Premium and Atrium Using Unity Pro

Electronic Cam Module

User Manual

07/2012

35006228.09

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 that is 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.

© 2012 Schneider Electric. All rights reserved.

Table of Contents

Safety Information **9**
About the Book **11**
Part I Cam module TSX CCY 1128 . . . . **13**
Chapter 1 Introduction to the electronic cam module **15**
General introduction to the electronic cam module 16
Operation of the electronic Cam Module 17
Position reading 18
Cam processing operation 19
Structuring the cam processes 22
Interface with the PLC program 24
Software installation functions for the module 26
General ergonomics for the module implementation functions 28
Installation Phase Overview 30
Chapter 2 Introduction to the TSX CCY 1128 electronic cam module **33**
Description of the TSX CCY 1128 in its environment 34
Physical description of the TSX CCY 1128 35
Electronic cam function of the TSX CCY 1128 37
Chapter 3 Installation methodology **39**
Methodology for setting up module software 39
Chapter 4 Types of applications **41**
Areas of application 42
Alternating movement 43
Use in alternating motion 45
Rotary movements 46
Use for Rotary Motion 47
Cyclic movements 48
Use for cyclic motion 50
Continuous movement 52
Use for continuous motion 53

Chapter 5 Examples 55
5.1 Simple example 56
Introduction to a simple example 57
Entering configuration parameters. 58
Programming 61
5.2 Advanced example 64
Introduction to a detailed example. 65
Operating mode 67
The Recipes. 68
Entering configuration parameters. 69
Programming 70
Special Features 76
Part II Hardware installation of the module TSX CCY 1128 77
Chapter 6 General instructions for installing the TSX CCY 1128 module 79
Installing the TSX CCY 1128 in a PLC station rack. 80
Installing the TSX CCY 1128 in a PLC station. 81
Number of application-specific channels managed by one PLC station . 83
Installation precautions for the TSX CCY 1128 84
General wiring instructions. 85
Selecting and protecting auxiliary power supplies 86
Choice of encoders for the TSX CCY 1128 87
Chapter 7 Connecting an incremental and absolute SSI encoder to the TSX CCY 1128 91
Principles for connecting an encoder to a TSX CCY 1128 92
Connecting an incremental encoder with RS422 outputs to the TSX CCY 1128. 93
Connecting an incremental encoder with Totem Pole outputs to the TSX CCY 1128 96
Connecting an absolute SSI encoder to the TSX CCY 1128. 99
Connecting an encoder supply monitor to the TSX CCY 1128 102
Connecting the encoder supply to the TSX CCY 1128 104
TSX CAP S15 connection accessory. 107
TSX TAP S1505/S1524 and TSX CCP S15x Connecting Accessories. . 108
Chapter 8 Connecting the auxiliary inputs and track outputs of the TSX CCY 1128 113
Introduction to the connection interfaces of the TSX CCY 1128 114
Connecting the auxiliary inputs of the TSX CCY 1128 117
Connecting the track outputs of the TSX CCY 1128 125

Chapter 9 Connecting an absolute encoder with parallel outputs to TSX CCY 1128 133
Principle for connecting an absolute encoder to a TSX CCY 1128 134
TELEFAST ABE-7CPA11 base. 135
Pinout configuration of the 15-pin SUB-D connectors of the module and the TELEFAST 138
Connecting an absolute encoder with parallel outputs 140
Wiring rules and precautions specific to the TELEFAST. 143
Configuration of the TELEFAST connector. 147
Chapter 10 TSX CCY 1128 module displays 149
10.1 Electrical properties of the TSX CCY 1128 module 150
General electrical characteristics of the TSX CCY 1128. 151
Characteristics of the auxiliary inputs of the TSX CCY 1128 152
Characteristics of the encoder supply feedback monitor for the TSX CCY 1128 153
Characteristics of the auxiliary inputs of the TSX CCY 1128 154
Characteristics of the track outputs of the TSX CCY 1128 155
10.2 Displays of the TSX CCY 1128 module 156
Introduction to the display block of the TSX CCY 1128 module 157
The various states of the LEDs on the TSX CCY 1128 and their meaning 158
Part III Software installation cam TSX CCY 1128 159
Chapter 11 Configuration of the electronic cam module 161
Accessing the module configuration parameters 162
Configuring the electronic cam parameters. 163
Configuring acquisition parameters. 164
Configuring an incremental encoder 165
Configuration of an absolute encoder 167
Reading format configuration 169
Configuring the position recalibration function for incremental encoders . 170
Capture function configuration 171
Cam processor configuration 172
Connector configuration 174
Confirmation of the configuration 175
Chapter 12 Programming 177
Synopsis of the module functions 179
Confirmation of the axis functions 181
Confirming the cam processor functions. 183
Confirming the events 184
Overview of event management 185
Language interface 186
Processor and module exchanges 187
System exchanges 188
WRITE_PARAM : Transfer of current recipe parameters 190
READ_PARAM : Transfer of current recipe parameters. 192

RESTORE_PARAM : Transferring initial parameters 193
SAVE_PARAM : Transferring initial parameters 194
MOD_PARAM: Axis adjustment 195
MOD_TRACK: Adjusting a track 198
MOD_CAM: Adjusting a cam 201
TRF_RECIPE: Recipe transfer functions. 204
TRF_RECIPE: Functions for storing recipes 205
TRF_RECIPE: Loading a new recipe 206
TRF_RECIPE: Saving a new 208
DETAIL_OBJECT : Operator dialogue interface 210
DETAIL_OBJECT: Transferring cam details 211
DETAIL_OBJECT: Transferring track details. 214
Reconfiguration in connected mode 217

Chapter 13 Entering recipe adjustment parameters for the electronic cam module 219

Accessing the recipe adjustment parameters for the module 220
Entering acquisition parameters for an incremental encoder. 221
Entering the acquisition parameters for an absolute encoder 223
Parametering for the parts counter 225
Activating/deactivating the tracks. 226
Parametering for the tracks 227
Creating cams 229
Parametering the cams 230
Cam in position 231
Monostable cam. 234
Breaking cam 235
Parametrizing the confirmation condition associated with a cam 236
Confirming the recipe adjustment parameters. 237
Saving the recipe adjustment parameters 238
Restoring the recipe adjustment parameters. 239

Chapter 14 Debugging and adjusting 241

Description of the debug screen 242
Description of the main debugging zone 243
Description of the debugging zone: "Acquisition" 244
Description of the debugging zone: "Parts counter" 246
Description of the adjustment zone 247
Description of the debug zone: "Group x" 250
Description of the adjustment screen. 252

Chapter 15 Diagnostics. 255

Module status level 256
Channel status level. 257
Error codes 259
Controls 262
Checking the integrity of the module 263

Monitoring the encoder 264
Checking auxiliary inputs. 266
Monitoring track outputs 267
Questions and Answers 269
Chapter 16 Performance and limitations 271
Global precision in controlling the actuators 272
Actuator commands 275
General time-division performance characteristics 279
Functional limitations. 281
Chapter 17 The language objects of the independent axes application. 283
Description of the language objects of the axis application-specific function 284
Implicit Exchange Language Objects Associated with the Application-Specific Function. 285
Explicit Exchange Language Objects Associated with the Application-Specific Function. 286
Management of Exchanges and Reports with Explicit Objects. 288
Details of the Language Objects of the T_GEN_MOD-Type IODDT 292
Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP0 293
Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP0 297
Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3 306
Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3 307
Configuration constants. 311
Glossary 315
Index 325

Safety Information

Important Information

NOTICE

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

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

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

DANGER

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

WARNING

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

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

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

PLEASE NOTE

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

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

About the Book

At a Glance

Document Scope

This manual describes the installation of the electronic cam application in the Premium and Atrium PLCs, from Unity Pro software.

Validity Note

This documentation is valid for Unity Pro from version 7.0.

Product Related Information

WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this product.

Follow all local and national safety codes and standards.

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

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Cam module TSX CCY 1128

I

Aim of this part

This part concerns the cam module TSX CCY 1128.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Introduction to the electronic cam module	15
2	Introduction to the TSX CCY 1128 electronic cam module	33
3	Installation methodology	39
4	Types of applications	41
5	Examples	55

Introduction to the electronic cam module

1

Subject of this chapter

This chapter presents the main characteristics and functions of the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
General introduction to the electronic cam module	16
Operation of the electronic Cam Module	17
Position reading	18
Cam processing operation	19
Structuring the cam processes	22
Interface with the PLC program	24
Software installation functions for the module	26
General ergonomics for the module implementation functions	28
Installation Phase Overview	30

General introduction to the electronic cam module

Introduction

The TSX CCY 1128 electronic cam module is designed to control 24 discrete physical outputs independently and with a very short response time (<0.2ms).

The TSX CCY 1128 module operates on a Premium PLC (software version 3.3 or above).

The software installation is performed using the Unity Pro software.

Areas of application

The TSX CCY 1128 module can process applications with the following types of movement:

- non-reversing rotary (eg.: mechanical presses)
- alternating (eg.: hydraulic presses, transfer machines)
- cyclical, with periodic entry of parts to be processed (eg.: packaging machines)
- continuous, with random entry of parts to be processed (eg.: conveyors)

Main characteristics

This table summarizes the main functional characteristics of the electronic cam module.

Characteristic	Value
Number of cams	128 maximum
Number of tracks	32 (24 directly associated with the 24 physical outputs, 8 logical)
Position encoder inputs	incremental or absolute
Outputs controlled	24 discrete outputs 24V, 0.5A
Types of cam	position, monostable, brake
Associated Functions	taking up axis backlash, recalibrating the position, capturing the readings, switching feed forward, parts counter, generation of events.

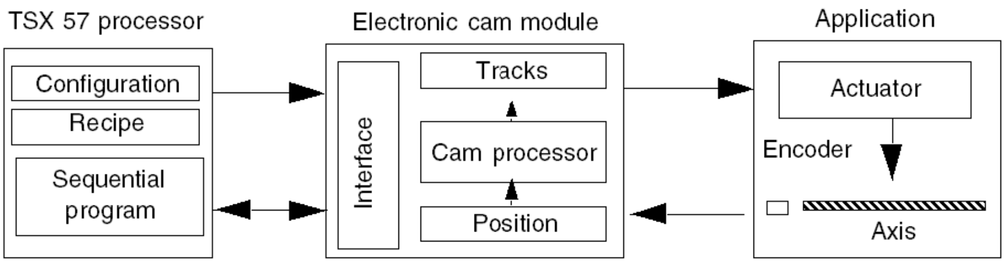
Operation of the electronic Cam Module

Introduction

The module calculates the position's reading from an absolute or incremental encoder connected to its inputs. The module controls its outputs depending on this position and on the transferred cam program (acceptance).

Illustration

The following diagram illustrates the operation of the cam module.



Description

Position: the module calculates the position's reading (angular and number of turns) according to the parameters provided by the position encoder.

Cam processor: defines when the cams switch from 1 to 0 according to the configuration and acceptance parameters transmitted by the PLC processor.

Tracks: control the outputs of the module according to the status of the cams associated with them.

PLC interface: this enables:

- the transfer of the configuration and acceptance parameters to the module,
- hardware faults to be recognized
- in the sequential program: the management of the machine operating modes and the extension of the module functions by acting directly on the outputs.

Position reading

Calculation of the Position Value

Using an incremental or absolute position encoder, the module calculates:

- the angular position value of the machine,
- the turn number value (for multi-turn processes).

All the operations performed in the module are based on the angular value. The number of turns may be taken into account in the sequential program.

Position Encoder

The module accepts 2 encoder types

Encoder type	Characteristics
Incremental	• 500 kHz bandwidth • multiplication by 4 • line check
Absolute	• SSi frame, any type of 8..25 bit format • transmission frequency determined automatically • reduction of resolution by 2, 4, 8, 16 and 32

NOTE: The cam module also takes parallel output absolute encoders (via Telefast ABE 7CPA11).

Associated Functions

In addition to the basic functions, the electronic cam module provides the following functions:

Function	Role
Readjustment *(see page 170)*	initializes the position reading value
Captures *(see page 171)*	take various types of reading, such as: length of parts, number of pulses per turn, incoming angle of parts, slip
Compensation for wear *(see page 221)*	compensates for play when there is a change of direction of movement

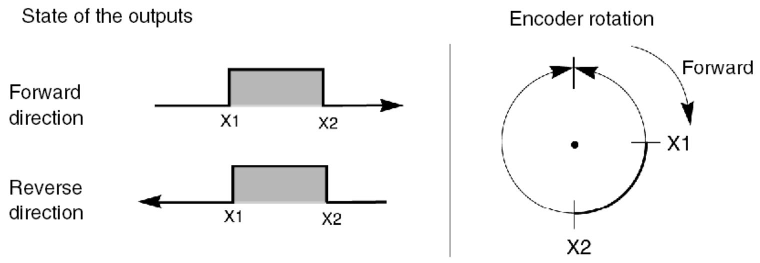
Cam processing operation

Role of the Tracks and the Cams

A track is made up of one or more cams. It controls a physical output of the module.

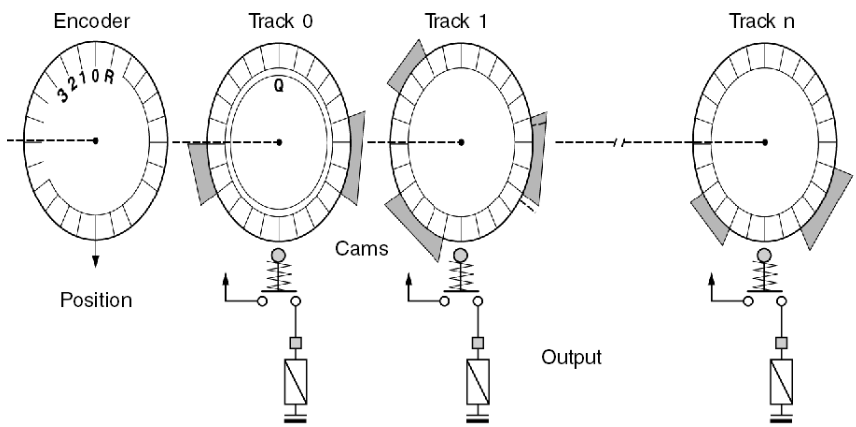
A cam defines an action on the output in a position reading interval.

Example of a cam in active position in both forward and reverse directions: the output associated to the track is activated when the angular reading is between 2 position values, X1 and X2 (when moving in either direction.).



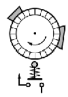
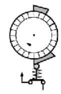
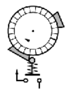
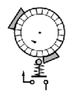
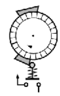
Analogy with Mechanical Cams

The following illustration shows a mechanical equivalent of the electronic cam.



Operation

The following table describes the operation of a track to which 2 cams are associated:

Phase	Description	Phase
1	The axis turns forward and drives the position encoder. The module calculates the position reading by evaluating the encoder increments. Track i is not active, output Qxy.i is at 0.	
2	When threshold X1 on cam 0 is reached: ● track i becomes active, ● output Qxy.i switches to 1.	
3	When threshold X2 on cam 0 is reached: ● track i becomes inactive, ● output Qxy.i switches to 0.	
4	When threshold X1 on cam 1 is reached, ● track i becomes active, ● output Qxy.i switches to 1.	
5	When threshold X2 on cam 1 is reached, ● track i becomes inactive, ● output Qxy.i switches to 0.	
6	The axis continues to rotate and the process is repeated (returns to phase no. 1).	

3 types of cam

The following table describes the 3 types of cam available. These cams can be active in forward or reverse direction, or forward and reverse simultaneously.

Type of cam	Role
Position *(see page 231)*	A position cam is a cam whose logic state depends on the position of the axis in relation to 2 thresholds.
Monostable *(see page 234)*	A monostable cam is one which switches to 1 when a threshold is crossed and returns to 0 after a time delay. This function is suitable for slow shaft detection.
Braking *(see page 235)*	A brake cam is one which switches to 1 when a threshold is crossed and returns to 0 when the same threshold is crossed in the opposite direction. This function is suitable for controlling braking at the top dead center of the machine.

Associated Functions

In addition to the basic functions, the electronic cam module provides the following functions:

Function	Role
Feedforward *(see page 227)*	Compensates for the delay caused by the machine's actuators.
Parallel track *(see page 227)*	Makes 2 tracks in one group parallel.
Event *(see page 227)*	Triggers an event each time the track is switched.
Parts counter *(see page 225)*	Manages the number of parts processed or cycles performed and reacts on the process when the counter has reached the limit value.

Structuring the cam processes

General

The cam process is done by a maximum of 128 cams divided between 32 tracks. The tracks are associated with the outputs of the module.

Processing is organized into 4 groups of 8 tracks, with groups 0 and 1 being associated with connector 0 on the module and groups 2 and 3 associated with connector 1.

Distributing the Tracks and the Cams

The following table describes the complete processing structure and the relationships with the module outputs.

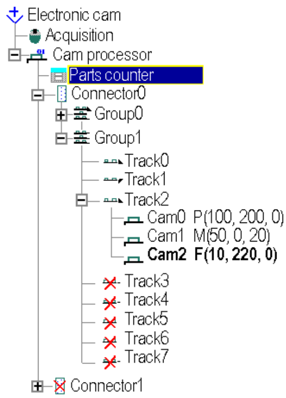
Connector	**0**		**1**	
Group	0	1	2	3
Maximum Number of Cams	32	32	32	32
Tracks	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Outputs	Q0. 0 1 2 3 4 5 6 7	Q1. 0 1 2 3	Q2. 0 1 2 3 4 5 6 7	Q3. 0 1 2 3
Connector	**0**		**1**	

Up to 32 cams can be assigned to one track, but in this case it is no longer possible to assign cams to the other tracks in the same group.

Tracks 4, 5, 6 and 7 of groups 1 and 3 are not linked to any physical output, and can be paralleled with tracks 0, 1, 2 and 3 of the same groups, or they can be used to control an output of a discrete module via the sequential program.

Example

In the example below, 3 cams have been assigned to track 2 of group 1, which is associated with connector 0. This track controls output Q1.2



Interface with the PLC program

General

Although it is independent, the processing of the electronic cam module is controlled by the sequential program of the PLC processor.

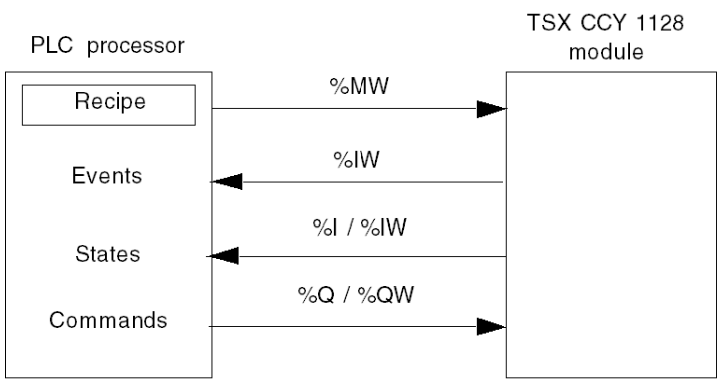
The electronic cam module benefits from:

- performances due to the independent module process (independence in relation to the cycles of PLC tasks)
- automatic cyclic exchanges with the main program in order to communicate with the other parts of the application.

States are read and commands written via the PLC sequential program using language objects associated with the module: %I, %Q, %M, etc accessible via mnemonics.

Illustration of exchanges

The following diagram illustrates the various exchanges between the electronic cam module and the PLC processor.



Description of the data exchanged

The following table describes the main data that is exchanged.

Type of data	Description
Recipe	The recipe groups together the data required for the module to control the machine for a series of parts. The recipe can be modified or completely changed by the PLC application. All this information is contained in the %MW PLC words in the memory zone reserved for the module. It contains the: • data for acquisition: the value of recalibration, play, parameters of the encoder (offset, reduction factor) etc. • descriptors of the tracks: parts counter,anticipation… • cam descriptors: cam type, threshold values…
Statuses	This data is used to control the electronic cam application in order to supervise, diagnose, or perform operations on the other elements of the application. It contains the values: • of the states of the physical inputs • of the position reading (angle and number of turns) • of the parts counter • of the registers of capture • of the states of the tracks and the outputs • of any faults
Commands	This data is used to control the electronic cam application from the PLC processor program to perform operations on the operating modes, enable the functions, force the outputs, etc. It contains the commands: • for starting or stopping the cam program • enabling the functions: recalibration, capture, position counting • direct commands on the following functions: recalibration, capture, parts counting • enabling cams and tracks • forcing outputs • masking events
Event	Activates the PLC processor event task. The module transmits data to the PLC processor regarding: • the source of the event • the angle values and number of turns which have been captured

Software installation functions for the module

General

Unity Pro software is used for the software installation of the TSX CCY 1128 module.

Description of the functions

The following table describes the functions specific to the electronic camapplication provided by Unity Pro for installing a TSX CCY 1128 module.

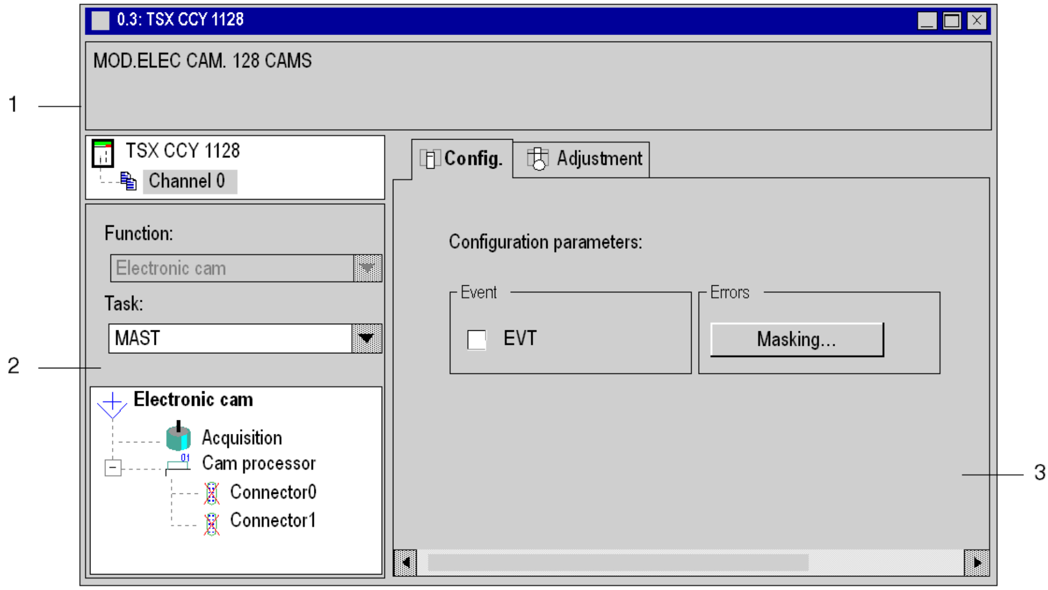
Functions	Description	Operating mode of the terminal	Operating mode of the cam processor
Configuration	Used for entering the configuration parameters of the module: • acquisition: type of encoder, reading format, type of recalibration, type of capture... • cam processor: resetting the tracks, behavior with faults... • connector: inversion of the tracks.	Offline or online	Stop
Adjusting the recipe	Used for entering the adjustment parameters of the recipe: • acquisition: number of points per cycle, backlash, recalibration value.. • cam processor: association of the cams with the tracks, track parameters, cam parameters, etc • parts counter: threshold	Offline or online	Stop
Adjustment	Used to modify certain adjustment parameters of the module recipe without stopping the cam processor: • acquisition: backlash, recalibration value, • cam processor: freeforwardfactor, thresholds and values of cam time delays.	Connected	Stop or Run

Functions	Description	Operating mode of the terminal	Operating mode of the cam processor
Debugging	Used to execute the following commands: • validation of cams, • enabling, forcing outputs • enabling, activating recalibration, capturing. It can also be used to display the following states: position reading, inputs, etc, and to perform module and application diagnostics.	Connected	Stop or Run

General ergonomics for the module implementation functions

Screen

The functions for implementing the module are accessed from the TSX CCY 1128 module’s hardware configuration screen.



This table defines the different addresses:

Address	Role
1	Module description zone.
2	This function zone offers access to the channel parameters.
3	Entry zone for entering parameters or commands (debug screen). This zone depends on the tab selected.

Browser

The browser for the electronic cam function presents the contents of an electronic cam application in a tree structure.

It can be used to navigate within the application, providing direct access to the parameter entry or debug screens associated with the following functions:

- Acquisition
- Cam processor
- Parts counter
- Connectors
- Groups
- Tracks
- Cams

Parameter entry

The colors of the parameters in the entry fields have the following meanings:

- Black: parameters which can be modified
- Grayed out: parameters which cannot be modified
- Blue: parameters which can be modified in adjust mode
- Red : incorrect parameter value entered

Installation Phase Overview

Introduction

The software installation of the application-specific modules is carried out from the various Unity Pro editors:
- in offline mode
- in online mode

If you do not have a processor to connect to, Unity Pro allows you to carry out an initial test using the simulator. In this case the installation *(see page 31)* is different.

The following order of installation phases is recommended but it is possible to change the order of certain phases (for example, starting with the configuration phase).

Installation Phases with Processor

The following table shows the various phases of installation with the processor:

Phase	Description	Mode
Declaration of variables	Declaration of IODDT-type variables for the application-specific modules and variables of the project.	Offline (1)
Programming	Project programming.	Offline (1)
Configuration	Declaration of modules.	Offline
	Module channel configuration.	
	Entry of configuration parameters.	
Association	Association of IODDTs with the channels configured (variable editor).	Offline (1)
Generation	Project generation (analysis and editing of links).	Offline
Transfer	Transfer project to PLC.	Online
Adjustment/Debugging	Project debugging from debug screens, animation tables.	Online
	Modifying the program and adjustment parameters.	
Documentation	Building documentation file and printing miscellaneous information relating to the project.	Online (1)
Operation/Diagnostic	Displaying miscellaneous information necessary for supervisory control of the project.	Online
	Diagnostic of project and modules.	
Key:		
(1)	These various phases can also be performed in the other mode.	

Implementation Phases with Simulator

The following table shows the various phases of installation with the simulator.

Phase	Description	Mode
Declaration of variables	Declaration of IODDT-type variables for the application-specific modules and variables of the project.	Offline (1)
Programming	Project programming.	Offline (1)
Configuration	Declaration of modules.	Offline
	Module channel configuration.	
	Entry of configuration parameters.	
Association	Association of IODDTs with the modules configured (variable editor).	Offline (1)
Generation	Project generation (analysis and editing of links).	Offline
Transfer	Transfer project to simulator.	Online
Simulation	Program simulation without inputs/outputs.	Online
Adjustment/Debugging	Project debugging from debug screens, animation tables.	Online
	Modifying the program and adjustment parameters.	
Key:		
(1)	These various phases can also be performed in the other mode.	

NOTE: The simulator is only used for the discrete or analog modules.

Introduction to the TSX CCY 1128 electronic cam module

2

Subject of this chapter

This Chapter introduces the TSX CCY 1128 electronic cam module

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Description of the TSX CCY 1128 in its environment	34
Physical description of the TSX CCY 1128	35
Electronic cam function of the TSX CCY 1128	37

Description of the TSX CCY 1128 in its environment

Introduction

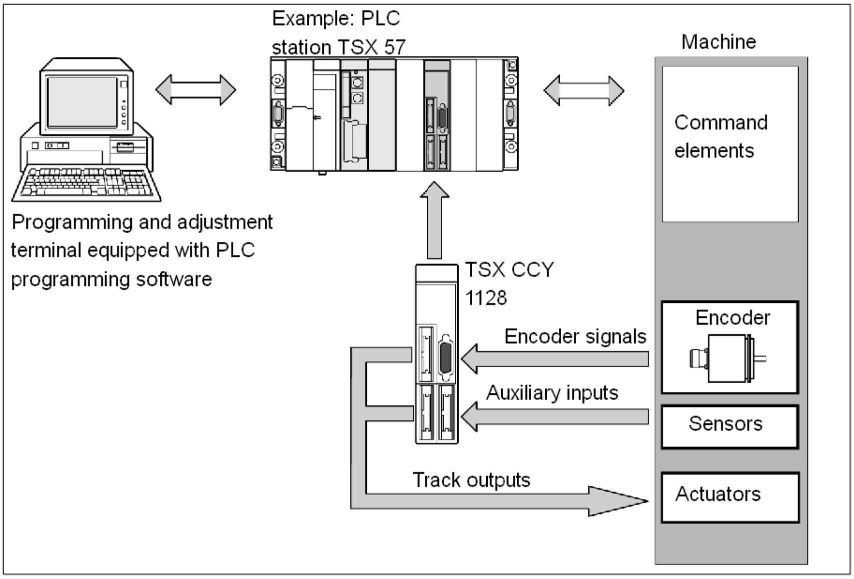
The TSX CCY 1128 module is a standard format application-specific module of the Premium range, which is built onto a TSX RKY rack •• of a PLC station TSX/PMX/PCX/PCIX 57. It performs the "electronic cam" function for a rotary, alternating, cyclic or endless axis, managed by an incremental or absolute encoder.

