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
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

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.
Preface

Scope and standards
This document is part of the Engineering System Handling documentation package.

Scope of validity
This manual is valid for SIMOTION SCOUT V4.4.

SIMOTION Documentation
An overview of the SIMOTION documentation can be found in the SIMOTION Documentation Overview document.
This documentation is included as electronic documentation in the scope of delivery of SIMOTION SCOUT. It comprises ten documentation packages.
The following documentation packages are available for SIMOTION V4.4:

- SIMOTION Engineering System Handling
- SIMOTION System and Function Descriptions
- SIMOTION Service and Diagnostics
- SIMOTION IT
- SIMOTION Programming
- SIMOTION Programming - References
- SIMOTION C
- SIMOTION P
- SIMOTION D
- SIMOTION Supplementary Documentation

Hotline and Internet addresses

Additional information
Click the following link to find information on the following topics:

- Ordering documentation / overview of documentation
- Additional links to download documents
- Using documentation online (find and search manuals/information)

http://www.siemens.com/motioncontrol/docu
My Documentation Manager

Click the following link for information on how to compile documentation individually on the basis of Siemens content and how to adapt it for the purpose of your own machine documentation:

http://www.siemens.com/mdm

Training

Click the following link for information on SITRAIN - Siemens training courses for automation products, systems and solutions:

http://www.siemens.com/sitrain

FAQs

Frequently Asked Questions can be found in SIMOTION Utilities & Applications, which are included in the scope of delivery of SIMOTION SCOUT, and in the Service&Support pages in Product Support:

http://support.automation.siemens.com

Technical support

Country-specific telephone numbers for technical support are provided on the Internet under Contact:

http://www.siemens.com/automation/service&support
# Table of contents

**Preface** .......................................................................................................................................................... 3

1 **Safety notes** ............................................................................................................................................... 9
  1.1 Fundamental safety instructions ................................................................................................................ 9
  1.1.1 General safety instructions .................................................................................................................... 9
  1.1.2 Safety instructions for electromagnetic fields (EMF) .......................................................................... 12
  1.1.3 Handling electrostatic sensitive devices (ESD) ..................................................................................... 13
  1.1.4 Industrial security .................................................................................................................................. 13
  1.1.5 Residual risks of power drive systems ................................................................................................. 15
  1.2 Specific safety information ...................................................................................................................... 16

2 **Getting Started with SIMOTION SCOUT** .............................................................................................. 17
  2.1 Aim of Getting Started ............................................................................................................................ 17
  2.2 Sample project ......................................................................................................................................... 17
  2.3 Preconditions .......................................................................................................................................... 20
  2.4 General information ................................................................................................................................. 21
    2.4.1 SIMOTION SCOUT online help ........................................................................................................ 21
    2.4.2 Available documentation ................................................................................................................ 22
    2.4.3 Project generator ................................................................................................................................ 22
    2.4.4 Utilities & applications ..................................................................................................................... 23

3 **Prepare the configuration** ...................................................................................................................... 25
  3.1 Restore factory settings ............................................................................................................................ 25
  3.2 Set up interface for online communication .......................................................................................... 26
    3.2.1 Prepare the Ethernet interface of the PG/PC .................................................................................. 28
    3.2.2 Define the access point of the SIMOTION device ........................................................................ 29
  3.3 Result in the sample project .................................................................................................................... 30

4 **Create a project** .................................................................................................................................. 31
  4.1 Overview .................................................................................................................................................. 31
  4.2 Project ..................................................................................................................................................... 31
  4.3 Create new project .................................................................................................................................. 31
  4.4 Result in the sample project .................................................................................................................... 33

5 **Create SIMOTION device and configure online communication** ......................................................... 35
  5.1 Overview ................................................................................................................................................ 35
  5.2 SIMOTION D platform .............................................................................................................................. 35
  5.3 Create SIMOTION device ......................................................................................................................... 36
    5.3.1 Steps ................................................................................................................................................ 36
    5.3.2 Create SIMOTION device ............................................................................................................... 37
    5.3.3 Configure the PROFINET interface ............................................................................................... 38
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.4</td>
<td>Set up PG/PC communication</td>
<td>39</td>
</tr>
<tr>
<td>5.4</td>
<td>Result in the sample project</td>
<td>40</td>
</tr>
<tr>
<td>6</td>
<td>Download the project to the target system</td>
<td>41</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview</td>
<td>41</td>
</tr>
<tr>
<td>6.2</td>
<td>Save and compile the project</td>
<td>41</td>
</tr>
<tr>
<td>6.3</td>
<td>Connect to selected target devices – Go online</td>
<td>42</td>
</tr>
<tr>
<td>6.4</td>
<td>Download the project to the target system</td>
<td>45</td>
</tr>
<tr>
<td>7</td>
<td>Configure the drive</td>
<td>47</td>
</tr>
<tr>
<td>7.1</td>
<td>Overview</td>
<td>47</td>
</tr>
<tr>
<td>7.2</td>
<td>Drive</td>
<td>47</td>
</tr>
<tr>
<td>7.3</td>
<td>Automatic drive configuration</td>
<td>47</td>
</tr>
<tr>
<td>7.4</td>
<td>Automatic configuration of the drive</td>
<td>48</td>
</tr>
<tr>
<td>7.5</td>
<td>Result in the sample project</td>
<td>51</td>
</tr>
<tr>
<td>8</td>
<td>Configure the infeed</td>
<td>53</td>
</tr>
<tr>
<td>8.1</td>
<td>Overview</td>
<td>53</td>
</tr>
<tr>
<td>8.2</td>
<td>Configuring an infeed without DRIVE-CLiQ interface</td>
<td>54</td>
</tr>
<tr>
<td>9</td>
<td>Configure the axis</td>
<td>57</td>
</tr>
<tr>
<td>9.1</td>
<td>Overview</td>
<td>57</td>
</tr>
<tr>
<td>9.2</td>
<td>Technology object axis</td>
<td>57</td>
</tr>
<tr>
<td>9.3</td>
<td>Axis wizard</td>
<td>57</td>
</tr>
<tr>
<td>9.4</td>
<td>Creating an axis</td>
<td>58</td>
</tr>
<tr>
<td>10</td>
<td>Test the axis with the axis control panel</td>
<td>63</td>
</tr>
<tr>
<td>10.1</td>
<td>Overview</td>
<td>63</td>
</tr>
<tr>
<td>10.2</td>
<td>Axis control panel</td>
<td>63</td>
</tr>
<tr>
<td>10.3</td>
<td>Working with the axis control panel</td>
<td>64</td>
</tr>
<tr>
<td>10.4</td>
<td>Result in the sample project</td>
<td>68</td>
</tr>
<tr>
<td>11</td>
<td>Configure digital outputs</td>
<td>69</td>
</tr>
<tr>
<td>12</td>
<td>Programming the SIMOTION application</td>
<td>71</td>
</tr>
<tr>
<td>12.1</td>
<td>Overview</td>
<td>71</td>
</tr>
<tr>
<td>12.2</td>
<td>Variables</td>
<td>72</td>
</tr>
<tr>
<td>12.2.1</td>
<td>Variable types</td>
<td>72</td>
</tr>
<tr>
<td>12.2.2</td>
<td>Variables of the sample project</td>
<td>73</td>
</tr>
<tr>
<td>12.2.3</td>
<td>Creating global device variables</td>
<td>74</td>
</tr>
<tr>
<td>12.2.4</td>
<td>Creating I/O variables</td>
<td>75</td>
</tr>
<tr>
<td>12.2.5</td>
<td>Back up the configuration</td>
<td>79</td>
</tr>
<tr>
<td>12.3</td>
<td>Programming</td>
<td>80</td>
</tr>
<tr>
<td>12.3.1</td>
<td>Programming languages in the sample project</td>
<td>80</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>12.3.2</td>
<td>MCC Motion Control Chart</td>
<td>80</td>
</tr>
<tr>
<td>12.3.2.1</td>
<td>The programming language MCC</td>
<td>80</td>
</tr>
<tr>
<td>12.3.2.2</td>
<td>Creating the MCC unit</td>
<td>81</td>
</tr>
<tr>
<td>12.3.2.3</td>
<td>Creating the MCC chart</td>
<td>81</td>
</tr>
<tr>
<td>12.3.2.4</td>
<td>Inserting command blocks into an MCC chart</td>
<td>82</td>
</tr>
<tr>
<td>12.4</td>
<td>Expanding the MCC sample program: control of the infeed</td>
<td>98</td>
</tr>
<tr>
<td>12.5</td>
<td>Program flow</td>
<td>98</td>
</tr>
<tr>
<td>12.5.1</td>
<td>System function call _LineModule_control[FB]</td>
<td>99</td>
</tr>
<tr>
<td>12.5.2</td>
<td>Create UNTIL loop</td>
<td>102</td>
</tr>
<tr>
<td>12.5.3</td>
<td>Copy blocks</td>
<td>103</td>
</tr>
<tr>
<td>12.5.4</td>
<td>Adapt the system function call _linemodule_control</td>
<td>104</td>
</tr>
<tr>
<td>12.5.5</td>
<td>Create additional MCC programs for the sample project</td>
<td>105</td>
</tr>
<tr>
<td>12.6</td>
<td>Back up MCC sample programs</td>
<td>106</td>
</tr>
<tr>
<td>12.7</td>
<td>LAD/FBD ladder logic/function block diagram</td>
<td>107</td>
</tr>
<tr>
<td>12.8</td>
<td>The LAD and FBD programming languages</td>
<td>107</td>
</tr>
<tr>
<td>12.8.1</td>
<td>Create LAD/FBD unit</td>
<td>108</td>
</tr>
<tr>
<td>12.8.2</td>
<td>Create LAD/FBD program</td>
<td>109</td>
</tr>
<tr>
<td>12.8.3</td>
<td>Creating a LAD sample program</td>
<td>111</td>
</tr>
<tr>
<td>12.8.4</td>
<td>Back up LAD/FBD sample program</td>
<td>114</td>
</tr>
<tr>
<td>12.9</td>
<td>Other programming languages</td>
<td>115</td>
</tr>
<tr>
<td>12.10</td>
<td>Result in the sample project</td>
<td>116</td>
</tr>
<tr>
<td>13</td>
<td>Configure execution system</td>
<td>117</td>
</tr>
<tr>
<td>13.1</td>
<td>Overview</td>
<td>117</td>
</tr>
<tr>
<td>13.2</td>
<td>Execution system</td>
<td>117</td>
</tr>
<tr>
<td>13.3</td>
<td>Assign programs to tasks</td>
<td>118</td>
</tr>
<tr>
<td>13.4</td>
<td>Result in the sample project</td>
<td>121</td>
</tr>
<tr>
<td>14</td>
<td>Starting and stopping the system</td>
<td>123</td>
</tr>
<tr>
<td>14.1</td>
<td>Overview</td>
<td>123</td>
</tr>
<tr>
<td>14.2</td>
<td>RUN and STOP operating states</td>
<td>124</td>
</tr>
<tr>
<td>14.3</td>
<td>Mode selector switch on the software side and the hardware side</td>
<td>125</td>
</tr>
<tr>
<td>14.4</td>
<td>Start program control of the sample project</td>
<td>127</td>
</tr>
<tr>
<td>15</td>
<td>Monitor the application</td>
<td>129</td>
</tr>
<tr>
<td>15.1</td>
<td>Overview</td>
<td>129</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>15.2</td>
<td>Monitoring program execution</td>
<td>130</td>
</tr>
<tr>
<td>15.3</td>
<td>Monitoring variables</td>
<td>132</td>
</tr>
<tr>
<td>15.4</td>
<td>Recording signals with the trace</td>
<td>133</td>
</tr>
<tr>
<td>15.4.1</td>
<td>Trace</td>
<td>133</td>
</tr>
<tr>
<td>15.4.2</td>
<td>Working with the trace</td>
<td>134</td>
</tr>
<tr>
<td>15.4.3</td>
<td>Further diagnostic functions</td>
<td>138</td>
</tr>
<tr>
<td>15.5</td>
<td>Result in the sample project</td>
<td>139</td>
</tr>
<tr>
<td>A</td>
<td>ESD directives</td>
<td>141</td>
</tr>
<tr>
<td>A.1</td>
<td>ESD definition</td>
<td>141</td>
</tr>
<tr>
<td>A.2</td>
<td>Electrostatic charging of individuals</td>
<td>142</td>
</tr>
<tr>
<td>A.3</td>
<td>Basic measures for protection against discharge of static electricity</td>
<td>143</td>
</tr>
<tr>
<td>Index</td>
<td></td>
<td>145</td>
</tr>
</tbody>
</table>
Safety notes

1.1 Fundamental safety instructions

1.1.1 General safety instructions

<table>
<thead>
<tr>
<th>DANGER</th>
</tr>
</thead>
</table>

**Danger to life due to live parts and other energy sources**

Death or serious injury can result when live parts are touched.
- Only work on electrical devices when you are qualified for this job.
- Always observe the country-specific safety rules.

Generally, six steps apply when establishing safety:
1. Prepare for shutdown and notify all those who will be affected by the procedure.
2. Disconnect the machine from the supply.
   - Switch off the machine.
   - Wait until the discharge time specified on the warning labels has elapsed.
   - Check that it really is in a no-voltage condition, from phase conductor to phase conductor and phase conductor to protective conductor.
   - Check whether the existing auxiliary supply circuits are de-energized.
   - Ensure that the motors cannot move.
3. Identify all other dangerous energy sources, e.g. compressed air, hydraulic systems, or water.
4. Isolate or neutralize all hazardous energy sources by closing switches, grounding or short-circuiting or closing valves, for example.
5. Secure the energy sources against switching on again.
6. Ensure that the correct machine is completely interlocked.

After you have completed the work, restore the operational readiness in the inverse sequence.

<table>
<thead>
<tr>
<th>WARNING</th>
</tr>
</thead>
</table>

**Danger to life from hazardous voltage when connecting an unsuitable power supply**

Touching live components can result in death or severe injury.
- Only use power supplies that provide SELV (Safety Extra Low Voltage) or PELV (Protective Extra Low Voltage) output voltages for all connections and terminals of the electronics modules.
Safety notes

1.1 Fundamental safety instructions

WARNING
Danger to life from touching live parts on damaged devices
Improper handling of devices can result in damage.
For damaged devices, hazardous voltages can be present at the enclosure or at exposed components; if touched, this can result in death or severe injury.
- Observe the limit values specified in the technical specifications during transport, storage, and operation.
- Do not use damaged devices.

WARNING
Danger to life through electric shock due to unconnected cable shields
Hazardous touch voltages can occur through capacitive cross-coupling due to unconnected cable shields.
- As a minimum, connect cable shields and the cores of power cables that are not used (e.g. brake cores) at one end at the grounded housing potential.

WARNING
Danger to life due to electric shock when not grounded
For missing or incorrectly implemented protective conductor connection for devices with protection class I, high voltages can be present at open, exposed parts, which when touched, can result in death or severe injury.
- Ground the device in compliance with the applicable regulations.

