SIEMENS

SIMOTION

SIMOTION SCOUT
Standard function for SINAMICS S120 line modules

Function Manual

02/2012

Legal information

Warning notice system

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

⚠DANGER

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

⚠WARNING

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

⚠CAUTION

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

CAUTION

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

NOTICE

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

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

Qualified Personnel

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

Proper use of Siemens products

Note the following:

⚠WARNING

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

Trademarks

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

Disclaimer of Liability

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

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
GERMANY

Foreword

Contents of the function manual

This document is part of the SIMOTION programming references documentation package.

Function block

The _LineModule_control function block is part of the command library of the "SIMOTION SCOUT" engineering system.

SIMOTION Documentation

An overview of the SIMOTION documentation can be found in a separate list of references.

This documentation is included as electronic documentation in the scope of delivery of SIMOTION SCOUT. It comprises 10 documentation packages.

The following documentation packages are available for SIMOTION V4.3:

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

Hotline and Internet addresses

Additional information

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

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

http://www.siemens.com/motioncontrol/docu

Please send any questions about the technical documentation (e.g. suggestions for improvement, corrections) to the following e-mail address:
docu.motioncontrol@siemens.com

My Documentation Manager

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

http://www.siemens.com/mdm

Training

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

http://www.siemens.com/sitrain

FAQs

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

http://support.automation.siemens.com

Technical support

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

http://www.siemens.com/automation/service&support

Table of contents

Description 1

1.1 General

The **_LineModule_control** function block can only be used in the SIMOTION system

- For the SIMOTION D4x5/D4x5-2 hardware platform or
- For the SIMOTION C, P, and D hardware platforms with a subordinate SINAMICS S120 CU320/CU320-2

in conjunction with a SINAMICS S120 infeed (Line Module) with a DRIVE-CLiQ connection.

From SIMOTION V4.2 onward, the symbolic assignment is activated as standard for newly created projects. For this purpose, no message frames must be configured for the infeed. If the symbolic assignment is deactivated, SIEMENS message frame 370 must be configured.

The following software versions are required for the standard function described in this documentation:

- SIMOTION SCOUT V4.2 or higher
- SIMOTION Kernel V4.2 or higher

Note

For versions earlier than V4.2, a function block can be found in "SIMOTION Utilities & Applications" for controlling SINAMICS S120 Line Modules (under **Applications > Cross-Sector Applications > Function Block for Controlling Line Modules**).

"SIMOTION Utilities & Applications" is provided free of charge and as part of the SIMOTION SCOUT scope of delivery.

1.2 Product description

You can use the _LineModule_control function block to switch on and off all SINAMICS S120 Line Modules with DRIVE-CLiQ connections from SIMOTION, as well as to perform straightforward diagnostics tasks.

The following infeeds are supported by the _LineModule_control function block:

- Active Line Modules (ALM)
- Basic Line Modules (BLM)
- Smart Line Modules (SLM)

The _LineModule_control function block is provided for communication between the higher-level SIMOTION control and SINAMICS S120 Line Modules using the command library of the SIMOTION SCOUT engineering system.

The function block is described in this manual.

Note

Line Modules (Active Line Modules, Basic Line Modules, Smart Line Modules) of different types must not be operated simultaneously on the same DC link.

Note

For explanations relating to the status and control word of SINAMICS S120 Line Modules, refer to the

- *SINAMICS S120 Drive Functions* Function Manual
- *SIMOTION D4x5* Commissioning and Hardware Installation Manual

These documents are provided as part of the SIMOTION SCOUT scope of delivery in electronic format.

Parameter assignment / addressing 2

2.1 Overview

Communication between the SIMOTION device and the Line Module takes place via the internal PROFIBUS DP (for SIMOTION D4x5/D4x5-2) or the external PROFIBUS DP or PROFINET (for SIMOTION C, P, D with subordinate SINAMICS S120 CU320/CU320-2).

From SIMOTION V4.2 onward, the symbolic assignment is activated as standard for newly created projects. For this purpose, no message frames must be configured for the infeed.

If the symbolic assignment is deactivated, SIEMENS message frame 370 must be configured for use of the **_LineModule_control** function block function. Addresses from 256 are recommended for configuring the Line Modules, where I/O variables are used for the write and read I/O access procedures.

2.2 Addressing the Line Module for SINAMICS S120

Creating I/O variables

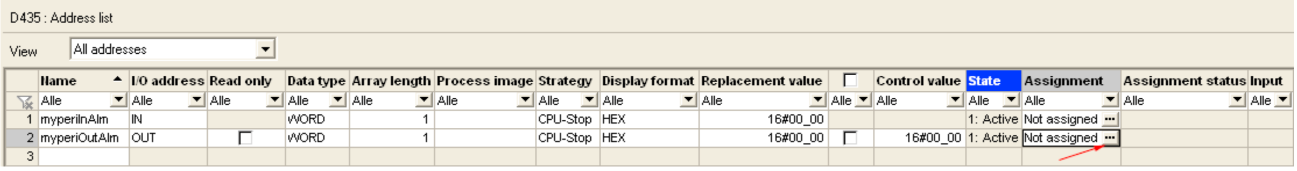
In the SIMOTION user project, two I/O variables with the configured Line Module I/O addresses must be created for the purpose of reading the status word (ZSW) and writing the control word (STW) of the configured Line Module.

The I/O variables must be created in the symbol browser in SIMOTION SCOUT. You are free to choose any names for them. The following shows an example of an Active Line Module created as I/O variable **myperiInAlm** and **myperiOutAlm** . These are also used in the application example: refer to the chapter titled Example of an application (Page 27).

Activated symbolic assignment

In the case of **activated symbolic assignment,** create the I/O variables in the address list as follows:

1. Create two I/O variables in the address list (e. g. **myperiInAlm** and **myperiOutAlm**), see Figure "Assigning an I/O variable".
2. The status word must be assigned to the I/O variable **myperiInALM**. For this purpose, select the entry "IN" as the I/O address and under data type select the entry "empty" or WORD. In the "Assignment" column, click on the "..." button.



D435 : Address list

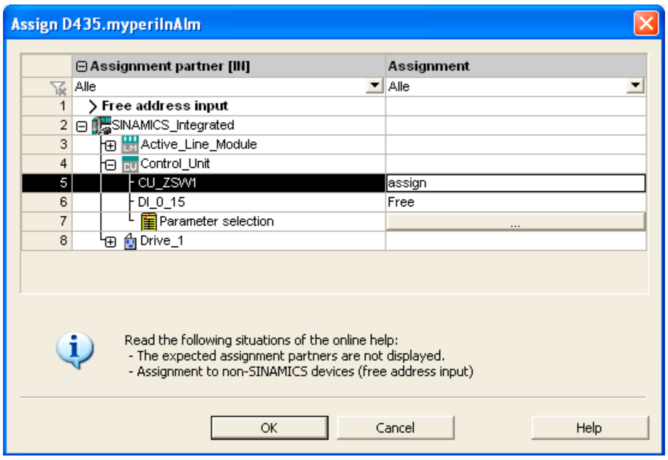
View: All addresses

	Name	I/O address	Read only	Data type	Array length	Process image	Strategy	Display format	Replacement value	□	Control value	State	Assignment	Assignment status	Input
	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle
1	myperiInAlm	IN		WORD	1		CPU-Stop	HEX	16#00_00			1: Active	Not assigned		
2	myperiOutAlm	OUT	□	WORD	1		CPU-Stop	HEX	16#00_00	□	16#00_00	1: Active	Not assigned		
3															

Figure 2-1 Assigning an I/O variable

The "Assigning myperiInALM" window opens.

