

SoMachine

TeSys Motor Starters Functions

TeSys Library Guide

04/2014

EI00000000657.04

www.schneider-electric.com

Schneider
 **Electric**

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2014 Schneider Electric. All rights reserved.

Table of Contents



Safety Information	5
About the Book.....	9
Chapter 1 TeSys Function Blocks Offer Overview	11
System Requirements	12
CANopen Function Blocks.....	13
Motor Control Through Different Operation Modes	14
Chapter 2 Network Configuration for TeSys	15
CANopen Network Configuration Procedure for TeSysU Motor Starters	15
Chapter 3 TeSysU_CtrlCmdCyc_CANopen.....	23
Functional Description	24
Input and Output Pin Description.....	25
Instantiation and Usage Example	27
Chapter 4 TeSysT_CtrlCmdCyc_CANopen	29
Functional and Pin Description	30
Input and Output Pin Description.....	31
Instantiation and Usage Example	33
Chapter 5 TeSysU_IO: Controlling the Motor with the TeSysU..	35
Functional Description	36
Input and Output Pin Description.....	39
Structures Used.....	41
Control Double Word Bits Description.....	42
Status Double Word Bits Description	43
Instantiation, Usage Example and Limitations.....	44
Chapter 6 MOT1D1S: Motor, One Direction, One Speed	45
Functional Description	46
Input and Output Pin Description.....	48
Structures Used.....	50
Control Word Bits Description	51
Status Word Bits Description	52
Chapter 7 MOT1D2S: Motor, One Direction, Two Speed	53
Functional Description	54
Input and Output Pin Description.....	57
Structures Used.....	59
Control Double Word Bits Description.....	60
Status Double Word Bits Description	61

Chapter 8	MOT2D1S: Motor, Two Direction, One Speed	63
Functional Description	64	
Input and Output Pin Description	67	
Structures Used	69	
Control Double Word Bits Description	70	
Status Double Word Bits Description	71	
Chapter 9	MOT2D2S: Motor, Two Direction, Two Speed	73
Functional Description	74	
Input and Output Pin Description	77	
Structures Used	79	
Control Word Bits Description	80	
Status Word Bits Description	81	
Instantiation and Usage Example	82	
Glossary	83
Index	87

Safety Information



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a Danger safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation which, if not avoided, **can result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation which, if not avoided, **can result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

WARNING

UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only the user can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine; therefore, only the user can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, the user should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

CAUTION

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in injury or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book



At a Glance

Document Scope

This document describes the functions of the SoMachine TeSys Library.

Validity Note

This document has been updated with the release of SoMachine V4.0.

Related Documents

Title of Documentation	Reference Number
TeSys T LTM R Modbus Motor Management Controller User Manual	1639501
TeSys T LTM R CANopen Motor Management Controller User Manual	1639503
TeSys U LUCM and LUCMT Multifunction Control Unit User Manual	1743237
TeSys U LULC08 CANopen Communication Module User Manual	1744084
TeSys U LULC032-033 Modbus Communication Module User Manual	1743234
TeSysU Communication Variables User Manual	1744082

You can download these technical publications and other technical information from our website at www.schneider-electric.com.

Product Related Information

WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.¹
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

Chapter 1

TeSys Function Blocks Offer Overview

Overview

This chapter describes the function blocks offer overview of TeSys Library.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
System Requirements	12
CANopen Function Blocks	13
Motor Control Through Different Operation Modes	14

System Requirements

Using the Library

WARNING

UNINTENDED EQUIPMENT OPERATION

- Verify the SoMachine libraries contained in your program are the correct version after updating SoMachine software.
- Verify that the library versions updated are consistent with your application specifications.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

For more detailed information, see Schneider Electric Libraries (see *SoMachine, Functions and Libraries User Guide*).

CANopen Function Blocks

Function Block Offer Description

This table describes the function block offer when TeSys are connected through a CANopen network:

Function Block	Description	For more information
TeSysU_CtrlCmdCyc_CANopen	This function block is dedicated to the control and command of a single TeSysU starter controller (up to 32 A/15 kW or 20 hp) with any control unit and a LULC08 CANopen communication module.	Refer TeSysU_CtrlCmdCyc_CANopen chapter (see page 23)
TeSysT_CtrlCmdCyc_CANopen	This function block is dedicated to the control and command of a single TeSys T LTMR•C• CANopen controller with or without the LTM E expansion module.	Refer TeSysT_CtrlCmdCyc_CANopen chapter (see page 29)

Motor Control Through Different Operation Modes

Function Blocks Description

These function blocks lets you control motors in various conditions (one direction, two directions, one speed, two speeds) through 3 different modes; automatic, manual using push-buttons, or manual using HMI.

Depending on the selected operation mode, one of 3 different groups of inputs is used to control the function block outputs.

The function blocks outputs can activate a motor starter connected through field bus or in parallel through digital inputs/outputs.

The table below describes the various possibilities of motor control:

Function Block	Description	For more information
TeSysU_IO	This function block is dedicated to the control and command of a single TeSysU starter-controller (up to 32 A/ 15 kW or 20 hp) one or two directions (reverser) with any control unit connected in parallel through digital Inputs/Outputs or through a communication module Modbus LULC033 or CANopen LULC08.	Refer TeSysU_IO chapter (see page 35)
MOT1D1S	This generic function block is dedicated to the control and command of motors with one direction of rotation and one speed through any actuator.	Refer MOT1D1S chapter (see page 45)
MOT1D2S	This generic function block is dedicated to the control and command of motors with one direction of rotation and two speeds through any actuator.	Refer MOT1D2S chapter (see page 53)
MOT2D1S	This generic function block is dedicated to the control and command of motors with two directions of rotation and one speed through any actuator.	Refer MOT2D1S chapter (see page 63)
MOT2D2S	This generic function block is dedicated to the control and command of motors with two directions of rotation and two speeds through any actuator.	Refer MOT2D2S chapter (see page 73)

Chapter 2

Network Configuration for TeSys

CANopen Network Configuration Procedure for TeSysU Motor Starters

Overview

This procedure describes the integration of M238 logic controller and TeSysU motor starters in a CANopen network using SoMachine.

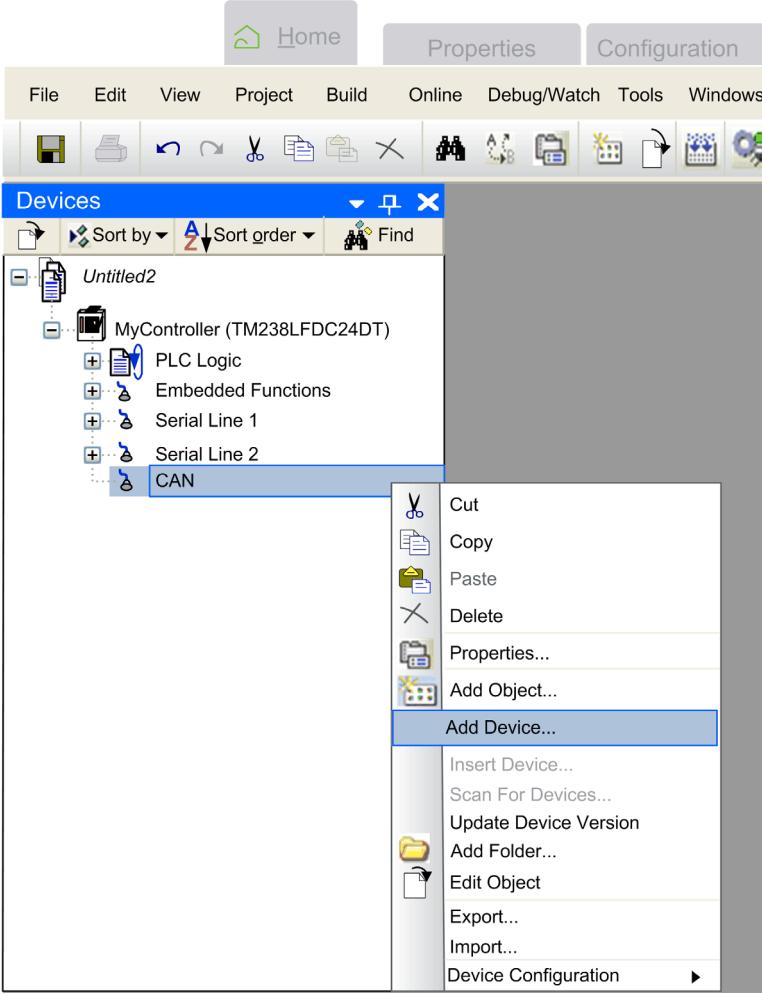
NOTE: The principles to integrate a TeSysT motor management system are the same.

Important Information

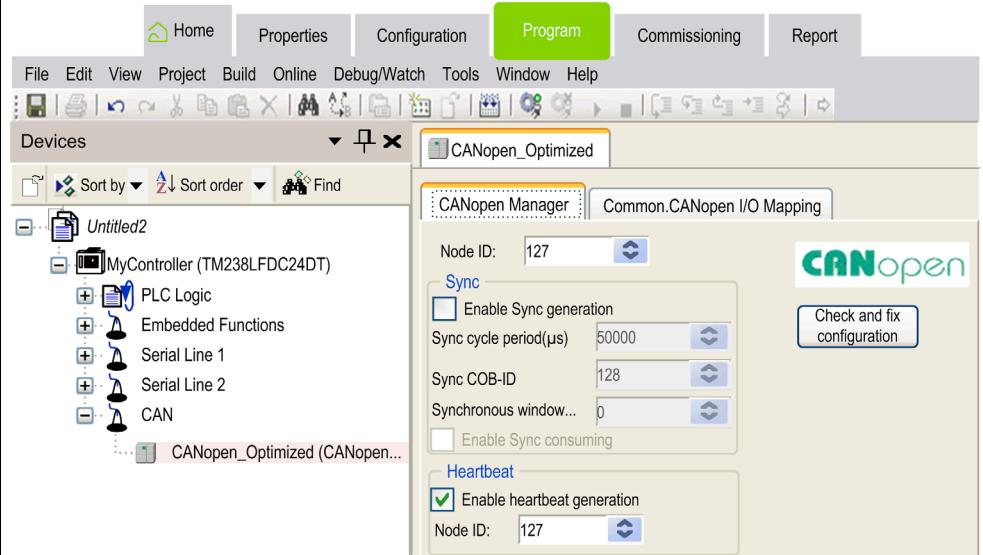
For electrical connection, refer to manuals mentioned in Related Documents.