WARNING
Danger to life due to fire spreading if housing is inadequate
Fire and smoke development can cause severe personal injury or material damage.
- Install devices without a protective housing in a metal control cabinet (or protect the device by another equivalent measure) in such a way that contact with fire inside and outside the device is prevented.
- Ensure that smoke can only escape via controlled and monitored paths.
1.1 Fundamental safety instructions

**WARNING**

**Danger to life from unexpected movement of machines when using mobile wireless devices or mobile phones**

Using mobile radios or mobile phones with a transmit power > 1 W closer than approx. 2 m to the components may cause the devices to malfunction, influence the functional safety of machines therefore putting people at risk or causing material damage.

- Switch off wireless devices or mobile phones in the immediate vicinity of the components.

**WARNING**

**Danger to life due to fire if overheating occurs because of insufficient ventilation clearances**

Inadequate ventilation clearances can cause overheating of components followed by fire and smoke development. This can cause death or serious injury. This can also result in increased downtime and reduced service life for devices/systems.

- Ensure compliance with the specified minimum clearance as ventilation clearance for the respective component.

**WARNING**

**Danger of an accident occurring due to missing or illegible warning labels**

Missing or illegible warning labels can result in accidents involving death or serious injury.

- Check that the warning labels are complete based on the documentation.
- Attach any missing warning labels to the components, in the national language if necessary.
- Replace illegible warning labels.

**WARNING**

**Danger to life when safety functions are inactive**

Safety functions that are inactive or that have not been adjusted accordingly can cause operational faults on machines that could lead to serious injury or death.

- Observe the information in the appropriate product documentation before commissioning.
- Carry out a safety inspection for functions relevant to safety on the entire system, including all safety-related components.
- Ensure that the safety functions used in your drives and automation tasks are adjusted and activated through appropriate parameterizing.
- Perform a function test.
- Only put your plant into live operation once you have guaranteed that the functions relevant to safety are running correctly.
1.1 Fundamental safety instructions

**Note**

**Important safety notices for safety functions**

If you want to use safety functions, you must observe the safety notices in the safety manuals.

<table>
<thead>
<tr>
<th>WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Danger to life or malfunctions of the machine as a result of incorrect or changed parameterization</strong></td>
</tr>
<tr>
<td>As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.</td>
</tr>
<tr>
<td>• Protect the parameterization (parameter assignments) against unauthorized access.</td>
</tr>
<tr>
<td>• Respond to possible malfunctions by applying suitable measures (e.g. EMERGENCY STOP or EMERGENCY OFF).</td>
</tr>
</tbody>
</table>

1.1.2 Safety instructions for electromagnetic fields (EMF)

<table>
<thead>
<tr>
<th>WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Danger to life from electromagnetic fields</strong></td>
</tr>
<tr>
<td>Electromagnetic fields (EMF) are generated by the operation of electrical power equipment such as transformers, converters or motors.</td>
</tr>
<tr>
<td>People with pacemakers or implants are at a special risk in the immediate vicinity of these devices/systems.</td>
</tr>
<tr>
<td>• Ensure that the persons involved are the necessary distance away (minimum 2 m).</td>
</tr>
</tbody>
</table>
1.1.3 Handling electrostatic sensitive devices (ESD)

Electrostatic sensitive devices (ESD) are individual components, integrated circuits, modules or devices that may be damaged by either electric fields or electrostatic discharge.

**NOTICE**

**Damage through electric fields or electrostatic discharge**

Electric fields or electrostatic discharge can cause malfunctions through damaged individual components, integrated circuits, modules or devices.

- Only pack, store, transport and send electronic components, modules or devices in their original packaging or in other suitable materials, e.g conductive foam rubber of aluminum foil.
- Only touch components, modules and devices when you are grounded by one of the following methods:
  - Wearing an ESD wrist strap
  - Wearing ESD shoes or ESD grounding straps in ESD areas with conductive flooring
- Only place electronic components, modules or devices on conductive surfaces (table with ESD surface, conductive ESD foam, ESD packaging, ESD transport container).

1.1.4 Industrial security

**Note**

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, solutions, machines, equipment and/or networks. They are important components in a holistic industrial security concept. With this in mind, Siemens’ products and solutions undergo continuous development. Siemens recommends strongly that you regularly check for product updates.

For the secure operation of Siemens products and solutions, it is necessary to take suitable preventive action (e.g. cell protection concept) and integrate each component into a holistic, state-of-the-art industrial security concept. Third-party products that may be in use should also be considered. For more information about industrial security, visit http://www.siemens.com/industrialsecurity.

To stay informed about product updates as they occur, sign up for a product-specific newsletter. For more information, visit http://support.automation.siemens.com
WARNING

Danger as a result of unsafe operating states resulting from software manipulation

Software manipulation (e.g. by viruses, Trojan horses, malware, worms) can cause unsafe operating states to develop in your installation which can lead to death, severe injuries and/or material damage.

- Keep the software up to date.
  Information and newsletters can be found at:
  http://support.automation.siemens.com

- Incorporate the automation and drive components into a state-of-the-art, integrated industrial security concept for the installation or machine.
  For more detailed information, go to:
  http://www.siemens.com/industrialsecurity

- Make sure that you include all installed products into the integrated industrial security concept.
1.1.5 **Residual risks of power drive systems**

The control and drive components of a drive system are approved for industrial and commercial use in industrial line supplies. Their use in public line supplies requires a different configuration and/or additional measures.

These components may only be operated in closed housings or in higher-level control cabinets with protective covers that are closed, and when all of the protective devices are enabled.

These components may only be handled by qualified and trained technical personnel who are knowledgeable and observe all of the safety instructions on the components and in the associated technical user documentation.

When assessing the machine's risk in accordance with the respective local regulations (e.g. EC Machinery Directive), the machine manufacturer must take into account the following residual risks emanating from the controller and drive components of a drive system:

1. Unintentional movements of driven machine components during commissioning, operation, maintenance, and repairs caused by, for example:
   - Hardware faults and/or software errors in sensors, controllers, actuators, and connection systems
   - Response times of the controller and drive
   - Operating and/or ambient conditions outside of the specification
   - Condensation / conductive contamination
   - Parameterization, programming, cabling, and installation errors
   - Use of radio devices/cellular phones in the immediate vicinity of the controller
   - External influences / damage

2. In the event of a fault, exceptionally high temperatures, including an open fire, as well as emissions of light, noise, particles, gases, etc. can occur inside and outside the inverter, for example:
   - Component malfunctions
   - Software errors
   - Operating and/or ambient conditions outside of the specification
   - External influences / damage

Inverters of the Open Type / IP20 degree of protection must be installed in a metal control cabinet (or protected by another equivalent measure) such that the contact with fire inside and outside the inverter is not possible.
3. Hazardous touch voltages caused by, for example:
   - Component malfunctions
   - Influence of electrostatic charging
   - Induction of voltages in moving motors
   - Operating and/or ambient conditions outside of the specification
   - Condensation / conductive contamination
   - External influences / damage

4. Electrical, magnetic and electromagnetic fields generated in operation that can pose a risk to people with a pacemaker, implants or metal replacement joints, etc. if they are too close.

5. Release of environmental pollutants or emissions as a result of improper operation of the system and/or failure to dispose of components safely and correctly.

---

**Note**

The components must be protected against conductive contamination (e.g. by installing them in a control cabinet with degree of protection IP54 according to IEC 60529 or NEMA 12).

Assuming that conductive contamination at the installation site can definitely be excluded, a lower degree of cabinet protection may be permitted.

For more information about residual risks of the components in a drive system, see the relevant sections in the technical user documentation.

### 1.2 Specific safety information

Make sure your training system is fully disconnected from productive operation.

Observe the safety notes in the documentation of the devices used.
2.1 Aim of Getting Started

Getting Started introduces you to working with the SIMOTION SCOUT engineering system. You will create a simple sample project and in so doing, you will work through the typical steps involved in configuring devices, drives, and axes. You will become familiar with the most important tools that SIMOTION SCOUT provides for configuring, programming, and diagnostics.

2.2 Sample project

Getting Started provides you with instructions for creating a simple sample project.

Configuring steps

Prepare the configuration
- You reset the SIMOTION device to the factory settings.
- You configure the interface for network communication between the PG/PC and the SIMOTION device.

Create project, configure SIMOTION device and network communication with the PG/PC
- You create a project.
- You create a SIMOTION device and set up the network communication between the PG/PC and the SIMOTION device.

Configure the drive
- You commission the drive.

Configure the infeed
- You interconnect the infeed with the drive.

Configure and test the axis
- You set up an axis.
- You interconnect the axis with the drive.
- You test the axis with the axis control panel.
Configure inputs/outputs

- You configure I/Os for use in the sample program.

Program, set up and monitor SIMOTION

- You write a simple SIMOTION user program that controls the configured axis.
  - You create the variables required by the program.
  - You create the program and additional auxiliary programs with the graphical editors.
- You assign the finished programs to the tasks of the execution system.
- You start program execution in the SIMOTION runtime system.
- You monitor the program-controlled axis movement.
  - You monitor program execution.
  - You monitor the values in the symbol browser.
  - You compile values in a watch table.
  - You record the course of the axis motion with the trace.

Scope of the sample project

The SIMOTION documentation contains two versions of Getting Started with different sample project scopes:

- **Print version of Getting Started**: The present print version deals with configuring a SIMOTION D435-2 device. Auto configuration is used for configuring the drive, and only this is described.

- **Online help version of Getting Started**: The Getting Started you will find in the SIMOTION SCOUT online help is more general in nature. The sample project presented there takes account of the three platforms SIMOTION C, SIMOTION D, and SIMOTION P. The online help version also describes:
  - Configuring the drive with the drive wizard
  - Testing the drive with the drive control panel
  - Configuring a virtual axis. The virtual axis does not require a drive

You can find the general version of Getting Started in the online help under **Getting Started SIMOTION SCOUT**.
**Completed sample project**

The completed sample project is included in the SIMOTION Utilities & Applications. You can find it there under **Examples > Getting Started**.

You can find information on the Utilities & Applications in the following section Utilities & applications (Page 23).

**Following Getting Started**

We recommend that, following Getting Started, you continue to familiarize yourself with SIMOTION SCOUT using the sample projects of the Utilities & Applications.

You can find information on the Utilities & Applications in the following section Utilities & applications (Page 23).
2.3 Preconditions

Training system

To create the sample project, you require a training system with a few components:

- SIMOTION D435-2 device with the latest firmware V4.4
- PG/PC with free Ethernet interface
  A USB Ethernet adapter is also suitable for the Ethernet connection.
- SINAMICS drive with infeed, power unit, and motor for operating an axis
- Full DRIVE-CLiQ wiring of the components; motor with DRIVE-CLiQ interface and thus with automatic encoder identification (SMI Sensor Module Integrated)
- SIMOTION SCOUT Engineering System

A SIMOTION D435-2 DP/PN device is used in the sample project. However, you can use any D4x5-2 device.

The PROFINET interface of the SIMOTION device is not used in the sample project. In the project, reference is made to PROFINET only when creating the SIMOTION device for the first time, in order to fully represent the work sequence.

Note

Getting Started deals with automatic drive configuration and not configuration using the drive wizard. To be able to carry out automatic drive configuration fully, full DRIVE-CLiQ wiring of the components involved is a necessary requirement.

Preparation of the training system

Your training system has been prepared for configuring with SIMOTION SCOUT:

- The hardware is ready installed and wired.
- The CF card with the latest firmware V4.4 is plugged in.
- The PG/PC and SIMOTION D435-2 are connected direct via Ethernet cable. The Ethernet X127 interface on the SIMOTION D435-2 is used for the connection.
- SIMOTION SCOUT has been installed on the PG/PC and correctly licensed.
- You have started SIMOTION SCOUT. The SIMOTION SCOUT Workbench is visible on the screen of the PG/PC.
- No project is open in SIMOTION SCOUT.
2.4 General information

2.4.1 SIMOTION SCOUT online help

SIMOTION SCOUT has a comprehensive online help. This is divided into general help and context-sensitive help.

You open the general help as follows

- Select Help > Help topics in the menu, or
- press the F1 key.

You open the context-sensitive help as follows

- Click the Help button of the dialog or window, or
- press the key combination Shift+F1, or
- click in the toolbar on the Help button.

With the mouse pointer (changed to a question mark), click the parameter or the window for which you require help.

You can find detailed information on using the context-sensitive help in the online help under SIMOTION SCOUT > General > Use Help.

Full text search in the online help

You can find important information on full text searching in the online help under Basic > Use Online Help and Function Block Diagrams > Use Online Help > Full Text Search.

Note

Full text search in the online help

Enclose the search term between asterisks * to include search results that contain the search term as part of a longer character string, e.g.

- variable finds only the whole words "variable" or "Variable",
- *variable* also finds "system variables", "variable assignment", etc.
2.4.2 Available documentation

Electronic documentation

The SIMOTION documentation is included in electronic form in the scope of delivery of SIMOTION SCOUT (SIMOTION SCOUT DVD Documentation, Utilities & Applications). You can search for all PDF documents in the electronic documentation with an index (SIMOTION.pdx). You can find an overview of the structure and content of the SIMOTION PDF documentation in the separate document Overview of the SIMOTION Documentation.

The content of the documents is also available in the SIMOTION SCOUT online help, with a few exceptions.

In the online help, you can find the Overview of the SIMOTION Documentation under Overview of the SIMOTION Documentation > Overview of the SIMOTION Documentation

Overview of SIMOTION

You can find a brief system overview of SIMOTION in the online help under Basic > Basic Functions > System Overview.

The manuals contain comprehensive information especially for configuring and commissioning a SIMOTION D, see the SIMOTION D4x5-2 Manual as well as the SIMOTION D4x5-2 Commissioning and Hardware Installation Manual. You can also find the SIMOTION D manuals in the online help under SIMOTION Devices > SIMOTION D.

2.4.3 Project generator

With SIMOTION easyProject, basic functions required in just about every SIMOTION application can be quickly and conveniently integrated into an existing project or a new project. The desired components and functions are selected and configured here via dialogs. There is thus no need for time-consuming programming, and a uniform and standardized project configuration is guaranteed while simultaneously reducing potential errors. As of SIMOTION SCOUT V4.4, the project generator can be started direct from SIMOTION SCOUT.

The project generator is not used in Getting Started.
2.4.4 Utilities & applications

The free SIMOTION Utilities & Applications provide you with a wealth of important background information on all aspects of SIMOTION, tools, special functions, blocks, SIMOTION sample projects, as well as off-the-shelf standard applications for illustration or for use in your projects. There you can also find detailed information on scripting and a host of sample scripts that facilitate working with SIMOTION.

1. FAQs
   Interesting FAQs such as control of hydraulic axes or communication issues.

2. Scripts
   A host of scripts, helpful tips, but also extensive solutions that make recurring tasks easier, for example.

