>
<td><img src="image" alt="Edit" /></td>
<td>Standard</td>
<td>Edit &gt; Paste</td>
<td>Inserts the contents of the clipboard at the cursor position.</td>
</tr>
</tbody>
</table>
### 7.1 Icons and menu commands

<table>
<thead>
<tr>
<th>Icon</th>
<th>Toolbar</th>
<th>Menu command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![View Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Accumulators</td>
<td>Displays Accumulators 1 to 4 and the status word.</td>
</tr>
<tr>
<td>![View Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Block Registers</td>
<td>Displays the address registers and data blocks registers.</td>
</tr>
<tr>
<td>![View Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Stacks</td>
<td>Displays the MCR stacks and nesting stacks.</td>
</tr>
<tr>
<td>![View Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Toolbars</td>
<td>Displays specific toolbars (on/off).</td>
</tr>
<tr>
<td>![View Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Status Bar</td>
<td>Displays the status bar (on/off).</td>
</tr>
<tr>
<td>![Standard Icon]</td>
<td>![Toolbar Icon]</td>
<td>View &gt; Always On Top</td>
<td>Always displays the simulation on top.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Input Variable</td>
<td>Displays an input variable.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Output Variable</td>
<td>Displays an output variable.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Bit Memory</td>
<td>Displays a bit memory.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Timer</td>
<td>Displays a timer.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Counter</td>
<td>Displays a counter.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Generic</td>
<td>Displays a numeric display.</td>
</tr>
<tr>
<td>![Insert Icon]</td>
<td>![Toolbar Icon]</td>
<td>Insert &gt; Vertical Bits</td>
<td>Displays a byte.</td>
</tr>
</tbody>
</table>

### Target system

<table>
<thead>
<tr>
<th>Icon</th>
<th>Menu command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![CPU Mode Icon]</td>
<td>PLC &gt; Power On</td>
<td>Turns on the PLC.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>PLC &gt; Power Off</td>
<td>Turns off the PLC.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>PLC &gt; Clear/Reset</td>
<td>Deletes the control program and variable memory.</td>
</tr>
</tbody>
</table>

### Execute

<table>
<thead>
<tr>
<th>Icon</th>
<th>Menu command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Key Switch Position</td>
<td>Puts the key switch of the CPU in the selected mode.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Startup Switch Position</td>
<td>Sets the startup switch position.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Scan Mode</td>
<td>Sets the mode.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Next Scan</td>
<td>Runs the next scan.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Pause</td>
<td>Halts the program immediately.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Automatic Timers</td>
<td>Sets all timers to automatic mode.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Manual Timers</td>
<td>Sets all timers to manual mode.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Reset Timers</td>
<td>Resets one or all timers.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Trigger Error OB</td>
<td>Triggers an error OB.</td>
</tr>
<tr>
<td>![CPU Mode Icon]</td>
<td>Execute &gt; Scan Cycle Monitoring</td>
<td>Is used to set and activate the scan cycle monitoring time.</td>
</tr>
</tbody>
</table>
## 7.1 Icons and menu commands

<table>
<thead>
<tr>
<th>Icon</th>
<th>Toolbar</th>
<th>Menu command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Tools</td>
<td>Tools &gt; Record/Playback</td>
<td>Record or play back a series of events.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Tools &gt; Options &gt; Attach Symbols</td>
<td>Searches for the symbol table of the downloaded program.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Tools &gt; Options &gt; Show Symbols</td>
<td>Displays the symbol of a variable.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Tools &gt; Options &gt; Reference Data</td>
<td>Displays current reference data for the current program.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Tools &gt; Options &gt; Symbol Table</td>
<td>Opens the current symbol table.</td>
</tr>
<tr>
<td></td>
<td>Window</td>
<td>Window &gt; Cascade</td>
<td>Arranges all of the open windows so that they are overlapping.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td>Window &gt; Tile Ordered</td>
<td>Arranges all of the open windows in a logical order.</td>
</tr>
<tr>
<td></td>
<td>Window</td>
<td>Window &gt; Arrange Icons</td>
<td>Arranges the symbols along the bottom edge of the window.</td>
</tr>
<tr>
<td></td>
<td>Window</td>
<td>Window &gt; 1,2,3 ... 9</td>
<td>Activates a view object that is already open.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Help &gt; Contents</td>
<td>Displays the index of help topics.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Help &gt; Introduction</td>
<td>Describes the functional scope of this application.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Help &gt; Getting Started</td>
<td>Describes the essential steps for using this application.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Help &gt; Using Help</td>
<td>Displays information about using the Help.</td>
</tr>
<tr>
<td></td>
<td>Help</td>
<td>Help &gt; About</td>
<td>Displays information about the current version of this application.</td>
</tr>
<tr>
<td></td>
<td>Standard</td>
<td></td>
<td>Displays help on buttons, menus and dialogs.</td>
</tr>
</tbody>
</table>
## 7.2 S7-PLCSIM Numeric Data Formats