3. Highlight the status word under infeed and confirm by pressing "OK".



Assign D435.myperiInAlm

	Assignment partner [III]	Assignment
	Alle	Alle
1	Free address input	
2	SINAMICS_Integrated	
3	Active_Line_Module	
4	Control_Unit	
5	CU_ZSW1	assign
6	DI_0_15	Free
7	Parameter selection	...
8	Drive_1	

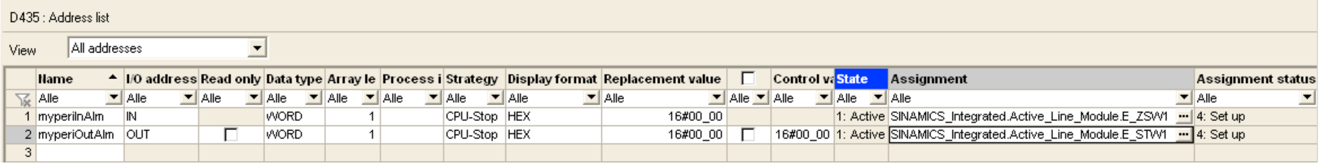
Read the following situations of the online help:
- The expected assignment partners are not displayed.
- Assignment to non-SINAMICS devices (free address input)

Figure 2-2 Highlighting the status word

The I/O variable is assigned to the status word.

4. Repeat steps 2 to 3 for assigning the control word with the I/O variable **myperiOutAlm**.

   The I/O variables are now assigned to the status and control word.



D435 : Address list

View: All addresses

	Name	I/O address	Read only	Data type	Array le	Process i	Strategy	Display format	Replacement value	□	Control va	State	Assignment	Assignment status
	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle
1	myperiInAlm	IN		WORD	1		CPU-Stop	HEX	16#00_00			1: Active	SINAMICS_Integrated.Active_Line_Module.E_ZSW1	4: Set up
2	myperiOutAlm	OUT	□	WORD	1		CPU-Stop	HEX	16#00_00	□	16#00_00	1: Active	SINAMICS_Integrated.Active_Line_Module.E_STW1	4: Set up
3														

Figure 2-3 Assigned I/O variables in the address list

Continue with the chapter entitled Parameter transfer at FB _LineModule_control (Page 12).

Deactivated symbolic assignment

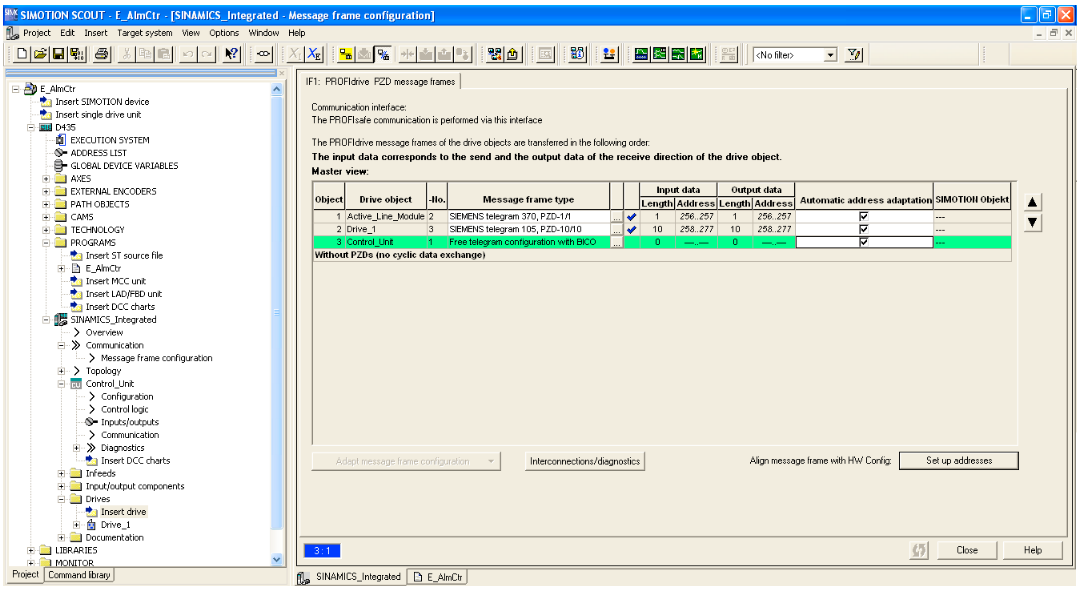
In the case of **deactivated symbolic assignment,** proceed as follows to determine the addresses of the Line Module:

1. In the project navigator, double-click "Communication" followed by "Message frame configuration" underneath the SINAMICS drive unit.

2. The "PZD message frames" window opens. Check whether message frame 370 is selected for the infeed. Click on "Set up address". The assigned addresses of the message frames are then displayed. Select the start address for the input data (see figure below).

Example

Determining the addresses from the message frame configuration of the SINAMICS drive unit (using an ALM as an example):



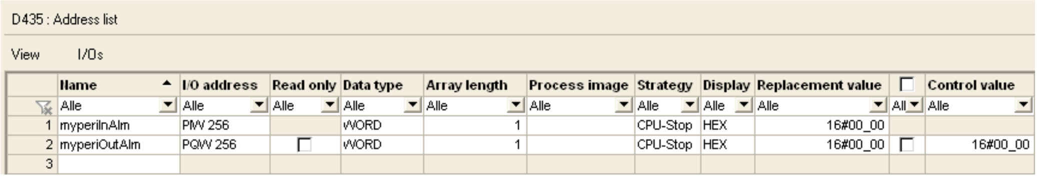
Communication interface:
The PROFIsafe communication is performed via this interface

The PROFIdrive message frames of the drive objects are transferred in the following order:
The input data corresponds to the send and the output data of the receive direction of the drive object.
Master view:

Object	Drive object	-No.	Message frame type			Input data		Output data		Automatic address adaptation	SIMOTION Objekt
						Length	Address	Length	Address		
1	Active_Line_Module	2	SIEMENS telegram 370, PZD-1/1	...	✓	1	256..257	1	256..257	☑	---
2	Drive_1	3	SIEMENS telegram 105, PZD-10/10	...	✓	10	258..277	10	258..277	☑	---
3	Control_Unit	1	Free telegram configuration with BICO	...		0	—.—	0	—.—	☑	---
Without PZDs (no cyclic data exchange)											

Figure 2-4 Message frame configuration addresses

Creating the I/O variables in the symbol browser:


D435 : Address list

View I/Os

	Name	I/O address	Read only	Data type	Array length	Process image	Strategy	Display	Replacement value	□	Control value
	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	Alle	All	Alle
1	myperiInAlm	PIW 256		WORD	1		CPU-Stop	HEX	16#00_00		
2	myperiOutAlm	PQW 256	□	WORD	1		CPU-Stop	HEX	16#00_00	□	16#00_00
3											

Figure 2-5 Addressing with I/O variables (example)

Note

Make sure that the addresses are accepted correctly from the message frame configuration. In your user project, when the **_LineModule_control** FB is called you must assign these two I/O variables to the **periIn** and **periOut** input/output parameters. Only once you have done this will the Line Module status word that has been read be transferred to the FB, and the output data prepared by the FB for the control word be transferred to the Line Module.