3. Tools and documentation
   You are provided with easy-to-use tools and in-depth documentation for many tasks.

4. Examples
   Sample projects for first-time-users, e.g. "Getting Started", as well as examples of special topics.

5. Applications
   For SIMOTION, there is a host of applications available to you that provide you with a sound basic framework. With the aid of the supplied documentation, you can use the applications as a basis for your own application, and adapt and expand them.

   Furthermore, functions are available to you under "Cross-Sector applications" that are of general help when creating your own applications, e.g. a LDPV1 library for drive communication via DPV1 services (read and write SINAMICS parameters, read errors and warnings from SINAMICS, deactivate objects, ramp-up coordination and much more), or the LCom library with functions for communication with TCP/IP for SIMOTION and SIMATIC.

   The new SIMOTION easyProject project generator significantly speeds up the creation of a standardized project basis for machine applications, and therefore reduces costs.

6. SIMOTION IT
   Under SIMOTION IT, you can find innovative examples and tools for bringing control and drive solutions a little closer to the IT world. These include a trace function that is executed via an Internet browser, as well as examples for user-defined Web pages for the SIMOTION IT Web server or for using OPC XML DA.
Prepare the configuration

3.1 Restore factory settings

For the sample project, it is useful to reset the SIMOTION device to the factory settings.

In this way, you restore the default communications parameters and you delete user data installed on the device and the CF card by a previous configuration. The runtime licenses are retained.

You restore the factory settings on the SIMOTION device as follows

1. Switch off the power supply of the SIMOTION device.
2. Set the mode selector switch ③ of the SIMOTION device to position 3.

![Diagram of SIMOTION D435-2 module front]

- LED display
- 7-segment display
- Mode selector switch in position 3

Figure 3-1 SIMOTION D435-2 module front

**NOTICE**

Damage from electrostatic discharge

The rotary switch can be destroyed by static electricity.

Operate the rotary switch only with an insulated screwdriver.

Observe the ESD regulations.
3. Switch on the power supply of the SIMOTION device.

The default settings are loaded. SIMOTION D435-2 switches to STOP mode.

Wait for the procedure to finish. The elements on the front of the module indicate completion:

- The 7-segment display ② shows status digit 6: SIMOTION D435-2 has started up.
- LEDs ①:
  - LED RDY flashes green (0.5 Hz): the drive is ready for commissioning.
  - LED STOP shows a yellow light: the SIMOTION device is in STOP mode.
  - All other LEDs are off.

4. Turn the mode selector switch ③ to position 0.

- The LED RUN shows a green light: the SIMOTION device is in RUN mode.

Result

The SIMOTION device has been restored to its factory settings and is ready for commissioning.
3.2 **Set up interface for online communication**

Online communication of the PG/PC with the SIMOTION device can be set up via PROFIBUS, PROFINET, or Industrial Ethernet. The sample project is restricted to the most frequent application case: communication via Industrial Ethernet.

Configuration of the Ethernet interface encompasses two steps:

- You insert the Ethernet address of the PG/PC into the default subnet of the SIMOTION device.
- You assign the Ethernet address to an access point in SIMOTION SCOUT.

**Preconditions**

- The PG/PC and the SIMOTION device are connected via an Ethernet cable.
- The X127 PN/IE interface on the SIMOTION device is used for the Ethernet connection.

---

**Ethernet interface X127 PN/IE**

![SIMOTION D435-2 module front](image)

- The Ethernet interface X127 PN/IE has the default address:
  - IP address: 169.254.11.22 - Subnet: 255.255.0.0
3.2.1 Prepare the Ethernet interface of the PG/PC

The PG/PC must be located in the same subnet as the SIMOTION device. The subnet is specified by the factory settings of the SIMOTION device.

You insert the PG/PC into the subnet of the SIMOTION device as follows:

1. In the Windows Control Panel of the PG/PC, open the Properties window of the network connection used, and then open the dialog Properties of Internet protocol Version 4 (TCP/IPv4).

   ![Image of Network Properties](Figure 3-3 IP address of the PG/PC in subnet 169.254)

2. Assign the following address:
   
   IP address: **169.254.11.99** - Subnet: **255.255.0.0**

3. Confirm with **OK**. Close the dialog.
3.2.2 Define the access point of the SIMOTION device

SIMOTION SCOUT has two access points for communication with controllers and individual drives:

- S7ONLINE
- DEVICE

In the sample project, the S7ONLINE access point is used for communication with the PG/PC.

You define the access point as follows

1. Go to SIMOTION SCOUT.
2. Select **Options > Set PG/PC interface** in the menu bar.
   
   The **Set PG/PC interface** dialog opens.
3. In the field **Access point of the application**, select the access point **S7ONLINE (STEP7)**.
   
   The interface currently still assigned to the access point is shown after the arrow pointing to the right.
4. Select the prepared Ethernet interface in the list **Interface programming used**.

   ![Set PG/PC interface](image)

   Figure 3-4  Set PG/PC interface

5. Confirm with **OK**.
   
   The S7ONLINE access point is activated.
   
   The Ethernet interface is assigned to the S7ONLINE access point.
3.3 Result in the sample project

The factory settings of the SIMOTION device have been restored.

The Ethernet interface has been prepared and can be used for configuring online communication between the PG/PC and the SIMOTION device.
Create a project

4.1 Overview

Aim of Getting Started

In this part of Getting Started, you create the sample project Sample_1. All of the subsequent configuring steps refer to this sample project.

4.2 Project

A project contains all the information that describes a machine and its function: configuration data, programs, motion profiles, drive data.

A project can contain several SIMOTION devices.

4.3 Create new project

You create a new project as follows

You start a new configuration in SIMOTION SCOUT by creating a new project.

1. Select Project > New from the menu bar. The New Project dialog is displayed.
2. Enter the project name under Name, e.g. Sample_1.
3. Under Storage location (path), enter the path where you wish to store the project. The default path is already set.
4. Acknowledge with OK.

The dialog closes.

Additional information about creating a new project

Default path of the project

The default path depends on the operating system:

- Windows XP
  
  C:\Program Files\Siemens\Step7\s7proj

- Windows 7
  
  C:\Program Files (x86)\Siemens\Step7\s7proj
**Project name and project directory name**

The name of a SIMOTION SCOUT project can contain not more than 24 characters. The project appears under the full name in the dialogs.

When initially saving the project, SIMOTION SCOUT creates the directory name from the first 8 characters of the project name. SIMOTION SCOUT uses a numerical counter "_1", "_2", … to resolve conflicting names resulting from the abbreviation of the 8 characters. The counter replaces the last characters of the directory name.

**Note**

It is useful to select project names in such a way that they can be uniquely distinguished by their first 8 characters. The project name and the directory name derived from it thus uniquely identify the same project.
4.4 Result in the sample project

The sample project of Getting Started has been created in SIMOTION SCOUT. The project folder Sample_1 is visible in the project navigator.

Figure 4-1 SIMOTION SCOUT Workbench, newly created sample project Sample_1
Create a project

4.4 Result in the sample project
Create SIMOTION device and configure online communication

5.1 Overview

Aim of Getting Started

This part of Getting Started shows you how to create a SIMOTION device in the project and how to set up communication between the PG/PC and the SIMOTION device.

5.2 SIMOTION D platform

SIMOTION D is the drive-based version of the SIMOTION motion control system, based on the SINAMICS family of drives. With SIMOTION D, the SIMOTION motion control functionalities and the SINAMICS drive software run on a SINAMICS-type closed-loop control hardware device. SIMOTION D devices have the following characteristic features:

- Motion control functionality and control functionality integrated direct in the drive
- Suitable for modular machine concepts with fast isochronous coupling

On SIMOTION D, the SINAMICS functionality of the closed-loop-control module of the SINAMICS S120 multi-axis drive system is integrated (SINAMICS Integrated).

You will find a brief introduction to SIMOTION D in the online help using the example of the D435 Control Unit. Click Help > Tutorials > SIMOTION Drive-Based in the SIMOTION SCOUT menu bar.
5.3 Create SIMOTION device

5.3.1 Steps

Creating a new SIMOTION device in the project involves three steps. SIMOTION SCOUT combines the steps into one coherent process:

1. Create SIMOTION device.
2. Configure PROFINET interface of the device.
   
   This step is omitted in the sample project. PROFINET is not used. However, SIMOTION SCOUT opens the configuration dialog if the PROFINET-supporting device version D4x5-2 DP/PN is installed in your training system.
3. Set up communication between the PG/PC and the SIMOTION device.

The newly created SIMOTION device appears in the project tree.
5.3.2 Create SIMOTION device

You create a SIMOTION D435-2 device in the project as follows:

1. In the project navigator, double-click **Insert SIMOTION device**.

   The **Insert SIMOTION device** dialog appears.

   ![Insert SIMOTION device dialog](image)

   Figure 5-1 Select SIMOTION device

2. In the **Device** list field, the **SIMOTION D** platform is already pre-selected.
3. In the **Device version** list, select the SIMOTION device **D435-2 DP/PN**, and under **SIMOTION version**, select the firmware version of the device used.

**Note**

The selected firmware version must match the firmware version on the memory card of the SIMOTION device. Otherwise, you will receive an error message when going online with the device.

4. With the checkbox **Open HW Config**, you define whether the window for hardware configuration **HW Config** is to be opened after the new device has been created.

   Leave the checkbox activated.

5. Acknowledge with **OK**.

   The **Insert SIMOTION device** dialog is closed. SIMOTION SCOUT takes you to the next step Configure the PROFINET interface (Page 38).

### 5.3.3 Configure the PROFINET interface

If the SIMOTION device has a PROFINET IO interface, the dialog **Properties – Ethernet interface PNxIO** appears. The dialog enables integration of the SIMOTION device into an existing PROFINET IO subnet. If no subnet is known, you can create it here.

PROFINET is not used in the sample project. Click **Cancel**. The dialog closes. SIMOTION SCOUT takes you to the next step Set up PG/PC communication (Page 39).

Figure 5-2 Dialog for selecting an Ethernet interface
5.3.4 Set up PG/PC communication

Interface Selection dialog

SIMOTION SCOUT opens the Interface Selection dialog. You configure the network communication between the PG/PC and the SIMOTION device in this dialog.

Figure 5-3  Set up PG/PC communication

Note
If a PG/PC is already available in the project and it is connected with a subnet, SIMOTION SCOUT automatically establishes the connection between the PG/PC and the SIMOTION device. In this case, the Interface Selection dialog does not appear.

The list field Interface selection for PG/PC connection in the upper half of the dialog provides the list of SIMOTION device interfaces for selection.

The lower field Interface parameterizations in the PG/PC lists the interfaces of the PG/PC. The upper field functions as a filter so that only the interfaces with suitable transmission protocol are available for selection.

You set up data communication between the SIMOTION device and the PG/PC as follows

1. Select the Ethernet interface Ethernet PNxlE (X127) for the SIMOTION D435-2 device.
2. Then select the prepared Ethernet interface of the PG/PC.
3. Confirm the configuration with OK.

The Interface Selection dialog is closed.
5.4 Result in the sample project

Newly created SIMOTION device

Configuring of the SIMOTION device and network communication between the PG/PC and the SIMOTION device is complete.

- The newly created SIMOTION D435-2 device is shown in the project tree of the workbench.
- The PG/PC is connected with the SIMOTION device via Ethernet.
- SIMOTION SCOUT automatically opens the HW Config window and thus leads you to hardware configuration.

HW Config

HW Config shows the newly created SIMOTION device, the integrated SINAMICS drive unit (SINAMICS_Integrated), and the integrated PROFIBUS (DP-Integrated).

![HW Config](image)

Figure 5-4 HW Config, SIMOTION D device with SINAMICS_Integrated

No other hardware configuration is required for the sample project. You can close HW Config. In the menu bar of HW Config, select the command Station > Exit.
6.1 Overview

Aim of Getting Started

In this configuring step, you create the prerequisites for configuring the drive.

- You back up the created sample project to the hard disk.
- You compile the project into executable code.
- You establish online communication with the SIMOTION device.
- You download the project from the PG/PC to the SIMOTION device.

6.2 Save and compile the project

To be able to download a project created in SIMOTION SCOUT to the target system, the project must be saved in executable code.

The command Save project and compile changes combines both steps. The project is backed up to the hard disk. SIMOTION SCOUT searches the entire project for changes and compiles only the changes.

Note

Use the command Save project and compile changes for preference for day-to-day work.

Several variations of the "Save" command are available for selection under the menu title Project. You can find information on this in the online help under Save and compile.

You save and compile a project as follows

Select the menu command Project > Save and compile changes or click the relevant button in the toolbar.

SIMOTION SCOUT shows progress indicators for symbolic assignment and for compilation. The compilation run is logged in the detail view of the workbench. Information, warning and compilation errors are shown there in plain text.

Switch on detail view

The detail view might be switched off. Click the menu item View > Detail view to activate the view.
6.3 Connect to selected target devices – Go online

To download project data from SIMOTION SCOUT to the hardware, or to transfer machine data in the other direction from the hardware to the SIMOTION SCOUT project, communication between the PG/PC and the SIMOTION device must be activated.

The status of the network communication is displayed in the footer of the workbench:

- **Online mode**: Network communication is switched on. Data can be exchanged.
- **Offline mode**: The network connection is switched off.
You go online as follows

1. Select the menu command **Project > Connect to selected target devices** or click the relevant button in the toolbar.

   ![Target Device Selection dialog](image)

   When the command is first called, SCOUT opens the **Target Device Selection** dialog. The dialog enables individual selection of the devices to which SIMOTION SCOUT is to connect.

2. In the dialog, select the configured SIMOTION device **D435** and the integrated drive **SINAMICS_Integrated**.

   ![Figure 6-1 Target Device Selection dialog](image)
3. Click **OK**.

   SIMOTION SCOUT establishes the online connection.

4. When first going online, SIMOTION SCOUT reports that access to the drive is not possible.

   Close the message. Track the consistency check to completion on the **Target system output** tab in the detail area.

   In the project tree, the connector symbols on the SIMOTION device and on the integrated SINAMICS drive change. Only the connector symbol of the SIMOTION device is partially colored in green. Green indicates the existing online connection.

   ![Project tree after first going online](image)

   **Device is in ONLINE mode.** There is an inconsistency between the project data of the device in the PG/PC and the current values of the device in the target system.

   **Element is not in ONLINE mode.**

   **Figure 6-2** Project tree after first going online

---

**No connection to the drive when first going online**

To allow SIMOTION SCOUT to connect with the drive, the project must be downloaded to the SIMOTION device. A project download has not yet been carried out in the sample project. You therefore receive the corresponding error messages.
6.4 Download the project to the target system

A complete project download is only possible in STOP mode. If required, SIMOTION SCOUT offers a change of operating mode during the download procedure.

You download the project as follows

1. Select Project > Download to target system in the menu, or click the Download project to target system button in the toolbar.