**Supported Numeric Data Formats**

Refer to the table below for the numeric data formats supported by S7-PLCSIM.

<table>
<thead>
<tr>
<th>Numeric Data Formats</th>
<th>Size</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>Bit, Byte</td>
<td></td>
</tr>
<tr>
<td>Binary</td>
<td>Byte, Word</td>
<td>1001_0011</td>
</tr>
<tr>
<td>Decimal</td>
<td>Byte, Word, Double Word</td>
<td>232</td>
</tr>
<tr>
<td>Hex (Hexadecimal)</td>
<td>Byte, Word, Double Word</td>
<td>9A</td>
</tr>
<tr>
<td>S7 format</td>
<td>Byte, Word, Double Word</td>
<td>dw#16#9a2ff23</td>
</tr>
<tr>
<td>Integer</td>
<td>Word, Double Word</td>
<td>632, -2370</td>
</tr>
<tr>
<td>BCD (Binary-coded Decimal)</td>
<td>Word, Double Word</td>
<td>400</td>
</tr>
<tr>
<td>Real</td>
<td>Double word</td>
<td>1.234567e+023</td>
</tr>
<tr>
<td>Char (Character)</td>
<td>Byte, Word, Double Word</td>
<td>'C', 'AB'</td>
</tr>
<tr>
<td>String</td>
<td>254 Alphanumeric Characters</td>
<td>This is a string</td>
</tr>
<tr>
<td>DT (DATE_AND_TIME)</td>
<td>8 bytes</td>
<td>2006-12-25-08:01:01</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The DT numeric data format is does not support milliseconds. If all 8 bytes are 0, the default DT display is: 1999-11-30-00:00:00.</td>
</tr>
<tr>
<td>S5TIME</td>
<td>WORD</td>
<td>3m5s00ms</td>
</tr>
<tr>
<td>Date</td>
<td>WORD</td>
<td>1998-06-18</td>
</tr>
<tr>
<td>Timer</td>
<td>Double word</td>
<td>9h26m53s703ms</td>
</tr>
<tr>
<td>TOD</td>
<td>Double word</td>
<td>9:26:53.702</td>
</tr>
<tr>
<td>Slider: Dec</td>
<td>Byte, Word, Double Word</td>
<td></td>
</tr>
<tr>
<td>Slider: Int</td>
<td>Word, Double Word</td>
<td></td>
</tr>
<tr>
<td>Slider: Real</td>
<td>Double word</td>
<td></td>
</tr>
</tbody>
</table>
### 7.3 Troubleshooting Tips

#### Problems and Recommended Corrective Actions

The following table describes some problems that you can encounter when using S7-PLCSIM. The possible causes of the problem and recommended corrective actions are listed.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Possible Causes and Corrections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your program does not download to the simulated CPU.</td>
<td>Verify that the CPU is in either STOP mode or RUN-P mode. You cannot download your program if the simulated CPU is in RUN mode, unless you have configured CiR (Configuration in RUN) elements in STEP 7. CiR objects are the only objects that can be downloaded to S7-PLCSIM in RUN mode.</td>
</tr>
<tr>
<td></td>
<td>If your program contains a System Data Block (SDB), verify that the CPU is in STOP mode. As with a real CPU, you can download SDBs only when the CPU is in STOP mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the CPU view object is in RUN-P mode, STEP 7 prompts you to change to STOP mode so that the hardware configuration can be downloaded.</td>
</tr>
<tr>
<td></td>
<td>Verify that the CPU and the program use the same node address and subnet name. The node address defined for the program must match the node address of the CPU.</td>
</tr>
<tr>
<td>The S7-PLCSIM application does not respond and appears to be &quot;locked up.&quot;</td>
<td>Check the execution mode. The simulation can appear to be locked up when paused or in Single Scan mode.</td>
</tr>
<tr>
<td>You enter a peripheral variable and get an &quot;Invalid Address&quot; error, even though the address value is valid. - or - You get a peripheral access error in your program, even though your S7-300 project contains the correct configuration.</td>
<td>Only the CPU 315-2DP, CPU 316-2DP, and CPU 318-2 download an I/O configuration. If you download a program from another S7-300 CPU, the system data does not include the I/O configuration. This causes errors when you attempt to access peripheral I/O in S7-PLCSIM. To avoid these errors, first of all create a hardware configuration with configured I/O modules in the system data. This way you can define which CPU modules should be made available. To do this, create a project and configure a S7-300 CPU in which the I/O are not automatically configured, for example the CPU 315-2DP, CPU 316-2DP or the CPU 318-2. Download this hardware configuration to the S7-PLCSIM. Then you can download the program blocks from any S7 programs. The I/Os are applied error-free.</td>
</tr>
<tr>
<td>Scan exceeded with cyclic interrupts</td>
<td>When simulating systems you must ensure that there is sufficient time between the start events of the individual cyclic interrupt OBs for processing the cyclic interrupts. It may be necessary to extend the intervals of the cyclic interrupts proportionally.</td>
</tr>
</tbody>
</table>
Index