Operating Principle

The module independently manages up to 128 cams, which can be distributed over a maximum of 32 tracks to which up to 24 physical outputs and 8 logical outputs can be allocated. After configuration and adjustment information has been sent by the PLC's processor, the module processes the cam program and controls the track outputs independently from the cycle of the PLC.

Installation diagram

The figure below represents a basic configuration of an installation



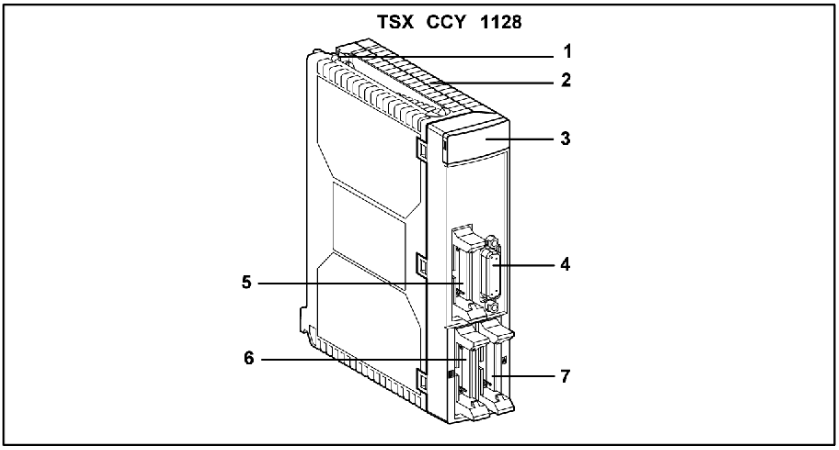
Physical description of the TSX CCY 1128

Introduction

The interface connectors of the inputs and outputs towards the machine are distributed on the front face of the module. The connector for connecting to the X bus is found to the back. All the command signals and data of the processor pass through this bus.

View of the module

The diagram below shows the TSX CCY 1128 module and its various elements.



Elements and their functions

Addresses	Elements	Functions
1	Screw	Attaches the module to the TSX RKY •• rack
2	Module casing	Performs the following functions: • support and protection of the electronic boards, • locking of the module into its slot, • Supports connectors.

Addresses	Elements	Functions
3	Display block comprising 4 LEDs: • green RUN LED • red ERR LED • red I/O LED • green CH0 LED	Display the statuses and faults of the module as well as channel diagnostics. Indicates the operating mode of the module, Indicates an internal module error, Indicates an external module error or an application fault, Provides channel diagnostics. The various LED-status displays and their meanings *(see page 158)*
4	SUB-D 15 points connector	Enables connection to the encoder.
5	HE 10 connector	Enables connection of the track outputs of groups 0 and 1
6	HE 10 connector	Enables connection of the track outputs of groups 2 and 3.
7	HE 10 connector	Enables connection of the auxiliary inputs and encoder power supply.

Electronic cam function of the TSX CCY 1128

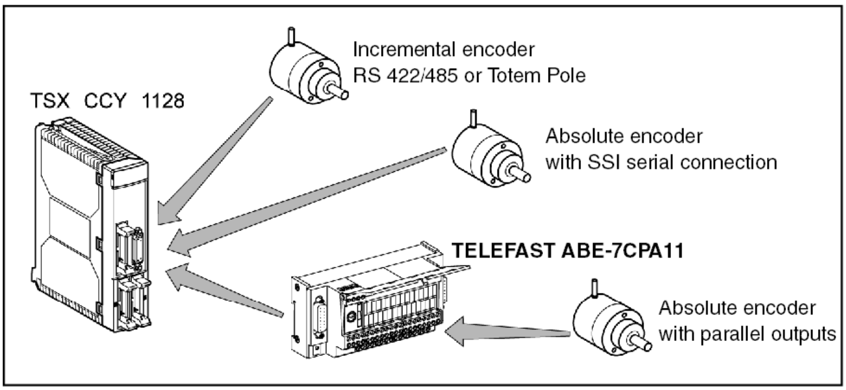
Introduction

The electronic cam function controls the module's outputs according to the position of the encoder installed on the machine. Several encoder types can be used:

- incremental encoder with RS 422/ RS 485 line transmitter outputs,
- incremental encoder with Totem Pole outputs,
- absolute SSI encoder with series outputs,
- absolute encoder with parallel outputs. The use of this encoder requires a TELEFAST ABE-7CPA11 interface, which converts parallel output signals into series output signals.

Illustration

The diagram below represents the various types of encoder, which can be connected to the module.



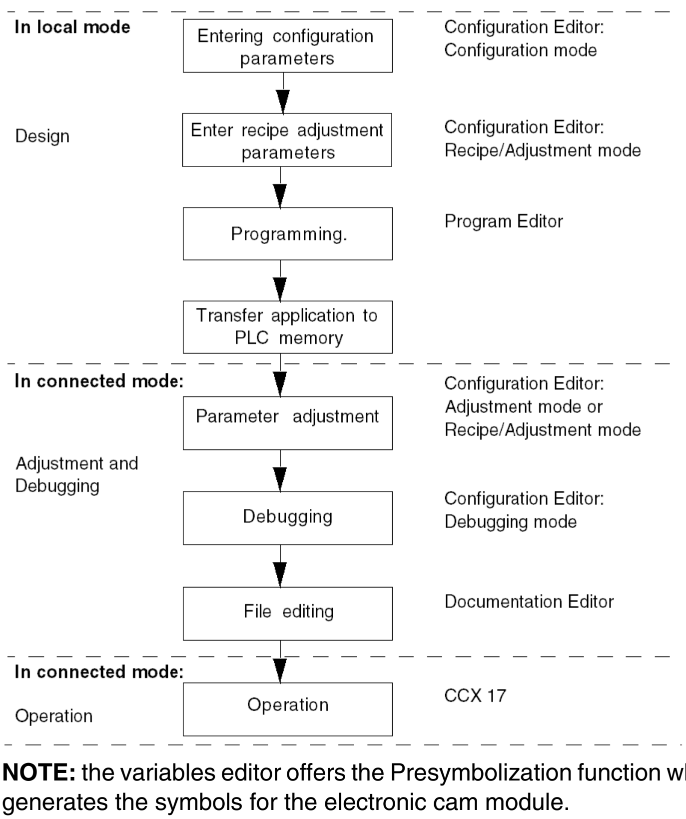
Installation methodology

3

Methodology for setting up module software

Illustration

The following flowchart summarizes the various implementation phases for an electronic cam application.



NOTE: the variables editor offers the Presymbolization function which automatically generates the symbols for the electronic cam module.

Types of applications

4

Subject of this chapter

This chapter describes the various fields and types of application an electronic cam can process.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Areas of application	42
Alternating movement	43
Use in alternating motion	45
Rotary movements	46
Use for Rotary Motion	47
Cyclic movements	48
Use for cyclic motion	50
Continuous movement	52
Use for continuous motion	53

Areas of application

Areas of Application

The module manages high-speed machines autonomously.

The areas of application are:

- Mechanical or hydraulic presses,
- Revolving transfer machines,
- Transfer machines
- Packaging machines,
- Woodworking machines

The machines in these various areas are characterized according to the type of **movement** performed by the machine axis.

Alternating movement

Area

This is for revolving transfer machines and hydraulic presses.

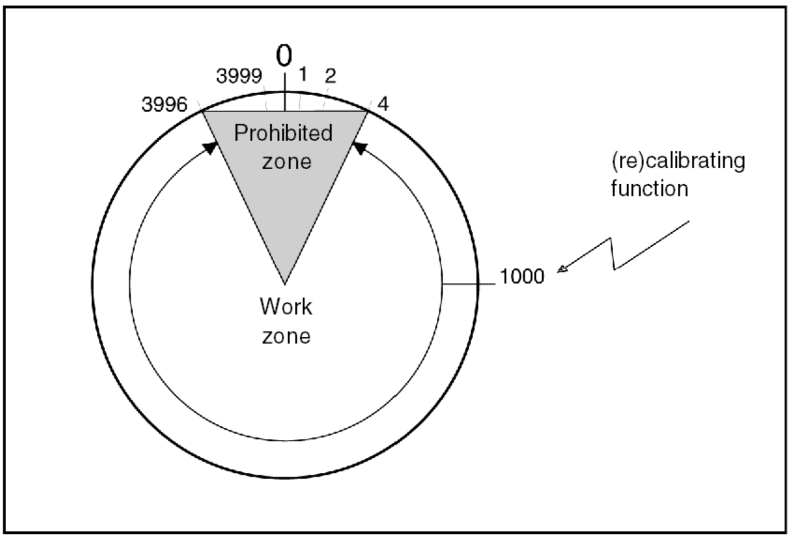
The alternating movement of the axis is "to-and-fro".

The rotation speed can vary greatly during the machine cycle.

Outputs are positioned in both directions. The forward program can differ from the reverse program. There may be a prohibited zone defined as a protective zone (brake control).

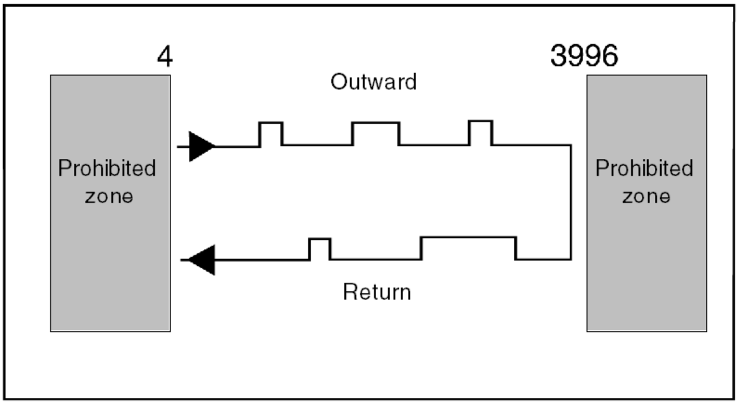
Example of an alternating axis

In this application, the encoder resolution is 4000 pulses per cycle. The permitted movement zone is 3992 pulses. Recalibration is performed at 90° of the mechanical origin (0) which is not accessible.



Timing diagram of an output

Timing diagram



Use in alternating motion

Using the module

In this type of application, the user indicates the angular resolution. The number of counter pulses in 360° (a cycle), are indicated. The angle measurement will be done in the "0" to "resolution -1" interval with:

- An incremental encoder, recalibrating the reading at the machine origin (cams reference) can be done with a value other than "0". Each time the machine is started up with an incremental encoder. At each machine cycle if the axis drive slips.
- With an absolute encoder, the module can correct the offset of the encoder on the machine 0. This type of encoder is suitable for this type of application if the drive does not slip.

The module can correct the error due to the axis play when the direction of movement is reversed.

Brake control can be programmed on an output to generate the protective function of the "prohibited zone".

The parts counter function can be programmed to indicate the number of cycles being run.

Anticipation

For high-speed machines, it is possible to correct the following very precisely: the actuator ascent time and the internal module processing time, if required.

A fine resolution and a large number of pulses per cycle, ensure precise anticipation.

For example: if the user wishes to anticipate the command of an actuator by 1ms, the encoder must be able to provide more than 5 pulses in that time. The speed must be stabilized before switching to the cam. This is particularly important after a reversal of the direction of rotation.

Rotary movements

Area

This is for mechanical presses and packaging machines.

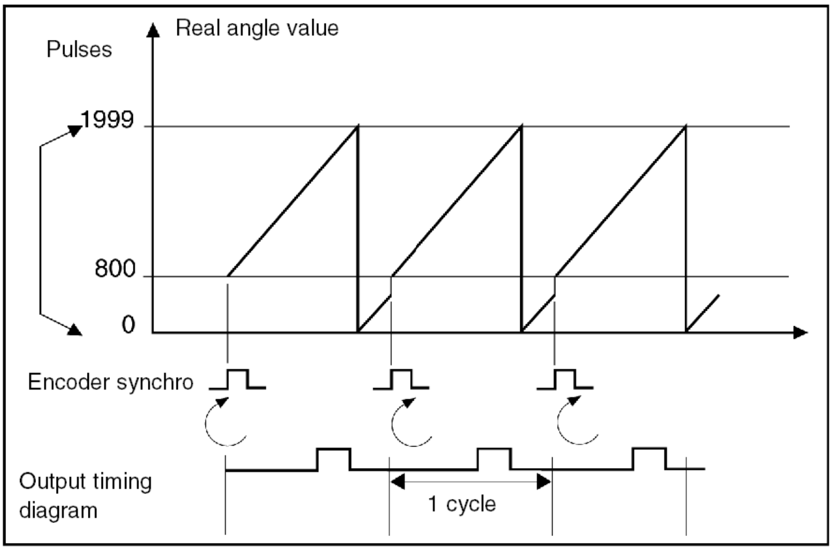
The main movement of the axis is in a single direction of rotation. The positioning of the outputs is the same for each cycle. Outputs can be activated between two cycles.

The linear speed is not necessarily constant during movement. The machine must stop within a precise zone.

Two types of encoders can be used, absolute or incremental. For drives with slip, it is preferable to use an incremental encoder.

Example of a rotating axis

A press cycle represents 2000 encoder pulses. Synchronization of the axis (correction of slip) will start at 800 pulses in relation to the physical origin of the tools. Recalibration is carried out systematically at each cycle.



Use for Rotary Motion

Using the module

For rotary movement, the user will indicate the number of pulses in a machine cycle.

- The cams can be declared for all the values of a cycle.
- The activity zone of an output can be programmed between two cycles.

Recalibrating the position reading on the machine origin (cam reference) can be done with a value other than "0":

- Each time the machine is started up,
- On each machine cycle for drives with slip,
- With an absolute encoder, the module can correct the offset on the machine 0. It can also apply an angle of deviation in relation to this same machine 0.

Module capture functions allow the following to be read:

- The number of counter pulses in a cycle,
- The axis slip (value of the angle before recalibration),
- The angle that parts arrive at, or their dimensions.

The parts counter function can be programmed to indicate the number of presses performed.

Anticipation

For high-speed machines, it is possible to correct the following very precisely: the actuator ascent time and the internal module processing time, if required.

- A fine resolution and a large number of pulses per cycle, ensure precise anticipation. For example: if the user wishes to anticipate the command of an actuator by 1ms, the encoder must be able to provide more than 5 pulses in that time.

NOTE: speed variation must be negligible in a period corresponding to the largest anticipation value requested.

Cyclic movements

Area

This is for packaging and bottling machines.

The main movement of the axis is in a single direction of rotation.

The linear speed is constant during movement, or varies slowly in relation to the machine rate. The positioning of the outputs is the same for each cycle.

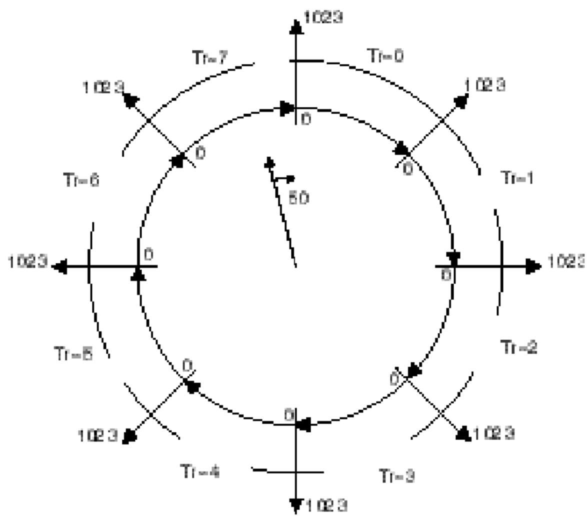
Parts arrive periodically:
- One part for each cycle. All the parts should enter systematically for the same position angle value in each cycle (mechanical problem).
- There can be active cams on several cycles.
- Certain grouping operations, for example, may be performed in a single cycle (one cycle in 4, for example).

In this type of application, recipe means a usage value. The recipe groups together the data required for the module to control the machine for a series of parts. The recipe can be modified or completely changed by the PLC application.

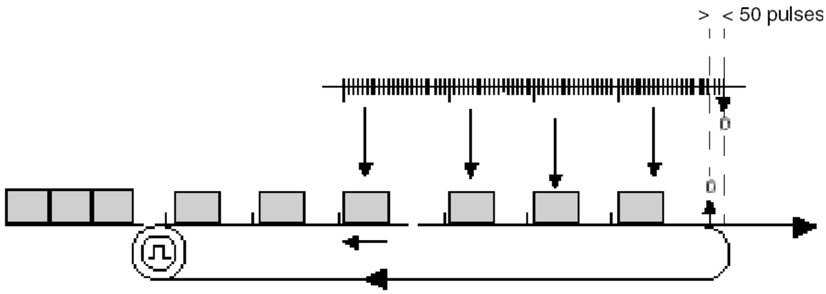
Two types of absolute encoder can be used, single-turn or multi-turn. The number of turns is not necessarily a power of 2.

Example of a cyclical axis

A cycle is made up of 1024 pulses and corresponds to the distance between two parts. The complete machine cycle is performed over 8 cycles. The start of the cycle (arrival of the part) is offset by 50 pulses in relation to the machine 0.



Reflex operations on parts:



Point 0 of the machine is defined when the tool box and the conveyor are synchronized (indexes facing).

Use for cyclic motion

Using the Module

For cyclic movement, the user indicates the resolution in a basic cycle and the number of turns per machine cycle.

The number of encoder pulses for a cycle (up to 32767) will be indicated; the position reading will be taken in the range ["0" to "resolution -1"].

The number of cycles per machine cycle (up to 32767) will also be indicated.

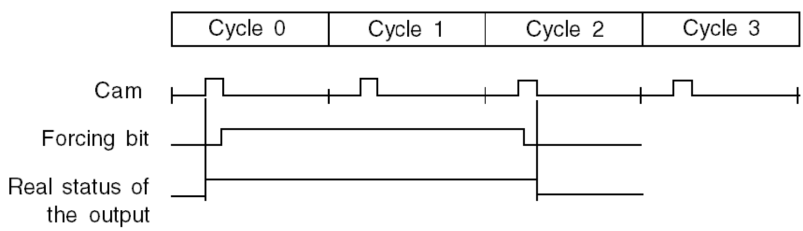
The cycle number value will be set in the range ["0" to "No. of cycles -1"].

The angle reading and the reading of the number of cycles can be (re)calibrated, independently or simultaneously, on the machine 0 position:
- When a machine is debugged (with an absolute encoder).
- Each time the machine is started up (with an incremental encoder).
- The cams used can be declared for all the values of a cycle.
- The activity zone of an output can be programmed between two cycles.
- From one series to the next, the parts may arrive at different angles.

It is possible to adjust the angle at which the parts arrive in the periodical process without modifying either (re)calibration on position 0 of the machine or the whole cam program in the absolute encoder.

It is possible to construct a cam profile that reacts with good precision over several cycles.

Option to force the cam output using the application program.



The use of an event cam is recommended in this case.

The "parts counter" function can be programmed to indicate the number of parts completed or to enable the action of a cam once a certain number of parts is reached.

The module can indicate the linear speed of the axis.

It can also provide the number of counter pulses in a cycle (only when using an incremental encoder).

The "slow shaft detection" function can be obtained on a track associated to a time delay cam.

Anticipation

For high-speed machines, it is possible to correct the following very precisely: the actuator ascent time and the internal module processing time, if required.

A fine resolution and a large number of pulses per cycle ensure precise anticipation. For example: if the use wishes to anticipate the command of an actuator by 1ms, the encoder must be able to provide more than 5 pulses in that time.

Continuous movement

Area

Continuous movement is for packaging machines, conveyors, tool boxes (woodworking machines, for example).

The main movement of the axis is in a single direction of rotation.

The machine operates continuously and drives an incremental encoder.

The linear speed is constant during operation or varies slowly in relation to the machine rate.

Parts arrive on the machine randomly, but they are all processed in the same way.

There must be a sensor that sets a reference point for each part as it arrives.

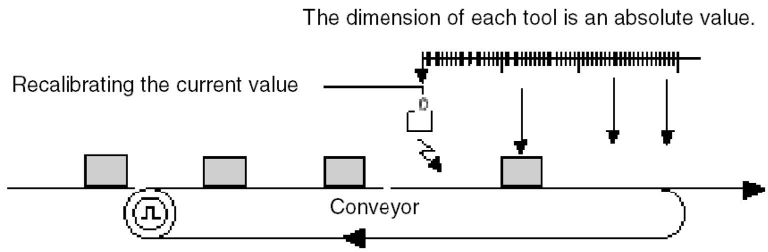
Machines that set a reference point on 1 part

The process operates on one part at a time (one station).

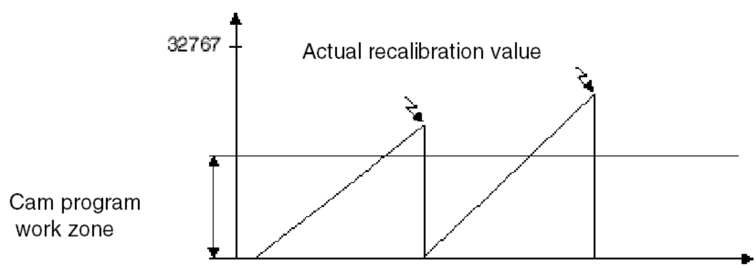
The reflex functions are finished before the next part arrives. The reference point for each part is set by recalibrating the current value.

Example

The position value is recalibrated to 0.



The work area is less than 32767 pulses. The position value remains outside the work zone until adjustment.



Use for continuous motion

Use

The position reading has a fixed dynamic of 32768. Programming the cams is done in this group of pulses.

Example: a 16 m belt with an encoder resolution of 0.5 mm.

Counter recalibration must be configured on the rising edge on Irec. The recalibration value can be adjusted with a value other than 0.

The length of the part can be measured using the capture functions.

The application of events can be used, for example, to:
- set the processor cam to Run after recalibration (movement of parts).
- set the cam processor to Stop on recalibration if processing of a part has not finished.

Anticipation

For high-speed machines, it is possible to correct the following very precisely: the actuator ascent time and the internal module processing time, if required.

A fine resolution and a large number of pulses per cycle, ensure precise anticipation. For example: if the user wishes to anticipate the command of an actuator by 1ms, the encoder must be able to provide more than 5 pulses in that time.

Examples

5

Subject of this chapter

This chapter presents two examples of using the TSX CCY 1128 electronic cam module. The simple example shows the elementary programming required to actuate an output on detection of a cam. The advanced example implements several cams with generation of events when switching modulo and full counters with production recipe management. They will enable you to implement a cam module quickly.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
5.1	Simple example	56
5.2	Advanced example	64

5.1 Simple example

Subject of this section

This section deals with the following subjects:

What Is in This Section?

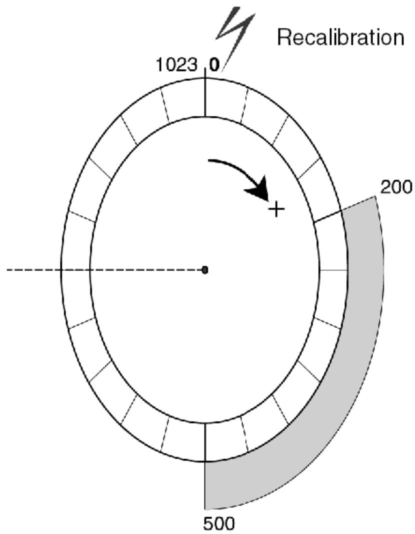
This section contains the following topics:

Topic	Page
Introduction to a simple example	57
Entering configuration parameters	58
Programming	61

Introduction to a simple example

Example

This is the implementation of an output on detection of a cam. The axis used is a type 1 rotating axis.



The electronic cam is set between 200 and 500. It activates output 0 on track 0.

Entering configuration parameters

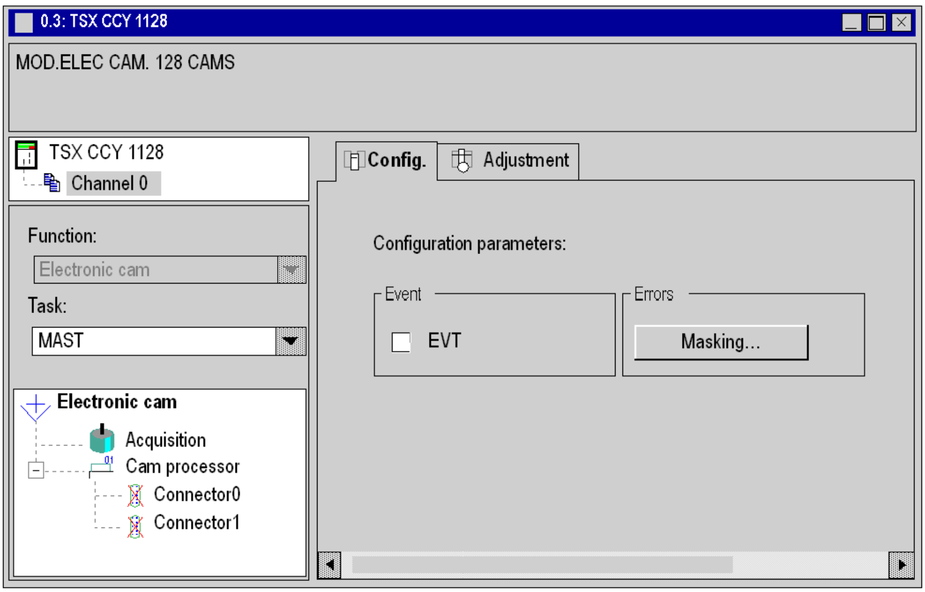
Hardware configuration

The hardware configuration is as follows:

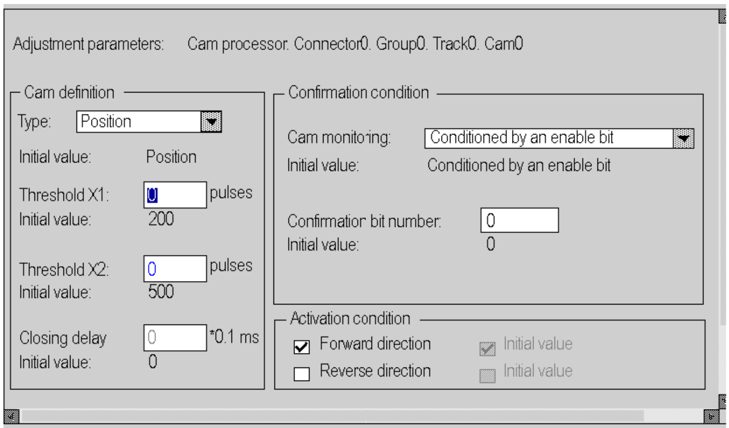
- TSX PSY 5500 double format power supply,
- TSX P57 204 double format processor,
- TSX CCY 1128 electronic cam module (in slot 3, rack 0),

Procedure

The stages are as follows:

Step	Action
1	• Select the TSX CCY 1128 module at its position n° 3 in rack n° 0. • Double-click on the module. Result: the module hardware configuration screen appears: 
2	The **Configuration** tab is selected. • Double-click with the left button on Acquisition in the browser
3	Select • For the Input interface: Incremental encoder. • For the Measurement Format: Type 1. • For the position Recalibration on Irec: Short cam • Confirm with the toolbar button.

Step	Action
4	• Click on the Adjust tab. • Double-click with the left button on Acquisition in the Browser. • Confirm with the toolbar button.
5	• Configure 1024 for No. of pulses/cycle.
6	• Click on the Configuration tab. • Confirm. • Click with the right button on Connector 0. • Unlock.
7	• Click on the Adjust tab. • Confirm with yes. • Double-click with the left button on Connector 0. • Double-click with the left button on Group 0. • Click with the right button on track 0. • Activate. • Click with the right button on track 0. • Create Cam ... • Select 0. • Confirm with OK.