   The dialog Download to target system appears.

2. Activate the checkbox After loading, copy RAM to ROM. This saves the RAM of the SIMOTION device to the memory card (ROM) of the SIMOTION device. In this way, the configuration is retained after the power supply has been switched off and on again.

3. Start the download operation with Yes.

   The project data and the data of the hardware configuration are downloaded to the RAM of the target system.

   If you are asked whether the CPU is to be switched to STOP, confirm with Yes.

SIMOTION SCOUT carries out numerous checks that are logged on the Target system output tab in the detail area. You will find there the concluding entry Download to target system completed successfully.

---

**Note**

The first download to the target system also downloads the data of the technology package. This operation can take several minutes.

---

**Note**

Depending on the firmware version on the CF card and on the SINAMICS components (DRIVE-CLiQ components such as Line Module, Motor Modules, Terminal Modules, etc.), the firmware of the components is automatically upgraded or downgraded.

The update can take several minutes and its progress is tracked by corresponding messages appearing in the output window of SIMOTION SCOUT.

A firmware update on DRIVE-CLiQ components is signaled by red-green flashing of the RDY LED:

- FW update in progress: RDY LED flashes slowly (0.5 Hz)
- FW update finished: RDY LED flashes quickly (2 Hz), POWER ON required

These flashing patterns are displayed additionally via a yellow RDY LED on the SIMOTION D, indicating that components connected to SIMOTION D are carrying out a firmware update or that all components have completed their firmware update.

Components requiring POWER ON following a firmware update signal this by means of the fast flashing RDY LED. Go offline with SIMOTION SCOUT and switch the 24 V supply to the relevant components off/on (POWER ON) to initialize.
Automatically established connection to the drive

SIMOTION SCOUT automatically establishes the online connection to the drive immediately following the project download.

In the project tree, the connector symbol changes on the SIMOTION device, the integrated SINAMICS drive, and the SINAMICS Control Unit. The connector symbol is entirely green, indicating that the project data in the PG/PC is identical with the project data in the target system.

Green connector symbol: Element is in online mode. The project data in the PG/PC is identical with the project data saved in the target system.

Figure 6-3   Project tree

From this point, you can access the drive online with the PG/PC.
7

Configure the drive

7.1 Overview

Aim of Getting Started

In this part of Getting Started, you configure the integrated drive of the SIMOTION D435-2 device. You use auto-configuration for this purpose.

Preconditions

- You have downloaded the sample project to the target system, refer to the section Download the project to the target system (Page 41).
- SIMOTION SCOUT is online with the SIMOTION device and the integrated drive (green connector symbols).
- The SINAMICS drive components Infeed and Power unit, as well as Motor and Encoder, are connected to the SIMOTION Control Unit via DRIVE-CLiQ. This requirement for auto-configuration was referred to at the start of Getting Started, see the section Preconditions (Page 20).

7.2 Drive

The speed and current for controlling the motor are regulated in the drive.

7.3 Automatic drive configuration

SIMOTION SCOUT can read out the electronic type plates of the SINAMICS drive components via the DRIVE-CLiQ interface and can use this data to configure the drive automatically. Manual data entry is thus dispensed with. The sample configuration of a SIMOTION D435-2 device shown here requires full DRIVE-CLiQ wiring.

As an alternative to automatic drive configuration, you can also configure a D4x5-2 device offline. Offline configuring is shown in the online help version of Getting Started for SIMOTION C and SIMOTION P.
7.4 Automatic configuration of the drive

You open automatic configuration as follows:

Double-click on **Automatic Configuration** in the project navigator under the drive SINAMICS_Integrated.

The **Automatic Configuration** dialog is displayed.

Figure 7-1 Call Project tree, Automatic Configuration

**Note**

In the project tree, the element **Automatic Configuration** is only visible if you have downloaded the project to the target system, and SIMOTION SCOUT is online with the SIMOTION device. The project must be consistent; see the section Download the project to the target system (Page 41).
You cause the drive to be configured automatically as follows

1. Click the **Configure** button in the **Automatic Configuration** dialog.
2. Confirm the prompt regarding restoring the factory settings with **Yes**.
   
   The confirmation prompt appears if the drive unit is not in the “First commissioning” state.

3. In the **Automatic Commissioning** dialog, you can specify whether you are using a drive object of the type servo or vector.

   Select **Servo**.

4. Click the **Create** button in the **Automatic Commissioning** dialog.

   Automatic configuring is started.

**Note**

**Firmware update**

If the firmware version on the DRIVE-CLIQ components is different to that on the CF card, a firmware update is performed automatically at this position.

In this case, proceed as follows:

- Wait for the procedure to finish. This can take several minutes.
- Go offline.
- Switch the power supply to the SIMOTION device off and then on again.

5. If a firmware update has not been performed, remain online to be able to see the changes in the project tree.
6. Open the **Infeeds** system folder in the project tree under the integrated drive.
   - If you are using an infeed with DRIVE-CLiQ interface, the auto configuration has created the infeed there.
   - If you are using an infeed without DRIVE-CLiQ interface, the folder is empty. The infeed is not known in the project.
   The other configuring requirements resulting in this way are dealt with in the next section Configure the infeed (Page 53).

7. Open the **Drives** system folder in the project tree under the integrated drive. The folder contains the drive detected by the auto configuration.

8. Go offline.
   Click in the toolbar on the **Disconnect from target system** button.

   ![Disconnect from target system button](image)

   In the footer of the workbench, **Offline mode** is indicated. Network communication between the PG/PC and the SIMOTION device has been cleared down.
Change the name of the drive

The automatic configuration also assigns the object name of the drive; in this example, SERVO_03.

You can change the name later. Changing is only possible in offline mode.

Proceed as follows:

1. Open the context menu of the drive by right-clicking. Select the Rename command there.
2. Assign the new name. Then confirm with OK.

As with all changes carried out in the project offline, the name change makes the project inconsistent. To restore the consistency between the "Project on the PG/PC" and the "Project on the SIMOTION device", another project download is necessary. The procedure is described in the section Download the project to the target system (Page 41).

7.5 Result in the sample project

If you are using an infeed with DRIVE-CLiQ interface, the drive is ready for operation. It can be interconnected with an axis.
Configure the drive

7.5 Result in the sample project
Configure the infeed

8.1 Overview

Aim of Getting Started

In this part of Getting Started, you configure the infeed.

The integrated SINAMICS Control Unit only starts the drive when the infeed is ready. The project must therefore know the interface via which the drive receives the ready signal of the infeed.

Two cases must be distinguished here:

- Infeed with DRIVE-CLiQ interface
  
  If an infeed with DRIVE-CLiQ connection has already been created, the ready signal of the infeed (r0863.0) is automatically interconnected with "Infeed operation, p0864" of the drive when drives are inserted (only applies to drives that are attached to the same drive unit as the infeed).

  You can continue immediately with the next configuring step, see the section Configure the axis (Page 57).

- Infeed without DRIVE-CLiQ interface
  
  If you are using an infeed without a DRIVE-CLiQ interface, e.g. a Smart Line Module, you must wire the ready signal of the infeed via terminals. The following section describes the procedure.

Preconditions

- You have configured the integrated drive of the SIMOTION D435-2 device, see the section Configure the drive (Page 47).

- SIMOTION SCOUT is in offline mode.
8.2 Configuring an infeed without DRIVE-CLiQ interface

An infeed without DRIVE-CLiQ interface provides the ready signal (p0863.0) via an output terminal. In the project, you specify the input (r0722) of the integrated SINAMICS Control Unit at which the signal is active. The drive supplied by the infeed uses the signal as a ready signal (p0864).

You interconnect the ready signal of the infeed as follows:

In the sample project, the ready signal of the infeed (terminal "DO: Ready" of the infeed) is wired to the DI 0 of the D435-2.

1. Open the configuration dialog of the drive. To do so, double-click Configuration in the project tree below the drive.

2. Click in the working area on the Configure DDS button with the yellow background.

![Configure DDS](image)

The drive wizard opens.

3. Click Next in the drive wizard until you reach the dialog Configuration - SINAMICSIntegrated - Power Unit BICO Technology.
4. In the **Power unit BICO** dialog, input field **p0864**, select the digital input to which the ready signal of the infeed is wired (e.g. DI 0).

![BICO interconnection in the drive wizard](image)

Figure 8-2  **BICO interconnection in the drive wizard**

5. Click **Next**. Run through all the other dialogs without change until the final **Summary** dialog.

6. Click **Finish**.

   The configuration is thus completed.

7. The configuration dialog of the drive is no longer required. Click **Close** at the bottom right of the dialog box.
Configure the infeed

8.2 Configuring an infeed without DRIVE-CLiQ interface
Configure the axis

9.1 Overview

Aim of Getting Started

This part of Getting Started shows you how to create and configure an axis in the project, and how to interconnect it with the integrated drive of the SIMOTION D device. SIMOTION SCOUT provides the axis wizard for this purpose.

Preconditions

You have configured the integrated drive of the SIMOTION D device, see the section Configure the drive (Page 47).

9.2 Technology object axis

Technology objects represent the respective real objects (e.g. a position axis) in the controller.

The technology object axis offers the user a technological view of the drive and the encoder (actuator and encoder), provides technological functions for this, and conceals the actual hardware interface.

The technology object axis contains extensive functionality, e.g. communication with the drive, actual value processing, position control, and positioning functionality. It executes control and motion commands and indicates states and actual values.

Technological limitations and mechanical values of the axis and encoder (e.g. leadscrew pitch and gears) are set on the axis. You can then work exclusively with technological variables.

9.3 Axis wizard

Creation of new axes in the project

SIMOTION SCOUT provides the axis wizard for creating a new axis. The wizard scans the basic settings and interconnects the TO axis with a drive.

The wizard can only be run through once. Later changes to the configuration are possible in the corresponding dialogs of the TO.
9.4 Creating an axis

For the sample project, create the axis \textbf{Axis\_2}. Assign the axis to the drive you have configured in the section Configure the drive (Page 47).

Run through the wizard and confirm all standard settings with \textbf{Next}.

\textbf{Note}

\textbf{Axis\_2}

In the online help version of Getting Started, 2 axes are created, a virtual axis \textbf{Axis\_1} and a real axis \textbf{Axis\_2}. In agreement with the designation there for the real axis, the designation \textbf{Axis\_2} is used here.

You can find Getting Started in the online help under \textit{Getting Started SIMOTION SCOUT}.

You create a real axis in the project as follows

1. Open the \textbf{AXES} folder in the project navigator.
2. Double-click \textbf{Insert axis}. The \textbf{Insert axis} dialog appears.
3. Name the axis in the \textbf{Name} field. Use the designation \textbf{Axis\_2} for the axis of the sample project.
4. Leave the preset axis technology at the default \textbf{Positioning}.

![Axis wizard, create real axis](image)

Figure 9-1  Axis wizard, create real axis
5. Click **OK**.
   The axis configuration wizard appears.

6. Define the axis type.
   For the sample project, select preferably a linear axis with an interconnection to an electrical drive.
   Activate the fields **linear** and **electrical**.

![Axis configuration wizard, determine axis type](image)

7. Click **Next**.
   The **Drive assignment** dialog appears.
8. Assign the axis to the drive you have configured in the section Configure the drive (Page 47).

To do so, click the drive unit in the **Assignment partner** column. Click in the tree under the drive object on the drive object you want to interconnect. **Assign** appears in the **Assignment** column. The axis/drive assignment is thus defined.

![Axis configuration - Axis 2 - Drive assignment](image)

**Figure 9-3**  Axis wizard, assign drive

Click **Next**.
Configure the axis

9.4 Creating an axis

---

**Note**

As an alternative to assigning the axis to an already configured drive, the axis wizard offers two further selection options:

- **Define the axis/drive assignment later:**
  The axis is to be created and not assigned to a drive until later. Programming and simulation of the axis are also possible here.

- **Create drive:**
  The drive wizard can be called up from the axis wizard (offline configuring). The axis can thus be created in one step along with the drive, and assigned to the drive.

The alternative procedures are not considered further in Getting Started.

---

9. Assign the motor encoder.

   In the **Encoder assignment** window, you use the motor encoder connected to the drive as the standard setting.

   Click **Next**.

10. The axis wizard then assembles the configuration data in an overview. You thus have an opportunity to check and correct the data before accepting it into the project.

   Close the dialog with **Finish**.

   The configured axis is displayed in the project navigator.

---

**Automatic settings of the engineering system**

The engineering system automatically defines the PROFldrive axis telegrams required for communication, as well as the addresses used.

In the same way, telegrams are extended and interconnections automatically created in the drive, depending on the selected TO technology (e.g. SINAMICS Safety Integrated).

Drive and encoder data, as well as reference variables, maximum variables, torque limits, and granularity in torque reduction of the SINAMICS S120 are accepted automatically for the configuration of the SIMOTION technology objects “TO axis” and “TO external encoder”.

This data no longer has to be entered in SIMOTION.
Further axis configuration

Further axis configuration is possible via dialogs that can be accessed in the project navigator under the axis. No other configuration is required for the sample project.

Download the axis configuration to the target system

- Save and compile the project.
- Download the sample project with the axis configuration to the target system to be able to test the functioning of the axis in the next configuration step.

See also

Download the project to the target system (Page 41)
10.1 Overview

Aim of Getting Started

In this part of Getting Started, you test the configured axis. SIMOTION SCOUT provides you with the axis control panel for this purpose.

Preconditions

- You have configured the infeed, see the section Configure the infeed (Page 53).
- You have created and fully configured an axis in the sample project, see the section Configure the axis (Page 57).
- The project with the axis configuration has been downloaded to the target system, see the section Download the project to the target system (Page 41).
- SIMOTION SCOUT is in online mode.

10.2 Axis control panel

Control and monitoring of individual axes

The axis control panel is used for controlling and monitoring individual axes. You can use it to traverse axes along with the drive.

You can use the control panel for the following tasks, for example:

- Test each part of the system individually before program-driven axis motions are initiated.
- In the event of an error, test whether the individual axes and drives can be traversed from the control panel.
- Moving the axes for optimization purposes (controller optimization).
- Carry out homing.
- Setting and revoking an axis enable.
10.3 Working with the axis control panel

In the sample project, you traverse the set-up axis in jog mode to test the correct functioning of the axis.

You open the axis control panel as follows

Open the AXES folder in the project navigator. Double-click below the axis on Control panel.
The axis control panel is opened in the detail view.

Reference is made to the circled digits in the text below.

Figure 10-2  Axis control panel
Test the axis with the axis control panel
10.3 Working with the axis control panel

You traverse an axis with the axis control panel as follows

1. The field at top left of the control panel ① shows the currently selected axis.

2. Showing areas in the control panel:
   The control panel is divided into a control area and a diagnostics area. The areas may be hidden.
   Click the Show/hide control area button ② and/or the Show/hide diagnostics area button ③ to show the respective area.