2.3 Parameter transfer at FB _LineModule_control

The I/O variable for the I/O inputs (status word) must be transferred to the **periIn** input parameter.

The prepared data for the I/O outputs (control word) are supplied by the **_LineModule_control** FB at the **periOut** output parameter. The **periOut** output parameter must be assigned to the I/O variable for the I/O outputs.

At the **typeLM** input parameter, specify the type of Line Module to be controlled.

Programming 3

3.1 _LineModule_control function block

3.1.1 Overview of _LineModule_control FB

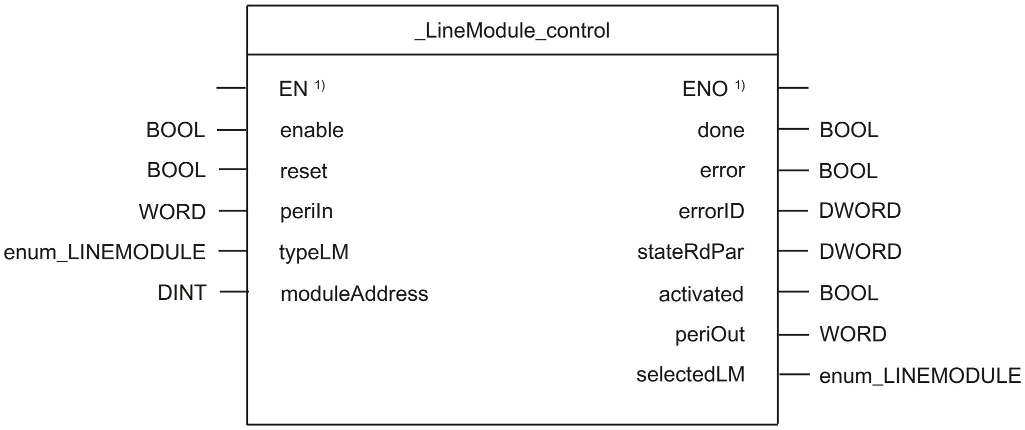
Task

You can use the **_LineModule_control** FB to switch on and off infeeds (Line Modules) for SINAMICS S120 with a DRIVE-CLiQ connection via your user program. The **_LineModule_control** FB transfers the commands to the selected Line Module, reads the response data provided, and monitors the status signals from the Line Module.

Note

From SIMOTION V4.2 onward, the symbolic assignment is activated as standard for newly created projects. For this purpose, no message frames must be configured for the infeed. If the symbolic assignment is deactivated, SIEMENS message frame 370 must be configured.

Call (LAD representation)



[1] LAD-specific parameters

3.1.2 Parameter description

The table below contains all the parameters of the **_LineModule_control** FB.

Table 3- 1 Parameters of the _LineModule_control FB

Name	P type [1]	Data type	Default	Meaning
enable	IN	BOOL	FALSE	Switch line module on/off **TRUE** = Depending on the **typeLM** input parameter, the line module type is determined and the line module is then switched on (FALSE → TRUE status change) **FALSE** = Switch off
reset	IN	BOOL	FALSE	Acknowledge pending errors with rising edge
periIn	IN	WORD	16#0000	The I/O variable for the status word read from the line module is assigned to this input parameter, see Section Parameter assignment / addressing (Page 9)
typeLM	IN	enum_ LINEMODULE	NOT_DEFINED	• Selection of line module to be controlled: ACTIVE_LINE_MODULE = 1 SMART_LINE_MODULE = 2 BASIC_LINE_MODULE = 3 • Automatic detection of line module type: AUTO_DETECT = 4 (only recommended in exceptional cases) The typeLM input parameter can only be changed when the line module is switched off. If the **typeLM** input parameter is changed in the activated state, there is no reaction. Only after deactivation (enable=FALSE) and subsequent activation (enable=TRUE) is the value in the **typeLM** input parameter taken over.
moduleAddress	IN	DINT	0	Logical address of the line module Only required if **typeLM** = AUTO_DETECT.
done	OUT	BOOL	FALSE	Job completed without errors. Signal present for just one cycle.
error	OUT	BOOL	FALSE	Line module error status **TRUE** = error during the request processing
errorID	OUT	DWORD	16#00000000	Specification of the error For **error** = **TRUE**, the **errorID** parameter contains the error information (refer to the "Error messages" table)
stateRdPar	OUT	DWORD	16#00000000	Error messages or status messages of system function _readDriveParameter() Only relevant if **typeLM** = AUTO_DETECT.
activated	OUT	BOOL	FALSE	Line module operating state: **TRUE** = line module switched on **FALSE** = line module switched off

Name	P type [1]	Data type	Default	Meaning
periOut	OUT	WORD	16#0000	This output parameter must be linked with the I/O variable for the line module control word; refer to Section Parameter assignment / addressing (Page 9).
selectedLM	OUT	enum_ LINEMODULE	NOT_DEFINED	Selected or automatically detected line module: • NOT_DEFINED = 0: – If the readDriveParameter() system function was not completed without errors. – If no line module type was selected at the **typeLM** input parameter on a rising edge at the **enable** input parameter. The **activated** output parameter is also set to FALSE in this status. • ACTIVE_LINE_MODULE = 1 • SMART_LINE_MODULE = 2 • BASIC_LINE_MODULE = 3

[1] Parameter types: IN = input parameter, OUT = output parameter

3.1.3 Function description

At the **typeLM** input parameter, specify the type of Line Module to be controlled. In exceptional cases, e.g. generic blocks for modular machines, automatic detection of the Line Module can also be configured.

At the **selectedLM** output parameter, the selected Line Module or the Line Module determined in the **_LineModule_control** FB is displayed.

A rising edge at the **enable** input parameter switches on the Line Module.

The switch from TRUE to FALSE at the **enable** input parameter switches off the Line Module. Errors present at the **error** output parameter must be acknowledged with a rising edge at the **reset** input parameter. Only the error is acknowledged. Once the error has been successfully acknowledged, the Line Module must be switched on again using a rising edge on the **enable** input parameter. The Line Module will not be switched on if errors are present (**error** = TRUE) while there is a rising edge at the **enable** input parameter. The error must be acknowledged first.

The error reset and the enable can also be executed together. Setting **enable** and **reset** from FALSE to TRUE simultaneously first acknowledges a pending error and then switches on the Line Module.

- If an error has been acknowledged with **reset** = TRUE and no others errors are pending, **error** = FALSE and **errorID** = 0.
- If an error is still present, **error** = TRUE remains and the **errorID** will be updated.

3.1.4 Manual selection of the Line Module

Select the Line Module to be controlled via the parameter **typeLM** :

- **typeLM** = ACTIVE_LINE_MODULE
- **typeLM** = SMART_LINE_MODULE
- **typeLM** = BASIC_LINE_MODULE

If no errors are present, the Line Module selected at the **typeLM** input parameter will be switched on when there is a rising edge at the **enable** input parameter.
The **_LineModule_control** FB does not check the selected Line Module. The _readDriveParameter() system function is not called in the **_LineModule_control** FB. The FB does not evaluate the **moduleAddress** input parameter in this case.

3.1.5 Automatic detection of the Line Module (only in exceptional cases)

General information

In exceptional cases, e.g. generic blocks for modular machines, automatic detection of the Line Module type can be configured:

typeLM = AUTO_DETECT

Note

When using this function, you must take potential programming conflicts with other parallel DPV1 jobs into account, e.g. from the library functions used. Manual selection is recommended for familiar Line Module types.

