SoMachine

Conveying Application Functions
Conveying Library Guide

11/2016

EIO0000000201.07

www.schneider-electric.com

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.

© 2016 Schneider Electric. All Rights Reserved.

Table of Contents

	Safety Information	9
	About the Book	13
Part I	Conveying Library System	15
Chapter 1	System Requirements	17
	Conveying System Requirements	18
	Technical Specifications and Limitations	20
	Function Block Location in the Conveying Library	21
Chapter 2	General Presentation of Application Function Blocks	23
	General Description	23
Chapter 3	Matrix of Functions by Type of Equipments	31
	Functional Overview	31
Chapter 4	Architectures	33
4.1	Equipment Application Function Blocks	34
	Overview of Equipment AFBs	34
Chapter 5	Operating Mode	37
	Operating Mode for Application Function Blocks	38
Chapter 6	Alarm Handling	43
6.1	Alarm Messages	44
	Alarm Conditions	45
	Alarm Words of Conveyor, Input, Output	46
	Alarm Words of Deskzone Complex	48
	Alarm Words of Deskzone Simple	49
	Alarm Words of ERC	50
	Alarm Words of TurnTable	53
	Alarm Word of Tracking Complex	56
	Alarm Word of Tracking Simple	57
	Alarm Words of Alarm Handling	58
	Alarm Word of AlrmHdlgTextFile	59
6.2	Alert Messages	60
	Alert Words of Conveying, Input, Output	61
	Alert Word of Deskzone (Complex, Simple)	63
	Alert Word of ERC	64
	Alert Word of TurnTable	66

Alert Word of Tracking Complex 68
Alert Word of Tracking Simple 69
Alert Word of Alarm Handling 70
Alert Word of AlrmHdlgTextFile 71
Chapter 7 Handshaking - Information Dataflow Between Conveying Equipments 73
Functional Overview. 74
Timing Diagram 78
Chapter 8 Predefined Equipment Types 85
Predefined Equipment Types 85
Chapter 9 Use Cases 87
Simple Conveying Line 88
Complex Conveying Line 92
Part II Conveyors 95
Chapter 10 Conveyor: Transfer of Goods 97
10.1 Functional and System Overview 98
Functional Overview. 99
System Overview 100
10.2 Function Block Description. 101
`Conveyor` Function Block 101
10.3 Pin Description 102
Input Pin Description 103
Output Pin Description 106
Input/Output Pin Description 110
10.4 Interfaces 111
Conveyor Interfaces 112
Linking the Function Blocks 113
Chapter 11 Input: Loading of Goods 119
11.1 Functional and System Overview. 120
Functional Overview. 121
System Overview 122
11.2 Function Block Description. 123
`Input` Function Block 123
11.3 Pin Description 124
Input Pin Description 125
Output Pin Description 128
Input/Output Pin Description 132

11.4 Interfaces 133
Input Interfaces 134
Linking the Function Blocks 135
Chapter 12 Output: Exit of Goods 141
12.1 Functional and System Overview 142
Functional Overview 143
System Overview 144
12.2 Function Block Description 145
`Output` Function Block 145
12.3 Pin Description 146
Input Pin Description 147
Output Pin Description 150
Input/Output Pin Description 154
12.4 Interfaces 155
Output Interfaces 156
Linking the Function Blocks 157
Part III Deskzone 163
Chapter 13 DeskzoneCmpx: Interface to HMI 165
13.1 Functional Overview 166
Functional Overview 166
13.2 Function Block Description 167
`DeskzoneCmpx` Function Block 167
13.3 Pin Description 168
Input Pin Description 169
Output Pin Description 170
Input/Output Pin Description 171
13.4 Interfaces 174
DeskzoneCmpx Interfaces 175
Linking the Function Blocks 176
Chapter 14 DeskzoneSimp: Interface to HMI 179
14.1 Functional Overview 180
Functional Overview 180
14.2 Function Block Description 181
`DeskzoneSimp` Function Block 181
14.3 Pin Description 182
Input Pin Description 183
Output Pin Description 184
Input/Output Pin Description 185

14.4 Interfaces 187
DeskzoneSimp Interfaces 188
Linking the Function Blocks 189
Part IV ERC 191
Chapter 15 ERC: Transfer of Goods in 4 Directions 193
15.1 Functional and System Overview. 194
Functional Overview. 195
System Overview 196
15.2 Function Block Description. 197
ERC Function Block 197
15.3 Pin Description 198
Input Pin Description 199
Output Pin Description 202
Input/Output Pin Description 206
15.4 Interfaces 207
ERC Interfaces. 208
Linking the Function Blocks 209
Part V Turntable 215
Chapter 16 TurnTable: Transfer of Goods in 4 Directions 217
16.1 Functional and System Overview 218
Functional Overview. 219
System Overview 220
16.2 Function Block Description. 221
TurnTable Function Block 221
16.3 Pin Description 222
Input Pin Description 223
Output Pin Description 226
Input/Output Pin Description 230
16.4 Interfaces 231
TurnTable Interfaces 232
Linking the Function Blocks 233
Part VI Tracking 239
Chapter 17 TrckCmpx: Handling of Tracking Data for TurnTable and Transfer Table 241
17.1 Functional Overview 242
Functional Overview. 242
17.2 Function Block Description. 243
TrckCmpx Function Block 243

17.3 Pin Description 244
Input Pin Description 245
Output Pin Description. 249
Input/Output Pin Description 251
17.4 Interfaces. 253
TrckCmpx Interfaces 254
Linking the Function Blocks. 255
Chapter 18 TrckSimp: Handling of Tracking Data for Conveyor 261
18.1 Functional Overview 262
Functional Overview 262
18.2 Function Block Description 263
`TrckSimp` Function Block. 263
18.3 Pin Description 264
Input Pin Description 265
Output Pin Description. 267
Input/Output Pin Description 269
18.4 Interfaces. 271
TrckSimp Interfaces. 272
Linking the Function Blocks. 273
Part VII Alarm Handling 279
Chapter 19 AlrmHdlg: Managing Alarms and Alerts 281
19.1 Functional Overview 282
Functional Overview 282
19.2 Function Block Description 283
`AlrmHdlg` Function Block. 283
19.3 Pin Description 284
Input Pin Description 285
Output Pin Description. 287
19.4 Interfaces. 291
AlrmHdlg Interfaces 292
Linking the Function Blocks. 293
19.5 Alarmtext File. 296
Alarmtext File. 296
Chapter 20 AlrmHdlgTextFile 299
20.1 Functional and System Overview 300
Functional Overview 300
20.2 Function Block Description 302
`AlrmHdlgTextFile` Function Block 302

20.3 Pin Description 303
Input Pin Description 304
Output Pin Description 306
Part VIII Commonly Used Sub-Structures 309
Chapter 21 Commonly Used Sub-Structures 311
Sub-Structure Descriptions 311
Part IX Troubleshooting 321
Chapter 22 Troubleshooting 323
Troubleshooting 323
Glossary 333
Index 337

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, service, 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” or “Warning” 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 a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

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

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could 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 EQUIPMENT

- 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 you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, 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, you 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.

⚠ WARNING

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 death, serious 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 temporary 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 Conveying Library.

Validity Note

This document has been updated for the release of SoMachine V4.2.

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.

Part I
Conveying Library System

At a Glance

This part describes the system requirements for the Conveying Library function blocks.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	System Requirements	17
2	General Presentation of Application Function Blocks	23
3	Matrix of Functions by Type of Equipments	31
4	Architectures	33
5	Operating Mode	37
6	Alarm Handling	43
7	Handshaking - Information Dataflow Between Conveying Equipments	73
8	Predefined Equipment Types	85
9	Use Cases	87

Chapter 1
System Requirements

At a Glance

This chapter describes the system requirements for the Conveying Library function blocks.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Conveying System Requirements	18
Technical Specifications and Limitations	20
Function Block Location in the Conveying Library	21

Conveying System Requirements

Overview

The following requirements are listed here:

- Hardware requirements
- Software requirements

Hardware Requirements

All the Conveying function blocks are designed to execute on either of the following controllers. The type of the communication interface used, depends on the user application.

- Modicon M241 Logic Controller
- Modicon M251 Logic Controller

Software Requirements

All the Conveying function blocks are running on application programs created with the programming software SoMachine 4.1.

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)*.

For IEC 61131-3 compatibility, the ability to add the EN/ENO input/output automatically to Function Blocks of certain programming languages is available to the programmer. However, for certain applications that require the complex interaction of multiple function blocks, the use of the IEC 61131-3 input to disable a function block in a series of interrelated functions affecting a process may lead to unintended operation of the system as a whole. For the functions contained in the Library that is the topic of the current document, this is especially true.

The EN/ENO inputs and outputs as defined by IEC 61131-3 are maladapted to, and therefore inappropriate for, the targeted application of these functions. Suddenly disabling one function by a falling edge on the EN input would require all outputs of the function block to immediately fall to their default states, and such an unanticipated action would cause in abrupt change to the entire process. The implication is that such an event would have deleterious results that may invoke undesirable consequences. Therefore, the EN/ENO inputs/outputs as defined by IEC 61131-3 are incompatible with the functions contained within this library.

⚠ WARNING

UNINTENDED MACHINE OPERATION

Do not use the EN/ENO functionality defined by IEC 61131-3 to control the behavior of the Application Function blocks.

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

NOTE: Verify that the EN/ENO option is disabled in the compiler options menu of SoMachine.

Technical Specifications and Limitations

Conveying Library

Deskzone Application Function Blocks

The `DeskzoneCmpx` and `DeskzoneSimp` function blocks are capable of controlling up to 25 equipments.

The memory consumed by these function blocks is always predefined by the total amount of equipments possible. This means that the memory consumption is always the same irrespective of the equipments used with the Deszone function blocks.

In case the amount of equipments exceeds 25, a second instance of the Deskzone function block is required. Due to memory constraints it is in general recommended to use only 1 Deskzone function block per HMI.

Retained Memory Used

A specific amount of retained memory is required by each function block.

The table below shows the memory consumption by application function blocks:

Application Function Block	Memory Consumption
`TrckCmpx`	136 bytes
`TrckSimp`	127 bytes
`TurnTable`	159 bytes
`ERC`	159 bytes
`Conveyor`	145 bytes
`Input`	145 bytes
`Output`	145 bytes
`DeskzoneCmpx`	001 bytes
`DeskzoneSimp`	001 bytes
`AlrmHdlg`	004 bytes
`AlrmHdlgTextFile`	94 bytes

Function Block Location in the Conveying Library

Function Block Location

Folder	Sub-Folder	Application Function Block

Conveying	Alarm Handling	`AlrmHdlg`
		`AlrmHdlgTextFile`
		Additional data sheets:
		● Conveying.Alarmtextfile.pdf
		● Conveying.Alarmtextfile.xls

	Conveyors	`Conveyor`
		`Input`
		`Output`

	Deskzone	`DeskzoneCmpx` `DeskzoneSimp`
	Tracking	`TrckCmpx` `TrckSimp`
	Transfer	`ERC`
	Turntable	`TurnTable`

Chapter 2
General Presentation of Application Function Blocks

General Description

Overview

The Conveying library contains Application Function Blocks (AFB) to manage a conveying line consisting of different mechanical equipments.

The graphic below shows a simple example of a conveying line and the application function blocks required to make it work:

- On the left you see the physical, mechanical equipment.
- On the right are the corresponding function blocks in your application.
- Conveyor equipment transport goods or materials from one end ot the other across the various conveyor equipment sections.
- At each transfer (cross-over) point between equipment sections, there is a data exchange (handshaking) between the function blocks, represented by the arrows in the graphic, in order to assure the correct flow of goods or material.

Main Types of Application Function Blocks

The Conveying library includes 2 main types of Application Function Blocks (AFB):

- Equipment functions
  - To realize individual equipment in a conveying line. For example, a `TurnTable` function block can be used to operate a physical turntable equipment in the controller.
- Transverse functions
  - To complement the Conveying equipment functions included in the Conveying library.

Names of the AFBs

Name of the AFB	Equipment function	Comment
`Conveyor`	Conveyor	-
`Input`	Input conveyor	-
`Output`	Output conveyor	-
`ERC`	Transfer table	ERC = Eccentric Roller Conveyor
`TurnTable`	Turntable	-

Name of the AFB	Transverse Functions	Comment
`AlrmHdlg`	Alarm Handling	-
`TrckCmpx`	Tracking	Used with `ERC` and `TurnTable` AFBs
`TrckSimp`		Used with `Conveyor`, `Input` and `Output` AFBs
`DeskzoneCmpx`	Deskzone	Used if the Tracking AFB is used
`DeskzoneSimp`		Used if the Tracking AFB is not used

The equipment function blocks of the Conveying library - representing different mechanical equipments - are classified into 3 main groups:
- Conveyors
- Eccentric Roller Conveyor (ERC)
- Turntable

The conveyor group contains 3 different sub-types of conveyor implementations:
- Conveyor
- Input
- Output

Input and output conveyors are special types of conveyors. They both have limited capabilities in that they only transfer material in one direction (only forward direction).

Legend on sensor positions:
(a) stop pallet
(b) switch transfer speed
(c) stop position (turn/lift)
(d) switch turn speed

The previous graphic (Conveyor and Turntable equipment) depicts a typical section of conveyor equipment and some of the devices to support its/their operation. The following table describes the equipment functions supported by each of the AFBs in the Conveying library, along with the devices required for those functions.

Equipment	AFB	Functions Supported by the Equipment	Number of Equipment Required					
				Sensor				
			Motor	Stop Pallet	Switch Transfer Speed	*Option: Unload Completed*	Stop Position Turn/Lift	Switch Turn Speed
Conveyor	Conveyor	1 speed/1 direction to transfer	1	1	-	*1*	-	
		1 speed/2 direction to transfer	1	1	-	-		
		2 speed/1 direction to transfer	1	1	1	*1*		
		2 speed/2 direction to transfer	1	2	2	-		
	Input	1 speed/1 direction to transfer	1	1	-	-		
		2 speed/1 direction to transfer	1	1	1	-		
	Output	1 speed/1 direction to transfer	1	1	-	*1*		
		2 speed/1 direction to transfer	1	1	1	*1*		
Turntable	TurnTable	1 speed/2 direction to transfer 1 speed/2 direction to turn	2	2	-	-	2	-
		2 speed/2 direction to transfer	2	2	2		2	-
		1 speed/2 direction to turn	2	2 *)			2	-
		1 speed/2 direction to transfer 2 speed/2 direction to turn	2	2	-		2	2
		2 speed/2 direction to transfer	2	2	2		2	2
		2 speed/2 direction to turn	2	2 *)			2	2
ERC	ERC	1 speed/2 direction to transfer 1 speed/2 direction to lift	3	2	-	-	2	-
		2 speed/2 direction to transfer	3	2	2			
		1 speed/2 direction to lift	3	2 *)				

***)** These equipment types realize the function of switching the speed by usage of the same sensor and sensor connection as for stopping the pallet.

For detailed description to the table, see below:

AFB: In this column are the names of the 5 equipment-controlling application function blocks included in the Conveying library.

Functions Supported by the Equipment : In this column, the 2 main attributes of speed and transfer directions are listed for each of the application function blocks.

- **Speeds** - Where applicable, the function blocks support 2 speeds of the equipment it is controlling. To accomplish this function, the following additional equipment must be present:
  - a drive or motor capable of multiple speeds
  - sensors for switching between the speeds
- **Transfer direction** - Generally, there are up to 2 transfer directions supported by the function blocks: forward and reverse. The exceptions are the `Input` and `Output` function blocks which support only the forward direction.
  The `TurnTable` function block adds turn (rotation), and the `ERC` function block adds the lift movement to its support of transfer directions.
  In addition to the equipment listed above for speeds, it is necessary to add additional motors and sensors to accomplish transfer directions beyond forward and reverse.

Number of Equipment Required : In this group of columns, the number and type of devices required to support the equipment function are indicated.

- **Motors** - This indicates the number of motors that are required to perform the induvidual equipment functions supported by the function block.
- **Sensors** - The sensors used for different purposes may or may not be the same type of sensor, depending on the materials or goods being transported on the conveyor equipment. They are referenced by their purpose as opposed to their type.
- **Stop Pallet** - In each transfer direction one sensor has to be provided to signal the stop of the pallet on the equipment. When the *stop pallet* sensor is reached, the pallet is occupying only the equipment where the *stop pallet* sensor is mounted.
- **Switch Transfer Speed** - For equipments which support 2 transfer speeds, one sensor for *switching the transfer speed* is necessary per direction. The sensor is mounted in the transfer direction before reaching the stop pallet position. The purpose of this *switch transfer speed* sensor is to reduce the velocity of the pallet in order to achieve a shorter stopping distance when the *stop pallet* sensor is triggered.
- **Upload Completed (optional)** - This sensor is optional and only possible for conveyor equipments where only one transfer direction is supported. During a pallet transfer process a pallet will be unloaded while the succeeding equipment will be loaded with this pallet. In general, the unload process will be finished when the stop pallet sensor is reached. This unloading process can be speeded up by the unload complete sensor to signal that the source equipment that the pallet has been taken over completely.

- **Stop Position (Turn/Lift)** - To move a pallet to the right or to the left of the initial direction, turntables or Eccentric Roller Conveyors (ERC) are used. To detect on turntables in which direction the rollerbed is positioned, or if the lift on ERC is in up or down position, 2 *stop position* sensors are required for `TurnTable` and `ERC` function blocks.
- **Switch Turn Speed** - If the turntable equipment allows the usage of 2 turn speeds, additional 2 sensors have to be placed on the equipment. The switch turn speed sensors are mounted between the *stop position (turn/lift)* sensors. On triggering the *switch turn speed* sensor the turning speed of the turntable will be reduced. The effect of this is to achieve a more precise stop position when approaching the *stop position (turn/lift)* sensor.

Chapter 3
Matrix of Functions by Type of Equipments

Functional Overview

Matrix Representation

The cross table between the type of equipments and the transverse function blocks is described below:

Equipment types\ Transverse AFB	Conveyor	Input	Output	Transfer Table ERC	Turntable
`AlrmHdlg`	Yes	Yes	Yes	Yes	Yes
`TrckSimp`	Yes	Yes	Yes	No	No
`TrckCmpx`	No	No	No	Yes	Yes
`DeskZoneSimp`	Used if the Tracking is not used				
`DeskZoneCmpx`	Used if the Tracking is used				

Legend	
Yes	Function blocks are intended to work together
No	Function blocks are not intended to work together, and cannot be connected

Chapter 4
Architectures

Section 4.1
Equipment Application Function Blocks

Overview of Equipment AFBs

Equipment AFB Description

All equipment application function blocks are built on the same structure as shown in the following simplified figure.

The equipment application function blocks are built with some common functions as described below:

- Initial preset
  - Only at the first cycle after system start, this function validates the equipment type and the plausibility of all configurable timers.
  - The initial status can set manually to TRUE.
- Equipment Alarm-Handling
  - This function receives the alarm and alert words from the other function and provides the information to the `AlrmHdlg` AFB.
  - This function receives from the `AlrmHdlg` AFB the reset information of the alarm and provides these information to the other function blocks.
- Sensor-control
  - This function does sensor noise filtering of the connected sensors on the OTB and monitors the noise detection.
  - According to the equipment type, the sensor inputs are monitored and the plausibility of the signals.
- Mode-control
  - This function receives the demand to switch the operation mode from automatic to manual and also from manual to automatic and switch to the mode after release from the function state machine.
  - This function provides the information to the other functions which mode is active.
- Init control
  - This function is provided by SoMachine, whether the Fieldbus node is running without problems.
- State machine
  - This function controls the mode change and allows a change only in empty or loaded state.
  - This function decides the material flow according to the status of the preceding and succeeding equipment and its type.
  - This function sends the start, speed and selection of the motor to the function motor-control.
  - If tracking is enabled, the function goes to unloading process only when tracking data Ok status is set by the `TrckSimp` or `TrckCmpx` function.
- Loading
  - This function receives the release information from the function state machine and it controls the loading process regarding the equipment type, the handshaking signals and the sensor status.
- Unloading
  - This function receives the release information from the function state machine and it controls the unloading process regarding the equipment type, the handshaking signals and the sensor status.

- Motor-control
  - This function receives the release of the motor start from the function state machine.
  - This function controls the feedback information of the controlled motors.

Usage of motors to the mechanical equipment is given in the below table:

	Motor #1	Motor #2	Motor #3
Conveyor	Conveyor	-	-
Input	Conveyor	-	-
Output	Conveyor	-	-
Turntable	Conveyor	Turntable	-
ERC	Conveyor	Chain transfer	Lifting

Chapter 5
Operating Mode

Operating Mode for Application Function Blocks

Operating Mode for Equipment AFB

The operating mode for the equipment application function blocks is defined below. Refer to Equipment Application Function Blocks *(see page 34)* for the description of each state.

Operating Mode for Alarm Handling

The operating mode for the `AlrmHdlg` AFB is defined below.

Operating Mode for Tracking

The `TrckSimp` and `TrckCmpx` are able to provide information along the conveying line in order to follow goods.

Operating Mode for Deskzone

The `DeskzoneSimp` and `DeskzoneCmpx` are used to exchange e.g. alarms, alerts, status between AFBs and XBTGT.

Chapter 6
Alarm Handling

At a Glance

This chapter describes the alarm and alert messages in the Conveying Library system.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
6.1	Alarm Messages	44
6.2	Alert Messages	60

Section 6.1
Alarm Messages

At a Glance

This section describes the alarm messages in the Conveying Library system.

What Is in This Section?

This section contains the following topics:

Topic	Page
Alarm Conditions	45
Alarm Words of Conveyor, Input, Output	46
Alarm Words of Deskzone Complex	48
Alarm Words of Deskzone Simple	49
Alarm Words of ERC	50
Alarm Words of TurnTable	53
Alarm Word of Tracking Complex	56
Alarm Word of Tracking Simple	57
Alarm Words of Alarm Handling	58
Alarm Word of AlrmHdlgTextFile	59

Alarm Conditions

Overview

Following are the alarm conditions:

If an alarm…	then…
occurs,	the corresponding bit within the alarm word is set to TRUE.
occurs,	the alarm release bit has to be set to FALSE.
is not active,	the corresponding bit within the alarm word is set to FALSE.

Acknowledge of Alarms

The alarms are acknowledged as follows:

- An alarm is present as long as the equipment alarm handling does not reset the alarm.
- The reset information is checked in every cycle.
- An alarm stays if it cannot be reset.

NOTE: For the `DeskzoneCmpx` function block the behavior is different, because an acknowledge of the alarm is not possible.

If...	then...
an alarm occurs,	fix the issue with a cold start of the controller.
the cause of alarm is not fixed,	the alarm rises again.

Alarm Words of Conveyor, Input, Output

Alarm Word 0

Alarm bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Equipment type mismatch alarm	Init control
1	Speed change delay time configuration forward alarm	Init control
2	Speed change delay time configuration reverse alarm	Init control
3	Reserved	–
4	Conveyor motor forward feedback alarm	Motor control
5	Conveyor motor reverse feedback alarm	Motor control
6...9	Reserved	–
10	Pallet detected in empty state alarm	Sensor control
11	Pallet detection lost in loaded state forward alarm	Sensor control
12	Pallet detection lost in loaded state reverse alarm	Sensor control
13	Sensitivity pallet sensor forward alarm	Sensor control
14	Sensitivity pallet sensor reverse alarm	Sensor control
15	Sensitivity speed change sensor forward alarm	Sensor control

Alarm Word 1

Alarm bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Sensitivity speed change sensor reverse alarm	Sensor control
1...8	Reserved	–
9	Time out load forward alarm	Load control
10	Time out load reverse alarm	Load control
11	Speed change timer configuration alarm	Load control
12	Conveyor sensor sequence alarm	Load control
13	Handshake logic alarm	Load control
14	Time out unload forward alarm	Unload control
15	Time out unload reverse alarm	Unload control

Alarm Word 2

Alarm bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Handshake logic alarm	Unload control
1	Reserved	–
2	Collision detection forward alarm	State machine
3	Collision detection reverse alarm	State machine
4, 5	Reserved	–
6	Flow inconsistency alarm	State machine
7	Both detected pallet sensors are in initial state alarm	State machine
8	Speed change sensor detected is in initial state alarm	State machine
9	Load cotnrol timer feedback elapsed alarm	State machine
10	Unload control timer feedback elapsed alarm	State machine
11...15	Reserved	–

Alarm Word 3

Alarm bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Reserved	–
1	Local alarm	Equipment alarm handling
2	Fieldbus Node detected error alarm	Equipment alarm handling
3	Motor circuit switch detected error Conveyor alarm	Equipment alarm handling
4	Drive detected error Conveyor motor alarm	Equipment alarm handling
5...8	Reserved	–
9	Parameter alarm of zone ID	Init control
10	Reserved	–
11	Equipment and equipment group ID are equal	Init control
12	Sensitivity alarm input conveyor release	Sensor control
13	Sensitivity alarm output conveyor release	Sensor control
14...15	Reserved	–

Alarm Words of Deskzone Complex

Alarm Word 0

Alarm bit description of the `DeskzoneCmpx` function block is given in the below table:

Bit	Notification	Source
0	Compare check alarm of zone ID in `DeskzoneCmpx` function block	Application program
1	Compare check alarm of equipment ID in `DeskzoneCmpx` function block	Application program
2	Compare check alarm of equipment type in `DeskzoneCmpx` function block	Application program
3	Parameter alarm of zone ID	Application program
4	Equality check alarm zone ID	Application program
5...15	Reserved	-

Alarm Words of Deskzone Simple

Alarm Word 0

Alarm bit description of the `DeskzoneSimp` function block is given in the below table:

Bit	Notification	Source
0...2	Reserved	-
3	Parameter alarm of zone Id	Application program
4	Equality check alarm zone Id	Application program
5...15	Reserved	-

Alarm Words of ERC

Alarm Word 0

Alarm bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Equipment type parameter alarm	Init control
1	Timer configuration alarm speed change delay forward	Init control
2	Timer configuration alarm speed change delay reverse	Init control
3	Reserved	–
4	Conveyor (360°/180°) motor forward feedback alarm	Motor control
5	Conveyor (360°/180°) motor reverse feedback alarm	Motor control
6	Conveyor (90°/270°) motor forward feedback alarm	Motor control
7	Conveyor (90°/270°) motor reverse feedback alarm	Motor control
8	Vertical move motor up feedback alarm	Motor control
9	Vertical move motor down feedback alarm	Motor control
10	Pallet detected in empty state alarm	Sensor control
11	Pallet detection lost in loaded state alarm	Sensor control
12	Reserved	–
13	Sensitivity pallet sensor forward alarm	Sensor control
14	Sensitivity pallet sensor reverse alarm	Sensor control
15	Sensitivity speed change sensor forward alarm	Sensor control

Alarm Word 1

Alarm bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Sensitivity speed change sensor reverse alarm.	Sensor control
1	Sensitivity position 180° ↔ 360° sensor alarm	Sensor control
2	Sensitivity position 90° ↔ 270° sensor alarm	Sensor control
3...7	Reserved	–
8	ERC position lost alarm	Sensor control
9	Time out load forward alarm	Load control
10	Time out load reverse alarm	Load control
11	Speed change timer configuration alarm	Load control

