Documentation of the library
WagoAppSMI
Release 1.0.1.3
# Contents

1 Description

2 10 Documentation
   2.1 doc01_Foreword (FB) .................................................. 3

3 20 Program Organization Units
   3.1 10 Communication ...................................................... 5
       3.1.1 FbSMI_Master (FB) .............................................. 5
       3.1.2 PrgSMIConfigurator (PRG) ................................... 6
   3.2 20 Basic Commands .................................................... 10
       3.2.1 FbSMI_Diagnostic (FB) ....................................... 10
       3.2.2 FbSMI_Down (FB) .............................................. 12
       3.2.3 FbSMI_Move_Pos (FB) .......................................... 13
       3.2.4 FbSMI_Move_Pos1 (FB) ....................................... 15
       3.2.5 FbSMI_Move_Pos2 (FB) ....................................... 17
       3.2.6 FbSMI_Move_Pos_Turn (FB) ................................... 18
       3.2.7 FbSMI_Read_Pos (FB) .......................................... 20
       3.2.8 FbSMI_Read_Pos1 (FB) ....................................... 21
       3.2.9 FbSMI_Read_Pos2 (FB) ....................................... 22
       3.2.10 FbSMI_Read_Syn (FB) ....................................... 23
       3.2.11 FbSMI_Step_Down (FB) ..................................... 24
       3.2.12 FbSMI_Step_Up (FB) ........................................ 26
       3.2.13 FbSMI_Stop (FB) ............................................ 28
       3.2.14 FbSMI_Up (FB) ............................................... 29
       3.2.15 FbSMI_Write_Pos1 (FB) .................................... 31
       3.2.16 FbSMI_Write_Pos2 (FB) .................................... 32
   3.3 30 Adressing .......................................................... 34
       3.3.1 FbSMI_Address (FB) ........................................... 34
       3.3.2 FbSMI_SlaveAdr_Write (FB) ................................ 36
       3.3.3 FbSMI_SlaveId_Read (FB) .................................. 37
       3.3.4 FbSMI_SlaveId_Search (FB) ................................. 38
       3.3.5 FbSMI_SystemImage (FB) ................................... 39
   3.4 40 Motor ............................................................... 40
       3.4.1 FbSMI_Motor (FB) ............................................. 40
       FbSMI_Motor.RemainingOverrideTime (PROP) .................. 45
   3.5 50 Parametrization ................................................... 45
       3.5.1 FbSMI_Read_Par (FB) ...................................... 45
       3.5.2 FbSMI_Write_Par (FB) ...................................... 46

4 29 Types
   4.1 typSMI_Adressing (STRUCT) ....................................... 49
   4.2 typSMI_Config (STRUCT) ............................................ 49
CHAPTER 1

Description

This document is automatically generated. Because of this, the chapter 30 Visualization is not shown in this document. If you are interested in getting to know more about visualization, we refer to the library manager of e!Cockpit.

Subject to Changes

WAGO Kontakttechnik GmbH Co. KG reserves the right to provide for any alterations or modifications. WAGO Kontakttechnik GmbH Co. KG owns all rights arising from the granting of patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

All tasks that are carried out with libraries made for the e!COCKPIT software must only be performed by qualified electrical specialists instructed in PLC programming according to IEC 61131-3.

All tasks that have an effect on the properties or the behavior of automation hardware or software products must only be performed by qualified employees with a thorough knowledge of handling the products concerned.

Intended Use of e!COCKPIT Libraries

Libraries created for the e!COCKPIT software are used to simplify the development of application projects in the IEC 61131-3 programming languages.

For automation tasks, WAGO offers programmable logic controllers in a wide variety of performance classes. In combination with a wide range of I/O modules, the controllers can process standard types of field signals. Controllers can be implemented centrally or in decentralized configurations. The controllers offer interfaces for the most commonly used fieldbuses for use in decentralized configurations. Fieldbus independent I/O modules are then linked via fieldbus couplers. WAGO controllers offer a runtime environment for user programs called e!RUNTIME. Software projects for implementation in e!RUNTIME environments can be created in e!COCKPIT. The programming environment in e!COCKPIT is based on the established CODESYS 3 industrial standard. Users with a previous knowledge of CODESYS 3 will thus find this environment largely familiar. The following programming languages of the IEC 61131-3 standard are available:

- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)
- Sequential Function Chart (SFC)
- Continuous Function Chart (CFC)

The individual programming languages can also be combined as required during the development of the software. A portfolio of prepared libraries can be accessed for many frequently used functions in order to make software development more efficient. This document provides an overview of the WagoAppSMI that WAGO offers for e!COCKPIT.
This library is for standard motor interface (SMI) drives.

The function blocks of this library are NOT thread safe and must be called from one CODESYS task only! Concurrent calls from different tasks may cause loss or corruption of data. ¹

Further library information are summarized here:

- **Company**: WAGO
- **Title**: WagoAppSMI
- **Version**: 1.0.1.3
- **Categories**: Application; WAGO LayerView|App; WAGO BusinessView|Building Automation
- **Namespace**: WagoAppSMI
- **Author**: WAGO / u090996
- **Placeholder**: WagoAppSMI

The content of this file was automatically generated with None on 13.08.2019, 20:27:26
To ensure fast installation and start-up of the units, we strongly recommend that the following information and explanations are carefully read and adhered to.

## 2.1 doc01_Foreword (FB)

### Copyright

This document, including all figures and illustrations contained therein, is subject to copyright. Any use of this document that infringes upon the copyright provisions stipulated herein is prohibited. Reproduction, translation, electronic and phototechnical filing/archiving (e.g., photocopying), as well as any amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG, Minden, Germany. Non-observance will entail the right of claims for damages.

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from granting patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

### Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards. WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

### Intended Use

For each individual application, the components are supplied from the factory with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in this document. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

---

**Warning:** Do not send movement commands in PLC mode when controller lock is active! The movement commands executed via the SMI Configurator have higher priority than any controller lock set in the PLC software. Therefore, it is possible to move the SMI drives even when the controller lock is active. This only applies if the SMI Configurator is connected to an SMI master module via PLC mode. Any time you send movement commands with the SMI Configurator via PLC mode, first check whether the controller lock is active. Do not send movement commands with the SMI Configurator as long as the controller lock is active.

### Scope of Applicability
This application note is based on the stated hardware and software from the specific manufacturer, as well as the associated documentation. This application note is therefore only valid for the described installation. New hardware and software versions may need to be handled differently.

Please note the detailed description in the specific manuals.
3.1 10 Communication

3.1.1 FbSMI_Master (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bPortSMI</td>
<td>BYTE</td>
<td>Specifies which SMI module is to be addressed at the controller. Counting is from left to right.</td>
</tr>
<tr>
<td></td>
<td>I_Port</td>
<td>WagoTypesModule_753_163x.I_Module</td>
<td>Interface to the 753-163x module</td>
</tr>
<tr>
<td></td>
<td>xEnableEnergySaver-Mode</td>
<td>BOOL</td>
<td>Set module in energy saver mode</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xEnergySaver-Mode</td>
<td>BOOL</td>
<td>Shows status of the energy saver mode (True := active, False := inactive)</td>
</tr>
<tr>
<td></td>
<td>xStateDigIn</td>
<td>BOOL</td>
<td>Shows status of the digital input (True := high level, False := low level)</td>
</tr>
</tbody>
</table>

Function:

The FbSMI_Master function block is required for connection of the SMI master module (753-1630) and the SMI master module LoVo (753-1631) to the WAGO-I/O-SYSTEM. All other function blocks communicate with the SMI Multi-Master module via this function block. The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

This function block detects all queued commands of the other SMI function blocks in the program and causes their execution. The commands are made available in the form of a data table via the global variable g_typSMI. The variable “g_typSMI” contains all relevant call parameters and data for the receiving telegrams/telegrams that are to be sent for communication with the SMI I/O module.

The FBSMI_Master function block enables access to the WAGO-SMI-configurator while the PLC is running.

The SMI I/O module can be put into energy saving mode via the xSmiEnableEnergySaverMode input. If this input is active, only SMI commands for motions are sent out. After a send out command the energy saving mode will be left by the module. To entry energy saving mode again a new rising edge via xSmiEnableEnergySaverMode input is required. Via the xSmiEnergySaverMode output, the current status of the energy saving mode of the SMI I/O module is displayed.
The SMI I/O module can be put into energy saving mode via the xSmiEnableEnergySaverMode input. If the SMI I/O module is in energy saving mode, you can exit energy saving mode via the following travel commands:

- DOWN
- UP
- STOP
- Step DOWN
- Step UP
- Move into position
- Move into position 1
- Move into position 2

Sending additional SMI commands is disabled. If a travel command is sent, there must be a rising edge at the “xSmiEnableEnergySaverMode” input to change the SMI I/O module to energy saving mode.

An error can be identified by the current communication status as displayed at output sStatus. The xStateDigIn output signals the status of the digital input of the SMI I/O module.

**Note:** All SMI function blocks should be called up in cycles within the same program task as the FbSMI_Master function block. The assignment of SMI I/O modules to the FbSMI_Master module must be performed with constants; otherwise run-time errors can occur.

**Example:**

The input I_Port will be connected with the module, which is defined in the project structure. Therefore you select the user defined name of the module at the input.

Additional you have to fill in the number of the SMI module.

**Graphical Illustration**

![Graphical Interface of FbSMI_Master](image)

**Fig. 3.1: Graphical Interface of FbSMI_Master**

### 3.1.2 PrgSMIConfigurator (PRG)

**Function:**

Visualization to adjust module settings and configure drives within a view. The programm may only be used with the SMI master function block (FbSMI_Master).