If the user sets the **typeLM** input parameter to AUTO_DETECT, the **_LineModule_control** FB automatically detects the type of Line Module once when there is a rising edge at the **enable** input parameter. In this mode, the _readDriveParameter() system function is called internally in the FB and the type of Line Module is determined by means of a parameter request. For this function, the configured Line Module address must also be specified at the **moduleAddress** input parameter (see Section "Determining the Logical Address of the Line Module"). Otherwise, the _readDriveParameter() system function called by the **_LineModule_control** FB will signal an error. In the case of error "50001" at the **errorID** output parameter, the **stateRdPar** output parameter will contain specific information on the error from the _readDriveParameter() system function.

Note

You must ensure that only one parameter request is ever active for each drive unit (e.g. SINAMICS Integrated, CU320, CX32). Otherwise, conflicts with other DPV1 jobs may arise, e.g. from the library functions used. Any additional requests sent to the same drive unit will be rejected with the error 16#FFFF81C7 (on output parameter functionResult of the parameter job or on output parameter **stateRdPar** of the **_LineModule_control** FB).

In the case of automatic Line Module detection, no parameter requests from the user program may be active at the drive unit of the Line Module when there is a rising edge at the **enable** input parameter. If the Line Module type cannot be determined automatically, an error is generated and the Line Module is not switched on.

You can recognize an active parameter request in the **_LineModule_control** FB from the values 0x00007001 or 0x00007002 in the **stateRdPar** output parameter.

The values 0x0000000 or FFFF8xxx (parameter request aborted with an error) in the **stateRdPar** output parameter indicate that the **_LineModule_control** FB is not processing any parameter requests internally.

For additional information on acyclic reading and writing of parameters (using DP-V1 services) with the _readDriveParameter() system function, please refer to the *SIMOTION Communication* System Manual, as well as the *SIMOTION D4x5* Commissioning and Hardware Installation Manual, Section "Acyclic communication with the drive".

Effect of the _readDriveParameter() system function in the _LineModule_control FB

The _readDriveParameter() system function enables the **_LineModule_control** FB to determine the type of Line Module when **typeLM** is set to AUTO_DETECT. For this purpose, the system function reads out parameter r107 of the Line Module. Parameter r107 indicates the Line Module type.

The parameter is read out once before the start of the Line Module switch-on sequence.

Once the Line Module type has been successfully read out, the switch-on sequence is executed.

If the Line Module type is read out incorrectly, the **error** output parameter is set to TRUE, the **selectedLM** output parameter is set to NOT_DEFINED, and the switch-on sequence is not started. Additionally, more detailed information on the error is specified in the **errorID** and **stateRdPar** output parameters.

Requirement for reading out the Line Module type/call:

The _readDriveParameter() system function call starts when there is a rising edge at the **enable** input parameter and **typeLM** is set to AUTO_DETECT for the **_LineModule_control** FB. The system function is called repeatedly until either 0x00000000 (OK) or 0xFFFF8xxx (parameter request aborted with error) is signaled at its output parameter, functionResult.

During the read process, the values 0x00007001 (first call) or 0x00007002 (intermediate call) are output (parameter request active).

All values of the functionResult output parameter of system function _readDriveParameter() are output in exactly the same format at the **stateRdPar** output parameter of the **_LineModule_control** FB. The error functionResult = 0xFFFF81C7 (parameter request active) is an exception to this rule. This error is only output once a timeout has expired at the output parameters of the **_LineModule_control FB** (see the "Timeout behavior" description below).

You can use the **stateRdPar** output parameter to execute (start/end) reading of the internal FB parameters.

Timeout behavior in relation to the _readDriveParameter() system function:

If a parameter request from a user or from the SIMOTION device is already active at the Line Module (functionResult = 0xFFFF81C7 error message), the parameter is read out repeatedly within a timeout period of 1 s.

The **stateRdPar** output parameter of the **_LineModule_control** FB remains at value 0x00007002 during the repeated readout process taking place within the timeout.

The parameter is only read again if there is a functionResult = 0xFFFF81C7 error. If the system function signals any other errors, an error is output immediately at the **_LineModule_control** FB and the parameter reading process is aborted with the **_readDriveParameter** system function.

If the functionResult = 0xFFFF81C7 error is still present once the timeout period has elapsed, the following error is output at the output parameters of the **_LineModule_control** FB:

error	= TRUE
errorID	= 50001
stateRdPar	= 0xFFFF81C7
selectedLM	= NOT_DEFINED

Output parameters in relation to the _readDriveParameter() system function:

Table 3- 2 Output parameters in relation to _readDriveParameter():

Parameters of the _LineModule_control FB	Task
stateRdPar	Output the functionResult output parameter of the _readDriveParameter() system function
selectedLM	Output the Line Module type determined If a parameter has been assigned incorrectly (e.g. the wrong module address) or an error occurs during parameter reading, this output parameter will be set to NOT_DEFINED.

Determining the logical address of the Line Module

If you are using the automatic detection method, you need to assign a value for the module address (logical address) of the Line Module at the **moduleAddress** input parameter of the **_LineModule_control** FB (see Chapter Determining the module address (Page 19)).

3.1.6 Determining the module address

To select a diagnostic interrupt in the PeripheralFaultTask, the module address (logical address) of the Line Module must be determined and made readily available in a variable.

In the case of activated symbolic assignment, determine the module address using the system function **_getLogicalAddressOfIoVariable** from the name of the I/O variable for the status or control word of the Line Module.

In the case of deactivated symbolic assignment, proceed in the same way as for determining the addresses of I/O variables, but this time determine the module address for the status and control word of the Line Module. Use the same address value (start address of input data) in the same way as for I/O variables for the status and control word of the Line Module, see Addressing the Line Module for SINAMICS S120 (Page 9).