Bit	Notification	Source
12	Sensor sequence alarm	Load control
13	Handshake logic alarm	Load control
14	Time out unload forward alarm	Unload control
15	Time out unload reverse alarm	Unload control

Alarm Word 2

Alarm bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Handshake logic alarm	Unload control
1	Reserved	–
2	Collision detection -> 360° alarm	State machine
3	Collision detection -> 180° alarm	State machine
4	Collision detection -> 90° alarm	State machine
5	Collision detection -> 270° alarm	State machine
6	Flow inconsistency alarm	State machine
7	Reserved	–
8	Undefined sensor detection in init state	State machine
9	Load control timer feedback elapsed alarm	State machine
10	Unload control timer feedback elapsed alarm	State machine
11	Vertical move control timer feedback elapsed alarm	State machine
12	Destination address invalid alarm	State machine
13	Reserved	–
14	Timeout vertical move elapsed alarm	Vertical move control
15	Reserved	–

Alarm Word 3

Alarm bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Reserved	–
1	Local alarm	Equipment alarm handling
2	Fieldbus Node error detected	Equipment alarm handling

Bit	Notification	Source
3	Motor 1 circuit switch detected error alarm (Conveyor)	Equipment alarm handling
4	Drive 1 detected error Conveyor motor alarm (Conveyor)	Equipment alarm handling
5	Motor 2 circuit switch detected error alarm (Lifting/Chain transfer)	Equipment alarm handling
6	Drive 2 detected error alarm (Lifting/Chain transfer)	Equipment alarm handling
7	Motor 3 circuit switch detected error alarm (Lifting)	Equipment alarm handling
8	Drive 3 detected error alarm (Lifting)	Equipment alarm handling
9	Parameter alarm of zone ID	Init control
10	Reserved	–
11	Equipment and equipment group ID are equal	Init control
12...15	Reserved	–

Alarm Words of TurnTable

Alarm Word 0

Alarm bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Equipment type parameter alarm	Init control
1	Timer configuration alarm speed change delay forward	Init control
2	Timer configuration alarm speed change delay reverse	Init control
3	Timer configuration turn move time	Init control
4	Conveyor motor forward feedback alarm	Motor control
5	Conveyor motor reverse feedback alarm	Motor control
6	Turn motor forward feedback alarm	Motor control
7	Turn motor reverse feedback alarm	Motor control
8, 9	Reserved	-
10	Pallet detected in empty state alarm	Sensor control
11	Pallet detection lost in loaded state alarm	Sensor control
12	Reserved	-
13	Sensitivity pallet sensor forward alarm	Sensor control
14	Sensitivity pallet sensor reverse alarm	Sensor control
15	Sensitivity speed change sensor forward alarm	Sensor control

Alarm Word 1

Alarm bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Sensitivity speed change sensor reverse alarm	Sensor control
1	Sensitivity position up sensor alarm	Sensor control
2	Sensitivity position down sensor alarm	Sensor control
3	Sensitivity 180° position sensor alarm	Sensor control
4	Sensitivity 180° speed sensor alarm	Sensor control
5	Sensitivity 90° position sensor alarm	Sensor control
6	Sensitivity 90° speed sensor alarm	Sensor control
7	Position interlock sensor detected error in load/empty alarm	Sensor control
8	Turntable position lost alarm	Sensor control
9	Time out load forward alarm	Load control

Bit	Notification	Source
10	Time out load reverse alarm	Load control
11	Speed change timer configuration alarm	Load control
12	Sensor sequence alarm	Load control
13	Handshake logic alarm	Load control
14	Time out unload forward alarm	Unload control
15	Time out unload reverse alarm	Unload control

Alarm Word 2

Alarm bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Handshake logic alarm	Unload control
1	Reserved	-
2	Collision detection -> 360° alarm	State machine
3	Collision detection -> 180° alarm	State machine
4	Collision detection -> 90° alarm	State machine
5	Collision detection -> 270° alarm	State machine
6	Flow inconsistency alarm	State machine
7	Reserved	-
8	Undefined sensor detection in init state	State machine
9	Load control timer feedback elapsed alarm	State machine
10	Unload control timer feedback elapsed alarm	State machine
11	Turn move control timer feedback elapsed alarm	State machine
12	Destination address invalid alarm	State machine
13	Both interlock sensors (B5/B6) not on in init state	State machine
14	Timer turn move elapsed alarm	Turn control
15	Sensor sequence alarm in turn move process	Turn control

Alarm Word 3

Alarm bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Timer speed change turn move configuration alarm	Turn control
1	Local alarm	Equipment alarm handling

Bit	Notification	Source
2	Fieldbus node detected error	Equipment alarm handling
3	Motor 1 circuit switch detected error alarm (Conveyor)	Equipment alarm handling
4	Drive 1 detected error alarm (Conveyor)	Equipment alarm handling
5	Motor 2 circuit switch detected error alarm (Turntable)	Equipment alarm handling
6	Drive 2 detected error alarm (Turntable)	Equipment alarm handling
7, 8	Reserved	-
9	Parameter alarm of zone ID	Init control
10	Reserved	-
11	Equipment and equipment group ID are equal	Init control
12...15	Reserved	-

Alarm Word of Tracking Complex

Alarm Word 0

Alarm bit description of the `TrckCmpx` function block

Bit	Notification	Source
0	Parameter detected error: Value for zone ID = 0	Application program
1	Parameter detected error: Value for equipment ID = 0	Application program
2	Parameter detected error: Value for equipment type = 0	Application program
3	Parameter detected error: Value for length of tracking data is > 49	Application program
4	Parameter detected error: Value for length of tracking data is < 1	Application program
5	Tracking alarm: End sign <> 16#F0F0 is in loaded state	Application program
6	Tracking alarm: Source address is invalid	Application program
7	Tracking alarm: Current destination pointer is invalid	Application program
8	Tracking alarm: Found destination address is invalid	Application program
9...15	Reserved	–

Alarm Word of Tracking Simple

Alarm Word 0

Alarm bit description of the `TrckSimp` function block is given in the below table:

Bit	Notification	Source
0	Parameter detected error: Value for Zone ID = 0	Application program
1	Parameter detected error: Value for equipment ID = 0	Application program
2	Parameter detected error: Value for equipment type = 0	Application program
3	Parameter detected error: Value length of tracking data is > 49	Application program
4	Parameter detected error: Value length of tracking data is < 1	Application program
5	Tracking alarm: End Sign <> 16#F0F0 in loaded state	Application program
6...9	Tracking alarm: No direction return from state machine	Application program
10...15	Reserved	–

Alarm Words of Alarm Handling

Overview

The function block `AlrmHdlg` provides 2 alarm words:

- The first alarm word contains alarms, which effect the alarm release of the relevant zone.
- The second alarm word has no effect to the alarm release of the relevant zone. This word is for user utilization.

Alarm Word Zone

Alarm bit description of the `AlrmHdlg` function block for `Word 1`, external alarms, is given in the below table:

Bit	Notification	Source
0	Emergency Stop in zone	Input
1	Fieldbus detected error	Input
2...15	Reserved	-

Alarm bit description of the `AlrmHdlg` function block for `Word 2`, internal alarms, is given in the below table:

Bit	Notification	Source
0	Parameter detected error: No zone ID parameterized	Input
1	Parameter detected error: Inconsistency zone ID in project	Input
2...15	Reserved	-

Alarm Word of AlrmHdlgTextFile

Overview

The function block `AlrmHdlgTextFile` provides 1 alarm word:

- The individual bits provide by their state the causation of a possible alarm state of this function block.
- In case of alarm state an alarm reset operation is required.
- The function block does not execute while in an alarm state.

Alarm Word Zone

The alarm bit description of the `AlrmHdlgTextFile` function block is given in the below table:

Bit	Notification	Source
0	error detected on File.Open	File.Open
1	error detected on Dir.Create	Dir.Create
2	error detected on File.GetPos	File.GetPos
3	error detected on File.Write	File.Write
4	error detected on File.Close	File.Close
5	error detected on File.Delete	File.Delete
6	error detected on File.Rename	File.Rename
7	value of `i_byFileName` out of range (preferred input should fall in 1-4)	Input
8	internal error, value of `ret_uiCaseVar` is invalid/undefined	internal
9	internal error during WriteFile, value of `wAuxAreaToWrite` was invalid/undefined	internal
10	reserved	-
11	error detected as Pointer Not Configured	Input
12...15	reserved	-

Section 6.2
Alert Messages

At a Glance

This section describes the alert messages in the Conveying Library system.

What Is in This Section?

This section contains the following topics:

Topic	Page
Alert Words of Conveying, Input, Output	61
Alert Word of Deskzone (Complex, Simple)	63
Alert Word of ERC	64
Alert Word of TurnTable	66
Alert Word of Tracking Complex	68
Alert Word of Tracking Simple	69
Alert Word of Alarm Handling	70
Alert Word of AlrmHdlgTextFile	71

Alert Words of Conveying, Input, Output

Overview

If an alert has occurred, the function block continues to execute the code.

Alert Word 0

Alert bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Both pallet sensor detected in loaded state alert	Sensor control
1	Time out preload alert forward alert	State machine
2	Time out preload alert reverse alert	State machine
3	Reserved	–
4	Selected equipment type is invalid alert (only in manual mode)	State machine
5	Timer load control parameter alert	Init control
6	Speed change delay parameter alert	Init control
7	Motor stop delay parameter alert	Init control
8	Timer stage finish control parameter alert	Init control
9	Timer unload control parameter alert	Init control
10	Timer preload control parameter alert	Init control
11	Motor delay to start parameter alert	Init control
12	Constant flow timer configuration parameter alert	Init control
13	Reserved	–
14	Timer motor feedback parameter alert	Init control
15	Sensor filter parameter alert	Init control

Alert Word 1

Alert bit description of the `Conveyor`, `Input` and `Output` function block is given in the below table:

Bit	Notification	Source
0	Sensitivity non-operational detection count alert	Init control
1...6	Reserved	–
7	Acknowledge stage finish timeout alert	Load control
8	Stage finish timeout alert	Unload control

Bit	Notification	Source
9	Tracking alert	State machine
10	Stage finish control parameter alert	Init control
11	Handshake logic alert	State machine
12	Higher level alarm release not on	Equipment alarm handling
13	Fieldbus available bit not on	Equipment alarm handling
14, 15	Reserved	–

Alerts `Word 2` and `Word 3` are reserved.

Alert Word of Deskzone (Complex, Simple)

Overview

If an alert has occurred, the function block continues to execute the code.

Alert Word 1

Alert bit description of the `DeskzoneCmpx` and `DeskzoneSimp` function blocks are given in the below table:

Bit	Notification	Source
0	Selected zone ID alert is invalid in popup window 1	Application Program
1	Selected zone ID alert is unknown in popup window 1	Application Program
2	Selected equipment ID alert is unknown in popup window 1	Application Program
3	Selected equipment ID alert is in multiple use in popup window 1	Application Program
4	Life bit alert in popup window 1	Application Program
5	Selected zone ID alert is invalid in popup window 2	Application Program
6	Selected zone ID alert is unknown in popup window 2	Application Program
7	Selected equipment ID alert is unknown in popup window 2	Application Program
8	Selected equipment ID alert is in multiple use in popup window 2	Application Program
9	Life bit alert in popup window 2	Application Program
10	Selected zone ID alert is invalid in popup window 3	Application Program
11	Selected zone ID alert is unknown in popup window 3	Application Program
12	Selected equipment ID alert is unknown in popup window 3	Application Program
13	Selected equipment ID alert is in multiple use in popup window 3	Application Program
14	Life bit alert in popup window 3	Application Program
15	Parameter alert of timer life bit control	Application Program

Alert Word of ERC

Overview

If an alert occurs, the function block continues to execute the code.

Alert Word 0

Alert bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Reserved	–
1	Time out preload alert forward alert	State machine
2	Time out preload alert reverse alert	State machine
3	Reserved	–
4	Selected equipment type is invalid alert (only in manual mode)	State machine
5	Timer load control parameter alert	Init control
6	Speed change delay parameter alert	Init control
7	Motor stop delay parameter alert	Init control
8	Timer stage finish control parameter alert	Init control
9	Timer unload control parameter alert	Init control
10	Timer preload control parameter alert	Init control
11	Motor delay to start parameter alert	Init control
12	Constant flow timer configure parameter alert	Init control
13	Motor delay to switch over parameter alert	Init control
14	Motor feedback timer configure parameter alert	Init control
15	Sensor filter parameter alert	Init control

Alert Word 1

Alert bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	Sensitivity non-operational detection count alert	Init control
1	Time to vertically move parameter alert	Init control
2	Vertical move stop delay timer configure parameter alert	Init control
3	Reserved	–
4	Home position address is invalid alert	Init control
5	Priority logic address alert	Init control

Bit	Notification	Source
6	Back to home position timer configure parameter alert	Init control
7	Acknowledge stage finish timeout alert	Load control
8	Stage finish timeout alert unload process	Unload control
9	Tracking alert	State machine
10	Stage finish control parameter alert	Init control
11	Handshake logic alert	State machine
12	Higher level alarm release not on	Equipment alarm handling
13	Fieldbus available bit not on	Equipment alarm handling
14, 15	Reserved	-

Alert Word 2

Alert bit description of the ERC function block is given in the below table:

Bit	Notification	Source
0	All source stations disabled	Equipment alarm handling
1...15	Reserved	-

Alert `Word 3` is reserved.

Alert Word of TurnTable

Overview

If an alert has occurred, the function block will continue to execute the code.

Alert Word 0

Alert bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Reserved	–
1	Time out preload alert forward alert	State machine
2	Time out preload alert reverse alert	State machine
3	Reserved	–
4	Selected equipment type is invalid alert (only in manual mode)	State machine
5	Timer load control parameter alert	Init control
6	Speed change delay parameter alert	Init control
7	Motor stop delay parameter alert	Init control
8	Timer stage finish control parameter alert	Init control
9	Timer unload control parameter alert	Init control
10	Timer preload control parameter alert	Init control
11	Motor delay to start parameter alert	Init control
12	Constant flow timer configure parameter alert	Init control
13	Motor delay to switch over parameter alert	Init control
14	Motor feedback timer configure parameter alert	Init control
15	Sensor filter parameter alert	Init control

Alert Word 1

Alert bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	Sensitivity non-operational detection count alert	Init control
1	Time to turn move parameter alert	Init control
2	Turning move stop delay timer configure parameter alert	Init control
3	Turning speed change delay timer configure parameter alert	Init control
4	Home position address is invalid alert	Init control

Bit	Notification	Source
5	Priority logic address alert	Init control
6	Back to home position timer configure parameter alert	Init control
7	Acknowledge stage finish timeout alert	Load control
8	Stage finish timeout alert	Unload control
9	Tracking alert	State machine
10	Stage finish control parameter alert	Init control
11	Handshake logic alert	State machine
12	Higher level alarm release not on	Equipment alarm handling
13	Fieldbus available bit not on	Equipment alarm handling
14, 15	Reserved	–

Alert Word 2

Alert bit description of the `TurnTable` function block is given in the below table:

Bit	Notification	Source
0	All source stations disabled	Equipment alarm handling
1...15	Reserved	–

Alert `Word 3` is reserved.

Alert Word of Tracking Complex

Overview

If an alert has occurred, the function block continues to execute the code.

Alert Word 0

Alert bit description of the `TrckCmpx` function block is given in the below table:

Bit	Notification	Source
0	Equipment type sent by HMI is not identified	Application Program
1	Detected syntax error for HMI command	Application Program
2	Tracking alert: End sign <> 16#F0F0 during loading state	Application Program
3	Parameter alert: Destination pointer for source address 360° is invalid	Application Program
4	Parameter alert: Destination pointer for source address 90° is invalid	Application Program
5	Parameter alert: Destination pointer for source address 180° is invalid	Application Program
6	Parameter alert: Destination pointer for source address 270° is invalid	Application Program
7	Destination pointer sent by HMI is invalid	Application Program
8	Tracking function is disabled	Application Program
9...15	Reserved	–

Alert Word of Tracking Simple

Overview

If an alert has occurred, the function block continues to execute the code.

Alert Word 0

Alert bit description of the `TrckSimp` function block is given in the below table:

Bit	Notification	Source
0	Equipment type sent by HMI is not identified	Application program
1	Syntax error for HMI command	Application program
2	Tracking alert: End sign <> 16#F0F0 during the loading state	Application program
3...7	Reserved	–
8	Tracking function is disabled	Application program
9...15	Reserved	–

Alert Word of Alarm Handling

Overview

If an alert occurs, the `AlrmHdlg` function block continues to execute the code.

Alert Word Zone

`Word 1`, external alerts, is reserved.

The alert bit description of `AlrmHdlg` function block for `Word 2`, internal alerts, is given in the below table:

Bit	Notification	Source
0	Parameter detected error: Time for alarm release delayed on out of range	Input
1	Reset function: Selected zone ID unknown	Input
2	Reset function: Selected equipment ID unknown	Input
3...15	Reserved	-

Alert Word of AlrmHdlgTextFile

Overview

If an alert occurs, the `AlrmHdglTextFile` function block continues to execute the code.

Alert Word Zone

The alert bit description of `AlrmHdlgTextFile` function block is given in the below table:

Bit	Notification	Source
0	reserved	-
1	reserved	-
2	reserved	-
3	assigned value of `i_wMsgeCnt` is greater than 1000	Input
4...15	reserved	-

Chapter 7
Handshaking - Information Dataflow Between Conveying Equipments

At a Glance

This chapter describes the functionality and implementation of handshaking mechanism used throughout the Conveying Library.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functional Overview	74
Timing Diagram	78

Functional Overview

Handshake Mechanism - Dataflow

The handshake mechanism works considering a predefined transfer direction. That means, a pin of type `iq_stHdskNMnus1` must be linked with a pin of type `iq_stHdskNPlus1`.

The name of the handshake pins on the `TurnTable` and the ERC function blocks are different.

The graphic below shows the logical direction by default of the handshake pins identified by N+1 and N-1.

In some conveying applications it is needed to link handshake pins of the same type (N+1 or N-1) together for stations which are neighboured to a turntable or a transfer table (ERC = Eccentric Roller Conveyor). In this case the logical direction of the handshake pins on a `TurnTable` or an `ERC` function block can be switched between the types N+1 and N-1. Therefore the configuration parameters of the sub-structure `stConfOfHdskSwap` are provided.

In the following table the possible combinations are listed with the demanded value for the handshake swap parameter:

Handshake Pin of the Station	Linked to `iq_stHdsk90Deg`	Linked to `iq_stHdsk180Deg`	Linked to `iq_stHdsk270Deg`	Linked to `iq_stHdsk360Deg`
	Parameter: `x90DegEn`	Parameter: `x180DegEn`	Parameter: `x270DegEn`	Parameter: `x360DegEn`
`iq_stHdskNMnus1`	FALSE	TRUE	TRUE	FALSE
`iq_stHdskNPlus1`	TRUE	FALSE	FALSE	TRUE
`iq_stHdsk90Deg`	TRUE	FALSE	FALSE	TRUE
`iq_stHdsk180Deg`	FALSE	TRUE	TRUE	FALSE
`iq_stHdsk270Deg`	FALSE	TRUE	TRUE	FALSE
`iq_stHdsk360Deg`	TRUE	FALSE	FALSE	TRUE

NOTE: In the case that 2 handshake pins of the same type must be linked together, the handshake swap might be enabled only for either pin. This could cause if a `TurnTable` and/or an ERC function block must be linked.

Functional Description

The handshaking mechanism is explained with a simple application built with 3 conveyors of type 2 Speed and 2 Directions. The Conveyors are defined based on their positions. If the goods are transported to the right, then the middle conveyor is defined as Conveyor N, the right Conveyor is the succeeding Conveyor (Conveyor N+1), and the equipment to the left is the preceding Conveyor (Conveyor N-1).

Each Conveyor interfaces through the handshaking structure to their direct neighbor. These input/output structures carry the signals which are necessary to provide the information for taking over or handing over goods between two equipments. The direction of the information or signal dataflow is depicted in the diagram with arrows.

Example

Equipment Conveyor N carries a pallet and transports it to Conveyor N+1 (forward move).

Precondition before any movement:

Phase	Description
1	Auto mode is activated for both Conveyors, `xAuto` is ON.
2	Equipment type is existing and in the valid range `ebyEqupType`, for more details refer to chapter Predefined Equipment Types *(see page 85)*.

Preparation to hand over a pallet:

Phase	Description
1	From the Conveyor N perspective, the purpose is to unload the pallet. By switching on the signal `xPreUnld`, the Conveyor indicates the load status in the automatic mode and the readiness to unload.
2	Conveyor N+1 switches on the Conveyor motor and indicates its readiness to take over the pallet by setting `xLoadRdy` to TRUE.

Handing over the pallet:

Phase	Description
1	Conveyor confirms the ongoing unloading process by a high signal on `xUnldRdy`.
2	The handing over of the pallet continues unless the signals `xUnIdRdy` (Conveyor N) and `xLoadRdy` (Conveyor N+1) are not reset.
3	When the pallet reaches the final position of Conveyor (sensor N+1: B1), the Conveyor N+1 confirms by `xStagFnsh` that the pallet has reached its intended position and the Conveyor (N+1) motor stops.

Post-processing of pallet hand over:

Phase	Description
1	Conveyor N switches off the Conveyor Motor (N) and answers the `xStagFnsh` signal by setting the `xAcknStagFnsh` to TRUE.

The unloading/loading process is now completed and Conveyor N is empty. The equipment indicates the readiness to load a new cycle to the preceding Conveyor (N-1).

Timing Diagram

Handshake Timing Diagram

This timing diagram explains the handshake mechanism and its associated state transition for a pallet to travel continuously between N to N+1.

Pallet Transfer Conveyor Sensors

Pallet Transfer Handshake Signal

Pallet Transfer State Machine

Stage a: Pallet Reaches B3 Sensor of N-1 Conveyor

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N-1 turns on the Pre-Unload signal to N.
2	Conveyor N enters the pre-loading phase there by running the motor in empty state. This state of running the motor in empty state (pre-loading) enables the Conveyor to run the motor before the pallet arrives.

Stage b: Pallet Reaches B1 Sensor of N-1 Conveyor

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N-1 enters the loaded state.
2	Conveyor N-1 turns off the Pre-Unload signal and turns on the Unload Ready signal to N.
3	As N is ready to load and N-1 is ready to unload the pallet transfer starts between Conveyors N-1 and N.
4	Conveyors N enters loading state from empty.
5	After one scan, N-1 enters unloading state from loaded and the motor continues to run.

Stage c: Pallet Reaches B2 Sensor of Conveyor N (Negative Edge of Opposite Pallet Presence Sensor for Conveyor N)

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N turns on the Stage Finished signal to Conveyor N-1.
2	Conveyor N turns off the Load Ready signal to Conveyor N-1.

NOTE: Instance at stage c are those where the Stage Finished signal is generated by Conveyor N and N+1 respectively. Note that in Conveyor N, the Stage Finished signal is generated at falling edge of its B2 sensor, whereas for Conveyor N+1 the Stage Finished signal is generated at rising edge of its B3 sensor. This is because rising edge of B3 sensor comes before the falling edge of B2 sensor.

Stage d: One Scan After Stage c

During this instance, following are the reactions in phases:

Phase	Description
1	To acknowledge the Stage Finished signal from Conveyor N, the Conveyor N-1 turns on the Acknowledge Stage Finished signal to Conveyor N.
2	Conveyor N turns off its Stage Finished signal as N-1 Conveyor acknowledged the signal.
3	Conveyor N-1 turns off its Unload Ready signal to Conveyor N.
4	Conveyor N-1 enters constant flow delay state from unloading state, there by running the motor in empty state. The state machine remains in this state for a user defined time and then enters the empty state when the motor stops. This constant flow delay state allows the motor of the conveyor to continue running even after unloading so that the next loading process starts without stopping and starts the motor again.

Stage e: One Scan After Stage d

During this instance, following is the reaction in phase:

Phase	Description
1	Conveyor N-1 turns off its Acknowledge Stage Finished signal to Conveyor N as its Stage Finished signal is turned off.

Stage f: Pallet Reaches B3 Sensor of Conveyor N

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N turns on Pre-Unload signal to N+1.
2	Conveyor N+1enters pre-loading, there by running the motor in empty state. This state of running the motor in empty state (pre-loading) enables the Conveyor to run the motor before the pallet arrives.

Stage g: Constant Flow Delay Time Elapses for Conveyor N-1

During this instance, following is the reaction in phase:

Phase	Description
1	After the constant flow delay time is elapsed in empty state, the motor stops.

Stage h: Pallet Reaches B1 Sensor of Conveyor N

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N enters loaded state.
2	Conveyor N turns off the Pre-Unload signal and turns on the Unload Ready signal to Conveyor N.
3	As N+1 is ready to load and N is ready to unload, the pallet transfer starts between Conveyors N and N+1.
4	Conveyor N+1 enters loading state from empty.
5	After one scan, N enters unloading state from loaded and the motor continues to run.

Stage i: Pallet Reaches B3 Sensor of N+1 Conveyor (Rising Edge of Speed Change Sensor for N+1 Conveyor)

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N enters loaded state.
2	Conveyor N+1 turns off its Load Ready signal to Conveyor N.

NOTE: Instance at stage i are those where the Stage Finished signal is generated by Conveyor N and N+1 respectively. Note that in Conveyor N, the Stage Finished signal is generated at falling edge of its B2 sensor, whereas for Conveyor N+1 the Stage Finished signal is generated at rising edge of its B3 sensor, because rising edge of B3 sensor comes before the falling edge of B2 sensor.

Stage j: One Scan After Stage i

During this instance, following are the reactions in phases:

Phase	Description
1	To acknowledge the Stage Finished signal from Conveyor N+1, the Conveyor N turns on the Acknowledge Stage Finished signal to Conveyor N+1.
2	Conveyor N+1 turns off its Stage Finished signal as Conveyor N acknowledged it.
3	Conveyor N turns off its Unload Ready signal to Conveyor N+1.
4	Conveyor N enters constant flow delay state from unloading state, there by running the motor in empty state. The state machine stays in this state for a user defined time and enters the empty state when the motor stops. This constant flow delay state allows the motor of the conveyor to remain running even after unloading so that the next loading process starts without stopping and starts the motor again.

Stage k: One Scan After Stage j

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N turns off its Acknowledge Stage Finished signal to Conveyor N+1 as its Stage Finished signal turned off.

Stage l: Pallet Reaches B1 Sensor of N+1 Conveyor

During this instance, following are the reactions in phases:

Phase	Description
1	Conveyor N+1 enters loaded state.
2	Considering the succeeding equipment to N+1 is not ready to load, and after entering into loaded state the motor stops.

Stage m: Pallet Reaches B1 Sensor of N Conveyor and From Their ON

During this instance, following are the reactions in phases:

Phase	Description
1	Like shown from stage b to h, pallet transfer continues away from Conveyor N+1 in forward direction.

Chapter 8
Predefined Equipment Types

Predefined Equipment Types

Overview

The conveyor equipment application function blocks are configured through predefined equipment types. The equipment types specify the different modes of operations and sequences of instructions for the motor operation. You must define the equipment type for each of these AFBs used in your application.