Step	Action
8	● Double-click with the left button on cam 0. Result: the parameter adjustment screen appears. 
9	Configure ● for ThresholdX1:200, and ● for ThresholdX2: 500 Select ● for Cam monitoring (in the Enable conditions): Conditioned by an enable bit ● for the Activation Condition: Forward Enable the configuration entered with the toolbar button.

Programming

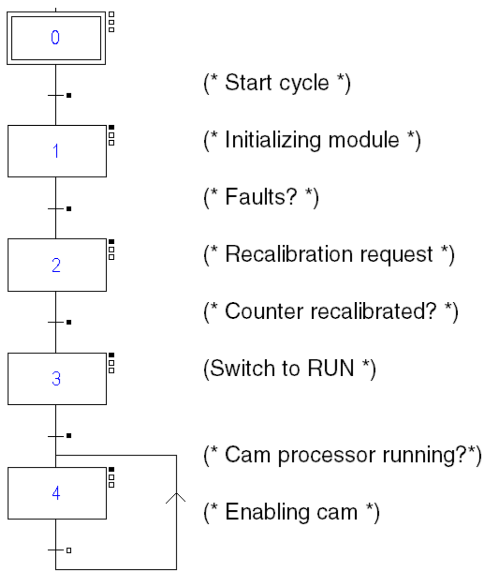
Introduction

The module calculates the position reading from an incremental encoderconnected to its inputs. Depending on this position and the cam programtransferred which describes the recipe, the module controls its outputs.

This program contains the minimum number of steps required to ensure that the cam function starts up correctly. It is sufficient here for each step to set the commands required to activate the output to 1. For this example, we have used an IODDT called `Ex_cam` of type `T_CCY_GROUP0`.

SFC

This illustration describes an example of SFC.



Step 0 -> 1

(* start cycle? *)

%M0

Action on activation step 1

(* Initializing enable and assignment parameters *)

(* assigning track 0 to output 0 *)

Ex_cam.Group0_And_Bits := 16#0001;

(* Enabling group 0 outputs *)

SET (Ex_cam.outs_Enable);

(* Reset faults *)

SET (Ex_cam.ack_flt);

Step 1 -> 2

(* default? *)

NOT Ex_cam.ch_error

Action on activation step 2

(* Enabling angle recalibration function *)

SET (Ex_cam.preset_ang_Enable);

Step 2 -> 3

(* Counter recalibrated? *)

Ex_cam.ang_ok

Action on activation step 3

(*setting the cam processor to run *)

SET (Ex_cam.pcam_start_stop);

Step 3 -> 4

(* cam processor in RUN? *)

Ex_cam.pcam_on

Action on activation step 4

(* enabling cam *)

SET (Ex_cam. ENAB_GROUP0_BITS.0);

Variables used

Address	Symbol	Comments
%M0	Dcy	Start cycle
%I0.3.0.Err	Ex_cam.ch_error	Channel error bit
%I0.3.0.0	Ex_cam.ang_ok	Valid angle reading
%I0.3.0.3	Ex_cam.pcam_on	Cam processor RUN/STOP
%Q0.3.0.0	Ex_cam.preset_ang_Enable	Enables (re)calibration function on angle value only
%Q0.3.0.5	Ex_cam.pcam_start_stop	Set: Cam processor start/ cam processor reset stop
%Q0.3.0.15	Ex_cam.ack_flt	Command to acknowledge faults present
%Q0.3.0.25	Ex_cam.outs_Enable	Global enabling of track outputs
%QW0.3.0.1	Ex_cam.Group0_And_Bits	8 bits for assigning tracks to group outputs
%QW0.3.0.0	Ex_cam.ENAB_GROUP0_BITS	Enabling cam

5.2 Advanced example

Subject of this section

This section deals with the following subjects:

What Is in This Section?

This section contains the following topics:

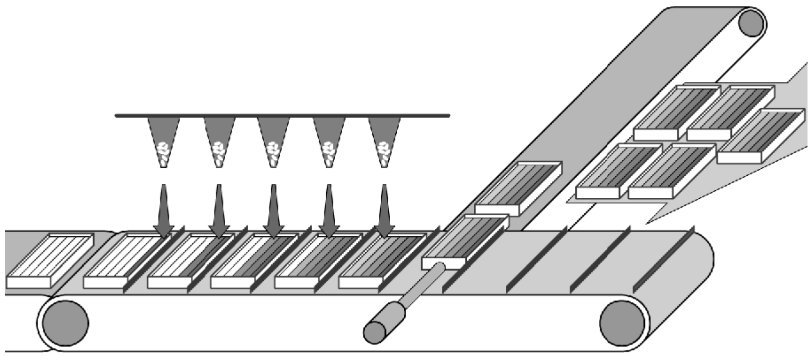
Topic	Page
Introduction to a detailed example	65
Operating mode	67
The Recipes	68
Entering configuration parameters	69
Programming	70
Special Features	76

Introduction to a detailed example

General

An automatic device controls the filling and storage of boxes of medicine. During filling, each box is wedged into a mechanical stop. Once they have been filled, the boxes are stored in lots of 5. Each lot is then sent to the storage area.

Illustration of automatic device



NOTE: This example is given for information only and is not an exact representation of an industrial application.

Description of the cycles

The machine is divided into 6 cycles of 1024 pulses:

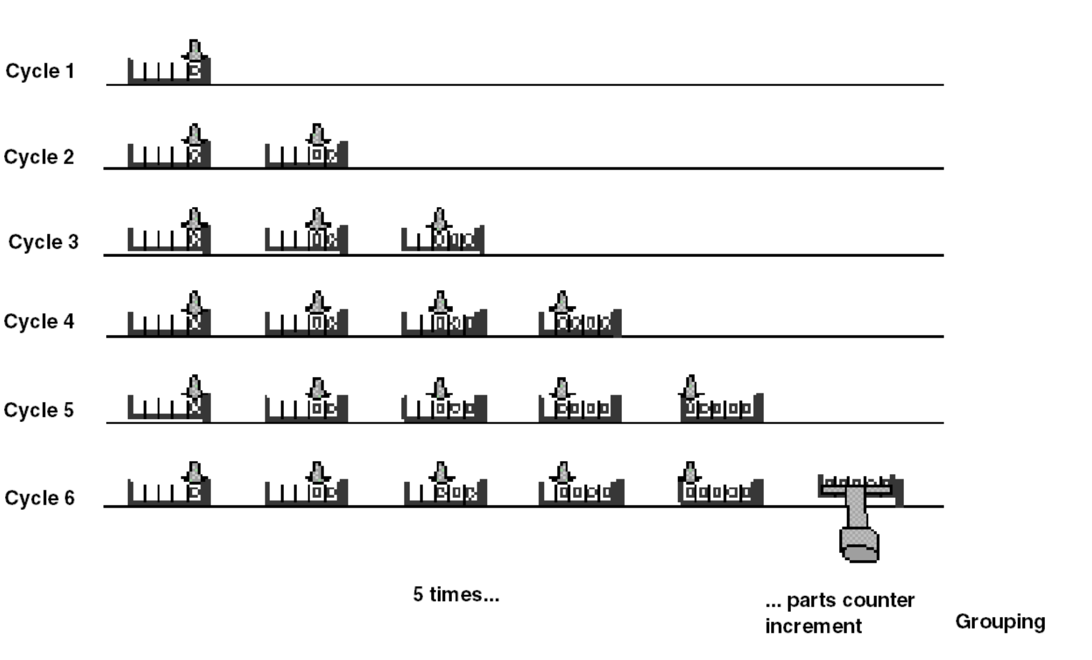
- 5 filling cycles consisting of depositing the medicine into the box,
- 1 storage and grouping cycle.

The parts counter counts the parts to be grouped for entry to the storage area. If the storage area is full, grouping is transferred.

The application program manages the transient machine loading and unloading phases. During loading, the stations are activated one at a time according to the cycle number.

When the machine is fully loaded, all stations work in parallel during each cycle.

	Filling the machine					Clearing	
	track 0	track 1	track 2	track 3	track 4	track 5	track 6
	station 1	station 2	station 3	station 4	station 5	station 6	station 7
	forward motion of the belt						



Stations 1 to 5 are controlled by the outputs of tracks 0 to 4 and only have one output per cycle (1 cam). Track 5 performs operations on the indexer for station 6, track 6 performs operations on the grouping actuator at station 7.

Operating mode

Start-up

- Recalibration is performed before every start-up.
- The machine is started gradually according to the arrival of the boxes.
- The parts counter is incremented for every full box. When this counter equals 5, the collected batch is pushed towards the packaging area.
- managing 2 types of stop:
  - a controlled stop to empty the string of boxes in progress.
  - an immediate stop.

The Recipes

At a glance

Two recipes are managed. These recipes are stored in the following %MW words:

Recipe 1	%MW100 to %MW816
Recipe 2	%MW850 to %MW1566

Parameters for recipe 1

This table describes the parameters for recipe 1:

Track	**Cam**	**Positions**		**Direction**	**Enabling**
Track 0	Cam 0	position X1=800	X2=820	Forward	enable by bit:0
Track 1	Cam 1	position X1=600	X2=620	Forward	enable by bit:1
Track 2	Cam 2	position X1=400	X2=420	Forward	enable by bit:2
Track 3	Cam 3	position X1=200	X2=220	Forward	enable by bit:3
Track 4	Cam 4	position X1=0	X2=20	Forward	enable by bit:4
Track 5	Cam 5	position X1=800	X2=900	Forward	enable by bit:5
Track 6	Cam 6	position X1=200	Tempo 100 ms	Forward	full counter
The parts counter is incremented by track 5. The parts counter is reset by track 6.					

Parameters for recipe 2

This table describes the parameters for recipe 2:

Track	**Cam**	**Positions**		**Direction**	**Enabling**
Track 0	Cam 0	position X1=800	X2=900	Forward	enable by bit:0
Track 1	Cam 1	position X1=600	X2=700	Forward	enable by bit:1
Track 2	Cam 2	position X1=400	X2=500	Forward	enable by bit:2
Track 3	Cam 3	position X1=200	X2=300	Forward	enable by bit:3
Track 4	Cam 4	position X1=0	X2=100	Forward	enable by bit:4
Track 5	Cam 5	position X1=800	X2=900	Forward	enable by bit:5
Track 6	Cam 6	position X1=200	Tempo 100 ms	Forward	full counter
The parts counter is incremented by track 5. The parts counter is reset by track 6.					

Entering configuration parameters

Hardware configuration

The hardware configuration is as follows:

- TSX PSY 5500 double format power supply,
- TSX P57 204 double format processor,
- TSX CCY 1128 electronic cam module (in slot 3, rack 0),

Procedure

The axis configuration and adjustment parameter entry is the same as for the simple example.

Programming

Introduction

The module calculates the position reading from an incremental encoderconnected to its inputs. Depending on this position and the cam programtransferred which describes the recipe, the module controls its outputs.

We will use the IODDT `Channel_0` of type `T_CCY_GROUP0` which we will associate with channel 0 of the TSX CCY module situated in slot 3 of the basic rack.

We use one section to carry out the preliminary processing and another to carry out the subsequent processing.

We will also need an event task.

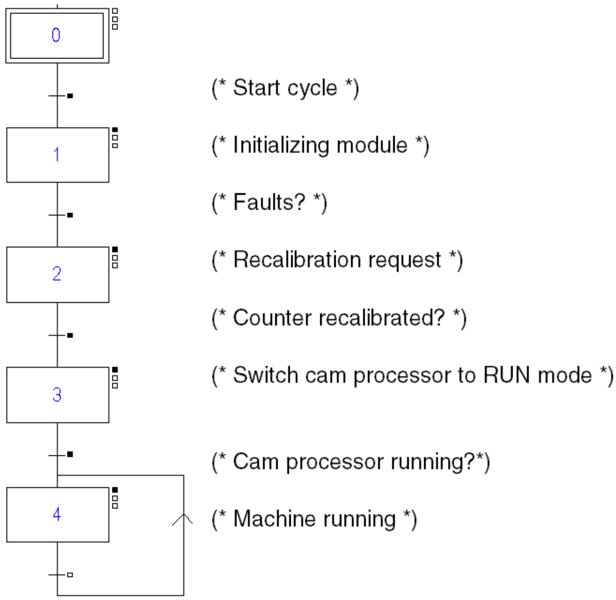
Preliminary section

```
(*SFC Chart initialization*)
IF NOT %M0 THEN SET %S21;
END_IF;
(* immediate cycle stop *)
IF %M3 THEN SET(%S21);RESET(Channel_0.Pcam_start_stop);
Channel_0.ENAB_GROUP0_BITS:=0;RESET(%M3);RESET(%M0);
END_IF;
(* Power outage management *)
IF %S1 THEN SET(%S21);RESET(Channel_0.Pcam_start_stop);
Channel_0.ENAB_GROUP0_BITS:=0;RESET(Channel_0.OUTS_ENABLE);
            END_IF;
(* Recipe management *)
(* saving recipe 1 in words %mw100 onwards *)
IF %M11 THEN TRF_RECIPE (Channel_0, 1,100);RESET(%M11)
        END_IF;
(* saving recipe 2 in words %MW850 onwards *)
IF %M12 THEN TRF_RECIPE (Channel_0,1,850);RESET(%M12);
        END_IF;
(*Restoring recipe 1 from the %MW100 word*)
IF %M13 THEN TRF_RECIPE
(Channel_0,0,100);RESET(%M13);RESET(Channel_0.Pcam_start_sto
p);
        END_IF;
```

```
(*Restoring recipe 2 from the %mw850 word*)
IF %M14 THEN TRF_RECIPE
(Channel_0,0,850);RESET(%M14);RESET(Channel_0.Pcam_start_sto
p);
        END_IF;
(* restoring the recipe switches the processor cam to STOP and
so the command must be reset to zero *)
```

SFC

This illustration describes the Chart of the example.



Step 0 -> 1

```
(* start cycle? *)
%M0
```

Action on activation step 1

```
(* Initializing enable and assignment parameters *)
(* assigning tracks to GRP outputs '7 tracks / 7 outputs' *)
Channel_0.Group0_and_bits:=16#007F;
(* Reset parts counter *)
SET(Channel_0.Came_pieces_reset);
(* Authorization to control group 0 outputs *)
SET(Channel_0.OUTS_ENABLE);
(* Event authorization when angle rolls over *)
SET(Channel_0.Evt_ang_enable);
(* Acknowledge faults during start-up phase *)
SET(Channel_0.Came_ack_flt);
```

Step 1 -> 2)

```
(* fault? *)
NOT Channel_0.Ch_error;
```

Action on activation step 2

```
(* Type 1 function therefore recalibrate angle *)
SET(Channel_0.Preset_ang_enable);
```

Action on deactivation step 2

```
(* Reset recalibration *)
RESET(Channel_0.Preset_ang_enable);
```

Step 2 -> 3

```
(* Counter recalibrated? *)
Channel_0.Ang_ok;
```

Action on activation step 3

```
(*setting the cam processor to run *)
SET(Channel_0.Pcam_start_stop);
```

Step 3 -> 4

```
(* cam processor in RUN? *)
Channel_0.Pcam_on;
```

Action on activation step 4

```
(* Enabling parts counter *)
RESET(Channel_0.Came_pieces_reset);
SET(Channel_0.Pieces_enable);
```

Subsequent section

```
(* Managing faults *)
IF Channel_0.Ch_error THEN SET(%S21); END_IF;
(* Fault acknowledgement *)
IF %M1     THEN SET(Channel_0.Came_ack_flt);RESET(%M1);
        ELSE RESET(Channel_0.Came_ack_flt);    END_IF;
(* Reading fault type *)
IF %M2 THEN READ_STS(Channel_0);RESET(%M2);    END_IF;
```

EVENT-EVT1

```
(* Managing cycle rollovers for starting *)
(* Detecting angle number rollovers *)
IF Channel_0.Evt_ang AND %M0 THEN INC(%MW0);END_IF;
(* Action on detection of cycle rollovers, validation of cams
*)
IF(%MW0=1)AND %M0 THEN
SET(Channel_0.ENAB_GROUP0_BITS.1);END_IF;
IF(%MW0=2)AND %M0 THEN
SET(Channel_0.ENAB_GROUP0_BITS.2);END_IF;
IF(%MW0=3)AND %M0 THEN
SET(Channel_0.ENAB_GROUP0_BITS.3);END_IF;
IF(%MW0=4)AND %M0 THEN
SET(Channel_0.ENAB_GROUP0_BITS.4);END_IF;
IF(%MW0=5)AND %M0 THEN
SET(Channel_0.ENAB_GROUP0_BITS.5);END_IF;
IF(%MW0>6)AND %M0 THEN %MW0:=6;END_IF;
(* Managing cycle rollovers for stopping *)
```

```
(* Detecting angle number rollovers *)
IF Channel_0.Evt_ang AND NOT %M0 THEN DEC(%MW0);END_IF;
(* Action on detection of cycle rollovers, disabling cams *)
IF(%MW0=5)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.0);END_IF;
IF(%MW0=4)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.1);END_IF;
IF(%MW0=3)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.2);END_IF;
IF(%MW0=2)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.3);END_IF;
IF(%MW0=1)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.4);END_IF;
IF(%MW0=0)AND NOT %M0 THEN
RESET(Channel_0.ENAB_GROUP0_BITS.5);END_IF;
IF(%MW0<1)AND NOT %M0 THEN %MW0:=0;END_IF;
(* Managing event stack overflow on cam card *)
(* Detecting overflow *)
IF Channel_0.Direction_evt THEN SET(%M15);END_IF;
```

Variables used

Table

Address	Symbol	Comments
%M0	Dcy	Start cycle
%M1	Ack_def	Acknowledging fault
%M2	Read_def	Reading fault
%M3	Stop_imm	Immediate stop of the machine
%M5	Counter_plein	
%M11	Save_recipe1	Saving Recipe 1
%M12	Save_recipe2	Saving Recipe 2
%M13	Restore_recipe1	Restoring Recipe 1
%M14	Restore_recipe2	Restoring Recipe 2
%M15		(* Detecting overflow *)
%MW0	Comp_modulo_ang	Evt. rollover counter
%MW2	Comp_comp_pieces	
%I0.3.0.Err	Channel_0.Ch_error	Channel error bit

Address	Symbol	Comments
%I0.3.0	Channel_0.Ang_ok	Valid angle reading
%I0.3.0.3	Channel_0.Pcam_on	Cam processor RUN/STOP
%IW0.3.0	Channel_0.Group0_strack	State of tracks i.0 to i.7
%IW0.3.0.12.0	Channel_0.Evt_ang	Angle value rollover event
%IW0.3.0.12.6	Channel_0.Evt_pieces_full	Reaching limit value for the parts counter event
%IW0.3.0.12.15	Channel_0.Direction_evt	Direction of movement on ANG_EVT or TURN_EVT
%Q0.3.00.0	Channel_0.Preset_ang_enable	Enables (re)calibration function on angle value only
%Q0.3.00.5	Channel_0.Pcam_start_stop	Set: Cam processor start/ cam processor reset stop
%Q0.3.00.7	Channel_0.Pieces_enable	Enables the parts counter function
%Q0.3.00.8	Channel_0.Evt_ang_enable	Enables the ANG_EVT event source
%Q0.3.00.14	Channel_0.Evt_piece_full_enable	Enables the PIECES_FULL_EVT event source
%Q0.3.00.15	Channel_0.Came_ack_flt	Command to acknowledge faults present
%Q0.3.00.23	Channel_0.Came_pieces_reset	Reset parts counter
%Q0.3.0.25	Channel_0.OUTS_ENABLE	Global enabling of track outputs
%QW0.3.0	Channel_0.ENAB_GROUP0_BITS	8 enable bits associated with the group
%QW0.3.0.1	Channel_0.Group0_and_bits	8 bits for assigning tracks to group outputs

Special Features

Managing the box counter reset

When the box counter is full, cam 6 is active.

In the example, the counter full reset is performed on activation of cam 6.

This function can only be used if the cam is a Monostable type.

NOTE: If the cam is not a monostable type it will set and reset immediately.

The parts counter can also be reset to zero by:
- the application, but pay particular attention to any problems between tasks: Do not reset the counter before cam execution.
- by a dummy cam.

%MW for the recipes

The recipes use 716 %MW words.

This zone must not be used by the system.

Saving the recipes

To be able to save the various recipes, use a paginated PCMCIA.

Hardware installation of the module TSX CCY 1128

II

Aim of this part

This part describes the hardware installation of the cam module TSX CCY 1128.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
6	General instructions for installing the TSX CCY 1128 module	79
7	Connecting an incremental and absolute SSI encoder to the TSX CCY 1128	91
8	Connecting the auxiliary inputs and track outputs of the TSX CCY 1128	113
9	Connecting an absolute encoder with parallel outputs to TSX CCY 1128	133
10	TSX CCY 1128 module displays	149

General instructions for installing the TSX CCY 1128 module

6

Subject of this chapter

This Chapter describes the general instructions for installing the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Installing the TSX CCY 1128 in a PLC station rack	80
Installing the TSX CCY 1128 in a PLC station	81
Number of application-specific channels managed by one PLC station	83
Installation precautions for the TSX CCY 1128	84
General wiring instructions	85
Selecting and protecting auxiliary power supplies	86
Choice of encoders for the TSX CCY 1128	87

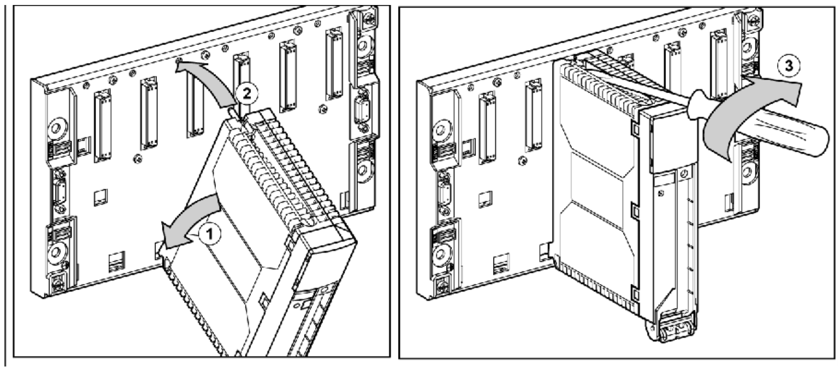
Installing the TSX CCY 1128 in a PLC station rack

Introduction

The TSX CCY 1128 module can be installed into all the available slots of the TSX RKY •• rack of a TSX 57/PMX 57/PCX 57 PLC station, with the exception of the slots taken up by the supply and processing modules.

Illustration

The diagrams below show the procedure for installing a module from the Premium range onto the TSX RKY •• rack.



Procedure

The table below describes the procedure:

Step	Action
1	Insert the pins at the back of the module into the centering holes located on the lower part of the rack.
2	Twist the module to bring it into contact with the support.
3	Fix the module onto the rack using the screw located on the upper part of the module. Maximum screw tightening: 2.0 N. m

Installing the TSX CCY 1128 in a PLC station

Introduction

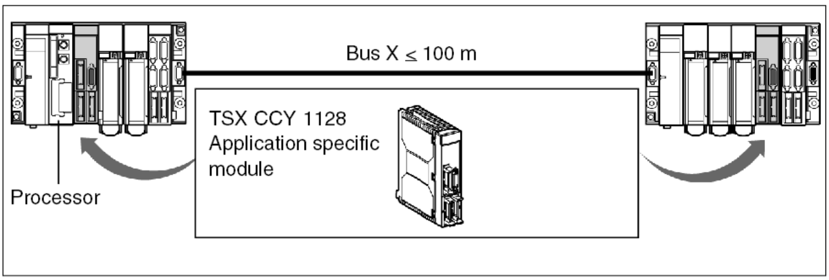
The TSX CCY 1128 module can be installed onto all PLC station racks:

- TSX RKY•• extendable racks located on the main segment of bus X, (the segment on which the rack supporting the processor is installed).
- TSX RKY•• extendable racks located on the remote segments of bus X.

Installation onto a rack belonging to the main segment of bus X

The diagram below represents the installation of the module onto a rack belonging to the main segment of bus X. The module can be installed onto the rack supporting the processor and onto all other racks present on bus X. The distance between the rack supporting the module and the rack supporting the processor must not exceed 100 meters.

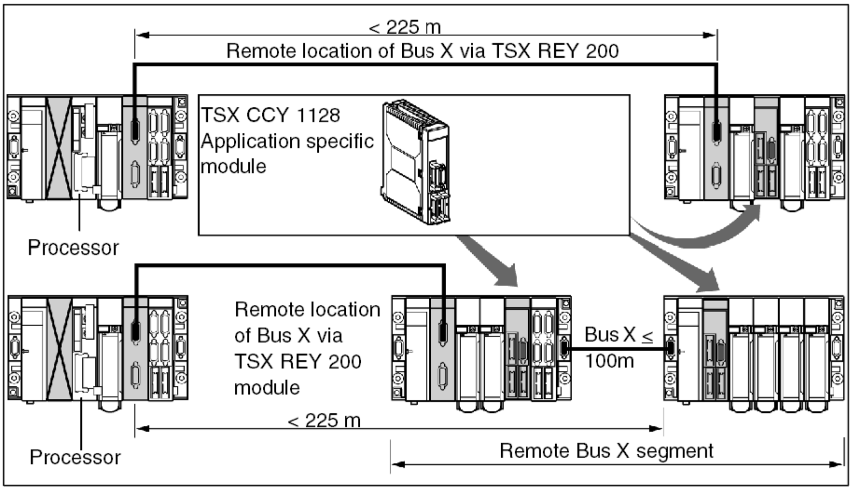
Installing the module



Installing onto a rack belonging to a remote segment of bus X

The diagram below represents the installation of the module onto a rack belonging to the remote segments of bus X. The distance between the rack supporting the module and the rack supporting the processor must, under no circumstances, exceed 225 meters.

Installing the module



Number of application-specific channels managed by one PLC station

Reasons for counting the number of specific-application channels in a station

To determine:

- the power of the CPU to be installed
- the maximum number of specific-application modules that can be installed in the station.

Number of "application-specific" channels supported:

- Premium *(see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual)*
- Atrium *(see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual)*

Installation precautions for the TSX CCY 1128

Installation

In order to guarantee good working order, it is necessary to take certain precautions during its installation and removal, when plugging and unplugging the connectors on the front panel of the module, and when adjusting its fixing screws and the SUB D 15-pin connector.

Installing and removing the module

The module can be installed or removed without cutting the supply to the rack. The design of the module allows this action to be carried out with the power on in order to ensure the availability of the device.

Plugging and unplugging the connectors on the front panel of the module

It is not recommended that you plug in or unplug the connectors located at the front panel of the module when the sensor/pre-sensor supply is switched on.

Reasons:

- the encoders will not tolerate a simultaneous start-up or outage of the signals and supplies.
- The track outputs can become damaged if they are in state 1 and connected to an inductive supply

Adjusting the screws and locking the HE10 connectors in place

In order to ensure good electrical contact between the devices and by doing so create effective resistance to electrostatic and electromagnetic interference:

- the fixing screws on the module and the SUB D 15-pin connector must be correctly screwed in.
  - tightening on the module's fixing screw: 2.0 N.m
  - tightening on the SUB D 15-pin connector's fixing screw: 0.5 N.m
- The HE10 connectors must be correctly locked.

General wiring instructions

Introduction

In order to guarantee that the automatism operates correctly, it is necessary to respect some basic rules.

Section of wires used

Must be of sufficient size to avoid on-line voltage falls and overheating.

Cable path.

The encoder connector cables, the other sensors and the pre-actuators must be kept away from any source of radiation resulting from high-power electric circuit switches and which could cause malfunctions.

Encoder signal connector cables

The module/encoder connector cables must adhere to the following rules:

- They must be shielded using a high quality shielding,
- they must only carry related signals to the encoder,
- the cable shielding must be linked to the protective ground connection both at the module and the encoder,
- the grounding must be continuous throughout the connection.

Selecting and protecting auxiliary power supplies

Introduction

Encoders, sensors and pre-actuators associated with the module require auxiliary power supplies (5VDC and/or 24VDC).

Type of power supply

Only use regulated power supplies to:

- ensure optimum reliable response time for sensors and pre-actuators,
- increase the reliability of devices by minimum heating of module I/O circuits.