3.1.7 Graphical overview of the functionality

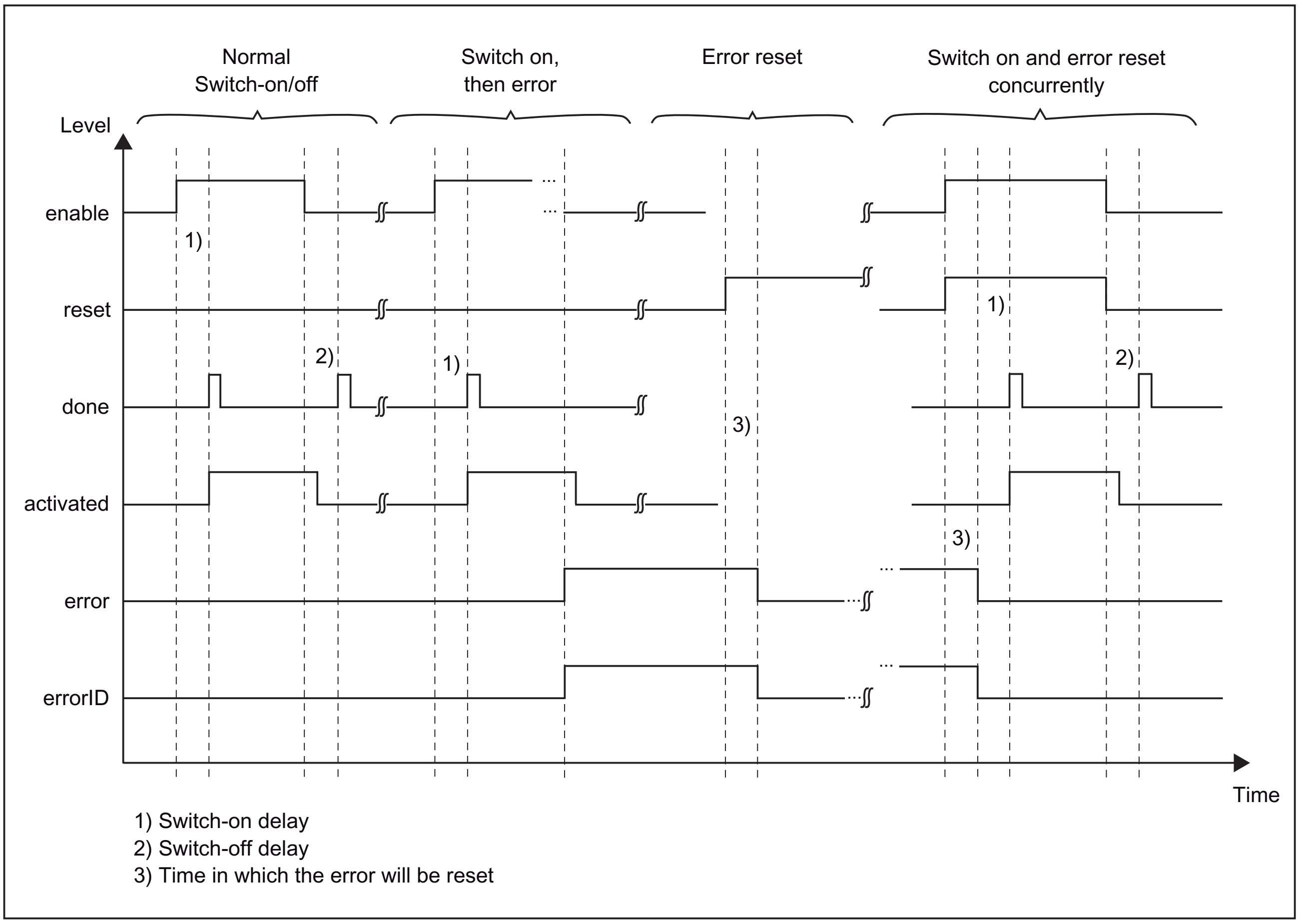


Figure 3-1 Signal propagation diagram

3.1.8 Task integration (call)

The **_LineModule_control** FB is designed to be called in a cyclic task and must be called in this task during each task pass. Processing a job can take several cycles. The user decides which cyclic task of the **_LineModule_control** FB the call is made in. No restrictions are applied with respect to the functionality of the **_LineModule_control** FB.

3.2 Calling the function block

Procedure

Proceed as follows to work with the **_LineModule_control** function block in your user project (the numbers shown in the program segment below correspond to the steps listed):

1. Create an instance of the **_LineModule_control** function block.
2. Call the function block instance and transfer input parameters.
3. The output parameters of the function block are accessed with <instance name of FB>.<name of output parameter>.
4. The data for the I/O outputs (control word of the Active Line Module) prepared by the FB must be assigned by the user program to the I/O variables for the purpose of writing the control word. It is recommended that you use the _setSafeValue system function for this assignment task.

Note

The program segment is an extract from the application example supplied. The application example is included on the "SIMOTION Utilities & Applications" DVD and is available for various SIMOTION hardware platforms.

"SIMOTION Utilities & Applications" is provided free of charge and as part of the SIMOTION SCOUT scope of delivery.

Note

For additional information, see the following sources:

- *SIMOTION SCOUT* online help
- Programming manual of the corresponding programming language, e.g.:
  - *SIMOTION ST*, *Structured Text* programming manual
  - *SIMOTION MCC*, *Motion Control Chart* programming manual
  - *SIMOTION LAD/FBD, Ladder Diagram and Function Block Diagram* Programming Manual

These documents are provided as part of the SIMOTION SCOUT scope of delivery in electronic format.

Program segment from the sample program:

```
UNIT E_AlmCtr;

INTERFACE

VAR_GLOBAL
  myFbAlmCtrl : _LineModule_control;    // Instance of FB _LineModule_control                    (1)
END_VAR
```

```
PROGRAM StartUpAlm;
PROGRAM BackGrndAlm;
PROGRAM PeripheralFaultAlm;

END_INTERFACE

IMPLEMENTATION

PROGRAM BackGrndAlm
// *********************************************************************
// call instance of FB _LineModule_control
// *********************************************************************

  myFbAlmCtrl ( enable    := myAlmCtrl_In.enable,                                                 (2)
                reset     := myAlmCtrl_In.reset,
                typeLM    := ACTIVE_LINE_MODULE,
                periIn    := myAlmCtrl_In.periIn
              );
  myAlmCtrl_Out.error     := myFbAlmCtrl.error;                                                   (3)
  myAlmCtrl_Out.errorID   := myFbAlmCtrl.errorID;
  myAlmCtrl_Out.done      := myFbAlmCtrl.done;
  myAlmCtrl_Out.activated := myFbAlmCtrl.activated;
  myAlmCtrl_Out.periOut   := myFbAlmCtrl.periOut;

//****************************************************************************
// write the output parameters of FB _LineModule_control - control WORD ALM -
// to i/o-variable myperiOutAlm
//****************************************************************************
  s_eRetVal := _setSafeValue (                                                                    (4)
               variable   := myperiOutAlm,
               value      := myAlmCtrl_Out.periOut,
               accessmode := default_value,
               setvalue   := s_setValue
              );

END_PROGRAM

END_IMPLEMENTATION
```

3.3 Error messages

The value TRUE at the error output parameter indicates an error status in the Line Module. More details of the error are provided at the errorID output parameter. The assignment of the error cause is made in this output parameter.

If an internal Line Module (200xx group) fault occurs, the user must read out the exact cause using the appropriate SIMOTION system functions, e.g. _readDriveFaults.

For a detailed description of the system functions, refer to the *SIMOTION System Functions/Variables Device* Parameter Manual. This document is provided as part of the SIMOTION SCOUT scope of delivery in electronic format.

Error groups

The errors signaled in the errorID output parameter may be allocated to the following error groups:

Group 100xx: A time-out is present.

The Line Module does not respond to the _LineModule_control FB commands, or responds too late.

Group 200xx: An internal Line Module fault is present.

"Fault active" bit of the Line Module status word = TRUE

The exact cause of the error must be read out using the appropriate SIMOTION system functions (e.g. _readDriveFaults).

Group 300xx: The Line Module has withdrawn the "Controlled by PLC" bit in the status word during operation.

Group 400xx: An error occurred during a reset.

Group 500xx: An error occurred during the execution of the _readDriveParameter() system function used internally by the _LineModule_control FB. More detailed error information can be obtained from the statusRdPar output parameter.

Error messages

Note

Statuses S1 to S4 are contained in the errorID output parameter while the Line Module is in the process of being switched on or off.

You can find the meanings of the statuses (S1 to S4) described in the table below in the Appendix Flow diagrams for switching the Line Modules on and off (Page 31).

For explanations relating to the status and control word of SINAMICS S120 Line Modules, refer to the

- *SINAMICS S120 Drive Functions* Function Manual
- *SIMOTION D4x5* Commissioning and Hardware Installation Manual

These documents are provided as part of the SIMOTION SCOUT scope of delivery in electronic format.

The error groups and error messages listed below are to be viewed as being in decimal format.