The conveyor equipment AFBs are configured by assigning a value to a parameter element within one of the function block input structures. Specifically, the `i_stInit` input structure element `ebyEqupType` has to be given a predefined constant value. These values are contained within enumerated values defined in the Conveying library.

NOTE: An *enumeration* is a user defined data type, and can be constructed as a list of constant values.
There are 3 enumerated lists defined in the Conveying library:

- `EqupType` - includes all provided equipment types
- `EqupTypeSimp` - includes the equipment types for `Conveyor`, `Input` and `Output` function blocks
- `EqupTypeCmpx` - includes the equipment types for `TurnTable` and `ERC` function blocks.

For example, to establish (configure) a conveyor function block as being a 2 speed, 2 direction type of equipment, you would add the following line of code to set the correct constant vlaue for the parameter element of the input structure:

```
g_insConv_ID_10.i_stInit.ebyEqupType := EqupTypeSimp.Conv_2S2D
```

The following table summarizes the equipment types valid for each conveyor equipment function block:

Equipment	AFB	Enumerator Label	Enumerator Type			Pictogram *) slow speed / Fast speed	Value
			EqupType	EqupTypeSimp	EqupTypeCmpx		
Conveyor	Conveyor	Conv_1S1D	X	X	-		11
		Conv_1S2D	X	X	-		12
		Conv_2S1D	X	X	-		21
		Conv_2S2D	X	X	-		22
	Input	InputConv_1S	X	X	-		51
		InputConv_2S	X	X	-		52
	Output	OutputConv_1S	X	X	-		61
		OutputConv_2S	X	X	-		62
Turntable	TurnTable	Ttbl_1S_1TS	X	-	X		31
		Ttbl_2S4Sens_1TS	X	-	X		32
		Ttbl_2S2Sens_1TS	X	-	X		33
		Ttbl_1S_2TS	X	-	X		34
		Ttbl_2S4Sens_2TS	X	-	X		35
		Ttbl_2S2Sens_2TS	X	-	X		36
ERC	ERC	ERC_1S	X	-	X		41
		ERC_2S4Sens	X	-	X		42
		ERC_2S2Sens	X	-	X		43

X supported
- not supported
***)** The pictogram indicates what supported equipment function (speed, transfer direction) is defined for each AFB.

Chapter 9
Use Cases

At a Glance

This chapter describes the use cases in the simple and complex conveying line.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Simple Conveying Line	88
Complex Conveying Line	92

Simple Conveying Line

Overview

The following use cases of simple conveying line will be shown on the next pages:

- Use Case 1 - Without Tracking and Alarm Management Functions *(see page 88)*
- Use Case 2 - With Tracking Function and Without Alarm Management Function *(see page 89)*
- Use Case 3 - Without Tracking Function and with Alarm Management Function *(see page 90)*
- Use Case 4 - With Tracking and Alarm Management Functions *(see page 91)*

The use cases consist of

- 1 Input conveyor
- 1 Conveyor
- 1 Output conveyor

It shows the links between AFBs which have to be done for different level of functionalities.

Use Case 1 - Without Tracking and Alarm Management Functions

1 Link from sensors and to the motor control through distributed I/Os
2 Link to manage handshaking function between AFBs to follow goods on the conveying line

Use Case 2 - With Tracking Function and Without Alarm Management Function

1 Link from sensors and to the motor control through distributed I/Os
2 Link to manage handshaking function between AFBs to follow goods on the conveying line
3 Exchange of tracking information
4 Exchange of tracking data between AFBs to follow goods on the conveying line
5 Exchange of tracking data for Human Machine Interface (HMI)
6 Tracking data command from HMI
7 Exchange of control, status and tracking information
11 Exchange of control, status and operation modes

Use Case 3 - Without Tracking Function and with Alarm Management Function

NOTE: The `DeskzoneSimp` is used due to the fact that the `TrckSimp` or `TrckCmpx` is not used.

1 Link from sensors and to the motor control through distributed I/Os
2 Link to manage handshaking function between AFBs to follow goods on the conveying line links
7 Exchange of control, status and tracking information
8 Alarm and alert information to HMI. Alarms acknowledge and reset from HMI
9 Exchange alarm and alert words from machine's AFBs. Exchange of alarm reset and Alarm release to machine's AFBs
10 Exchange of alarm and alert information
11 Exchange of control, status and operation modes

Use Case 4 - With Tracking and Alarm Management Functions

All the functionalities are included in this case and are defined below.

1 Link from sensors and to the motor control through distributed I/Os
2 Link to manage handshaking function between AFBs to follow goods on the conveying line
3 Exchange of tracking information
4 Exchange of tracking data between AFBs to follow goods on the conveying line
5 Exchange of tracking data for Human Machine Interface (HMI)
6 Tracking data command from HMI
7 Exchange of control, status and tracking information
8 Alarm and alert information to HMI. Alarms acknowledge and reset from HMI.
9 Exchange alarm and alert words from machine's AFBs. Exchange of Alarm reset and Alarm release to machine's AFBs
10 Exchange of alarm and alert information
11 Exchange of control, status and operation modes

Complex Conveying Line

Overview

The following use cases of complex conveying line will be shown on the next pages:

- Use Case 1 - Without Tracking and Alarm Management Functions *(see page 92)*
- Use Case 2 - With Tracking Function and Without Alarm Management Function *(see page 93)*

The use cases use

- 5 Conveyors
- 1 Turntable
- 1 Transfer Table/ERC

Use Case 1 - Without Tracking and Alarm Management Functions

2 Link to manage handshaking function between AFBs to follow goods on the conveying line

Use Case 2 - With Tracking Function and Without Alarm Management Function

This case shows which functions have to be used while using turn tables, transfer tables and simple conveyors. `TrckSimp` and `TrckCmpx` function blocks are different:

- AFB `TrckSimp` is used if the equipment is a simple conveyor.
- AFB `TrckCmpx` is used if the equipment can sort goods and send it to several ways such as a turntable or a transfer table (ERC).

2 Link to manage handshaking function between AFBs to follow goods on the conveying line
3 Exchange of tracking information

NOTE:
The previous diagram has to be completed with the `DeskzoneCmpx` in the same way as the one showed in the previous chapters Case 2 - With Tracking Function and Without Alarm Management function *(see page 89)* for simple conveying line.

If an alarm management has to be used, then the use case is similar to Use Case 4 - With Tracking and Alarm Management Functions *(see page 91)* in the simple conveying line.

Part II
Conveyors

At a Glance

This part describes the following function blocks in the Conveying Library system:

- `Conveyor`
- `Input`
- `Output`

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
10	Conveyor: Transfer of Goods	97
11	Input: Loading of Goods	119
12	Output: Exit of Goods	141

Chapter 10
Conveyor: Transfer of Goods

At a Glance

This chapter describes the functionality and implementation of the `Conveyor` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
10.1	Functional and System Overview	98
10.2	Function Block Description	101
10.3	Pin Description	102
10.4	Interfaces	111

Section 10.1
Functional and System Overview

At a Glance

This section describes the functional and system overview of the `Conveyor` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Functional Overview	99
System Overview	100

Functional Overview

Functional Description

The function block `Conveyor` is part of the Conveying library with the following features:

- Operation Modes: Automatic, Manual and Local
- Moves Conveyor in forward or reverse direction with one or two speeds
- Receives control information from the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Provides status information to the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Receives status information from the function block `TrckSimp` if Tracking is enabled
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

System Overview

System View

If the Conveyor is used to deliver in 360° direction (considered in general as forward), only sensors B1 and B3 are taking effect on the function of the Conveyor. As B3 sensor is merely used to change the speed of the pallet, this sensor is only applicable for Conveyors which support two speeds. All sensors are wired to OTBs which are providing their status by the OTB structures with the input `i_stOtbIput`.

If the Conveyor is used to deliver in 180° direction (considered in general as reverse), only sensors B2 and B4 are taking effect on the function of the Conveyor. As B4 sensor is merely used to change the speed of the pallet, this sensor is only applicable for Conveyors which support two speeds. All sensors are wired to OTBs which are providing their status by the OTB structures with the input `i_stOtbIput`.

Section 10.2
Function Block Description

Conveyor Function Block

Pin Diagram

Section 10.3
Pin Description

At a Glance

This section gives the pin description of the `Conveyor` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	103
Output Pin Description	106
Input/Output Pin Description	110

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stEqupCmd	stEqupCmd	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to structure parameter description i_stEqupCmd *(see page 103)*.
i_stOtbIput	stConvOtbIput	Connects to OTB where the content of this structure provides the status information of the equipment (like drive and sensor). Refer also to structure parameter description i_stOtbIput *(see page 104)*.
i_stInit	stConvInit	Initializes values of timer and the enable/disable function during the start up phase. Refer also to structure parameter description i_stInit *(see page 104)*.
i_xFbAvai	BOOL	Indicates the availability of the Fieldbus [depends on the start-up time].
i_xFbNodeOk	BOOL	Indicates the availabilty of Fieldbus Node.
i_stTrckStat	stTrckStat	Connects to function block TrckSimp. Refer also the structure parameter description i_stTrckStat *(see page 104)*.
i_xEmgy	BOOL	TRUE: No local stop FALSE: Local stop
i_stAlrmCtrl	stAlrmCtrl	Connects to function block AlrmHdlg. Refer also the structure parameter description i_stAlrmCtrl *(see page 105)*.
i_xProgRstAlrm	BOOL	External reset of alarm.
i_xDoInit	BOOL	TRUE: Initial state is enabled FALSE: Initial state is disabled

i_stEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Contains ● Mode selection ● Manual command ● Conveyor direction command ● Equipment ID, type ID, and zone ID ● Alarm reset
xManCmdVldt	BOOL	Validates the manual movement

i_stOtbIput (Data Type: stConvOtbIput)

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Displays the status of Conveyor sensors
stMotStat	stMotStat	Displays the status of the connected motor
stMotFbck	stMotFbck	Displays the running direction of the motor
stRemMod	stRemMod	Displays the remote mode (local or maintenance)

i_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

i_stTrckStat (Data Type: stTrckStat)

Structure Parameter	Data Type	Description
xInfoOk	BOOL	Indicates if tracking data are valid as result of the End sign check.
xFuncOk	BOOL	Indicates the status of function block TrckSimp. TRUE: Function block is ready for operation.

i_stAlrmCtrl (Data Type: stAlrmCtrl)

Structure Parameter	Data Type	Description
xRstAlrm	BOOL	Resets alarm from function block AlrmHdlg in zone
xAlrmRele	BOOL	TRUE: Overall alarm is not active in zone FALSE: Overall alarm is active in zone

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stEqupStat	stEqupStat	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to structure parameter description of q_stEqupStat *(see page 106)*.
q_stStmStatSimp	stStmStatSimp	Connects to function block TrckSimp. Refer also to structure parameter description of q_stStmStatSimp *(see page 107)*.
q_stOtbOput	stConvOtbOput	Elements of this structure must be linked to physical outputs to control the equipment. Refer also to structure parameter description of q_stOtbOput *(see page 108)*.
q_stInit	stConvInit	Indicates the parameterized values of timer and the enable/disable function. Refer also to structure parameter description of q_stInit *(see page 108)*.
q_stDevAlrm	stDevAlrm	Connects to function block AlrmHdlg. Refer also to structure parameter description of q_stDevAlrm *(see page 109)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stEqupStat (Data Type: stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of mode status
stStmStatCmpx	stStmStatCmpx	Information of complex state machine status
stConvSenFltr	stConvSenFltr	Information of filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSen	Information of the filtered sensors status from the Turntable [not used in Conveyor; all structure elements are in FALSE state]

Structure Parameter	Data Type	Description
stErcSenFltr	stErcSen	Information of the sensors from the ERC position (up or down) [not used in Conveyor, all structure elements are in FALSE state]
dwMot1HrsofOp	DWORD	Information of the operation hour of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hour of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback [not used in Conveyor, all structure elements are in FALSE state]
dwMot3HrsofOp	DWORD	Information of the operation hour of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback [not used in Conveyor, all structure elements are in FALSE state]
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light). Used only by Input and Output AFB.
xAlrm	BOOL	Indicates that an alarm is active
xAlrt	BOOL	Indicates that an alert is active

q_stStmStatSimp (Data Type: stStmStatSimp)

Structure Parameter	Data Type	Description
xEmty	BOOL	Status information is empty
xLdng	BOOL	Status information is loading
xLded	BOOL	Status information is loaded
xUldg	BOOL	Status information is unloading
xFwd	BOOL	State machine is decided to transfer in forward direction
xRev	BOOL	State machine is decided to transfer in in reverse direction
xManModinPrgs	BOOL	Manual mode is in progress, selected by user
xStagFnsh	BOOL	Handshake signal stage is complete to indicate that tracking data is assumed
xCycRst	BOOL	Cycle restart is active
xCyclRstInPrgs	BOOL	Cycle restart is in progress
xDolnitInPrgs	BOOL	Indicates that the Go to Intialize Mode is in progress
xRst	BOOL	Alarm reset information for the relevant function block TrckSimp or TrckCmpx
xStmAuto	BOOL	Status information is in automatic mode

Structure Parameter	Data Type	Description
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

q_stOtbOput (Data Type: stConvOtbOput)

Structure Parameter	Data Type	Description
stMotCmd	stMotCmd	Motor control commands
xAlrm	BOOL	Alarm information to switch on the local alarm lamp

q_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

q_stDevAlrm (Data Type: stDevAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
wEqupId	WORD	Equipment ID of the equipment where the alarm/alert-event occurred
ebyEqupType	EqupType	Equipment type of the equipment where the alarm/alert-event occurred
awAlrmStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alarms. See also Alarm Words of Conveyor, Input, Output *(see page 46)*.
awAlrtStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alerts See also Alert Words of Conveyor, Input, Output *(see page 61)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
`iq_stHdskNMnus1`	`stHdkIntf`	Connects to UME180°. Refer also to the below structure parameter description `iq_stHdskNMnus1` and `iq_stHdskNPlus` *(see page 110)*.
`iq_stHdskNPlus1`	`stHdkIntf`	Connects to UME360°. Refer also to the below structure parameter description `iq_stHdskNMnus1` and `iq_stHdskNPlus` *(see page 110)*.

`iq_stHdskNMnus1` and `iq_stHdskNPlus1` (Data Type: `stHdskIntf`)

Both the structures have a set of structure elements with identical names.

Structure Parameter	Data Type	Description
`stFwd`	`stHdsk`	Handshake signals if material flow is in forward direction.
`stRev`	`stHdsk`	Handshake signals if material flow is in reverse direction.

NOTE: N minus 1(N-1) is the preceding element/equipment/conveyor and N plus 1 (N+1) is the subsequent element/equipment/conveyor.

Section 10.4
Interfaces

At a Glance

This section gives an overview how the `Conveyor` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
Conveyor Interfaces	112
Linking the Function Blocks	113

Conveyor Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Handshaking	Information needs to be exchanged between 2 neighboring pieces of equipment in order to manage the material flow.

Optional interface connections are related to:

Tracking	Supervision of the material flow by managing and validating the tracking data.
HMI operation	Together with a Deskzone AFB, a group of equipment can be operated by and monitored using an HMI (e.g., Magelis HMI).
Alarm and alert monitoring	In conjunction with the `AlrmHdlg` FB, the alarms and alerts signaled by the Equipment function block can be logged and summarized.

Linking the Function Blocks

Handshake Interface

To allow the material flow from one equipment to another, handshake information between these 2 equipment function blocks need to be shared. This interface is realized by using a variable of type `stHdskItf` linked to the In_Output pins of the concerned Equipment function blocks. For more detailed information refer to the chapter *Handshaking – Information Dataflow Between Conveying Equipments* *(see page 73)*.

On one hand, the `Conveyor` FB needs to read from the structure the relevant information from the neighboring equipment, on the other hand the `Conveyor` FB has to write its own status in the same structure (handshake variable). In order to share this information by using the identical structure, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Handshake interface with preceding Conveyor	`iq_stHdskNMnus1`	`iq_stHdskNPlus1`	`Conveyor`
Handshake interface with succeeding Conveyor	`iq_stHdskNPlus1`	`iq_stHdskNMnus1`	
Handshake interface with preceding Input-Conveyor	`iq_stHdskNMnus1`	`iq_stHdskNPlus1`	`Input`
Handshake interface with succeeding Output-Conveyor.	`iq_stHdskNPlus1`	`iq_stHdskNMnus1`	`Output`
Handshake interface with a Turntable or ERC. Seen from the position of the Turntable or ERC, the Conveyor is located on station 90, 180, 270 or 360 degree.	`iq_stHdskNPlus1` / `iq_stHdskNMnus1`	`iq_stHdsk90Deg`/ `iq_stHdsk180Deg` / `iq_stHdsk270Deg` / `iq_stHdsk360Deg`	`TurnTable` / `ERC`

The following screen capture shows, how the interfacing can be done in SoMachine on example of an `Input` and a `Conveyor` function block:

Tracking Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one `Conveyor` AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from `Conveyor` to `TrckSimp` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information from `TrckSimp` AFB about the state and the validity of the tracking data.	`i_stTrckStat`	`q_stStat`	`TrckSimp`
Information to the `TrckSimp` AFB about the state for managing the tracking data transfer.	`q_stStmStatSimp`	`i_ststmStatSimp`	

Deskzone Interface

The Deskzone FB (`DeskzoneSimp` and `DeskzoneCmpx`) manages a group of equipments by gathering and providing the necessary information from the equipment and to the equipment. This information will be shared with the HMI (e.g. Magelis HMI) in order to be able to provide the status information to the HMI and to receive the command from the HMI to operate the Conveying line.

The Deskzone FB uses arrays of structures to exchange the data between `Conveyor` and Deskzone function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Commands to control the Conveyor by the operator via the Deskzone AFB what is linked to an HMI.	`i_stEqupCmd`	`q_astEqupCmd[x]`	`DeskzoneSimp` / `DeskzoneCmpx`
State information for monitoring by the operator via the Deskzone AFB what is linked to an HMI.	`q_stEqupStat`	`iq_astEqupStat[x]`	

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 43)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `Conveyor` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the `Conveyor` AFB from `AlrmHdlg` AFB.	`i_stAlrmCtrl`	`q_astAlrmCtrl[x]`	`AlarmHdlg`
`Conveyor` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert lds.	`q_stDevArm`	`iq_astEqupAlrm[x].stDevAlrm`	

Chapter 11
Input: Loading of Goods

At a Glance

This chapter describes the functionality and implementation of the `Input` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
11.1	Functional and System Overview	120
11.2	Function Block Description	123
11.3	Pin Description	124
11.4	Interfaces	133

Section 11.1
Functional and System Overview

At a Glance

This section describes the functional and system overview of the `Input` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Functional Overview	121
System Overview	122

Functional Overview

Functional Description

The `Input` function block is part of the Conveying library with the following features:

- Operation Mode: Automatic, Manual and Local
- Moves Conveyor in forward direction with one or two speeds
- Receives Control Information from the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Provides status information to the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Receives the status information from the function block `TrckSimp` if Tracking is enabled
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

System Overview

System View

Section 11.2
Function Block Description

`Input` Function Block

Pin Diagram

Section 11.3
Pin Description

At a Glance

This section gives the pin description of the `Input` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	125
Output Pin Description	128
Input/Output Pin Description	132

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stEqupCmd	stEqupCmd	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to structure parameter description i_stEqupCmd *(see page 125)* .
i_stOtbIput	stIpcnvOtbIput	Connects to OTB where the content of this structure provides the status information of the equipment (like drive and sensors). Refer also to structure parameter description i_stOtbIput *(see page 126)*.
i_stInit	stConvInit	Initializes the values of timer and the enable/ disable function during the start up phase. Refer also to structure parameter description i_stInit *(see page 126)*.
i_xFbAvai	BOOL	Indicates the availability of the Fieldbus [depends on the start-up time].
i_xFbNodeOk	BOOL	Indicates the availabilty of Fieldbus Node.
i_stTrckStat	stTrckStat	Connects to function block TrckSimp. Refer also to structure parameter description i_stTrckSta *(see page 127)* .
i_xEmgy	BOOL	TRUE: No local stop FALSE: Local stop
i_stAlrmCtrl	stAlrmCtrl	Connects to function block AlrmHdlg. Refer also to structure parameter description i_stAlrmCtrl *(see page 127)* .
i_xProgRstAlrm	BOOL	External reset of alarm.
i_xDoInit	BOOL	TRUE: Initial state is enabled FALSE: Initial state is disabled

i_stEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Contains ● Mode selection ● Manual command ● Conveyor direction command ● Equipment ID, type ID, and zone ID ● Alarm reset
xManCmdVldt	BOOL	Validation of the manual movement

i_stOtbIput (Data Type: stIpcnvOtbIput)

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Displays the status of the Conveyor sensors
xIputRele	BOOL	TRUE: New pallet FALSE: Disabled
xIputAckn	BOOL	Acknowledge signal Conveyor input
stMotStat	stMotStat	Displays the status of the connected motor
stMotFbck	stMotFbck	Displays the running direction of the motor
stRemMod	stRemMod	Displays the remote mode (local or maintenance)

i_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

i_stTrckStat (Data Type: stTrckStat)

Structure Parameter	Data Type	Description
xInfoOk	BOOL	Indicates if tracking data are valid as result from the End sign check.
xFuncOk	BOOL	Indicates the status of function block TrckSimp. TRUE: Function block is ready for operation.

i_stAlrmCtrl (Data Type: stAlrmCtrl)

Structure Parameter	Data Type	Description
xRstAlrm	BOOL	Resets alarm from function block AlrmHdlg in zone
xAlrmRele	BOOL	TRUE: Overall alarm is not active in zone FALSE: Overall alarm is active in zone

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stEqupStat	stEqupStat	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to the below structure parameter description q_stEqupStat *(see page 128)*.
q_stStmStatSimp	stStmStatSimp	Connects to function block TrckSimp. Refer also to structure parameter description q_stStmStatSimp *(see page 129)* .
q_stOtbOput	stIpcnvOtbOput	Elements of this structure must be linked to physical outputs to control the equipment. Refer also to structure parameter description q_stOtbOput *(see page 130)*.
q_stInit	stConvInit	Indicates the parameterized values of timer and the enable/disable functions. Refer also to structure parameter description q_stInit *(see page 130)* .
q_stDevAlrm	stDevAlrm	Connects to function block AlrmHdlg. Refer also to structure parameter description q_stDevAlrm *(see page 131)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stEqupStat (Data Type: stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment Group ID, to control all equipments of the zone
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of the status of the mode
stStmStatCmpx	stStmStatCmpx	Information of the status of the complex state machine
stConvSenFltr	stConvSenFltr	Information of the filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSen	Information of the filtered sensors status from the Turntable [not used in Conveyor, all structure elements on FALSE state]

Structure Parameter	Data Type	Description
stErcSenFltr	stErcSen	Information of the sensors from the ERC position (up or down) [not used in Conveyor, all structure elements on FALSE state]
dwMot1HrsofOp	DWORD	Information of the operation hour of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hour of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback [not used in Conveyor, all structure elements on FALSE state]
dwMot3HrsofOp	DWORD	Information of the operation hour of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback [not used in Conveyor, all structure elements on FALSE state]
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light).
xAlrm	BOOL	Indicates that an alarm is active
xAlrt	BOOL	Indicates that an alert is active

q_stStmStatSimp (Data Type: stStmStatSimp)

Structure Parameter	Data Type	Description
xEmty	BOOL	Status information is empty
xLdng	BOOL	Status information is loading
xLded	BOOL	Status information is loaded
xUldg	BOOL	Status information is unloading
xFwd	BOOL	State machine is decided to transfer in forward direction
xRev	BOOL	State machine is decided to transfer in in reverse direction
xManModinPrgs	BOOL	Manual mode is in progress, selected by user
xStagFnsh	BOOL	Handshake signal stage is complete to indicate that tracking data is assumed
xCycRst	BOOL	Cycle restart is active
xCyclRstInPrgs	BOOL	Cycle restart is in progress
xDolnitInPrgs	BOOL	Indicates that the Go to Intialize Mode is in progress
xRst	BOOL	Alarm reset information for the relevant function block TrckSimp or TrckCmpx
xStmAuto	BOOL	Status information is in automatic mode

Structure Parameter	Data Type	Description
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

q_stOtbOput (Data Type: stIpcnvOtbOput)

Structure Parameter	Data Type	Description
stMotCmd	stMotCmd	Motor control commands
xAlrm	BOOL	Alarm information to switch on the local alarm lamp
stBecn	stBecn	Indicates the status of the input loading process

q_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID, to control all equipments of the zone
xTrckEn	BOOL	Tracking is enabled

q_stDevAlrm (Data Type: stDevAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
wEqupID	WORD	Equp ID of the equipment where the alarm/alert-event occurred
ebyEqupType	EqupType	Type of the equipment where the alarm/alert-event occurred
awAlrmStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alarms See also Alarm Words of Conveyor, Input, Output *(see page 46)*.
awAlrtStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alerts See also Alert Words of Conveyor, Input, Output *(see page 61)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
`iq_stHdskNPlus1`	`stHdskIntf`	Connects to UME360°. Refer also to structure parameter descripition `iq_stHdskNplus1` given below.

`iq_stHdskNPlus1` (Data Type: `stHdskIntf`)

Structure Parameter	Data Type	Description
`stFwd`	`stHdsk`	Handshake signals if material flow is in forward direction
`stRev`	`stHdsk`	Handshake signals if material flow is in reverse direction

Section 11.4
Interfaces

At a Glance

This section gives an overview how the `Input` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Interfaces	134
Linking the Function Blocks	135

Input Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Handshaking	Information needs to be exchanged between 2 neighboring pieces of equipment in order to manage the material flow.

Optional interface connections are related to:

Tracking	Supervision of the material flow by managing and validating the tracking data.
HMI operation	Together with a Deskzone AFB, a group of equipment can be operated by and monitored using an HMI (e.g., Magelis HMI).
Alarm and alert monitoring	In conjunction with the `AlrmHdlg` FB, the alarms and alerts signaled by the Equipment function block can be logged and summarized.

Linking the Function Blocks

Handshake Interface

To allow the material flow from one equipment to another, handshake information between these 2 equipment function blocks need to be shared. This interface is realized by using a variable of type `stHdskItf` linked to the In_Output pins of the concerned Equipment function blocks. For more detailed information refer to the chapter *Handshaking – Information Dataflow Between Conveying Equipments* *(see page 73)*.

On one hand, the `Input` FB needs to read from the structure the relevant information from the neighboring equipment, on the other hand the `Input` FB has to write its own status in the same structure (handshake variable). In order to share this information by using the identical structure, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Handshake interface with succeeding Conveyor	`iq_stHdskNPlus1`	`iq_stHdskNMnus1`	`Conveyor`
Handshake interface with succeeding Output-Conveyor	`iq_stHdskNPlus1`	`iq_stHdskNMnus1`	`Output`
Handshake interface with a Turntable or ERC. Seen from the position of the Turntable or ERC, the Conveyor is located on station 90, 180, 270 or 360 degree.	`iq_stHdskNPlus1`	`iq_stHdsk90Deg`/ `iq_stHdsk180Deg` / `iq_stHdsk270Deg` / `iq_stHdsk360Deg`	`TurnTable` / `ERC`

The following screen capture shows, how the interfacing can be done in SoMachine on example of an `Input` and a `Conveyor` function block:

Tracking Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one `Input` AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from `Input` to `TrckSimp` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information from `TrckSimp` AFB about the state and the validity of the tracking data.	`i_stTrckStat`	`q_stStat`	`TrckSimp`
Information to the `TrckSimp` AFB about the state for managing the tracking data transfer.	`q_stStmStatSimp`	`i_ststmStatSimp`	

Deskzone Interface

The Deskzone FB (`DeskzoneSimp` and `DeskzoneCmpx`) manages a group of equipments by gathering and providing the necessary information from the equipment and to the equipment. This information will be shared with the HMI (e.g. Magelis HMI) in order to be able to provide the status information to the HMI and to receive the command from the HMI to operate the Conveying line.