**Note:** If bPortSMI and I_Port of a FbSMI_Master function block doesn’t match, an error message is shown in the status bar “No interface connected...”. Ensure that the relative number fit to the logical position of I_Port. Only a new configuration of the FbSMI_Master and a restart of the SPS application removes the error.
Graphical Illustration

![Graphical Interface of PrgSMIConfigurator](image)

**Fig. 3.2: Graphical Interface of PrgSMIConfigurator**

**Visualization:**

- **Configuration:** The *Configuration* visualization supports the startup of a SMI line and its drives. *PrgSMIConfigurator* must be integrated in order to use the visualization.

1. The relative number of the inserted SMI I/O module 753-1963x is selected via combo box. After this in input, the user’s attention is drawn to the change of bus with a security prompt, which can be acknowledged with [Yes] or [Cancel]. After the SMI I/O module is changed, the displayed list 10 is deleted, but not automatically updated. During a New addressing or Read system image operation, a change is not possible. If the selected SMI I/O module is accessed via an external tool, then all elements but 1, 17 and 18 are grayed out, and External access to the module is output.

2. The [Read] button updates all display fields in list 10. The displayed system image corresponds to the current status of the SMI line.

3. The [Write] button causes all comments in list 10 to be written to the system image. Individual selection of the comments is not possible.

### 3.1. 10 Communication
4. For [New] addressing, the existing list including comments is deleted, and all drives connected to the SMI I/O module receive a new address depending on their slave ID. The operation can take several minutes depending on the drives connected.

5. The [Expansion] is used when drives are to be added to an existing list or when removed drives are to be deleted. If the system expansion is started after replacing a drive, the new drive is written to the old address, and the comments are included.

6. The [Change] button is used to sort the drives. The button is only enabled if two drives are selected.

7. The [Delete] button is only active if at least one drive was marked as missing by Read system image (“Missing” is displayed in the comments) and one is selected via 9.

8. Buttons for manufacturer selection for controlling drives of only one manufacturer: If a drive is selected via 9, the manufacturer selection is deactivated. If multiple drives are selected, the manufacturer can again be selected within this group.

9. Button to select the drives: The button labeling corresponds to the slave address of the selected drive. Different functions can be used in the startup tool depending on the number of drives selected.

10. For New addressing and Read system image, the drives found appear in a list. It contains the assigned slave address, the slave ID and the manufacturer ID. To improve clarity, a comment on the installation location can be added (max. 12 characters) to each drive. The comments indicate drives that are not found (“Missing”) or address conflicts (“Conflict”).

11. Buttons for controlling one or more drives: The [Up] and [Down] motion commands are executed until the [Stop] button is pressed or the respective end position is reached. For the [On step] and [From step] motion commands, degree values can be specified for lamella adjustment in the adjacent input field.

12. The [Position] button can be used to move the drives to the position entered as a percentage in the neighboring field. The neighboring [Read] button can be used to display the current position of a drive.

13. The [Diagnostics] button can be used to query which motion commands are executed directly. Multiple namings are possible in the neighboring display field, e.g., “UP STOP” means at least one drive moves up and at least one drive stays. Only the “ALL xxx” messages affect all drives.

14. Pressing the [Pos 1] or [Pos 2] button moves the drives to the position saved in the drive. The [Read] buttons in the associated input fields cause the saved values to be displayed as percentages. The input fields and [Write] buttons can be used to save new values to a drive. The read and write functions are only enabled if an individual drive is selected. The buttons are otherwise disabled.

15. The number of drives found is shown in [Slaves].

16. Indication of the status of the SMI communication or messages of the SMI I/O module: Besides the events (Listed in eStatusbarmessage).

17. Tab to open visualization Configuration.

18. Tab to open visualization ModuleSettings.

19. Indication the status (WagoSysErrorBase.WagoTypes.eSeverity.error) of the IEC application: Besides the events (Listed in eStatus).

• Modulesetting: The Modulesetting visualization supports parameterization of an SMI I/O module.
This visualization offers the following control and display options:

1. Tab to open visualization Configuration.

2. Tab to open visualization ModuleSettings.

3. The relative number of the inserted SMI I/O module 753-1963x is selected via combox box. After this in input, the user’s attention is drawn to the change of bus with a security prompt, which can be acknowledged with [Yes] or [Cancel]. After the SMI I/O module is changed, the displayed list 10 is deleted, but not automatically updated. During a New addressing or Read system image operation, a change is not possible. If the selected SMI I/O module is accessed via an external tool, then all elements but 1, 17 and 18 are grayed out, and External access to the module is output.

4. The [Read] button in Action causes the current settings to be read from the SMI I/O module.

5. The [Write] button in Action causes the current settings to be written to the SMI I/O module.

6. The [Write] button in Factorysettings causes the settings of the SMI I/O module to be reset to the state they were in at the time of delivery.

7. The [Globally activate diagnostics] checkbox can be used to enable/disable the diagnostics of the SMI I/O module globally.

8. The [Activate internal data bus watchdog] checkbox is used to configure the behavior of the SMI I/O module in the event of an internal data bus error.

9. If the [In case of overrun, overwrite ring buffer data] box is checked, the oldest entry in the log buffer is written if the buffer is full and another entry needs to be stored in the log.
the [In case of overrun, overwrite ring buffer data] box is unchecked, no new entry is stored in the log when it is full.

10. If the [Activate auto-replace function] box is checked, the SMI I/O module automatically performs an auto replace operation when the conditions for this are met. Further information on auto replace can be found in the SMI I/O module manual. If the [Activate Auto-Replace-Function] box is unchecked, no automatic replacement is performed by the SMI I/O module.

11. The [Internal data bus timeout and control error monitoring] input field is used to specify the time span after which the SMI I/O module detects an internal data bus or controller failure.

12. The [Timeout energy saving mode] input field is used to configure the time span of the idle phase of the SMI line after which the SMI I/O module automatically goes into energy saving mode. Inputting “0 minutes” deactivates the automatic energy saving mode.

13. The [System delay] input field is used to specify the duration until readiness for operation of the start-up phase of the SMI drives.

14. Energy saving mode is linked to the digital output of the SMI Master Module. The digital output is controlled via the energy saving mode. The [Drive at DO = 24 V (on or off)] checkbox is used to configure the behavior of the digital output, so that either 0 V or 24 V is applied at the digital output when energy saving mode is activated. If the checkbox is selected, 0 V is applied at the digital output when energy saving mode is activated.

15. The [Selection of digital input] radio button configures the behavior in the event of a change of level specified by 15 on the CAGE CLAMP® of the SMI I/O module.

16. The [Logic of the input signal (break/make contact)] checkbox is used to configure 1-button operation. If the box is checked when the digital input is set to 1-button operation, the input signal is interpreted as a break contact.

17. The [Forced position] radio button is used to configure its behavior if it enters a forced position (e.g., wind alarm or internal data bus failure).

18. Display of the firmware version of the selected SMI I/O module.

19. Display of the hardware version of the selected SMI I/O module.

20. Indication of the status of the SMI communication or messages of the SMI I/O module: Besides the events (Listed in eStatusbarmessage).

3.2 20 Basic Commands

3.2.1 FbSMI_Diagnostic (FB)

Interface variables
### Scope

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer’s address</td>
</tr>
</tbody>
</table>

### Output

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td>eSMI_Diagnostic</td>
<td>eSMI_Diagnostic</td>
<td></td>
<td>Indication of motor failure</td>
</tr>
<tr>
<td>xMotor_Failure</td>
<td>BOOL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Function:

The FbSMI_Diagnostic function block makes it possible to initiate the request for an SMI diagnostic response. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The request for a diagnostic response is initiated by a rising edge at the xStart input.

The inputs xManufacturer and bAddress define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE
  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE
  
  bAddress = manufacturer address 1-15

- **Random addressing**
  
  xManufacturer = FALSE
  
  bAddress = 0-15

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by

---

3.2. 20 Basic Commands
the current communication status as displayed at output sStatus.

The eSMI_Diagnostic output returns the diagnostic response. The response can be used to deduce which motion commands (if any) are to be executed. A readout is also made to detect a possible drive failure. The status of this request is displayed at the xMotor_Failure output.

Graphical Illustration

![Graphical Interface of FbSMI_Diagnostic](image)

Fig. 3.3: Graphical Interface of FbSMI_Diagnostic

### 3.2.2 FbSMI_Down (FB)

#### Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer's address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in sStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

Function:

3.2. 20 Basic Commands
The FbSMI_Down function block sends commands for the DOWN motion command. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  - xManufacturer = TRUE
  - bAddress = 0

- **Manufacturer addressing**
  - xManufacturer = TRUE
  - bAddress = manufacturer address 1-15

- **Random addressing**
  - xManufacturer = FALSE
  - bAddress = 0-15

- **Group addressing**
  - xManufacturer = TRUE
  - bAddress = 0-15
  - wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

**Graphical Illustration**

![Graphical Interface of FbSMI_Down](image)

3.2.3 FbSMI_Move_Pos (FB)

**Interface variables**
### Function:

The FbSMI_Move_Pos function block moves a blind’s position. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE
  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE
  
  bAddress = manufacturer address 1-15
• Random addressing
  \[ x_{\text{Manufacturer}} = \text{FALSE} \]
  \[ b_{\text{Address}} = 0-15 \]

• Group addressing
  \[ x_{\text{Manufacturer}} = \text{TRUE} \]
  \[ b_{\text{Address}} = 0-15 \]
  \[ w_{\text{Group}} = \text{eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4} \]

Communication with the SMI interface is activated when the \[ x_{\text{Ready}} \] output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output \[ s_{\text{Status}} \].