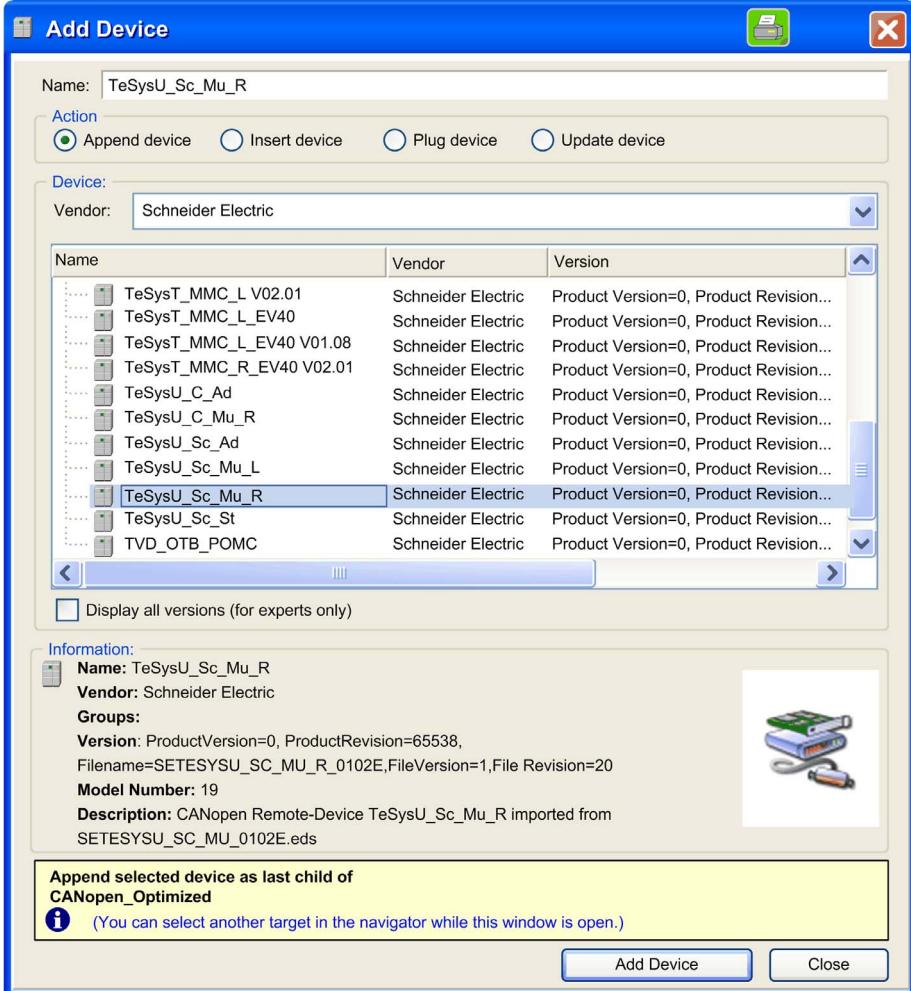
Before instantiating the TeSys device function blocks in SoMachine application project, configure the CANopen network as given in the following procedure:

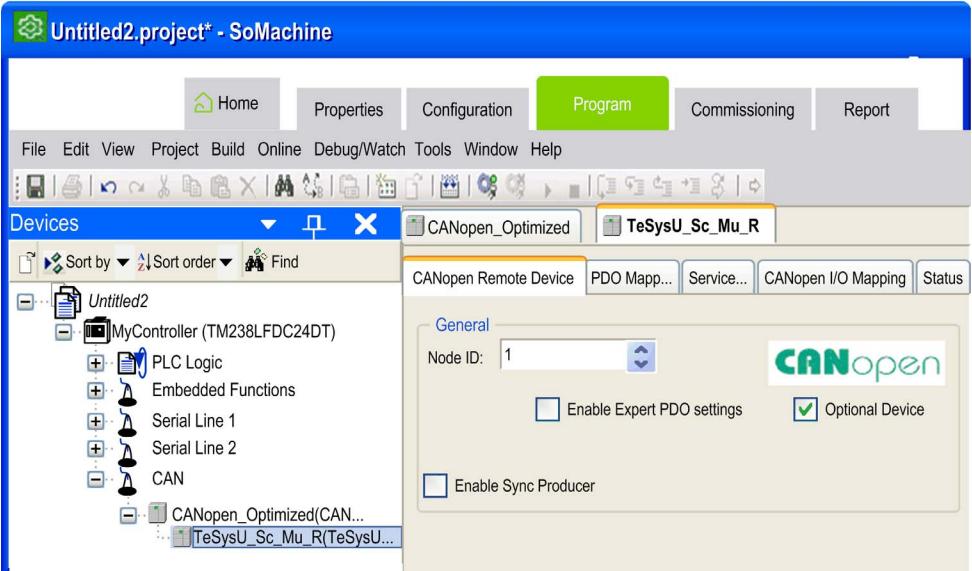
Configuring the CANopen Network

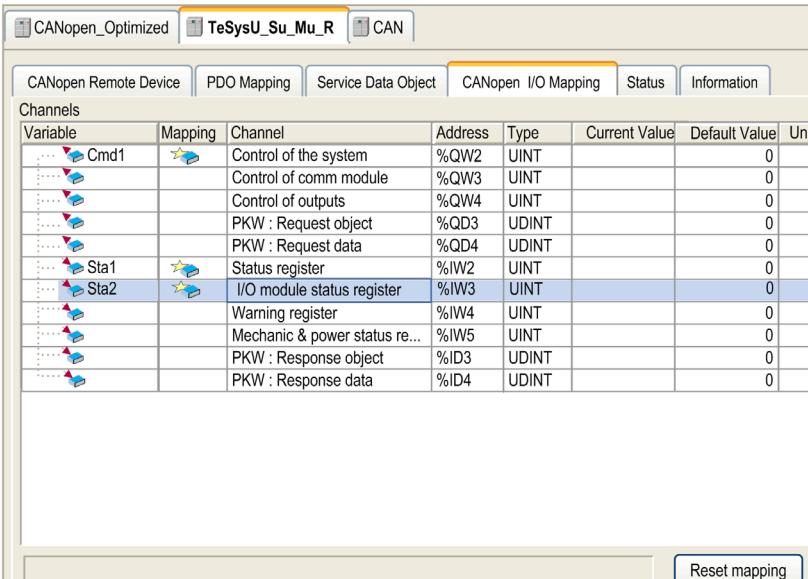
Step	Action
1	Open new SoMachine project.
2	Right-click CAN and select Add Device as shown in the figure below.  <p>The screenshot shows the SoMachine software interface. The top menu bar includes Home, Properties, Configuration, File, Edit, View, Project, Build, Online, Debug/Watch, Tools, and Windows. Below the menu is a toolbar with various icons. The main window displays a tree view under the heading 'Devices'. The tree structure is as follows:</p> <ul style="list-style-type: none">Untitled2<ul style="list-style-type: none">MyController (TM238LFDC24DT)<ul style="list-style-type: none">PLC LogicEmbedded FunctionsSerial Line 1Serial Line 2CAN <p>A context menu is open over the 'CAN' node, listing the following options: Cut, Copy, Paste, Delete, Properties..., Add Object..., Add Device..., Insert Device..., Scan For Devices..., Update Device Version, Add Folder..., Edit Object, Export..., Import..., and Device Configuration. The 'Add Device...' option is highlighted with a blue selection bar.</p> <p>Result: The Add Device window appears.</p>

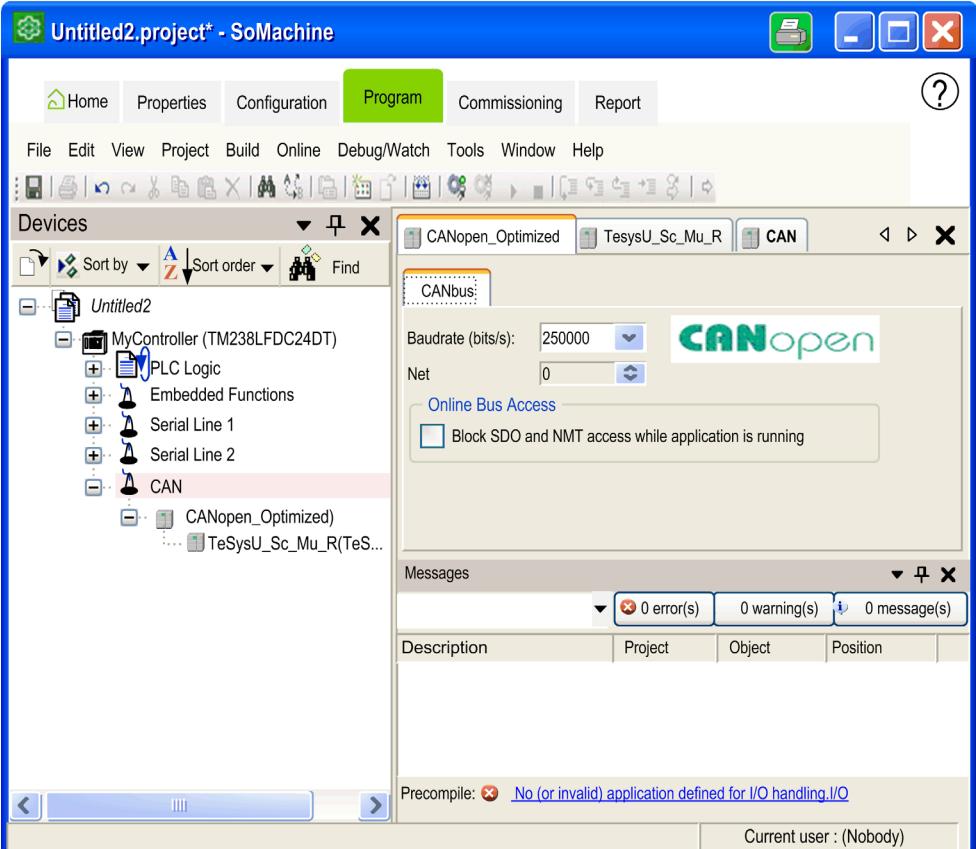
Step	Action
3	<p>Select CANopen Optimized as shown in the figure below.</p> <p>Result: CANopen_Optimized (CANopen Optimized) is added under CAN. Refer figure in Step 5.</p>
4	<p>Close the Add Device window and double-click CANopen_Optimized (CANopen Optimized).</p>

Step	Action
5	<p>Select the CANopen Manager tab. In the Node ID text field, enter a unique node number for CANopen_Optimized as shown in the figure below.</p>  <p>The screenshot shows the CANopen Manager configuration window. The Node ID is set to 127. Other settings include Sync cycle period at 50000 µs, Sync COB-ID at 128, and Synchronous window at 0. There is also an option to enable Sync consuming and another for Enable heartbeat generation, which is checked. A 'Check and fix configuration' button is visible.</p>
6	<p>Right-click CANopen_Optimized (CANopen Optimized) and select Add Device. Result: The Add Device window appears, showing the TeSysU motor starters with multifunction control unit. Refer figure in Step 7.</p>