3. Assume control priority – Observe safety regulations:
   To be able to traverse the axis from the PG/PC, the PG/PC must obtain control priority.
   Click the Assume Control Priority button ④.
   The Assume Control Priority dialog opens.

   ![Assume Control Priority Dialog](image)

   Figure 10-3  Axis control panel, assume control priority

   Observe the safety regulations and confirm with Accept.
   The PG/PC now has control priority.

Note
You can stop the axis at any time by pressing the spacebar.
4. Enable the axis:

   Click the **Set/remove enables** button ⑤ to enable the axis.

   Confirm the **Switch axis enable** dialog that appears with **OK**.

   The switches **Start motion**, **Stop motion** and **Jog** ⑦ are enabled.

5. Homing the axis:

   Click the **Home axis** button.

   In the dialog box, select the **Homing type Set home position**, and enter the value 0 under **Home position coordinates**.

   Confirm the dialog box with **OK** and start homing the axis.

6. Click the **Position-controlled traversing of the axis** button ⑥.

   The **Start axis position-controlled** dialog appears.

   Specify the velocity of the axis. Click **OK** to close the dialog.

   The velocity appears in the field \( v= \) ⑨.

7. Start the axis motion:

   Click **Jog** ⑦.

   The axis is traversed while you press the switch. In the fields **Velocity** and **Position** ⑧, you can monitor the traverse motion.

   The axis test is thus executed.

8. Deactivate axis enable:

   Click **Set/remove enables** ⑤.

   Confirm the **Disable axis** dialog with **OK**.

   **Note**

   If you want to traverse the axis again, acknowledge all alarms in the "Alarms" window.

9. Return control priority:

   Click **Give up control priority** ④ to deactivate control of the axes from the PG/PC. In this operating state, the axes can no longer be controlled from the PG/PC.

10. Go offline:

   Select **Project > Disconnect from target system** in the menu. Or click in the toolbar on the **Disconnect from target system** button.
10.4 Result in the sample project

You have traversed the axis of the sample project with the axis control panel and thus ensured its correct functioning. Configuring the axis is thus completed.
Configure digital outputs

Aim of Getting Started

In this part of Getting Started, you configure two of the digital inputs/outputs available on the SIMOTION D device as outputs.

You require the outputs for the program you will write in the subsequent configuring step, see the section Programming the SIMOTION application (Page 71).

Preconditions

SIMOTION SCOUT is in offline mode.

I/O channels of the terminal X142

You use the I/O channels of the terminal X142 for the sample project.

![Image of SIMOTION D445-2 DP/PN, position of the digital interface X142]

Figure 11-1 SIMOTION D445-2 DP/PN, position of the digital interface X142

The I/Os of the terminal X142 are permanently assigned to SIMOTION D4x5-2. The terminal is thus visible in the project tree under the SIMOTION device.
Configure digital outputs

You define the digital outputs as follows

1. Double-click the element Inputs/Outupts X142 in the project tree below the SIMOTION device.

   The I/O properties dialog opens.

2. Go to the Channels 0-7 tab.

   The terminal X142 is represented on the tab, specifying the pin numbers and the current use of the channels.

3. Channels 0 and 1 on pins 3 and 4 are preset at the factory as digital inputs DI 0 and DI 1. Define the two channels as digital outputs DO 0 and DO 1. Select the value DO in each case in the Function list field.

   ![I/O properties](image)

   Figure 11-2   I/O properties

4. Confirm with OK.

   Configuring the digital outputs is thus completed.
Programming the SIMOTION application

12.1 Overview

Aim of Getting Started
In this section, you write a simple SIMOTION user program for the sample project.
- You create the variables required by the program.
- You create the program and additional auxiliary programs with the graphical editors supplied by SIMOTION SCOUT.

Purpose of the program
The program controls the axis set up in the sample project.
- The infeed is switched on via a SINAMICS system call.
- The axis is traversed to a specific position and returned to the original position.
- The program states "Program started" and "Program end reached" are applied to digital outputs to evaluate them in the I/O.

Preconditions
- You have configured the infeed, see the section Configure the infeed (Page 53).
- You have configured an axis and tested its function, see the sections Configure the axis (Page 57) and Test the axis with the axis control panel (Page 63).
- You have prepared two digital outputs for use in the program, see the section Configure digital outputs (Page 69).
- SIMOTION SCOUT is in offline mode.
12.2 Variables

12.2.1 Variable types

Variable types

Several variable types are distinguished in SIMOTION:

- **System variables**
  Each SIMOTION device and each technology object has specific system variables. You can access system variables within the SIMOTION device from all programs.

- **I/O variables**
  An I/O variable is a symbolic variable name that is assigned to an I/O address of the SIMOTION device or to the I/O. Direct access to the I/O is thus possible.
  I/O variables are valid across all devices. All programs of the SIMOTION device have access to them.

- **Global device variables, unit variables, and local variables** are user-defined variables with a limited scope of validity:
  All programs of a SIMOTION device can access global device variables.
  Unit variables can be accessed by all programs, function blocks and functions defined within the same unit, e.g. ST unit, MCC unit, LAD/FBD unit.
  A unit (source) is a logic unit that you can create in your project and that can contain programs, functions and function blocks.
  Local variables can only be accessed within the program, the function or the function block in which they are defined.

You can find further information on variables in the online help under **Variable model**.
12.2.2 Variables of the sample project

You require the following variables for the sample project of Getting Started:

2 global device variables
- g_bo_start
- g_bo_ready

2 I/O variables
- q_bo_output0
- q_bo_output1

Additionally when using an infeed with DRIVE-CLiQ interface:

2 I/O variables
- LineModule_STW
- LineModule_ZSW

Instance
- myFB_LineControl

Use of the variables in the sample project

The variables g_bo_start and g_bo_ready redundantly describe the program states "Program started" and "Program end reached".

The I/O variables q_bo_output0 and q_bo_output1 apply the program state to the configured digital outputs of the SIMOTION device, e.g. for a display.

The instance myFB_LineControl and the I/O variables LineModule_STW and LineModule_ZSW control the infeed.
12.2.3 Creating global device variables

You create global device variables in the Symbol browser tab of the detail view.

You create global device variables as follows

Create the global device variables `g_bo_start` and `g_bo_ready` for the sample project as follows.

The variable definition encompasses:

- Variable name
- BOOL data type

Note

You can only create global device variables in offline mode.

1. Click in the project navigator under the SIMOTION device on the element GLOBAL DEVICE VARIABLES.
   The Global device variables table is displayed in the symbol browser.
2. Click in the Name column on the first free cell and enter the variable name `g_bo_start`.
   Press RETURN or TAB. The input focus jumps to the Data type field. Alternatively, you can click in the field to move the input focus there.
3. Enter the data type BOOL in the Data type field.
4. Press RETURN to confirm.
   The variable is created and is available in the project. A new empty line is opened for input in the symbol browser.
5. Create the global device variable `g_bo_ready` accordingly.

The figure below shows the full definition.

![Global device variables of the sample project](image)

Figure 12-1  Global device variables of the sample project
12.2.4 Creating I/O variables

When configuring I/Os, SIMOTION SCOUT (as of Version 4.2) supports the symbolic assignment of inputs and outputs located on SIMOTION/SINAMICS components. To be able to access the drive-level I/O with an I/O variable, it is only necessary to assign the I/O variable to the I/O channel. SIMOTION SCOUT automatically sets up telegrams, BICO interconnections and addresses.

You create the I/O variables for output to the configured digital outputs as follows

You create the I/O variables `q_bo_output0` and `q_bo_output1` for the sample project as follows.

The variable declaration encompasses:

- Variable name
- Definition as output variable OUT
- BOOL data type
- Assignment to a digital output

**Note**

I/O variables can only be created in offline mode.

1. Double-click the element **ADDRESS LIST** in the project navigator below the SIMOTION device.

   The **Address list** tab opens in the detail view.

2. Click the first free cell in the **Name** column. Enter the variable name `q_bo_output0`.

   Press RETURN or TAB. The input focus jumps to the **I/O address** field. Alternatively, you can click in the field to move the input focus there.

3. Enter the keyword `OUT` in the **I/O address** field.

4. Enter the data type `BOOL` in the **Data type** field, or leave the field empty.
5. Assign the configured digital output to the variable:
   - Click the button in the **Assignment** cell. The Assignment dialog opens.
   - Open the **D435** element in the Assignment dialog.
   - Click the digital output **DO_0** below the **D435** element. **Assign** appears in the **Assignment** column.

   ![Assignment Dialog]

   **Figure 12-2 Assigning I/O variable and I/O channel**

6. Confirm with **OK**.

   The assignment of I/O variable and I/O channel has been completed. The I/O variable is created and is available in the project. A new empty line is opened for input in the **Address list** table.
7. Create the I/O variable `q_bo_output1` in the same way. Assign the variable to the configured digital output DO 1. The figure below shows the full declaration:

![Figure 12-3 I/O variables of the sample project](image)

8. Leave the **Address list** tab open for declaring further I/O variables.
You create the I/O variables for controlling the infeed as follows

If you are using an infeed with DRIVE-CLiQ interface, create also the I/O variables LineModule_STW and LineModule_ZSW.

Note

I/O variables can only be created in offline mode.

Proceed as for the definition of the I/O variables q_bo_output0 and q_bo_output1.

1. Open the Address list tab in the detail area if not already visible.

2. Create the I/O variable LineModule_STW.
   - I/O address: OUT
   - Data type: WORD
   - Assignment: Control word E_STW1 of the SINAMICS infeed

3. Create the I/O variable LineModule_ZSW.
   - I/O address: IN
   - Data type: WORD
   - Assignment: Status word E_ZSW1 of the SINAMICS infeed
The figure below shows the full declaration:

![Image of I/O variables](image)

Figure 12-5  I/O variables of the sample project

### 12.2.5 Back up the configuration

Back up the variables created in the sample project.

To do so, click in the toolbar on the **Save project** or **Save project and compile changes** button.
12.3 Programming

12.3.1 Programming languages in the sample project
Within the scope of the sample project, you create a simple user program with the programming languages MCC and LAD/FBD.
For this, you use the variables you created in the previous section Variables (Page 72).

12.3.2 MCC Motion Control Chart

12.3.2.1 The programming language MCC
MCC (Motion Control Chart) is a graphical programming language.
The programmed motion sequences (machine sequences) are shown in the MCC as flowcharts (MCC charts). The structure of the flowcharts is oriented around the actual operating sequence of the machine. A sequential motion sequence is programmed.

System in SIMOTION SCOUT
The PROGRAMS folder under the SIMOTION device contains the MCC units created in the project.
An MCC unit contains the MCC programs that are to run on the SIMOTION device.
"MCC program" is a collective term for different program organization units (POU).
A POU can be a program, a function, or a function block. The type of the POU is indicated in the project navigator by an icon:

- Program
- Function
- Function block

An MCC chart is the graphical representation of a program organization unit POU.
An MCC unit can contain several MCC charts.

Program creation steps
Creation of an MCC program encompasses the following steps:
1. Creating the MCC unit.
2. Creating the MCC charts in the MCC unit.
3. Inserting MCC commands in the MCC chart and parameterizing the commands.
12.3.2.2 Creating the MCC unit

You create the MCC unit motion for the sample project as follows.

You create an MCC unit in the project as follows

1. Open the PROGRAMS folder under the SIMOTION device in the project navigator. Double-click Insert MCC unit. The Insert MCC unit window appears.
2. Enter the name motion.
3. Go to the Compiler tab. For diagnostics purposes, activate the options Permit program status and Permit single step. In this way, you can monitor program execution later in online mode.
4. Confirm with OK.

The MCC unit is created.

– The unit appears in the project navigator under the PROGRAMS branch.
– In the working area of the workbench, the declaration table of the unit opens. The variables declared there apply within the MCC unit and can be linked in other units.

No other variable declaration is required for the sample project. You have already created the necessary variables as global device variables in the symbol browser.

12.3.2.3 Creating the MCC chart

You create the MCC chart pos_axis for the sample project as follows. The MCC chart pos_axis is a POU of the type program.

You create an MCC chart as follows

1. Open the PROGRAMS folder under the SIMOTION device in the project navigator.
2. Open the MCC unit motion in the PROGRAMS folder.
3. Double-click Insert MCC chart.

The Insert MCC Chart window appears.
4. Enter the name pos_axis.
5. Select the creation type Program.
6. Confirm with OK.

The MCC chart is created in the project.

– The created MCC chart pos_axis appears in the PROGRAMS folder under the motion unit.
– The MCC editor is opened in the working area of the workbench. The start and end nodes are already pre-defined. You can start MCC programming.
12.3.2.4 Inserting command blocks into an MCC chart

Every newly created MCC chart already contains a start and end node.

![MCC chart, start and end block](image)

You insert the MCC command blocks between these. The commands are processed in the direction from the start to the end node.

The MCC commands are available to you via:

- MCC editor toolbar
- MCC Chart > Paste menu command
- Context menu of the command block

You work with the MCC editor toolbar as follows

Open the toolbar

The MCC editor toolbar becomes visible in the workbench as soon as you open an MCC chart.

The commands are arranged into command groups. The sample project uses commands from the command groups Basic commands, Program structures, and Single axis commands.

![MCC editor toolbar](image)

1. Command groups
2. Basic commands
3. Program structures
4. Single axis commands

Figure 12-7 MCC editor toolbar
Note

If you do not see the toolbar, check that the display is switched on: Open the menu View > Toolbars. Activate the checkbox for MCC editor in the Toolbars window.

Open the command group

Move the mouse over the colored buttons of the toolbar to show the command groups.

Figure 12-8 Open the command group

Keeping command groups open or closed continuously

Click the window title of a command group to keep the command group open continuously.
Select Hide in the context menu of a command group to close the command group.

Placing the command group as required

Drag the toolbar or the command groups of the toolbar with the mouse to any location on the workbench.

Docking the toolbar

Drag the toolbar or the command groups of the toolbar with the mouse to the edge areas of the workbench to dock them there.

Showing a tooltip for the command

Hold the mouse pointer briefly over a command button. The designation of the command is shown.

You insert commands into an MCC chart with the help of the MCC editor toolbar as follows

1. Click in the active MCC chart on the connecting line between two commands, or click the command after which the new command is to be inserted.
   The connecting line or the border of the command button is marked in blue. The marking flashes.
2. Select the command group in the MCC editor toolbar. Click the desired command in the command group.
   The command is inserted into the chart.

The newly inserted command is empty. It must then be parameterized; e.g. for a variable assignment, the name of the variable and the assigned value must be specified.

The following section Creating an MCC sample program: basic framework (Page 84) describes the procedure in detail.
12.4 Creating an MCC sample program: basic framework

12.4.1 Overview

In the sample project, you create the MCC program \texttt{pos_axis}. This program traverses an axis to a target position and back to the starting position.