The Deskzone FB uses arrays of structures to exchange the data between `Input` and Deskzone function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Commands to control the Input-Conveyor by the operator via the Deskzone AFB what is linked to an HMI.	`i_stEqupCmd`	`q_astEqupCmd[x]`	`DeskzoneSimp` / `DeskzoneCmpx`
State information for monitoring by the operator via the Deskzone AFB what is linked to an HMI.	`q_stEqupStat`	`iq_astEqupStat[x]`	

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 43)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `Input` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the `Input` AFB from `AlrmHdlg` AFB.	`i_stAlrmCtrl`	`q_astAlrmCtrl[x]`	`AlarmHdlg`
`Input` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert lds.	`q_stDevArm`	`iq_astEqupAlrm[x].stDevAlrm`	

Chapter 12
Output: Exit of Goods

At a Glance

This chapter describes the functionality and implementation of the `Output` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
12.1	Functional and System Overview	142
12.2	Function Block Description	145
12.3	Pin Description	146
12.4	Interfaces	155

Section 12.1
Functional and System Overview

At a Glance

This section describes the functional and system overview of the `Output` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Functional Overview	143
System Overview	144

Functional Overview

Functional Description

The `Output` function block is part of the Conveying library with the following features:

- Operation Modes: Automatic, Manual and Local
- Moves Conveyor in forward direction with one or two speeds
- Receives control information from the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Provides status information to the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Receives status information from the function block `TrckSimp` if Tracking is enabled
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

System Overview

System View

Section 12.2
Function Block Description

`Output` Function Block

Pin Diagram

Section 12.3
Pin Description

At a Glance

This section gives the pin description of the `Output` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	147
Output Pin Description	150
Input/Output Pin Description	154

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stEqupCmd	stEqupCmd	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to structure parameter description i_stEqupCmd *(see page 147)* .
i_stOtbIput	stOpcnvOtbIput	Connects to OTB where the content of this structure provides the status information of the equipment (like drive and sensors). Refer also to structure parameter description i_stOtbOput *(see page 148)*.
i_stInit	stConvInit	Initializes the values of timer and the enable/ disable function during the start up phase. Refer also to structure parameter description i_stInit *(see page 148)*.
i_xFbAvai	BOOL	Indicates the availability of the Fieldbus [depends on the start-up time]
i_xFbNodeOk	BOOL	Indicates the availability of the Fieldbus Node.
i_stTrckStat	stTrckStat	Connects to function block TrckSimp. Refer also to structure parameter description i_stTrckStat *(see page 149)*.
i_xEmgy	BOOL	TRUE: No local stop FALSE: Local stop
i_stAlrmCtrl	stAlrmCtrl	Connects to function block AlrmHdlg. Refer also to structure parameter description i_stAlrmCtrl *(see page 149)*.
i_xProgRstAlrm	BOOL	External reset of alarm.
i_xDoInit	BOOL	TRUE: Initial state is enabled FALSE: Initial state is disabled

i_stEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Contains ● Mode selection ● Manual command ● Conveyor direction command ● Equipment ID, type ID, and zone ID ● Alarm reset
xManCmdVldt	BOOL	Validates the manual movement

i_stOtbOput (Data Type: stOpcnvOtbIput)

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Displays the status of the Conveyor sensors
xOputRele	BOOL	TRUE: Pallet is removed from system FALSE: Disabled
xOputAckn	BOOL	Acknowledge signal Output Conveyor
stMotStat	stMotStat	Displays the status of the connected motor
stMotFbck	stMotFbck	Information of the motor feedback
stRemMod	stRemMod	Displays the remote mode (local or maintenance)

i_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

i_stTrckStat (Data Type: stTrckStat)

Structure Parameter	Data Type	Description
xInfoOk	BOOL	Indicates if tracking data are valid as result from the End sign check.
xFuncOk	BOOL	Indicates the status of function block TrckSimp . TRUE: Function Block is ready for operation.

i_stAlrmCtrl (Data Type: stAlrmCtrl)

Structure Parameter	Data Type	Description
xRstAlrm	BOOL	Resets alarm from function block AlrmHdlg in zone
xAlrmRele	BOOL	TRUE: Overall alarm is not active in zone FALSE: Overall alarm is active in zone

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stEqupStat	stEqupStat	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to the below structure parameter description q_stEqupStat *(see page 150)*.
q_stStmStatSimp	stStmStatSimp	Connects to function block TrckSimp. Refer also to structure parameter description q_stStmStatSimp *(see page 151)* .
q_stOtbOput	stOpcnvOtbOput	Elements of this structure must be linked to physical outputs to control the equipment. Refer also to structure parameter description q_stOtbOput *(see page 152)* .
q_stInit	stConvInit	Indicates the parameterized values of timer and the enable/disable function. Refer also to structure parameter description q_stInit *(see page 152)* .
q_stDevAlrm	stDevAlrm	Connects to function block AlrmHdlg. Refer also to structure parameter description q_stDevAlrm *(see page 153)* .
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stEqupStat (Data Type: stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID, to control all equipments of the zone
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of the status of the mode
stStmStatCmpx	stStmStatCmpx	Information of the status of the complex state machine
stConvSenFltr	stConvSenFltr	Information of the filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSen	Information of the filtered sensors status from the Turntable [not used in Conveyor, all structure elements on FALSE state]

Structure Parameter	Data Type	Description
stErcSenFltr	stErcSen	Information of the sensors from the ERC position (up or down) [not used in Conveyor, all structure elements on FALSE state]
dwMot1HrsofOp	DWORD	Information of the operation hour of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hour of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback [in Conveyor all FALSE]
dwMot3HrsofOp	DWORD	Information of the operation hour of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback [in Conveyor all FALSE]
xAlrm	BOOL	Indicates that an alarm is active
xAlrt	BOOL	Indicates that an alert is active
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light).

q_stStmStatSimp (Data Type: stStmStatSimp)

Structure Parameter	Data Type	Description
xEmty	BOOL	Status information is empty
xLdng	BOOL	Status information is loading
xLded	BOOL	Status information is loaded
xUldg	BOOL	Status information is unloading
xFwd	BOOL	State machine is decided to transfer in forward direction
xRev	BOOL	State machine is decided to transfer in in reverse direction
xManModinPrgs	BOOL	Manual mode is in progress, selected by user
xStagFnsh	BOOL	Handshake signal stage is complete to indicate that tracking data is assumed
xCycRst	BOOL	Cycle restart is active
xCyclRstInPrgs	BOOL	Cycle restart is in progress
xDolnitInPrgs	BOOL	Indicates that the Go to Intialize Mode is in progress
xRst	BOOL	Alarm reset information for the relevant function block TrckSimp or TrckCmpx
xStmAuto	BOOL	Status information is in automatic mode

Structure Parameter	Data Type	Description
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

q_stOtbOput (Data Type: stOpcnvOtbOput)

Structure Parameter	Data Type	Description
stMotCmd	stMotCmd	Motor control commands
xAlrm	BOOL	Alarm information to switch on the local alarm lamp
stBecn	stBecn	Indicates the status of the output unloading process

q_stInit (Data Type: stConvInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoadCtrl	Initial values for the internal function block Conveyor Loading Control
stUnldCtrl	stInitOfUnldCtrl	Initial values for the internal function block Conveyor Unloading Control
stStm	stStmInit	Initial values for the internal function block Conveyor State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeSimp	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

q_stDevAlrm (Data Type: stDevAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupID	WORD	Equipment ID [user defined]
wEqupGrpID	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
ebyEqupType	EqupType	Equipment type [system defined]
awAlrmStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alarms See also Alarm Words of Conveyor, Input, Output *(see page 46)*.
awAlrtStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate the alerts See also Alert Words of Conveyor, Input, Output *(see page 61)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
`iq_stHdskNMnus1`	`stHdskIntf`	Connects to UME180°. Refer also to the below structure parameter description `stHdsNMinus1` *(see page 154)*.

`stHdsNMinus1` (Data Type: `stHdskIntf`)

Structure Parameter	Data Type	Description
`stFwd`	`stHdsk`	Handshake signals if material flow is in forward direction.
`stRev`	`stHdsk`	Handshake signals if material flow is in reverse direction.

Section 12.4
Interfaces

At a Glance

This section gives an overview how the `Output` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
Output Interfaces	156
Linking the Function Blocks	157

Output Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Handshaking	Information needs to be exchanged between 2 neighboring pieces of equipment in order to manage the material flow.

Optional interface connections are related to:

Tracking	Supervision of the material flow by managing and validating the tracking data.
HMI operation	Together with a Deskzone AFB, a group of equipment can be operated by and monitored using an HMI (e.g., Magelis HMI).
Alarm and alert monitoring	In conjunction with the `AlrmHdlg` FB, the alarms and alerts signaled by the Equipment function block can be logged and summarized.

Linking the Function Blocks

Handshake Interface

To allow the material flow from one equipment to another, handshake information between these 2 equipment function blocks need to be shared. This interface is realized by using a variable of type `stHdskItf` linked to the In_Output pins of the concerned Equipment function blocks. For more detailed information refer to the chapter *Handshaking – Information Dataflow Between Conveying Equipments* *(see page 73)*.

On one hand, the `Output` FB needs to read from the structure the relevant information from the neighboring equipment, on the other hand the `Output` FB has to write its own status in the same structure (handshake variable). In order to share this information by using the identical structure, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Handshake interface with preceding Conveyor	`iq_stHdskNMnus1`	`iq_stHdskNPlus1`	`Conveyor`
Handshake interface with preceding Input-Conveyor	`iq_stHdskNMnus1`	`iq_stHdskNPlus1`	`Input`
Handshake interface with a Turntable or ERC. Seen from the position of the Turntable or ERC, the Output-Conveyor is located on station 90, 180, 270 or 360 degree.	`iq_stHdskNMnus1`	`iq_stHdsk90Deg`/ `iq_stHdsk180Deg` / `iq_stHdsk270Deg` / `iq_stHdsk360Deg`	`TurnTable` / `ERC`

The following screen capture shows, how the interfacing can be done in SoMachine on example of an `Output` and a `Conveyor` function block:

Tracking Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one `Output` AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from `Output` to `TrckSimp` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information from `TrckSimp` AFB about the state and the validity of the tracking data.	`i_stTrckStat`	`q_stStat`	`TrckSimp`
Information to the `TrckSimp` AFB about the state for managing the tracking data transfer.	`q_stStmStatSimp`	`i_ststmStatSimp`	

Deskzone Interface

The Deskzone FB (`DeskzoneSimp` and `DeskzoneCmpx`) manages a group of equipments by gathering and providing the necessary information from the equipment and to the equipment. This information will be shared with the HMI (e.g. Magelis HMI) in order to be able to provide the status information to the HMI and to receive the command from the HMI to operate the Conveying line.

The Deskzone FB uses arrays of structures to exchange the data between `Output` and Deskzone function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Commands to control the Output-Conveyor by the operator via the Deskzone AFB what is linked to a HMI.	`i_stEqupCmd`	`q_astEqupCmd[x]`	`DeskzoneSimp` / `DeskzoneCmpx`
State information for monitoring by the operator via the Deskzone AFB what is linked to a HMI.	`q_stEqupStat`	`iq_astEqupStat[x]`	

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 43)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `Output` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the `Output` AFB from `AlrmHdlg` AFB.	`i_stAlrmCtrl`	`q_astAlrmCtrl[x]`	`AlarmHdlg`
`Output` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert lds.	`q_stDevArm`	`i_astEqupAlrm[x].stDevAlrm`	

Part III
Deskzone

At a Glance

This part describes the function blocks `DeskzoneCmpx` and `DeskzoneSimp` in the Conveying Library system.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
13	DeskzoneCmpx: Interface to HMI	165
14	DeskzoneSimp: Interface to HMI	179

Chapter 13
DeskzoneCmpx: Interface to HMI

At a Glance

This chapter describes the functionality and implementation of the `DeskzoneCmpx` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
13.1	Functional Overview	166
13.2	Function Block Description	167
13.3	Pin Description	168
13.4	Interfaces	174

Section 13.1
Functional Overview

Functional Overview

Functional Description

The function block `DeskzoneCmpx` is part of the Conveying library with the following features:

- Provides status information to the Deskzone HMI
- Receives control information from the Deskzone HMI
- Receives status information from the controlled UME
- Provides control information to the controlled UME
- Receives the tracking data from the function block `TrckCmpx` or `TrckSimp`
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

Section 13.2
Function Block Description

DeskzoneCmpx Function Block

Pin Diagram

Section 13.3
Pin Description

At a Glance

This section gives the pin description of the `DeskzoneCmpx` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	169
Output Pin Description	170
Input/Output Pin Description	171

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_wZoneId	WORD	Zone ID [user defined]
i_tLifeCtrl	TIME	Parameterizes between 1 and 3 s
i_xManMode	BOOL	TRUE: Manual mode in zone 1 is enabled FALSE: Manual mode in zone 1 is disabled
i_xAutoMode	BOOL	TRUE: Automatic mode in zone 1 is enabled FALSE: Automatic mode in zone 1 is disabled
i_xManCmdVldt	BOOL	TRUE: Validates the manual move command FALSE: Disabled

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_xBusy	BOOL	TRUE: Function block is enabled FALSE: Function block is disabled
q_astEqupCmd	ARRAY [0...24] of stEqupCmd	Connects to Equipment AFBs. Refer also to the below structure parameter description q_astEqupCmd *(see page 170)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert
q_stAlrm	stDeskAlrm	Connects to function block AlrmHdlg. Refer also to the below structure parameter description .q_stAlrm *(see page 170)*

q_astEqupCmd (Data Type: ARRAY [0...24] of stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Equipment control information of mode, manual move commands and initial direction
xManCmdVldt	BOOL	Validates the manual move commands

q_stAlrm (Data Type: stDeskAlrm)

Structure Parameter	Data Type	Description
wAlrmStat	WORD	Alarm word of the function block See also Alarm Words of Deskzone Complex *(see page 48)*.
wAlrtStat	WORD	Alert word of the function block See also Alert Word of Deskzone (Complex, Simple *(see page 63)*).
wZoneId	WORD	Zone ID of the equipment where the alarm / alert-event occurred

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
iq_astHmiCtrlPopUp	ARRAY [0...2] of stHmiCtrlPopUp	Connects to HMI. Refer also to the below structure parameter description iq_astHmiCtrlPopUp *(see page 171)*.
iq_astEqupStat	ARRAY [0...24] of stEqupStat	Connects to function block Equipment. Refer also to the below structure parameter description iq_astEqupStat *(see page 172)*.
iq_astTrckHmiStat	ARRAY [0...24] of stTrckHmiStat	Connects to function block TrckSimp or TrckCmpx. Refer also to the below structure parameter description iq_astTrckHmiStat *(see page 173)*.

iq_astHmiCtrlPopUp (Data Type: ARRAY [0...2] of stHmiCtrlPopUp)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Equipment control information of mode, manual move commands and initial direction
stEqupStat	stEqupStat	Equipment status information
xLifeBit	BOOL	Toggling life bit indicates the data exchange to the HMI

iq_astEqupStat (Data Type: ARRAY [0...24] of stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Group ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of the status of the mode
stStmStatCmpx	stStmStatCmpx	Information of the status of the complex state machine
stConvSenFltr	stConvSenFltr	Information of the filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSen	Information of the filtered sensors status from the Turntable
stErcSenFltr	stErcSen	Information of the sensors from the ERC position [up or down]
dwMot1HrsOfOp	DWORD	Information of the operation hour of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hour of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback
dwMot3HrsofOp	DWORD	Information of the operation hour of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light). Used only by Input and Output AFB.
xAlrm	BOOL	Indicates that an alarm is active
xAlrt	BOOL	Indicates that an alert is active

iq_astTrckHmiStat (Data Type: ARRAY [0...24] of stTrckHmiStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID of the function block TrckSimp or TrckCmpx
wEqupId	WORD	Equipment ID of the function block TrckSimp or TrckCmpx
ebyEqupType	EqupType	Equipment type of the function block TrckSimp or TrckCmpx
stDestPntr	stDestPntr	Information of the Word and Nibble index where the destination address is stored
awTrckData	ARRAY [0...49] of WORD	Tracking data field

Section 13.4
Interfaces

At a Glance

This section gives an overview how the `DeskzoneCmpx` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
DeskzoneCmpx Interfaces	175
Linking the Function Blocks	176

DeskzoneCmpx Interfaces

Overview

The graphic below demonstrates the principle which is followed to integrate the `DeskzoneCmpx` FB into the user's application.

Linking the Function Blocks

HMI

The main function of the Deskzone FB is, to simplify the interface between the operator and the Equipment FBs.

This interface implements the following features:
- Forwarding the status information from the connected Equipment AFBs to the HMI.
- Forwarding the control commands from the HMI to the connected Equipment AFBs.

Therefore the Deskzone FB provides pins to interface an HMI.

Description	Pin on AFB	Needs to be Linked to...
Array of 3 structures to control up to 3 Equipment AFBs at a time. Each structure forwards: ● State information of the selected equipment to the HMI. ● Control commands from the operator to the selected equipment.	`iq_astHmiCtrlPopUp`	Usually linked to a HMI (e.g. Magelis operator panel)
Command signal to switch all connected Equipment AFBs to manual operation mode.	`i_xManMode`	Usually linked to an operator push button
Command signal to switch all connected Equipment AFBs to automatic operation mode.	`i_xAutoMode`	Usually linked to an operator push button
Command signal validates the preset commands for manual motor operations on the Equipment AFBs. Preset commands means the commands linked to the pin `iq_astHmiCtrlPopUp`.	`i_xManCmdVldt`	Usually linked to an operator push button

Equipment Interface

As described in the *HMI* section the Deskzone FB is forwarding the control and the state information between the HMI and the Equipment AFBs. The pins which are provided to link the Equipment AFBs with the Deskzone FB are listed in the following table.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State information from the Equipment AFBs, to forward them to the HMI on demand.	`iq_astEqupStat[x]`	`q_astEqupStat[x]`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`
Control commands which have been received from the HMI forwarded to the demanded Equipment AFBs.	`q_astEqupCmd[x]`	`i_stEqupCmd[x]`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`

Tracking Data Interface

The `DeskzoneCmpx` FB provides one additional feature. It checks the consistency of the defined equipment parameter within the application. The `DeskzoneCmpx` FB crosschecks the Equipment parameter of the Equipment FB and the associated Tracking FBs for plausibility

These parameters are:
- `wZoneId`
- `wEqupId`
- `ebyEqupType`

Therefore the Tracking FBs must be linked to the `DeskzoneCmpx` FB in addition to the Equipment FBs.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
`DeskzoneCmpx` AFB does read only the Equipment parameter from the connected Tracking FBs to compare them with appropriated Equipment FB.	`iq_astTrckStat[x]`	`iq_stTrckHmiStat`	`TrckSimp` / `TrckCmpx`

Alarm Handling Interface

In case of detecting inconsistent equipment parameter or instable communication with an HMI, the Deskzone FB generates alarms and alerts. These are signaled to the `AlrmHdlg` FB for logging and summarizing.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Deskzone AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`q_stAlrm`	`i_stDeskAlrm`	`AlarmHdlg`

Chapter 14
DeskzoneSimp: Interface to HMI

At a Glance

This chapter describes the functionality and implementation of the `DeskzoneSimp` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
14.1	Functional Overview	180
14.2	Function Block Description	181
14.3	Pin Description	182
14.4	Interfaces	187

Section 14.1
Functional Overview

Functional Overview

Functional Description

The function block `DeskzoneSimp` is part of the Conveying library with the following features:

- Provides status information to the Deskzone HMI
- Receives control information from the Deskzone HMI
- Receives status information from the controlled equipment
- Provides control information to the controlled equipment
- Handles and generates alerts and provides the information to the function block `AlrmHdlg`

NOTE: In comparison with the `DeskzoneCmpx`, this function block does not support the tracking functionalities.

Section 14.2
Function Block Description

DeskzoneSimp Function Block

Pin Diagram

Section 14.3
Pin Description

At a Glance

This section gives the pin description of the `DeskzoneSimp` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	183
Output Pin Description	184
Input/Output Pin Description	185

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_wZoneId	WORD	Zone ID [user defined]
i_tLifeCtrl	TIME	Time to monitor the LifeBit toggled by the HMI. Range is 1 s...3 s. Default value: 1 s
i_xManMode	BOOL	A rising edge of this input sends a request for switching to Manual mode to all connected equipment function blocks.
i_xAutoMode	BOOL	A rising edge of this input sends a request for switching to Automatic mode to all connected equipment function blocks.
i_xManCmdVldt	BOOL	TRUE: Validates the manual move command FALSE: Disabled

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_xBusy	BOOL	TRUE: Function block is enabled FALSE: Function block is disabled
q_astEqupCmd	ARRAY [0...24] of stEqupCmd	Connects to UME. Refer also to the below structure parameter description q_astEqupCmd *(see page 184)*
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert
q_stAlrm	stDeskAlrm	Connects to function block AlrmHdlg. Refer also to the below structure parameter description q_stAlrm *(see page 184)*.

q_astEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupCmd	Equipment control information of mode, manual move commands and initial direction
xManCmdVldt	BOOL	Validates the manual move commands

q_stAlrm (Data Type: stDeskAlrm)

Structure Parameter	Data Type	Description
wAlrmStat	WORD	Alarm word of the function block See also Alarm Words of Deskzone Simple *(see page 49)*.
wZoneId	WORD	Zone ID of the equipment where the alarm / alert-event occurred
wAlrtStat	WORD	Alert word of the function block See also Alert Word of Deskzone (Complex, Simple *(see page 63)*).

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
iq_astHmiCtrlPopUp	ARRAY [0...2] of stHmiCtrlPopUp	Connects to HMI. Refer also to the below structure parameter description iq_astHmiCtrlPopUp *(see page 185)*
iq_astEqupStat	ARRAY [0...24] of stEqupStat	Connects to function block Equipment. Refer also to the below structure parameter description iq_astEqupStat *(see page 185)*

iq_astHmiCtrlPopUp (Data Type: ARRAY [0...2] of stHmiCtrlPopUp)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Equipment control information of mode, manual move commands and initial direction
stEqupState	stEqupStat	Equipment status information
xLifeBit	BOOL	Toggling Life Bit (Heartbeat) must be generated on the HMI. Indicates the established connection between the HMI and the controller and is monitored by the Deskzone AFB.

iq_astEqupStat (Data Type: ARRAY [0...24] stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Group ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of the status of the mode
stStmStatCmpx	stStmStatCmpx	Information of the status of the complex state machine
stConvSenFltr	stConvSenFltr	Information of the filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSen	Information of the filtered sensors status from the Turntable

Structure Parameter	Data Type	Description
stErcSenFltr	stErcSen	Information of the sensors from the ERC position [up or down]
dwMot1HrsofOp	DWORD	Information of the operation hour of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hour of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback
dwMot3HrsofOp	DWORD	Information of the operation hour of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light). Used only by Input and Output AFB.
xAlrm	BOOL	Indicates that an alarm is active
xAlrt	BOOL	Indicates that an alert is active

Section 14.4
Interfaces

At a Glance

This section gives an overview how the `DeskzoneSimp` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
DeskzoneSimp Interfaces	188
Linking the Function Blocks	189

DeskzoneSimp Interfaces

Overview

The graphic below demonstrates the principle which is followed to integrate the `DeskzoneSimp` FB into the user's application.

Linking the Function Blocks

HMI

The main function of the Deskzone FB is, to simplify the interface between the operator and the Equipment FBs.

This interface implements the following features:
- Forwarding the status information from the connected Equipment AFBs to the HMI.
- Forwarding the control commands from the HMI to the connected Equipment AFBs.

Therefore the Deskzone FB provides pins to interface an HMI.