**Graphical Illustration**

![Function Block Diagram](image)

**Fig. 3.5: Graphical Interface of FbSMI_Move_Pos**

### 3.2.4 FbSMI_Move_Pos1 (FB)

**Interface variables**
### Function:

The function blocks `FbSMI_Move_Pos1` or `FbSMI_Move_Pos2` are used to move the blind to one of the two positions that were configured at the motor. If the command is sent, the motor is shifted to the configured position. The function block may only be used with the SMI master function block (`FbSMI_Master`). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

The command is sent by a rising edge at the `xStart` input.

The three inputs `xManufacturer`, `bAddress` and `wGroup` define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  `xManufacturer = TRUE
  bAddress = 0`

- **Manufacturer addressing**
  
  `xManufacturer = TRUE
  bAddress = manufacturer address 1-15`

- **Random addressing**
  
  `xManufacturer = FALSE
  bAddress = 0-15`

- **Group addressing**
  
  `xManufacturer = TRUE`
bAddress = 0-15
wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

Graphical Illustration

![Graphical Interface of FbSMI_Move_Pos1](image)

3.2.5 FbSMI_Move_Pos2 (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer’s address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

Function:

3.2. 20 Basic Commands
The function blocks FbSMI_Move_Pos1 or FbSMI_Move_Pos2 are used to move the blind to one of the two positions that were configured at the drive. If the command is sent, the drive is shifted to the configured position. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE  
  bAddress = manufacturer address 1-15

- **Random addressing**
  
  xManufacturer = FALSE  
  bAddress = 0-15

- **Group addressing**
  
  xManufacturer = TRUE  
  bAddress = 0-15  
  wGroup = e.g., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

**Graphical Illustration**

![Graphical Interface of FbSMI_Move_Pos2](image)

**Fig. 3.7: Graphical Interface of FbSMI_Move_Pos2**

### 3.2.6 FbSMI_Move_Pos_Turn (FB)

**Interface variables**
### Function:

The `FbSMI_Move_Pos_Turn` function block is used to move the blind to the height position. After reaching the height position, the lamella opening angle can be adjusted. The function block may only be used with the SMI master function block (`FbSMI_Master`). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

A sequence command (two commands in one telegram) is sent to the SMI drive by a rising edge at the `xStart` input. The commands are executed by the drive one after the other. First, the motion command occurs on the height position indicated at the `rPosition` input. The lamella opening angle is then adjusted to the angle set at the `rRad` input. The direction of the lamella adjustment is defined by the value at the `xTurn_Up_Down` input.