Handling the infeed

If you are using an infeed without a DRIVE-CLiQ interface, the basic framework for program-controlled operation of the axis dealt with in this section is sufficient. The ready signal of the infeed is wired to the digital input DI 0 of the D435-2, see section Configure the infeed (Page 53).

If you are using an infeed with a DRIVE-CLiQ interface, the program must switch the infeed on. Getting Started deals with this additional program sequence separately in the section Expanding the MCC sample program: control of the infeed (Page 98).
12.4.2  Program flow

The finished MCC chart

The figure below shows the finished MCC chart of the sample program.

The first two command blocks initialize the variables `g_bo_ready` and `g_bo_start`. The axis is then enabled, referenced, and traversed to the target position. When the position has been reached, the axis returns to the starting position. The enable is revoked. Finally, the variable `g_bo_start` is reset and the variable `g_bo_ready` is set.

Command groups and individual commands

In the figure above, the icons of the command group and of the individual command that you must click in sequence to insert the command into the MCC chart are shown in front of every MCC command.

The sections below describe the procedure in detail.
12.4.3 **Variable assignment g_bo_ready:=false / g_bo_start:=true**

The variable `g_bo_ready` is set to FALSE to set the variable to a defined starting point. The variable `g_bo_start` is set to TRUE to indicate the start of the axis.

To assign the variables in the MCC editor, proceed as follows:

1. Mark the line between the start and end nodes.

```
Start
START

End
END
```

![Figure 12-9 Empty MCC chart](image)

2. Open the **Basic commands** command group in the **MCC editor** toolbar. Click the **Variable assignment** command in the command group.

![Figure 12-10 MCC editor toolbar, variable assignment command](image)

The **Variable assignment** command block is inserted into the MCC chart after the start node.

```
Start
START

Variable assignment
Variable assignment

End
END
```

![Figure 12-11 MCC chart with inserted variable assignment command block](image)
3. Assign the value FALSE to the variable **g_bo_ready**. You transfer the variable name from the symbol browser to the command block:

- Double-click the **Variable assignment** command block.

  The **Variable assignment** window appears.

- Select the value **Formula** in the drop-down list.

  A table with the columns **Variables** and **Expression** appears.

- Open the symbol browser. To do so, click **GLOBAL DEVICE VARIABLES** under the SIMOTION device in the project navigator.

- Select the variable **g_bo_ready** in the symbol browser. To do so, click in the first column of the symbol browser on the preceding line number. The line is shown on a black background.

  ![Figure 12-12 Symbol browser](image)

  Figure 12-12 Symbol browser

- Drag and drop the marked line to the **Variable** column of the **Variable assignment** window. The variable is inserted and its name displayed.

- In the **Expression** column, enter the value **false** and confirm with RETURN.

  ![Figure 12-13 Variable assignment](image)

  Figure 12-13 Variable assignment
4. Assign the value **true** to the variable `g_bo_ready` in the same way.

5. Click **OK** to close the window.

   You have set the global device variables to the values **false** and **true**.
12.4.4 Switch axis enable

You switch the axis enable as follows

1. Insert the statement block **Switch axis enable** into the MCC chart in front of the end node:
   - Click the last block before the end node. The node is marked in blue.
   - Open the **Single axis commands** command group in the **MCC editor** toolbar. Click the **Switch axis enable** command in the command group.

   ![MCC editor toolbar, Switch axis enable command](image)

   Figure 12-16 MCC editor toolbar, Switch axis enable command

   The **Axis enable** command block is inserted into the MCC chart after the selected block and marked in blue.

2. Parameterize the axis enable:
   - Double-click the inserted **Switch axis enable** command block in the MCC chart. The **Switch axis enable** window is displayed.
   - In the **Axis** field, select the axis that is controlled by the program.
   - Make sure that the **Delay program execution** option is selected for the sample project. The command is only finished when the axis is enabled.
   - Leave the other parameters at the default values.
3. Click **OK** to close the window.

   The command is parameterized.
12.4.5 Home axis

The position values of an axis are always in relation to the axis coordinate system. The homing procedure defines where the zero point of the coordinate system is.

Homing of the axis is implemented as follows

1. Insert the statement block Home axis into the MCC chart in front of the end node:
   - Click the last block before the end node. The node is marked in blue.
   - Open the Single axis commands command group in the MCC editor toolbar. Click the command Home axis in the command group.

![MCC editor toolbar, Home axis command](image)

   Figure 12-18 MCC editor toolbar, Home axis command

   The Home axis command block is inserted into the MCC chart after the selected block and marked in blue.

2. Parameterize axis homing:
   - Double-click the inserted Home axis command block in the MCC chart. The Home axis window appears.
   - In the Axis field, select the axis that is controlled by the program.
   - In the Parameters tab, select the homing type Set home position. Enter the value 0 in the Home position coordinates field.

   With this homing type, the value of the home position coordinates are assigned to the current position of the axis (actual value). There is no active traversing motion.

   - Leave the other parameters at the default values.
3. Click **OK** to close the window.

   The command is parameterized.
12.4.6 Position the axis to the target position

The program is to traverse the axis at a velocity of 500 mm/s to the position 2000 mm.

You traverse the axis to the target position as follows

1. Insert the statement block Position axis into the MCC chart in front of the end node:
   
   - Click the last block before the end node. The node is marked in blue.
   
   - Open the Single axis commands command group in the MCC editor toolbar. Click the Position axis command in the command group.

   ![Figure 12-20 MCC editor toolbar, Position axis command](image)

   The Position axis command block is inserted into the MCC chart after the selected block and marked in blue.

2. Parameterize the axis motion:
   
   - Double-click the inserted Position axis command block in the MCC chart. The Position axis window appears.
   
   - In the Axis field, select the axis that is controlled by the program.
   
   - Enter the position 2000 in the Parameters tab. Set the velocity to 500 mm/s.
   
   - Leave the other parameters at the default values.
3. Click **OK** to close the window.

   The command is parameterized.
12.4.7 Position the axis to the starting position

The program is to return the axis to the starting position 0 mm at a velocity of 500 mm/s.

You return the axis to the starting position as follows

Proceed as for the step Position the axis to the target position (Page 93).

1. Insert the statement block **Position axis** into the MCC chart in front of the end node.
2. Parameterize the axis motion:
   - Enter the position 0 in the **Parameters** tab. Set the velocity to 500 mm/s.
   - Leave the other parameters at the default values.

3. Click **OK** to close the window.

The command is parameterized.
12.4.8 Disable axis

You disable the axis as follows

1. Insert the statement block **Disable axis** into the MCC chart in front of the end node:
   - Click the last block before the end node. The node is marked in blue.
   - Open the **Single axis commands** command group in the **MCC editor** toolbar. Click the **Disable axis** command in the command group.

   ![MCC editor toolbar, Disable axis command](image)

   Figure 12-23 MCC editor toolbar, Disable axis command

   The **Disable axis** command block is inserted into the MCC chart after the selected block and marked in blue.

2. Parameterize the command:
   - Double-click the inserted **Disable axis** command block in the MCC chart. The **Disable axis** window is displayed.

   ![Disable axis window](image)

   Figure 12-24 MCC chart, Disable axis

   - Remove pos. controller enable
   - Remove enable to be removed
   - Remove drive enable
   - Remove pulse enable

   Follow-up mode: [Follow-up setpoint]

   Delay program execution

   ![OK, Cancel, Accept, Help buttons](image)
In the **Axis** field, select the axis that is controlled by the program.

- Leave the other parameters at the default values.

3. Click **OK** to close the window.

The command is parameterized.

### 12.4.9 Variable assignment g_bo_start:=false / g_bo_ready:=true

The axis motion is finished. The variable **g_bo_start** is set to FALSE. The variable **g_bo_ready** is set to TRUE.

**To assign the variables in the MCC editor, proceed as follows:**

Proceed as for the step Variable assignment g_bo_ready:=false / g_bo_start:=true  (Page 86)

1. Insert the statement block **Variable assignment** into the MCC chart in front of the end node.

2. Assign the value **false** to the variable **g_bo_start** and the value **true** to the variable **g_bo_ready**.

3. Click **OK** to close the window.

The command is parameterized.
12.5 Expanding the MCC sample program: control of the infeed

12.5.1 Program flow

If you are using an infeed with DRIVE-CLiQ interface, the programmed controller must switch on the infeed before the axis commands can then be issued. The program sequence shown below handles this task. The pos_axis program created until now must be expanded accordingly.

The finished MCC chart

The two times two blocks _linemodule_control system function call and LineModule_STW variable assignment are located in the MCC chart pos_axis before the first command for axis control Switch axis enable.

The first two blocks before the UNTIL loop reset the operating parameters of the infeed to the start values. Without this initialization, it may not be possible to switch the infeed on. In addition, this also acknowledges alarm messages that prevent enabling of the infeed.

The blocks within the UNTIL loop switch the infeed on. The loop repeats the switching command until the infeed is ready for operation.

Note

To avoid excess load on the performance of the round robin execution level (background task, motion tasks) during endless loops, it makes sense to use a _waitTime (T#0ms) in the loop while actively waiting for an event.
12.5.2 System function call _LineModule_control[FB]

You create the system function call _LineModule_control[FB] as follows

1. Insert the statement block System function call into the MCC chart:
   - Click the command block $g_{bo\_start} := \text{true}$.
     The node is marked in blue.
   - Select the Basic commands > System function call command in the MCC editor toolbar.

   ![MCC editor toolbar, System function call command](image)

   Figure 12-25  MCC editor toolbar, System function call command

   The command block is inserted into the MCC chart.

2. Parameterize the function call:
   - Double-click the empty System function call command block in the MCC chart.
     The System function call [pos_axis] dialog appears.
   - Open the command library. You can find the command library in the screen area of the project navigator as a separate tab.

   ![Open the command library](image)

   Figure 12-26  Open the command library

   - Go to the Drives > SINAMICS branch of the command library.
   - Drag & drop the _LineModule_control[FB] function call from the command library to the System function field of the System function call[pos_axis] window.
3. Define an instance for the system function call:

- Enter the instance name `myFB_LineControl` in the **Instance** field of the **System function call [pos_axis]** window.

As soon as you exit the field, e.g. by clicking on another element or with the TAB or RETURN keys, the **Variables declaration** dialog appears. The instance variable `myFB_LineControl` is assigned to the variable type **VAR** as default. The variable is thus valid locally within the program **pos_axis**.

![Variables Declaration](Image)

Figure 12-27 Variable declaration of the instance variable `myFB_LineControl`

Accept the default. Click **OK** to confirm the dialog.

- Set the input variables **VAR_INPUT** and the output variable **VAR_OUTPUT** of the instance to the following values.

Make use of the convenience of the autocomplete operator function **Ctrl+spacebar**:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset</td>
<td>true</td>
<td>Reset the infeed</td>
</tr>
<tr>
<td>periIn</td>
<td>LineModule_ZSW</td>
<td>Status word of the infeed</td>
</tr>
<tr>
<td>typeLM</td>
<td>ACTIVE_LINE_MODULE SMART_LINE_MODULE BASIC_LINE_MODULE</td>
<td>Type of the module to be controlled depending on the infeed used</td>
</tr>
<tr>
<td>periOut</td>
<td>LineModule_STW</td>
<td>Control word, sequence control infeed</td>
</tr>
</tbody>
</table>
4. Confirm with **OK**. The command is parameterized.
12.5.3 Create UNTIL loop

You create the UNTIL loop as follows:

1. Click the connecting line between the blocks System function call and the subsequent axis command Switch axis enable. The reference point is marked in blue.

2. In the MCC editor toolbar, select the command Program structures > UNTIL: Loop with condition at end

![MCC editor toolbar, UNTIL loop](image)

The UNTIL block with loop is inserted.

![MCC editor, UNTIL loop](image)

3. Parameterize the loop condition:

   The loop is to be run through until the infeed is ready for operation.
   - Double-click the UNTIL block. The window UNTIL: Loop with condition at end[pos_axis] appears.
   - Select the value Formula in the field at top left.
   - Enter the loop condition in the Until field: myFB_LineControl.activated=TRUE.

   You can use the autocomplete function Ctrl+spacebar in this field too.

4. Confirm with OK.

   The loop is created and parameterized in the program.
12.5.4 Copy blocks

You copy the statement blocks in the MCC chart as follows:

Copy the statement blocks **System function call _LineModule_control[FB]** and **Variable assignment LineModule_STW:=myFB_LineControl.periOut** into the UNTIL loop. Use the operator functions **Copy** and **Paste** for this purpose.

1. Mark the blocks:
   - Hold the **Shift** key down.
   - Click the blocks one after the other. Both blocks are edged in blue.

2. Copy the selected blocks to the clipboard:
   - Open the context menu with the right mouse key.
   - Select the **Copy** command in the context menu.
   Or use the **Ctrl+C** key function.

3. Select the target of the copy operation in the MCC chart:
   - Click the connecting line within the **UNTIL** loop. The connecting piece is marked in blue.

![Figure 12-32 MCC chart, mark the UNTIL loop internal area](image)

4. Insert the copied blocks at the target position:
   - Open the context menu again with the right mouse key.
   - Select the **Paste** command in the context menu.
   Or use the **Ctrl+V** key function.
   The blocks are inserted within the loop.
Adapt the system function call _linemodule_control

You change the system function call _linemodule_control as follows

The system function call _linemodule_control within the UNTIL loop switches the infeed on.

1. Open the system function block by double-clicking.

2. Adapt the parameter variables of the myFB_LineControl instance as shown below:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>true</td>
<td>Switches the infeed on</td>
</tr>
<tr>
<td>reset</td>
<td>Empty field. The variable is thus set to false</td>
<td></td>
</tr>
<tr>
<td>periIn</td>
<td>LineModule_ZSW</td>
<td>Status word of the infeed</td>
</tr>
<tr>
<td>typeLM</td>
<td>ACTIVE_LINE_MODULE, SMART_LINE_MODULE, BASIC_LINE_MODULE</td>
<td>Type of the module to be controlled depending on the infeed used</td>
</tr>
</tbody>
</table>

3. Confirm with OK.
12.6 Create additional MCC programs for the sample project

Handling system events

The execution system that you will set up in the later section Configure execution system (Page 117) requires two further MCC programs:

- technology_fault
- peripheral_fault

These programs are used for handling system events. Handling is generally necessary. If a system event occurs that is not handled by the user program, the CPU goes to STOP mode.

Within the scope of the sample project, no specific handling of system events is necessary. The programs technology_fault and peripheral_fault therefore remain empty.

You create the MCC programs technology_fault and peripheral_fault as follows

1. Create the MCC unit fault.

   Open the PROGRAMS folder under the SIMOTION device in the project navigator. Double-click Insert MCC unit. Assign the name fault to the MCC unit.

2. Create the MCC chart technology_fault.

   Open the MCC unit fault in the PROGRAMS folder. Double-click Insert MCC chart. Assign the name technology_fault to the MCC chart. Select the creation type Program.

   The program remains empty.