Step	Action
7	<p>To add the TeSysU motor starters with multifunction control unit in remote configuration mode, double-click TeSysU_Sc_Mu_R and close the Add Device window. Refer figure below.</p>  <p>The screenshot shows the 'Add Device' dialog box. In the 'Name' field, 'TeSysU_Sc_Mu_R' is entered. Under 'Action', 'Append device' is selected. In the 'Device' section, 'Vendor' is set to 'Schneider Electric'. A list of products is displayed, with 'TeSysU_Sc_Mu_R' highlighted. Below the list, there's a checkbox for 'Display all versions (for experts only)'. The 'Information' panel shows details for 'TeSysU_Sc_Mu_R', including its name, vendor, version information, and a description as a CANopen Remote-Device. A note at the bottom says 'Append selected device as last child of CANopen_Optimized' with a tip about selecting another target. At the bottom right are 'Add Device' and 'Close' buttons.</p> <p>NOTE: The letter at the end of the EDS file name indicates if configuration is managed remotely through the network (R) or locally through the configuration port (L). For proper communication, the TeSys device should be configured in accordance (parameter 601 : configuration mode) to be able to communicate properly.</p> <p>Result: TeSysU_Sc_Mu_R is added under CAN. Refer figure in step 8.</p>

Step	Action
8	<p>Double-click TeSysU_Sc_Mu_R. Under the CANopen Remote Device tab, set a unique node number in the Node ID text field as shown in the figure below.</p>  <p>NOTE: The same node number should be set in the slave device using DIP switches for TeSysU and PowerSuite/SoMove through the HMI port for TeSysT as applicable. Refer to the user manual of the respective device for more information.</p>
9	<p>From the same window (figure in step 8), select CANopen I/O Mapping tab. Result: A new window appears with channels description. Refer figure in step 10.</p>

Step	Action																																																																																																
10	<p>Under the column Variable, enter the variable name (shown in the below figure as Cmd1, Sta1, and Sta2 for control register 1, for status register and for I/O module status register respectively for TeSysT) against the Address column based on the information to be read/written from the CANopen device.</p>  <table border="1" data-bbox="259 399 1067 677"> <thead> <tr> <th>Variable</th> <th>Mapping</th> <th>Channel</th> <th>Address</th> <th>Type</th> <th>Current Value</th> <th>Default Value</th> <th>Unit</th> </tr> </thead> <tbody> <tr> <td>Cmd1</td> <td></td> <td>Control of the system</td> <td>%QW2</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Control of comm module</td> <td>%QW3</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Control of outputs</td> <td>%QW4</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>PKW : Request object</td> <td>%QD3</td> <td>UDINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>PKW : Request data</td> <td>%QD4</td> <td>UDINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td>Sta1</td> <td></td> <td>Status register</td> <td>%IW2</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr style="background-color: #d9e1f2;"> <td>Sta2</td> <td></td> <td>I/O module status register</td> <td>%IW3</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Warning register</td> <td>%IW4</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>Mechanic & power status re...</td> <td>%IW5</td> <td>UINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>PKW : Response object</td> <td>%ID3</td> <td>UDINT</td> <td></td> <td>0</td> <td></td> </tr> <tr> <td></td> <td></td> <td>PKW : Response data</td> <td>%ID4</td> <td>UDINT</td> <td></td> <td>0</td> <td></td> </tr> </tbody> </table> <p style="text-align: right;">Reset mapping</p>	Variable	Mapping	Channel	Address	Type	Current Value	Default Value	Unit	Cmd1		Control of the system	%QW2	UINT		0				Control of comm module	%QW3	UINT		0				Control of outputs	%QW4	UINT		0				PKW : Request object	%QD3	UDINT		0				PKW : Request data	%QD4	UDINT		0		Sta1		Status register	%IW2	UINT		0		Sta2		I/O module status register	%IW3	UINT		0				Warning register	%IW4	UINT		0				Mechanic & power status re...	%IW5	UINT		0				PKW : Response object	%ID3	UDINT		0				PKW : Response data	%ID4	UDINT		0	
Variable	Mapping	Channel	Address	Type	Current Value	Default Value	Unit																																																																																										
Cmd1		Control of the system	%QW2	UINT		0																																																																																											
		Control of comm module	%QW3	UINT		0																																																																																											
		Control of outputs	%QW4	UINT		0																																																																																											
		PKW : Request object	%QD3	UDINT		0																																																																																											
		PKW : Request data	%QD4	UDINT		0																																																																																											
Sta1		Status register	%IW2	UINT		0																																																																																											
Sta2		I/O module status register	%IW3	UINT		0																																																																																											
		Warning register	%IW4	UINT		0																																																																																											
		Mechanic & power status re...	%IW5	UINT		0																																																																																											
		PKW : Response object	%ID3	UDINT		0																																																																																											
		PKW : Response data	%ID4	UDINT		0																																																																																											

Step	Action
11	<p>Double-click CAN tab. In the Baudrate (bits/s) list, select the CANopen network baudrate as shown in the figure below.</p>  <p>NOTE: The same baudrate should be set in the slave device using DIP switches for TeSysU and PowerSuite/SoMove through the HMI port for TeSysT as applicable. Refer to the user manual of the respective device for more information.</p> <p>Result: CANopen network configuration is complete.</p>

Chapter 3

TeSysU_CtrlCmdCyc_CANopen

Overview

This chapter explains the TeSysU_CtrlCmdCyc_CANopen function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	24
Input and Output Pin Description	25
Instantiation and Usage Example	27

Functional Description

Function Block Description

The TeSysU_CtrlCmdCyc_CANopen function block is dedicated to the control and command of a single TeSysU motor starters (up to 32 A/15 kW/20 hp) through TeSysU registers exchanged through CANopen PDOs.

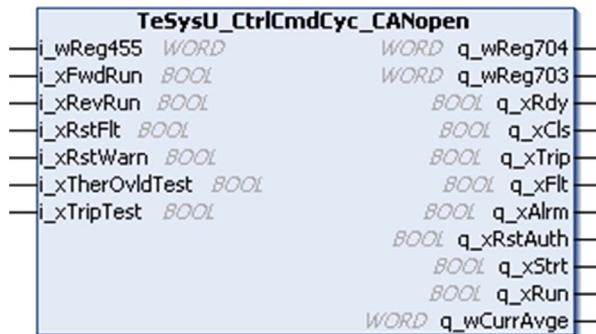
TeSysU Compliance

The TeSysU_CtrlCmdCyc_CANopen function blocks are compliant with the following TeSysU sub assemblies:

Type	Subassembly Name
Power base	<ul style="list-style-type: none"> LUBxx non-reversing power base (up to 32 A/15 kW/20 hp) LU2Bxx reversing power base (up to 32 A/15 kW/20 hp)
Control unit	<ul style="list-style-type: none"> LUCA standard control unit LUCB, LUCC, and LUCD advanced control units LUCM multi-function control unit LUCL magnetic control unit
Communication module	<ul style="list-style-type: none"> LULC08 CANopen communication module

Pin Diagram

This figure shows the pin diagram of the TeSysU_CtrlCmdCyc_CANopen function block:



Input and Output Pin Description

Input Pin Description

This table describes the input pins of the TeSysU_CtrlCmdCyc_CANopen function block:

Input	Data Type	Description	Control Unit
i_wReg455	WORD	This input must be linked to the status register that is transported in the CANopen TxPDO1 parameter.	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
i_xFwdRun	BOOL	Motor run forward command	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
i_xRevRun	BOOL	Motor run reverse command	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
i_xRstFlt	BOOL	Reset detected error	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
i_xRstWarn	BOOL	Reset communication loss alert	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
i_xTherOvldTest	BOOL	Automatic thermal overload test	<ul style="list-style-type: none"> ● LUCM
i_xTripTest	BOOL	Overcurrent trip test	<ul style="list-style-type: none"> ● LUCM

Output Pin Description

This table describes the output pins of the TeSysU_CtrlCmdCyc_CANopen function block:

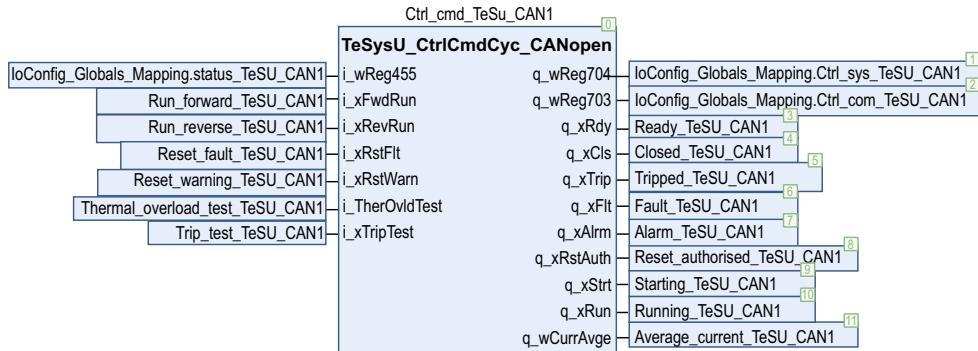
Output	Data Type	Description	Control Unit
q_wReg704	WORD	This output must be linked to control of the system register that is transported in the CANopen RxPDO1 parameter.	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_wReg703	WORD	This output must be linked to the control of communication module register that is transported in the CANopen RxPDO1 parameter.	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM

Output	Data Type	Description	Control Unit
q_xRdy	BOOL	TeSysU rotary handle is turned to On position and there is no detected error.	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_xCls	BOOL	Pole status: closed	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_xTrip	BOOL	TeSysU rotary handle is turned to Trip position.	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_xFlt	BOOL	All detected errors	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_xAlrm	BOOL	All alerts	<ul style="list-style-type: none"> ● LUCA/LUCL ● LUCB/C/D ● LUCM
q_xRstAuth	BOOL	Detected error reset authorized status	<ul style="list-style-type: none"> ● LUCB/C/D ● LUCM
q_xStrt	BOOL	Start in progress 0 = Descending current is lower than 150% FLA 1 = Ascending current is greater than 10% FLA	<ul style="list-style-type: none"> ● LUCB/C/D ● LUCM
q_xRun	BOOL	Motor runs with detection of current, if greater than 10% FLA.	<ul style="list-style-type: none"> ● LUCB/C/D ● LUCM
q_wCurrAvge	WORD	Average motor current (% FLA) 32 = 100% FLA 63 = 200% FLA Range: 0...63	<ul style="list-style-type: none"> ● LUCB/C/D ● LUCM

Instantiation and Usage Example

Instantiation and Usage Example

This figure shows an instantiation example of the TeSysU_CtrlCmdCyc_CANopen function block:



This figure shows a visualization for the associated CANopen I/O Mapping dialog of TeSysU:

CANopen I/O Mapping								
Variable	Map...	Channel	Address	Type	Current Value	Default Value	Unit	Description
Ctrl_sys...	Control of the system	%QW2	UINT			0		
Ctrl_com...	Control of comm module	%QW3	UINT			0		
...	Control of outputs	%QW4	UINT			0		
...	PKW : Request object	%QD3	UDINT			0		
...	PKW : Request data	%QD4	UDINT			0		
...	Status_T...	%IW2	UINT			0		
...	I/O module status register	%IW3	UINT			0		
...	PKW : Response object	%ID2	UDINT			0		
...	PKW : Response data	%ID3	UDINT			0		

Chapter 4

TeSysT_CtrlCmdCyc_CANopen

Overview

This chapter explains the `TeSysT_CtrlCmdCyc_CANopen` function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional and Pin Description	30
Input and Output Pin Description	31
Instantiation and Usage Example	33

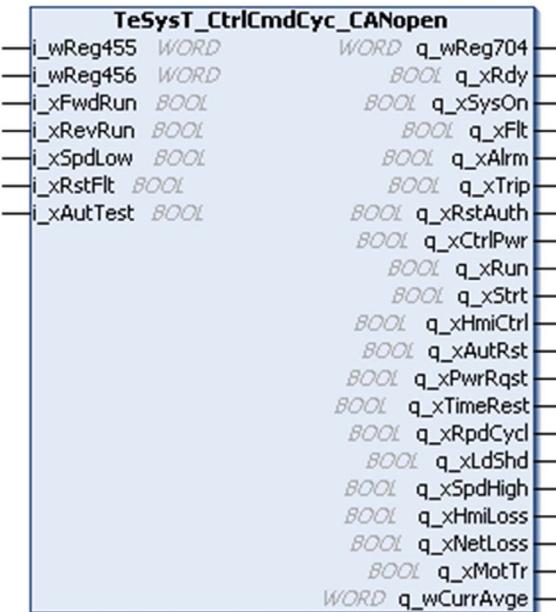
Functional and Pin Description

Function Block Description

The TeSysT_CtrlCmdCyc_CANopen function block is dedicated to the control and command of a single TeSys T LTM R•C• CANopen controller with TeSys T registers exchanged through CANopen PDOs.

Pin Diagram

This figure shows the pin diagram of the TeSysT_CtrlCmdCyc_CANopen function block:



Requirements

The TeSysT_CtrlCmdCyc_CANopen function block is compliant with all the TeSys T LTM R•C• CANopen controller versions, with or without the LTM E expansion module.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of the TeSysT_CtrlCmdCyc_CANopen function block:

Input	Data Type	Description
i_wReg455	WORD	This input must be linked to the System Register 1 that is transported in the CANopen TxPDO1 parameter.
i_wReg456	WORD	This input must be linked to the System Register 2 that is transported in the CANopen TxPDO1 parameter.
i_xFwdRun	BOOL	Motor run forward command
i_xRevRun	BOOL	Motor run reverse command
i_xSpdLow	BOOL	Motor low speed command (Optional, depends on TeSysT operation mode)
i_xRstFlt	BOOL	Detected error reset command
i_xAutTest	BOOL	Self test command

Output Pin Description

This table describes the output pins of the TeSysT_CtrlCmdCyc_CANopen function block:

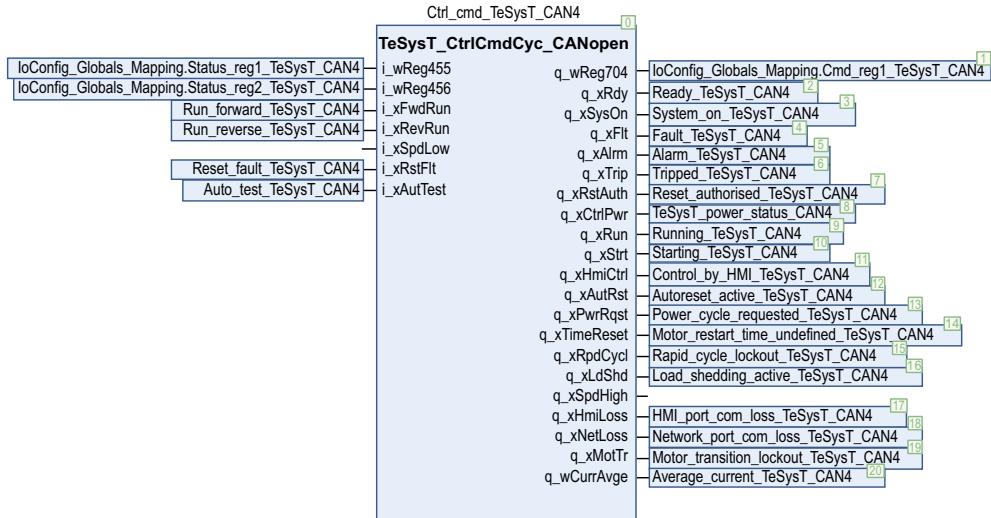
Output	Data Type	Description
q_wReg704	WORD	This output must be linked to the Command Register 1 that is transported in the CANopen RxPDO1 parameter.
q_xRdy	BOOL	System ready
q_xSysOn	BOOL	System On
q_xFlt	BOOL	System error detection
q_xAlrm	BOOL	System alarm
q_xTrip	BOOL	System tripped
q_xRstAuth	BOOL	Detected error reset authorized
q_xCtrlPwr	BOOL	Controller power status
q_xRun	BOOL	Motor running (with detection of a current, if greater than 10% FLC)
q_xStrt	BOOL	Motor starting (start in progress) 0 = descending current is less than 150% FLC 1 = ascending current is greater than 10% FLC
q_xHmiCtrl	BOOL	Control through HMI
q_xAutRst	BOOL	Auto-reset active

Output	Data Type	Description
q_xPwrRqst	BOOL	Power cycle requested
q_xTimeRest	BOOL	Motor restart time undefined
q_xRpdCycl	BOOL	Rapid cycle lockout
q_xLdShd	BOOL	Load shedding
q_xSpdHigh	BOOL	Motor speed 0 = FLC1 setting is used 1 = FLC2 setting is used (Optional, depends on TeSysT operation mode)
q_xHmiLoss	BOOL	HMI port communication loss
q_xNetLoss	BOOL	Network port communication loss
q_xMotTr	BOOL	Motor transition lockout
q_wCurrAvge	WORD	Motor average current ratio 32 = 100% FLC 63 = 200% FLC Range: 0...63

Instantiation and Usage Example

Instantiation and Usage Example

This figure shows an instantiation example of the TeSysT_CtrlCmdCyc_CANopen function block:



This figure shows a visualization for the associated CANopen I/O Mapping dialog of TeSysT:

CANopen Remote Device	PDO Mapping	Service Data Object	CANopen I/O Mapping	Status	Information
Channels					
Variable	Mapping	Channel	Address	Type	Current Value
Cmd_reg1_TeSysT_CAN4	Command register 1	%QW26	UINT		0
	Analog output 1 command	%QW27	UINT		0
	Logic outputs command	%QW28	UINT		0
PKW_req_obj_TeSysT_CAN4	PKW : Request object	%QD15	UDINT		0
	PKW : Request data	%QD16	UDINT		0
Status_reg1_TeSysT_CAN4	System status register 1	%IW24	UINT		0
Status_reg2_TeSysT_CAN4	System status register 2	%IW25	UINT		0
	Logic inputs status	%IW26	UINT		0
	Logic outputs status	%IW27	UINT		0
PKW_res_obj_TeSysT_CAN4	PKW : Response object	%ID14	UDINT		0
PKW_res_data_TeSysT_CAN4	PKW : Response data	%ID15	UDINT		0

Chapter 5

TeSysU_IO: Controlling the Motor with the TeSysU

Overview

This chapter describes the TeSysU_IO function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	36
Input and Output Pin Description	39
Structures Used	41
Control Double Word Bits Description	42
Status Double Word Bits Description	43
Instantiation, Usage Example and Limitations	44

Functional Description

Overview

The TeSysU_IO function block is dedicated to the control and command of a single TeSysU motor starters (up to 32 A/15 kW or 20 hp) in one or two directions (reverser) with any control unit connected in parallel through digital inputs/outputs or through a communication module Modbus LULC033 or CANopen LULC08.

The motor can be controlled through 3 different operating modes; automatic, manual using push buttons, or manual using HMI.

TeSysU Compliance

The TeSysU_IO function block is compliant with the following TeSysU sub-assemblies:

Type	Subassembly Name
Power base	<ul style="list-style-type: none"> LUBxx non-reversing power base (up to 32 A/15 kW/20 hp) LU2Bxx reversing power base (up to 32 A/15 kW/20 hp)
Control unit	<ul style="list-style-type: none"> LUCA standard control unit LUCB, LUCC, and LUCD advanced control units LUCL magnetic control unit
Communication module	Without (parallel connection through removable screw terminal blocks) <ul style="list-style-type: none"> LULC033 Modbus serial line communication module LULC08 CANopen communication module

Pin Diagram

This figure shows the pin diagram of the TeSysU_IO function block:



Operation Modes

The TeSysU_IO function block is used for controlling the motor with the TeSysU. It has three modes of operation:

- **Automatic Mode:** The Automatic mode is selected through the input pin `i_xAut`. In Automatic mode, the motor is started and stopped in the forward direction through the input pin `i_xFwdAut` regardless of the activation of local mode. Similarly, the reverse direction is handled with the input pin `i_xRevAut`. These digital inputs are controlled by the controller process application during normal functioning.
- **Manual Mode:** The Manual mode is activated by the pin `i_xMan`.
 - Case 1: Local mode is not activated. The motor is started and stopped through the bit commands of the signal `i_dwCtrl`. This double word can be associated to an external HMI equipped with a keypad.
 - Case 2: Local mode is activated through the input pin `i_xLoc`. The motor is started through the inputs `i_xFwdLoc` and stopped through `i_xLocStop` in forward direction, and the inputs `i_xRevLoc` and `i_xLocStop` are used to manually control the motor in reverse direction. While the motor is running in one direction, it cannot be run in opposite direction until it is stopped by `i_xLocStop`. These digital inputs can be linked to connect push-buttons.
- **Local Mode:** The Local mode is activated by an input pin `i_xLoc` and is set additionally to the automatic or manual mode. The local mode does not influence the automatic mode, but changes the source for manual operation.

The block is de-activated on controller start and remains in the same operation mode, unless a new one is selected. If both automatic and manual modes are selected simultaneously (inputs `i_xAut` and `i_xMan` are set to TRUE), the operation mode is invalid, which is indicated at the `q_xErr` output.

CAUTION

UNINTENDED MOTOR STOP

Do not change from manual mode to automatic mode when the motor is being used, doing so will make the motor stop.

Failure to follow these instructions can result in injury or equipment damage.

Any other change of the operation mode does not affect the motor operation

Starting in Reverse Direction

The motor cannot be started in reverse direction, when already running in forward (output `q_xFwd` set to TRUE) and vice versa. This direct change between the two directions can be avoided by proper time supervision. The delay time for the change of direction can be freely chosen through the structure element `iRevDly` at the input `i_strPara`. If the time is set to 0, the change takes place immediately.