The inputs `xManufacturer` and `bAddress` define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

```markdown
- **BYTE**
  - `bModule_753_163x` (Initial: 1) Relative number of the inserted SMI I/O module 753-163x in the node
  - `xStart` (Type: BOOL) A rising edge starts the command
  - `xManufacturer` (Type: BOOL) Activation of manufacturer address
  - `bAddress` (Type: BYTE) Slave address or manufacturer's address

- **REAL**
  - `rPosition` (Type: REAL) Drive shifts up by x angular degrees

- **BOOL**
  - `xTurn_Up_Down` (Type: BOOL) Direction of adjustment of the lamella opening angle:
    - TRUE = Angle adjustment down
    - FALSE = Angle adjustment up
    - etc.

- **WagoSysErrorBase.FbResult** (Type: STRING) Error object. (Listed in eStatus) The content of the error object could be displayed via the `FbShowResult` from the `WagoSysErrorBase` library.

- **STRING** (Type: STRING) Error description as string (Listed in eStatus)

- **BOOL** (Type: BOOL) TRUE = communication deactivated, FALSE = communication activated
• Broadcast
  
  xManufacturer = TRUE
  bAddress = 0

• Manufacturer addressing
  
  xManufacturer = TRUE
  bAddress = manufacturer address 1-15

• Random addressing
  
  xManufacturer = FALSE
  bAddress = 0-15

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

Graphical Illustration

![Fig. 3.8: Graphical Interface of FbSMI_Move_Pos_Turn](image)

3.2.7 FbSMI_Read_Pos (FB)

Interface variables
### Function:

The `FbSMI_Read_Pos` function block requests the current position of a blind. The function block may only be used with the SMI master function block (`FbSMI_Master`). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

The command for reading out the current position is sent to the SMI slave by a rising edge at the `xStart` input. The position value is displayed at the `rPosition` output. Reading out the position is only possible in random addressing. The slave address (0-15) that is to be read out is indicated at the `bAddress` input.

Communication with the SMI interface is activated when the `xReady` output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output `sStatus`.

#### Graphical Illustration

![Function Block Diagram](image)

**Fig. 3.9: Graphical Interface of FbSMI_Read_Pos**

#### 3.2.8 FbSMI_Read_Pos1 (FB)

**Interface variables**

### Table 3.20: Basic Commands

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in <code>eStatus</code>) The content of the error object could be displayed via the <code>FbShowResult</code> from the <code>WagoSysErrorBase</code> library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in <code>eStatus</code>)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>rPosition</td>
<td>REAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Function:

The function blocks `FbSMI_Read_Pos1` and `FbSMI_Read_Pos2` request the values from both fixed positions that were configured at the drive. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

The command for reading out the fixed position is sent by a rising edge at the `xStart` input. The position value is displayed at the `rPosition` output. Reading out the position is only possible in random addressing. The slave address (0-15) that is to be read out is indicated at the `bAddress` input.

Communication with the SMI interface is activated when the `xReady` output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output `sStatus`.

### Graphical Illustration

![FunctionBlock](image)

**Fig. 3.10: Graphical Interface of FbSMI_Read_Pos1**

### 3.2.9 FbSMI_Read_Pos2 (FB)

**Interface variables**

---

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in <code>eStatus</code>) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in <code>eStatus</code>)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>rPosition</td>
<td>REAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Function:

The function blocks `FbSMI_Read_Pos1` and `FbSMI_Read_Pos2` request the values from both fixed positions that were configured at the drive. The function block may only be used with the SMI master function block (`FbSMI_Master`). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

The command for reading out the fixed position is sent by a rising edge at the `xStart` input. The position value is displayed at the `rPosition` output. Reading out the position is only possible in random addressing. The slave address (0-15) that is to be read out is indicated at the `bAddress` input.

Communication with the SMI interface is activated when the `xReady` output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output `sStatus`.

### Graphical Illustration

![Graphical Interface of FbSMI_Read_Pos2](image)

### 3.2.10 FbSMI_Read_Syn (FB)

**Interface variables**
### Function:

The `FbSMI_Read_Syn` function block reads out the manufacturer ID and the motor type of an SMI slave. The function block may only be used with the SMI master function block (`FbSMI_Master`). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input.

The command for reading out the manufacturer ID and the motor type is sent by a rising edge at the `xStart` input. The manufacturer ID is displayed at the `bManufac_ID` output, and the motor type is displayed at the `bMotor_Type` output. Reading out these values is only possible in random addressing. The slave address (0-15) that is to be read out is indicated at the `bAddress` input.

Communication with the SMI interface is activated when the `xReady` output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output `sStatus`.

**Graphical Illustration**

![Graphical Interface of FbSMI_Read_Syn](image)

Fig. 3.12: Graphical Interface of FbSMI_Read_Syn

### 3.2.11 FbSMI_Step_Down (FB)

**Interface variables**

---

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td><code>bModule_753_163x</code></td>
<td><code>BYTE</code></td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td><code>xStart</code></td>
<td><code>BOOL</code></td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td><code>bAddress</code></td>
<td><code>BYTE</code></td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td>Output</td>
<td><code>oStatus</code></td>
<td><code>WagoSysErrorBase.FbResult</code></td>
<td></td>
<td>Error object. (Listed in <code>eStatus</code>) The content of the error object could be displayed via the <code>FbShowResult</code> from the <code>WagoSysErrorBase</code> library.</td>
</tr>
<tr>
<td></td>
<td><code>sStatus</code></td>
<td><code>STRING</code></td>
<td></td>
<td>Error description as string (Listed in <code>eStatus</code>)</td>
</tr>
<tr>
<td></td>
<td><code>xReady</code></td>
<td><code>BOOL</code></td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td><code>bManufac_ID</code></td>
<td><code>BYTE</code></td>
<td></td>
<td>Manufacturer ID</td>
</tr>
<tr>
<td></td>
<td><code>bMotor_Type</code></td>
<td><code>BYTE</code></td>
<td></td>
<td>Motor type (manufacturer specific)</td>
</tr>
</tbody>
</table>
WagoAppSMI, Release 1.0.1.3

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer’s address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>bRad</td>
<td>BYTE</td>
<td></td>
<td>Motor shifts down by x angular degrees</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the Fb-ShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

Function:

The FbSMI_Strip_Down function block sends the command to shift the lamella. If the command is sent, the motor is shifted downward by the angular degree set at the bRad input. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- Broadcast
  
xManufacturer = TRUE
  
bAddress = 0

- Manufacturer addressing
  
xManufacturer = TRUE
  
bAddress = manufacturer address 1-15

- Random addressing
  
xManufacturer = FALSE
  
bAddress = 0-15

- Group addressing

3.2. 20 Basic Commands
xManufacturer = TRUE
bAddress = 0-15
wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the \texttt{xReady} output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output \texttt{sStatus}.

**Graphical Illustration**

![Graphical Interface of FbSMI_Step_Down](image)

Fig. 3.13: Graphical Interface of FbSMI_Step_Down

### 3.2.12 FbSMI_Step_Up (FB)

**Interface variables**
The FbSMI_Step_Up function block sends the command to shift the lamella. If the command is sent, the motor is shifted up by the angular degree set at the bRad input. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- Broadcast
  
  xManufacturer = TRUE
  bAddress = 0

- Manufacturer addressing
  
  xManufacturer = TRUE
  bAddress = manufacturer address 1-15

- Random addressing
  
  xManufacturer = FALSE
  bAddress = 0-15

- Group addressing

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td>A rising edge starts the command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td>Activation of manufacturer address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td>Slave address or manufacturer’s address</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td>Drive shifts up by x angular degrees</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td>Error description as string (Listed in eStatus)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>
xManufacturer = TRUE
bAddress = 0-15
wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

Graphical Illustration

![Graphical Interface of FbSMI_Step_Up](image)

**3.2.13 FbSMI_Stop (FB)**

**Interface variables**

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer’s address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

3.2. 20 Basic Commands
Function:

The FbSMI_Stop function block can be used to stop the movement of a blind. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE
  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE
  
  bAddress = manufacturer address 1-15

- **Random addressing**
  
  xManufacturer = FALSE
  
  bAddress = 0-15

- **Group addressing**
  
  xManufacturer = TRUE
  
  bAddress = 0-15
  
  wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

**Graphical Illustration**

![Graphical Interface of FbSMI_Stop](image)

Fig. 3.15: Graphical Interface of FbSMI_Stop

### 3.2.14 FbSMI_Up (FB)

**Interface variables**
### Scope

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address or manufacturer’s address</td>
</tr>
<tr>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Function:

The FbSMI_Up function block sends commands for the UP motion command. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command is sent by a rising edge at the xStart input.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE
  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE
  
  bAddress = manufacturer address 1-15

- **Random addressing**
  
  xManufacturer = FALSE
  
  bAddress = 0-15

- **Group addressing**
  
  xManufacturer = TRUE
  
  bAddress = 0-15
  
  wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

### Output

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>
Communication with the SMI interface is activated when the \texttt{xReady} output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output \texttt{sStatus}.

Graphical Illustration

![Graphical Interface of FbSMI_Up](image)

**FunctionBlock**

\begin{Verbatim}
FbSMI_Up
\end{Verbatim}

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rPosition</td>
<td>REAL</td>
<td></td>
<td>Position 0 - 100%</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in \texttt{eStatus}) The content of the error object could be displayed via the \texttt{FbShowResult} from the \texttt{WagoSysErrorBase} library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in \texttt{eStatus})</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

**3.2.15 FbSMI_Write_Pos1 (FB)**

Interface variables

Function:

The function blocks \texttt{FbSMI_Write_Pos1} and \texttt{FbSMI_Write_Pos2} write both fixed positions that were configured at the drive. The function block may only be used with the SMI master function block (\texttt{FbSMI_Master}). The relative number of the inserted SMI I/O module 753-163x is selected via the \texttt{bModule_753_163x} input.
The command is sent by a rising edge at the \textit{xStart} input.

The three inputs \textit{xManufacturer}, \textit{bAddress} and \textit{wGroup} define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  \begin{itemize}
  \item \textit{xManufacturer} = TRUE
  \item \textit{bAddress} = 0
  \end{itemize}

- **Manufacturer addressing**
  \begin{itemize}
  \item \textit{xManufacturer} = TRUE
  \item \textit{bAddress} = manufacturer address 1-15
  \end{itemize}

- **Random addressing**
  \begin{itemize}
  \item \textit{xManufacturer} = FALSE
  \item \textit{bAddress} = 0-15
  \end{itemize}

- **Group addressing**
  \begin{itemize}
  \item \textit{xManufacturer} = TRUE
  \item \textit{bAddress} = 0-15
  \item \textit{wGroup} = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4
  \end{itemize}

Communication with the SMI interface is activated when the \textit{xReady} output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output \textit{sStatus}.

**Graphical Illustration**

![Graphical Interface of FbSMI_Write_Pos1](image)

**Fig. 3.17: Graphical Interface of FbSMI_Write_Pos1**

### 3.2.16 FbSMI_Write_Pos2 (FB)

**Interface variables**
### Function:

The function blocks FbSMI_Write_Pos1 and FbSMI_Write_Pos2 write both fixed positions that were configured at the drive. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the `bModule_753_163x` input. The command is sent by a rising edge at the `xStart` input.

The three inputs `xManufacturer`, `bAddress` and `wGroup` define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE  
  bAddress = manufacturer address 1-15

- **Random addressing**
  
  xManufacturer = FALSE  
  bAddress = 0-15

- **Group addressing**
  
  xManufacturer = TRUE  
  bAddress = 0-15  
  wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>rPosition</td>
<td>REAL</td>
<td></td>
<td>Position 0 - 100%</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in <code>eStatus</code>) The content of the error object could be displayed via the <code>FbShowResult</code> from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in <code>eStatus</code>)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

3.2. 20 Basic Commands 33
Communication with the SMI interface is activated when the $x\text{Ready}$ output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output $s\text{Status}$.

**Graphical Illustration**

![Graphical Interface of FbSMI_Write_Pos2](image)

**3.3 30 Adressing**

### 3.3.1 FbSMI_Addressing (FB)

**Interface variables**

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xNewAdressing</td>
<td>BOOL</td>
<td></td>
<td>Start signal for new addressing of SMI slaves</td>
</tr>
<tr>
<td></td>
<td>xExtendedAdressing</td>
<td>BOOL</td>
<td></td>
<td>Start signal for addressing SMI slaves for system extension</td>
</tr>
<tr>
<td></td>
<td>xAbortAdressing</td>
<td>BOOL</td>
<td></td>
<td>Reset or stop the addressing</td>
</tr>
<tr>
<td></td>
<td>xSwap_Address</td>
<td>BOOL</td>
<td></td>
<td>Swap addresses</td>
</tr>
<tr>
<td></td>
<td>bSwap-Slave_Adr_1</td>
<td>BYTE</td>
<td></td>
<td>First address swap (Range of values: 0-15)</td>
</tr>
<tr>
<td></td>
<td>bSwap-Slave_Adr_2</td>
<td>BYTE</td>
<td></td>
<td>Second address swap (Range of values: 0-15)</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in $e\text{Status}$) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in $e\text{Status}$)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>bNumber</td>
<td>BYTE</td>
<td></td>
<td>Number of connected SMI slaves</td>
</tr>
<tr>
<td>Inout</td>
<td>typSMI_SlaveIDList</td>
<td>IDpSMI_SlaveIDList</td>
<td></td>
<td>Structure of the connected SMI line see typSMI_SlaveIDList</td>
</tr>
</tbody>
</table>

**Function:**

The FbSMI_Addressing module is used to assign drive addresses. Each drive has a unique SlaveID. It is normally unknown at commissioning and is automatically determined by the module. With the specified SlaveID, one of the 16 addresses (0-15) is assigned to each drive. The function block may only be used with the SMI master.
function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The following functions are carried out by the function block:  - If a rising edge is identified at the xNewAddressing input, then all drives connected to the SMI interface are assigned an address (0-15). - When a rising edge is identified at input xExtendedAddressing, only the SMI drives newly added are addressed. - An ongoing addressing process can be interrupted from a rising edge at input xReset. - After automatic addressing is completed, the addresses can be manually swapped or moved to an open address. A rising edge at the input xSwap_Address means that the addresses specified at the inputs bSwapSlave_Adr_1 and bSwapSlave_Adr_2 are swapped.

The typSMI_SLAVEID variable contains a list in which the reference between the slave addresses (0-15) and the 32-bit SlaveIDs is represented. In addition, the list contains the manufacture ID of the drives and a possibly assigned comment. This is illustrated in the following:

```plaintext
typSMI_SLAVEID_163x
  typSMI_SLAVEID_163x[0] Slave address = 0
    dwSlaveID = 16#15330001
    bManuf_ID = 16#06
    sComment = 'Motor A'
  typSMI_SLAVEID_163x[1]
    dwSlaveID = 16#15340001
    bManuf_ID = 16#06
    sComment = 'Motor B'
  typSMI_SLAVEID_163x[2]
    dwSlaveID = 16#15340007
    bManuf_ID = 16#06
    sComment = 'Motor C'
  typSMI_SLAVEID_163x[3] Slave address = 3
    dwSlaveID = 16#00000000
    bManuf_ID = 16#00
    sComment = '
```

Communication with the SMI I/O module is activated when the xReady output is FALSE. An active running addressing operation is signaled in addition. During this time, the other SMI modules are disabled in the program call. After the communication is completed, the output switches to TRUE. Addressing errors are displayed at the output sStatus.

The output bNumber shows the number of drives found during the automatic addressing procedure.

**Note:** Please note that the other SMI modules are blocked in the program call during automatic addressing. It is then impossible, for example, to stop an ongoing motion command of a drive.
Note: An addressing operation can take several minutes. The time depends on the number of SMI slaves connected and the program cycle time. A signal change from FALSE to TRUE of the output “xReady” indicates that addressing is complete.

Graphical Illustration

![Fig. 3.19: Graphical Interface of FbSMI_Addressing](image)

### 3.3.2 FbSMI_SlaveAdr_Write (FB)

#### Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bManuf_Adr</td>
<td>BYTE</td>
<td></td>
<td>Manufacturer’s address</td>
</tr>
<tr>
<td></td>
<td>dwSlave_ID</td>
<td>DWORD</td>
<td></td>
<td>32 bit SlaveID</td>
</tr>
<tr>
<td></td>
<td>bSlave_Adr</td>
<td>BYTE</td>
<td></td>
<td>New slave address (Range of values: 0-15)</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

**Function:**

The FbSMI_SlaveAdr_Write function block is used to write a new slave address when the 32-bit SlaveID of the motor is known. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command for writing the slave address is sent by a rising edge at the xStart input. Manufacturer addressing must be used for this function block. The bManuf_Adr parameter defines the manufacturer’s address. The necessary and unique 32 bit SlaveID of the motor must be entered at the dwSlave_ID input, and the new slave address is entered at the bSlave_Adr input.

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal.

An error can be identified by the current communication status as displayed at output sStatus.

Graphical Illustration

3.3. 30 Adressing
### 3.3.3 FbSMI_SlaveId_Read (FB)

#### Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>32 bit SlaveID</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSys-</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sError-Base.FbResult</td>
<td></td>
<td>be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>dwSlave_ID</td>
<td>DWORD</td>
<td></td>
<td>32Bit motor slave-ID</td>
</tr>
</tbody>
</table>

#### Function:

The FbSMI_SlaveAddr_Write function block is used to write a new slave address when the 32-bit SlaveID of the motor is known. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command for writing the slave address is sent by a rising edge at the xStart input. Manufacturer addressing must be used for this function block. The bManuf_Adr parameter defines the manufacturer’s address. The necessary and unique 32 bit SlaveID of the motor must be entered at the dwSlave_ID input, and the new slave address is entered at the bSlave_Adr input.

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal.

An error can be identified by the current communication status as displayed at output sStatus.

#### Graphical Illustration

![Graphical Interface of FbSMI_SlaveAddr_Write](image_url)
3.3.4 FbSMI_SlaveId_Search (FB)

Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bManuf_Adr</td>
<td>BYTE</td>
<td></td>
<td>Manufacturer’s address</td>
</tr>
<tr>
<td></td>
<td>dwSearch_ID</td>
<td>DWORD</td>
<td>16#80000000</td>
<td>Search specification for the 32 bit SlaveID. Default setting = 16#8000000</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>xLT</td>
<td>BOOL</td>
<td></td>
<td>There are shorter IDs</td>
</tr>
<tr>
<td></td>
<td>xGT</td>
<td>BOOL</td>
<td></td>
<td>There are longer IDs</td>
</tr>
<tr>
<td></td>
<td>xEQ</td>
<td>BOOL</td>
<td></td>
<td>The slave ID being searched for exists</td>
</tr>
<tr>
<td></td>
<td>xNO_Adr</td>
<td>BOOL</td>
<td></td>
<td>The slave address of the drive is 0</td>
</tr>
</tbody>
</table>

Function:

The FbSMI_SlaveId_Search function block can be used to identify the individual SlaveIDs that have been activated on the SMI interface. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command for searching for a SlaveID is sent by a rising edge at the xStart input. Manufacturer addressing must be used for this function block. The bManuf_Adr parameter defines the manufacturer’s address. The SlaveID being sought must be specified at the dwSearch_ID input.

The outputs xLT, xGT and xEQ indicate whether there are motors in the SMI installation that have a shorter, longer or identical SlaveIDs as the number specified at the dwSearch_ID input. Relevant algorithms can therefore be used to identify all activated Slave_IDs of the SMI motors.

The TRUE signal at the xNO_Adr output signals that all motors in the SMI installation have a slave address of 0. The system thus has an initialization stage, since the default slave address of the SMI motors is always 0 in its initial state.

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal.

An error can be identified by the current communication status as displayed at output sStatus.

Graphical Illustration
### 3.3.5 FbSMI_SystemImage (FB)

#### Interface variables

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node.</td>
</tr>
<tr>
<td></td>
<td>xReadSystemImage</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command to read the system image.</td>
</tr>
<tr>
<td></td>
<td>xWriteComments</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command to write the comments.</td>
</tr>
<tr>
<td></td>
<td>bDeleteAddress</td>
<td>BYTE</td>
<td></td>
<td>Address to be deleted from system image</td>
</tr>
<tr>
<td></td>
<td>xDeleteAddress</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command to delete a missing SMI drive.</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSys-Base.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td>Inout</td>
<td>typSMI_SlaveID</td>
<td>typSMI_SlaveIDList</td>
<td></td>
<td>Structure of the connected SMI line see typSMI_SlaveIDList.</td>
</tr>
</tbody>
</table>

**Function:**

The FbSMI_SystemImage function block is used to create or edit a current image of the SMI drives connected to the SMI I/O module 753-163x. This function block can only be used with the SMI Master module (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The following functions are carried out by the function block in the order given below (if multiple queries are pending simultaneously):

- If a rising edge is detected on the xWriteComments input, the command to write the comments contained in the typSMI_SlaveID input/output is sent.
- If a rising edge is detected on the xReadSystemImage input, the command to read the system image is sent.
- If a rising edge is detected on the xDeleteAddress input, the SMI address on the bAddress input is deleted from the system image if it was identified as missing from the SMI I/O module 163x by a cyclic scan.

### 3.3. 30 Addressing
Communication with the SMI I/O module is enabled when the $x_{\text{Ready}}$ output is FALSE. After the communication is completed, the output switches to TRUE. An error can be identified by the current communication status as displayed at output $s_{\text{Status}}$.

**Graphical Illustration**

![Graphical Interface of FbSMI_SystemImage](image)

Fig. 3.23: Graphical Interface of FbSMI_SystemImage

### 3.4 40 Motor

#### 3.4.1 FbSMI_Motor (FB)

**Interface variables**
<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave Addresss</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xUp</td>
<td>BOOL</td>
<td></td>
<td>Switch command Blind/Lamella UP</td>
</tr>
<tr>
<td></td>
<td>xDown</td>
<td>BOOL</td>
<td></td>
<td>Switch command Blind/Lamella DOWNn</td>
</tr>
<tr>
<td></td>
<td>xSafetyPosition</td>
<td>BOOL</td>
<td></td>
<td>Signal input: safety position</td>
</tr>
<tr>
<td></td>
<td>xLockBlind</td>
<td>BOOL</td>
<td></td>
<td>Blind lock, see warning below!</td>
</tr>
<tr>
<td></td>
<td>xSetPosition</td>
<td>BOOL</td>
<td></td>
<td>Move into position</td>
</tr>
<tr>
<td></td>
<td>rSetPosition_Blind</td>
<td>REAL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>rSetPosition_Lamella</td>
<td>REAL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>xMoveToShadowPosition</td>
<td>BOOL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>rShadowPosition_Blind</td>
<td>REAL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>rShadowPosition_Lamella</td>
<td>REAL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>xMoveToShadowPosition</td>
<td>BOOL</td>
<td></td>
<td>Move to shadow position</td>
</tr>
<tr>
<td></td>
<td>xSetOverride</td>
<td>BOOL</td>
<td></td>
<td>Set the manual override</td>
</tr>
<tr>
<td></td>
<td>xResetOverride</td>
<td>BOOL</td>
<td></td>
<td>Reset the manual override</td>
</tr>
</tbody>
</table>

**Height position of the Blind**

- 0% = Upper end position
- 100% = Lower end position

**Lamella position of the Blind**

- 0% = Upper end position
- 100% = Lower end position

**Height position of the Blind**

- 0% = Upper end position
- 100% = Lower end position

**Lamella position of the Blind**

- 0% = Upper end position
- 100% = Lower end position

**3.4. 40 Motor**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>typSMI_Config</td>
<td>typSMI_Config</td>
<td>SMI - Configuration data</td>
</tr>
<tr>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td>Error object. (Listed</td>