   Figure 12-33 Empty program technology fault

3. Create the MCC chart peripheral_fault within the MCC unit fault.

   This program also remains empty.

   Figure 12-34 MCC programs technology_fault and peripheral_fault in the MCC unit fault
12.7 Back up MCC sample programs

Back up the created MCC programs.

To do so, click in the toolbar on the Save project or Save project and compile changes button.

Additional information

As an alternative to the command Save project and compile changes, you can find the command Accept and compile in the MCC editor toolbar.

This command compiles the currently selected program as well as all other programs of the same unit.

However, the command does not save the changes.

You thus have the option of accepting changes to a program into the project without having to save or compile the entire project again.
12.8 LAD/FBD ladder logic/function block diagram

12.8.1 The LAD and FBD programming languages

LAD (ladder logic)
LAD stands for ladder logic. LAD is a graphical programming language. The syntax for the statements is similar to a circuit diagram. LAD enables simple tracking of the signal flow between conductor bars via inputs, outputs and operations. The LAD statements comprise elements and blocks that are connected graphically to form networks. LAD operations follow the rules of Boolean logic.

FBD (function block diagram)
FBD stands for function block diagram. FBD is a graphical programming language. To represent the logic relationships, it uses the logic boxes familiar from Boolean algebra. In addition, complex functions (e.g. mathematical functions) can be represented directly in conjunction with the logic boxes.

System in SIMOTION SCOUT
The PROGRAMS folder under the SIMOTION device contains the LAD/FBD units created in the project.

A LAD/FBD unit contains the LAD/FBD programs that are to run on the SIMOTION device. A LAD/FBD unit can contain several LAD/FBD programs.

"LAD/FBD program" is a collective term for different program organization units (POU).

A POU can be a program, a function, or a function block. The type of the POU is indicated in the project navigator by an icon:

- □ Program
- □ Function
- □ Function block

Program creation steps
Creation of a LAD/FBD program encompasses the following steps:

1. Creating a LAD/FBD unit.
2. Creating a LAD/FBD program in the LAD/FBD unit.
3. Inserting LAD/FBD commands in the LAD/FBD program and parameterizing the commands.
4. Saving and compiling the LAD/FBD program.
Switching the programming language

SIMOTION SCOUT allows simple switching between ladder logic and function block diagram. The LAD/FBD editor contains the command **LAD/FBD program > Switch to FBD** or **Switch to LAD**.

12.8.2 Create LAD/FBD unit

You create the LAD/FBD unit **bg_out** for the sample project as follows.

You create a LAD/FBD unit in the project as follows

1. Open the **PROGRAMS** folder under the SIMOTION device in the project navigator. Double-click **Insert LAD/FBD unit**. The **Insert LAD/FBD unit** window appears.
2. Enter the name **bg_out** for the unit.
3. Go to the **Compiler** tab. For diagnostics purposes, activate the option **Permit program status**. In this way, you can monitor program execution later in online mode.
4. Confirm with **OK**.

The LAD/FBD unit is created.

- The LAD/FBD unit **bg_out** appears in the **PROGRAMS** folder.

![Figure 12-35 MCC and LAD/FBD units in the PROGRAMS folder](image)

- In the working area of the workbench, the declaration table of the unit opens. The variables declared there apply within the LAD/FBD unit and can be linked in other units.

No other variable declaration is required for the sample project. You have already created the necessary variables as global device variables in the symbol browser.
12.8.3 Create LAD/FBD program

You create the LAD/FBD program LAD_1 within the LAD/FBD unit bg_out as follows.

You create a LAD/FBD program in a LAD/FBD unit as follows

1. Open the PROGRAMS folder under the SIMOTION device in the project navigator.
2. Open the LAD/FBD unit bg_out in the PROGRAMS folder. Double-click Insert LAD/FBD program. The Insert LAD/FBD program window appears.
3. Enter the name LAD_1. The name must be unique throughout the project.
4. Select the creation type Program.

![Insert LAD/FBD program window](image)

Figure 12-36 Insert LAD/FBD program
5. Confirm with **OK**.

The LAD/FBD program **LAD_1** is created in the project.

- The LAD/FBD program appears in the **PROGRAMS** folder.

![Figure 12-37 LAD program LAD_1 in the PROGRAMS folder](image)

- The LAD/FBD editor is opened in the working area of the workbench. You can start programming.
12.8.4 Creating a LAD sample program

You write the LAD program **LAD_1** for the sample project.

In this program, the global device variables `g_bo_start` and `g_bo_ready` are read cyclically, and the I/O variables `q_bo_output0` and `q_bo_output1` are set accordingly.

Transfer of the program status to the digital outputs DO 0 and DO 1 could also be programmed direct in the MCC chart **pos_axis**. However, Getting Started implements the assignment in an autonomous LAD program to introduce you to LAD programming.

You create the LAD program **LAD_1** for the sample project as follows

1. Open the LAD/FBD program if it is not already visible in the working area of the workbench. For this purpose, double-click the LAD/FBD program under the LAD/FBD unit in the **PROGRAMS** folder of the project navigator.

   In the sample program, the empty LAD/FBD program **LAD_1** is already open.

   ![Figure 12-38 Empty LAD program following first-time creation in the LAD/FBD unit](image)

2. Insert a network:
   - Click **Insert network** in the LAD/FBD toolbar.

   ![LAD/FBD Toolbar](image)

   The **001** block with a coil is inserted.
3. Insert an NO contact in front of the coil:
   – Select the coil.

   ![LAD program, selected coil](image)

   Figure 12-39  LAD program, selected coil

   – Click **NO Contact** in the toolbar.

   ![LAD/FBD toolbar](image)

   The NO contact is inserted in front of the coil.

4. Assign the **g_bo_start** variable to the NO contact and the **q_bo_output0** variable to the coil:
   – Click above the symbols for NO contact and coil on ???.

   ![LAD program, opened field for entering the variable name](image)

   Figure 12-40  LAD program, opened field for entering the variable name
– Transfer the variable names from the symbol browser or the ADDRESS LIST by dragging and dropping. Or use the Autocomplete function. Use RETURN to confirm each entry.

5. Insert a network 002:
   – Click the block 001 to select the block.
     A selected block has a blue background on the left edge.
   – Click Insert network in the LAD/FBD toolbar.
     The block 002 is inserted after the block 001.

6. Insert an NO contact in front of the coil in network 002.

7. Assign the g_bo_ready variable to the NO contact and the q_bo_output1 variable to the coil.

The LAD program is finished.
12.8.5 Back up LAD/FBD sample program

Back up the created LAD/FBD sample program.

To do so, click in the toolbar on the **Save project** or **Save project and compile changes** button.

Additional information

As an alternative to the command **Save project and compile changes**, you can find the command **Accept and compile** in the LAD/FBD toolbar.

This command compiles the currently selected program as well as all other programs of the same unit.

However, the command does not save the changes.

You thus have the option of accepting changes to a program into the project without having to save or compile the entire project again.
12.9 Other programming languages

SIMOTION SCOUT provides other programming languages in addition to MCC and LAD/FBD.

- **ST (Structured Text)**
  
  ST is a text-based, PASCAL-based programming language. The language is based on the international standard IEC 61131-3. This standardizes the programming languages for programmable controllers (PLC). ST is based on the Structured Text part of this standard. In addition to the standardized programming language in accordance with IEC 61131-3, SIMOTION ST features technological commands.

  An easy-to-use text editor is provided for creating programs. The ST compiler compiles the edited program into executable code and indicates every syntax error, specifying the program line and the cause of the error.

  In the Command library tab of the project navigator, the commands and functions required for programming are shown in a tree. You can use these in all programming languages, e.g. when programming.

- **DCC (Drive Control Chart)**

  Drive Control Chart (DCC) means graphic configuration and expansion of the device functionality by means of freely available control, calculation and logic blocks.

  DCC comprises the DCC editor and the DCB library (block library with standard DCC blocks).

  The user-friendly DCC Editor enables easy graphical configuration and a clear representation of control loop structures as well as a high degree of reusability of existing diagrams.

  The open-loop and closed-loop control functionality is defined by using multi-instance-capable blocks (Drive Control Blocks (DCBs)) from a predefined library (DCB library) that are selected and graphically linked by dragging and dropping.

- **Libraries**

  Libraries allow modular software development and provide you with user-defined data types, functions and function blocks that you can use from all SIMOTION devices.

  Libraries can be written in all programming languages and used in all program sources (e.g. ST units, MCC units).

  All programming languages enable you to structure the application using programs, functions and function blocks so that the application is manageable and re-usable.

  To test the programs, there are comprehensive test functions available to you with program status and breakpoints in all languages. You can visualize and test your programs online.

  For further information on programming languages, please refer to the respective manuals.
12.10 Result in the sample project

You have created the programs required for program-controlled traversing of the axis in the sample project of Getting Started.
13 Configure execution system

13.1 Overview

Aim of Getting Started

In this part of Getting Started, you get to know the SIMOTION execution system. You assign the sample project programs to the tasks of the execution system. You thus transfer the programs from the SIMOTION SCOUT engineering system to the SIMOTION runtime system. The programs control the system as soon as the operating mode of the SIMOTION device is switched to RUN.

Preconditions

You have created the programs pos_axis, technology_fault, peripheral_fault, and LAD_1 for the sample project, see the section Programming the SIMOTION application (Page 71).

13.2 Execution system

Execution levels and tasks

Execution levels define the chronological sequence of tasks in the execution system. One level can contain several tasks. The tasks provide the framework for program execution. One task can contain several programs. By assigning the created programs to the tasks, you can, for example, define the priority, the time frame, or the order in which the programs are to be executed.
13.3 Assign programs to tasks

Below you will assign the sample project programs to the tasks of the execution system.

<table>
<thead>
<tr>
<th>Programs of the sample project</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>pos_axis</td>
<td>MotionTask_1</td>
</tr>
<tr>
<td>lad_1</td>
<td>BackgroundTask</td>
</tr>
<tr>
<td>technology_fault</td>
<td>TechnologicalFaultTask</td>
</tr>
<tr>
<td>peripheral_fault</td>
<td>PeripheralFaultTask</td>
</tr>
</tbody>
</table>

You assign programs to tasks in the execution system as follows

1. Double-click in the project navigator under the SIMOTION device on EXECUTION SYSTEM.

   The EXECUTION SYSTEM window appears in the working area.

2. Assign the MCC program `motion.pos_axis` to the task MotionTask_1:
   - In the tree of the execution system, select the branch ExecutionLevels > OperationLevels > MotionTasks > MotionTask_1.

   ![Execution system, tree view of the execution levels and tasks](image)

   The MotionTasks window appears on the right of the working area. The programs pos_axis and LAD_1 as well as auxiliary programs of the fault unit are visible on the Program assignment tab under Programs.
13.3 Assign programs to tasks

- Select the MCC program `motion.pos_axis` and click the >> button. The program is displayed under Programs used. It is thus assigned to the task `MotionTask_1`.

![Figure 13-2 Execution system, MotionTasks window](image)

The assignment is visible in the tree of the execution system. The program `motion.pos_axis` appears below the branch `MotionTask_1`.

![Figure 13-3 Execution system, MotionTask_1 with assigned program pos_axis](image)

- Activate the checkbox Activation after StartupTask in the Task configuration tab. This executes the MCC program immediately after the SIMOTION device is started. If this checkbox is not activated, the program must be started explicitly by the call from another program that is assigned to the StartupTask or another active task.

3. Assign the LAD program `bg_out.kop_1` to the task BackgroundTask:

- In the tree of the execution system, select the branch ExecutionLevels > OperationLevels > BackgroundTask. Assign the LAD program `bg_out.kop_1` to this task.
Configure execution system

13.3 Assign programs to tasks

4. Assign the error handling routines:
   – In the tree of the execution system, select the branch ExecutionLevels > OperationLevels > SystemInterruptTasks > TechnologicalFaultTask. Assign the MCC program fault.technology_fault to this task.
   – In the tree of the execution system, select the branch ExecutionLevels > OperationLevels > SystemInterruptTasks > PeripheralFaultTask. Assign the MCC program fault.peripheral_fault to this task.

5. Click Close. Confirm with Yes if you are prompted to save.

The execution system is configured.

Loading the configured execution system to the target system

- Save and compile the project.
- Go online.
- Download the sample project with the configured execution system to the target system.

See also

Download the project to the target system (Page 41)
13.4 Result in the sample project

Configuring the axis control is thus completed:

- You have set up an axis in the sample project.
- You have created a program for traversing an axis, as well as other programs that are necessary for operation.
- You have assigned these programs to the tasks of the SIMOTION runtime system.

In the following configuring steps, you will start the axis control on the SIMOTION device and monitor the program-controlled axis motion.
Configure execution system

13.4 Result in the sample project
14 Starting and stopping the system

14.1 Overview

Aim of Getting Started

You switch the SIMOTION CPU of the sample project to RUN mode to start execution of the pos_axis program. You can see on the hardware that the axis rotates twice for approximately 4 seconds. After execution of the program, switch back to the STOP state.

You switch the operating mode in the dialog Control Operating State.

Preconditions

- You have created the programs pos_axis, technology_fault, peripheral_fault and LAD_1 for the sample project, and you have assigned them to the tasks of the SIMOTION execution system, see the sections Programming the SIMOTION application (Page 71) and Configure execution system (Page 117).

- The project has been compiled and downloaded to the target system, see the section Download the project to the target system (Page 41).

- SIMOTION SCOUT is in online mode.
14.2 RUN and STOP operating states

Operating states
In the SIMOTION SCOUT dialog **Control Operating State**, you can switch a SIMOTION CPU to RUN or STOP mode.

**RUN mode**
SIMOTION executes the user program and the associated system services:
- Read process image input
- Execution of the user programs assigned to the execution system
- Write process image output

The technology packages are active in this state. They can execute commands from the user program.

**STOP mode**
SIMOTION does not process any user program.
- It is possible to load a complete user program.
- All system services (communication, etc.) are active.
- The I/O modules are in a secure state. (This means, for example, the digital outputs are at "LOW level" and the analog outputs are de-energized)
- The technology packages are inactive, that is, all enables are deleted. No axis motions can be executed.

You can find the full description of the operating states in the online help under **SIMOTION device: Operating state**, or in the **SIMOTION SCOUT Configuration Manual**.
14.3 Mode selector switch on the software side and the hardware side

Control Operating State dialog

The menu command **Target system > Control operating state** or the button **Control operating state** in the toolbar of the workbench open the dialog **Control Operating State**.

The dialog lists all configured CPUs. The **State** column shows the current operating mode. The assigned switches in the **Control** column switch the CPU to the RUN or STOP mode. The switches are deactivated when the CPU is in offline mode.

![Control Operating State dialog](image)

Figure 14-1 Control Operating State dialog

**Note**

As of SIMOTION V4.4, the **Control Operating State** dialog represents all configured SIMOTION CPUs. The previous dialog had to be opened individually for each CPU.