Supervising the Motor

The operation of the motor is supervised by feedback signal (`i_xFbckRun`), which indicates the running motor. The feedback signal must adjust its value to the value of the related output `q_xFwd/q_xRev` within a defined time. If the time exceeds, the block indicates an error detection. The time can be set through the structure element `iFbckDly` at the input `i_strPara`. The supervision can be switched off by the structure element `xFbckEn` at the input `i_strPara`.

Running the Motor

The motor can run, only if the interlock input `i_xLock` is set to FALSE. An active interlock signal inhibits the start of the motor, and stops a running motor. When the interlock signal returns to FALSE, the motor is restarted. An active interlock is indicated at the output `q_xLock`.

The motor can only be run if the output detected error is set to FALSE. An active detected error signal inhibits the start of the motor or stops a running motor. The function block sets the detected error signal, if the detected error input `i_xErr` is set to TRUE (external detected error) or in case of an invalid operation mode, a missing voltage signal (`i_xFbckVolt` = FALSE), a detected trip error (`i_xFbckErr`) or a missing run feedback signal (internal detected error). The single errors are indicated in the HMI as alarms. To reset the detected error output the detected error has to be acknowledged by a rising edge on the input `i_xAckn` or by using bit 16 of the signal `i_dwCtrl`.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of the TeSysU_IO function block:

Input	Data Type	Description
i_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
i_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
i_xLoc	BOOL	TRUE: Local mode enabled FALSE: Disabled
i_xFwdLoc	BOOL	Rising edge from 0 to 1 manually starts the motor in forward direction in local mode. Manual and Local modes to be set to 1 simultaneously.
i_xRevLoc	BOOL	Rising edge from 0 to 1 manually starts the motor in reverse direction in local mode. Manual and Local modes to be set to 1 simultaneously.
i_xLocStop	BOOL	Rising edge from 0 to 1 manually stops the motor operation in local mode. Manual and Local modes to be set to 1 simultaneously.
i_xFwdAut	BOOL	TRUE: Forward command is active in automatic mode FALSE: Forward command is inactive.
i_xRevAut	BOOL	TRUE: Reverse command is active in automatic mode FALSE: Reverse command is inactive
i_xFbckVolt	BOOL	TRUE: Feedback signal from TeSysU is active: voltage available, switch is in On position FALSE: No feedback
i_xFbckErr	BOOL	TRUE: Feedback signal from TeSysU is active: detected error, switch in position Trip FALSE: No feedback
i_xFbckRun	BOOL	TRUE: Feedback signal from TeSysU is active: motor running (forward or reverse direction) FALSE: No feedback

Input	Data Type	Description
i_xLock	BOOL	TRUE: Interlock enabled FALSE: Disabled Interlock input. Operation is inhibited, when the input is set to 1.
i_xErr	BOOL	TRUE: Externally detected error is active FALSE: No externally detected error
i_xAckn	BOOL	Acknowledgement is done with a rising edge. Input to acknowledge internally and externally detected errors that are indicated at the output q_xErr.
i_strPara	STRUCT Par_Mot_TesysU	Structure with parameters for this block. Refer to the structures used (see page 41).
i_dwCtrl	DWORD	Command bits to interact from an HMI keypad Range: 0...4294967295 Refer to the control bit description (see page 42).

Output Pin Description

This table describes the output pins of the TeSysU_IO function block:

Output	Data Type	Description
q_xAut	BOOL	TRUE: Automatic mode is enabled FALSE: Disabled
q_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
q_xFwd	BOOL	TRUE: Forward command is active FALSE: Disabled
q_xRev	BOOL	TRUE: Reverse command is active FALSE: Disabled
q_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock Indicates that the operation is blocked by an interlock (input i_xLock).
q_xErr	BOOL	TRUE: Detected error is active FALSE: No detected error
q_dwStat	DWORD	Status bits to be displayed in an HMI Refer to status double word bits description (see page 43).

Structures Used

Par_Mot_TesysU

Structure Element	Data Type	Description
xFbckEn	BOOL	Enable feedback signal supervision.
iFbckDly	INT	Delay time in seconds to get the feedback signal from the TeSysU.
iRevDly	INT	Delay time in seconds, to reverse the direction of the running motor.

Control Double Word Bits Description

Functionality

This table describes the control double word bits that can be managed by an HMI keypad:

Bit Position	Description
0...7	Not used
8	Runs the motor forward in manual mode (rising-edge triggered).
9	Runs the motor reverse in manual mode (rising-edge triggered).
10, 11	Not used
12	Stops motor (rising-edge triggered) in manual mode.
13...15	Not used
16	Acknowledges internally and externally detected errors that are indicated at the output <code>q_xErr</code> (rising-edge triggered).
17...31	Not used

Status Double Word Bits Description

Functionality

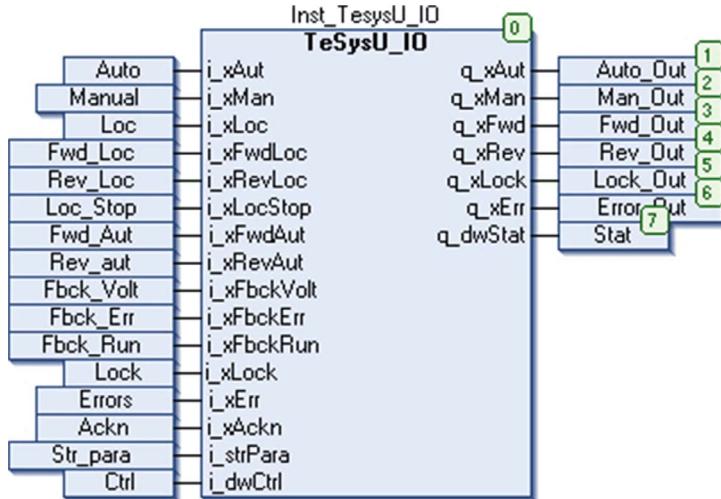
This table describes the status double word bits that can be managed by an HMI display:

Bit Position	Description
0	Automatic mode is activated.
1	Manual mode is activated.
2	Indicates that block is in local mode.
3	Indicates that the operation is blocked by an interlock.
4...7	Not used
8	Signal to run the motor in forward direction.
9	Signal to run the motor in reverse direction.
10...13	Not used
14	If <i>i_strPara.xFbckEn</i> = 0, then Feedback= (<i>q_xFwd</i> or <i>q_xRev</i>). Else Feedback = <i>i_xFbckRun</i> .
16	Resets the detected error indicated at <i>q_xErr</i> .
17	Indicates that the operation is blocked by an internally or externally (Input <i>i_xErr</i>) detected error, which is not acknowledged.
19	Enables feedback signal supervision.
18...23	Not used
24	Invalid operating mode error detection.
25	Externally detected error.
26	Missing feedback detected error.
27	No voltage error detection.
28	Device trip error detection.
29...31	Not used

Instantiation, Usage Example and Limitations

Instantiation and Usage Example

This figure shows an instance of the TeSysU_IO function block in online mode of SoMachine:



Auto mode is selected through the pin `i_xAut` which is indicated at the output `q_xAut`. The motor is given command to run in forward direction through the input pin `i_xFwdAut` as indicated at the output `q_xFwd`. The feedback signal `i_xFbckRun` signal is high.

Limitations

The motor is running (`q_xFwd = 1` or `q_xRev = 1`), and feedback signal is active (`i_xFbckRun = 1`). Now if interlock occurs (`i_xLock = 1`) for a duration more than `iFbckDly` seconds specified at the input `i_strPara`, an error is detected.

Chapter 6

MOT1D1S: Motor, One Direction, One Speed

Overview

This chapter describes the MOT1D1S function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	46
Input and Output Pin Description	48
Structures Used	50
Control Word Bits Description	51
Status Word Bits Description	52

Functional Description

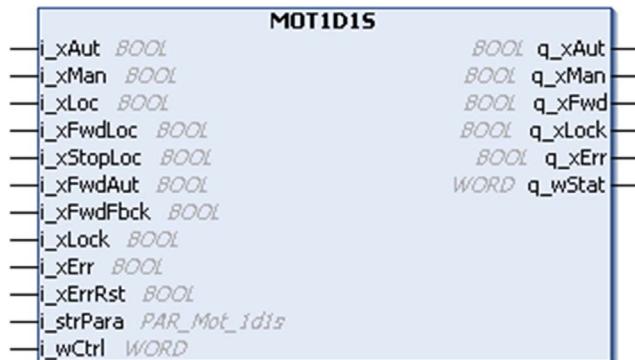
Function Block Description

The MOT1D1S generic function block is dedicated to the control and command of motors with one direction of rotation and one speed through any actuator connected in parallel through digital I/Os or through communication networks like Modbus Serial Line or CANopen.

The motor can be controlled through 3 different operating modes : automatic, manual using push-buttons, or manual using HMI.

Pin Diagram

This figure shows the pin diagram of the MOT1D1S function block:



Limitations

If both forward run and stop command are given at the same time then stop has higher priority than forward run.

Operation Modes

The MOT1D1S function block supports three modes of operation:

- **Automatic Mode:** The automatic mode is selected through the input pin *i_xAut*. In automatic mode, the motor is started and stopped through the input pin *i_xFwdAut*, regardless of the activation of local mode. The output *q_xFwd* remains high, as long as the input *i_xFwdAut* remains high, and there is no detected error or interlock. The *i_xFwdAut* input is controlled by the controller process application during normal functioning.

- **Manual Mode:** The manual mode is activated by the pin `i_xMan`.
Case 1: Local mode is not activated. The motor is started and stopped through the bit commands of the signal `i_dwCtrl`. This double word can be associated to an external HMI equipped with a keypad.
Case 2: Local mode is activated through the input pin `i_xLoc`. The motor is started via the rising edge of input `i_xFwdMan` and stopped via the rising edge of `i_xManStop`. These digital inputs can be linked to connect push-buttons.
- **Local Mode:** The local mode is activated by an input pin `i_xLoc`, and is set additionally to the automatic or manual mode. The local mode does not influence the automatic mode, but changes the source for manual operation.

The block is de-activated on controller start and remains in the same operation mode, unless a new one is selected. If both automatic and manual modes are selected simultaneously (inputs `i_xAut` and `i_xMan` are set to 1), the operation mode is invalid which is indicated at the `q_xErr` output.

CAUTION

UNINTENDED MOTOR STOP

Do not change from manual mode to automatic mode when the motor is being used, doing so will make the motor stop.

Failure to follow these instructions can result in injury or equipment damage.

Supervising the Motor

The operation of the motor is supervised by feedback signal at pin `i_xFwdFbck`, which indicates the running motor. The feedback signal must adjust its value to the value of the related output `q_xFwd` within a defined time. If the time is exceeded, the block indicates a detected error. The time can be set via the structure element `tFbTime` at input `i_strPara`. The feedback supervision can be switched off by the structure element `xEnFb` at the input `i_strPara`.