These power supplies must be independent enough (> 10ms) to override micro-power outages and ensure the module continues to run effectively.

Protecting power supplies

The power supplies for encoders, others sensors and pre-actuators need calibered fast-blow fuses against overloads and short-circuits.

Connection of the 0V supply to the protective ground:

The 0V supply must be connected to the protective ground nearest to the supply module output.

General rules for installing the encoder power supply module

- this must be used only for supplying the encoder,
- it must be independent enough to override micro-power outages (> 10ms).
- it must be placed as close as possible to the TSX CCY 1128 module to reduce circuit capacities to the maximum.

Choice of encoders for the TSX CCY 1128

Introduction

The TSX CCY 1128 module inputs are able to receive signals from the following encoders:

- incremental,
- absolute with SSI serial outputs,
- absolute with parallel outputs. This last type requires the use of a specific interface TELEFAST ABE-7CPA11.

The user can choose from these encoder types according to the requirements.

Encoder output interface

The table below summarizes the main characteristics of the output interface for the encoder types normally used.

Type of encoder	Supply voltage	Output voltage	Types of interface
Incremental	5 VDC	5 VDC differential	Outputs with line transmitters to RS 422 standard, with 2 outputs per signal A+/A-, B+/B-, Z+/Z-
	10...30 VDC	10...30 VDC	Totem Pole outputs with one output per signal A, B, Z
Absolute with SSI outputs	10...30 VDC	5 VDC differential	Output with line transmitters to RS 422 standard for the data signal (Data SSI) RS 422 compatible input for the clock signal (CLK SSI).
Absolute with parallel outputs	5 VDC or 10...30 VDC	5 VDC or 10...30 VDC	Parallel outputs. Require the use of the Telefast ABE-7CPA11 interface to transform parallel output signals into serial signals.

Encoder supply

The design of the module allows an encoder supply of:

- 5 VDC
- 24 VDC, standardized voltage in the 10…30 VDC format.

The choice of supply voltage is dependent on the encoder supply voltage.

5 VDC encoder supply

For encoders with a 5 VDC supply, voltage falls must be taken into account. These are dependent upon:

- the length of the cable between the module and the encoder (double length),
- the section of wire,
- the encoder consumption.

The acceptable voltage fall for the encoder is generally 10% of the nominal voltage.

The table below gives the on-line voltage fall, according to the section of the wire, for a 100 meter length of wire with a given encoder consumption.

Section of wire	**Voltage fall for a 100 meter length of wire with an encoder consumption of:**			
	50 mA	**100 mA**	**150 mA**	**200 mA**
0.22 mm = gauge 24	0.4 V	-	-	-
0.34 mm = gauge 22	0.25 V	0.5 V	-	-
0.5 mm	0.17 V	0.34 V	0.51 V	-
1 mm	0.09 V	0.17 V	0.24 V	0.34 V

CAUTION

MODULE DESTRUCTION

Do not compensate an online voltage fall by the raise of the encoder power supply voltage.

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

24 VDC encoder supply

Encoders with a supply voltage of 24 VDC are recommended for the following reasons:

- the supply source does not need to be completely accurate. As a general rule, these encoders use a supply format of 10…30 V.
- an on-line voltage fall is of little significance due to a substantial distance between the module and the encoder.

Ground connection continuity

In order to ensure correct operation during interference, it is vital:

- to choose an encoder with a metal casing that is referenced to the protective ground of the connected device.
- that the ground connection is continuous between:
  - the encoder,
  - the shielding of the connector cable,
  - the module.

Connecting an incremental and absolute SSI encoder to the TSX CCY 1128

7

Subject of this chapter

This Chapter describes the procedure for connecting an incremental and absolute SSI encoder to the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Principles for connecting an encoder to a TSX CCY 1128	92
Connecting an incremental encoder with RS422 outputs to the TSX CCY 1128	93
Connecting an incremental encoder with Totem Pole outputs to the TSX CCY 1128	96
Connecting an absolute SSI encoder to the TSX CCY 1128	99
Connecting an encoder supply monitor to the TSX CCY 1128	102
Connecting the encoder supply to the TSX CCY 1128	104
TSX CAP S15 connection accessory	107
TSX TAP S1505/S1524 and TSX CCP S15x Connecting Accessories	108

Principles for connecting an encoder to a TSX CCY 1128

Connection interfaces

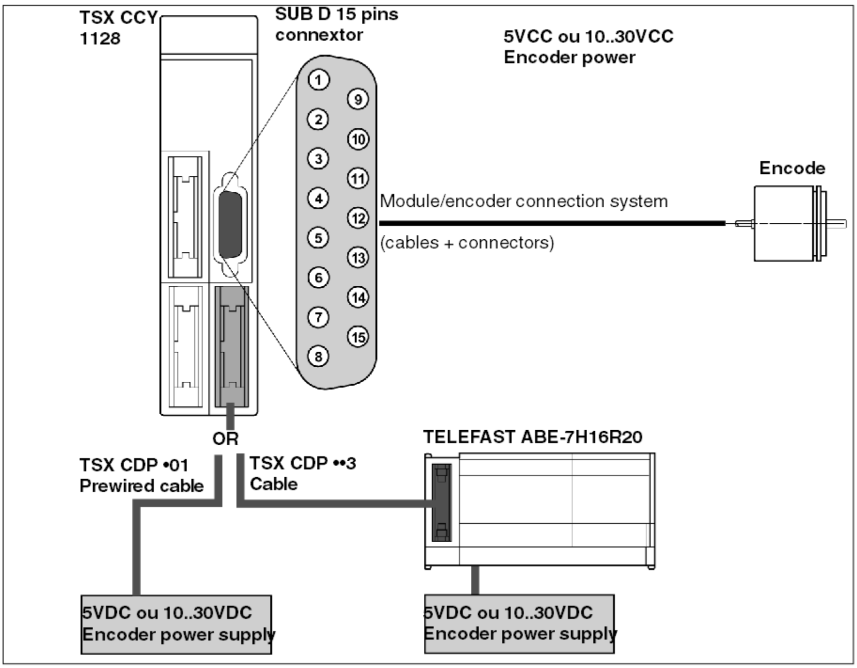
The 15-pin SUB D connector, located on the front panel of the module, allows the connection of encoder and module. This connector carries:

- all signals travelling to or from the encoder,
- the encoder supply source, which is also connected:
  - either by a TELEFAST ABE-7H16R20 cable interface.
  - or directly, using a TSX CDP •01 prewired strand

The design of the module allows the encoder to be connected at either 5VDC or 10…30VDC.

Illustration

The diagram below shows the principles for connecting an encoder to a TSX CCY 1128 module.



Connecting an incremental encoder with RS422 outputs to the TSX CCY 1128

Pinouts of the module's 15-pin SUB D connector

The pinout configuration of the module's 15-pin SUB D connector, for connection to an incremental encoder with RS 422 outputs, are as follows:

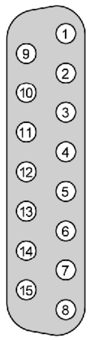
Diagram (front view)	Pin No.	Signal	Designation
	1	A+ 5V	Encoder input, pulse A+ (5VDC)
	2	A-	Encoder input, pulse A-
	3	-	-
	4	Z+ 5V	Encoder input, zero latch pulse Z+ (5VDC)
	5	Z-	Encoder input, zero latch pulse Z-
	6	-	-
	7	10…30 V	Encoder supply output (+ 10…30VDC)
	8	0 V	Encoder supply output (- 0VDC)
	9	-	-
	10	B+	Encoder input, pulse B+ (5VDC)
	11	B-	Encoder input, pulse B-
	12	-	-
	13	EPSR	Positive encoder supply feedback input. Receives positive supply feedback from the encoder, which allows the module to verify the encoder's presence.
	14	-	-
	15	5 V	Encoder supply output (+ 5VDC)

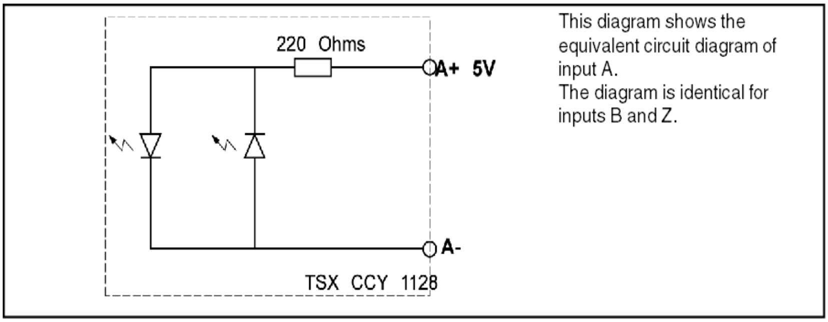
Diagram showing the A, B and Z encoder inputs of the module

The diagram below shows an A, B or Z encoder input used with an incremental encoder equipped with:

- a high-level output with a line transmitter,
- a 5VDC output voltage to the RS 422 standard.

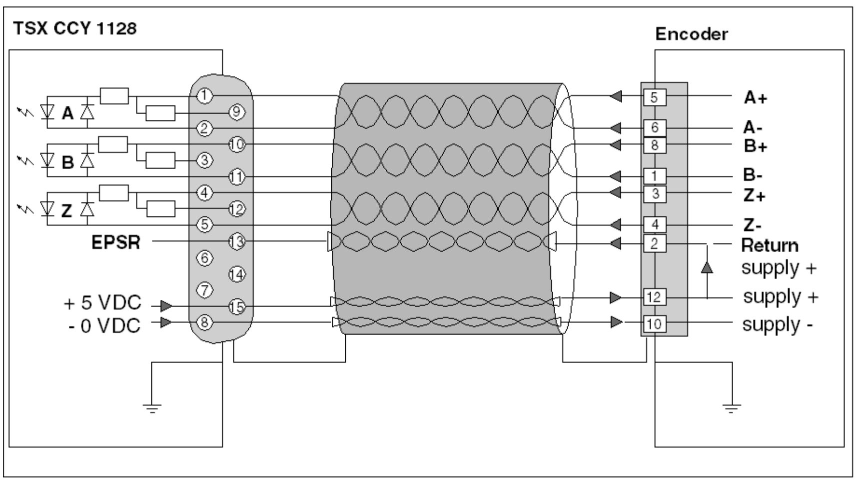
NOTE: each A, B and Z input uses a differential line monitor.

Diagram of input A



Connection process diagram

The diagram below describes the process of connecting a TSX CCY 1128 module to an incremental encoder with RS 422 outputs and a 5VDC supply.



Recommendations

When connecting, adhere to the following recommendations:

- Connect each signal encoder A+/A-, B+/B-, Z+/Z-, using a twisted pair.
- Connect each supply point using a twisted pair, in order to reduce the on-line voltage falls.
- Connect both terminals of the cable shielding to the protective ground.

⚠ CAUTION
ENCODER DAMAGE - PINOUT CONFIGURATION
Before connecting the encoder to the module, check the pinout configuration given by the encoder’s manufacturer.
Failure to follow these instructions can result in injury or equipment damage.

Connecting an incremental encoder with Totem Pole outputs to the TSX CCY 1128

Pinout configuration of the module’s 15-pin SUB D connector

The pinout configuration of the module’s 15-pin SUB D connector, for connection to an incremental encoder with Totem Pole outputs, is as follows:

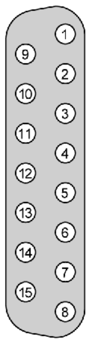
Diagram (front view)	Pin No.	Signal	Designation
	1	-	-
	2	A-	Input to be connected to 0V encoder
	3	B+ 24V	Encoder input, pulse B+ (10…30VDC)
	4	-	-
	5	Z-	Input to be connected to 0V encoder
	6	-	-
	7	10…30 V	Encoder supply output (+ 10…30VDC)
	8	0 V	Encoder supply output (- 0VDC)
	9	A+ 24V	Encoder input, pulse A+ (10…30VDC)
	10	-	-
	11	B-	Input to be connected to 0V encoder
	12	Z+ 24V	Encoder input, zero latch pulse Z+ (10…30VDC)
	13	EPSR	Positive encoder supply feedback input. Receives positive supply feedback from the encoder, which allows the module to verify the encoder’s presence.
	14	-	-
	15	5 V	Encoder supply output (+ 5VDC)

Diagram showing the A, B and Z encoder inputs of the module

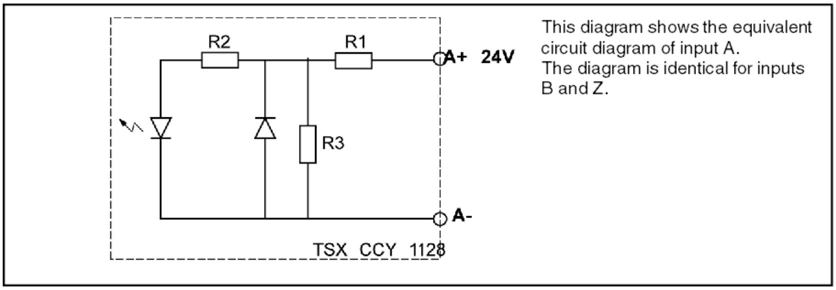
The diagram below shows an A, B or Z encoder input used with an incremental encoder equipped with:

- a Totem Pole type high-level output,
- an output voltage of 10…30VDC.

NOTE:

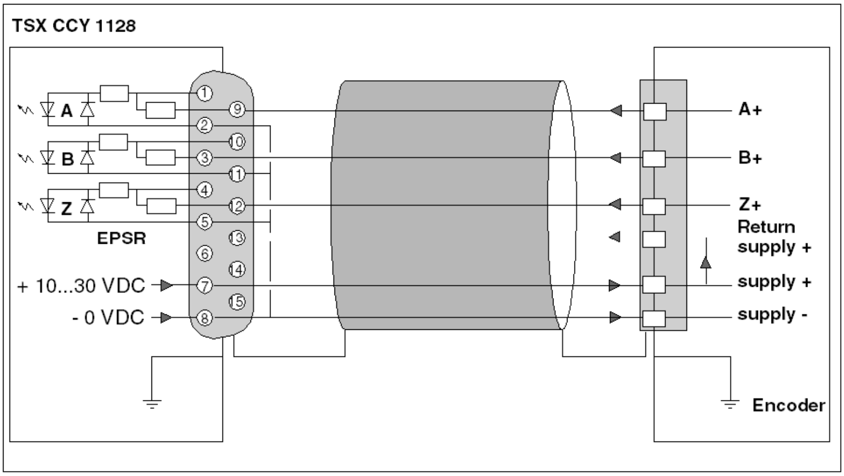
- differential assembly is not possible, the negative polarity of each input (A-, B- and Z-), must be linked to the 0V encoder and the positive inputs (A+, B+ and Z+) to encoder outputs A+, B+ and Z+.
- No line monitoring.

Diagram of input A



Connection process diagram for the encoder

The diagram below describes the process of connecting an incremental encoder with Totem Pole outputs, with a 10...30VDC supply.



Recommendations

- connect the positive encoder supply to pin 7 of the module's 15-pin SUB D connector,
- connect the module's EPSR input to the positive supply if the encoder does not have a positive supply feedback output.
- connect both terminals of the cable shielding to the protective ground.

⚠ CAUTION
ENCODER DAMAGE - PINOUT CONFIGURATION
Before connecting the encoder to the module, check the pinout configuration given by the encoder's manufacturer.
Failure to follow these instructions can result in injury or equipment damage.

Connecting an absolute SSI encoder to the TSX CCY 1128

Pinout configuration of the module’s 15-pin SUB D connector

The pinout configuration of the module’s 15-pin SUB D connector, for connection to an absolute SSI encoder, is as follows:

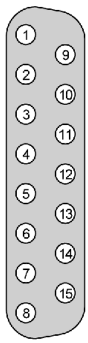
Diagram (front view)	Pin No.	Signal	Designation
	1	positive SSI data	Encoder input, positive SSI data (5VDC)
	2	negative SSI data	Encoder input, negative SSI data
	3	-	-
	4	-	-
	5	-	-
	6	CLK +	Encoder output, positive SSI CLK (5VDC)
	7	10…30 V	Encoder supply output (+ 10…30VDC)
	8	0 V	Encoder supply output (- 0VDC)
	9	-	-
	10	-	-
	11	-	-
	12	-	-
	13	EPSR	Positive encoder supply feedback input. Receives positive supply feedback from the encoder, which allows the module to verify the encoder’s presence.
	14	CLK -	Encoder output, negative SSI CLK
	15	5 V	Encoder supply output (+ 5VDC)

Diagram of the module's SSI Data encoder input

The diagram below shows the SSI Data encoder input used with an absolute SSI encoder equipped with:

- a high-level output with a line transmitter,
- a 5VDC output voltage to the RS 422/RS 485 standard.

NOTE: the SSI Data input uses a differential line monitor.

Diagram of input A:

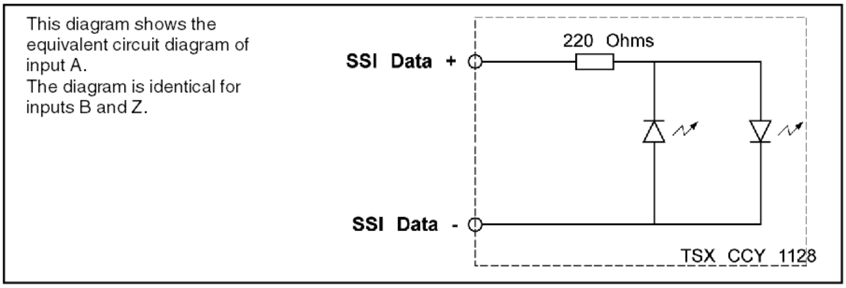
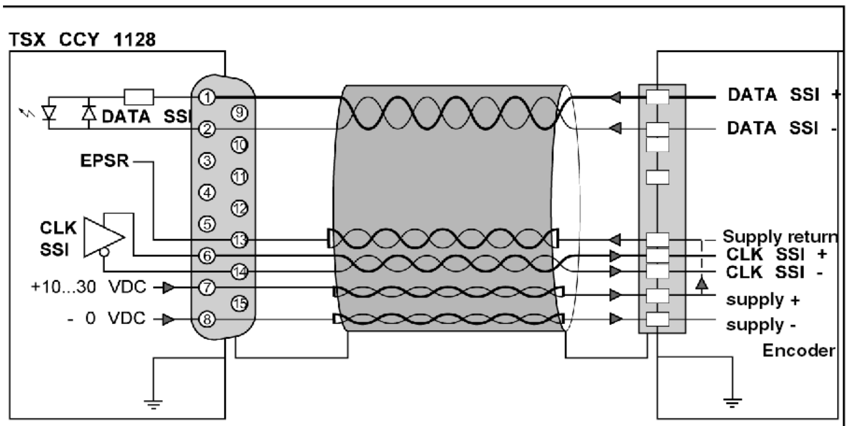


Diagram showing the process for connecting an absolute SSI encoder

The diagram below describes the process for connecting an absolute SSI encoder with a serial link, a 10…30VDC supply and outputs with standard RS 422 line transmitters.



Recommendations

- Connect each DATA SSI+/SATA SSI- encoder signal using a twisted pair,
- Connect each supply point using a twisted pair, in order to reduce the on-line voltage falls
- Connect both terminals of the cable shielding to the protective ground.
- If the encoder does not have a positive supply feedback output, connect the module's EPSR input to the positive supply of the encoder.

⚠ CAUTION
ENCODER DAMAGE - PINOUT CONFIGURATION Before connecting the encoder to the module, check the pinout configuration given by the encoder's manufacturer. **Failure to follow these instructions can result in injury or equipment damage.**

Connecting an encoder supply monitor to the TSX CCY 1128

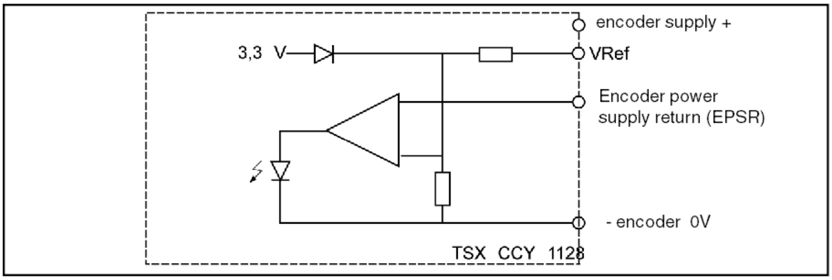
Principle

The EPSR input signal from the encoder is comparable to:

- either a fixed internally generated voltage of 3.3V if the Vref input is not connected.
- or a voltage equal to 66% of the voltage recorded at the Vref input, positive polarity of the encoder supply voltage.

Diagram of the encoder return supply monitor

The diagram below is of the encoder supply feedback monitor.



Using the VRef input

The table below summarizes the use of the Vref input according to the encoder supply voltage.

If	Then
The encoder is using a 5V supply	The Vref input is not connected to the positive encoder supply. The EPSR signal is comparable to the internal voltage of 3.3V. OK if > 3.3V
The encoder is using a 10…30V supply	The Vref input is connected to the positive encoder supply. The ESPR signal is comparable to 66% of the encoder supply voltage. OK if >66%

Diagram showing the connection process for an encoder on a 10…30V supply

The diagram below shows the connection process for the encoder supply feedback monitor if the encoder is using a 10…30 V supply.

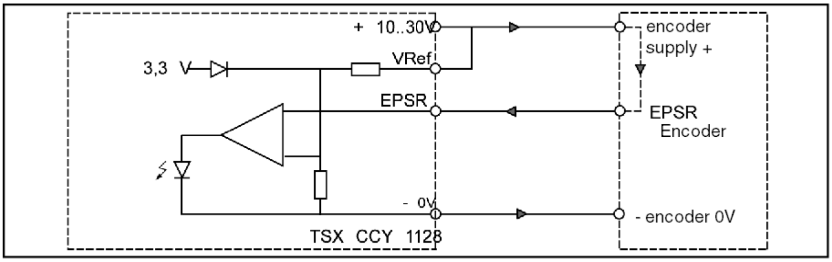


Diagram showing the connection process for an encoder on a 5V supply

The diagram below gives the process for connecting the encoder supply feedback monitor if the encoder is using a 5V supply.

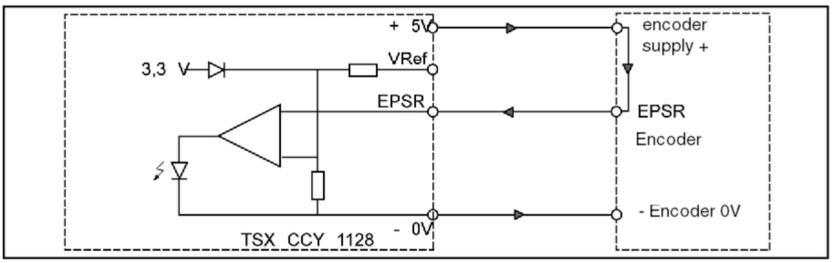
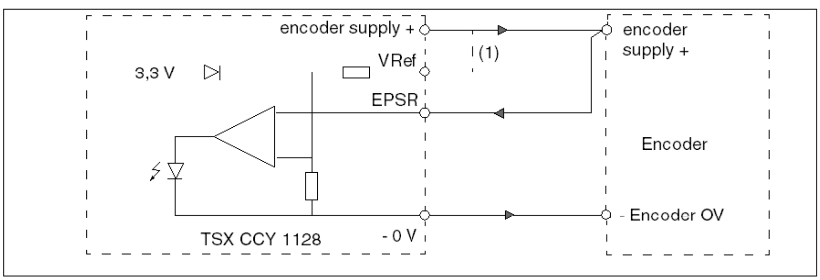


Diagram of the connection process if the encoder has no supply feedback

In this case, the EPSR input is connected to the positive supply at the encoder.



Connecting the encoder supply to the TSX CCY 1128

Introduction

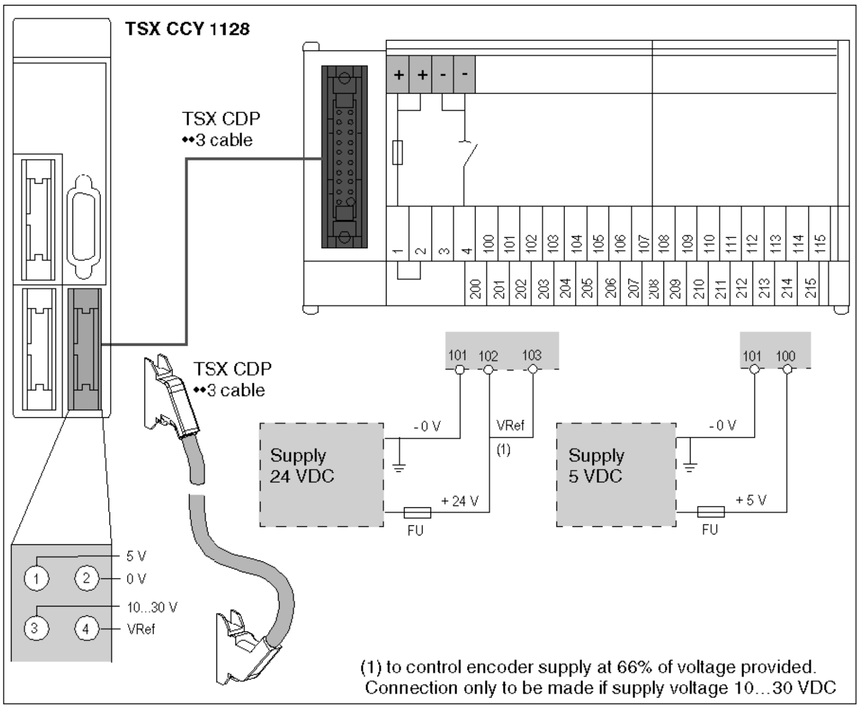
The encoder supply can be connected:

- either by using a TELEFAST ABE-7H16R20 cable interface, which is then connected to the module using a TSX CDP ••3 cable.
- or directly, using a TSX CDP •01 pre-wired strand

Process diagram for connecting the encoder supply to the TELEFAST interface

The diagram below shows the process for connecting the encoder supply.

- At 24 VDC for an encoder with a 10…30 VDC supply format,
- and at 5 VDC for an encoder with a 5 VDC supply.



Catalog of TSX CDP ••3 connector cables

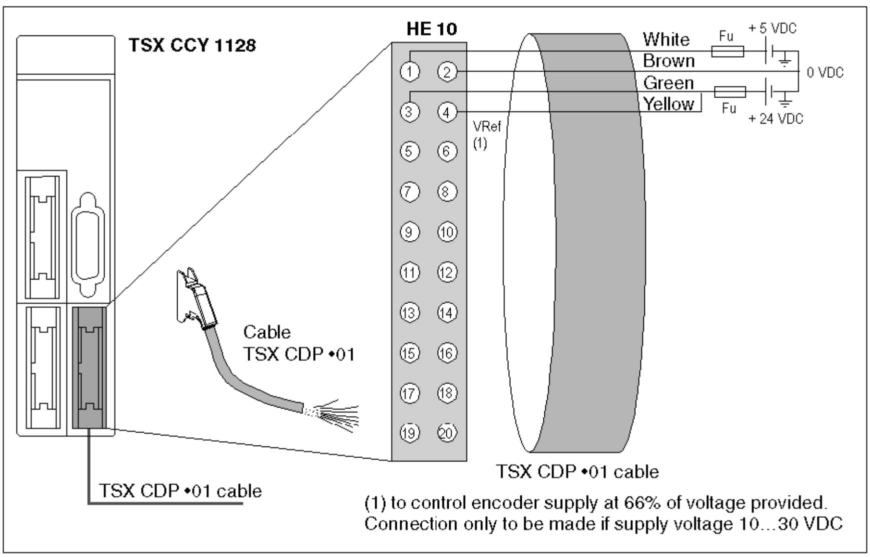
The table below gives the different references for the cables connecting the TELEFAST to the module, and their respective lengths.

Cable references	Cable lengths
TSX CDP 053	0.5 meters
TSX CDP 103	1 meter
TSX CDP 203	2 meters
TSX CDP 303	3 meters
TSX CDP 503	5 meters

Diagram showing the process for connecting the supply using a TSX CDP •01 pre-wired strand

The diagram below shows the process for connecting the encoder supply.

- At 24 VDC for an encoder with a 10…30 VDC supply format,
- and at 5 VDC for an encoder with a 5 VDC supply.