Description	Pin on AFB	Needs to be Linked to...
Array of 3 structures to control up to 3 Equipment AFBs at a time. Each structure forwards: ● State information of the selected equipment to the HMI. ● Control commands from the operator to the selected equipment.	`iq_astHmiCtrlPopUp`	Usually linked to a HMI (e.g. Magelis operator panel)
Command signal to switch all connected Equipment AFBs to manual operation mode.	`i_xManMode`	Usually linked to an operator push button
Command signal to switch all connected Equipment AFBs to automatic operation mode.	`i_xAutoMode`	Usually linked to an operator push button
Command signal validates the preset commands for manual motor operations on the Equipment AFBs. Preset commands means the commands linked to the pin `iq_astHmiCtrlPopUp`.	`i_xManCmdVldt`	Usually linked to an operator push button

Equipment Interface

As described in the *HMI* section the Deskzone FB is forwarding the control and the state information between the HMI and the Equipment AFBs. The pins which are provided to link the Equipment AFBs with the Deskzone FB are listed in the following table.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State information from the Equipment AFBs, to forward them to the HMI on demand.	`iq_astEqupStat[x]`	`q_astEqupStat[x]`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`
Control commands which have been received from the HMI forwarded to the demanded Equipment AFBs.	`q_astEqupCmd[x]`	`i_stEqupCmd[x]`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`

Alarm Handling Interface

In case of detecting inconsistent equipment parameter or instable communication with an HMI, the Deskzone FB generates alarms and alerts. These are signaled to the `AlrmHdlg` FB for logging and summarizing.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Deskzone AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`q_stAlrm`	`i_stDeskAlrm`	`AlarmHdlg`

Part IV
ERC

Chapter 15
ERC: Transfer of Goods in 4 Directions

At a Glance

This chapter describes the functionality and implementation of the ERC function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
15.1	Functional and System Overview	194
15.2	Function Block Description	197
15.3	Pin Description	198
15.4	Interfaces	207

Section 15.1
Functional and System Overview

At a Glance

This section describes the functional and system overview of the (Eccentric Roller Conveyor) ERC function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Functional Overview	195
System Overview	196

Functional Overview

Functional Description

The ERC function block is part of the Conveying library with the following features:

- Operation modes: Automatic, Manual and Local
- Movement of goods in forward or reverse direction on the Conveyor
- Lifts eccentric up or down for movement of goods on the Chain or on the Conveyor
- Control of up to 3 motors that can be adjusted during initialization (disables automatically when alarm is generated)
- Receives control information from function block `DeskzoneSimp` or `DeskzoneCmpx`
- Provides status information to function block `DeskzoneSimp` or `DeskzoneCmpx`
- Receives status information and destination address from the function block `TrckCmpx` if Tracking is enabled
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

System Overview

System View

M1 Conveyor motor (roller section)
M2 Conveyor motor (chain section)
M3 Transfer motor (vertical movement)
B1 Pallet presence sensor forward
B2 Pallet presence sensor reverse
B3 Pallet speed change sensor forward
B4 Pallet speed change sensor reverse
B5 180° and 360° position sensor
B6 90° and 270° position sensor

Section 15.2
Function Block Description

ERC Function Block

Pin Diagram

Section 15.3
Pin Description

At a Glance

This section gives the pin description of the ERC function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	199
Output Pin Description	202
Input/Output Pin Description	206

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stEqupCmd	stEqupCmd	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to structure parameter description i_stEqupCmd *(see page 199)*.
i_stOtbIput	stErcOtbIput	Connects to OTB where the content of this structure provides the status information of the equipment (like drive and sensors). Refer also to structure parameter description i_stOtbOput *(see page 200)*.
i_stInit	stErcInit	Initializes the values of timer and the enable/ disable functions during the start up phase. Refer also to structure parameter description i_stInit *(see page 200)*.
i_xFbAvai	BOOL	Indicates the availability of Fieldbus [depends on the start-up time].
i_xFbNodeOk	BOOL	Indicates the availability of Fieldbus Node.
i_stDest	stDest	Connects to function block TrckCmpx. Refer also to structure parameter description i_stDest *(see page 201)*.
i_xEmgy	BOOL	TRUE: No local stop FALSE: Local stop
i_stAlrmCtrl	stAlrmCtrl	Connects to function block AlrmHdlg. Refer also to structure parameter description i_stAlrmCtrl *(see page 201)*.
i_xProgRstAlrm	BOOL	External reset of alarm.
i_xDoInit	BOOL	TRUE: Initial state is enabled FALSE: Initial state is disabled

i_stEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Contains ● Mode selection ● Manual commands ● Conveyor direction command ● Equipment ID, type, and zone ID ● Alarm reset
xManCmdVldt	BOOL	Validates manual movement

i_stOtbOput (Data Type: stErcOtbIput)

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Displays the status of Conveyor sensors
stErcSen	stErcSen	Displays the status of ERC sensors
stMot1Stat	stMotStat	Displays the status of connected motor1
stMot2Stat	stMotStat	Displays the status of connected motor2
stMot3Stat	stMotStat	Displays the status of connected motor3
stMot1Fbck	stMotFbck	Information of the motor1 feedback
stMot2Fbck	stMotFbck	Information of the motor2 feedback
stMot3Fbck	stMotFbck	Information of the motor3 feedback
stRemMod	stRemMod	Displays remote mode (local or maintenance)

i_stInit (Data Type: stErcInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoad Ctrl	Initial values for the internal function block Turntable Loading Control
stUnldCtrl	stInitOfUnld Ctrl	Initial values for the internal function block Turntable Unloading Control
stErcTtblStm	stErcTtblStm Init	Initial values for the internal function block Turntable State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
stVertMovCtrl	stInitOfVertMov Ctrl	Initial values for the internal function block Vertical Move Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeCmpx	Equipment type [system]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID, to control all equipments of the zone
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled
stConfofHdskSwap	stConfOfHdsk-Swap	Initial values for configuration of swapping the handshaking signals based on flow direction

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

i_stDest (Data Type: stDest)

Structure Parameter	Data Type	Description
stTrckStat	stTrckStat	Information of tracking data status
wDestPntrAddr	WORD	Destination address for unloading

i_stAlrmCtrl (Data Type: stAlrmCtrl)

Structure Parameter	Data Type	Description
xAlrmRele	BOOL	TRUE: No overall alarm is active in zone FALSE: Overall alarm is active in zone
xRstAlrm	BOOL	Resets alarm in zone

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stEqupStat	stEqupStat	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to the below structure parameter description q_stEqupStat *(see page 202)*
q_stStmStatCmpx	stStmStatCmpx	Connect to function block TrckCmpx. Refer also to structure parameter description q_stStmStatCmpx *(see page 203)*
q_stOtbOput	stErcOtbOput	Elements of this structure must be linked to physical outputs to control the equipment. Refer also to structure parameter description q_stOtbOput *(see page 204)*
q_stInit	stErcInit	Initializes the values of timer and the enable/disable functions during the start up phase. Refer also to structure parameter description q_stInit *(see page 204)*
q_stDevAlrm	stDevAlrm	Connects to function block AlrmHdlg. Refer also to structure parameter description q_stDevAlrm *(see page 205)*
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stEqupStat (Data Type: stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment Group ID, to control all equipments of the zone]
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of the status of the mode
stStmStatCmpx	stStmStatCmpx	Information of the status of the complex state machine
stConvSenFltr	stConvSenFltr	Information of the filtered sensors status from the Conveyor
stTtblSenFltr	stTtblSenFltr	Information of the filtered sensors status from the Turntable [not used in Conveyor, all structure elements on FALSE state]

Structure Parameter	Data Type	Description
stErcSenFltr	stErcSenFltr	Information of the sensors from the ERC position (up or down) [not used in Conveyor, all structure elements on FALSE state]
dwMot1HrsofOp	DWORD	Information of the operation hours of motor1
stMot1Fbck	stMotFbck	Information of the motor1 feedback
dwMot2HrsofOp	DWORD	Information of the operation hours of motor2
stMot2Fbck	stMotFbck	Information of the motor2 feedback [not used in Conveyor, all structure elements on FALSE state]
dwMot3HrsofOp	DWORD	Information of the operation hours of motor3
stMot3Fbck	stMotFbck	Information of the motor3 feedback [not used in Conveyor, all structure elements on FALSE state]
xAlrm	BOOL	Indicates that an alarm is active
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light). Used only by Input and Output AFB.
xAlrt	BOOL	Indicates that an alert is active

q_stStmStatCmpx (Data Type: stStmStatCmpx)

Structure Parameter	Data Type	Description
xEmty	BOOL	Status information is empty
xLdng	BOOL	Status information is loading
xLded	BOOL	Status information is loaded
xUldg	BOOL	Status information is unloading
xFwd	BOOL	Motor is running in forward direction
xRev	BOOL	Motor is running in reverse direction
xManModInPrgs	BOOL	Manual mode is in progress (selected by user)
xStagFnsh	BOOL	Handshake signal stage is complete and tracking data is assumed
xCyclRst	BOOL	Restart cycle is active
xCyclRstInPrgs	BOOL	Cycle Restart is in progress
xDoInitInPrgs	BOOL	Go to initialization mode is in progress
xRst	BOOL	Alarm reset information for the relevant function block TrckSimp or TrckCmpx
xStmAuto	BOOL	Status information is in automatic mode
xConvInUnldPos	BOOL	Conveyor is in unloading position
xPosChg	BOOL	Position is changed (ERC up/down)

Structure Parameter	Data Type	Description
wSrcAddr	WORD	Source address of the next goods
wDestAddr	WORD	Destination Address
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

q_stOtbOput (Data Type: stErcOtbOput)

Structure Parameter	Data Type	Description
stMot1Cmd	stMotCmd	Motor1 control commands
stMot2Cmd	stMotCmd	Motor2 control commands
stMot3Cmd	stMotCmd	Motor3 control commands
xAlrm	BOOL	Alarm information switches on the local alarm lamp

q_stInit (Data Type: stErcInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial value for the internal function block Sensor Control
stLoadCtrl	stInitOfLoad Ctrl	Initial value for the internal function block Turntable Loading Control
stUnldCtrl	stInitOfUnld Ctrl	Initial value for the internal function block Turntable Unloading Control
stErcTtblStm	stErcTtblStm Init	Initial values for the internal function block Turntable State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
stVertMovCtrl	stInitOfVertMov Ctrl	Initial values for the internal function block Vertical Move Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeCmpx	Equipment type [system defined]
wEqupGrpId	WORD	Equipment Group ID, to control all equipments of the zone
wEqupId	WORD	Equipment ID [user defined]
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled
stConfofHdskSwap	stConfOfHdsk Swap	Initial values for configuration of swapping handshaking signals based on the flow direction

q_stDevAlrm (Data Type: stDevAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
wEqupId	WORD	Equp ID of the equipment where the alarm/alert-event occurred
ebyEqupType	EqupType	Equipment type [system defined]
awAlrmStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate alarms See also Alarm Words of ERC *(see page 50)*.
awAlrtStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate alerts See also Alert Word of ERC *(see page 64)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
`iq_stHdsk360Deg`	`stHdskIntf`	Connects to UME360°. Refer also to the below structure parameter description `iq_stHdsk`*xxx*`Deg` *(see page 206)*.
`iq_stHdsk90Deg`	`stHdskIntf`	Connects to UME90°. Refer also to the below structure parameter description `iq_stHdsk`*xxx*`Deg` *(see page 206)*.
`iq_stHdsk180Deg`	`stHdskIntf`	Connects to UME180°. Refer also to the below structure parameter description `iq_stHdsk`*xxx*`Deg` *(see page 206)*.
`iq_stHdsk270Deg`	`stHdskIntf`	Connects to UME270°. Refer also to the below structure parameter description `iq_stHdsk`*xxx*`Deg` *(see page 206)*.

`iq_stHdsk`*xxx*`Deg` (Data Type: `stHdskIntf`)

Structure Parameter	Data Type	Description
`stFwd`	`stHdsk`	Handshake signals if the material flow is in forward direction.
`stRev`	`stHdsk`	Handshake signals if the material flow is in reverse direction.

Section 15.4
Interfaces

At a Glance

This section gives an overview how the ERC function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
ERC Interfaces	208
Linking the Function Blocks	209

ERC Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Handshaking	Information needs to be exchanged between 2 neighboring pieces of equipment in order to manage the material flow.

Optional interface connections are related to:

Tracking	Supervision of the material flow by managing and validating the tracking data.
HMI operation	Together with a Deskzone AFB, a group of equipment can be operated by and monitored using an HMI (e.g., Magelis HMI).
Alarm and alert monitoring	In conjunction with the `AlrmHdlg` FB, the alarms and alerts signaled by the Equipment function block can be logged and summarized.

Linking the Function Blocks

Handshake Interface

To allow the material flow from one equipment to another, handshake information between these 2 equipment function blocks need to be shared. This interface is realized by using a variable of type `stHdskItf` linked to the In_Output pins of the concerned Equipment function blocks. For more detailed information refer to the chapter *Handshaking – Information Dataflow Between Conveying Equipments* *(see page 73)*.

On one hand, the ERC FB needs to read from the structure the relevant information from the neighboring equipment, on the other hand the ERC FB has to write its own status in the same structure (handshake variable). In order to share this information by using the identical structure, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Handshake interface with preceding Conveyor	`iq_stHdsk90Deg/ iq_stHdsk180Deg / iq_stHdsk270Deg / iq_stHdsk360Deg`	`iq_stHdskNPlus1 / iq_stHdskNMnus1`	`Conveyor`
Handshake interface with neighbored Input-Conveyor		`iq_stHdskNPlus1`	`Input`
Handshake interface with neighbored Output-Conveyor		`iq_stHdskNMnus1`	`Output`
Handshake interface with a Turntable or ERC. Seen from the position of the Turntable or ERC, the ERC is located on station 90, 180, 270 or 360 degree.		`iq_stHdsk90Deg/ iq_stHdsk180Deg / iq_stHdsk270Deg / iq_stHdsk360Deg`	`TurnTable`
			`ERC`

The following screen capture shows, how the interfacing can be done in SoMachine on example of an ERC and a `Conveyor` function block:

Tracking Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one ERC AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from ERC to `TrckCmpx` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information from `TrckCmpx` AFB about the state and the validity of the tracking data and the destination address read out of them.	`i_stDest`	`q_stStat`	`TrckCmpx`
Information to the `TrckCmpx` AFB about the state for managing the tracking data transfer.	`q_stStmStatCmpx`	`i_ststmStatCmpx`	

Deskzone Interface

The Deskzone FB (`DeskzoneSimp` and `DeskzoneCmpx`) manages a group of equipments by gathering and providing the necessary information from the equipment and to the equipment. This information will be shared with the HMI (e.g. Magelis HMI) in order to be able to provide the status information to the HMI and to receive the command from the HMI to operate the Conveying line.

The Deskzone FB uses arrays of structures to exchange the data between ERC and Deskzone function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Commands to control the ERC by the operator via the Deskzone AFB what is linked to an HMI.	`i_stEqupCmd`	`q_astEqupCmd[x]`	`DeskzoneSimp` / `DeskzoneCmpx`
State information for monitoring by the operator via the Deskzone AFB what is linked to an HMI.	`q_stEqupStat`	`iq_astEqupStat[x]`	

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 43)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between ERC and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the ERC AFB from `AlrmHdlg` AFB.	`i_stAlrmCtrl`	`q_astAlrmCtrl[x]`	`AlarmHdlg`
ERC AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`q_stDevArm`	`i_astEqupAlrm[x].stDevAlrm`	

Part V
Turntable

Chapter 16
TurnTable: Transfer of Goods in 4 Directions

At a Glance

This chapter describes the functionality and implementation of the `TurnTable` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
16.1	Functional and System Overview	218
16.2	Function Block Description	221
16.3	Pin Description	222
16.4	Interfaces	231

Section 16.1
Functional and System Overview

At a Glance

This section describes the functional and system overview of the `TurnTable` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Functional Overview	219
System Overview	220

Functional Overview

Functional Description

The `TurnTable` function block is part of the Conveying library with the following features:

- Operation Modes: Automatic, Manual and Local
- Moves goods in forward or reverse direction
- Turns table left or right in forward direction and back in reverse direction
- Receives control information from the function block `DeskzoneSimp` or function block `DeskzoneCmpx`
- Provides status information to the function block `DeskzoneSimp` or function block `DeskzoneSimp`
- Receives status information from the function block `TrckCmpx` if Tracking is enabled
- Receives the destination address from the function block `TrckCmpx` if Tracking is enabled
- Handles and generates alarms and alerts and provides the information to the function block `AlrmHdlg`

System Overview

System View

B1 Pallet presence sensor forward
B2 Pallet presence sensor reverse
B3 Pallet speed change sensor forward
B4 Pallet speed change sensor reverse
B5 Position 1 interlock sensor
B6 Position 2 interlock sensor
B7 180° and 360° position sensor
B8 180° and 360° position speed change sensor
B9 90° and 270° position speed change sensor
B10 90° and 270° position sensor

Section 16.2
Function Block Description

TurnTable Function Block

Pin Diagram

Section 16.3
Pin Description

At a Glance

This section gives the pin description of the `TurnTable` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	223
Output Pin Description	226
Input/Output Pin Description	230

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stEqupCmd	stEqupCmd	Connects to function block DeskzoneSimp or DeskzomeCmpx. Refer also to structure parameter description i_stEqupCmd *(see page 223)*.
i_stOtbIput	stTtblOtbIput	Connects to OTB where the content of this structure provides the status information of the equipment (like drive and sensors). Refer also to structure parameter description i_stOtbOput *(see page 224)*.
i_stInit	stTblInit	Initializes values of timer and enables/disables the function during start up phase. Refer also to structure parameter description i_stInit *(see page 224)*.
i_xFbAvai	BOOL	Indicates the availability of Fieldbus [depends on the start-up time].
i_xFbNodeOk	BOOL	Indicates the availability of Fieldbus Node.
i_stDest	stDest	Connects to function block TrckCmpx. Refer also to structure parameter description i_stDest *(see page 225)*.
i_xEmgy	BOOL	TRUE: No local stop FALSE: Local stop
i_stAlrmCtrl	stAlrmCtrl	Connects to function block AlrmHdlg. Refer also the structure parameter description i_stAlrmCtrl *(see page 225)*.
i_xProgRstAlrm	BOOL	External reset of alarm.
i_xDoInit	BOOL	TRUE: Initial state is enabled FALSE: Initial state is disabled

i_stEqupCmd (Data Type: stEqupCmd)

Structure Parameter	Data Type	Description
stEqupHmiCmd	stEqupHmiCmd	Contains • Mode selection • Manual command • Conveyor direction command • Equipment ID, type ID, and zone ID • Alarm reset
xManCmdVldt	BOOL	Validates the manual movement

i_stOtbOput (Data Type: stTtblOtbIput)

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Displays the status of Conveyor sensors
stTtblSen	stTtblSen	Displays the status of Turntable sensors
stMot1Stat	stMotStat	Displays the status of connected motor1
stMot2Stat	stMotStat	Displays the status of connected motor2
stMot1Fbck	stMotFbck	Information of the motor1 feedback
stMot2Fbck	stMotFbck	Information of the motor2 feedback
stRemMod	stRemMod	Displays the remote mode (local or maintenance)

i_stInit (Data Type: stTtblInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSen Ctrl	Initial values for internal function block Sensor Control
stLoadCtrl	stInitOfLoad Ctrl	Initial values for internal function block Turntable Loading Control
stUnldCtrl	stInitOfUnld Ctrl	Initial values for internal function block Turntable Unloading Control
stErcTtblStm	stErcTtblStm Init	Initial values for internal function block Turntable State Machine
stMotCtrl	stInitOfMot Ctrl	Initial values for internal function block Motor Control
stTurnMovCtrl	stInitOfTurn MovCtrl	Initial values for internal function block Turning Move Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeCmpx	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment group ID to control all equipments of the zone]
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled
stConfofHdskSwap	stConfOfHdsk Swap	Initial values for the configuration of swapping the handshaking signals based on the flow direction

NOTE: Normally, once installed, an equipment would be considered to remain as originally designed in terms of its mechanical characteristics. However, if its mechanical characteristics are later modified, a cold start is required in order for the adaptation to be accepted; an initialization is insufficient as the former equipment type will be retained.

i_stDest (Data Type: stDest)

Structure Parameter	Data Type	Description
stTrckStat	stTrckStat	Information of tracking data status
wDestPntrAddr	WORD	Destination address for unloading

i_stAlrmCtrl (Data Type: stAlrmCtrl)

Structure Parameter	Data Type	Description
xAlrmRele	BOOL	TRUE: No overall alarm is active in FALSE: Overall alarm is active in zone
xRstAlrm	BOOL	Resets alarm in zone

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stEqupStat	stEqupStat	Connects to function block DeskzoneSimp or DeskzoneCmpx. Refer also to the below structure parameter description q_stEqupStat *(see page 226)*.
q_stStmStatCmpx	stStmStatCmpx	Connects to function block TrckCmpx. Refer also to structure parameter description q_stStmStatCmpx *(see page 227)*.
q_stOtbOput	stTtblOtbOput	Elements of this structure must be linked to physical outputs to control the equipment. Refer also to structure parameter description q_stOtbOput *(see page 228)*.
q_stInit	stTtblInit	Indicates values of timer and enables/disables function during start up phase. Refer also to structure parameter description q_stInit *(see page 228)*.
q_stDevAlrm	stDevAlrm	Connects to function block AlrmHdlg. Refer also to structure parameter description q_stDevAlrm *(see page 229)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stEqupStat (Data Type: stEqupStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment Group ID, to control all equipments of the zone
wEqupId	WORD	Equipment ID [user defined]
stModStat	stModStat	Information of mode status
stStmStatCmpx	stStmStatCmpx	Information of complex state machine status
stConvSenFltr	stConvSenFltr	Information of filtered sensor status from the Conveyor
stTtblSenFltr	stTtblSen	Information of filtered sensor status from the Turntable [not used in Conveyor, all structure elements on FALSE state]
stErcSenFltr	stErcSen	Information of sensors from the ERC position (up or down) [not used in Conveyor, all structure elements on FALSE state]

Structure Parameter	Data Type	Description
dwMot1HrsofOp	DWORD	Information of motor1 operation hour
stMot1Fbck	stMotFbck	Information of motor1 feedback
dwMot2HrsofOp	DWORD	Information of motor2 operation hour
stMot2Fbck	stMotFbck	Information of the motor2 feedback [not used in Conveyor, all structure elements on FALSE state]
dwMot3HrsofOp	DWORD	Information of motor3 operation hour
stMot3Fbck	stMotFbck	Information of motor3 feedback [not used in Conveyor, all structure elements on FALSE state]
xAlrm	BOOL	Indicates that an alarm is active
stBecn	stBecn	Elements of this structure indicate the state of the signals to control the connected Indicator bank (tower light). Used only by Input and Output AFB.
xAlrt	BOOL	Indicates that an alert is active

q_stStmStatCmpx (Data Type: stStmStatCmpx)

Structure Parameter	Data Type	Description
xEmty	BOOL	Status information is empty
xLdng	BOOL	Status information is loading
xLded	BOOL	Status information is loaded
xUldg	BOOL	Status information is unloading
xFwd	BOOL	Motor is running in forward direction
xRev	BOOL	Motor is running in reverse direction
xManModInPrgs	BOOL	Manual mode is in progress (selected by user)
xStagFnsh	BOOL	Handshake signal stage is complete and tracking data is assumed
xCyclRst	BOOL	Restart cycle is active
xCyclRstInPrgs	BOOL	Cycle Restart is in progress
xDoInitInPrgs	BOOL	Go to initialization mode is in progress
xRst	BOOL	Alarm reset information for the relevant function block TrckSimp or TrckCmpx
xStmAuto	BOOL	Status information is in automatic mode
xConvInUnldPos	BOOL	Conveyor is in unloading position
xPosChg	BOOL	Position is changed (ERC up/down)
wSrcAddr	WORD	Source address of the next goods
wDestAddr	WORD	Destination Address

Structure Parameter	Data Type	Description
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

q_stOtbOput (Data Type: stTtblOtbOput)

Structure Parameter	Data Type	Description
stMot1Cmd	stMotCmd	Motor 1 control commands (Conveyor)
stMot2Cmd	stMotCmd	Motor 2 control commands (Turntable)
xAlrm	BOOL	Alarm Information to switch on the local alarm lamp

q_stInit (Data Type: stTtblInit)

Structure Parameter	Data Type	Description
stSenCtrl	stInitOfSenCtrl	Initial values for the internal function block Sensor Control
stLoadCtrl	stInitOfLoad Ctrl	Initial values for the internal function block Turntable Loading Control
stUnldCtrl	stInitOfUnld Ctrl	Initial values for the internal function block Turntable Unloading Control
stErcTtblStm	stErcTtblStm Init	Initial values for the internal function block Turntable State Machine
stMotCtrl	stInitOfMotCtrl	Initial values for the internal function block Motor Control
stTurnMovCtrl	stInitOfTurnMov Ctrl	Initial values for the internal function block Turning Move Control
wZoneId	WORD	Zone ID [user defined]
ebyEqupType	EqupTypeCmpx	Equipment type [system defined]
wEqupId	WORD	Equipment ID [user defined]
wEqupGrpId	WORD	Equipment Group ID, to control all equipments of the zone]
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled
stConfOfHdskSwap	stConfOfHdsk Swap	Initial values for configuration of swapping the handshaking signals based on flow direction

q_stDevAlrm (Data Type: stDevAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined]
wEqupGrpId	WORD	Equipment Group ID of the Equipment where the alarm/alert-event occurred
wEqupID	WORD	Equp ID of the equipment where the alarm/alert-event occurred
ebyEqupType	EqupType	Equipment type [system defined]
awAlrmStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate alarms See also Alarm Words of TurnTable *(see page 53)*.
awAlrtStat	ARRAY [0...3] of WORD	Bit-oriented word to indicate alerts See also Alert Word of TurnTable *(see page 66)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
iq_stHdsk360Deg	stHdskIntf	Connects to UME360°. Refer also to the below structure parameter description iq_stHdsk360Deg, iq_stHdsk90Deg, iq_stHdsk180Deg, iq_stHdsk270Deg *(see page 230)*.
iq_stHdsk90Deg	stHdskIntf	Connects to UME90°. Refer also to the below structure parameter description iq_stHdsk360Deg, iq_stHdsk90Deg, iq_stHdsk180Deg, iq_stHdsk270Deg *(see page 230)*.
iq_stHdsk180Deg	stHdskIntf	Connects to UME180°. Refer also to the below structure parameter description iq_stHdsk360Deg, iq_stHdsk90Deg, iq_stHdsk180Deg, iq_stHdsk270Deg *(see page 230)*.
iq_stHdsk270Deg	stHdskIntf	Connects to UME270°. Refer also to the below structure parameter description iq_stHdsk360Deg, iq_stHdsk90Deg, iq_stHdsk180Deg, iq_stHdsk270Deg *(see page 230)*.

iq_stHdsk360Deg, iq_stHdsk90Deg, iq_stHdsk180Deg, and iq_stHdsk270Deg (Data Type: stHdskIntf)

Structure Parameter	Data Type	Description
stFwd	stHdsk	Handshake signals if material flow is in forward direction
stRev	stHdsk	Handshake signals if material flow is in reverse direction

Section 16.4
Interfaces

At a Glance

This section gives an overview how the `TurnTable` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
TurnTable Interfaces	232
Linking the Function Blocks	233

TurnTable Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Handshaking	Information needs to be exchanged between 2 neighboring pieces of equipment in order to manage the material flow.

Optional interface connections are related to:

Tracking	Supervision of the material flow by managing and validating the tracking data.
HMI operation	Together with a Deskzone AFB, a group of equipment can be operated by and monitored using an HMI (e.g., Magelis HMI).
Alarm and alert monitoring	In conjunction with the `AlrmHdlg` FB, the alarms and alerts signaled by the Equipment function block can be logged and summarized.

Linking the Function Blocks

Handshake Interface

To allow the material flow from one equipment to another, handshake information between these 2 equipment function blocks need to be shared. This interface is realized by using a variable of type `stHdskItf` linked to the In_Output pins of the concerned Equipment function blocks. For more detailed information refer to the chapter *Handshaking – Information Dataflow Between Conveying Equipments* *(see page 73)*.

On one hand, the `TurnTable` FB needs to read from the structure the relevant information from the neighboring equipment, on the other hand the `TurnTable` FB has to write its own status in the same structure (handshake variable). In order to share this information by using the identical structure, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Handshake interface with neighbored Conveyor. Depending on the transfer direction, how the conveyor is located to the turntable.	`iq_stHdsk90Deg/` `iq_stHdsk180Deg` / `iq_stHdsk270Deg` / `iq_stHdsk360Deg`	`iq_stHdskNPlus1` / `iq_stHdskNMnus1`	`Conveyor`
Handshake interface with neighbored Input-Conveyor.		`iq_stHdskNPlus1`	`Input`
Handshake interface with neighbored Input-Conveyor.		`iq_stHdskNMnus1`	`Output`
Handshake interface with a Turntable or ERC. Seen from the position of the Turntable or ERC, the Turntable is located on station 90, 180, 270 or 360 degree.		`iq_stHdsk90Deg/` `iq_stHdsk180Deg` / `iq_stHdsk270Deg` / `iq_stHdsk360Deg`	`TurnTable`
			`ERC`

The following screen capture shows, how the interfacing can be done in SoMachine on example of a `TurnTable` and a `Conveyor` function block:

Tracking Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one `TurnTable` AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from `TurnTable` to `TrckCmpx` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information from `TrckCmpx` AFB about the state and the validity of the tracking data and the destination address read out of them.	`i_stDest`	`q_stStat`	`TrckCmpx`
Information to the `TrckCmpx` AFB about the state for managing the tracking data transfer.	`q_stStmStatCmpx`	`i_ststmStatCmpx`	

Deskzone Interface

The Deskzone FB (`DeskzoneSimp` and `DeskzoneCmpx`) manages a group of equipments by gathering and providing the necessary information from the equipment and to the equipment. This information will be shared with the HMI (e.g. Magelis HMI) in order to be able to provide the status information to the HMI and to receive the command from the HMI to operate the Conveying line.