Running the Motor

The motor can run, only if the interlock input `i_xLock` is set to 0. An active interlock signal inhibits the start/stop of the motor. When the interlock signal returns to 0 the motor is restarted. An active interlock is indicated at the output `q_xLock`. The motor can run only if the output `q_xErr` is set to 0.

An active detected error signal inhibits the start or stop of the motor. The function block sets the detected error signal, if the detected error input `i_xErr` is set to 1 (external detected error) or in case of an invalid operation mode or a missing feedback signal (internal detected error).

Resetting a Detected error

The detected errors are indicated in the HMI as alarms. To reset the detected error output the detected error has to be acknowledged by a rising edge on the input `i_xAckn` or by using the acknowledge bit of the input `i_wCtrl`.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of MOT1D1S function block:

Input	Data Type	Description
i_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
i_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
i_xLoc	BOOL	TRUE: Local mode enabled FALSE: Disabled
i_xFwdLoc	BOOL	Rising edge trigger from 0 to 1 runs the motor in forward direction. Manual and Local Mode to be set to 1 simultaneously. (Optional)
i_xStopLoc	BOOL	Rising edge trigger from 0 to 1 stops the motor in forward direction. Manual and Local Mode to be set to 1 simultaneously.
i_xFwdAut	BOOL	TRUE: Forward command enabled FALSE: Disabled (Optional)
i_xFwdFbck	BOOL	TRUE: Feedback is active FALSE: No feedback
i_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock
i_xErr	BOOL	TRUE: External detected error is active FALSE: No external detected error
i_xErrRst	BOOL	Acknowledgement is done with a rising edge. Input to acknowledge internal and external detected errors indicated at the output q_xErr.
i_strPara	STRUCT Par_Mot_1d1s	Structure with parameters for this block Refer to structures used (see page 50)
i_wCtrl	WORD	Command bits to interact from the HMI Range: 0...65535 Refer to control word bits description (see page 51)

Output Pin Description

This table describes the output pins of MOT1D1S function block:

Output	Data Type	Description
q_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
q_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
q_xFwd	BOOL	TRUE: Forward command active FALSE: Disabled
q_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock Indicates that the operation is blocked by an interlock (input i_xLock)
q_xErr	BOOL	TRUE: Detected error is active FALSE: No detected error
q_wStat	WORD	Status bits to be displayed in the HMI Range: 0..65535 Refer to status word bits description (see page 52).

Structures Used

Par_Mot_1d1s

Structure Element	Type	Description
xEnFb	BOOL	Enable feedback signal supervision
tFbTime	TIME	Delay time in seconds for the feedback signals

Control Word Bits Description

Functionality

This table describes the command word bits that can be managed by an HMI keypad:

Bit Position	Rising edge of this bit gives:
0, 1	Not used
2	Rising edge of this bit gives RUN command
3, 4	Not used
5	Rising edge of this bit gives STOP command
6...14	Not used
15	Rising edge of this bit gives acknowledgement to detected error

Status Word Bits Description

Functionality

This table describes the status word bits that can be managed by an HMI display:

Bit Position	Description
0	Auto mode is active
1	Manual mode is active
2	RUN command to motor
3	Local mode is active
4	Not used
5	Function block is locked by interlock input
6, 7	Not used
8	Detected error
9	Internally detected error
10	Externally detected error
11	Detected feedback error (missing feedback)
12...15	Not used

Chapter 7

MOT1D2S: Motor, One Direction, Two Speed

Overview

This chapter describes the MOT1D2S function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	54
Input and Output Pin Description	57
Structures Used	59
Control Double Word Bits Description	60
Status Double Word Bits Description	61

Functional Description

Function Block Description

The MOT1D2S generic function block is dedicated to the control and command of motors with one direction of rotation and two speeds through any actuator connected in parallel through digital I/Os or through communication networks like Modbus Serial Line or CANopen.

The motor can be controlled through 3 different operating modes : automatic, manual using push-buttons, or manual using HMI.

Pin Diagram

This figure shows the pin diagram of the MOT1D2S function block:



Operation Modes

The MOT1D2S function block supports three modes of operation:

- **Automatic Mode:** The automatic mode is selected through the input pin *i_xAut*. In automatic mode, the motor is started and stopped through the input pin *i_xFwdAut*, regardless of the activation of local mode. The output *q_xFwd* remains high, as long as the input *i_xFwdAut* remains high, and there is no detected error or interlock. The *i_xAutFast* input allows to activate fast speed. These inputs are controlled by the controller process application during normal functioning.

- **Manual Mode:** The manual mode is activated by the pin `i_xMan`.
Case 1: Local mode is not activated. The motor is started and stopped through the bit commands of the signal `i_dwCtrl`. This double word can be associated to an external HMI equipped with a keypad.
Case 2: Local mode is activated through the input pin `i_xLoc`. The motor is started and stopped through the input signals `i_xFwdLoc` and `i_xLocStop`. These digital inputs can be linked to connect push-buttons.
- **Local Mode:** The local mode is activated by an input pin `i_xLoc`, and is set additionally to the automatic or manual mode. The local mode does not influence the automatic mode, but changes the source for manual operation.

The block is de-activated on controller start and remains in the same operation mode, unless a new one is selected. If both automatic and manual modes are selected simultaneously (inputs `i_xAut` and `i_xMan` are set to 1), the operation mode is invalid which is indicated at the `q_xErr` output.

CAUTION

UNINTENDED MOTOR STOP

Do not change from manual mode to automatic mode when the motor is being used, doing so will make the motor stop.

Failure to follow these instructions can result in injury or equipment damage.

Enabling Fast Speed

To enable the fast speed in the automatic mode, the input `i_xAutFast` should be set to 1. The bit command of the signal `i_dwCtrl` must be set in the manual mode or `i_xLocFast` must be set, if the control is switched to local mode.

Supervising the Motor

The operation of the motor is supervised by feedback signal (`i_xFwdFbck`), which indicates the running motor. The feedback signal must adjust its value to the value of the related output `q_xFwd` within a defined time. If the time is exceeded, the block indicates a detected error (Missing Feedback detected error). The time can be set via the structure element `iFbckDly` at input `i_strPara`. The supervision can be switched off by the structure element `xEnFb` at the input `i_strPara`.

Running the Motor

The motor can run, only if the interlock input `i_xLock` is set to 0. An active interlock signal inhibits the start/stop of the motor. When the interlock signal returns to 0 the motor is restarted. An active interlock is indicated at the output `q_xLock`.

The motor can run only if the output `q_xErr` is set to 0. An active detected error signal inhibits the start or stop of the motor. The function block sets the detected error signal, if the detected error input `i_xErr` is set to 1 (external detected error) or in case of an invalid operation mode or a missing feedback signal (internal detected errors).

Resetting a Detected Error

The single detected errors are indicated in the HMI as alarms. To reset the detected error output, the detected error has to be acknowledged by a rising edge on the input `i_xAckn` or by using the 16th bit of the signal `i_dwCtrl`.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of the MOT1D2S function block:

Input	Data Type	Description
i_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
i_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
i_xLoc	BOOL	TRUE: Local mode enabled FALSE: Disabled
i_xFwdLoc	BOOL	Rising edge trigger from 0 to 1 runs the motor in forward direction. Manual and Local Mode to be set to 1 simultaneously.
i_xStopLoc	BOOL	Rising edge trigger from 0 to 1 stops the motor in forward direction. Manual and Local Mode to be set to 1 simultaneously.
i_xLocFast	BOOL	TRUE: Fast command is enabled in local mode FALSE: Disabled (Optional)
i_xFwdAut	BOOL	TRUE: Forward command enabled in auto mode FALSE: Disabled (Optional)
i_xAutFast	BOOL	TRUE: Fast command enabled in auto mode FALSE: Disabled (Optional)
i_xFwdFbck	BOOL	TRUE: Feedback is active. FALSE: No feedback
i_xLock	BOOL	TRUE: Interlock is active. FALSE: No interlock
i_xErr	BOOL	TRUE: External detected error is active. FALSE: No external detected error.
i_xErrRst	BOOL	Acknowledgement is done with a rising edge. Input to acknowledge internal and external detected errors indicated at the output q_xErr.
i_strPara	STRUCT Par_Mot_1d2s	Structure with parameters for this block. Refer to structures used (see page 59).
i_dwCtrl	DWORD	Command bits to interact from the HMI Refer to command double word description (see page 60).

Output Pin Description

This table describes the output pins of the MOT1D2S function block:

Output	Data Type	Description
q_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
q_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
q_xFwd	BOOL	TRUE: Forward command active FALSE: Disabled
q_xFast	BOOL	TRUE: Fast command active FALSE: Disabled
q_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock Indicates that the operation is blocked by an interlock (input i_xLock).
q_xErr	BOOL	TRUE: Detected error is active FALSE: No detected error
q_dwStat	DWORD	Status bits to be displayed in the HMI. Refer to status double word bit description (see page 61).

Structures Used

Par_Mot_1d2s

Structure Element	Type	Description
xEnFb	BOOL	Enable feedback signal supervision
iFbckDly	INT	Delay time in seconds to get the feedback signals from motor.

Control Double Word Bits Description

Functionality

This table describes the control double word bits that can be managed by an HMI keypad:

Bit Position	Description
0...7	Not used
8	Runs the motor forward in manual mode (rising-edge triggered)
9	Not used
10	Set to TRUE, enables manually, fast velocity for motor in manual mode
11	Not used
12	Stops manually motor (rising-edge triggered) in manual mode
13...15	Not used
16	Acknowledges internal and external detected errors indicated at the output <code>q_xErr</code> (rising-edge triggered)
17...31	Not used

Status Double Word Bits Description

Functionality

This table describes the status double word bits that can be managed by an HMI display:

Bit Position	Description
0	Auto mode is active
1	Manual mode is active
2	Sets the block to local mode
3	Indicates that the operation is blocked by an interlock (input <i>i_xLock</i>)
4...7	Not used
8	Signal to run the motor
9	Not used
10	Signal to run the motor with faster speed
11...13	Not used
14	If <i>i_strPara.xFbckEn</i> = 0, Feedback = <i>q_xFwd</i> . If <i>i_strPara.xFbckEn</i> = 1, Feedback = <i>i_xFwdFbck</i> .
15	Not used
16	Acknowledges internal and external detected errors indicated at the output <i>q_xErr</i>
17	Indicates that the operation is blocked by an internal or external (Input <i>i_xErr</i>) detected error, which is not acknowledged
18	Not used
19	Enable feedback signal supervision
20...23	Not used
24	Invalid operating mode detected error
25	External detected error
26	Missing feedback detected error
27...31	Not used

Chapter 8

MOT2D1S: Motor, Two Direction, One Speed

Overview

This chapter describes the MOT2D1S function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	64
Input and Output Pin Description	67
Structures Used	69
Control Double Word Bits Description	70
Status Double Word Bits Description	71

Functional Description

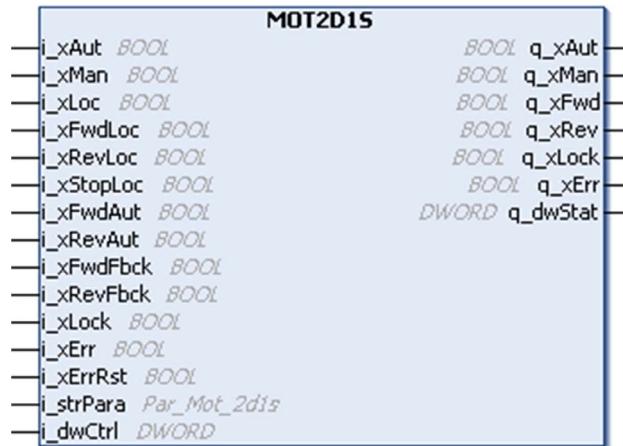
Function Block Description

The MOT2D1S generic function block is dedicated to the control and command of motors with two directions of rotation and one speed through any actuator connected in parallel through digital I/Os or through communication networks like Modbus Serial Line or CANopen.