Catalog of TSX CDP •01 connector cables

The table below gives the different references for the cables connecting the TELEFAST to the module, and their respective lengths.

Cable references	Cable lengths
TSX CDP 301	3 meters
TSX CDP 501	5 meters

Recommendations

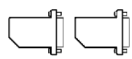
- Maximum length of wires between the supply outputs and the connection points on the TELEFAST: must be less than 0.5 meters.
- Protection on the + supply: although the module has several built-in protection systems to guard against wiring errors and accidental short-circuits on the cables, it is vital to install a 1A maximum non-delay fuse (Fu) on the + supply.
- Connection of the 0 V supply to the protective ground: must be as close as possible to the supply output.

TSX CAP S15 connection accessory

At a Glance

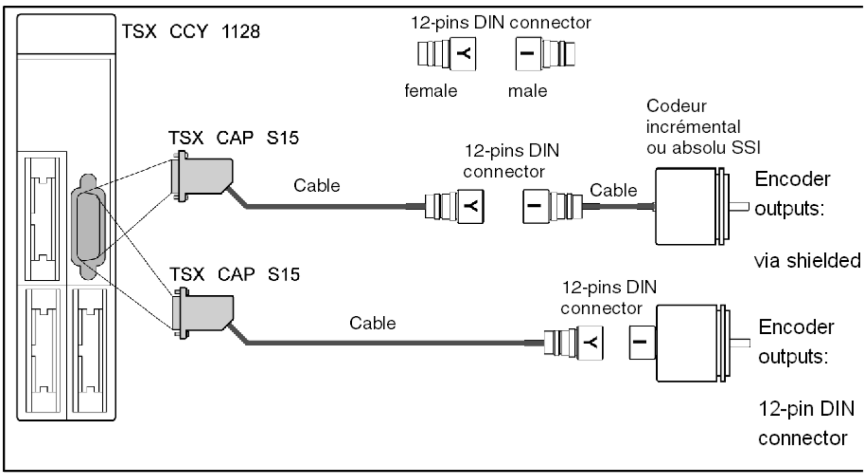
The TSX CAP S15 accessory comprises 2 15-pin SUB D connectors, and ensures the link between the module’s connection interface and the encoder’s connection string.

The table below introduces the TSX CAP S15 connection accessory.

Diagram	References	Use	Composition
	TSX CAP S15	Can be used in the module/encoder connection string	Kit comprises 2 15-pin SUB D connectors with covers.

Integrating the TSX CAP S15 into the connection string

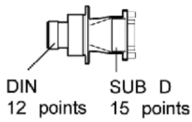
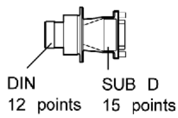
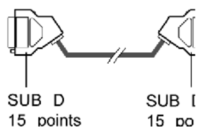
The diagram below shows how to integrate a 15-pin SUB D connector from the TSX CAP S15 kit into the encoder connection string.



TSX TAP S1505/S1524 and TSX CCP S15x Connecting Accessories

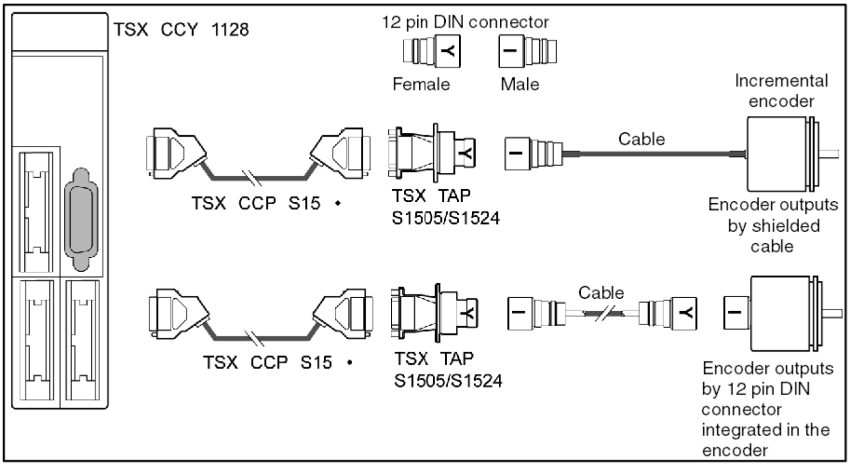
Presentation of the Various Accessories

The following table presents the various accessories, their uses and functions.

Representation	Product references	Use	Function
	TSX TAP S1505	May be used in the module/encoder connection series to connect a 5 VDC-powered incremental encoder with RS 422 outputs and A+/A-, B+/B- & Z+/Z- signals.	Mechanical interface, fitted with two connectors which enables conversion from a 15 pin SUB D connection system to a 12 pin DIN connection system.
	TSX TAP S1524	May be used in the module/encoder connection series to connect a 24 VDC-powered encoder with Totem Pole outputs and A, B & Z signals.	Mechanical interface, fitted with two connectors which enables conversion from a 15 pin SUB D connection system to a 12 pin DIN connection system.
	TSX CCP S15•	May be used in the module/encoder connection series to connect the module to the TSX TAP S1505 or TSX TAP S1524	Connector cables made up of a 24 gauge wiring cable and fitted with 15 pin SUB D connector at each end.

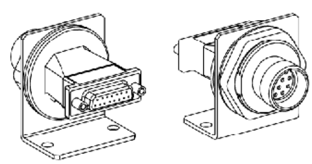
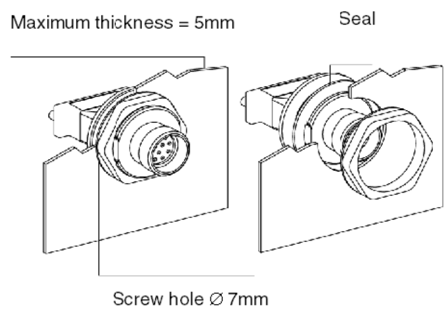
Integration of TSX TAP S1505/S1524 and TSX CCP S15• Accessories in the Connection Series.

The figure below shows the integration of an encoder for the TSX CCP S15, TSX TAP S1505 and TSX TAP S1524 accessories in the connection series.



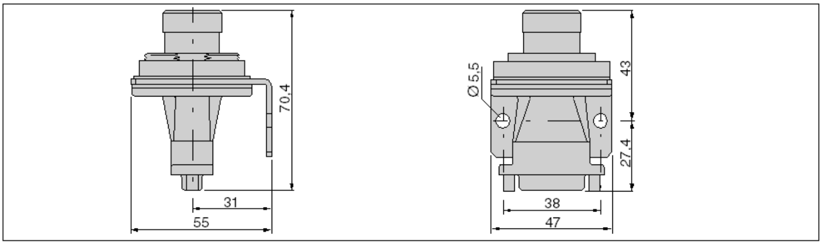
Mounting of the TSX TAP S1505/S1524 Accessories

The figures below show the two mounting procedures for these accessories.

Mounting on an AM1-PA• Telequick mounting grid	**Cabinet bushing mounting**
	

Dimensions of the TSX TAP S1505/S1524 Accessories

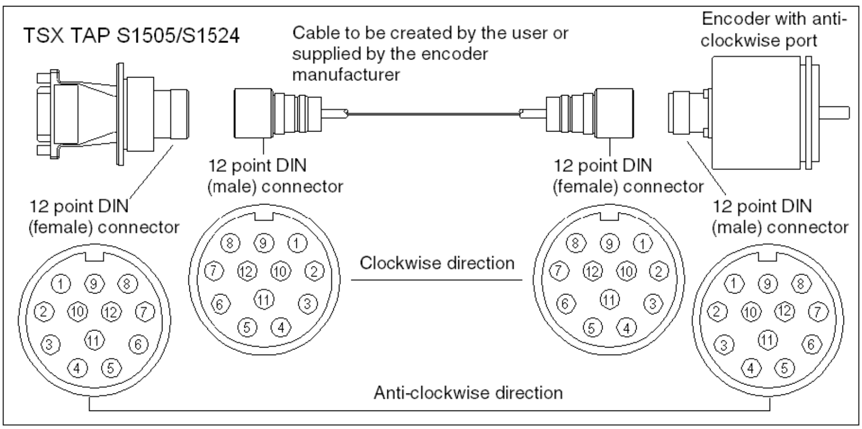
The below figure shows the dimensions and the mount center distances on the AM1-PA• Telequick mounting grid for the TSX TAP S1505 and TSX TAP S1524 accessories.



Anti-clockwise Direction of the Connection Series' 12 Pin DIN Connectors

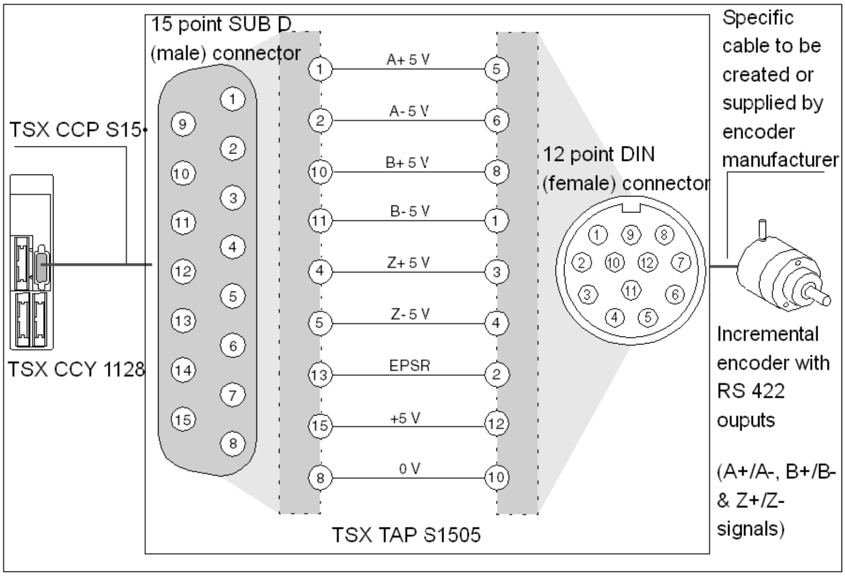
To facilitate the connections, the numbers of the pins on the connection series' various 12 pin DIN connectors must match up, one to one . For this, the connector pin assignment must be:

- in a clockwise direction for connectors that belong to the cable linking the TSX TAP S1505/S1524 accessory to the encoder,
- in an anti-clockwise direction for connectors that belong to the encoder and the TSX TAP S1505/S1524 accessory.



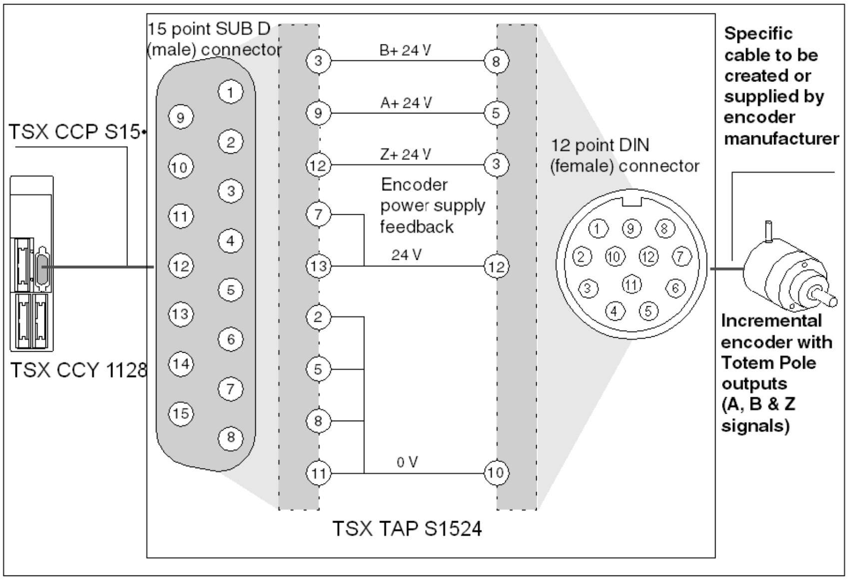
Pin Assignment for the TSX TAP S1505 Accessory

The below figure shows the pin assignment for the TSX TAP S1505 accessory on both the 15 pin SUB D connector and 12 pin DIN connector sides.



Pin Assignment for the TSX TAP S1524 Accessory

The below figure shows the pin assignment for the TSX TAP S1524 accessory on both the 15 pin SUB D connector and 12 pin DIN connector sides.



Connecting the auxiliary inputs and track outputs of the TSX CCY 1128

8

Subject of this chapter

This Chapter describes the connection procedure for the auxiliary inputs and track outputs of the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Introduction to the connection interfaces of the TSX CCY 1128	114
Connecting the auxiliary inputs of the TSX CCY 1128	117
Connecting the track outputs of the TSX CCY 1128	125

Introduction to the connection interfaces of the TSX CCY 1128

Connection interfaces

The three HE10 connectors located on the front panel of the module are for connecting:

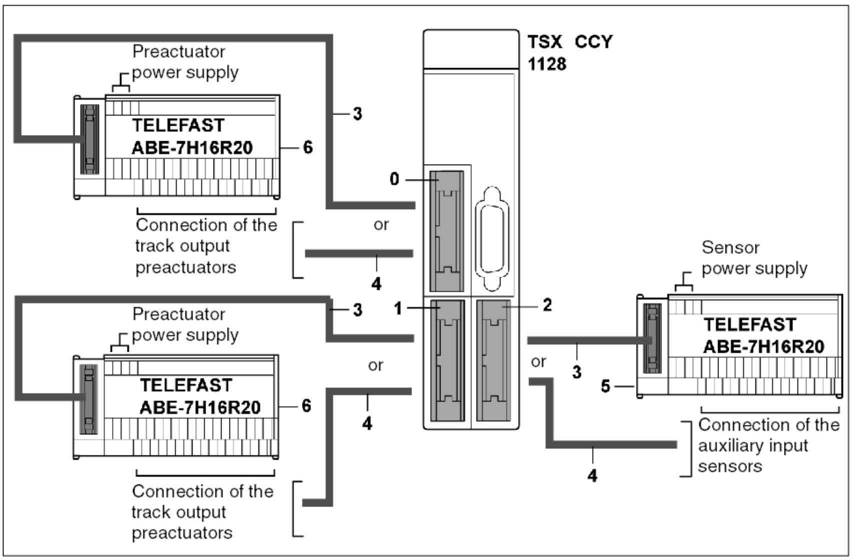
- the sensors and pre-actuators inline with the module inputs/outputs,
- the sensor and pre-actuators supplies inline with the module inputs/outputs.

The module is connected to the sensors and pre-actuators using:

- either TELEFAST connectors and TSX CDP ••3 cables (recommended),
- or TSX CDP •01 strips.

Illustration

The diagram below shows the input/output connection strip of a TSX CCY 1128 module.



Elements and their functions

The table below gives the different elements of the connection strip.

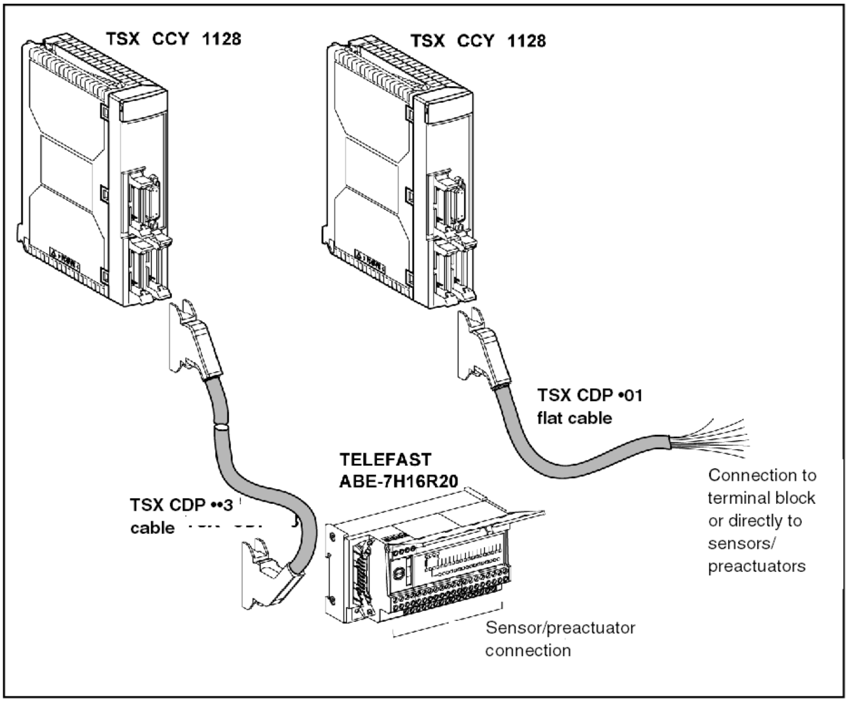
Number	Elements	Functions
0	20-pin HE10 type connector	Allows connection of: ● the pre-actuators controlled by the group 0 and 1 track outputs ● the pre-actuator supply
1	20-pin HE10 type connector	Allows connection of: ● the pre-actuators controlled by the group 2 and 3 track outputs ● the pre-actuator supply
2	20-pin HE10 type connector	Allows connection of: ● the sensors that control the auxiliary inputs, ● the sensor supply, ● the incremental or absolute SSI encoder supply
3	TSX CDP••3 cables, equipped with a HE10 connector at each terminal	Allow the module to be connected to the TELEFAST connector. 5 lengths are suggested: ● TSX CDP 053: 0.5 meters in length ● TSX CDP 103: 1 meter in length ● TSX CDP 203: 2 meters in length ● TSX CDP 303: 3 meters in length ● TSX CDP 503: 5 meters in length
4	TSX CDP•01 strips, equipped with a HE10 connector at one terminal, and free wires at the other, identified using a color code	Allow direct connection between the module inputs/outputs and the sensors and pre-actuators. 2 lengths are suggested: ● TSX CDP 301: 3 meters in length ● TSX CDP 501: 5 meters in length
5	TELEFAST ABE-7H16R20 connectors	Connects a HE10 connection and a screw terminal block connection, allowing a rapid connection of supplies, sensors and pre-actuators

Connection accessories

The sensors are connected to the auxiliary inputs, and the pre-actuators to the track outputs by:

- either using a TELEFAST ABE-7H16R20 connector and TSX CDP ••3 cable (recommended system),
- or directly, using TSX CDP •01 strips

The diagram below shows these two types of connection



Connecting the auxiliary inputs of the TSX CCY 1128

Number of auxiliary inputs

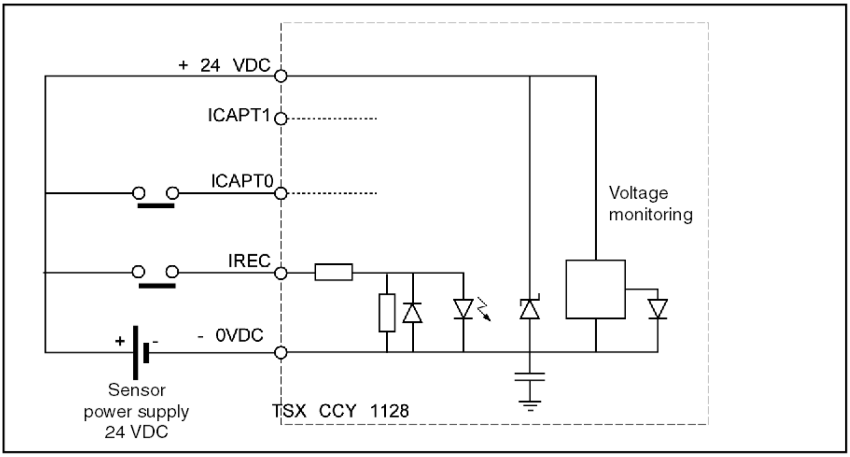
The TSX CCY 1128 module has 3 auxiliary inputs:

Auxiliary	Functions
IREC	Adjustment of the position measurement
ICAPT0	Position capture in register 0
ICAPT1	Position capture in register 1

Diagram

The auxiliary inputs use a 24 VDC from an external supply provided via the connector.

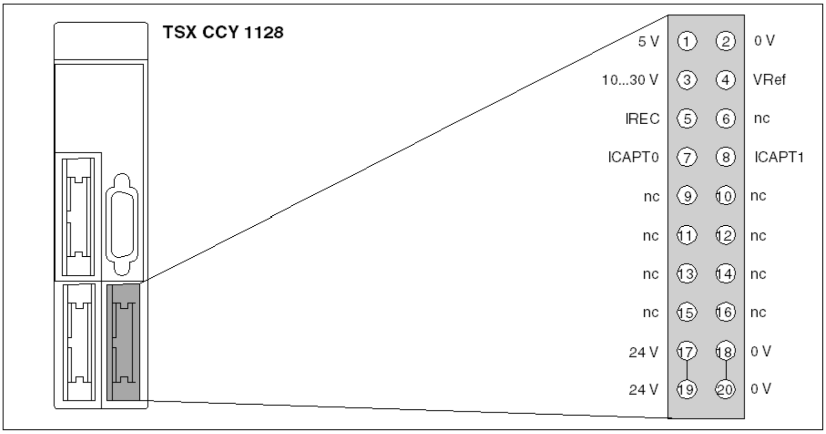
The diagram below shows this supply.



Positioning the HE10 connector, and identifying the signals

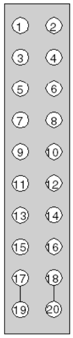
The diagram below represents the position of the HE10 connector on the module in relation to the auxiliary inputs and the identification of the different signals delivered by the connector.

This connector delivers exactly the same supply as the incremental or absolute SSI encoder. This information will be elaborated in Chapter 3.



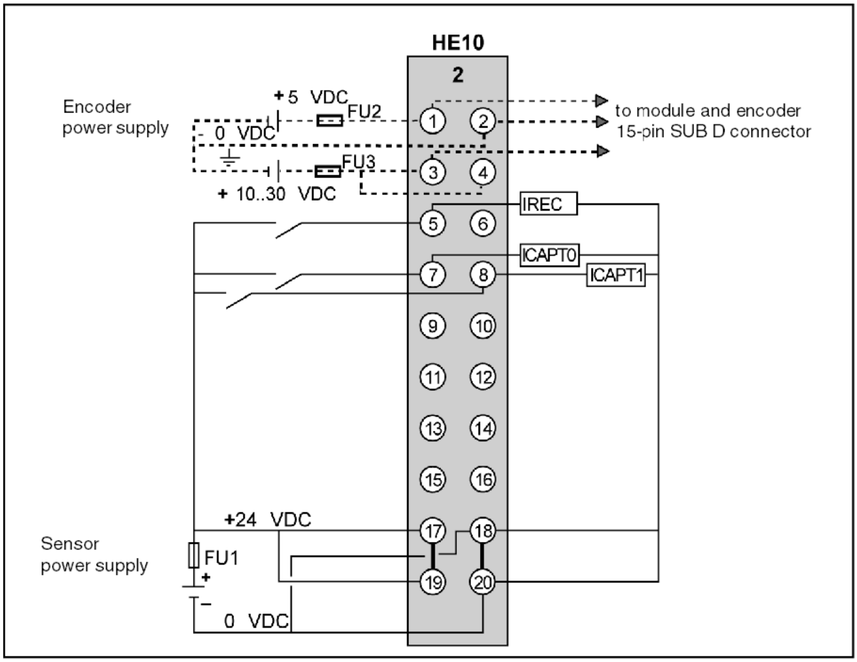
Pinout configuration of the module's HE10 connector

The pinout configuration of the HE10 connector in relation to the auxiliary inputs is as follows:

Diagram (front view)	Pin No.	Signal	Designation
	1	5 V	Encoder supply input +5 VDC
	2	0 V	Encoder supply input – 0VDC
	3	10..30V	Encoder supply input + 10…30VDC
	4	VRef	Reference input voltage for encoder supply monitoring
	5	IREC	Auxiliary adjustment input
	6	-	Not wired
	7	ICAPT0	Auxiliary capture input 0
	8	ICAPT1	Auxiliary capture input 1
	9	-	Not wired
	10	-	Not wired
	11	-	Not wired
	12	-	Not wired
	13	-	Not wired
	14	-	Not wired
	15		Not wired
	16		Not wired
	17	24 V	Sensor supply input + 24VDC
	18	0 V	Sensor supply input - 0VDC
	19	24 V	Sensor supply input + 24VDC
	20	0 V	Sensor supply input - 0VDC

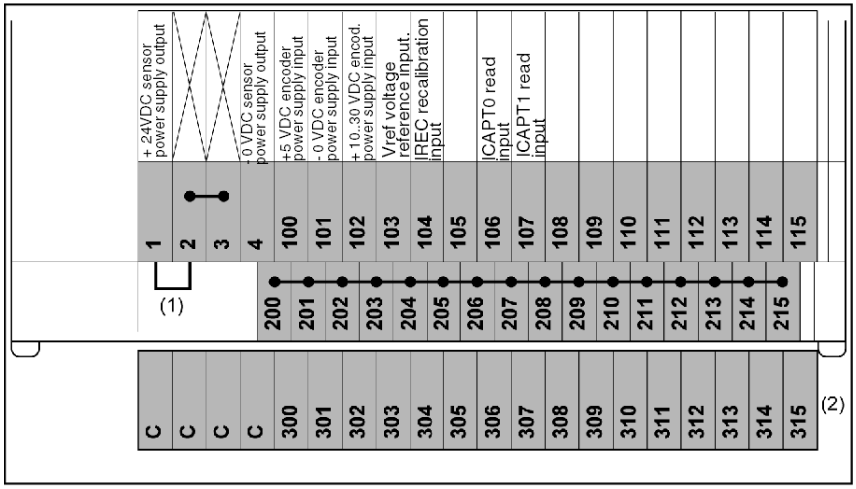
Process diagram

The diagram below represents the process for connecting the auxiliary inputs



Connection using TELEFAST connector and TSX CDP ••3 cable

Availability of signals at the screw terminal block of the TELEFAST connector:



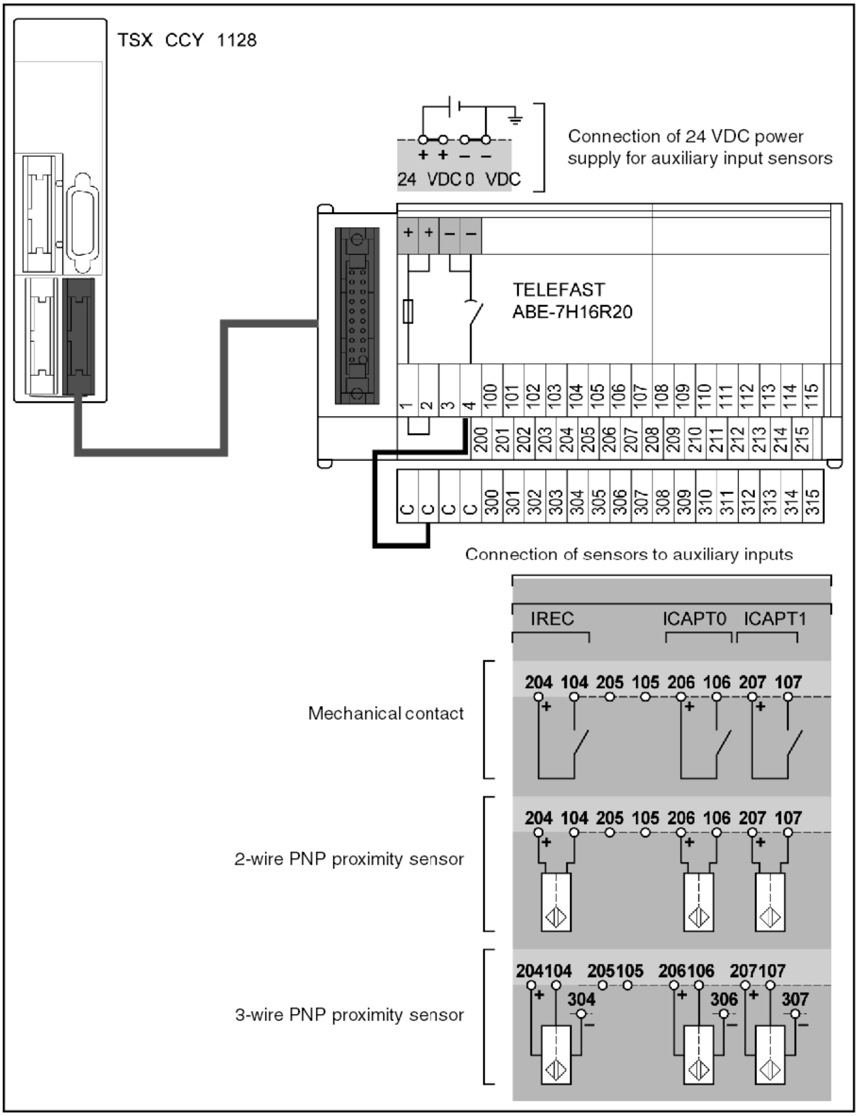
(1) At the TELEFAST ABE-7H16R20 connector, the position of the jumper wire determines the polarity of all terminals from 200 to 215:

- jumper wire in position 1 or 2: terminals 200 to 215 have positive polarity,
- jumper wire in position 2 or 4: terminals 200 to 215 have negative polarity.