Table 3- 3 Error messages

Error no. (errorID), decimal	Meaning
100xx	Timeout during the status transition of the Line Module. • 1000x: Time-out when switching on: Status x (S1…S4) was not attained. • 100x0: Time-out when switching off: Status x (S1…S4) was not attained. **Example:** 10030: Time-out error when switching off. Status S3 was either not attained at all or attained too late.
200xx	The Line Module reports an internal fault using the "Fault active" bit of the status word. • 2000x: Internal fault when switching on: Status x (S1…S4) was not attained. • 200x0: Internal fault when switching off: Status x (S1…S4) was not attained. **Example:** 20002: Internal fault when switching on: In status S2. The exact cause of the error must be read out using the appropriate SIMOTION system functions (e.g. _readDriveFaults).
30000	While there was a rising edge on the **enable** input parameter, the "Controlled by PLC" bit in the status word of the Line Module was set to **FALSE**. This means that the command could not be performed.
300xx	The Line Module has reset the "Controlled by PLC" bit in the status word. The FB resets all bits in the Line Module control word and waits for new commands. • 3000x: The "Controlled by PLC" bit in the status word was reset to Status x (S1...S4) when switching on. • 300x0: The "Controlled by PLC" bit in the status word was reset during operation or when switching off to Status x (S1…S4). **Example:** 30002: The "Controlled by PLC" bit in the status word was reset during switching on to Status S2.
40001	While there was a rising edge on the **reset**input parameter, the "Controlled by PLC" bit in the status word of the Line Module was set to **FALSE**. This means that the command could not be performed.
40003	While there was a rising edge on the **reset** input parameter, the Line Module did not reset the "Fault active" bit in the status word, even though the "Acknowledge error" bit was set in the control word.
50001	Only relevant if input parameter **typeLM** = AUTO_DETECT An error occurred during the execution of the _readDriveParameter() system function. Evaluate the **statusRdPar** output parameter. Information on the errors can be found in the *SIMOTION System Functions/Variables Device* List Manual.

Error correction

Use the methods outlined below to correct pending errors.

Table 3- 4 Information on correcting pending errors

Check the diagnostics LEDs on the Line Module and all SINAMICS components. • Check the RDY LED. • Check the DC-LINK LED. For descriptions of all the LEDs, please refer to the *SINAMICS S120 Booksize Power Units* Manual.
Check the communication between SIMOTION and the connected Line Module. Has SIEMENS message frame 370 been configured for the connected Line Module? See Section Parameter assignment / addressing (Page 9)
Check the programming for the _LineModule_control FB. • Does the symbol browser in your project contain one I/O variable for reading the status word and one for writing the control word? • Is the I/O variable for reading the status word assigned to input parameter **periIn**of the **_LineModule_control** FB? • Is output parameter **periOut** assigned to the I/O variable for writing the control word after the **_LineModule_control** FB is called? • Is the **_LineModule_control** FB called in a cyclic task and run during each cycle? See Section Parameter assignment / addressing (Page 9)
Check whether the connected Line Module has signaled any errors. • Check the error messages in the "Alarms" window of SIMOTION SCOUT. • In the Line Module expert list, check the error messages in the following parameters: p945 (fault code) p947 (fault number) p2131 (current fault code) • Check the error messages via the user program, using the _readDriveFaults() system function. For information on the causes of the errors read and how to remedy them, please refer to the *SINAMICS S* List Manual and the *SIMOTION SCOUT* online help. On SIMOTION Utilities & Applications, you will also find a DPV1 library containing a function block for reading drive fault and warning messages. SIMOTION Utilities & Applications is provided as part of the SIMOTION SCOUT scope of delivery.
Check the entries in the SINAMICS diagnostics buffer (D4x5 SINAMICS Integrated/CX32: SIMOTION V4.1 SP2 and higher; CU320: SINAMICS Firmware V2.6 and higher) 1. Right-click the SINAMICS device in the SIMOTION SCOUT project navigator. 2. Select "Target device" > "Device Diagnostics" from the context menu and open the "Diagnostics buffer" tab.

Check the status of the SINAMICS drive. **Line Module:** 1. Open the Line Module expert list in SIMOTION SCOUT. 2. Check the following parameters: r2 (status display) p10 = 0 (commissioning parameter filter) r46 (missing enables) **Control Unit:** 1. Open the Control Unit expert list in SIMOTION SCOUT. 2. Check the following parameters: r2 (status display) p9 = 0 (device commissioning parameter filter)
Check the Line Module wiring. • Check the supply voltage for the connected Line Module. • Check the supply voltage at the EP terminals. • Check the DRIVE-CLiQ topology. • Check the wiring of the Active Interface Module (if present).

See also

_LineModule_control function block (Page 13)

Example of an application 4

4.1 General

Task

The application example shows:

- How you can use the **_LineModule_control** FB to switch a Line Module on and off. This is shown using the example of an ALM.
- How error statuses are signaled by the **_LineModule_control** FB, and how you can reset the errors.
- Additional ALM diagnostic information.

The application example contains the following programs:

Table 4- 1 Application example programs

Program	Task	Meaning
StartUpALM	StartupTask	Program for start-up
PeripheralFaultALM	PeripheralFaultTask	Program for handling diagnostic alarms
BackGrndALM	BackgroundTask	Program for switching the ALM on and off

Structure of the example

In the application example, an Active Line Module (ALM) is used as the infeed. The application example is structured as follows:

- Type declarations
- Variable definitions
- Variable initialization
- Switch-on sequence
- Switch-off sequence
- Evaluation of diagnostic alarms in the **PeripheralFaultTask**

Hardware platform

The application example is available for various SIMOTION hardware platforms.

Note

If the application example is not available for your hardware platform, you must adapt the hardware configuration.

Calling the application example

The application example can be found on the "SIMOTION Utilities & Applications" DVD. "SIMOTION Utilities & Applications" is provided free of charge and as part of the SIMOTION SCOUT scope of delivery.

1. Dearchive and open the project containing the application example.
2. Check the axis configuration:
3. If necessary, modify the example project.
4. Save and compile the example project. You can then download the example to the SIMOTION device and switch to RUN mode.

Adapting the application example

The address of the Line Module in the frame configuration must match the addresses of the I/O variables in the application example. To enable selection of a diagnostic interrupt in the PeripheralFaultTask, the address of the Line Module must be adapted in a subsequent variable.

You must adapt the following settings where necessary:

- Address of I/O variables myperiInAlm and myperiOutAlm (default: 256), see Chapter Addressing the Line Module for SINAMICS S120 (Page 9)
- myAlmModuleAddress(default: 256)

  In the case of activated symbolic assignment, determine the module address via the system function _getLogicalAddressOfIoVariable, see Chapter Determining the module address (Page 19).

4.2 Sequence of the application example

Relevant variables in the application example

Table 4- 2 Overview of the relevant variables

Variable	Data type	Initial value	Meaning
mySwitchOn	BOOL	FALSE	**TRUE** = Start the switch-on sequence
mySwitchOff	BOOL	FALSE	**TRUE** = Start the switch-off sequence
myFirstRun	BOOL	TRUE	**TRUE** = Initialize the variables
myError	BOOL	FALSE	**TRUE** = Error on ALM or error during switching on/off
myErrorId	DWORD	16#00000000	Error specification
myDiagnosticAlarm	BOOL	FALSE	**TRUE** = Diagnostic alarm present on ALM
myProcessAlarm	BOOL	FALSE	**TRUE** = process alarm present
myAlmCtrl_In	Struct_AlmControlIn	-	Structure for input parameter of the **_LineModule_control** FB
myAlmCtrl_Out	Struct_AlmControlOut	-	Structure for output parameter of the **_LineModule_control** FB
myperiInAlm	WORD	16#0000	I/O variable with ALM address for status word
myperiOutAlm	WORD	16#0000	I/O variable with ALM address for control word
myAlmModuleAddress	DINT	256	ALM address for selection of diagnostic alarm in **PeripheralFaultTask**
myPftTsi	Struct_PeripheralFaultTaskTsi	-	Task start information for **PeripheralFaultTask**

StartUpALM program

In the StartUpAlm program, a flag for the initial run is set and then evaluated in the BackgroundTask. This allows for the implementation of a standard start-up sequence for not only the STOP – RUN transition, but also an appropriate user request.