The motor can be controlled through 3 different operating modes : automatic, manual using push-buttons, or manual using HMI.

Pin Diagram

This figure shows the pin diagram of the MOT2D1S function block:



Operation Modes

The MOT2D1S function block has three modes of operation:

- **Automatic Mode:** The Automatic mode is selected through the input `i_xAut`. In Automatic mode, the motor is started and stopped in the forward direction through the input pin `i_xFwdAut`, regardless of the activation of local mode. Similarly, the reverse direction is handled with the input `i_xRevAut`. These inputs are controlled by the controller process application during normal functioning.

- **Manual Mode:** The Manual mode is activated by the pin `i_xMan`.
Case 1: Local mode is not activated. The motor is started and stopped through the bit commands of the signal `i_dwCtrl`. This double word can be associated to an external HMI equipped with a keypad.
Case 2: Local mode is activated through the input pin `i_xLoc`. The motor is started through the inputs `i_xFwdLoc` and stopped via `i_xLocStop` in forward direction, and the inputs `i_xRevLoc` and `i_xLocStop` are used to manually control the motor in reverse direction. These digital inputs can be linked to connect push-buttons.
- **Local Mode:** The local mode is activated by an input pin `i_xLoc` and is set additionally to the automatic or manual mode. The local mode does not influence the automatic mode, but changes the source for manual operation.

The block is de-activated on controller start and remains in the same operation mode, unless a new one is selected. If both automatic and manual modes are selected simultaneously (inputs `i_xAut` and `i_xMan` are set to 1), the operation mode is invalid, which is indicated at the `q_xErr` output.

NOTE: If the operation mode is changed from manual mode to automatic mode, a running motor is turned off. Any other change of the operation mode does not affect the motor operation.

Starting in Reverse Direction

The motor cannot be started in reverse direction, when already running in forward (output `q_xFwd` set to 1) and vice versa. This direct change between the two directions can be avoided by proper time supervision. The delay time for the change of direction can be freely chosen through the structure element `iRevDly` at the input `i_strPara`. If the time is set to 0, the change takes place immediately.

Supervising the Motor

The operation of the motor is supervised by feedback signal (`i_xFwdFbck`), which indicates the running motor. The feedback signal must adjust its value to the value of the related output `q_xFwd` within a defined time. If the time is exceeded, the block indicates a detected error (Missing Feedback detected error). The time can be set via the structure element `iFbckDly` at input `i_strPara`. The supervision can be switched off by the structure element `xEnFb` at the input `i_strPara`.

Running the Motor

The motor can run, only if the interlock input `i_xLock` is set to 0. An active interlock signal inhibits the start/stop of the motor. When the interlock signal returns to 0 the motor is restarted. An active interlock is indicated at the output `q_xLock`.

The motor can run only if the output `q_xErr` is set to 0. An active detected error signal inhibits the start or stop of the motor. The function block sets the detected error signal, if the detected error input `i_xErr` is set to 1 (external detected error) or in case of an invalid operation mode or a missing feedback signal (internal detected errors).

Resetting a Detected Error

The single detected errors are indicated in the HMI as alarms. To reset the detected error output, the detected error has to be acknowledged by a rising edge on the input `i_xAckn` or by using the 16th bit of the signal `i_dwCtrl`.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of the MOT2D1S function block:

Input	Data Type	Description
i_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
i_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
i_xLoc	BOOL	TRUE: Local mode enabled FALSE: Disabled
i_xFwdLoc	BOOL	Rising edge trigger from 0 to 1 runs the motor in forward direction. Manual and Local Mode to be set to 1 simultaneously.
i_xRevLoc	BOOL	Rising edge trigger from 0 to 1 runs the motor in reverse direction. Manual and Local Mode to be set to 1 simultaneously.
i_xStopLoc	BOOL	Rising edge trigger from 0 to 1 stops the motor. Manual and Local Mode to be set to 1 simultaneously.
i_xFwdAut	BOOL	TRUE: Forward command enabled FALSE: Disabled
i_xRevAut	BOOL	TRUE: Reverse command enabled FALSE: Disabled
i_xFwdFbck	BOOL	TRUE: Forward Feedback is active. FALSE: Disabled
i_xRevFbck	BOOL	TRUE: Reverse Feedback is active. FALSE: Disabled
i_xLock	BOOL	TRUE: Interlock is active. FALSE: No interlock
i_xErr	BOOL	TRUE: External detected error is active. FALSE: No external detected error
i_xErrRst	BOOL	Acknowledgement is done with a rising edge. Input to acknowledge internal and external detected errors indicated at the output q_xErr.
i_strPara	STRUCT Par_Mot_2d1s	Structure with parameters for this block. Refer to used structures (see page 69).
i_dwCtrl	DWORD	Command bits to interact from the HMI Refer to control double word description (see page 70).

Output Pin Description

This table describes the output pins of the MOT2D1S function block:

Output	Data Type	Description
q_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
q_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
q_xFwd	BOOL	TRUE: Forward command active FALSE: Disabled
q_xRev	BOOL	TRUE: Reverse command active FALSE: Disabled
q_xLock	BOOL	TRUE: Interlock is active. FALSE: No interlock Indicates that the operation is blocked by an interlock (input i_xLock).
q_xErr	BOOL	TRUE: Detected error is active. FALSE: No detected error
q_dwStat	DWORD	Status bits to be displayed in the HMI Refer to status double word bits description (see page 71).

Structures Used

Par_Mot_2d1s

Structure Element	Type	Description
xEnFb	BOOL	Enable feedback signal supervision
iFbckDly	INT	Delay time in seconds to get the feedback signals from motor.
iRevDly	INT	Delay time in seconds, to reverse the direction of the running motor.

Control Double Word Bits Description

Functionality

This table describes the control double word bits that can be managed by an HMI keypad:

Bit Position	Description
0...7	Not used
8	Runs the motor forward in manual mode (rising-edge triggered).
9	Runs the motor reverse in manual mode (rising edge triggered).
10, 11	Not used
12	Stops manually motor (rising-edge triggered) in manual mode.
13...15	Not used
16	Acknowledges internal and external detected errors indicated at the output <code>q_xErr</code> (rising-edge triggered).
17...31	Not used

Status Double Word Bits Description

Functionality

This table describes the status double word bits that can be managed by an HMI display:

Bit Position	Description
0	Auto mode is active.
1	Manual mode is active.
2	Sets the block to local mode.
3	Indicates that the operation is blocked by an interlock.
4...7	Not used
8	Signal to run the motor in forward direction.
9	Signal to run the motor in reverse direction.
10...13	Not used
14	If <i>i_strPara.xFbckEn</i> = 0, Feedback = <i>q_xFwd</i> . If <i>i_strPara.xFbckEn</i> = 1, Feedback = <i>i_xFwdFbck</i> .
15	If <i>i_strPara.xFbckEn</i> = 0, Feedback = <i>q_xRev</i> Else Feedback = <i>i_xRevFbck</i> .
16	Resets the detected error indicated at the output <i>q_xErr</i> .
17	Indicates that the operation is blocked by an internal or external (Input <i>i_xErr</i>) detected error, which is not acknowledged.
19	Enable feedback signal supervision.
20...23	Not used
24	Invalid operating mode detected error.
25	External detected error.
26	Missing feedback detected error.
27...31	Not used

Chapter 9

MOT2D2S: Motor, Two Direction, Two Speed

Overview

This chapter describes the MOT2D2S function block.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Description	74
Input and Output Pin Description	77
Structures Used	79
Control Word Bits Description	80
Status Word Bits Description	81
Instantiation and Usage Example	82

Functional Description

Function Block Description

The MOT2D2S generic function block is dedicated to the control and command of motors with two directions of rotation and two speeds through any actuator connected in parallel through digital I/Os or through communication networks like Modbus Serial Line or CANopen.

The motor can be controlled through 3 different operating modes : automatic, manual using push-buttons, or manual using HMI.

Pin Diagram

This figure shows the pin diagram of the MOT2D2S function block:



Limitations

- There is no provision for giving speed reference value from inside the block. It has to be set from outside the block.
- If both forward and reverse run command are given at the same time then none of them will take effect on the output side, i.e there will be no run order in any direction.
- If both forward run and stop command are given at the same time then stop has higher priority than forward run.
- If both reverse run and stop command are given at the same time then stop has higher priority than reverse run.

Operation Modes

The MOT2D2S function block has three modes of operation:

- **Automatic Mode:** The automatic mode is selected through the input pin `i_xAut`. In automatic mode, the motor is started and stopped in the forward direction through the input pin `i_xFwdAut` regardless of the activation of local mode. Similarly, the reverse direction is handled with the input pin `i_xRevAut`. These inputs are controlled by the controller process application during normal functioning.
- **Manual Mode:** The Manual mode is activated by the pin `i_xMan`.
 - Case 1: Local mode is not activated. The motor is started and stopped through the bit commands of the signal `i_wCtrl`. This double word can be associated to an external HMI equipped with a keypad.
 - Case 2: Local mode is activated through the input pin `i_xLoc`. The motor is started through the inputs `i_xFwdLoc` and stopped via `i_xLocStop` in forward direction, and the inputs `i_xRevLoc` and `i_xLocStop` are used to manually control the motor in reverse direction. While the motor is running in one direction, it cannot be run in opposite direction until its stopped first by `i_xLocStop`. These digital inputs can be linked to connect push-buttons.
- **Local Mode:** The Local mode is activated by an input pin `i_xLoc` and is set additionally to the automatic or manual mode. The local mode does not influence the automatic mode, but changes the source for manual operation.

The block is de-activated on controller start and remains in the same operation mode, unless a new one is selected. If both automatic and manual modes are selected simultaneously (inputs `i_xAut` and `i_xMan` are set to 1), the operation mode is invalid, which is indicated at the `q_xErr` output.