(2) At the TELEFAST ABE-7H16R20 connector, it is possible to add an optional ABE-7BV20 bar to create a second shared sensor (positive or negative according to user’s choice).

Connection using TELEFAST connector and TSX CDP ••3 cable

Example for connecting sensors to the auxiliary inputs:



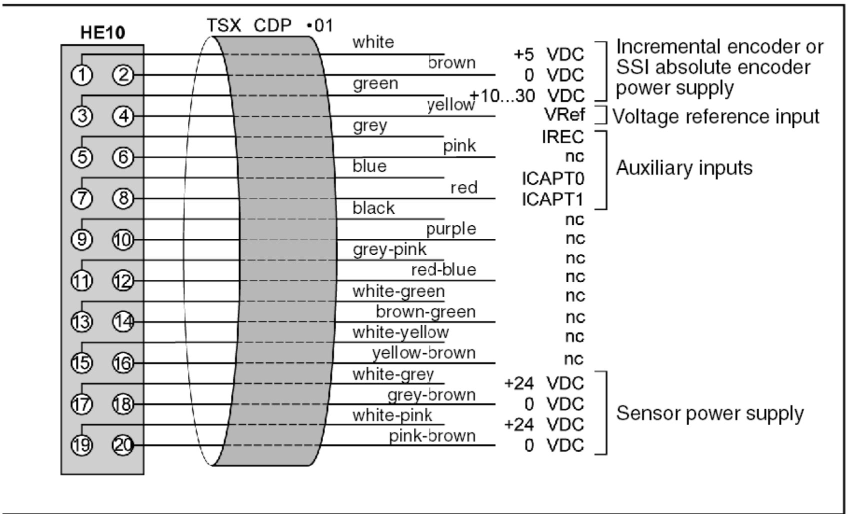
Correspondence between the TELEFAST terminal block and the module's HE10 connector:

TELEFAST screw terminal block (No. of terminal)	HE10 connector 20 points (pin number)	Type of signal	Functions
100	1	+ 5VDC	Encoder supply
101	2	- 0 VDC	
102	3	+10...30VDC	
103	4	VRef	Reference input voltage for encoder supply feedback monitoring
104	5	IREC	Adjustment input
105	6	Not connected	-
106	7	ICAPT0	Capture input register 0
107	8	ICAPT1	Capture input register 1
108 to 115	9 to 16	Not connected	-
+ 24VDC	17	-	Auxiliary input sensor supply
- 0 VDC	18	-	
+ 24VDC	19	-	
- 0 VDC	20	-	
1	-	-	Terminals 200 to 215 at + 24VDC if terminals 1 and 2 are connected
2	-	-	
3	-	-	Terminals 200 to 215 at - 24VDC if terminals 3 and 4 are connected
4	-	-	
200...215	-	-	Connecting shared sensors to: ● +24VDC if terminals 1 and 2 are connected ● - 0VDC if terminals 3 and 4 are connected
300...315	-	-	At the optional ABE-7BV20 bar, the terminals that can be used as a shared sensor must be connected, using a wire, to the desired shared voltage.

Connection using TSX CDP •01 strips

This type of connection is used to connect all signals travelling to or from the module directly:

- at a terminal block, or
- at the sensors.



nc – not connected

Connecting the track outputs of the TSX CCY 1128

Number of track outputs and their distribution

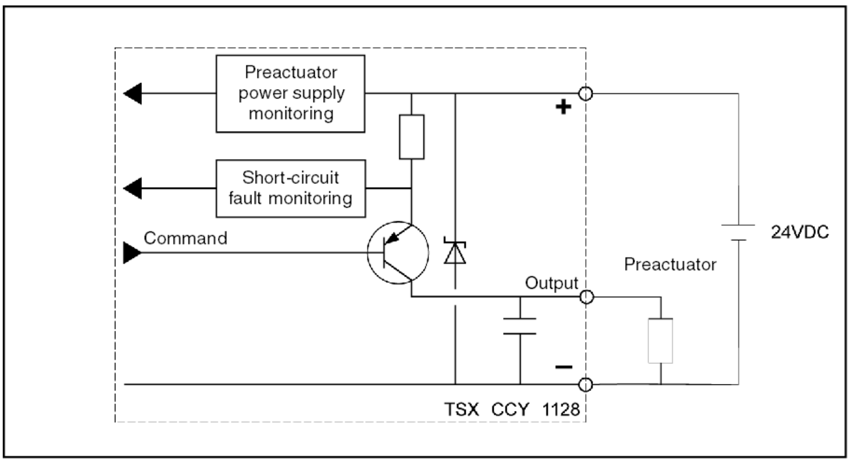
The TSX CCY 1128 module has 32 track outputs of which 24 are physically accessible.

These track outputs are divided into 4 groups on two HE10 connectors at the front panel of the module

Connectors	0		1	
Groups	0	1	2	3
Tracks	01234567	0123	01234567	0123
Outputs	Q0. 01234567	Q1. 0123	Q2. 01234567	Q3. 0123

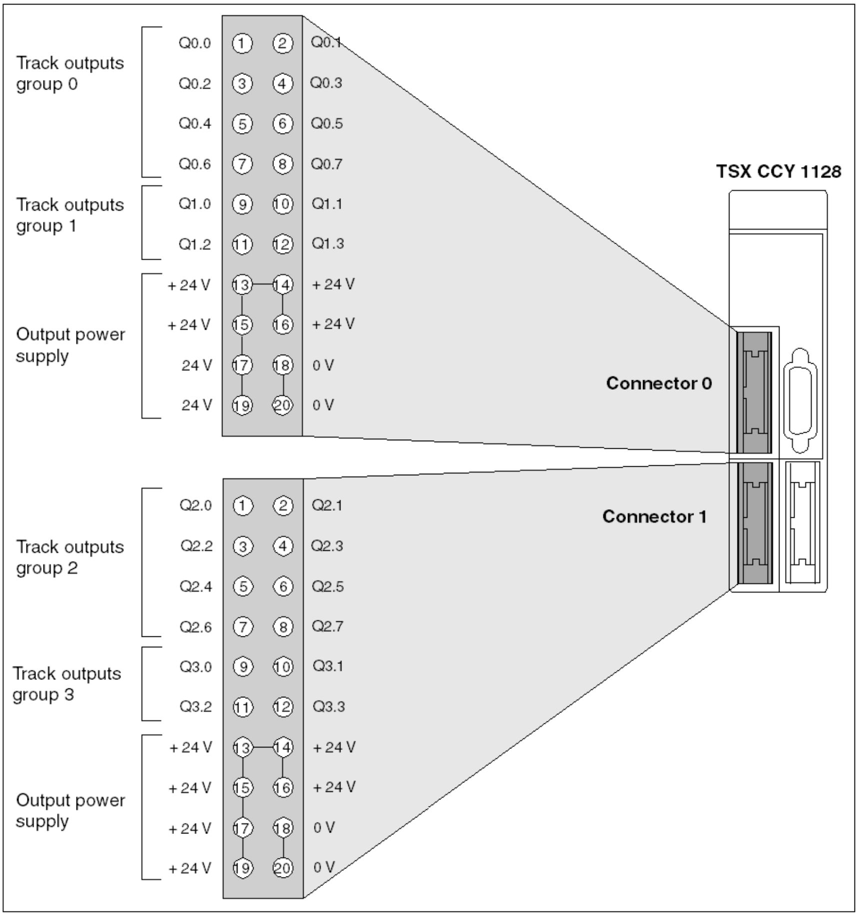
Diagram

The diagram below shows a track output.



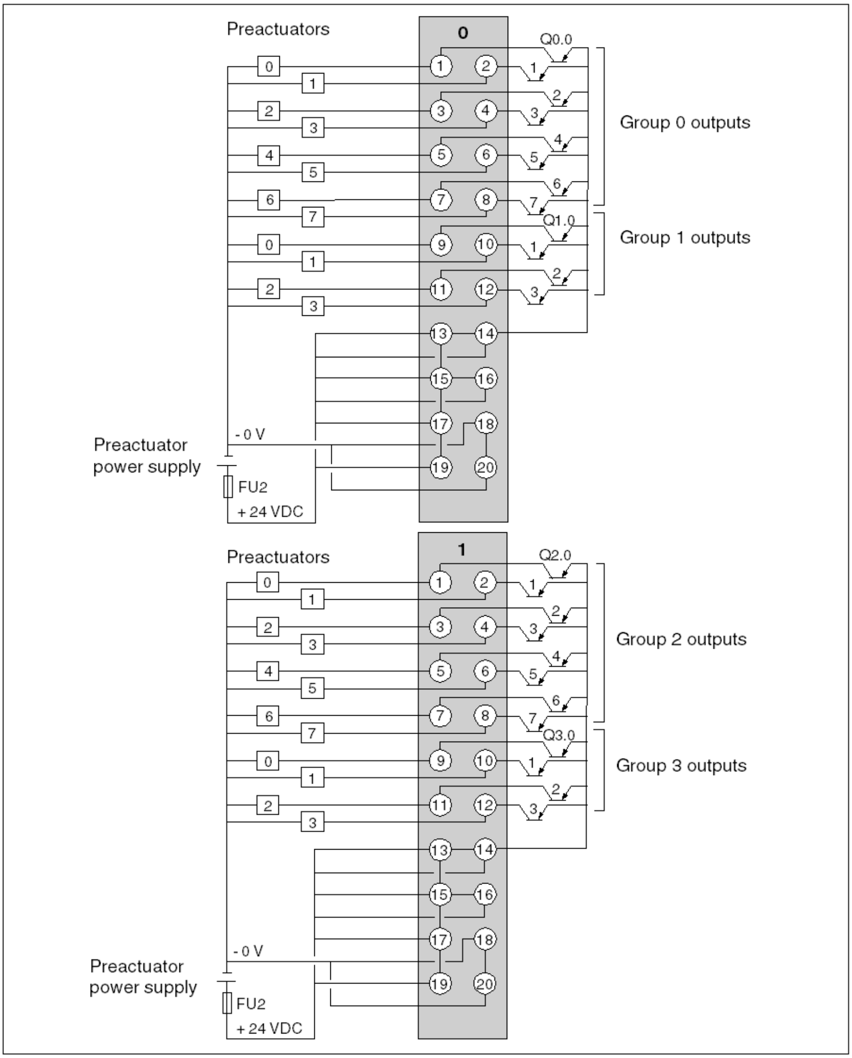
Positioning the HE10 connector, and identifying the signals

The diagram below represents the position of the HE10 connectors on the module in relation to the track outputs and the identification of the different signals delivered by the connectors.



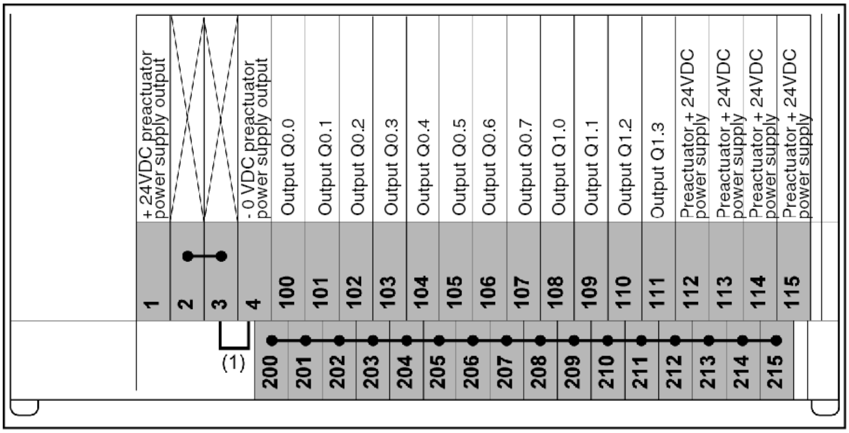
Process diagram

The diagrams below show the connection process.

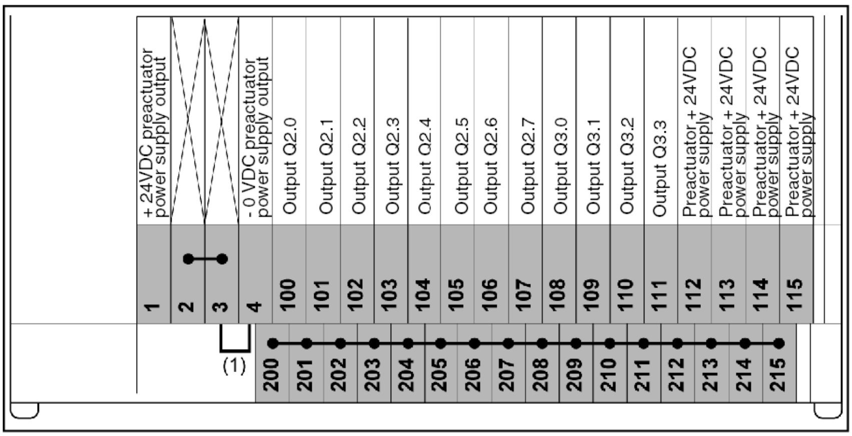


Connection using TELEFAST connector and TSX CDP ••3 cable

Availability of **connector 0** signals at the TELEFAST screw terminal block



Availability of **connector 1** signals at the TELEFAST screw terminal block

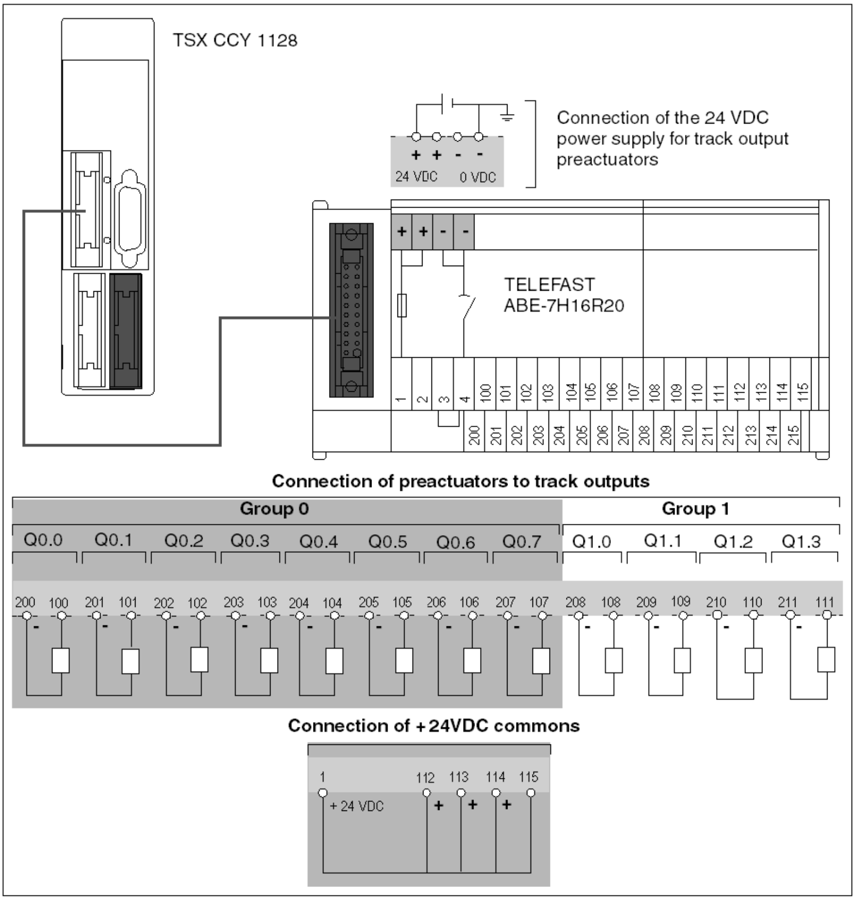


(1) At the TELEFAST ABE-7H16R20 connector, the position of the jumper wire determines the polarity of all terminals from 200 to 215:

- jumper wire in position 1 or 2: terminals 200 to 215 have positive polarity,
- jumper wire in position 3 or 4: terminals 200 to 215 have negative polarity.

Connection using TELEFAST connector and TSX CDP ••3 cable

Example for connecting pre-actuators to the track outputs of connector 0 (groups 0 and 1). Proceed in the same way for connector 1 (output groups 2 and 3).



The table below gives the correspondence between the TELEFAST screw terminal block and the HE10 connectors of the module (connectors 0 and 1).

No. of terminal on TELEFAST screw terminal block	Pin no. of HE10 connector	Type of signals at connectors		Functions at connectors	
		0	1	0	1
100	1	Q0.0	Q2.0	Group 0 track outputs	Group 2 track outputs
101	2	Q0.1	Q2.1		
102	3	Q0.2	Q2.2		
103	4	Q0.3	Q2.3		
104	5	Q0.4	Q2.4		
105	6	Q0.5	Q2.5		
106	7	Q0.6	Q2.6		
107	8	Q0.7	Q2.7		
108	9	Q1.0	Q3.0	Group 1 track outputs	Group 3 track outputs
109	10	Q1.1	Q3.1		
110	11	Q1.2	Q3.2		
111	12	Q1.3	Q3.3		
112	13	+ 24VDC		Pre-actuators receive a shared + 24VDC supply if the connection at terminal 1 of the TELEFAST is external	
113	14				
114	15				
115	16				
+ 24VDC	17-19	+ 24VDC		Track outputs' pre-actuator supply	
- 0 VDC	18-20	- 0 VDC			
1	-	+ 24VDC		Terminals 200 to 215 at + 24VDC if terminals 1 and 2 are connected	
2	-	Shared by terminals 200 to 215			
3	-	Shared by terminals 200 to 215		Terminals 200 to 215 at – 0 VDC if terminals 3 and 4 are connected	
4	-	- 0 VDC			

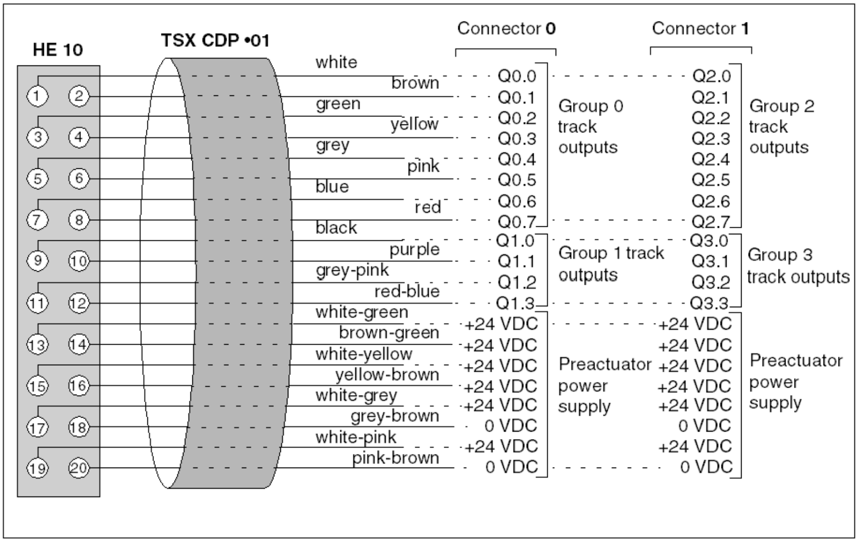
<table>
<tr><th rowspan="2">No. of terminal on TELEFAST screw terminal block</th><th rowspan="2">Pin no. of HE10 connector</th><th colspan="2">Type of signals at connectors</th><th colspan="2">Functions at connectors</th></tr>
<tr><th>0</th><th>1</th><th>0</th><th>1</th></tr>
<tr><td>200...215</td><td>-</td><td colspan="2">• + 24 VDC if terminals 1 and 2 are connected, - 0VDC if terminals 3 and 4 are connected
• + 24 VDC if terminals 1 and 2 are connected, - 0VDC if terminals 3 and 4 are connected</td><td colspan="2">Connecting shared sensors</td></tr>
</table>

Connection using TSX CDP •01 strips

This type of connection is used to connect all signals travelling to or from the module directly:

- at a terminal block, or
- at the pre-actuators.

The diagram below gives the correspondence between wire color and HE10 connector pin number for connectors 0 and 1



Connecting an absolute encoder with parallel outputs to TSX CCY 1128

9

Subject of this chapter

This Chapter describes the procedure for connecting an absolute encoder with parallel outputs to the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Principle for connecting an absolute encoder to a TSX CCY 1128	134
TELEFAST ABE-7CPA11 base	135
Pinout configuration of the 15-pin SUB-D connectors of the module and the TELEFAST	138
Connecting an absolute encoder with parallel outputs	140
Wiring rules and precautions specific to the TELEFAST	143
Configuration of the TELEFAST connector	147

Principle for connecting an absolute encoder to a TSX CCY 1128

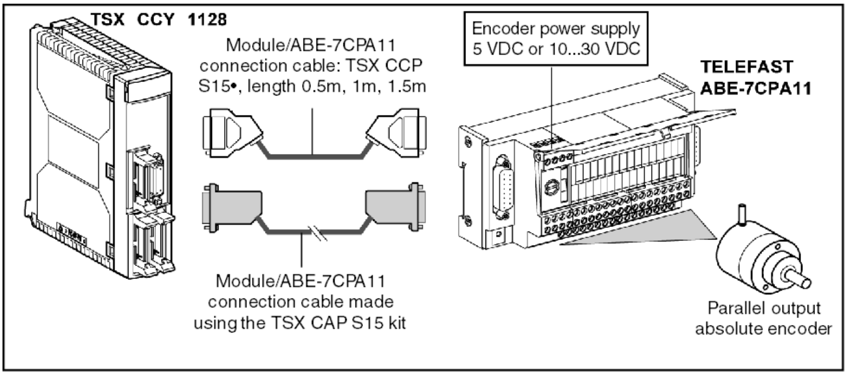
Connection string

The 15-pin SUB D connector, located on the front panel of the module, allows the connection of encoder and module using a TELEFAST ABE-7CPA11 connector.

- The TELEFAST connector receives:
  - all parallel signals from the encoder,
  - the 5VDC or 10…30VDC encoder supply source.
- The TELEFAST connector restores
  - signals for the module that have been encoded as RS 422 standard signals.

Illustration

The diagram below shows the principle for connecting an absolute encoder with parallel outputs to a TSX CCY 1128 module.



TELEFAST ABE-7CPA11 base

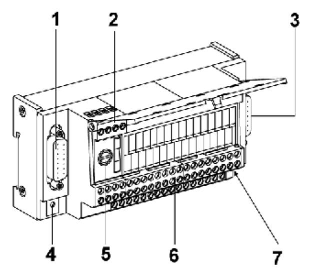
Role

The TELEFAST ABE-7CPA11 base ensures:

- the connection interface between:
  - the absolute encoder with parallel outputs, and
  - the TSX CCY 1128 module
- the conversion of the value position provided by the encoder with parallel outputs as RS 422 standard serial information. The absolute encoder must have pure or gray binary coding, with a maximum of 24 bits of data.

Appearance of the TELEFAST base

The diagram below represents the TELEFAST ABE-7CPA11 base and its various elements.



Elements and their functions

The table below gives the functions of the various elements of the TELEFAST.

Number	Elements	Functions
1	15 pin SUB-D connector.	Allows connection to the TSX CCY 1128 module, via a TSX CCP S15• cable.
2	Screw terminal block.	Allows connection of the encoder supply.
3	15 pin SUB-D connector.	Not used.
4	LED	When lit, shows that the TELEFAST is switched on.
5	Fuse	Ensures protection of the supply: ● calibre: 1A, ● type: non-delay fusion.
6	Screw terminal block	Allows connection to the encoder.
7	Microswitches	Allow configuration of the type of encoder relevant to the TELEFAST (gray or binary).

Characteristics of the TELEFAST base

General characteristics

The table below gives the general characteristics of the TELEFAST ABE-7CPA11 base.

Parameters	Values
Permitted voltage at 10…30 VDC	11...30 VDC
Permitted voltage at 5 VDC	5...6 VDC
Maximum frequency for changing the state of the lightweight bit	75 kHz
Read frequency of the serial frame	150 kHz...1 MHz
Current used (excluding encoder)	Typical: 90mA - Maximum: 130 mA
Dissipated power	Typical: 450mW - Maximum: 1,5 W
Limit of encoder supply feedback monitor	- 15% < V supply < + 15%
Insulation resistance	> 10MW at 500 VDC
Dielectric rigidity	1000 V eff -50/60 Hz per mn
Operating temperature	0…60° C
Storage temperature	-25° C…+70° C
Hygrometry	5%…95% without condensation
Operating altitude	0…2000 meters

Characteristics of the encoder reading inputs In0 to In23

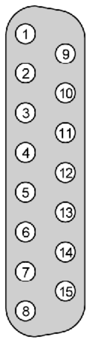
The table below gives the different parameters of the TELEFAST inputs connected to the encoder channels.

Parameters	Type	Values
Logical input	Positive	State 0: U < 2.5 V, state 1: U > 3.9V
	Negative	State 0: U > 3.9V, state 1: U < 2.5 V
Compatibility with encoder outputs	Totem Pole outputs	11...30 VDC
	TTL 5 V outputs	5 VDC
	Transistor outputs with NPN open collector	11...30 VDC
Maximum input voltage	-	+30 VDC
Maximum length of encoder/TELEFAST cable	-	200 m This maximum distance depends on the type of encoder used, and limits the frequency of changing the lightweight bit. See TELEFAST base configuration
Maximum length of encoder/TELEFAST cable	-	200 m. This maximum distance limits the frequency of the serial transmission clock. See the specific precautions and rules for cabling
Low limit for input voltage	-	0 VDC < VIL < 2.5 VDC
Upper limit for input voltage	-	3.9 VDC < VIH < 30 VDC

Pinout configuration of the 15-pin SUB-D connectors of the module and the TELEFAST

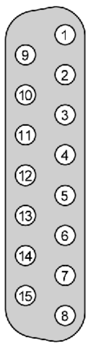
Pinout configuration of the module's 15-pin SUB-D connector

The pinout configuration for connecting an absolute encoder with parallel outputs using a TELEFAST ABE-7CPA11 is as follows.

Diagram (front view)	Pin No.	Signal	Designation
	1	Data+	Encoder input, + data (5 VDC)
	2	Data-	Encoder input, - data
	3	-	-
	4	-	-
	5	-	-
	6	CLK +	Encoder input, CLK + (5 VDC)
	7	10…30 V	Encoder supply input (+ 10…30 VDC)
	8	0 V	Encoder supply input (- 0 VDC)
	9	-	-
	10	-	-
	11	-	-
	12	-	-
	13	EPSR	+ encoder supply feedback input. Receives + supply feedback from the encoder, which allows the module to verify the encoder's presence.
	14	CLK -	Encoder input, CLK -
	15	5 V	Encoder supply input (+ 5 VDC)

Pinout configuration of the TELEFAST 15-pin SUB-D connector

The pinout configuration of the TELEFAST ABE-7CPA11 15-pin SUB-D connector is as follows.

Diagram (front view)	**Pin No.**	**Signal**	**Designation**
	1	Data+	Encoder output, + data signal (5 VDC)
	2	Data-	Encoder output, - data signal
	3	-	-
	4	-	-
	5	-	-
	6	CLK +	Encoder output, + CLK signal (5 VDC)
	7	10…30 V	Encoder supply output (+ 10…30 VDC)
	8	0 V	Encoder supply output (- 0 VDC)
	9	-	-
	10	-	-
	11	-	-
	12	-	-
	13	EPSR	+ encoder supply feedback input. Receives + supply feedback from the encoder, which allows the module to verify the encoder's presence.
	14	CLK -	Encoder output, - CLK signal
	15	5 V	Encoder supply output (+ 5 VDC)

Connecting an absolute encoder with parallel outputs

Introduction

An absolute encoder with parallel outputs must always be connected to the module using a TELEFAST ABE-7CPA11 base. The signals sent to the module are RS 422 standard SSI serial signals. The encoder supply is 10…30 VDC or 5 VDC depending on the encoder type.