A
Accumulators and Status Word, viewing, 53
Accumulators, similarity to S7-400 CPUs, 10
Address register, 53
Attaching symbols to a simulation, 43

B
Back button, Help window, 28
BCD (Binary-coded decimal) numeric format, 76
Binary, numeric format, 76
Bit memory, 14
Bit Memory
  Viewing and modifying, 57
Bits, numeric data format, 76
Block registers, viewing and modifying, 54

C
C (counter), 14
Char (Character), 76
CiR elements, 52
Close
  Simulation with connections, 77
Communication Redundancy Error (OB73), 67
Contents tab, Help window, 28
Continuous scan mode
  Setting, 35
  Use in debugging, 39
Control program, monitoring, 37
Counter
  Viewing and modifying, 58
CPU 317-T, simulation, 50
CPU accumulators, 53
CPU Indicators, 53
CPU operating modes, 52
CPU redundancy error (OB72), 85
Creating a new event file, 44
Customer support, 7

D
Data blocks, 14
Data formats, 76
Date, numeric data format, 76
DATE_AND_TIME, numeric data format, 76
DC indicator, 53
Debugging with S7-PLCSIM, 39
Decimal, numeric data format, 76
Delta button, record/playback, 44
Diagnostic buffer, differences from a "real" PLC,
Diagnostic interrupt (OB82), 68
Differences from a "real" PLC,
Documentation, related, 7
Download to CPU
  Problems, 77
DP Failure Tab, 71
DP indicator, 53
DT (DATE_AND_TIME), numeric data format, 76

E
Error OBs, 61
  OB70 (I/O redundancy error), 64
  OB72 (CPU redundancy error), 85
  OB73 (Communication Redundancy Error), 87
  OB80 (time error), 67
  OB85 (priority class error), 70
  OB86 (rack failure), 71
Expansion Rack FailureTab, 71

F
Formats, numeric data, 76
F-System peripheral I/O, 51, 55, 56

G
Generic variables, viewing and modifying, 58
Index

H
Hardware Interrupt (OB40-OB47)T, 63
Help, 28
  Buttons in the Help window, 28
  F1 access, 28
Hexadecimal (Hex), numeric data format, 76
Hide button, Help window, 28
Highlighting search terms, online help, 28

I
I (input), 14
I/O (input/output)
  Differences from a "real" PLC, 64
  Input Variable, 55
  Output Variable, 56
Index tab, Help window, 28
Input, 14
Input Variable
  Variable, viewing and modifying, 55
Insert Module Interrupt (OB83), 69
Insert/Remove Module Interrupt (OB83), 69
Integer, numeric data format, 76
Interrupt OBs, 61
  OB40-OB47 (Hardware Interrupt )T, 63
  OB82 (diagnostic interrupt), 68
  OB83 (insert/remove interrupt), 69

K
Key Switch Position Command, 52

L
LED indicators, 53

M
M (bit memory), 14
Manual timers, using, 42
Manuals, related, 7
Master Control Relay (MCR), 54
Maximum slider value, 38
MC Commands, 50
MCR (Master Control Relay ), 54
Memory areas, 14
  Differences from a "real" PLC, 10
  I/O, 10
  Process, 10
Minimum slider value, 38
Mode, scan, 35
Monitoring
  Simulated program, 37
Motion control simulation, 50
Multi-instance
  Communications, 24
  Downloading a STEP 7 project, 23
  Save, 23
  Select type of connection, 24
  Simulate, 23