BackGrndALM program

The BackGrndALM program contains 2 program sequences for switching the Active Line Module on and off. In both sequences, the switch-on/switch-off procedure is monitored for errors and tested to ensure it has completed successfully.

The switch-on procedure for the ALM is initiated using a positive edge on the **mySwitchOn** variable. The steps that follow involve testing the switch-on procedure to ensure it has completed successfully and checking it for errors. The **mySwitchOn** variable is set to FALSE once the switch-on procedure has begun.

The switch-off procedure for the ALM is initiated using a positive edge on the **mySwitchOff** variable. The value of the **mySwitchOff** variable is then set to FALSE. The step that follows involves testing the procedure to ensure it has completed successfully and checking it for errors.

Any errors that occurred during processing are displayed in the **myError** and **myErrorId** global variables and can be reset using the **myAlmCtrl_In.reset** variable.

PeripheralFaultALM program

If the module triggering an error is the ALM, the start information for the **PeripheralFaultTask** is written to the **myPftTsi** global variable. If the start information evaluation produces a diagnostic or process alarm signaled by the ALM, this is displayed in the **myDiagnosticAlarm** or **myProcessAlarm** global variable.

Appendix A

A.1 Flow diagrams for switching the Line Modules on and off

Overview

The flow diagrams below describe the statuses (S1 to S4) present while the Line Module is ramping up.

The Line Modules are switched off using the same procedure used to switch them on, but in the reverse order.

Note

For explanations relating to the status and control word of SINAMICS S120 Line Modules, refer to the

- *SINAMICS S120 Drive Functions* Function Manual
- *SIMOTION D4x5* Commissioning and Hardware Installation Manual

These documents are provided as part of the SIMOTION SCOUT scope of delivery in electronic format.

Flow diagram for switching on the Active Line Module (ALM)

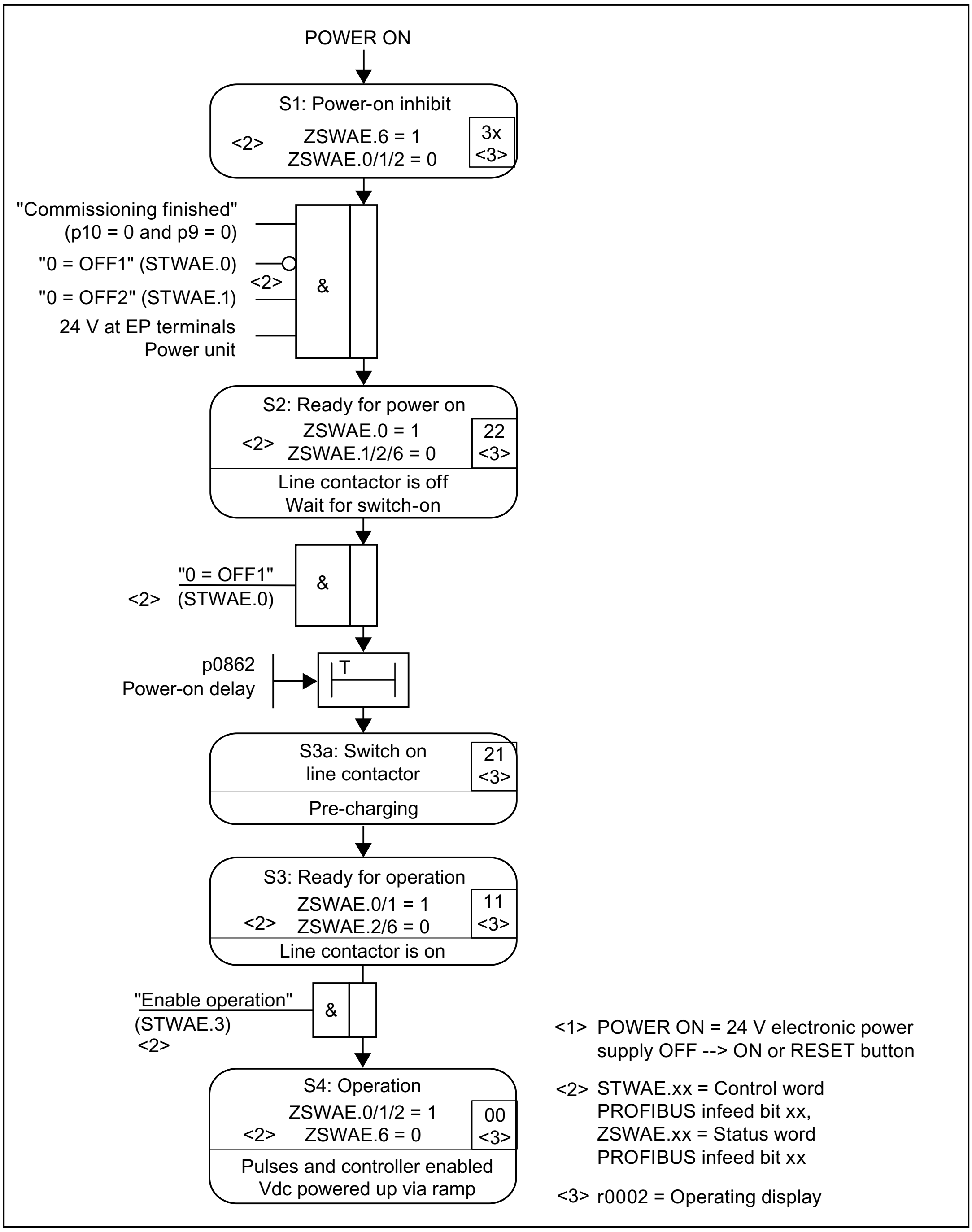


Figure A-1 Flow diagram: Switching on the ALM

Flow diagram for switching on the Basic Line Module (BLM)