Diagram showing the principle of connecting an absolute encoder with parallel outputs on a 10…30 VDC supply

The diagram below represents the connection string of an absolute encoder with parallel outputs on a 10…30 VDC supply.

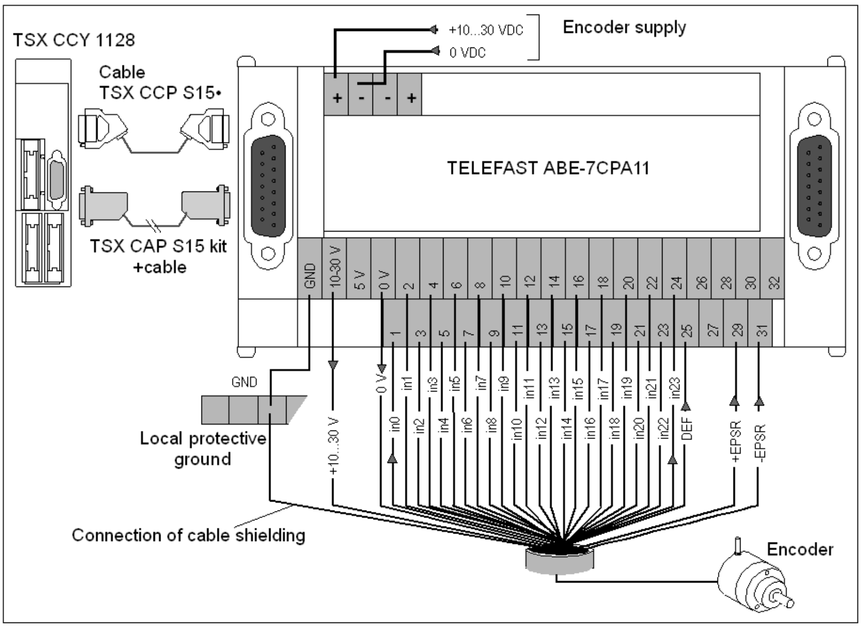
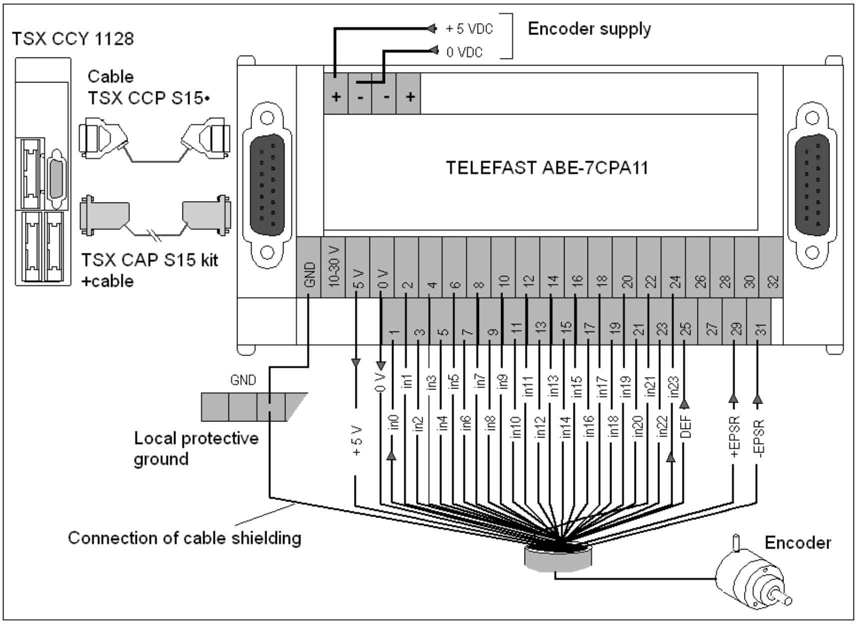


Diagram showing the principle of connecting an absolute encoder with parallel outputs on a 5 VDC supply

The diagram below represents the connection string of an absolute encoder with parallel outputs on a 5 VDC supply.



Functions of the TELEFAST screw terminal block

The TELEFAST base has two terminal blocks:

One located on the upper part, comprising four terminals and used to connect the supply source to the encoder,

Terminals on the screw terminal blocks	Signals	Functions
+	-	+ inputs, source of encoder supply
-	-	- inputs, source of encoder supply

Two terminal blocks comprising 36 terminals are located on the lower part. These are used to connect all signals travelling to or from the encoder.

Terminals on the screw terminal blocks	Signals	Functions
GND	-	Connection to TELEFAST protective ground. Ensures the continuity of the ground connection between the encoder and the module
+10..30 V	-	Connecting the + encoder supply for an encoder using a 10…30 VDC supply
+5 V	-	Connecting the + encoder supply for an encoder using a 5 VDC supply
0 V	-	Connecting the - encoder supply
1 to 24	In0 to In24	Parallel outputs of the encoder
25	ERR	Default output of the encoder
29	+ EPSR	+ return supply of the encoder. If there is no return encoder supply, connect the terminal to the +10…30 V or +5 V terminal according to the encoder supply
30	- EPSR	- return supply of the encoder. - return supply of the encoder. If there is no return encoder supply, connect the terminal to the 0 V terminal

Wiring rules and precautions specific to the TELEFAST

Connecting or disconnecting the TELEFAST

You should always connect or disconnect the TELEFAST's connectors and various connection wires when the voltage is SWITCHED OFF:

- connecting or disconnecting the cable connectors linking the module and the TELEFAST connector,
- connecting or disconnecting the wires linking the TELEFAST connector to the encoder.

Length of the connection cable between the module and the TELEFAST

The table below gives the clock frequency of the transmission series according to the distance.

If	then
cable length < to 10 meters	frequency of the transmission series clock: 1 MHz
cable length < to 20 meters	frequency of the transmission series clock: 750 kHz
cable length < to 50 meters	frequency of the transmission series clock: 500 kHz
cable length < to 100 meters	frequency of the transmission series clock: 375 kHz
cable length < to 150 meters	frequency of the transmission series clock: 200 kHz
cable length < to 200 meters	frequency of the transmission series clock: 150 kHz

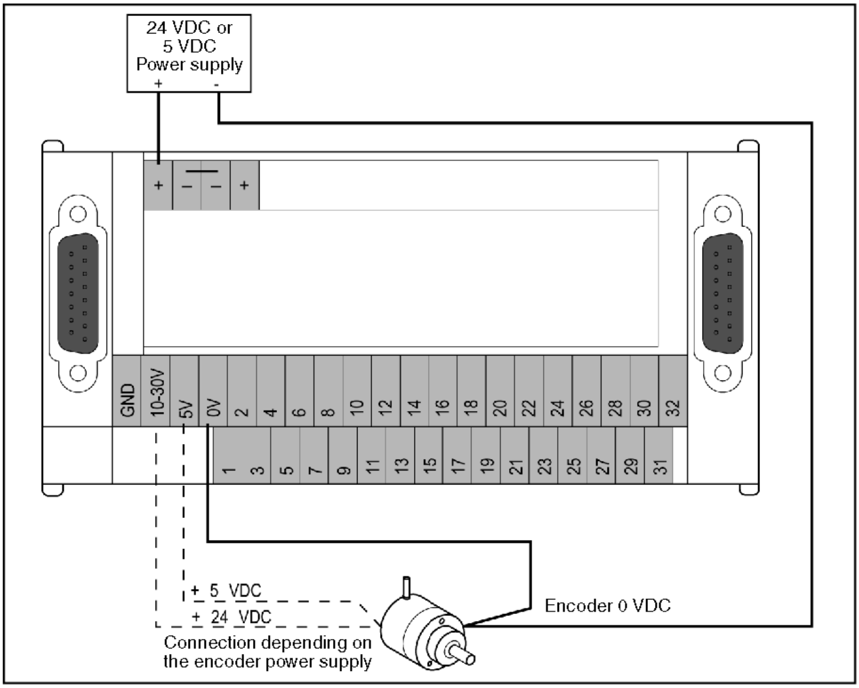
Cross-section of the wire connecting the module and the TELEFAST

In order to reduce the on-line voltage falls as much as possible, please respect the following points:

If	And	Then
The encoder is using a 5 VDC supply	The distance from the module to the TELEFAST is < 100m	Use a wire with minimum cross-section 0.08 mm (gage 28)
	The distance from the module to the TELEFAST is > 100m	Use a wire with minimum cross-section 0.34 mm (gage 22)

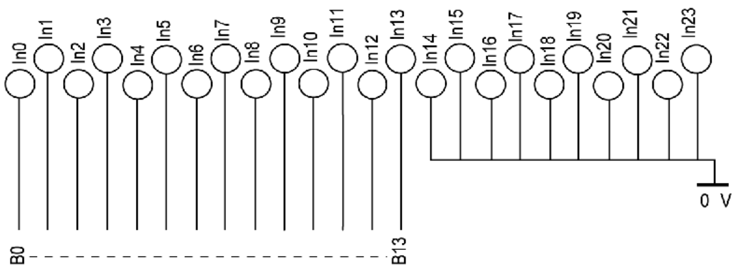
Connecting the encoder supply

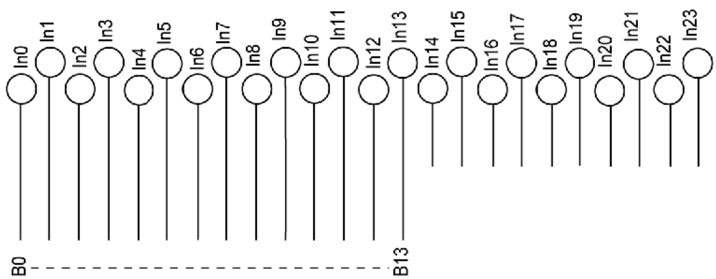
In order to limit voltage falls with a 0V, caused by the encoder supply current, we recommend that you wire the 0V as follows:



Wiring the encoder outputs on the TELEFAST

If the encoder outputs have positive or negative logic with a number lower than 24, use the following connection procedure:

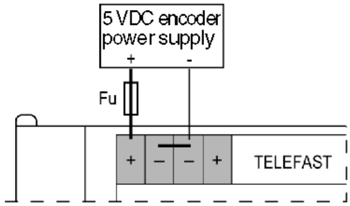
<table>
<tr><th>If</th><th>And</th><th>Then</th></tr>
<tr><td>the encoder outputs have positive logic</td><td>their number is lower than 24</td><td>• wire the encoder outputs to the TELEFAST inputs, working from the least significant to the most significant
• wire the unused TELEFAST inputs to the 0V terminal</td></tr>
<tr><td colspan="3">

Example: 14-bit encoder</td></tr>
<tr><td>the encoder outputs have negative Logic</td><td>their number is lower than 24</td><td>• wire the encoder outputs to the TELEFAST inputs, working from the least significant to the most significant
• do not wire (leave free) the unused TELEFAST inputs.</td></tr>
<tr><td colspan="3">

Example: 14-bit encoder</td></tr>
</table>

Protecting the encoder supply

According to the encoder supply voltage, the supply should be protected as follows:

If	Then
The encoder supply voltage is 10…30 VDC	The protective fuse is built into the TELEFAST: • size: 1A • type: fast-blow fusion.
The encoder supply voltage is 5 VDC	Provide a series fuse (Fu) for the positive supply: • calibre: to be determined by the user, dependent upon the TELEFAST and encoder consumption • type: fast-blow fusion 

Monitoring the encoder supply

If the encoder supply voltage decreases by more than 15%, the default (EPSR signal) is sent back to the module. If the encoder does not have a return supply, do the following:

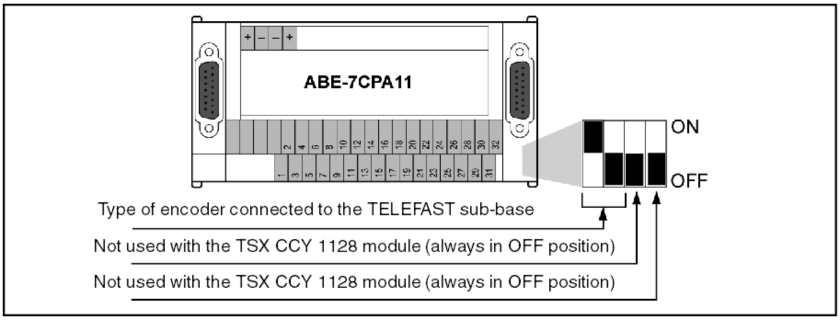
If	Then
No return encoder supply	Connect the positive and negative EPSR of the TELEFAST: • the positive EPSR terminal of the TELEFAST to the positive terminal of the encoder supply • the negative EPSR terminal of the TELEFAST to the negative terminal of the encoder supply

Configuration of the TELEFAST connector

At a Glance

The connector is configured by setting the 4 micro-switches located under the SUB D 15-pin connector to the right of this. For the TSX CCY 1128 module, the configuration is limited according to the type of connected encoder.

The diagram below shows the location and function of the 4 micro-switches.



The two micro-switches located on the right must always be in the OFF position. The two micro-switches on the left allow the link performance to be set according to the encoder output characteristics, and dependent upon the distance between the TELEFAST and the encoder.

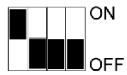
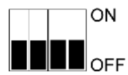
Positions of the micro-switches: encoder with positive Logic outputs

Encoder with positive Logic outputs, coded GRAY

Type of encoder output			**position of the micro-switch**	**Maximum length encoder/TELEFAST**	**Max. frequency for changing least significant bit**
Logic	**Output interface**	**Code**			
Positive	• Totem Pole • TTL • NPN open collector	Gray		50 meters	75 kHz

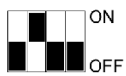
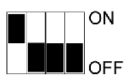
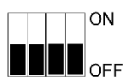
Positions of the micro-switches: encoder with negative Logic outputs

Encoder with negative Logic outputs, coded GRAY

Type of encoder output			position of the micro-switch	Maximum length encoder/TELEFAST	Max. frequency for changing least significant bit
Logic	**Output interface**	**Code**			
Negative	Totem Pole	Gray		50 meters	75 kHz
	TTL			100 meters	40 kHz
	NPN open collector			200 meters	5 kHz

Positions of the micro-switches: encoder with positive or negative Logic outputs,

Encoder with positive or negative logic outputs, Binary code

Type of encoder output			Positions of the micro-switches	Maximum length encoder/TELEFAST	Max. frequency for changing least significant bit
Logic	**Output interface**	**Code**			
Positive or negative	Totem Pole	Binary		10 meters	40 kHz
	TTL			30 meters	20 kHz
	NPN open collector			50 meters	5 kHz

TSX CCY 1128 module displays

10

Subject of this chapter

This Chapter introduces the various LEDs of the TSX CCY 1128 electronic cam module and their meanings.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
10.1	Electrical properties of the TSX CCY 1128 module	150
10.2	Displays of the TSX CCY 1128 module	156

10.1 Electrical properties of the TSX CCY 1128 module

Object

This section describes the various electrical properties of the TSX CCY 1128 cam module.

What Is in This Section?

This section contains the following topics:

Topic	Page
General electrical characteristics of the TSX CCY 1128	151
Characteristics of the auxiliary inputs of the TSX CCY 1128	152
Characteristics of the encoder supply feedback monitor for the TSX CCY 1128	153
Characteristics of the auxiliary inputs of the TSX CCY 1128	154
Characteristics of the track outputs of the TSX CCY 1128	155

General electrical characteristics of the TSX CCY 1128

Table of the general characteristics of the module

The following table gives the general characteristics of the module.

Parameter designation		**Values**	
		Typical	**Maximum**
Current used by the module	internal 5 V (with internal ventilator in the operating module)	0.66 A	1 A
	24V sensors/pre-actuators (auxiliary inputs and track outputs)	15 mA	18 mA
	10…30 V (using an absolute SSI encoder, and 24V supply only)	11 mA	20 mA
Power dissipated in the module		7 W (1)	10 W (2)
Sensor/pre-actuator supply monitoring		Yes	
Insulation resistance		> 10 MOhms at 500 VDC	
Dielectric rigidity with the ground connection or the 0 V PLC logic		1000 V eff. - 50/60Hz per min	
Operating temperature		0 to 60° C	
Storage temperature		-25° C to 70° C	
Hygrometry (without condensation)		5% to 95%	
Operating altitude		0 to 2000m	

(1) Under normal operating conditions: only one active auxiliary input, 24 VDC supply voltage, RS 422 standard signal.

(2) Under extreme operating conditions: all auxiliary inputs active, 30 VDC supply voltage, etc

Characteristics of the auxiliary inputs of the TSX CCY 1128

Characteristics of the encoder inputs

The table below gives the characteristics of the A, B and Z encoder inputs.

Input			**RS 422 use**	**Use at 10… 30 VDC**
Logic			Differential inputs	Positive or negative
Nominal values	Voltage		-	24 V
	Current		10 mA	15.5 mA
Thresholds	Voltage		< 5.5 V	< 30 V (possible up to 34V, limited to 1hr in 24)
	In state 1	Voltage	> 3 V (1)	> 11 V
		Current	> 5.8 mA (1)	> 5 mA
	In state 0	Voltage	< - 3 V	< 5 V
		Current	< - 5.8 mA	< 2 mA
Input impedance at nominal voltage			-	1.5 kOhms
Type of input			Resistant	Resistant
Maximum permitted frequency	Incremental encoders		500 kHz multiplied by 1 250 kHz multiplied by 4	

(1) The positive or negative differential voltage must be higher then 3 volts, and the current in the positive or negative loop must be higher than 5.8 volts to guarantee:

- the inclusion of count pulses up to 500 kHz,
- that the line control does not detect errors, irrespective of the frequency.

NOTE: Comparison of the RS 422 standard encoder outputs

An encoder with RS 422 standard outputs can control the inputs of two TSX CCY 1128 modules in parallel. In order to guarantee the necessary voltage levels, the encoder supply voltage must be higher than 4.5 V.

Characteristics of the encoder supply feedback monitor for the TSX CCY 1128

Characteristics of the EPSR input

The table below gives the characteristics of the encoder supply feedback monitor.

Parameters		Values
Thresholds for the EPSR input	Voltage	30 V (possible up to 34V, limited to 1h. in 24h)
	Current	< 1.5 mA
Voltage for OK state	Vref input free	OK if U> 3.3V
	VRef connected to + encoder supply	OK if U > 66% of the voltage recorded at the VRef input

Characteristics of the auxiliary inputs of the TSX CCY 1128

Table of the auxiliary inputs characteristics

The following table gives the characteristics of the IREC, ICAPT1 and ICAPT2 auxiliary inputs.

Parameter designation			Symbols	Values	Units
Nominal values	Voltage		Un	24	V
	Current		In	8	mA
	Sensor supply, ripple included		U1 Utime (1)	19...30 34	V
Thresholds	In state 1	Voltage	Uon	> 11	V
		Current at Uon	Ion	> 3	mA
	In state 0	Voltage	Uoff	< 5	V
		Current	Ioff	< 1,5	mA
Response time	State 0 to 1		Ton	< 100	ms
	State 1 to 0		Toff	< 100	ms
Sensor voltage check threshold	OK		Uok	> 18	V
	Fault		Udef	< 14	V
Input impedance			Re	3	kOhms
Type of input	Resistive				
Logic type	Positive (sink)				
IEC 1131 compatibility with sensors	Type 1				
3-wire/2-wire proximity sensor compatibility	• 3-wire proximity sensor: all 3-wire proximity sensors operating at 24 VDC • 2-wire proximity sensor: all 2-wire proximity sensors operating at 24 VDC have the following characteristics: Breakdown voltage when closed: < 7V Minimum switched current: < 2.5 mA Residual current when open: < 1.5 mA				
Dielectric rigidity with the ground connection	1500 V eff 50/60 Hz per mn				

(1) Utime: maximum permitted voltage for 1 hour in every 24.

Characteristics of the track outputs of the TSX CCY 1128

Table of characteristics of the track outputs

The following table gives the characteristics of the track outputs.

Parameter designation			Symbols	Values	Units
Nominal values	Voltage		Un	24	V
	Current		In	500	mA
Thresholds	Voltage		U1	19...30	V
			Utime (1)	34	V
	Maximum current per output for U= 30 V or 34 V		I1	600	mA
	Maximum current	per connector	I2	< 6	A
		per module	I3	< 12	A
Maximum power for lamp with tungsten filament			P1	10	W
Maximum switching frequency on inductive load			F	<0.6/LI	Hz
Electro unload time			T	< L/R	s
Pre-actuator voltage check threshold		OK	Uok	> 18	V
		Fault	Udef	< 14	V
Compatibility with direct current inputs			All direct current inputs with positive logic, and an input resistance of < at 15 kOhms		
Protection against overloads and short-circuits			Using current limiter and thermal circuit breaker (0.7A<Id<2A)		
Protection against overvoltage of the outputs			Using a Zener (breakdown) diode between the outputs and the +24V		
Protection against polarity inversions			Using a reverse diode on the supply		
Dielectric rigidity with the ground connection			1500 V eff 50/60 Hz per mn		
Compliance with IEC 1131-2			Yes		

10.2 Displays of the TSX CCY 1128 module

Object

This section describes the various LED indicators of the electronic cam module TSX CCY 1128 and their meanings.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction to the display block of the TSX CCY 1128 module	157
The various states of the LEDs on the TSX CCY 1128 and their meaning	158

Introduction to the display block of the TSX CCY 1128 module

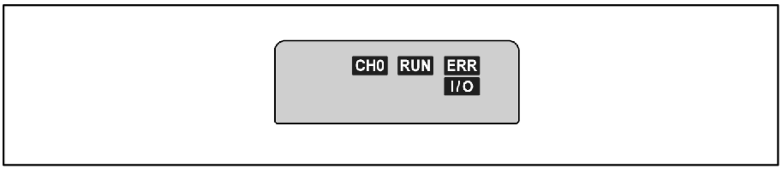
Role

The module's display block has four LEDs, which provide information on:

- the operating mode of the module, normal operating mode or module switched off or experiencing a fault
- internal or external module operating faults.

Physical presentation

The diagram below is a representation of the physical appearance of the module's display block and the location of its four LEDs



The various states of the LEDs on the TSX CCY 1128 and their meaning

Signaling operating mode

The table below gives the various states of the RUN lamp and their meanings

LED	Color	State	Meaning
RUN	Green	Lit	Module operating normally
		Off	Module switched off or experiencing a fault

Signaling faults

The table below gives the various states of the ERR, I/O and CH0 LEDs and their meanings.

LED	Color	State	Meaning
ERR	Red	Lit	Internal module error: ● module has broken down
		Flashing	● Error in communication with processor ● Application missing, invalid or experiencing a fault during execution
		Off	Operating normally, no faults
I/O	Red	Lit	External module fault: ● Cabling fault ● Encoder supply fault ● Configuration/adjustment parameters declined
		Flashing	Insignificant
		Off	Operating normally, no faults
CH0	Green	Lit	Operating normally, channel is active
		Flashing	The channel is not functioning correctly due to: ● an external fault ● a communication error
		Off	Channel inoperative: ● Channel not configured ● Channel incorrectly configured

Software installation cam TSX CCY 1128

III

Aim of this part

This part describes the software installation of the cam module TSX CCY 1128.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
11	Configuration of the electronic cam module	161
12	Programming	177
13	Entering recipe adjustment parameters for the electronic cam module	219
14	Debugging and adjusting	241
15	Diagnostics	255
16	Performance and limitations	271
17	The language objects of the independent axes application	283

Configuration of the electronic cam module

11

Subject of this chapter

This chapter describes the configuration operations for the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Accessing the module configuration parameters	162
Configuring the electronic cam parameters	163
Configuring acquisition parameters	164
Configuring an incremental encoder	165
Configuration of an absolute encoder	167
Reading format configuration	169
Configuring the position recalibration function for incremental encoders	170
Capture function configuration	171
Cam processor configuration	172
Connector configuration	174
Confirmation of the configuration	175

Accessing the module configuration parameters

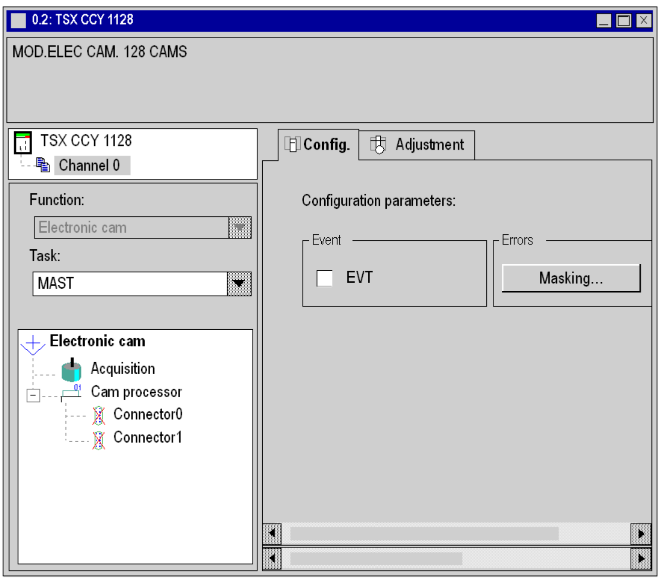
Role

This operation is used to access the configuration parameters of the TSXCCY1128 module.

In connected mode, the parameters: task, event, masking faults, encoder and encoder configuration cannot be modified. Performing any modification causes the cam function to stop Cam (processor on Stop).

Procedure

The table below describes the procedure:

Step	Action
1	Access the hardware configuration screen
2	Double-click on the position of the module in the rack. **Result :** 
3	Select the task (MAST or FAST) in which the module language objects are updated in the PLC processor: **Task** scrollable menu.
4	In the Browser, double-click on the element to be configured. ● Electronic cam ● Acquisition ● Cam processor ● Connector 0 or 1 after unlocking

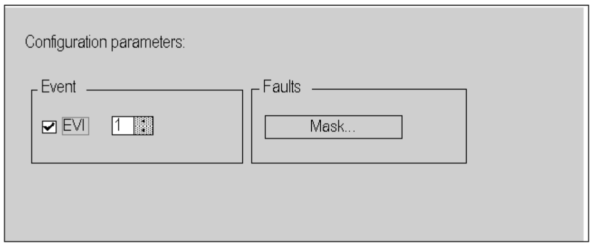
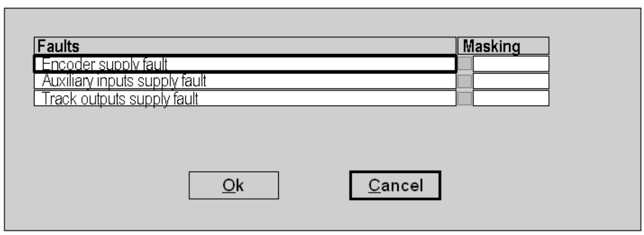
Configuring the electronic cam parameters

Role

This operation is used to:
- declare an event-triggered task associated with the module
- mask all or some of the application faults

Procedure

The following table describes the operations to be performed:

Step	Action
1	In the browser, double-click on **electronic cam**. **Result :** 
2	If an event-triggered task in the PLC processor is to be assigned to the module: ● click in the **EVT** check box ● select the number of the associated event-triggered task, from 0 to 63 (0 being the priority task).
3	Masking a fault prevents indication of the fault being taken into account in the general channel fault information. The associated check remains active, however, and affects the operating modes of the module. To mask faults: click on the **Mask** button. **Result:**  Check the box(es) of the faults which are to be masked and confirm.