</tr>
</tbody>
</table>
Function:

The FbSMI_Motor function block controls Blinds with an SMI interface. The module provides the following control functions:

- UP/DOWN motion command and lamella shift
- Move to a defined shadow position
- Move to a defined lamella position
- Move to safety position with interlocking feature (e.g., wind alarm)
- Blind lock
- Selection possibility between manual-automatic mode
- Acknowledgement of the Blind position

**Warning:** Do not send movement commands in PLC mode when controller lock is active! The movement commands executed via the SMI Configurator have higher priority than any controller lock set in the PLC software. Therefore, it is possible to move the SMI drives even when the controller lock is active. This only applies if the SMI Configurator is connected to an SMI master module via PLC mode. Any time you send movement commands with the SMI Configurator via PLC mode, first check whether the controller lock is active. Do not send movement commands with the SMI Configurator as long as the controller lock is active.

The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The two inputs bAddress and wGroup define whether the function block communicates with one or several SMI slaves. Performing group addressing is prioritized.

The following addressing types are possible:

- Random addressing
  
  bAddress = SlaveAdr (0-15)  
  wGroup = 0  
- Group addressing
  
  bAddress = 0  
  wGroup = eg., (2#0000 0000 0001 0001) for slave addresses 0 and 4

The SMI drive is controlled by two button inputs: xUP and xDOWN. An extended button press on one of these inputs (longer than the configured time, typSMI_Config.tShortPressTime) causes the drive to move to the upper or lower end position (configured at the drive). If the input signal is shorter than the configured time, a STOP telegram or telegram indicating lamella shift is sent.

The configurable safety position (upper end position) of the Blind (e.g., for the wind alarm) can be controlled via the xSafetyPosition input. When the blind has been moved to the safety position, it cannot be manually controlled until the xSafetyPosition input has been reset. For safety reasons, the function block sends the UP motion command every 3 minutes if the safety position is activated.

The xLockBlind can lock the blind control. Current motion commands are not canceled. If the xLock input is TRUE, the blind can only be moved to the safety position (e.g., for the wind alarm).

**Warning:** Do not send movement commands in PLC mode when controller lock is active! The movement commands executed via the SMI Configurator have higher priority than any controller lock set in the PLC software. Therefore, it is possible to move the SMI drives even when the

3.4. 40 Motor
controller lock is active. This only applies if the SMI Configurator is connected to an SMI master module via PLC mode. Any time you send movement commands with the SMI Configurator via PLC mode, first check whether the controller lock is active. Do not send movement commands with the SMI Configurator as long as the controller lock is active.

A rising edge at the xSetPosition input means that a motion command to the positions specified at the rSetPosition_Blind and rSetPosition_Blind inputs is executed.

The xMoveToShadowPosition input is primarily used to bring the blind into a shadow position. If the signal at this input is TRUE, the drive moves to the configured height and lamella angle position (rShadowPosition_Blind and rShadowPosition_Lamella). The height position is always approached from below to ensure a unique lamella position. As long as the xMoveToShadowPosition input signal is TRUE, value changes in the rShadowPosition_Blind and rShadowPosition_Lamella inputs are tracked. A minimum value change of 5% is considered as the hysteresis.

The automatic sun protection can be overridden. In other words, new commands via the xMoveToShadowPosition, rShadowPosition_Blind and rShadowPosition_Lamella inputs are not evaluated. The automatic sun protection is overridden for the configured time typSMI_Config_163x.tDisableAutomatic if:

1. A motion command was initiated via one of the xUp or xDown inputs.
2. A position was approached via the xSetPosition input.
3. The xSetOverride input with signal TRUE is connected. It should be noted that the time only runs if the signal is switched to FALSE again. That means that the automatic sun protection can be overridden longer than the time set.

Configuration parameters:

The typSMI_Config input variable includes all necessary configuration parameters for drive control:

- typSMI_Config.xAutoMoveUp defines if the UP command should be sent after completing the manual override (falling edge at the xAutomaticOverride output) or if the position of the drive should be maintained.
- typSMI_Config.bRadMechanicReverse is the value in angular degree for compensation of the mechanical dead time. Due to tightening of the webbing, a dead time elapses until the lamella first responds.
- typSMI_Config.bType defines the blind type and displays how the blind moves. The blind type is identified by the position of the lamella per direction of motion: Type 1: down closed / up open
- typSMI_Config.tShortPressTime specifies the extended period allocated to push the button.
- typSMI_Config.tDisableAutomatic specified the time for overriding the automatic function.
- typSMI_Config.rAngle_of_Tilt is the parameter for the possible adjustable range of the lamella angle from completely open to completely closed.
- typSMI_Config.bRad_MotorStep defines by how many angular degrees the lamellas should be shifted in the case of a brief button push.

Communication with the SMI interface is activated when the xReady output is FALSE. After the communication is completed, the output switches to the TRUE signal. An error can be identified by the current communication status as displayed at output sStatus.

The rPosition_Blind output returns the current position of the blind as an acknowledgement. If the function block recognizes a current motion command, the current position is requested every two seconds. If no motion command can be found on the function block, the position request recurs.
in 30-second cycles. The lamella position is reported via the `rPosition_Lamella` output. The lamella position is not returned as a direct value from the drive, but is calculated internally by the function block. An undefined position (e.g., after resetting the controller) appears as the value 255.

**Note:**
- In group addressing, the position acknowledgement always occurs via the smallest individual address within the group, since the position request cannot be carried out using a group command. Thus, using this function block means it cannot be determined whether a group’s motors have different positions.

**Note:**
- The positions are always approached from below. This means that the drive initially starts positioning from a lower position than the target position so that the target position can then be reached using an UP motion command. In this way, an exact lamella position is guaranteed.

**Note:**
- The `rShadowPosition_Blind` and `rShadowPosition_Lamella` position parameters should always be written synchronously when the `xMoveToShadowPosition` input is set because a STOP command is triggered for each change in position on one of the inputs.

**Note:**
- The `typSMI_Config_163x` structure should be declared RETAIN PERISISTENT, so that set parameters are retained after a controller reset or a program download.

**Note:**
- The module currently supports blind type 1 only.

**Graphical Illustration**

---

3.4. 40 Motor
### 3.5 50 Parametrization

#### 3.5.1 FbSMI_Read_Par (FB)

**Interface variables**

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td></td>
<td>bLength</td>
<td>BYTE</td>
<td>1</td>
<td>Byte length of the parameter value that is to be read. Value range: 1, 2 or 4. Default setting = 1</td>
</tr>
<tr>
<td></td>
<td>wPar_Addr</td>
<td>WORD</td>
<td></td>
<td>Parameter address. Value range: 16#00 - 16#FFF</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSys-</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
<tr>
<td></td>
<td>dwParameter</td>
<td>DWORD</td>
<td></td>
<td>Parameter value</td>
</tr>
</tbody>
</table>

**Function:**

The FbSMI_Read_Par function block reads manufacturer-specific parameters of the motors. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command for reading out a slave parameter is sent by a rising edge at the xStart input. This requires the designation of the parameter address (wPar_Addr) and the length (bLength) of the parameter value. Parame-
ters are either 1 byte, 2 bytes or 4 bytes. Reading out the values is only possible in random addressing. The slave address (0-15) that is to be read out is indicated at the `bAddress` input.

The `dwParameter` output returns the read out parameter value of the SMI slave.

Communication with the SMI interface is activated when the `xReady` output is FALSE. After the communication is completed, the output switches to the TRUE signal.

An error can be identified by the current communication status as displayed at output `sStatus`.

**Graphical Illustration**

![FunctionBlock](image)

Fig. 3.25: Graphical Interface of FbSMI_Read_Par

### 3.5.2 FbSMI_Write_Par (FB)

**Interface variables**
<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>bModule_753_163x</td>
<td>BYTE</td>
<td>1</td>
<td>Relative number of the inserted SMI I/O module 753-163x in the node</td>
</tr>
<tr>
<td></td>
<td>xStart</td>
<td>BOOL</td>
<td></td>
<td>A rising edge starts the command</td>
</tr>
<tr>
<td></td>
<td>xManufacturer</td>
<td>BOOL</td>
<td></td>
<td>Activation of manufacturer address</td>
</tr>
<tr>
<td></td>
<td>bAddress</td>
<td>BYTE</td>
<td></td>
<td>Slave address</td>
</tr>
<tr>
<td></td>
<td>wGroup</td>
<td>WORD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>bLength</td>
<td>BYTE</td>