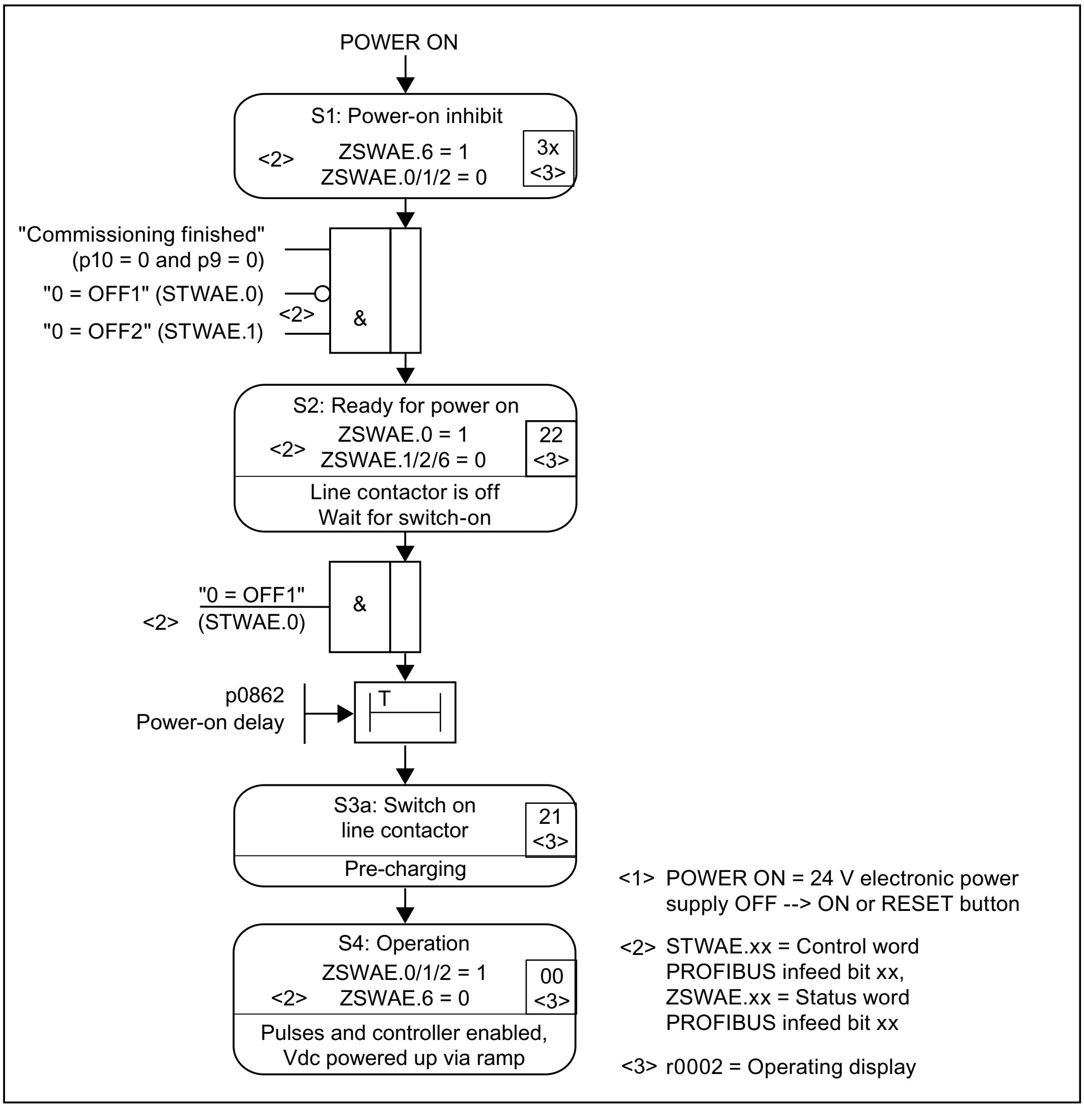


Figure A-2 Flow diagram: Switching on the BLM

Flow diagram for switching on the Smart Line Module (SLM)

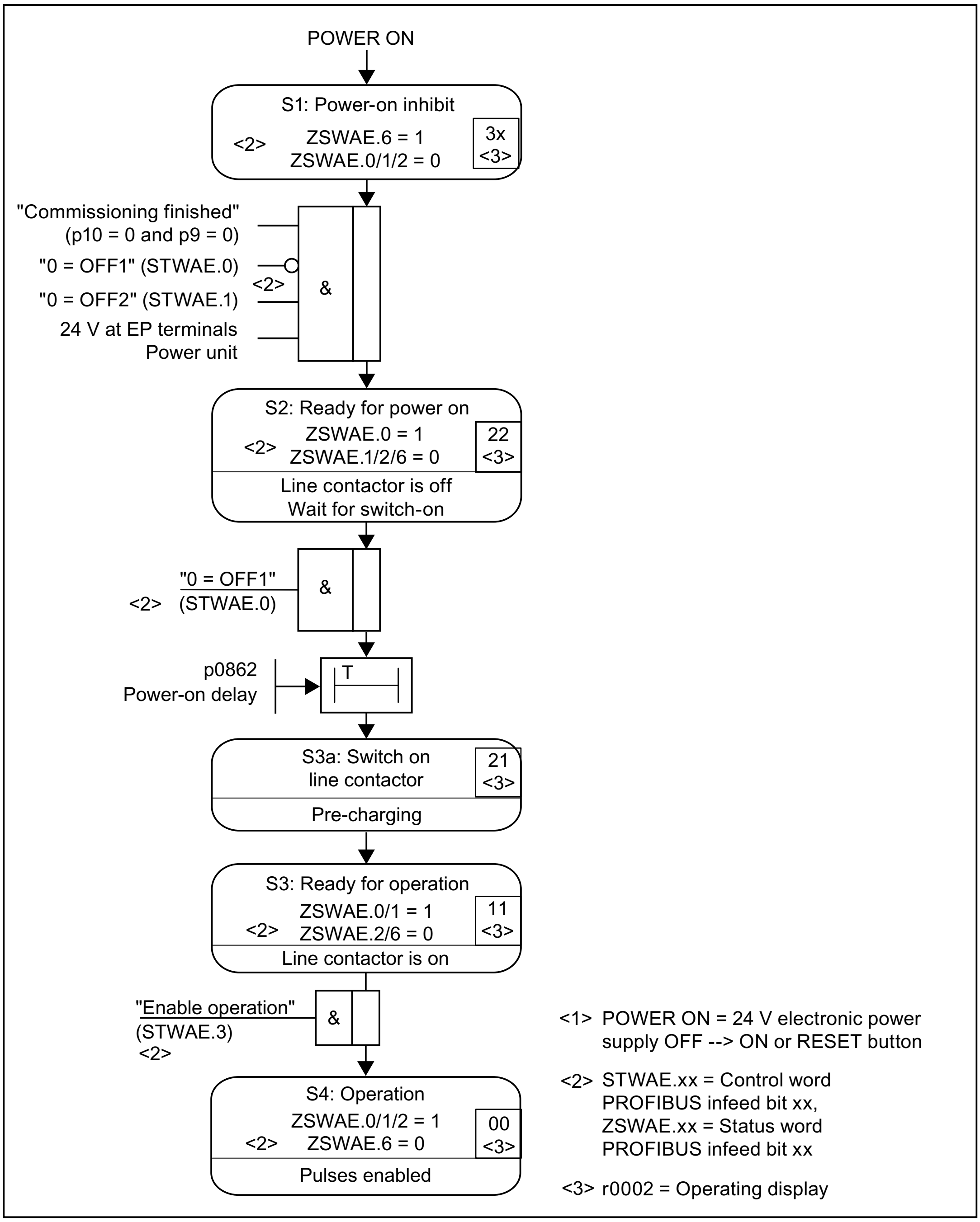


Figure A-3 Flow diagram: Switching on the SLM

A.2 List of abbreviations / acronyms

Table A- 1 Abbreviations

Abbreviation	Meaning
ALM	Active Line Module
BLM	Basic Line Module
DRIVE-CLiQ	DRIVE Component Link with IQ
FB	Function block
IN	Input parameters
I/O	Input/output
LAD	Ladder diagram
LED	Light Emitting Diode
OUT	Output parameter
PLC	Programmable Logic Controller
PZD	Process data
SLM	Smart Line Module
STW	Control word
ZSW	Status word

Index