N
Nesting stacks, 54
Next scan, executing, 35
Numeric Data Format, 76
Numeric data format, slider control, 38

O
OB40 to OB47, 63
OB70, 64
OB72, 65
OB73, 67
OB80, 67
OB82, 68
OB83, 69
OB85, 70
OB86, 71
OBs, 61
  Communication Redundancy Error (OB73)T, 67
  CPU redundancy error (OB72), 65
  Diagnostic interrupt (OB82), 68
  Hardware Interrupt (OB40-OB47)T, 63
  I/O Redundancy Error (OB70), 64
  Insert/Remove Module Interrupt (OB83), 69
  Priority Class Error (OB85)T, 70
  Rack Failure (OB86), 71
  Time Error (OB80), 67
Online help, 28
  Buttons in the Help window, 28
  F1 access, 28
Open
  Event file, 44
  Open connection message, 77
Operating modes, 52
Options button, Help window, 28
Output, 14
Output Variable
  Viewing and modifying, 56

S7-PLCSIM V5.4
Operating Manual, 07/2011, A5E00992424-03
Index

P
Pausing recording or playback, 44
Peripheral memory, 14
Access error, 77
Input Variable, 55
Output Variable, 56
Update differences from a "real" PLC,
Vertical Bits, 59
PI (peripheral input), 14
Playback, 44
PQ (peripheral output), 14
Print button, Help window, 28
Priority Class Error (OB85), 70
Process image, 14
Input Variable, 55
Output Variable, 56
Update differences from a "real" PLC,
Vertical Bits, 59
Program simulation, 37
ProSim, 7

Q
Q (output), 14

R
Rack Failure (OB86), 71
Real S7 PLC, S7-PLCSIM differences, 10
Real, numeric data format, 76
Record/Playback, 44
Record/playback status, 44
Redundancy error
Communication (OB73) T, 67
CPU (OB72), 65
I/O (OB70), 64
Reset Timers, 42, 57
RUN, 52
LED indicator, 53
RUN-P, 52

S
S5TIME, numeric data format, 76
S7 format, numeric data format, 76
S7-PLCSIM locked, 77
S7ProSim, 7
Safe state not changed, 10
Save
Event file, 44
Layout, 32
Scan mode options, 35
Searching online help, 28
Selecting
Maximum slider value, 38
Minimum slider value, 38
Numeric data format, 38
Scan mode options, 35
SF indicator, 53
Show button, Help window, 28
Showing and attaching symbols, 43
Simulated PLC
Monitoring, 37
Simulating a CPU 317-T, 50
Single scan mode
Setting, 35
Single scan program execution
Use in debugging, 39
Slider, numeric data format, 76
Station failure, 71
Status word, 53
STOP, 52
Differences from a "real" PLC,
LED indicator, 53
Stopping record/playback, 44
String, numeric data format, 76
Strings, 76
Symbol table, 43
Symbolic addressing, 43

T
T (timer), 14
Target system
Simulation, 37
Technical support, 7
Telephone numbers, support, 7
Time Error (OB80), 67
Time, numeric data format, 76
Timers, 14
Differences from a "real" PLC,
Manual, 42
Reset, 42
Viewing and modifying, 57
Tool tips for symbols, 43
Troubleshooting in the PLC simulation, 77
General, 77
Record/Playback, 44
Troubleshooting Tips, 77

S7-PLCSIM V5.4
Operating Manual, 07/2011, A5E00992424-03
81
Turning power on and off for a simulated CPU, 42

U

Using

- Help, 28
- Record/Playback, 44
- S7-PLCSIM to debug your program, 39
- S7-PLCSIM to monitor simulated program, 37
- Slider Control, 38
- Symbolic addressing, 43
- View objects, 51

V

Variable table, 10

Vertical bits, viewing and modifying, 59

View object

- ACCUs & Status Word, 53
- Bit Memory, 57
- Block Registers, 54
- Counter, 58
- Generic variable, 58
- Input Variable, 55
- Output Variable, 56
- Stacks, 54
- Timers, 57
- Vertical Bits, 59

View objects, 51

- CPU, 52
- F-System variable indication, 51
- Slider, 38
- Using, 37