<td>1</td>
<td>Byte length of the parameter value that is to be read. Value range: 1, 2 or 4. Default setting = 1</td>
</tr>
<tr>
<td></td>
<td>bPar_Addr</td>
<td>BYTE</td>
<td></td>
<td>Parameter address. Value range: 16#00 - 16#FFF</td>
</tr>
<tr>
<td></td>
<td>dwParameter</td>
<td>DWORD</td>
<td></td>
<td>Parameter value</td>
</tr>
<tr>
<td>Output</td>
<td>oStatus</td>
<td>WagoSysErrorBase.FbResult</td>
<td></td>
<td>Error object. (Listed in eStatus) The content of the error object could be displayed via the FbShowResult from the WagoSysErrorBase library.</td>
</tr>
<tr>
<td></td>
<td>sStatus</td>
<td>STRING</td>
<td></td>
<td>Error description as string (Listed in eStatus)</td>
</tr>
<tr>
<td></td>
<td>xReady</td>
<td>BOOL</td>
<td>TRUE</td>
<td>TRUE = communication deactivated, FALSE = communication activated</td>
</tr>
</tbody>
</table>

**Function:**

The FbSMI_Write_Par function block writes manufacturer-specific parameters into the motors. The function block may only be used with the SMI master function block (FbSMI_Master). The relative number of the inserted SMI I/O module 753-163x is selected via the bModule_753_163x input.

The command for writing a dwParameter slave parameter is sent by a rising edge at the xStart input. This requires the designation of the parameter address wPar_Addr and the length bLength of the parameter value. Parameters are either 1 byte, 2 bytes or 4 bytes.

The three inputs xManufacturer, bAddress and wGroup define whether the function block communicates with one or several SMI slaves.

The following addressing types are possible:

- **Broadcast**
  
  xManufacturer = TRUE
  
  bAddress = 0

- **Manufacturer addressing**
  
  xManufacturer = TRUE
  
  bAddress = manufacturer address 1-15
• Random addressing
  \( x_{\text{Manufacturer}} = \text{FALSE} \)
  \( b_{\text{Address}} = 0-15 \)

• Group addressing
  \( x_{\text{Manufacturer}} = \text{TRUE} \)
  \( b_{\text{Address}} = 0-15 \)
  \( w_{\text{Group}} = \text{eg., (2}\#0000 0000 0001 0001) \text{ for slave addresses 0 and 4} \)

Communication with the SMI interface is activated when the \( x_{\text{Ready}} \) output is \text{FALSE}. After the communication is completed, the output switches to the \text{TRUE} signal.

An error can be identified by the current communication status as displayed at output \( s_{\text{Status}} \).

**Graphical Illustration**

![Graphical Interface of FbSMI_Write_Par](image)

Fig. 3.26: Graphical Interface of FbSMI_Write_Par
29 Types

4.1 typSMI_Adressing (STRUCT)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>uiSlaveId</td>
<td>UINT</td>
</tr>
<tr>
<td>bManufacturerId</td>
<td>BYTE</td>
</tr>
<tr>
<td>sComment</td>
<td>STRING</td>
</tr>
</tbody>
</table>

4.2 typSMI_Config (STRUCT)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Initial</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>xAutoMoveUp</td>
<td>BOOL</td>
<td>FALSE</td>
<td>Automatic UP-Command after override state</td>
</tr>
<tr>
<td>bRadMechanicReverse</td>
<td>BYTE</td>
<td>0</td>
<td>Rad offset for mechanical reverse</td>
</tr>
<tr>
<td>bType</td>
<td>BYTE</td>
<td>1</td>
<td>Type of blind</td>
</tr>
<tr>
<td>tShortPressTime</td>
<td>TIME</td>
<td>TIME#500ms</td>
<td>#500ms</td>
</tr>
<tr>
<td>tDisableAutomatic</td>
<td>TIME</td>
<td>TIME#60m0s0ms</td>
<td>Period of time where automatic is disabled due to manual control</td>
</tr>
<tr>
<td>rAngle_of_Tilt</td>
<td>REAL</td>
<td>90</td>
<td>Total angle of tilt for lamella</td>
</tr>
<tr>
<td>bRad_MotorStep</td>
<td>BYTE</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

4.3 typSMI_SlaveID (STRUCT)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>bAdress</td>
<td>BYTE</td>
<td>Structure with the 32 bit SlaveIDs, manufacturer ID and comment.</td>
</tr>
<tr>
<td>dwSlaveID</td>
<td>DWORD</td>
<td>32 bit SlaveID</td>
</tr>
<tr>
<td>bManuf_ID</td>
<td>BYTE</td>
<td>Manufacturer ID</td>
</tr>
<tr>
<td>sComment</td>
<td>STRING(12)</td>
<td>Comment input, max.12 characters</td>
</tr>
</tbody>
</table>

Structure with the 32-bit slave IDs, manufacturer ID and comments

4.4 typSMI_SlaveIDList (ALIAS)

List of structures which contains 32-bit slave IDs, manufacturer ID and comments
# 5.1 Status (GVL)

<table>
<thead>
<tr>
<th>Scope</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>gc_Status</td>
<td>ARRAY [0..32] OF WagoSysErrorBase.WagoTypesErrorBase.typResultItem</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eStatus.TelegramOk</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Telegramm SMI ACK’</td>
</tr>
<tr>
<td>eStatus.TelegramNack</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Telegramm SMI NACK’</td>
</tr>
<tr>
<td>eStatus.TelegramLengthError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Telegramm length’</td>
</tr>
<tr>
<td>eStatus.TelegramTimeout</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Telegramm timeout’</td>
</tr>
<tr>
<td>eStatus.TelegramEchoError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Echo telegramm’</td>
</tr>
<tr>
<td>eStatus.TelegramCrcError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘CRC failure’</td>
</tr>
<tr>
<td>eStatus.EnergySaverModeIsActive</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Energy mode is active’</td>
</tr>
<tr>
<td>eStatus.LogFailure</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Logging failure’</td>
</tr>
<tr>
<td>eStatus.LogEmpty</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Log is empty’</td>
</tr>
<tr>
<td>eStatus.SystemImageError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Failure in systemimage’</td>
</tr>
<tr>
<td>eStatus.UnkownError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Unkown error’</td>
</tr>
<tr>
<td>eStatus.NewAdressingError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Error during auto addressing’</td>
</tr>
<tr>
<td>eStatus.MasterIsReady</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Master is ready to communicate’</td>
</tr>
<tr>
<td>eStatus.MasterInvalidAdress</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Adress is not valid’</td>
</tr>
<tr>
<td>eStatus.MasterModuleAdressing</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Module addressing is active’</td>
</tr>
<tr>
<td>eStatus.MasterRegisterCommunication</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Register communication is active’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsOk</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.info</td>
<td>‘Modulesettings are ok’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Failure during reading module’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsTimeout</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Timeout while reading module’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsUnknownError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Unknown error while reading module’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsComError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Error occurred during communication’</td>
</tr>
<tr>
<td>eStatus.ModuleSettingsGeneralComError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘General communication error’</td>
</tr>
<tr>
<td>eStatus.AutomaticAdressingError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Error during automatic addressing’</td>
</tr>
<tr>
<td>eStatus.ManualSwapFailed</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Swap of address failed’</td>
</tr>
<tr>
<td>eStatus.AddressingOk</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Addressing is ok’</td>
</tr>
<tr>
<td>eStatus.MismatchInterfaceModuleNumber</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Module and interface number differ’</td>
</tr>
<tr>
<td>eStatus.ExternalAccess</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘External access to module’</td>
</tr>
<tr>
<td>eStatus.SystemExtensionError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Error occurred during system expansion’</td>
</tr>
<tr>
<td>eStatus.SystemExtensionNoConflict</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘No conflict in systemimage for system extension’</td>
</tr>
<tr>
<td>eStatus.SwapFailedAdressingError</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘Swaping error because of a conflict’</td>
</tr>
</tbody>
</table>
Table 5.1 – continued from previous page

<table>
<thead>
<tr>
<th>Value</th>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eStatus.UndefiendInterface</td>
<td>WagoSysErrorBase.WagoTypes.eSeverity.error</td>
<td>‘No interface connected’</td>
</tr>
</tbody>
</table>

5.2 eStatus (ENUM)

InOut:

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial</th>
</tr>
</thead>
<tbody>
<tr>
<td>TelegrammOk</td>
<td>0</td>
</tr>
<tr>
<td>TelegrammNack</td>
<td>1</td>
</tr>
<tr>
<td>TelegrammLengthError</td>
<td>2</td>
</tr>
<tr>
<td>TelegrammTimeout</td>
<td>3</td>
</tr>
<tr>
<td>TelegrammEchoError</td>
<td>4</td>
</tr>
<tr>
<td>TelegrammCrcError</td>
<td>5</td>
</tr>
<tr>
<td>EnergySaverModeIsActive</td>
<td>6</td>
</tr>
<tr>
<td>LogFailure</td>
<td>7</td>
</tr>
<tr>
<td>LogEmpty</td>
<td>8</td>
</tr>
<tr>
<td>SystemImageError</td>
<td>9</td>
</tr>
<tr>
<td>UnknownError</td>
<td>10</td>
</tr>
<tr>
<td>NewAdressingError</td>
<td>28</td>
</tr>
<tr>
<td>MasterIsReady</td>
<td>13</td>
</tr>
<tr>
<td>MasterInvalidAdress</td>
<td>14</td>
</tr>
<tr>
<td>MasterModuleAdressing</td>
<td>15</td>
</tr>
<tr>
<td>MasterRegisterCommunication</td>
<td>16</td>
</tr>
<tr>
<td>ExternalAccess</td>
<td>17</td>
</tr>
<tr>
<td>ModuleSettingsOk</td>
<td>18</td>
</tr>
<tr>
<td>ModuleSettingsError</td>
<td>19</td>
</tr>
<tr>
<td>ModuleSettingsTimeout</td>
<td>20</td>
</tr>
<tr>
<td>ModuleSettingsUnknownError</td>
<td>21</td>
</tr>
<tr>
<td>ModuleSettingsComError</td>
<td>22</td>
</tr>
<tr>
<td>ModuleSettingsGeneralComError</td>
<td>23</td>
</tr>
<tr>
<td>ManualSwapFailed</td>
<td>24</td>
</tr>
<tr>
<td>AutomaticAdressingError</td>
<td>25</td>
</tr>
<tr>
<td>ManualAdressingAbortion</td>
<td>26</td>
</tr>
<tr>
<td>AdressingOk</td>