The Deskzone FB uses arrays of structures to exchange the data between `TurnTable` and Deskzone function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Commands to control the Turntable by the operator via the Deskzone AFB what is linked to an HMI.	`i_stEqupCmd`	`q_astEqupCmd[x]`	`DeskzoneSimp` / `DeskzoneCmpx`
State information for monitoring by the operator via the Deskzone AFB what is linked to an HMI.	`q_stEqupStat`	`iq_astEqupStat[x]`	

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 43)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `TurnTable` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the `TurnTable` AFB from `AlrmHdlg` AFB.	`i_stAlrmCtrl`	`q_astAlrmCtrl[x]`	`AlarmHdlg`
`TurnTable` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`q_stDevArm`	`i_astEqupAlrm[x].stDevAlrm`	

Part VI
Tracking

At a Glance

This part describes the `TrckCmpx` and `TrckSimp` function blocks in the Conveying Library system.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
17	TrckCmpx: Handling of Tracking Data for TurnTable and Transfer Table	241
18	TrckSimp: Handling of Tracking Data for Conveyor	261

Chapter 17
TrckCmpx: Handling of Tracking Data for TurnTable and Transfer Table

At a Glance

This chapter describes the functionality and implementation of the `TrckCmpx` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
17.1	Functional Overview	242
17.2	Function Block Description	243
17.3	Pin Description	244
17.4	Interfaces	253

Section 17.1
Functional Overview

Functional Overview

Functional Description

The function block `TrckCmpx` is part of the Conveying library with the following features:

- Handling of tracking data (read, write, cut and delete) in manual mode.
- Adaptation/selection of the destination address in manual mode.
- Validation of received tracking data by the End Sign check.
- Provides its state and the validity of the tracking data to the function block `TurnTable` or `ERC`.
- Control in automatic mode based on the provided information by the function block `TurnTable` or `ERC`.
- Receives tracking data from the preceding function block `TrckCmpx` or `TrckSimp` based on the given source address during load process.
- Sends tracking data to the succeeding function block `TrckCmpx` or `TrckSimp` based on the given destination address during unload process.
- Resolves a provided destination address out of the received tracking data.
- Provides the destination address to the function block `TurnTable` or `ERC`.
- Handles and generates alarms and alerts and provides information to the function block `AlrmHdlg`.

Section 17.2
Function Block Description

TrckCmpx Function Block

Pin Diagram

Section 17.3
Pin Description

At a Glance

This section describes the pin description of the `TrckCmpx` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	245
Output Pin Description	249
Input/Output Pin Description	251

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stInit	stTrckInit	Contains initialization parameters of the function block TrckCmpx. Refer also to the below structure parameter description i_stInit *(see page 245)*.
i_stDestPntr360	stDestPntr	Pointer to the destination address. Used on load process from station 360°. Refer also to structure parameter description for i_stDestPntr*xxx* *(see page 246)*.
i_stDestPntr90	stDestPntr	Pointer to the destination address. Used on load process from station 90°. Refer also to structure parameter description for i_stDestPntr*xxx* *(see page 246)*.
i_stDestPntr180	stDestPntr	Pointer to the destination address. Used on load process from station 180°. Refer also to structure parameter description i_stDestPntr*xxx* *(see page 246)*.
i_stDestPntr270	stDestPntr	Pointer to the destination address. Used on load process from station 270°. Refer also to structure parameter description i_stDestPntr*xxx* *(see page 246)*.
i_stStmStatCmpx	stStmStatCmpx	Connects to function block TurnTable or ERC. Refer also to structure parameter description i_stStmStatCmpx *(see page 247)*.
i_xInit	BOOL	On a rising edge a re-initialization will be executed. The associated Equipment application function block may not be in auto mode.
i_xAlrmRst	BOOL	External reset of alarm

i_stInit (Data Type: stTrckInit)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [must be the same as the associated Equipment application function block]
wEqupId	WORD	Equipment ID [must be the same as the associated Equipment application function block]
ebyEqupType	EqupType	Equipment type [must be the same as the associated Equipment application function block]
xTrckAlrmDsbl	BOOL	If the tracking alarm is disabled, then the tracking data will be not checked for the end sign (16#F0F0). TRUE: Tracking alarm is disabled FALSE: Tracking alarm is enabled

Structure Parameter	Data Type	Description
wTrckLenData	WORD	Length of tracking data. Defines the index of the Word inside the tracking data where the end sign (16#F0F0) is provided.

End Sign Check

If the tracking alarm is not disabled the tracking data are checked for validity. This check compares the Word with the last index (wTrckLenData) of the tracking data with the system defined value 16#F0F0. If the value of this Word is unequal to 16#F0F0 an alert is generated during load process and changes to an alarm in loaded state.

i_stDestPntr*xxx* (Data Type: stDestPntr)

The structure is used to point to a nibble (halfbyte) inside the tracking data array where the code for the destination address is provided.

The TrckCmpx FB provides 4 inputs of data type stDestPntr. For each source station another nibble can be pointed where the destination address is contained.

Structure Parameter	Data Type	Description
byIndx	BYTE	Index of the Word inside the tracking data where the destination address is provided. Range: 0...wTrckLenData
byNiblPntr	BYTE	Nibble (halfbyte) of the Word which provides the destination address. Range: 1...4

The following figure shows 2 examples how the TrckCmpx function block reads the destination address out of the tracking data with the use of the stDestPntr.

* Data type: SE_CONV.awTrckData (ARRAY [0...49] OF WORD)

Example 1:

The value of the 3rd nibble in the Word with index [1] is 16#B. This value is resolved to the destination address 90° and will be provided to the associated `TurnTable` or ERC function block.

Example 2:

The value of the 1st nibble in the Word with index [7] is 16#A. This value cannot be resolved to a destination address. The value for the destination address sent to the associated `TurnTable` or ERC function block is 0 (invalid). If the load process is finished (state Loaded) the alarm `Found destination address is invalid` is generated. This alarm is restrained if the associated `TurnTable` or ERC function block provides an already available destination address which was set by the user application.

The destination address codes are described in the following table:

Value of the Pointed Nibble (halfbyte)		Resolved Destination Address
BINARY	HEX	
2#0111	16#7	360
2#1011	16#B	90
2#1101	16#D	180
2#1110	16#E	270

`i_stStmStatCmpx` (Data Type: `stStmStatCmpx`)

Structure Parameter	Data Type	Description
`xEmty`	BOOL	Status information is empty
`xLdng`	BOOL	Status information is loading
`xLded`	BOOL	Status information is loaded
`xUldg`	BOOL	Status information is unloading
`xFwd`	BOOL	Motor is running in forward direction
`xRev`	BOOL	Motor is running in reverse direction
`xManModInPrgs`	BOOL	Manual mode is in progress (selected by user)
`xStagFnsh`	BOOL	Handshake signal stage is complete and tracking data is assumed
`xCyclRst`	BOOL	Restart cycle is active
`xCyclRstInPrgs`	BOOL	Cycle Restart is in progress
`xDoInitInPrgs`	BOOL	Go to initialization mode is in progress
`xRst`	BOOL	Alarm reset information for the relevant function block `TrckSimp` or `TrckCmpx`
`xStmAuto`	BOOL	Status information is in automatic mode

Structure Parameter	Data Type	Description
xConvInUnldPos	BOOL	Conveyor is in unloading position
xPosChg	BOOL	Position is changed (ERC up/down)
wSrcAddr	WORD	Source address of the next goods
wDestAddr	WORD	Destination Address
xTrckEn	BOOL	TRUE: Tracking is enabled FALSE: Tracking is disabled

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stStat	stDest	Connects to function block TurnTable or ERC. Refer also to the below structure parameter description q_stDest *(see page 249)*.
q_stTrckAlrm	stTrckAlrm	Connects to function block AlrmHdlg. Refer also to the below structure parameter description q_stTrckAlrm *(see page 250)* .
q_stInit	stTrckInit	Provides the actual used initial parameter. Refer also to the below structure parameter description q_stInit *(see page 249)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stStat (Data Type: stDest)

Structure Parameter	Data Type	Description
stTrckStat	stTrckStat	Information of tracking data status
wDestPntrAddr	WORD	Destination address for next upload process, read and generated by the TrckCmpx function block. The address is read out of the tracking data or is sent by the operator previously.

q_stInit (Data Type: stTrckInit)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [must be the same as the associated Equipment application function block]
wEqupId	WORD	Equipment ID [must be the same as the associated Equipment application function block]
ebyEqupType	EqupType	Equipment type [must be the same as the associated Equipment application function block]
xTrckAlrmDsbl	BOOL	If the tracking alarm is disabled, then the tracking data will be not checked for the end sign (16#F0F0). TRUE: Tracking alarm is disabled FALSE: Tracking alarm is enabled

Structure Parameter	Data Type	Description
wTrckLenData	WORD	Length of tracking data. Defines the index of the Word inside the tracking data where the end sign (16#F0F0) is provided.

End Sign Check

If the tracking alarm is not disabled the tracking data are checked for validity. This check compares the Word with the last index (wTrckLenData) of the tracking data with the system defined value 16#F0F0. If the value of this Word is unequal to 16#F0F0 an alert is generated during load process and changes to an alarm in loaded state.

q_stTrckAlrm (Data Type: stTrckAlrm)

Structure Parameter	Data Type	Description
wAlrmStat	WORD	Bit-orientated Word indicates alarms See also Alarm Word of Tracking Complex *(see page 56)*.
wAlrtStat	WORD	Bit-orientated Word indicates alerts See also Alert Word of Tracking Complex *(see page 68)*.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
iq_awTrckData360	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station on 360°.
iq_awTrckData90	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station on 90°.
iq_awTrckData180	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station on 180°.
iq_awTrckData270	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station on 270°.
iq_awTrckHmiData	ARRAY [0…49] of WORD	Link for tracking data handling by the HMI based on the commands send via structure iq_stTrckHmiCmd.
iq_stTrckHmiCmd	stTrckHmiCmd	Command structure to handle the tracking data by the HMI. Refer also to the below structure parameter description iq_stTrckHmiCmd *(see page 252)*.
iq_stTrckHmiStat	stTrckHmiStat	Link to provide the state to a HMI and/or to the function block DeskzoneCmpx. Refer also to the below structure parameter description iq_stTrckHmiStat *(see page 252)* .

iq_stTrckHmiCmd (Data Type: stTrckHmiCmd)

The function block TrckCmpx will execute the commands only if the selected ZoneId, EqupId and the EqupType are consistent with one another. Furthermore the associated Equipment application function block must not be in auto mode.

Structure Parameter	Data Type	Description
xTrckWr	BOOL	Command to write the tracking data on iq_awTrckHmiData into the function block.
xTrckRd	BOOL	Command to read the tracking data from the function block to iq_awTrckHmiData.
xTrckCut	BOOL	Command to cut (read & delete) the tracking data from the function block to iq_awTrckHmiData.
xTrckDel	BOOL	Command to read the current tracking data of the function block.
wZoneIdSel	WORD	Zone ID selected by the operator to handle the tracking data.
wEqupIdSel	WORD	Equipment ID selected by the operator to handle the tracking data.
ebyEqupTypeSel	EqupType	Equipment type selected by the operator to handle the tracking data.
wDestPntrAddr	WORD	Destination address selected by the operator for the next unload process.

iq_stTrckHmiStat (Data Type: stTrckHmiStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID currently used by the function block.
wEqupId	WORD	Equipment ID currently used by the function block.
ebyEqupType	EqupType	Equipment type currently used by the function block.
stDestPntr	stDestPntr	Currently used pointer to a nibble (halfbyte) inside the tracking data what contains the destination address.
awTrckData	ARRAY [0…49] of WORD	Actual tracking data

Section 17.4
Interfaces

At a Glance

This section gives an overview how the `TrckCmpx` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
TrckCmpx Interfaces	254
Linking the Function Blocks	255

TrckCmpx Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Tracking Data Transfer	The tracking information are transferred related to the material flow from one Tracking FB to another.
Equipment Interface	The data transfer is controlled by the operation of the appropriate Equipment FB and vice versa the Tracking FB returns the validity of the data.

Optional interface connections are related to:

HMI operation	Manual tracking data handling and monitoring using an HMI (e.g., Magelis HMI).
Deskzone	Deskzone FB checks the Equipment parameter of the Equipment FB and the appropriate Tracking FB for equality.
Alarm and alert monitoring	In conjunction with `AlrmHdlg` FB, the alarms and alerts signaled by the Tracking function block can be logged and summarized.

Linking the Function Blocks

Tracking Data Interface

To allow the tracking data transfer from one equipment to another, a Tracking FB corresponding to each equipment function is used. The neighboring Tracking FBs must communicate to each other by interface variables. These interface variables are of type `awTrckData` and are linked to the In_Output pins of the concerned Tracking function blocks.

On one hand, the `TrckCmpx` FB needs to read the data during the loading process, on the other hand the `TrckCmpx` FB has to write the data during the unloading process. In order to transfer the data by using the identical interface variable, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB

Data transfer with the neighboring Tracking FB.	`iq_awTrckData90Deg/`
	`iq_awTrckData180Deg /`
	`iq_awTrckData270Deg /`
	`iq_awTrckData360Deg`
		`iq_awTrckDataNMnus1 /`
		`iq_awTrckDataNPlus1`
			`TrckSimp`

		`iq_awTrckData90Deg/`
		`iq_awTrckData180Deg /`
		`iq_awTrckData270Deg /`
		`iq_awTrckData360Deg`
			`TrckCmpx`

The following screen capture shows, how the interfacing can be done in SoMachine on example of a `TrckSimp` and a `TrckCmpx` function block:

Equipment Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one Equipment AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from Equipment to `TrckCmpx` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information to ERC or TurnTable AFB about the state and the validity of the tracking data and the destination address read out of them.	q_stStat	i_stDest	TurnTable
			ERC
Information from the TurnTable or ERC AFB about the state for managing the tracking data transfer.	i_stStmStatCmpx	q_ststmStatCmpx	TurnTable
			ERC

HMI

By using an HMI (e.g. Magelis HMI) the tracking data can be manipulated in manual mode by the operator. The scope of these possible command operations comprises change, copy, delete and write of tracking data (data can be changed, copied, deleted and written by specific commands). For this purpose the Tracking FB provides 2 interfaces, one is the command structure and the other one is the data field.

Description	Pin on AFB	Needs to be Linked to...
Commands to handle the tracking data in manual mode.	iq_stTrckHmiCmd	HMI
Data field of tracking data to exchange the data with the HMI for manual data handling.	iq_awTrckHmiData	

Deskzone Interface

The `DeskzoneCmpx` FB crosschecks the Equipment parameter of the Equipment FB and the associated Tracking FBs for plausibility. In case of different values an alarm will be signaled by the `DeskzoneCmpx` FB.

The `DeskzoneCmpx` FB uses arrays of structures to link the data with the Equipment and Tracking function blocks. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Structure includes the Equipment parameter and the currently carried tracking data.	`iq_stTrckHmiStat`	`iq_astTrckHmiStat[x]`	`DeskzoneCmpx`

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked tracking function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 279)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `TrckCmpx` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
`TrckCmpx` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`q_stTrckAlrm`	`i_astEqupAlrm[x].stTrckAlrm`	`AlarmHdlg`

Chapter 18
TrckSimp: Handling of Tracking Data for Conveyor

At a Glance

This chapter describes the functionality and implementation of the `TrckSimp` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
18.1	Functional Overview	262
18.2	Function Block Description	263
18.3	Pin Description	264
18.4	Interfaces	271

Section 18.1
Functional Overview

Functional Overview

Functional Description

The function block `TrckSimp` is part of the Conveying library with the following features:

- Handling of tracking data (read, write, cut and delete) in manual mode.
- Validation of received tracking data by the end sign check.
- Provides its state and the validity of the tracking data to the function block `Conveyor`, `Input` or `Output`.
- Control in auto mode based on the provided information by the function block `Conveyor`, `Input` or `Output`.
- Receives tracking data from the preceding function block `TrckCmpx` or `TrckSimp` based on the given transfer direction during load process.
- Sends tracking data to the secceeding function block `TrckCmpx` or `TrckSimp` based on the given transfer direction during unload process.
- Handles and generates alarms and alerts and provides information to the function block `AlrmHdlg`

Section 18.2
Function Block Description

TrckSimp Function Block

Pin Diagram

Section 18.3
Pin Description

At a Glance

This section describes the pin description of the `TrckSimp` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	265
Output Pin Description	267
Input/Output Pin Description	269

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stInit	stTrckInit	Contains the initialization parameters of the function block TrckSimp. Refer also to the below structure parameter description i_stInit *(see page 265)*.
i_stStmStatSimp	stStmStatSimp	Connects to following function blocks ● Conveyor ● Input ● Output Refer also to the below structure parameter description i_stStmStatSimp *(see page 266)* .
i_xInit	BOOL	On a rising edge a re-initialization will be executed. The associated Equipment application function block does not have to be in auto mode.
i_xAlrmRst	BOOL	External reset of alarm.

i_stInit (Data Type: stTrckInit)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [must be the same as the associated Equipment application function block]
wEqupId	WORD	Equipment ID [must be the same as the associated Equipment application function block]
ebyEqupType	EqupType	Equipment type [must be the same as the associated Equipment application function block]
xTrckAlrmDsbl	BOOL	If the tracking alarm is disabled, then the tracking data will be not checked for the end sign (16#F0F0). TRUE: Tracking alarm is disabled FALSE: Tracking alarm is enabled
wTrckLenData	WORD	Length of tracking data. Defines the index of the Word inside the tracking data where the end sign (16#F0F0) is provided.

End Sign Check

If the tracking alarm is not disabled the tracking data are checked for validity. This check compares the Word with the last index (`wTrckLenData`) of the tracking data with the system defined value `16#F0F0`. If the value of this Word is unequal to `16#F0F0` an alert is generated during load process and changes to an alarm in loaded state.

`i_stStmStatSimp` (Data Type: `stStmStatSimp`)

Structure Parameter	Data Type	Description
`xEmty`	`BOOL`	Status information is empty
`xLdng`	`BOOL`	Status information is loading
`xLded`	`BOOL`	Status information is loaded
`xUldg`	`BOOL`	Status information is unloading
`xFwd`	`BOOL`	State machine is decided to transfer in forward direction
`xRev`	`BOOL`	State machine is decided to transfer in in reverse direction
`xManModinPrgs`	`BOOL`	Manual mode is in progress, selected by user
`xStagFnsh`	`BOOL`	Handshake signal stage is complete to indicate that tracking data is assumed
`xCycRst`	`BOOL`	Cycle restart is active
`xCyclRstInPrgs`	`BOOL`	Cycle restart is in progress
`xDolnitInPrgs`	`BOOL`	Indicates that the Go to Intialize Mode is in progress
`xRst`	`BOOL`	Alarm reset information for the relevant function block `TrckSimp` or `TrckCmpx`
`xStmAuto`	`BOOL`	Status information is in automatic mode
`xTrckEn`	`BOOL`	TRUE: Tracking is enabled FALSE: Tracking is disabled

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_stStat	stTrckStat	Connects to function block Conveyor, Input or Output. Refer also to the below structure parameter description q_stStat *(see page 267)*.
q_stTrckAlrm	stTrckAlrm	Connects to function block AlrmHdlg. Refer also to the below structure parameter description q_stTrckAlrm *(see page 267)*.
q_stInit	stTrckInit	Provides the actual used parameters for the initialization. Refer also to the below structure parameter description q_stInit *(see page 268)*.
q_xAlrm	BOOL	TRUE: at least 1 alarm is detected FALSE: no alarm
q_xAlrt	BOOL	TRUE: at least 1 alert is detected FALSE: no alert

q_stStat (Data Type: stTrckStat)

Structure Parameter	Data Type	Description
xFuncOk	BOOL	Indicates if tracking data are valid as result of the End sign check.
xInfoOk	BOOL	Indicates the status of function block TrckSimp. TRUE: Function block is ready for operation.

q_stTrckAlrm (Data Type: stTrckAlrm)

Structure Parameter	Data Type	Description
wAlrmStat	WORD	Bit-orientated word to indicate alarms See also Alarm Word of Tracking Simple *(see page 57)*.
wAlrtStat	WORD	Bit-orientated word to indicate alerts See also Alert Word of Tracking Simple *(see page 69)*.

q_stInit (Data Type: stInit)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [must be the same as the associated Equipment application function block]
wEqupId	WORD	Equipment ID [must be the same as the associated Equipment application function block]
ebyEqupType	EqupType	Equipment type [must be the same as the associated Equipment application function block]
xTrckAlrmDsbl	BOOL	If the tracking alarm is disabled, then the tracking data will be not checked for the end sign (16#F0F0). TRUE: Tracking alarm is disabled FALSE: Tracking alarm is enabled
wTrckLenData	WORD	Length of tracking data. Defines the index of the Word inside the tracking data where the end sign (16#F0F0) is provided.

End Sign Check

If the tracking alarm is not disabled the tracking data are checked for validity. This check compares the Word with the last index (wTrckLenData) of the tracking data with the system defined value 16#F0F0. If the value of this Word is unequal to 16#F0F0 an alert is generated during load process and changes to an alarm in loaded state.

Input/Output Pin Description

Input/Output Pin Description

Input/Output	Data Type	Description
iq_awTrckDataNMnus1	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station in reverse direction.
iq_awTrckDataNPlus1	ARRAY [0…49] of WORD	Link for tracking data transfer with the Tracking function block of the station in forward direction.
iq_awTrckHmiData	ARRAY [0…49] of WORD	Link for tracking data handling by the HMI based on the commands sent via structure iq_stTrckHmiCmd.
iq_stTrckHmiCmd	stTrckHmiCmd	Command structure to handle the tracking data by the HMI. Refer also to the below structure parameter description iq_stTrckHmiCmd *(see page 269)*.
iq_stTrckHmiStat	stTrckHmiStat	Link to provide the state to a HMI and/or to the function block DeskzoneCmpx. Refer also to the below structure parameter description iq_stTrckHmiStat *(see page 270)*.

iq_stTrckHmiCmd (Data Type: stTrckHmiCmd)

The function block TrckSimp will execute the commands only if the selected ZoneId, EqupId and the EqupType are conform to its own. Furthermore the associated Equipment application function block may not be in auto mode.

Structure Parameter	Data Type	Description
xTrckWr	BOOL	Command to write the tracking data on iq_awTrckHmiData into the function block.
xTrckRd	BOOL	Command to read the tracking data from the function block to iq_awTrckHmiData.
xTrckCut	BOOL	Command to cut (read & delete) the tracking data from the function block to iq_awTrckHmiData.
xTrckDel	BOOL	Command to read the currently tracking data of the function block.
wZoneIdSel	WORD	Zone ID selected by the operator to handle the tracking data.
wEqupIdSel	WORD	Equipment ID selected by the operator to handle the tracking data.
ebyEqupTypeSel	EqupType	Equipment type selected by the operator to handle the tracking data.

iq_stTrckHmiStat (Data Type: stTrckHmiStat)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID currently used by the function block.
wEqupId	WORD	Equipment ID currently used by the function block.
ebyEqupType	EqupType	Equipment type currently used by the function block.
stDestPntr	stDestPntr	not used by TrckSimp function block
awTrckData	ARRAY [0…49] of WORD	Actual tracking data

Section 18.4
Interfaces

At a Glance

This section gives an overview how the `TrckSimp` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
TrckSimp Interfaces	272
Linking the Function Blocks	273

TrckSimp Interfaces

Overview

In order for the function block to operate correctly, it is essential to share relevant information with other application function blocks. The FB provides several connections to interface and to communicate with AFBs out of the Conveying library. These interface connections can be divided into 2 types: mandatory and optional.

Mandatory interface connections are related to:

Tracking Data Transfer	The tracking information are transferred related to the material flow from one Tracking FB to another.
Equipment Interface	The data transfer is controlled by the operation of the appropriate Equipment FB and vice versa the Tracking FB returns the validity of the data.

Optional interface connections are related to:

HMI operation	Manual tracking data handling and monitoring using an HMI (e.g., Magelis HMI).
Deskzone	Deskzone FB checks the Equipment parameter of the Equipment FB and the appropriate Tracking FB for equality.
Alarm and alert monitoring	In conjunction with `AlrmHdlg` FB, the alarms and alerts signaled by the Tracking function block can be logged and summarized.

Linking the Function Blocks

Tracking Data Interface

To allow the tracking data transfer from one equipment to another, a Tracking FB corresponding to each equipment function is used. The neighboring Tracking FBs must communicate to each other by interface variables. These interface variables are of type `awTrckData` and linked to the In_Output pins of the concerned Tracking function blocks.

On one hand, the `TrckSimp` FB needs to read the data during the loading process, on the other hand the `TrckSimp` FB has to write the data during the unloading process. In order to transfer the data by using the identical interface variable, one variable for each direction needs to be defined.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Data transfer with the preceding Tracking FB.	`iq_awTrckDataNMnus1`	`iq_awTrckDataNPlus1`	`TrckSimp`
Data transfer with the succeeding Tracking FB.	`iq_awTrckDataNPlus1`	`iq_awTrckDataNMnus1`	
Data transfer with the neighboring Tracking FB. Seen from the position of the `TrckCmpx`, the `TrckSimp` is located on station 90, 180, 270 or 360 degree.	`iq_awTrckDataNMnus1` / `iq_awTrckDataNPlus1`	`iq_awTrckData90Deg`/ `iq_awTrckData180Deg` / `iq_awTrckData270Deg` / `iq_awTrckData360Deg`	`TrckCmpx`

The following screen capture shows, how the interfacing can be done in SoMachine on example of a `TrckSimp` and a `TrckCmpx` function block:

Equipment Interface

The identification of pallets and the check of valid data transferred along with the pallet requires a tracking functionality in a conveying line. For this purpose one Equipment AFB is associated with one Tracking function block.

As the designated pins can be directly linked (from Equipment to `TrckSimp` and vice versa), no variable between these 2 FBs is necessary.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Information to `Conveyor`, `Input` or `Output` AFB about the state and the validity of the tracking data.	`q_stStat`	`i_stTrckStat`	`Conveyor`
			`Input`
			`Output`
Information from the `TurnTable` or `ERC` AFB about the state for managing the tracking data transfer.	`i_stStmStatSimp`	`q_stStmStatSimp`	`Conveyor`
			`Input`
			`Output`

HMI

By using an HMI (e.g. Magelis HMI) the tracking data can be manipulated in manual mode by the operator. The scope of these possible command operations comprises change, copy, delete and write of tracking data (data can be changed, copied, deleted and written by specific commands). For this purpose the Tracking FB provides 2 interfaces, one is the command structure and the other one is the data field.

Description	Pin on AFB	Needs to be Linked to...
Commands to handle the tracking data in manual mode.	`iq_stTrckHmiCmd`	HMI
Data field of tracking data to exchange the data with the HMI for manual data handling.	`iq_awTrckHmiData`	

Deskzone Interface

The `DeskzoneCmpx` FB crosschecks the Equipment parameter of the Equipment FB and the associated Tracking FBs for plausibility. In case of different values an alarm will be signaled by the `DeskzoneCmpx` FB.