You can find a detailed description of the dialog in the *SIMOTION SCOUT* Configuration Manual.
Priority of the mode selector switch on the SIMOTION device

The setting of the mode selector switch on the SIMOTION device has priority. SIMOTION SCOUT can only switch a SIMOTION device to RUN mode if the mode selector switch on the device is set to 0 (RUN).

Figure 14-2  Mode selector switch of the SIMOTION D435-2, switch position 0 (RUN)

You can find detailed information on the mode selector switch of the SIMOTION D4x5-2 devices in the Manuals and Commissioning Manuals, as well as in the online help.
14.4 Start program control of the sample project

You start the axis control of the sample project on the SIMOTION device as follows

<table>
<thead>
<tr>
<th>WARNING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danger to life through unexpected machine movement</td>
</tr>
<tr>
<td>Make sure this presents no hazard to personnel or property.</td>
</tr>
</tbody>
</table>

1. Open the **Control Operating Mode** dialog. To do so, click **Control Operating Mode** in the toolbar.

2. Switch the SIMOTION CPU of the sample project to the RUN mode. To do so, click the assigned **RUN** switch in the **Control** column of the **Control Operating Mode** dialog.

   The axis starts to rotate immediately. The axis returns to a standstill after approximately 8 seconds.

   The SIMOTION device remains in the RUN mode after execution of the **pos_axis** program.

3. Switch the SIMOTION device to STOP mode. To do so, click the assigned **STOP** switch in the **Control** column.

**Note**

If you want to traverse the axis again, acknowledge all alarms in the "Alarms" window.
Monitor the application

15.1 Overview

Aim of Getting Started

In this part of Getting Started, you monitor the program-controlled axis motion.

- You monitor program execution.
- You monitor the values in the symbol browser.
- You compile certain values in a watch table.
- You record the course of the axis motion with the trace.

Preconditions

- You have created the programs pos_axis, technology_fault, peripheral_fault and LAD_1 for the sample project, and you have assigned them to the tasks of the SIMOTION execution system, see the section Configure execution system (Page 117).
- The project has been compiled and downloaded to the target system, see the section Download the project to the target system (Page 41).
- SIMOTION SCOUT is in online mode.
- Operating mode of the SIMOTION device:
  - The SIMOTION device of the sample project has been switched to STOP mode on the software side. The SIMOTION SCOUT mode selector switch, Control Operating State dialog, shows the STOP mode.
  - The SIMOTION device of the sample project has been switched to RUN mode on the hardware side. The mode selector switch on the SIMOTION device is at position 0 (RUN).
15.2 Monitoring program execution

SIMOTION SCOUT provides several functions for monitoring program execution.

In the sample project, you use the function Monitor. With this function, you can track execution of the command blocks in an MCC chart. The currently executed command block is shown against a yellow background in the chart. The marking runs at the same speed as the actual program execution. The marking is therefore only visible with the “slow” commands that wait for the machine to implement the command.

You monitor the execution of the commands in the MCC chart pos_axis as follows

1. Open the Control Operating State dialog.

   ![](image)

   The SIMOTION device must be in STOP mode.

2. Open the MCC chart pos_axis.

3. Switch the monitoring function on:
   - To do so, select the command MCC chart > Monitor in the menu bar of the workbench. The function is switched on if a checkmark is visible next to the menu item.
   - Or click in the MCC editor toolbar on Monitor.
4. Start program control: Switch the SIMOTION device to **RUN** mode in the **Control Operating State** dialog.

The **pos_axis** program is run through once. The currently executed command block is shown against a yellow background.

![Diagram of program execution](image)

**Figure 15-1** Monitor the program execution of the MCC chart pos_axis

5. Switch the SIMOTION device to **STOP** mode in the **Control Operating State** dialog.

**Further information**

You can find further information on program monitoring in the online help under **Monitoring program execution**.
15.3 Monitoring variables

In the symbol browser, you can monitor variables (read out status value) or assign values to them (assign control values).

For the sample project, you monitor the actual position of the axis during the program run.

Monitor variables in the symbol browser

1. In the AXES folder of the project navigator, select the axis created in the sample project. The system variables and configuration data of the axis are displayed on the Symbol browser tab of the detail view.

2. Open the system variable \texttt{positioningstate.actualposition} (actual position of the axis) in the symbol browser.

   You find the variable as follows:
   
   - Filter the list: You can specify a filter criterion in the filter line of the symbol browser. The last 5 criteria are saved and can be selected for re-use.
     
     Enter a suitable filter term, e.g. \texttt{positioningstate}, in the filter line. Press \texttt{RETURN} to confirm.
   
   - Search for variable: As an alternative to the filter function, you can search for the variable. Select the menu command \texttt{Edit > Find}. Enter a suitable search term in the Find dialog, e.g. \texttt{positioningstate}. Click \texttt{Find next}.

   The actual position of the axis is displayed in the Status value column of the symbol browser.

3. Start the program control via the dialog box \texttt{Control Operating State}.

   The program runs once. The changing values of the actual axis position are displayed in the symbol browser.

   The SIMOTION device is in the RUN state after the program has run.

4. Switch the SIMOTION device to STOP mode in the \texttt{Control Operating State} dialog.

Monitoring variables in watch table

Different variables, e.g. of several devices, can be combined into a table to be monitored.

To include a variable in a watch table, you must first create a watch table.

Proceed as follows:

1. Double-click in the project navigator under \texttt{MONITOR} on the entry \texttt{Insert watch table} and confirm your configuration.

2. Right-click in the symbol browser on the variable that you want to add to the watch table. Choose the Add to watch table command from the shortcut menu.

   The selected variable will be displayed in the table. In this way, you can add further variables to the watch table and monitor them.
You open a watch table as follows

You will find all created watch tables in the MONITOR folder of the project navigator. Double-click a watch table to open it.

You can find detailed information on this topic in the online help under Working with the SCOUT Workbench > Working with lists > Watch table.

15.4 Recording signals with the trace

15.4.1 Trace

You can use the trace to record and save signal characteristics and variable values. The recorded data can be used, for example, for diagnostics purposes in machine motion sequences, and for troubleshooting in user programs.

For the sample program, you record the actual position of the axis over time and represent it in the diagram.
15.4.2 Working with the trace

You call the trace as follows:

1. Select the SIMOTION device in the project navigator.
2. Select the menu command **Target system > Device trace** or click the **Device trace function generator** button in the **Trace toolbar**.

The **Device trace** window appears in the working area.

![Device trace window](image)

---

1. … 5 Reference is made to the circled digits in the text below.

Figure 15-2 Device trace window
You parameterize the trace for recording as follows

1. In the **Duration** field ① of the **Trace** tab, enter the recording duration 15000 ms.
2. Click the button ② in line **No 1** of the **Signals** table.
   The **Trace Signal Selection** window appears.
3. In the tree, select the branch **Sample_1 > (SIMOTION device, e.g. D435) > TO > Axis_2 > positioningstate**.
4. Select the system variable **actualPosition** in the variable table.
5. Click the button 1 to accept the system variable for channel 1.

   ![Signal selection Trace](image)

   The variable is displayed under **Signal name**.

---

**Figure 15-3 Trace signal selection**
6. Repeat the channel assignment for the variable \texttt{g\_bo\_start}:  

Select the branch (\texttt{Project}) > (\texttt{SIMOTION device}) > \texttt{GLOBAL DEVICE VARIABLES}. Select the system variable \texttt{g\_bo\_start} in the variable table. Click the button 2.

\underline{Note}

You can also drag & drop the variables from the symbol browser or the watch table to the signal field of the Trace dialog. Procedure for dragging & dropping, see the section \texttt{Variable assignment g\_bo\_ready:=false / g\_bo\_start:=true} (Page 86).

7. Confirm with OK.

The Trace Signal Selection window is closed.

The trace is now parameterized for recording.

You save the parameterization of the trace as follows

The trace parameterization is not saved in the project data. When you close the project, the trace parameterization is deleted.

To save the parameterization, click the \texttt{Accept in catalog} button 3 on the Trace tab. In the Catalog entry field, enter the name under which the setting/parameterization is to be saved in the catalog of the trace.

You record with the trace as follows

1. Go online.

2. Download the parameterization of the trace to the target system:

   \begin{itemize}
   \item Click the \texttt{Download parameterization} button 4 on the Trace tab.
   \item Click OK to confirm the dialog that appears following successful downloading.
   \end{itemize}

3. Open the Control Operating State dialog.

4. Open the Time diagram tab in the Device trace window.
5. Start recording of the trace:
   - Click **Trace start** in the **Device trace** window.
   - Then immediately switch the SIMOTION device to RUN mode in the **Control Operating State** dialog.

   The program is started. The actual position of the axis is recorded and represented in the time diagram. After expiry of the recording duration, the signal profile of the actual position is displayed.

6. If the recorded curve is only partially displayed, select the menu item **Auto-scaling** in the context menu of the time diagram.

7. Switch the SIMOTION device to STOP mode in the **Control Operating State** dialog box.
Carrying out several measurements

You can carry out several measurements. They are shown on the Measurements tab and can be displayed in the diagram. You can save and re-open recorded measurements for documentation purposes.

15.4.3 Further diagnostic functions

You can find a detailed overview of the extensive service and diagnostics options in the online help under Diagnostics, as well as in the product information SIMOTION SCOUT Overview of service and diagnostics options.
15.5 Result in the sample project

Creation of the trace time diagram concludes Getting Started.

We recommend that you continue to familiarize yourself with SIMOTION SCOUT using the sample projects of the Utilities & Applications.

You can find information on the Utilities & Applications in the section Utilities & applications (Page 23).
Monitor the application

15.5 Result in the sample project
ESD directives

A.1 ESD definition

What does ESD mean?

Electrostatic sensitive devices (ESDs) are individual components, integrated circuits, modules or devices that may be damaged by either electrostatic fields or electrostatic discharge.

<table>
<thead>
<tr>
<th>NOTICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Damage caused by electric fields or electrostatic discharge</td>
</tr>
</tbody>
</table>

Electric fields or electrostatic discharge can result in malfunctions as a result of damaged individual parts, integrated circuits, modules or devices.

- Only pack, store, transport and send electronic components, modules or devices in their original packaging or in other suitable materials, e.g. conductive foam rubber or aluminum foil.
- Only touch components, modules and devices if you are first grounded by applying one of the following measures:
  - Wearing an ESD wrist strap
  - Wearing ESD shoes or ESD grounding straps in ESD areas with conductive flooring
- Only place electronic components, modules or devices on conductive surfaces (table with ESD surface, conductive ESD foam, ESD packaging, ESD transport container).
A.2 Electrostatic charging of individuals

Any person who is not conductively connected to the electrical potential of the environment can accumulate an electrostatic charge.

This figure indicates the maximum electrostatic charges that can accumulate on an operator when he comes into contact with the indicated materials. These values comply with the specifications in IEC 801-2.

Figure A-1  Electrostatic voltage that can accumulate on operating personnel
A.3 Basic measures for protection against discharge of static electricity

Ensure sufficient grounding

When working with electrostatic sensitive devices, make sure that the you, your workstation, and the packaging are properly grounded. This prevents the accumulation of static electricity.

Avoid direct contact

You should only touch ESD components if unavoidable (for example, during maintenance work). When you touch modules, make sure that you do not touch either the pins on the modules or the printed conductors. If you follow these instructions, electrostatic discharge cannot reach or damage sensitive components.

If you have to take measurements on a module, make sure that you first discharge any static that may have accumulated in your body. To do this, touch a grounded metal object. Only use grounded measuring instruments.
ESD directives

A.3 Basic measures for protection against discharge of static electricity
Index

A
Axis
Assigning a drive, 60
Creating, 58
Testing, 66
Traversing, 66
Axis control panel, 63
Opening, 64
Axis wizard, 57

C
Connect to selected target devices, 43
Control panel
Opening, 64
Controlling the operating state, 125
Create new project, 31
Creating
Axis, 58
Creating I/O variables, 75
Creating the MCC chart, 81
Creating the MCC unit, 81

D
Digital outputs, 69
Disable axis, 96
Drive
Automatic configuration, 48

E
ESD guideline, 141
Execution levels, 117
Execution system, 117

F
FBD, 107

G
Getting Started with SIMOTION SCOUT
Assign programs to tasks, 118
Configure digital outputs, 69
Configure execution system, 117
Configure the axis, 57
Configure the drive, 47
Configure the infeed, 53
Connect to selected target devices, 42
Create LAD/FBD program, 109
Create LAD/FBD unit, 108
Create new project, 31
Create SIMOTION device in the project, 35
Creating a LAD sample program, 111
Creating global device variables, 74
Creating I/O variables, 75
Creating the MCC chart, 81
Creating the MCC unit, 81
Download the project to the target system, 45
MCC sample program, basic framework, 84
MCC sample program, infeed, 98
Monitor the application, 129
Overview, 17
PG/PC, 39
Programming languages in the sample project, 80
Programming the SIMOTION application, 71
Recording signals with the trace, 133
Sample project, configuring steps, 17
Sample project, preconditions, 20
Save project and compile changes, 41
Set up PG/PC communication, 35
Starting and stopping the system, 123
Switch on the infeed with program control, 98
System function call _LineModule_control[FB], 99
Test the axis with the axis control panel, 63
Variable types, 72
Variables of the sample project, 73
Global device variables, 74
Guideline
ESD, 141

H
Handling system events, 105
I
I/O channels of the terminal X142, 69
Infeed, 53

L
LAD, 107

M
MCC, 80
  Disable axis, 96
  Handling system events, 105
  Home axis, 91
  Position the axis to the starting position, 95
  Position the axis to the target position, 93
  Switch axis enable, 89
MCC editor toolbar, 82
Mode selector switch
  in SIMOTION SCOUT, 125
  SIMOTION hardware, 126
Monitoring
  Program flow, 130
  Variables, 132
Monitoring variables
  Symbol browser, 132
  Watch table, 132
Motion Control Chart, 80

O
Operating state, 124

P
peripheral_fault, 105
PG/PC, 39
Program flow
  monitor, 130
Programming language
  FBD, 107
  LAD, 107
  MCC, 80
  ST, 115
Project download, 45

R
References, 3

Restoring
  factory settings, 25

S
Safety notes, 16
Save and compile the project, 41
Save project and compile changes, 106
SIMOTION D, 35
SIMOTION SCOUT online help, 21
Switch axis enable, 89
Symbol browser, 74
  Monitoring variables, 132

T
Target device selection, 43
Tasks, 117
Technology object axis, 57
technology_fault, 105
Trace, 133

U
Unit
  LAD/FBD, 108
  MCC, 81
Utilities & Applications, 23

V
Variable assignment
  g_bo_ready:=false, 86
  g_bo_ready:=true, 97
  g_bo_start:=false, 97
  g_bo_start:=true, 86
Variable types, 72
Variables
  Adding to watch table, 132
  monitor, 132

W
Watch table
  Inserting, 132
  Monitoring variables, 132
  Opening, 133