<td>27</td>
</tr>
<tr>
<td>MismatchInterfaceModuleNumber</td>
<td>31</td>
</tr>
<tr>
<td>SystemExtensionError</td>
<td>29</td>
</tr>
<tr>
<td>SystemExtensionNoConflict</td>
<td>30</td>
</tr>
<tr>
<td>SwapFailedAdressingError</td>
<td>32</td>
</tr>
<tr>
<td>UndefinedInterface</td>
<td>33</td>
</tr>
</tbody>
</table>
### VersionHistory (GVL)

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Info</td>
<td>WagoSysVersion.ProjectInfo</td>
</tr>
</tbody>
</table>

WagoAppSMI.library
<table>
<thead>
<tr>
<th>date</th>
<th>version</th>
<th>author</th>
<th>change</th>
</tr>
</thead>
<tbody>
<tr>
<td>05.02.2019</td>
<td>1.0.1.3</td>
<td>u015842</td>
<td>Visu Dialogs called with Namespace</td>
</tr>
<tr>
<td>01.02.2019</td>
<td>1.0.1.2</td>
<td>u015652</td>
<td>More precise drive of the lamella position (WAT 28697)</td>
</tr>
<tr>
<td>21.01.2019</td>
<td>1.0.1.1</td>
<td>u015652</td>
<td>Position read in safety-position-mode after 4 sec.</td>
</tr>
<tr>
<td>08.01.2019</td>
<td>1.0.1.0</td>
<td>u015842</td>
<td>Properties: free placeholder added</td>
</tr>
<tr>
<td>19.11.2018</td>
<td>1.0.0.19</td>
<td>U015842</td>
<td>FsSMI_Motor new property RemainingOverrideTime</td>
</tr>
<tr>
<td>08.08.2018</td>
<td>1.0.0.18</td>
<td>WAGO / u015652</td>
<td>Fixed WAT26957</td>
</tr>
<tr>
<td>23.05.2018</td>
<td>1.0.0.17</td>
<td>WAGO / u0100408</td>
<td>Fixed WAT26414</td>
</tr>
<tr>
<td>26.04.2018</td>
<td>1.0.0.16</td>
<td>u010545</td>
<td>Update documentation</td>
</tr>
<tr>
<td>14.02.2018</td>
<td>1.0.0.15</td>
<td>WAGO / u0100408</td>
<td>Fixed WAT25623</td>
</tr>
<tr>
<td>18.08.2017</td>
<td>1.0.0.14</td>
<td>WAGO / u090230</td>
<td>changed datatype of gc_wSpsModePort_TCP from INT to WORD</td>
</tr>
<tr>
<td>17.08.2017</td>
<td>1.0.0.13</td>
<td>WAGO / u090230</td>
<td>Added warnings in the comments for the documentation</td>
</tr>
<tr>
<td>20.07.2017</td>
<td>1.0.0.12</td>
<td>WAGO / u090996</td>
<td>Fixed WAT23567</td>
</tr>
<tr>
<td>21.06.2017</td>
<td>1.0.0.11</td>
<td>WAGO / u090996</td>
<td>Edit user manual of PrgSMIC configurator</td>
</tr>
<tr>
<td>15.06.2017</td>
<td>1.0.0.10</td>
<td>WAGO / u090996</td>
<td>Edit user manual and fixed an issue appeared during development</td>
</tr>
<tr>
<td>23.05.2017</td>
<td>1.0.0.7</td>
<td>WAGO / u090996</td>
<td>Fixed issues appeared during development</td>
</tr>
<tr>
<td>23.05.2017</td>
<td>1.0.0.6</td>
<td>WAGO / u090996</td>
<td>Fixed issues appeared during development</td>
</tr>
<tr>
<td>17.05.2017</td>
<td>1.0.0.3</td>
<td>WAGO / u090996</td>
<td>Fixed WAT23419</td>
</tr>
<tr>
<td>17.05.2017</td>
<td>1.0.0.2</td>
<td>WAGO / u090996</td>
<td>Add proof of null reference of I_PORT</td>
</tr>
<tr>
<td>16.05.2017</td>
<td>1.0.0.1</td>
<td>WAGO / u090996</td>
<td>Add missing manual documentation</td>
</tr>
<tr>
<td>27.04.2017</td>
<td>1.0.0.0</td>
<td>WAGO / u090996</td>
<td>1. Release</td>
</tr>
</tbody>
</table>

**Release Notes:**

No known issues
Library Reference

This is a dictionary of all referenced libraries and their name spaces.

**Standard**

*Library Identification:*
olecules: Standard
Default Resolution: Standard, * (System)
Namespace: Standard

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**VisuDialogs**

*Library Identification:*
 molecules: VisuDialogs
Default Resolution: VisuDialogs, * (System)
Namespace: VisuDialogs

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**VisuElem3DPath**

*Library Identification:*
 molecules: System_VisuElem3DPath
Default Resolution: VisuElem3DPath, 3.5.10.0 (System)
Namespace: VisuElem3DPath

*Library Properties:*


Library Parameter:
Parameter: GC_POINTS_PER_POLYGON = 100

**VisuElemCamDisplayer**

*Library Identification:*
Placeholder: System_VisuElemCamDisplayer
Default Resolution: VisuElemCamDisplayer, 3.5.10.0 (System)
Namespace: VisuElemCamDisplayer

*Library Properties:*
- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

Library Parameter:
Parameter: GC_POINTS_PER_CAM = 100

**VisuElemMeter**

*Library Identification:*
Placeholder: System_VisuElemMeter
Default Resolution: VisuElemMeter, 3.5.10.0 (System)
Namespace: VisuElemMeter

*Library Properties:*
- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

**VisuElemTextEditor**

*Library Identification:*
Placeholder: System_VisuElemTextEditor
Default Resolution: VisuElemTextEditor, 3.5.10.0 (System)
Namespace: VisuElemTextEditor

*Library Properties:*
- LinkAllContent: False
• QualifiedOnly: False
• SystemLibrary: True
• Optional: False

VisuElemTrace

Library Identification:
Placeholder: System_VisuElemTrace
Default Resolution: VisuElemTrace, 3.5.10.0 (System)
Namespace: VisuElemTrace

Library Properties:

• LinkAllContent: False
• QualifiedOnly: False
• SystemLibrary: True
• Optional: False

VisuElemXYChart

Library Identification:
Placeholder: System_VisuElemXYChart
Default Resolution: VisuElemXYChart, 3.5.14.0 (System)
Namespace: VisuElemXYChart

Library Properties:

• LinkAllContent: False
• QualifiedOnly: False
• SystemLibrary: True
• Optional: False

VisuElems

Library Identification:
Placeholder: System_VisuElems
Default Resolution: VisuElems, 3.5.10.0 (System)
Namespace: VisuElems

Library Properties:

• LinkAllContent: False
• QualifiedOnly: False
• SystemLibrary: True
• Optional: False

VisuElemsAlarm

Library Identification:
Placeholder: System_VisuElemsAlarm
Default Resolution: VisuElemsAlarm, 3.5.10.0 (System)
Namespace: VisuElemsAlarm

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

**VisuElemsDateTime**

*Library Identification:*
- Placeholder: System_VisuElemsDateTime
- Default Resolution: VisuElemsDateTime, 3.5.10.0 (System)
- Namespace: VisuElemsDateTime

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

**VisuElemsSpecialControls**

*Library Identification:*
- Placeholder: System_VisuElemsSpecialControls
- Default Resolution: VisuElemsSpecialControls, 3.5.10.0 (System)
- Namespace: VisuElemsSpecialControls

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

**VisuElemsWinControls**

*Library Identification:*
- Placeholder: System_VisuElemsWinControls
- Default Resolution: VisuElemsWinControls, 3.5.10.0 (System)
- Namespace: VisuElemsWinControls

**Library Properties:**

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False
**VisuInputs**

*Library Identification:*

Placeholder: system_visuinputs

Default Resolution: VisuInputs, 3.5.14.0 (System)

Namespace: visuinputs

*Library Properties:*

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: True
- PublishSymbolsInContainer: True

**VisuNativeControl**

*Library Identification:*

Placeholder: System_VisuNativeControl

Default Resolution: VisuNativeControl, 3.5.10.0 (System)

Namespace: VisuNativeControl

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: True
- Optional: False

**WagoAppSocket**

*Library Identification:*

Placeholder: WagoAppSocket

Default Resolution: WagoAppSocket, * (WAGO)

Namespace: WagoAppSocket

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

.Library Parameter:

Parameter: UIMULTICONNECT_NINSTANCELISTSIZE = 20
Parameter: UIBACKLOGDEFAULT = 5
Parameter: TMULTICONNECT_TIMEOUT = TIME#0ms

**WagoAppString**

*Library Identification:*

Placeholder: WagoAppString
Library Properties:

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True

WagoSysErrorBase

Library Identification:
Placeholder: WagoSysErrorBase
Default Resolution: WagoSysErrorBase, * (WAGO)
Namespace: WagoSysErrorBase

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True
- Optional: False

Library Parameter:
Parameter: RES_LOG_MAX_FILESIZE = 2000
Parameter: RES_LOG_MAX_FILES = 1
Parameter: RES_LOG_MAX_ENTRIES = 200
Parameter: RES_LOG_NAME = 'WagoAppResultLogger'

WagoSysModuleBase

Library Identification:
Placeholder: WagoSysModuleBase
Default Resolution: WagoSysModuleBase, * (WAGO)
Namespace: WagoSysModuleBase

Library Properties:

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

Library Parameter:
Parameter: MAX_MBX_OUTPUT_SIZE = 47
Parameter: MAX_MODULE_QUANTITY = 250
Parameter: MAX_MODULE_INPUT_SIZE = 48
Parameter: MAX_RUNNABLES = MAX_MODULE_QUANTITY
Parameter: MBX_PIPE_SIZE = 1024
Parameter: MAX_MODULE_OUTPUT_SIZE = 48
Parameter: MAX_MBX_INPUT_SIZE = 47

**WagoSysPlainMem**

*Library Identification:*
Placeholder: WagoSysPlainMem
Default Resolution: WagoSysPlainMem, *(WAGO)*
Namespace: WagoSysPlainMem

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**WagoSysVersion**

*Library Identification:*
Name: WagoSysVersion
Version: 1.0.0.0
Company: WAGO
Namespace: WagoSysVersion

*Library Properties:*

- LinkAllContent: False
- QualifiedOnly: False
- SystemLibrary: False
- Optional: False

**WagoTypesCommon**

*Library Identification:*
Placeholder: WagoTypesCommon
Default Resolution: WagoTypesCommon, *(WAGO)*
Namespace: WagoTypes

*Library Properties:*

- LinkAllContent: False
- Optional: False
- QualifiedOnly: False
- SystemLibrary: False
- PublishSymbolsInContainer: True
WagoTypesModuleBase

Library Identification:
Placeholder: WagoTypesModuleBase
Default Resolution: WagoTypesModuleBase, * (WAGO)
Namespace: WagoTypesModuleBase

Library Properties:

• LinkAllContent: False
• QualifiedOnly: True
• SystemLibrary: False
• PublishSymbolsInContainer: True
• Optional: False

Library Parameter:
Parameter: MAX_MODULE_OUTPUT_SIZE = 48
Parameter: MAX_MODULE_QUANTITY = 250
Parameter: MAX_MODULE_INPUT_SIZE = 48
Parameter: MAX_RUNNABLES = MAX_MODULE_QUANTITY
Parameter: MBX_PIPE_SIZE = 1024
Parameter: MAX_MBX_SIZE = 18
Parameter: MAX_MBX1_SIZE = 18
Parameter: MAX_MBX_OUTPUT_SIZE = 47
Parameter: MAX_MBX_INPUT_SIZE = 47

WagoTypesModule_753_163x

Library Identification:
Placeholder: WagoTypesModule_753_163x
Default Resolution: WagoTypesModule_753_163x, * (WAGO)
Namespace: WagoTypesModule_753_163x

Library Properties:

• LinkAllContent: False
• QualifiedOnly: False
• SystemLibrary: False
• Optional: False

WagoVisuIcons

Library Identification:
Placeholder: WagoVisuIcons
Default Resolution: WagoVisuIcons, * (WAGO)
Namespace: WagoVisuIcons

Library Properties:

• LinkAllContent: False