The `DeskzoneCmpx` FB uses arrays of structures to link the data with the Equipment and Tracking function blocks. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Structure includes the Equipment parameter and the currently carried tracking data.	`iq_stTrckHmiStat`	`iq_astTrckHmiStat[x]`	`DeskzoneCmpx`

Alarm Handling Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked tracking function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. For more details on alarm handling, please refer to chapter *Alarm Handling* *(see page 279)*.

The `AlrmHdlg` FB uses arrays of structures to exchange the data between `TrckSimp` and `AlrmHdlg` function block. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
`TrckSimp` AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert lds.	`q_stTrckAlrm`	`i_astEqupAlrm[x].stTrckAlrm`	`AlarmHdlg`

Part VII
Alarm Handling

At a Glance

This part describes the `AlrmHdlg` function block in the Conveying Library system.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
19	AlrmHdlg: Managing Alarms and Alerts	281
20	AlrmHdlgTextFile	299

Chapter 19
AlrmHdlg: Managing Alarms and Alerts

At a Glance

This chapter describes the functionality and implementation of the `AlrmHdlg` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
19.1	Functional Overview	282
19.2	Function Block Description	283
19.3	Pin Description	284
19.4	Interfaces	291
19.5	Alarmtext File	296

Section 19.1
Functional Overview

Functional Overview

Functional Description

The function block `AlrmHdlg` is a part of the Conveying library with the following features:

- Receives alarm and alert status information from the connected function block Equipment, relevant to function blocks `TrckSimp` or `TrckCmpx` and `DeskzoneSimp` or `DeskzoneCmpx`
- Provides alarm and alert status information to HMI [FIFO principle]
- Receives reset information of alarms from HMI
- Provides reset and release information of alarms to the controlled function block Equipment
- Saves alarms into a separate *Alarm.txt* file
- Provision for user to match alarm text to the alarm ID in *Conveyor.txt* file

Section 19.2
Function Block Description

AlrmHdlg Function Block

Pin Diagram

Section 19.3
Pin Description

At a Glance

This section gives the pin description of the `AlrmHdlg` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	285
Output Pin Description	287

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_stInit	stInitOfHdlgOfAlrm	Initializes values of timer and enables/disables functions during the start up phase. Refer also to the below structure parameter description i_stInit *(see page 285)*.
i_xFbAvai	BOOL	Indicates the availability of Fieldbus [depends on start-up time]
i_stHmiRstCmd	stHmiRstCmd	Connects to HMI. Refer also to the below structure parameter description i_stHmiRstCmd *(see page 286)*.
i_stZoneAlrm	stZoneAlrm	Connects to user equipment. Refer also to the below structure parameter description i_stZoneAlrm *(see page 286)*.
i_stDeskAlrm	stDeskAlrm	Connect to function block DeskzoneSimp or DeskzoneCmpx. Refer also to the below structure parameter description i_stDeskAlrm *(see page 286)*.
i_astEqupAlrm	ARRAY [0..24] of stAlrmRprt	Indicates alarm and alert words. Refer also to the below structure parameter description i_astEqupAlrm *(see page 286)*.

i_stInit (Data Type: stInitOfHdlgOfAlrm)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID [user defined] If the value is 0, then zone is disabled
ptAlrmMsge	POINTER TO abyAlrmMsge	Pointer to an external data ARRAY [0...199, 0...54] where the alarm message will be written in ASCII code by the AlrmHdlg function block.
ptAlrtMsge	POINTER TO abyAlrmMsge	Pointer to an external data ARRAY [0...199, 0...54] where the alert message will be written in ASCII code by the AlrmHdlg function block.
tDlyForAlrmRele	TIME	After global alarm is reset, the release is set to TRUE with a time delay

For more information about abyAlrmMsge refer to the detailed data type description *(see page 305)*.

i_stHmiRstCmd (Data Type: stHmiRstCmd)

Structure Parameter	Data Type	Description
wZoneIdSel	WORD	Zone ID [user defined]
wEqupIdSel	WORD	Equipment ID [system defined]
xRst	BOOL	Reset bit

i_stZoneAlrm (Data Type: stZoneAlrm)

Structure Parameter	Data Type	Description
wAlrmExtl	WORD	Local alarm word of zone The bits 2...15 is used by the user.
wAlrtExtl	WORD	Local alert word of zone The bits 0...15 is used by the user.

i_stDeskAlrm (Data Type: stDeskAlrm)

Structure Parameter	Data Type	Description
wAlrmStat	WORD	Alarm status word from function block DeskzoneSimp or DeskzoneCmpx
wAlrtStat	WORD	Alert status word from function block DeskzoneSimp or DeskzoneCmpx
wZoneId	WORD	Zone ID of the equipment where the alarm/alert-event occurred

i_astEqupAlrm (Data Type: ARRAY [0...24] of stAlrmRprt)

Structure Parameter	Data Type	Description
stDevAlrm	stDevAlrm	Connects to function block TrckSimp or TrckCmpx Contains information from the function block equipment: ● Zone ID, equipment ID and type ● Alarm words ● Alert words
stTrckAlrm	stTrckAlrm	Connects to function block TrckSimp or TrckCmpx Contains information from the function block TrckSimp or TrckCmpx: ● Alarm words ● Alert words

Output Pin Description

Output Pin Description

Output	Data Type	Description
q_xBusy	BOOL	Indicates the function block is busy with checking the alarm and alert bit status.
q_astAlrmCtrl	ARRAY [0...24] of stAlrmCtrl	Connects to function block Equipment. Refer also to the below structure parameter description q_astAlrmCtrl *(see page 288)*.
q_astAlrmActv	ARRAY [0...49] of stAlrmMsgeSimp	Connects to user equipment. Indicates active alarms. Refer also to the below structure parameter description q_astAlrmActv *(see page 288)*.
q_astAlrtActv	ARRAY [0...49] of stAlrmMsgeSimp	Connects to user equipment. Indicates active alerts. Refer also to the below structure parameter description q_astAlrtActv *(see page 288)*.
q_astAlrmHist	ARRAY [0...99] of stAlrmMsgeCmpx	Connects to user equipment. Indicates alarms in the history field. Refer also to the below structure parameter description q_astAlrmHist *(see page 289)*.
q_astAlrtHist	ARRAY [0...99] of stAlrmMsgeCmpx	Connects to user equipment Indicates alerts in the history field. Refer also to the below structure parameter description q_astAlrtHist *(see page 289)*.
q_wAlrmMsgeCnt	WORD	Counter which indicates the actual alarm number
q_wAlrtMsgeCnt	WORD	Counter which indicates the actual alert number
q_stAlrmStat	stAlrmStat	Connects to user equipment. Refer also to the below structure parameter description q_stAlrmStat *(see page 290)*.
q_stSumryAlrmStat	stSumryAlrmStat	Indicates the alarm and alert status of zone and internal. Refer also to the below structure parameter description q_stSumryAlrmStat *(see page 290)*.

q_astAlrmCtrl (Data Type: ARRAY [0...24] of stAlrmCtrl)

Structure Parameter	Data Type	Description
xAlrmRele	BOOL	TRUE: No overall alarm is active in zone FALSE: Overall alarm is active in zone
xRstAlrm	BOOL	Resets alarm in zone

q_astAlrmActv (Data Type: ARRAY [0...49] of stAlrmMsgeSimp)

Structure Parameter	Data Type	Description
wZoneId	WORD	Zone ID of the equipment [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
wEqupId	WORD	Equipment ID of the equipment where the alarm/alert-event occurred
byAlrmId	BYTE	Unique alarm ID

q_astAlrtActv (Data Type: ARRAY [0...49] of stAlrmMsgeSimp)

Structure Parameter	Data Type	Description
wZoneId	WORD	Equipment zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID of the Equipment where the alarm/alert-event occurred
wEqupId	WORD	Equipment ID of the equipment where the alarm/ alert-event occurred
byAlrmId	BYTE	Unique alert ID

q_astAlrmHist (Data Type: ARRAY [0...99] of stAlrmMsgeCmpx)

Structure Parameter	Data Type	Description
wZoneId	WORD	Equipment zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID of the equipment where the alarm/alert-event occurred
wEqupId	BYTE	Equipment ID of the Equipment where the alarm/alert-event
byAlrmId	BYTE	Unique alarm ID
xEventStat	BOOL	TRUE: Indicates the presence of an alarm FALSE: Indicates the absence of an alarm
dtTimeStamp	DATE_AND_TIME	Time stamp of the event

q_astAlrtHist (Data Type: ARRAY [0...99] of stAlrmMsgeCmpx)

Structure Parameter	Data Type	Description
wZoneId	WORD	Equipment zone ID [user defined]
ebyEqupType	EqupType	Equipment type [system defined]
wEqupGrpId	WORD	Equipment group ID of the Equipment where the alarm/alert-event occurred
wEqupId	WORD	Equipment ID [user defined]
byAlrmId	BYTE	Unique alert ID
xEventStatus	BOOL	TRUE: Indicates the presence of alert FALSE: Indicates the absence of alert
dtTimeStamp	DATE_AND_TIME	Time stamp of the event

q_stAlrmStat (Data Type: stAlrmStat)

Structure Parameter	Data Type	Description
wZoneAlrmExtl	WORD	Indicates the local alarm word of zone See also Alarm Words of Alarm Handling *(see page 58)*.
wZoneAlrtExtl	WORD	Indicates the local alert word of zone See also Alert Word of Alarm Handling *(see page 70)*.
wAlrmInt	WORD	Indicates the internal alarm word See also Alarm Words of Alarm Handling *(see page 58)*.
wAlrtInt	WORD	Indicates the internal alert word See also Alert Word of Alarm Handling *(see page 70)*.

q_stSumryAlrmStat (Data Type: stSumryAlrmStat)

Structure Parameter	Data Type	Description
xZoneAlrm	BOOL	TRUE: Indicates alarm in zone FALSE: No alarm
xZoneAlrt	BOOL	TRUE: Indicates alert in zone FALSE: No alert
xAlrmIntl	BOOL	TRUE: Indicates internal alarm FALSE: No alarm
xAlrtIntl	BOOL	TRUE: Indicates internal alert FALSE: No alert

Section 19.4
Interfaces

At a Glance

This section gives an overview how the `AlrmHdlg` function block is linked to other AFBs out of the Conveying library.

What Is in This Section?

This section contains the following topics:

Topic	Page
AlrmHdlg Interfaces	292
Linking the Function Blocks	293

AlrmHdlg Interfaces

Overview

The graphic below demonstrates the principle which is followed to integrate the `AlrmHdlg` FB into the user's application

Linking the Function Blocks

HMI

The main function of the `AlrmHdlg` FB is, to log all the received alarm and alert information from the linked AFBs. These information will be treated and are provided on interface pins which can be linked to an HMI. In addition the `AlrmHdlg` FB will forward a reset command to the addressed Equipment AFBs.

Description	Pin on AFB	Needs to be Linked to...
Array what contains up to 50 active alarms with information about the source where the alarms were detected.	`q_astAlrmActv`	Usually linked to a HMI (e.g. Magelis operator panel)
Array what contains up to 50 active alerts with information about the source where the alerts were detected.	`q_astAlrtActv`	
Array what contains the last 100 alarm events with information about the source where the alarm events were occurred together with the timestamp and the state.	`q_astAlrmHist`	
Array what contains the last 100 alert events with information about the source where the alert events were occurred together with the timestamp and the state.	`q_astAlrtHist`	
Reset information from the HMI, will be forwarded to the addressed Equipment AFB.	`i_stHmiRstCmd`	

Equipment/Tracking Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from each linked equipment and tracking function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked. The `AlrmHdlg` FB provides information in a dedicated array, which is related to the equipment. These data apprise the equipment function block, if a zone alarm is active or a reset alarm command has been triggered.

The interfacing pins to the equipment and tracking FBs are provided as pins of data type array of structure. Each array element is representing one equipment in the conveying line. The index of this array (symbolized in the table below with **x**) is unique and must be kept through out the entire application.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
State of a zone alarm and command to reset all alarms and alerts of the Equipment AFB.	`q_astAlrmCtrl[x]`	`i_stAlrmCtrl`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`
Equipment AFBs provide the state of all alarms and alerts to the `AlrmHdlg` AFB for generating alarm and alert Ids.	`i_astEqupAlrm[x].stDevAlrm`	`q_stDevAlrm`	`Conveyor` / `Input` / `Output` / `TurnTable` / `ERC`
Tracking AFBs provide the state of all alarms and alerts to the `AlrmHdlg` AFB for generating alarm and alert Ids.	`i_astEqupAlrm[x].stTrckAlrm`	`q_stTrckAlrm`	`TrckSimp` / `TrckCmpx`

Deskzone Interface

The `AlrmHdlg` FB collects the detected alarms and alerts from the linked Deskzone function block. While these alarms and alerts arrive and leave, the status of these events need to be tracked.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Deskzone AFB provides the state of all alarms and alerts to `AlrmHdlg` AFB for generating alarm and alert Ids.	`i_stDeskAlrm`	`q_stAlrm`	`DeskzoneCmpx` / `DeskzoneSimp`

Zone Alarms/Alerts Interface

The `AlrmHdlg` FB provides an interface pin to link user defined alarm and alert information. These alarms and alerts will be summarized and logged by the `AlrmHdlg` FB.

Description	Pin on AFB	Needs to be Linked to...
Up to 16 user defined alarms can be linked to the AlrmHdlg FB.	`i_stZoneAlrm.wAlrmExt1`	Alarms detected by the user application, e.g. emergency stop device.
Up to 16 user defined alerts can be linked to the AlrmHdlg FB.	`i_stZoneAlrm.wAlrtExt1`	Alerts detected by the user application, e.g. maintenance interval elapsed.

AlrmHdlgTextFile Interface

In conjunction with the `AlrmHdlgTextFile` FB the logged alarm and alert events will be stored in a text file on the controller. The `AlrmHdlg` FB generates the alarm and alert event messages in ASCII code and stores them into 2 external Arrays. Each Array is linked to a single `AlrmHdlg-TextFile` FB. One is linked to alarms and the other one is linked to alerts. Two message counters provided as output pins on the `AlrmHdlg` FB complement the interface to the `AlrmHdlg-TextFile` FB. For more details on the `AlrmHdlgTextFile` FB, please refer to chapter *AlrmHdlgTextFile* *(see page 299)*.

Description	Pin on AFB	Interfacing	
		Needs to be Linked to...	AFB
Pointer to the external array to store the alarm event messages in ASCII code.	`i_stInit.ptAlrmMsge`	`i_ptMsgeData`	Instance of `AlrmHdlgtextFile` to write alarm messages into a file.
Counter indicates the amount of generated alarm event messages in ASCII code.	`q_wAlrmMsgeCnt`	`i_wMsgeCnt`	
Pointer to the external array to store the alert event messages in ASCII code.	`i_stInit.ptAlrtMsge`	`i_ptMsgeData`	Instance of `AlrmHdlgtextFile` to write alert messages into a file.
Counter indicates the amount of generated alert event messages in ASCII code.	`q_wAlrtMsgeCnt`	`i_wMsgeCnt`	

Section 19.5
Alarmtext File

Alarmtext File

Overview

Besides the `AlrmHdlg` and `AlrmHdlgTextFile` function blocks, the Conveying Library contains 2 files of *ConveyingAlarmTextFile* in two different formats:

- Microsoft Excel (*.xls*)
- Acrobat Portable Document Format (*.pdf*)

To be language independent, the function blocks of the Conveying Library support unique alarm and alert IDs with a value ranging from 0…255. Using the alarm textfiles the ID can be combined with the corresponding message in the HMI application.

To access the files, double-click the library browser of the Conveying Library and then store the respective file on the local hard disk.

The *ConveyingAlarmTextFile* (*.xls* or *.pdf*) contains the ID equations - separated into alarms and alerts in the following languages:

- English
- German
- French
- Italian
- Spanish
- Chinese

The screen capture below shows an extract of the alarm Ids in English language.

	A	B
1	Alarm ID	English
38	100	Parameter Alarm of Equipment Type in EquipmentAFB
39	101	Parameter Alarm of Preset value for Load Timer in forward direction
40	102	Parameter Alarm of Preset value for Load Timer in reverse direction
41	103	Parameter Alarm of Preset value for Turnmove Timer
42	104	Parameter Alarm of Preset value for Verticalmove Timer
43	105	Feedback Alarm of Motor 1 in forward direction
44	106	Feedback Alarm of Motor 1 in reverse direction
45	107	Feedback Alarm of Motor 2 in forward direction
46	108	Feedback Alarm of Motor 2 in reverse direction
47	109	Feedback Alarm of Motor 3 in forward direction
48	110	Feedback Alarm of Motor 3 in reverse direction

Chapter 20
AlrmHdlgTextFile

At a Glance

This chapter describes the functionality and implementation of `AlrmHdlgTextFile` function block.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
20.1	Functional and System Overview	300
20.2	Function Block Description	302
20.3	Pin Description	303

Section 20.1
Functional and System Overview

Functional Overview

Functional Description

The `AlrmHdlgTextFile` function block is designed to write alarm/alert messages into a file of the file system in the controller. These messages are created by the `AlrmHdlg` function block.

This function block is part of the Conveying Library and provides the following features:

- Creation of a file on the controller; file and folder name are predefined by the function block
- Writing of alarm and respective alert messages into this file; messages and count are provided by `AlrmHdlg` function block
- Creation of a copy (archive) of this file automatically if 1000 messages are written
- Creation of a copy (archive) of this file through command interface
- Deletion of existing files through command interface
- Generation of its own alarms and alerts
- Possibility to interlock this function block if other time-consuming applications execute
- Provides an output `q_xBusy` to interlock other time-consuming applications

Functional Overview

This overview describes the functionality and implementation of the `AlrmHdlgTextFile` function block.

The graphic above shows the principle of the `AlrmHdlgTextFile` function block in conjunction with the `AlrmHdlg` function block for one zone. The upper instance of `AlrmHdlgTextFile` function block is dedicated to write the alarmtext file in the file system of the controller. The lower function block instance represents the writing of the alert text file. The link between `i_xInlk` and `q_xbusy` avoids that both function block instances execute file operations at the same time which might result in data loss.

Section 20.2
Function Block Description

AlrmHdlgTextFile Function Block

Pin Diagram

Section 20.3
Pin Description

At a Glance

This section gives the pin description of the `AlrmHdlgTextFile` function block.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Pin Description	304
Output Pin Description	306

Input Pin Description

Input Pin Description

Input	Data Type	Description
i_xInlk	BOOL	This input is mainly relevant if more than one instance of this function block is used. This input is used to interlock with the q_xBusy output of a further instance of the AlrmHdlgTextFile function block to avoid performance issues.
i_ptMsgeData	POINTER TO abyAlrmMsge	This pointer input is used to connect and read the external data array. This data array contains the messages in ASCII code which have been written by the AlrmHdlg function block. See also abyAlrmMsge for more detailed description *(see page 305)*.
i_wMsgeCnt	WORD	This input must be connect to the according output of AlrmHdlg function block (q_wAlrmMsgeCnt/q_AlrtMsgeCnt). It reflects the number of written messages into the external data array.
i_byFileName	BYTE	The predefined filename is defined by values for the alarm or alert messages file. Valid range: 1...4
i_stCmd	stCmdHdlgAlrmTextFile	Command structure to control the AlrmHdlgTextFile function block.
i_udiWrTmot	UDINT	Write File Time out configuration parameter. Default value: 0 (t#0s) It is recommended to use with t#0s.

NOTE: The function block accesses the file system of the controller by using file handling routines out of the CAA file library. Up to 4 different files can be created by default with 4 different instances of AlrmHdlgTextFile function blocks.

Be sure that only one file write operation is executing at a time. Use the link to the q_xBusy pin of an active function block to the interlock pin i_xInlk of any other AlrmHdlgTextFile function blocks to avoid simultaneous file write operations. Otherwise, the result could be the loss of important alarm information.

NOTICE
LOSS OF DATA Be sure that only one instance of the AlrmHdlgTextFile function block is writing data at a time. **Failure to follow these instructions can result in equipment damage.**

abyAlrmMsge

This data type is part of the Conveying Library and includes an ARRAY `[0...199, 0...54]` OF BYTES. The log-register contains 200 messages. One message has a length of 55 bytes and includes separators and a line break. The messages are provided in ASCII code.

The following table shows the design of a message inside the log register and represents one data entry of an alarm or alert message.

MsgeCnt	ZoneID	EqupGrpId	EqupId	EqupType	AlrmId	EventStat	TimeStamp
001,	00001,	00017,	00012,	021,	112,	GONE,	2009-10-20-09:56:53$N

i_stCmd (Data Type: stCmdHdlgAlrmTextFile)

Structure Parameter	Data Type	Description
`xAcknArchCmplt`	`BOOL`	Resets on a rising edge the output of `q_xArchCmplt` which is set whenever an archive file has been created
`xAlrmRst`	`BOOL`	Resets alarms and alerts of the `HdlgAlrmTextFile` function block.
`xCreatArch`	`BOOL`	Creates an archive file on a rising edge, assumed the function block is in Ready state.
`xDelFile`	`BOOL`	Deletes the current file immediately on a rising edge of this input, assumed the function block is in Ready state.
`xDelArch`	`BOOL`	Deletes the selected archive file immediately on a rising edge of this input, assumed the function block is in Ready state.

i_byFileName

Four names are predefined for alarms and alerts which can be selected by following values:

- 1: Zone1Alarm.txt
- 2: Zone2Alarm.txt
- 3: Zone1Alert.txt
- 4: Zone2Alert.txt

Any value out of this range results in an alarm message of this function block.

Output Pin Description

Output Pin Description

Input	Data Type	Description
q_xInlk	BOOL	Reflects the state of corresponding input i_xInlk. TRUE: AlrmHdlgTextFile function block will not be executed FALSE: No limitation on usage of AlrmHdlgTextFile function block
q_xBusy	BOOL	Reflects an execution of any file handling action of the AlrmHdlgTextFile function block (open, close, write, delete and so on.)
q_xArchCmplt	BOOL	Confirms the completion of the archive file creation
q_wMsgeInFile	WORD	Number of messages which are written successfully into the current file
q_wMsgeInArch	WORD	Number of messages which are written successfully into the archive file
q_eiStat	StatHdlgAlrmTextFile	This output indicates the actual state of AlrmHdlgTextFile function block by an enumeration
q_stAlrmStat	stAlrmStatHdlgAlrmTextFile	Structure includes the information about the reason of an alarm/alert state of the function block.

q_eiStat (Enumeration Data Type: StatHdlgAlrmTextFile)

For a better diagnostic, the output q_eiStat returns the state of the AlrmHdlgTextFile function block by using the enumeration StatHdlgAlrmTextFile.

A description of this enumeration is given in the table below:

Element	Value	Description
Ready	0	Ready state without any alarm.
OpenFile	1	Opening a file.
CreateDir	2	Creating a directory.
WriteFile	3	Writing to the opened file.
CloseFile	4	Closing the file.
RenameFile	5	Renaming the file.
DeleteFile	6	Deleting the file.
Alarm	7	Alarm state
Interlocked	8	Interlocked with input i_xInlk.

q_stAlrmStat (Data Type: stAlrmStat)

Input	Data Type	Description
wAlrm	WORD	Bit orientated status WORD of an occurring alarm. See also Alarm Word of AlrmHdlgTextFile *(see page 59)*.
wAlrt	WORD	Bit orientated status WORD of an occurring alert See also Alert Word of AlrmHdlgTextFile *(see page 71)*.
eCAAErrorNr	FILE.ERROR (ENUM)	Reflects the detected error code of the used CoDeSys CAA file library which allows the file handling on the controller.

eCAAErrorNr (Enumeration Data Type: FILE.ERROR)

The element eCAAErrorNr of the output q_stAlrmStat provides the returned error code of the used function blocks out of the CAA.FILE library inside the AlrmHdlgTextFile function block. The data type is an enumeration name as FILE.ERROR and is also provided by the CAA.FILE library.

A description of this enumeration is given in the table below:

Element	Value	Description
FILE_NO_ERROR	0	No detected error
FILE_FIRST_ERROR	5100	First detected error (library-specific)
FILE_TIME_OUT	5101	Time limit is exceeded
FILE_ABORT	5102	Order has been aborted by
FILE_HANDLE_INVALID	5103	Invalid handle
FILE_NOT_EXIST	5104	Directory or file does not exist
FILE_EXIST	5105	Directory or file already exists
FILE_NO_MORE_ENTRIES	5106	No further entries are available
FILE_NOT_EMPTY	5107	File or directory is not empty
FILE_READ_ONLY_CAA	5108	File or directory is write protected
FILE_WRONG_PARAMETER	5109	Wrong parameters detected
FILE_ERROR_UNKNOWN	5110	Unknown detected error
FILE_WRITE_INCOMPLETE	5111	Not all data have been written
FILE_NOT_IMPLEMENTED	5112	Not implemented
FILE_FIRST_MF	5150	First detected error (manufacturer specific)
FILE_LAST_ERROR	5199	Last detected error (library specific)

Part VIII
Commonly Used Sub-Structures

Chapter 21
Commonly Used Sub-Structures

Sub-Structure Descriptions

Overview

The Conveying Library system consists of function blocks basing and interacting on pre-defined interfaces. These interfaces consist of variables used as structures for the different types (input, output, input / output).

In the previous chapters the function blocks have been described down to the next layer, below the input or output pin - which means that only the main structures have been described.

As some of these structures are also using sub-structures, the description of those follow in this chapter, as commonly used sub-structures.

- `stEqupHmiCmd` *(see page 312)*
- `stCtrl` *(see page 312)*
- `stConvInitDir` *(see page 312)*
- `stManCmd` *(see page 313)*
- `byMotSel` *(see page 313)*
- `stMod` *(see page 313)*
- `stModStat` *(see page 314)*
- `stConvSen` *(see page 314)*
- `stErcSen` *(see page 314)*
- `stMotStat` *(see page 314)*
- `stMotFbck` *(see page 314)*
- `stRemMod` *(see page 315)*
- `stTtblSen` *(see page 315)*
- `stBecn` *(see page 315)*
- `stMotCmd` *(see page 315)*
- `stConfOfHdskSwap` *(see page 316)*
- `stInitOfSenCtrl` *(see page 316)*
- `stInitOfLoadCtrl` *(see page 316)*
- `stInitOfTurnMovCtrl` *(see page 317)*
- `stInitOfUnldCtrl` *(see page 317)*
- `stStmInit` *(see page 318)*
- `stInitOfVertMotCtrl` *(see page 318)*
- `stInitOfMotCtrl` *(see page 318)*
- `stConvSenFltr` *(see page 319)*

- stTtblSenFltr (see page 319)
- stErcSenFltr (see page 319)
- stHdsk (see page 319)
- stLoad (see page 319)
- stUnld (see page 320)
- ErcTtblPara (see page 320)
- stErcTtblStmInit (see page 320)

stEqupHmiCmd

Structure Parameter	Data Type	Description
stMod	stMod	Mode structure
stManCmd	stManCmd	Manual command structure
stCtrl	stCtrl	Deskzone control structure
stConvInitDir	stConvInitDir	Conveyor init direction
xRstAlrm	BOOL	Reset alarm command from Deskzone
xCyclRst	BOOL	Restart cycle command from HMI through Deskzone

stCtrl

Structure Parameter	Data Type	Description
wZoneIdSel	WORD	Selected zone ID [user defined]
wEqupIdSel	WORD	Selected equipment ID [system defined]
ebyEqupTypeSel	EqupType	Selected equipment type [system defined]

stConvInitDir

Structure Parameter	Data Type	Description
xFwd	BOOL	Conveyor preference to run forward
xRev	BOOL	Conveyor preference to run reverse

stManCmd

Structure Parameter	Data Type	Description
byMotSel	BYTE	Motor selection
xFwd	BOOL	Forward direction selected
xRev	BOOL	Reverse direction selected
xSlow	BOOL	Slow operation enabled
xMotHrsOfOpRst	BOOL	Reset counter for hours of motor operation

byMotSel

The motor is selected based on the bit and the equipment type used. A manual command, e.g. a move in forward or reverse direction, requires, besides the command itself, the selection of 1 motor. If more than 1 motor or more than 1 direction is selected, all selections and commands will be ignored without raising any alarm or alert message. Up to 3 motors per equipment (ERC) can be controlled, but only 1 motor at a time. The value of the variable byMotSel selects the motor which should be controlled in manual mode.