CAUTION

UNINTENDED MOTOR STOP

Do not change from manual mode to automatic mode when the motor is being used, doing so will make the motor stop.

Failure to follow these instructions can result in injury or equipment damage.

Starting in Reverse Direction

The motor cannot be started in reverse direction, when already running in forward (output `q_xFwd` set to 1) and vice versa. This direct change between the two directions can be avoided by proper time supervision. The delay time for the change of direction can be freely chosen through the structure element `tRevTime` at the input `i_strPara`. If the time is set to 0 the change takes place immediately.

Enabling Fast Speed

To enable the fast speed the input `i_xAutFast` has to be set to 1 in automatic mode, the 4th bit from Control word `i_wCtrl` has to be set in manual mode or `i_xLocFast` set to 1, if the control is switched to local mode.

Supervising the Motor

The operation of the motor is supervised by feedback signal (*i_xFwdFbck*), which indicates the running motor. The feedback signal must adjust its value to the value of the related output *q_xFwd* within a defined time. If the time is exceeded, the block indicates a detected error (Missing Feedback detected error). The time can be set via the structure element *iFbckDly* at input *i_strPara*. The supervision can be switched off by the structure element *xEnFb* at the input *i_strPara*.

Running the Motor

The motor can run, only if the interlock input *i_xLock* is set to 0. An active interlock signal inhibits the start/stop of the motor. When the interlock signal returns to 0 the motor is restarted. An active interlock is indicated at the output *q_xLock*.

The motor can run only if the output *q_xErr* is set to 0. An active detected error signal inhibits the start or stop of the motor. The function block sets the detected error signal, if the detected error input *i_xErr* is set to 1 (external detected error) or in case of an invalid operation mode or a missing feedback signal (internal detected errors).

Resetting a Detected error

The single detected errors are indicated in the HMI as alarms. To reset the detected error output, the detected error has to be acknowledged by a rising edge on the input *i_xAckn* or by using the 16th bit of the signal *i_dwCtrl*.

Input and Output Pin Description

Input Pin Description

This table describes the input pins of the MOT2D2S function block:

Input	Data Type	Description
i_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
i_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
i_xLoc	BOOL	TRUE: Local mode enabled FALSE: Disabled
i_xFwdLoc	BOOL	Rising edge from 0 to 1 manually starts motor in forward direction in local mode. Manual and Local modes to be set to 1 simultaneously.
i_xRevLoc	BOOL	Rising edge from 0 to 1 manually starts motor in reverse direction in local mode. Manual and Local modes to be set to 1 simultaneously.
i_xStopLoc	BOOL	Rising edge trigger from 0 to 1 stops the motor. Manual and Local modes to be set to 1 simultaneously.
i_xLocFast	BOOL	Enables manually fast velocity for motor in local mode. Manual and Local modes to be set to 1 simultaneously. (Optional)
i_xFwdAut	BOOL	TRUE: Forward command enabled FALSE: Disabled
i_xRevAut	BOOL	TRUE: Reverse command enabled FALSE: Disabled
i_xAutFast	BOOL	Motor runs with fast velocity, if the input is set to 1 and the automatic mode is activated. (Optional)
i_xFwdFbck	BOOL	TRUE: Feedback is active FALSE: No feedback
i_xRevFbck	BOOL	TRUE: Reverse Feedback is active FALSE: No feedback (Optional)
i_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock

Input	Data Type	Description
i_xErr	BOOL	TRUE: External detected error is active FALSE: No external detected error
i_xErrRst	BOOL	Acknowledgement is done with a rising edge. Input to acknowledge internal and external detected errors indicated at the output q_xErr.
i_strPara	STRUCT Par_Mot_2d2s	Structure with parameters for this block. Refer to used structures (see page 79).
i_wCtrl	WORD	Command bits to interact from the HMI Range: 0...65535 Refer to control word bits description (see page 80).

Output Pin Description

This table describes the output pins of the MOT2D2S function block:

Output	Data Type	Description
q_xAut	BOOL	TRUE: Automatic mode enabled FALSE: Disabled
q_xMan	BOOL	TRUE: Manual mode enabled FALSE: Disabled
q_xFwd	BOOL	TRUE: Forward command active FALSE: Disabled
q_xRev	BOOL	TRUE: Reverse command active FALSE: Disabled
q_xFast	BOOL	Signal to set the motor to fast speed. It is set, if the appropriate input signals and one output q_xFwd or q_xRev are set.
q_xLock	BOOL	TRUE: Interlock is active FALSE: No interlock Indicates that the operation is blocked by an interlock (input i_xLock).
q_xErr	BOOL	TRUE: Detected error is active FALSE: No detected error
q_wStat	WORD	Status bits to be displayed in the HMI Range: 0...65535 Refer to status word bits description (see page 81).

Structures Used

Par_Mot_2d2s

Structure Element	Type	Description
xEnFb	BOOL	Enable feedback signal supervision
tTimeFbck	TIME	Delay time in seconds for the feedback signals
tRevTime	TIME	Delay time in seconds, to reverse the direction of the running motor.

Control Word Bits Description

Functionality

This table describes the control word bits that can be managed by an HMI keypad:

Bit Position	Description
0, 1	Not used
2	Runs the motor forward in manual mode (rising edge triggered)
3	Runs the motor reverse in manual mode (rising edge triggered)
4	Enables manually fast velocity for motor in manual mode
5	Stops manually motor (rising edge triggered) in manual mode
6..14	Not used
15	Acknowledges internal and external detected errors indicated at the output <code>q_xErr</code> (rising edge triggered)

Status Word Bits Description

Functionality

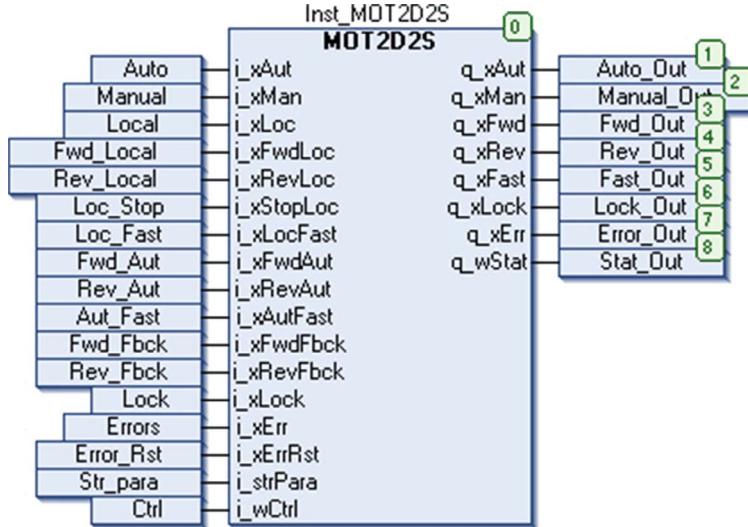
This table describes the status word bits that can be managed by an HMI display:

Bit Position	Description
0	Automatic mode is activated
1	Manual mode is active
2	Signal to run the motor in forward direction
3	Signal to run the motor in reverse direction
4	Enables fast velocity for motor
5	Indicates that the operation is blocked by an interlock
6	Sets the block to local mode
7	Both forward and reverse commands are activated simultaneously
8	Indicates that the operation is blocked by an internal or external (Input <i>i_xErr</i>) detected error, which is not acknowledged
9	Invalid operating mode detected error
10	External detected error
11	Missing feedback detected error
12...15	Not used

Instantiation and Usage Example

Instantiation and Usage Example

This figure shows an instance of the MOT2D2S function block:



As shown by output pin `q_xAut`, the block is operating in Auto mode. A reverse fast run command is issued at the input pins `i_xRevAut` and `i_xAutFast` which is shown at the output side on pins `q_xRev` and `q_xFast`.

Glossary



0-9

%

According to the IEC standard, % is a prefix that identifies internal memory addresses in the logic controller to store the value of program variables, constants, I/O, and so on.

A

application

A program including configuration data, symbols, and documentation.

C

CANopen

An open industry-standard communication protocol and device profile specification (EN 50325-4).

configuration

The arrangement and interconnection of hardware components within a system and the hardware and software parameters that determine the operating characteristics of the system.

controller

Automates industrial processes (also known as programmable logic controller or programmable controller).

D

digital I/O

(*digital input/output*) An individual circuit connection at the electronic module that corresponds directly to a data table bit. The data table bit holds the value of the signal at the I/O circuit. It gives the control logic digital access to I/O values.

E

EDS

(*electronic data sheet*) A file for fieldbus device description that contains, for example, the properties of a device such as parameters and settings.

element

The short name of the ARRAY element.

F

FLA

(*full load amps*) A motor characteristic that you can find on the motor plate. It represents the current that the motor draws at the rated voltage and rated load. This characteristic is named as the motor full load current.

FLC

(*full load current*) A motor characteristic that you can find on the motor plate. It represents the current that the motor draws at the rated voltage and rated load. This characteristic is named as the motor full load current.

function

A programming unit that has 1 input and returns 1 immediate result. However, unlike FBs, it is directly called with its name (as opposed to through an instance), has no persistent state from one call to the next and can be used as an operand in other programming expressions.

Examples: boolean (AND) operators, calculations, conversions (BYTE_TO_INT)

H

HMI

(*human machine interface*) An operator interface (usually graphical) for human control over industrial equipment.

I

I/O

(*input/output*)

M

Modbus

The protocol that allows communications between many devices connected to the same network.

N

network

A system of interconnected devices that share a common data path and protocol for communications.

node

An addressable device on a communication network.

P

PDO

(process data object) An unconfirmed broadcast message or sent from a producer device to a consumer device in a CAN-based network. The transmit PDO from the producer device has a specific identifier that corresponds to the receive PDO of the consumer devices.

R

run

A command that causes the controller to scan the application program, read the physical inputs, and write to the physical outputs according to solution of the logic of the program.

S

STOP

A command that causes the controller to stop running an application program.

T

terminal block

(terminal block) The component that mounts in an electronic module and provides electrical connections between the controller and the field devices.

V

variable

A memory unit that is addressed and modified by a program.

W

WORD

A type encoded in a 16-bit format.

Glossary



M

MOT1D1S, 45
MOT1D2S, 53
MOT2D1S, 63
MOT2D2S, 73

N

network configuration for TeSys, 15

T

TeSys
function blocks offer overview, 11
MOT1D1S, 45
MOT1D2S, 53
MOT2D1S, 63
MOT2D2S, 73
network configuration, 15
TeSysT_CtrlCmdCyc_CANopen, 29
TeSysU_CtrlCmdCyc_CANopen, 23
TeSysU_IO, 35
TeSys function blocks offer overview, 11
TeSysT_CtrlCmdCyc_CANopen, 29
TeSysU_CtrlCmdCyc_CANopen, 23
TeSysU_IO, 35