The following table describes the range of values and the selected motor based on the value and the Equipment application function block.

AFB	Motor		Value of byMotSel	
	Number	Function	Binary	Decimal
Conveyor	1	Transfer rev <-> fwd	2#0001	1
Input	1	Transfer rev <-> fwd	2#0001	1
Output	1	Transfer rev <-> fwd	2#0001	1
TurnTable	1	Transfer rev <-> fwd	2#0001	1
	2	Turnmove	2#0010	2
ERC	1	Transfer 180° <-> 360°	2#0001	1
	2	Transfer 270° <-> 90°	2#0010	2
	3	Lifting	2#0100	4

stMod

Structure Parameter	Data Type	Description
xAuto	BOOL	Auto mode
xMan	BOOL	Manual mode

stModStat

Structure Parameter	Data Type	Description
xAuto	BOOL	Auto mode
xMan	BOOL	Manual mode
xLoc	BOOL	Local mode
xMain	BOOL	Maintenance mode
xManModInPrgs	BOOL	Manual mode in progress

stConvSen

Structure Parameter	Data Type	Description
xPltPrsFwd	BOOL	Pallet present forward
xPltPrsRev	BOOL	Pallet present reverse
xPltSpdChgFwd	BOOL	Pallet speed change forward
xPltSpdChgRev	BOOL	Pallet speed change reverse

stErcSen

Structure Parameter	Data Type	Description
xPos180And360Deg	BOOL	Positioned for 180 and 360 degree
xPos90And270Deg	BOOL	Positioned for 90 and 270 degree

stMotStat

Structure Parameter	Data Type	Description
xDrvFlt	BOOL	Drive error detected
xMotSwc	BOOL	Status of motor switch

stMotFbck

Structure Parameter	Data Type	Description
xFwd	BOOL	Forward command
xRev	BOOL	Reverse command

stRemMod

Structure Parameter	Data Type	Description
xLoc	BOOL	Local mode
xMain	BOOL	Maintenance mode

stTtblSen

Structure Parameter	Data Type	Description
xPos1Inlk	BOOL	Position 1 interlock
xPos2Inlk	BOOL	Position 2 interlock
xPos180And360Deg	BOOL	180 & 360 degree position
x180And360DegSpdChg	BOOL	180 & 360 degree speed change
xPos90And270Deg	BOOL	90 & 270 degree position
x90And270DegSpdChg	BOOL	90 & 270 degree speed change

stBecn

Structure Parameter	Data Type	Description
xRed	BOOL	Beacon lamp red
xYell	BOOL	Beacon lamp yellow
xGreen	BOOL	Beacon lamp green

stMotCmd

Structure Parameter	Data Type	Description
xSel	BOOL	Selected
xFwd	BOOL	Forward command
xRev	BOOL	Reverse command
xFast	BOOL	Fast command
xSlow	BOOL	Slow command

stConfOfHdskSwap

Structure Parameter	Data Type	Description
x360DegEn	BOOL	Handshake swap 360 degree enable
x90DegEn	BOOL	Handshake swap 90 degree enable
x180DegEn	BOOL	Handshake swap 180 degree enable
x270DegEn	BOOL	Handshake swap 270 degree enable

stInitOfSenCtrl

Structure Parameter	Data Type	Description
xSenAddEn	BOOL	Additional sensor enable
tFltr	TIME	Filter time Range: 0...2 s Factory setting: 0 s
bySntyFailCnt	BYTE	Sensitivity fail detection count Range: 1...10 Factory settings: 3
xBothPltDeteAlrtDsbl	BOOL	Disables the alert generation on detection of both pallet presence sensors (fwd & rev) in loaded state. Parameter is used only by the Conveyor function block.

stInitOfLoadCtrl

Structure Parameter	Data Type	Description
xLdngAlrmDsbl	BOOL	Disables the loading alarm
tLoadCtrlFwd	TIME	Load control forward Range: 5 s...2 minutes Factory settings: 1 minute
tLoadCtrlRev	TIME	Load control reverse Range: 5 s...2 minutes Factory settings: 1 minute
tDlyToSpdChgFwd	TIME	Delay to speed change forward Range: 0…20 s Factory settings: 0 s
tDlyToSpdChgRev	TIME	Delay to speed change reverse Range: 0…20 s Factory settings: 0 s
tMotDlyToStopFwd	TIME	Motor delay to stop forward direction Range: 0…5 s Factory settings: 0 s

Structure Parameter	Data Type	Description
tMotDlyToStopRev	TIME	Motor delay to stop reverse direction Range: 0...5 s Factory settings: 0 s
tStagFnshCtrl	TIME	Stage finish control Range: 100 ms…3 minutes Factory settings: 100 ms

stInitOfTurnMovCtrl

Structure Parameter	Data Type	Description
tTimeToTurn	TIME	Normal time to finish turning Range: 5 s...2 minutes Factory settings: 1 minute
t180DegDlyToTurnStop	TIME	Delay to turn stop 180 degree Range: 0...5 s Factory settings:0 s
t90DegDlyToTurnStop	TIME	Delay to turn stop 90 degree Range: 0...5 s Factory settings: 0 s
t90DegDlyToSpdChgTurn	TIME	Speed change turn 180 degree Range: 0…5 s Factory settings: 0 s
t180DegDlyToSpdChgTurn	TIME	Delay to speed change reverse Range: 0...5 s Factory settings: 0 s

stInitOfUnldCtrl

Structure Parameter	Data Type	Description
xUldgAlrmDsbl	BOOL	Disables unloading alarm
tUnldCtrlFwd	TIME	Timer for unloading operation in forward direction Range: 5 s...2 minutes Factory settings: 1 minute
tUnldCtrlRev	TIME	Timer for unloading operation in forward direction Range: 5 s...2 minutes Factory settings: 1 minute
tStagFnshCtrl	TIME	Control timer for finishing stage control Range: 100 ms...3 s Factory settings: 100 ms

stStmInit

Structure Parameter	Data Type	Description
xPrldAlrtDsbl	BOOL	Disables pre-load alert
tPrldCtrl	TIME	Pre-load control Range: 0...20 s Factory settings: 0 s (disabled)
tMotDlyToStrt	TIME	Motor delay to start Range: 1...5 s Factory settings: 1 s
tMotDlyToSwcOver	TIME	Motor delay to switch over Range: 1...5 s Factory settings: 1 s
tDlyToFlowCnst	TIME	Constant flow control Range: 0...60 s Factory settings: 0 s

stInitOfVertMotCtrl

Structure Parameter	Data Type	Description
tTimeToVertMov	TIME	Time to vertical move Range: 5 s...2 minutes Factory settings: 1 minute
tDlyToVertMovStopUp	TIME	Delay to vertical move up stop Range: 0...5 s Factory settings: 0 s
tDlyToVertMovStopDown	TIME	Delay to vertical move down stop Range: 0...5 s Factory settings: 0 s

stInitOfMotCtrl

Structure Parameter	Data Type	Description
xFbckAlrmDsbl	BOOL	Disables the motor feedback alarm
tFbck	TIME	Defines the feedback time Range: 500 ms...5 s Factory settings: 3 s

stConvSenFltr

Structure Parameter	Data Type	Description
xSenAddEn	BOOL	Enables additional sensor
stConvSen	stConvSen	Structure composed of Conveyor sensors (position and speed change sensors)
xIpcnvOpcnvRele	BOOL	Input Conveyor and output Conveyor release
xIpcnvOpcnvAckn	BOOL	Input Conveyor and output Conveyor acknowledge

stTtblSenFltr

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Structure composed of Conveyor sensors (position and speed change sensors)
stTtblSen	stTtblSen	Structure composed of Turntable sensors (position and speed change sensors)

stErcSenFltr

Structure Parameter	Data Type	Description
stConvSen	stConvSen	Structure composed of Conveyor sensors (position and speed change sensors)
stErcSen	stErcSen	Structure composed of ERC sensors (position sensors)

stHdsk

Structure Parameter	Data Type	Description
stUnld	stUnld	Structure of signals used for the unloading process
stLoad	stLoad	Structure of signals used for the loading process
xAuto	BOOL	TRUE: Automatic mode is enabled FALSE: Automatic mode is disabled
ebyType	EqupType	Equipment type [system defined]

stLoad

Structure Parameter	Data Type	Description
xLoadRdy	BOOL	Equipment indicates readiness to load
xStagFnsh	BOOL	Equipment confirms end of loading process

stUnld

Structure Parameter	Data Type	Description
xUnldRdy	BOOL	Equipment indicates readiness to unload
xAcknStagFnsh	BOOL	Equipment confirms end of unloading process
xPreUnld	BOOL	Preparation to unload and initiating the switching on of the succeeding conveyor roller motor equipment for taking over the pallet

stERCTtblPara

Structure Parameter	Data Type	Description
wFrstPrty	WORD	First priority address (90,180, 270 or 360) Factory settings: 90 degree
wScndPrty	WORD	Second priority address (90,180, 270 or 360) Factory settings: 360 degree
wThrdPrty	WORD	Third priority address (90,180, 270 or 360) Factory setting: 270 degree
x360DegStanEn	BOOL	360 degree station enabled
x90DegStanEn	BOOL	90 degree station enabled
x180DegStanEn	BOOL	180 degree station enabled
x270DegStanEn	BOOL	270 degree station enabled
wHomeAddr	WORD	Home position address (90, 180, 270 or 360) Factory setting: 360 degree
xHomeAddrEnBack	BOOL	Home position enable back
tDlyToHome	TIME	Back to home position delay Range 0...10 s

stErcTtblStmInit

Structure Parameter	Data Type	Description
stStm	stStmInit	State machine initialization
stErcTtblPara	stErcTtblPara	Parameters used for ERC and TurnTable

Part IX
Troubleshooting

Chapter 22
Troubleshooting

Troubleshooting

Note

We assume that SoMachine is well known in terms of programming languages, configuration and operating modes for Controllers and HMIs. If not, please refer to the dedicated literature.

Programming or Software Configuration Issues

This table describes the software configuration issues and their solutions:

Issue	Cause	AFB impacted	Solution
Function block is instantiated, but is not working.	Function block might not have been enabled.	All function blocks using an `En` input.	Turn on `En` input. This enables the function block.
Tracking is not working.	Connection from `TrckSimp` or `TrckCmpx` function block to according equipment function block is not done.	• `TrckCmpx` • `TrckSimp`	Connect the function block as described in the graphic.
Alarm and/or alert messages are not working.	Connection from equipment function block to `AlrmHdlg` function block is not done.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable` • `TrckCmpx` • `TrckSimp` • `DeskzoneCmpx` • `DeskzoneSimp`	Connect the function block as described in the graphic.
Alarm and/or alert messages are not displayed.	The alarm texts are not defined in Vijeo Designer.	All Conveying library function blocks.	Define the alarm text messages in Vijeo Designer.
Alarm and/or alert messages are not displayed in the correct language.	The alarm texts are not defined in the correct language in Vijeo Designer.	All Conveying library function blocks.	Define the alarm text messages in Vijeo Designer in the correct language.

Issue	Cause	AFB impacted	Solution
Link to the UME90° is not working.	Connection from equipment function block to equipment in 90° is not done.	• Conveyor • Input • Output • ERC • TurnTable	Connect the function block after description.
Link to the UME180° is not working.	Connection from equipment function block to equipment in 180° is not done.	• Conveyor • Input • Output • ERC • TurnTable	Connect the function block as described in the graphic.
Link to the UME270° is not working.	Connection from equipment function block to equipment in 270° is not done.	• Conveyor • Input • Output • ERC • TurnTable	Connect the function block as described in the graphic.
Link to the UME360° is not working.	Connection from equipment function block to equipment in 360° is not done.	• Conveyor • Input • Output • ERC • TurnTable	Connect the function block as described in the graphic.
Handshake Logic Alarm	From the preceding/succeeding equipment AFB, the Ready to Load or Ready to Unload signal is missing.	• Conveyor • Input • Output • ERC • TurnTable	Check the handshake structure wiring to the preceding/succeeding equipment AFB.
Time Out Unload Forward Alarm	The timer has already elapsed, but the Pallet Presence Sensor is still TRUE.	• Conveyor • Input • Output • ERC • TurnTable	Change the configured value of the timer.
Time Out Unload Reverse Alarm	The timer has already elapsed, but the Pallet Presence Sensor is still TRUE.	• Conveyor • Input • Output • ERC • TurnTable	Change the configured value of the timer.
Collision Detection Alarm	The destination conveyor unloads the pallet into the opposite direction.	• Conveyor • Input • Output • ERC • TurnTable	Check the handshake structure wiring to the preceding/succeeding equipment AFB.

Issue	Cause	AFB impacted	Solution
Material Flow Inconsistency Alarm	The conveyor is loaded and the destination conveyor is not able to receive the pallet.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	• Check the build up of the application. • Check the handshake structure wiring to the preceding/succeeding equipment AFB.
Tracking source address is invalid.	The source address in the tracking data field is not valid.	• `TrckCmpx` • `TrckSimp`	Correct the address in the tracking field.
Tracking destination address is invalid.	The destination address in the tracking data field is not valid.	• `TrckCmpx` • `TrckSimp`	Correct the address in the tracking field.
Tracking destination pointer is invalid.	The pointer to the data field in the tracking array is not valid.	• `TrckCmpx` • `TrckSimp`	Correct the pointer.
Tracking End Sign Alarm	If the equipment is loaded, the function block detects that the end sign [F0F0] is not valid in the tracking data field.	• `TrckCmpx` • `TrckSimp`	Check the tracking data field and revise the end sign if it is not F0F0.
Destination Address Alarm	During the loading process, the function block detects that the destination address is not valid.	• `TrckCmpx` • `TrckSimp`	• Check the destination address coming from the equipment is correct. • Check the configured pointer to the destination address in the tracking data field is correct.
Length of Array Parameter Alarm	During the initializing phase, the function block detects the tracking array is too low or too high.	• `TrckCmpx` • `TrckSimp`	Check the parameter size of the tracking data field and correct. Data field: 0...49 words
Equipment Type Parameter Alarm	The equipment type is wrong.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable` • `TrckCmpx` • `TrckSimp`	Revise the equipment type. The documentation helps to choose the correct type.

Issue	Cause	AFB impacted	Solution
System in manual mode and it is not possible to be controlled by the `DeskzoneSimp` or `DeskzoneCmpx` function block.	During manual mode, the equipment cannot be controlled from HMI through `DeskzoneSimp` or `DeskzoneCmpx` function block.	• `DeskzoneSimp` • `DeskzoneCmpx`	• Check the data transfer from `DeskzoneSimp` or `DeskzoneCmpx` function block to the HMI is done [Life sign signal]. • Check the equipment command structure is connected to the equipment.
System in auto mode and free of alarms, but the conveyor is not running.	The equipment has a pallet on the conveyor, but cannot be unloaded.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	• Check the handshake structure. • Check the controller is in Run mode.
Speed Change Timer Configuration Alarm in load process.	The timer has already elapsed, but the sensor speed change is not reached.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Change the configured value of the timer.
Timer Turn Move Elapsed Alarm	The timer has already elapsed, but the Turn Position Sensor is not reached.	`TurnTable`	Change the configured value of the timer.
Timer Vertical move Elapsed Alarm	The timer has already elapsed, but the Vertical Move Position Sensor is not reached.	`ERC`	Change the configured value of the timer.
Time Out Load Alarm	The timer has already elapsed, but the Pallet Presence Sensor is not reached.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Change the configured value of the timer.
Function block is never Ready to Load	The signal Ready to Load is not provided to the preceding equipment.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Check the handshaking structure is connected properly.

Other issues

This table describes some solutions for problems that occur during installation. Ensure that you have checked the facility is operational both in terms of mechanical and electrical services.

Issue	Cause	AFB impacted	Solution
Speed Change Timer Configuration Alarm in Load Process	The timer has already elapsed, but the sensor speed change is not reached.	• Conveyor • Input • Output • ERC • TurnTable	Change the sensor installation position to reach the signal earlier.
Timer Turn move Elapsed Alarm	The timer has already elapsed, but the Turn Position Sensor is not reached.	TurnTable	Change the sensor installation position to reach the signal earlier.
Timer Vertical move Elapsed Alarm	The timer has already elapsed, but the Vertical Move Position Sensor is not reached.	ERC	Change the sensor installation position to reach the signal earlier.
Time Out Load Alarm	The timer has already elapsed, but the Pallet Presence Sensor is not reached.	• Conveyor • Input • Output • ERC • TurnTable	Change the sensor installation position to reach the signal earlier.
Function block is never Ready to Load.	The signal Ready to Load is not provided to the preceding equipment.	• Conveyor • Input • Output • ERC • TurnTable	Check the sensor signal is not permanent on 24 V.
Local stop is not working	The local stop button is not connected as required.	• Conveyor • Input • Output • ERC • TurnTable	Connect the local stop button as described in the graphics.
System doesn't start	Controller must start.	All function blocks using an En input.	Start the controller [Ctrl+F5].
Local stop Activated Alarm	Local stop input is turned off.	• Conveyor • Input • Output • ERC • TurnTable	Check the local stop input is high.
Motor Feedback Alarm	Motor did not start or did not stop.	• Conveyor • Input • Output • ERC • TurnTable	Check the hardware connection from OTB to ATV.

Issue	Cause	AFB impacted	Solution
Pallet detected in Empty State Alarm	One of the sensors detects a pallet, but the function block expects to be empty.	• `Conveyor` • `Input` • `Output`	Remove the pallet and acknowledge the alarm or start the cycle restart.
	This alarm may occur in any of the below state of equipment: • In Empty state - Any one of the sensor detects a pallet. • In position change state (Homing) - When the equipment performs cycle reset state, it checks for the equipment station's current position. It can be either in 90°/360°. If a failure is detected it tries to reach the home position. During the homing operation the pallet sensors are monitored. In case of any sensor detection this alarm will be generated.	• `TurnTable` • `ERC`	

Issue	Cause	AFB impacted	Solution
Pallet detection lost in Loaded State Forward Alarm.	The Pallet Presence Sensor in forward direction detects no pallet, but the function block expects to have a pallet on the conveyor.	• `Conveyor` • `Input` • `Output`	Move the pallet on the sensor and acknowledge the alarm or start the cycle restart.
Pallet detection lost in Loaded State.	This alarm may occur in any of the below state of equipment: • In Loaded state - The forward and reverse pallet presence sensors detects no pallet. • In position change state (Homing) - When the equipment performs cycle reset state, it checks for the equipment station's current position. It can be either in 90°/360°. If a failure is detected it tries to reach the home position. During the homing operation the pallet sensors are monitored to be ON. In case of an detected failure of both forward and reverse pallet presence sensors this alarm will be generated.	• `TurnTable`	
Pallet detection lost in Loaded State.	This alarm may occur in any of the below state of equipment: • In Loaded state - The forward and reverse pallet presence sensors detects no pallet. • In position change state (Homing) - When the equipment performs cycle reset state, it checks for the equipment station's current position. It can be either in 90°/360°. If a failure is detected it tries to reach the home position. During the homing operation the pallet sensors are monitored to be ON. In case of an detected failure of any sensor this alarm will be generated.	• `ERC`	
Sensitive Pallet Presence Sensor Alarm	The Pallet Presence Sensor jitters.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensor detection fails during Loaded or Empty State Alarm	In loaded state, the Pallet Presence Sensor signals is FALSE or in empty state the Pallet Presence Sensor signal is TRUE.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Verify the wiring.

Issue	Cause	AFB impacted	Solution
Sensitive 180° Position Sensor Alarm	The Position Sensor jitters.	• ERC • TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitive 90° Position Sensor Alarm	The Position Sensor jitters.	• ERC • TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitive 180° Speed Change Sensor Alarm	The Position Sensor jitters.	• ERC • TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitive 90° Speed Change Sensor Alarm	The Position Sensor jitters.	• ERC • TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitive Interlock Alarm	One Interlock Sensor jitter	• TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Turntable Position Lost Alarm	The Position Sensor jitters.	• TurnTable	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Vertical Position Lost Alarm	The Position Sensor jitters.	• ERC	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitivity Position Up Alarm	The Position Sensor jitters.	• ERC	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensitivity Position Down Alarm	The Position Sensor jitters.	• ERC	Verify that the sensor and the cable are wired to the remote or distributed I/O module.
Sensor Sequence Alarm.	The Pallet Presence Sensor is TRUE before the Speed Change Sensor turns TRUE.	• Conveyor • Input • Output • ERC • TurnTable	Check the wiring whether the signals have to be interchanged.
Turn Move Feedback Alarm.	The motor for turning moves is running, but no feedback information is received.	• Conveyor • Input • Output • ERC • TurnTable	• Check the motor status. • Check the wiring from OTB to drive.
Unable to reset alarms.	The alarm cannot reset.	All Conveying library function blocks.	The alarm stays because the reason of the alarm is not fixed.
Emergency Stop is released.	The Emergency Stop signal is missing.	AlrmHdlg	Check that the Emergency Stop button is not pressed.

Issue	Cause	AFB impacted	Solution
Fieldbus is not available	The fieldbus information from the system indicates that either a node is not present or functioning correctly, or that the fieldbus itself is not in an operational state.	All function blocks.	Verify that the fieldbus cabling is correctly wired and that the fieldbus devices are operating correctly.
Fieldbus Node is not available.	Signal is missing.	• `Conveyor` • `Input` • `Output` • `ERC` • `TurnTable`	Check the OTB module.
Power supply preventa fault	The signal from the Preventa power supply is FALSE.	`AlrmHdlg`	Verify the Preventa and the wiring from the Preventa to the OTB.
Fuse released power supply 400 V: Box 1-10	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.
Fuse released power supply 400 V: Box 11-20	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.
Fuse released power supply 400 V: Box 21-30.	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.
Fuse released power supply 24 V: Box 1-10	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.
Fuse released power supply 24 V: Box 11-20.	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.
Fuse released power supply 24 V: Box 21-30.	The signal from the auxiliary contact from the fuses are FALSE.	`AlrmHdlg`	Verify that the fuses are On.

Issue	Cause	Device impacted	Solution
Download to Controller not possible.	Retain memory of controller is exceeded.	`Controller`	Reduce the amount of equipments used in your Controller application in order to reduce the retain memory size. Details on existing limitations are given in chapter *Conveying Library System*.

Issue	Cause	Device impacted	Solution
Download to HMI is not possible.	Total amount of variables in HMI application is exceeded. (Magelis is able to manage up to 8000 variables for internal and external use.)	`HMI`	2 Solutions: ● Split the application by adding a second HMI connected to the same controller. ● Reduce the amount of equipments used in your HMI application in order to reduce the total amount of variables used. Details on existing limitations are given in chapter *Conveying Library System*.
Not all equipments can be supported by a single alarm group.	Limit on alarm groups is exceeded (Magelis is supporting 32 alarm groups).	`HMI`	2 Solutions: ● Split the application by adding a second HMI connected to the same controller. ● Reduce the amount of equipments used in your HMI application in order to reduce the amount of alarm groups. Details on existing limitations are given in chapter *Conveying Library System*.

Glossary

A

AFB

(*application function block*)

application

A program including configuration data, symbols, and documentation.

ARRAY

The systematic arrangement of data objects of a single type in the form of a table defined in logic controller memory. The syntax is as follows: `ARRAY [<dimension>] OF <Type>`

Example 1: `ARRAY [1..2] OF BOOL` is a 1-dimensional table with 2 elements of type `BOOL`.

Example 2: `ARRAY [1..10, 1..20] OF INT` is a 2-dimensional table with 10 x 20 elements of type `INT`.

ASCII

(*American standard code for Information Interchange*) A protocol for representing alphanumeric characters (letters, numbers, certain graphics, and control characters).

B

byte

A type that is encoded in an 8-bit format, ranging from 00 hex to FF hex.

C

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).

E

element

The short name of the ARRAY element.

ERC

(*eccentric roller conveyor*) Also referred as a transfer table.

expansion bus

An electronic communication bus between expansion I/O modules and a controller.

F

FB

(*function block*) A convenient programming mechanism that consolidates a group of programming instructions to perform a specific and normalized action, such as speed control, interval control, or counting. A function block may comprise configuration data, a set of internal or external operating parameters and usually 1 or more data inputs and outputs.

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)

I

I/O

(*input/output*)

ID

(*identifier/identification*)

input/output

The index of the ARRAY.

M

machine

Consists of several *functions* and/or *equipment*.

ms

(*millisecond*)

N

nibble

A half-byte (representing 4 bits of a byte).

node

An addressable device on a communication network.

P

pallet

A portable platform, which is used for storing or moving goods.

program

The component of an application that consists of compiled source code capable of being installed in the memory of a logic controller.

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

scan

A function that includes:

- reading inputs and placing the values in memory
- executing the application program 1 instruction at a time and storing the results in memory
- using the results to update outputs

STOP

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

U

UME

(*unitary mechanical equipment*) Equipment that is controlled by 1 function block (for example, a conveyor, input, output, turntable, or ERC).

W

WORD

A type encoded in a 16-bit format.

Index

A

alarm handling, *43*, *279*
AlrmHdlg, *281*
AlrmHdlgTextFile, *299*

C

Conveying
- alarm handling, *43*
- AlrmHdlg, *281*
- AlrmHdlgTextFile, *299*
- complex conveying line, *92*
- Conveyor, *97*
- DeskzoneCmpx, *165*
- DeskzoneSimp, *179*
- equipment application function blocks, *34*
- equipment types transverse AFB, *31*
- ERC, *193*
- function block location in the Conveying Library, *21*
- general description, *23*
- Input, *119*
- operating mode for equipment AFBs, *38*
- Output, *141*
- simple conveying line, *88*
- system requirements, *18*
- TrckCmpx, *241*
- TrckSimp, *261*
- troubleshooting, *323*
- TurnTable, *217*

Conveyor, *97*

D

DeskzoneCmpx, *165*
DeskzoneSimp, *179*

E

equipment
- equipment application function blocks, *34*
- equipment types transverse AFB, *31*

ERC, *193*

I

Input, *119*

O

Output, *141*
- Function Block Description, *145*
- Functional and System Overview, *142*
- Pin Description, *146*

S

system requirements, *18*

T

TrckCmpx, *241*
TrckSimp, *261*
TurnTable, *217*