Configuring acquisition parameters

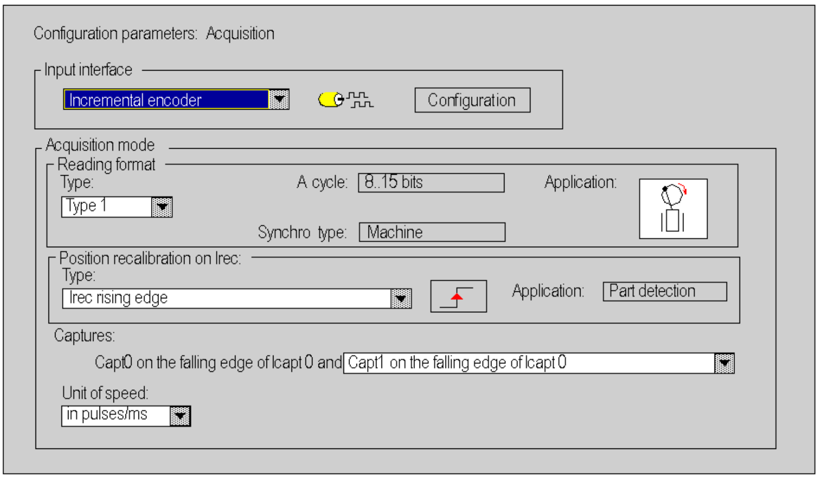
Role

Configuring acquisition parameters allows you to:

- select the type of encoder (incremental or absolute) and define its characteristics
- select the acquisition mode functions
  - Reading format
  - Recalibration
  - Capture
  - Unit of speed

Procedure

The following table describes the operations to be performed:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the browser, double-click on Acquisition.
Result:


</td></tr>
<tr><td>2</td><td>Select the field to be modified.</td></tr>
</table>

Configuring an incremental encoder

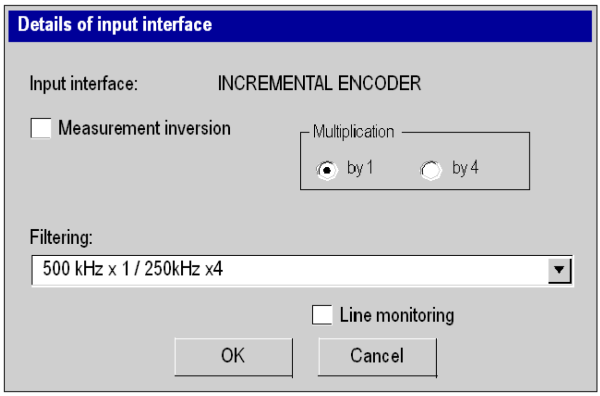
Role

Configuring an incremental encoder allows you to:

- declare that the inputs are connected to an incremental encoder,
- define the processing characteristics of the encoder signals.

Procedure

The table below describes the procedure:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select Incremental encoder in the input Interface field in the Acquisition configuration screen.</td></tr>
<tr><td>2</td><td>Click on Configuration to display the following dialogue box :

</td></tr>
<tr><td>3</td><td>Select the counter input filtering according to the maximum frequency supplied by the encoder :
● 125 kHz x1 / 125 kHz x4 enables good running for frequencies lower than 125 kHz in input (with or without multiplication by 4),
● 500 kHz x1 / 250 kHz x4 corresponds to the module’s limit values.</td></tr>
</table>

Step	Action
4	If necessary check the function **Reading inversion**. This parameter defines the direction in which the measurement changes in relation to the direction of rotation of the encoder.
5	Enable if necessary the function **Line check**. This function indicates any electrical faults (line break or short-circuit) on the links with the RS422 encoder. This function must not be used with an encoder with 10...30V totem pole outputs.
6	Select multiplication by 1 or by 4. Multiplication by 4 gives a resolution 4 times greater than the resolution of the encoder.
7	Click on **OK** to confirm the choices made.

Configuration of an absolute encoder

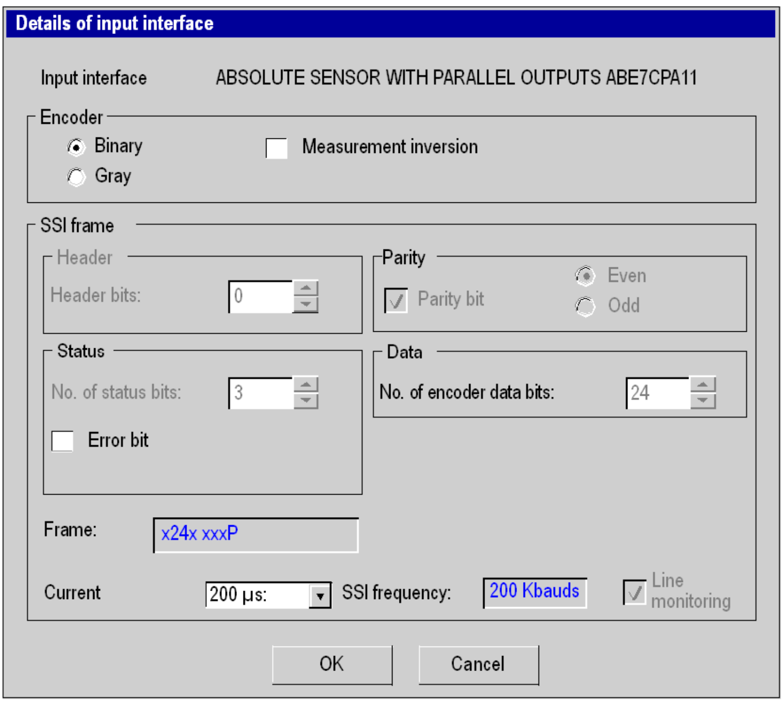
Role

Configuring an absolute encoder enables you to:

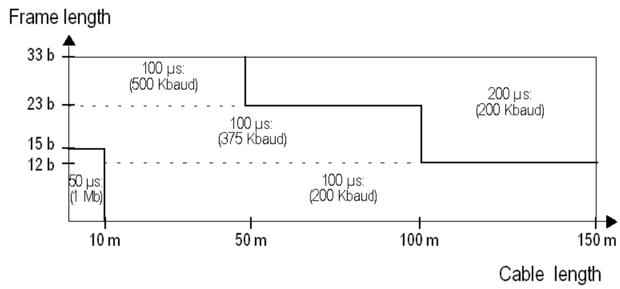
- declare that the inputs are connected to an absolute SSI encoder or to an absolute encoder with parallel outputs,
- define the characteristics of the encoder and the SSI frame.

Procedure

The table below describes the procedure:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select SSI absolute encoder or Parallel output absolute encoder in the input Interface field in the Acquisition configuration screen.</td></tr>
<tr><td>2</td><td>Click on Configuration... to display the following dialogue box:


</td></tr>
<tr><td>3</td><td>Select the type of coding used by the encoder: Binary or Gray.</td></tr>
</table>

Step	Action
4	If necessary check the function **Reading inversion**. This function modifies the value provided by the encoder in order to reverse the direction of the position.
5	Adjust the interrogation interval of the encoder according to the length of the frame and the length of the cable connecting the encoder. The module automatically determines the transmission frequency. The diagram below can be used to select the optimum period, the value in brackets gives the transmission frequency determined by the module. 
6	**For an SSI absolute encoder:** Set the SSI frame characteristics used by the encoder: ● number of bits on **the frame** heading are not significant N mini = 0, N maxi = 4 (0 is the default), ● encoder data **number** of bits: N min. = 8, N max. = 25, (8 is the default), ● number of bits of **status** N maxi = 3 (0 is the default), ● number of bits sent after the last data bit excluding the parity bit, ● presence or absence of the error bit (if the status field is not zero), ● setting of the error bit (Number from 1 to 3) in the status bit zone, ● logic level of the error bit (active at 0 or active at 1), ● presence of **parity** bit (by default it is absent) and type of parity: Even or Odd (odd parity is not checked by the module). Note: Number of header + data + status bits < or = 32 As the selections are made, the frame zone displays the elements of the frame. **Example**: Frame: xxxx x8x xxE P. xxxx = 4 header bits (one cross per header bit). x8x = 8 data bits. xxE = 3 status bits, one of which is an error bit positioned at number 1. P = presence of parity bit, parity type: Even. **For a parallel output absolute encoder:** Set the characteristics of the SSI frame used by the encoder bearing in mind that the following values are set as standard: ● number of frame **header** bits are not significant: 0 ● encoder data **number** of bits: 24 ● status **number of bits**: 3 ● number of error bit: 3 if error bit present ● presence of **parity** bit and type of parity Even
7	Click on **OK** to confirm the choices made.

Reading format configuration

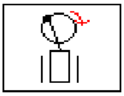
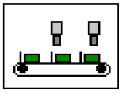
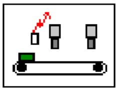
Role of the parameter

This parameter defines the format for the axis position reading, calculated by the module.

The format is chosen according to the type of movement.

Parameter selection

The following table can be used to select the format type (1, 2 or 3) depending on the type of application.

	Type 1	**Type 2**	**Type 3**
			
Type of movement	rotary and alternating	cyclic	continuous
Position reading	angular	angular + number of turns	counting the number of pulses
Format	8..15 bits (cycle) for incremental 8..14 bits for absolute	8..15 bits (cycle) for incremental 8..14 bits for absolute 1..15 bits (turn)	15 bits (cycle) ie. 32768 pulses
Synchro	machine	machine	parts detection
Type of encoder	incremental or absolute	incremental or absolute	incremental

The choice is made via a pull-down menu. The other fields are for information only.

Configuring the position recalibration function for incremental encoders

Role of recalibration

This is the module function which is used to calibrate the axis in relation to the machine zero or to synchronize the axis in relation to the entry of a part.

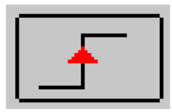
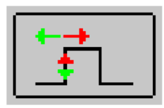
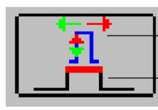
Recalibration forces the position reading to a value which is predefined by the "recalibration value" parameter (within the pulse field of the cycle).

This function is used to compensate for any possible slip in the reading. It is applied to the incremental encoders. Each time the moving part passes in front of the detector (wired to the Irec recalibration input), the reading is recalibrated.

The configuration operation of the recalibration function consists of defining the type of signal detected on the Irec recalibration input.

Procedure

The type of recalibration is selected using a pull-down menu:

For a synchro...	If you want recalibration on detection...	then select...
Part	• of a rising edge on the module **Irec** input	Irec rising edge 
Machine without turn pulse marker	• of a rising edge in forward movement • of a falling edge in reverse movement on the module **Irec** input.	Rising edge +direction, falling edge -direction 
Machine with turn pulse marker	• of a rising edge in forward movement • of a falling edge in reverse movement on the Zero Pulse Marker input when the **Irec** input is at 1.	short Cam  Zero Irec

Capture function configuration

Role of the capture function

This function is used to sample the value of the position of the axis on detection of an event. The capture function has no impact on the axis values, nor on the cam processor.

Implementing this function enables the application to manage the process better, for example control of: the number of pulses sent by the encoder, the size of the parts, the slip of the axis, the entry angle of the parts.

The module has:
- 2 physical capture inputs **Icapt0** and **Icapt1**,
- 4 register words that can be accessed via the sequential program:
  - 0 Registers: always store the actual position value of the angle (CAPT0_ANG) and the number of turns (CAPT0_TURN). The capture function always operates on detection of a rising input front Icapt0.
  - 1 Registers: the values captured in these registers (CAPT1_ANG and CAPT1_TURN) depend on the type of capture requested (see table below).

Procedure

The type of capture is chosen via a pull-down menu (only the first 2 choices can be accessed for an absolute encoder):

If you require capture on detection of a rising edge on the Icapt0 input (for 0 registers) and ...	then select...
a falling edge on the **Icapt0** input (for 1 registers) (E.g.: measure the size of the parts, in comparison with 0 registers)	Capt1 on falling edge of Icapt0.
a rising edge on the **Icapt1** input (for 1 registers) (E.g.: measure the entry angle of the parts)	Capt1 on rising edge of Icapt1.
storage of the number of pulses in the 1 registers on each encoder turn (E.g.: to check the encoder link)	Capt1 = Number of pulses per encoder turn.
capturing the value of the angle before recalibration in the 1 registers (E.g.: to evaluate the deviation due to the slip and corrected by the recalibration)	Capt1 before recalibration.
of a rising edge on zero pulse marker input (for 1 registers) (E.g.: to evaluate the deviation due to the slip, without recalibrating the axis)	Capt1 on rising edge of zero marker

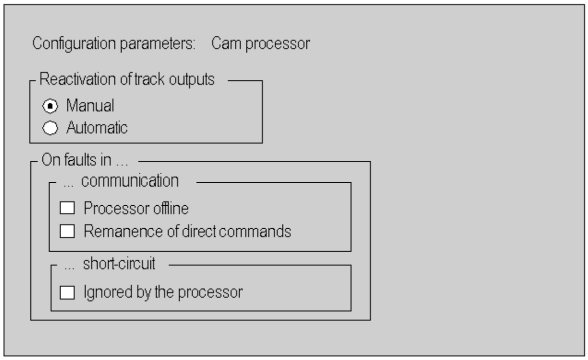
Cam processor configuration

Role

This operation defines the cam processor performance and the module outputs on a fault.

Procedure

The following table describes the operations to be performed:

Step	Action
1	In the browser, click on **cam processor**. **Result:** 
2	Select the type of output reactivation: **Manual** or **Automatic**. When an overcurrent is detected on an output, it is tripped, and it can be reactivated: ● **Manually**: it can be done from the debug screen or via the application. ● **Automatically**: performed automatically, 10 seconds after tripping.
3	Select the behavior of the cam processor module on a communication fault between the PLC processor and the electronic cam module. See table below.
4	Select the behavior of the cam processor on a short-circuit fault: ● box not selected or default selection: the cam processor is stopped if an output becomes faulty. ● **ignored by the processor** : tripping one of the track outputs does not stop the cam processor.

Behavior of the module on a communication fault

The behavior of the module on a communication fault between the PLC processor and the electronic cam module is summarized in the following table:

Check boxes selected	Cam processor state	State of the logic outputs
None	change to stop	set to 0
Processor offline	normal operation, (remains in RUN)	set to 0
Remanence of direct commands	change to stop	same as the last direct commands transmitted by the PLC processor;
Processor offline and Remanence of direct commands	normal operation, (remains in RUN)	or logic between: • the state of the associated tracks (or the reverse if inversion is requested in the adjustment parameter) • the last direct commands transmitted by the PLC processor

The actual state of a physical output is equal to the logic state of the output, or its reverse if inversion has been requested in the connector configuration.

Connector configuration

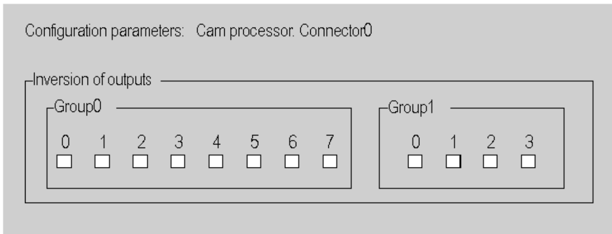
Role

The logic state of thetracks is assigned to the physical outputs of the module.

The configuration of the connectors enables the inverse state to be electrically applied to an output:
- 24 V for a 0 state,
- 0 V for a 1 state,

Procedure

The following table describes the operations to be performed:

Step	Action
1	Click with the right mouse button on **Connector 0 or 1** in the browser and select **Unlock**. **Result**: the cross above the connector icon disappears.
2	Double click on Connector 0 or 1 **in the browser**. **Result**: 
3	Check the boxes of the outputs to be inverted

Confirmation of the configuration

Procedure

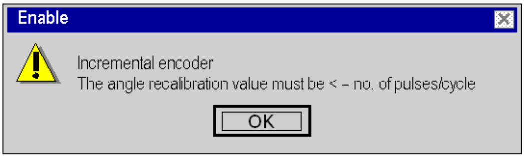
When all the configuration parameters are entered, enable the configuration obtained using the command **Edit/Enable** or click on the enabling icon.

If one or more parameter values is outside the permitted limits, an error message appears indicating the parameter concerned. Correct the parameter then confirm.

NOTE: Incorrect parameters are displayed in red.

Important

- Recipe adjustment parameters are initialized on the first request to confirm the configuration. It is therefore possible that, following modifications of the configuration values, the recipe adjustment parameters will no longer be correct. In this case a message specifies the parameter affected:



  Access the recipe adjustment parameters screen, correct the parameter, then confirm.
- **The configuration parameters are actually taken into account:**
  - when each of the configuration and adjustment parameters is correct,
  - when they are confirmed in the main screen of the configuration editor.

Programming

12

Subject of this chapter

This chapter describes the different programming steps and functions of an electronic cam module

What Is in This Chapter?

This chapter contains the following topics:

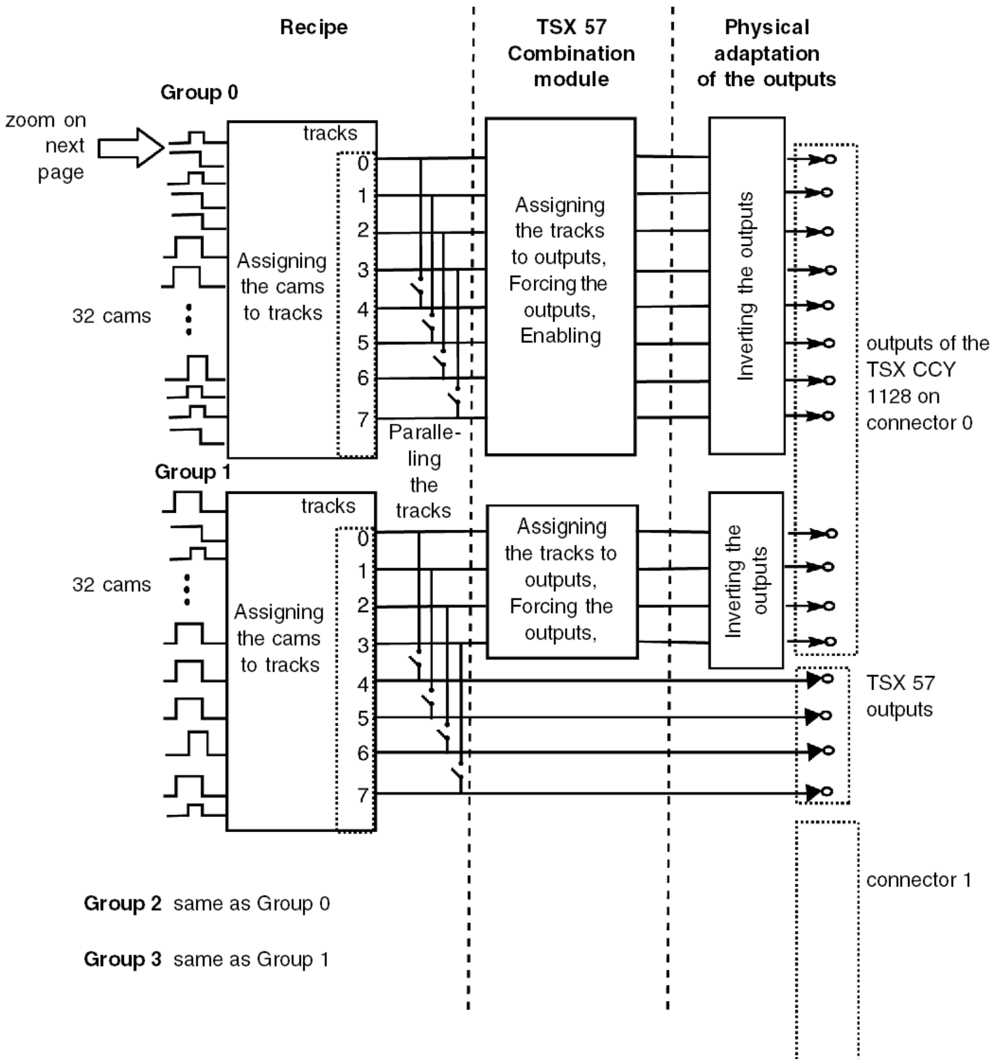
Topic	Page
Synopsis of the module functions	179
Confirmation of the axis functions	181
Confirming the cam processor functions	183
Confirming the events	184
Overview of event management	185
Language interface	186
Processor and module exchanges	187
System exchanges	188
WRITE_PARAM : Transfer of current recipe parameters	190
READ_PARAM : Transfer of current recipe parameters	192
RESTORE_PARAM : Transferring initial parameters	193
SAVE_PARAM : Transferring initial parameters	194
MOD_PARAM: Axis adjustment	195
MOD_TRACK: Adjusting a track	198
MOD_CAM: Adjusting a cam	201
TRF_RECIPE: Recipe transfer functions	204
TRF_RECIPE: Functions for storing recipes	205
TRF_RECIPE: Loading a new recipe	206
TRF_RECIPE: Saving a new	208
DETAIL_OBJECT : Operator dialogue interface	210

Topic	Page
DETAIL_OBJECT: Transferring cam details	211
DETAIL_OBJECT: Transferring track details	214
Reconfiguration in connected mode	217

Synopsis of the module functions

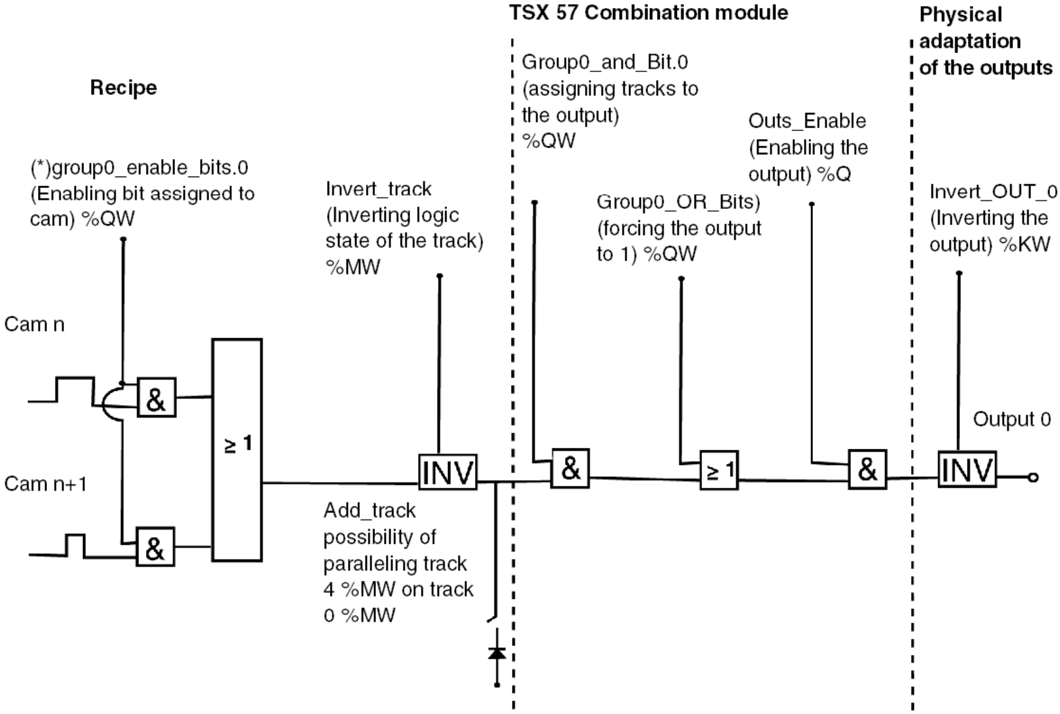
General overview

This synopsis describes a general overview of the functions of the module:



Detail for output 0 and 0 of group 0

This synopsis describes the zoom from the previous page:



NOTE: The cams assigned to one track cannot be assigned to another track.

(*) There are two other possibilities:
- always valid
- parts counter full

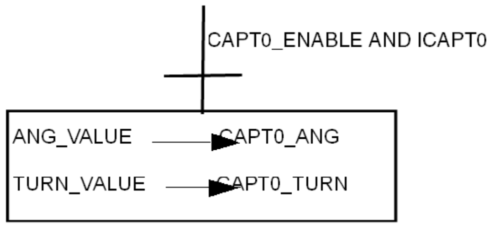
Confirmation of the axis functions

Capture

The module is able to capture the following:
- the current angle value,
- the current value of the number for the cycle, (for type 2)

AND FOR THE INCREMENTAL ENCODER ONLY
- the number of points per turn of the encoder,
- the current value before recalibration,
- and the current value on detection of the zero pulse marker

Example: Capture Angle and number of cycles in CAPT0 register



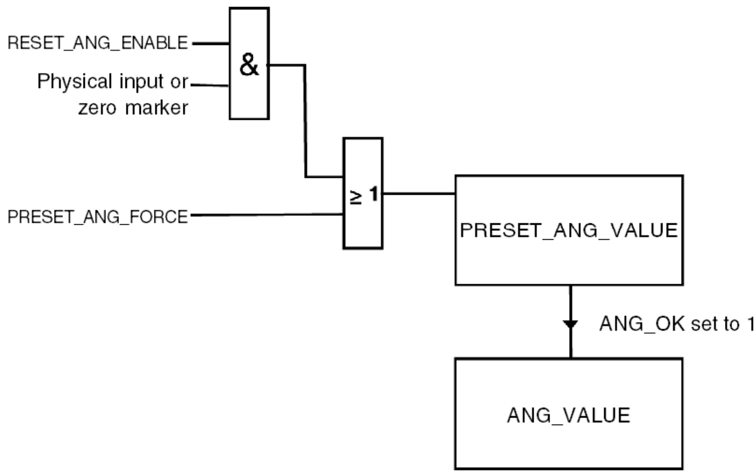
Recalibration

With an incremental encoder, the counter using the position value must have been recalibrated at least once for the processor to switch to RUN on a PCAM_START_STOP command.

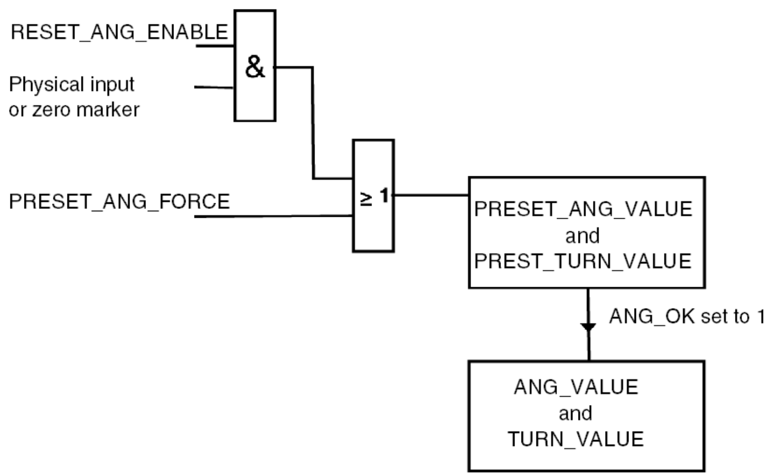
The implicit commands:
- PRESET_ANG_ENABLE or PRESET_ANG_FORCE (for Type 1 and Type 3)
- PRESET_ALL_ENABLE or PRESET_ALL_FORCE (for Type 2) must be positioned at 1 for the recalibration to be carried out.

Information ANG_OK changes to 1 when the counter is stalled.

Type 1 and Type 3



Type 2



Confirming the cam processor functions

RUN

The PCAM_START_STOP command object is used to set the cam processing to RUN.

This will be effective on the rising edge of the command bit:
- If the axis is recalibrated,
- if there is no external or application fault detected

If the cam processor is at STOP all the tracks are at 0.

NOTE: The forcing command and physical inversion of the outputs remain active.

Confirming the events

Event sources

The TSX CCY 1128 module includes 7 sources of events. Each of these can produce a maximum of one event per ms.

Confirming events

For a source to be able to produce its events, its validation bit must be set to 1. (for example: EVT_CAPT0_ENABLE for the capture event 0).

Processing

All the events sent by the module, regardless of the source, call up the same single event task in the PLC system.

There is normally only one type of event signaled per call. The switch to angle modulo information is signaled when the switch to Cycle modulo occurs. The source producing the call is determined in the event task via the Events input variable (%IWxy.0.12). This variable is updated at the beginning of event task processing.

Conditions

The number of the event task must be declared on the module configuration screen.

The module cannot send out more than one event per ms. This flow can be stopped if several modules send out events simultaneously on bus X.

The module has a buffer with 7 spaces that can be used to store several events awaiting transmission.

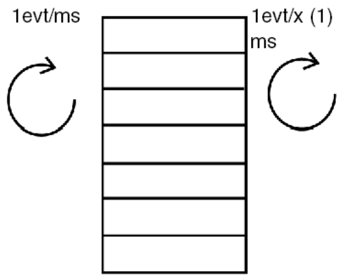
If the module cannot send all the internally produced events, the OVERRUN_EVT bit of the Events variable switches to 1.

For the "Modulo crossing" event, the configured current value format must be either type 2 (cyclic) or type 3 (infinite).

Overview of event management

At a Glance

This synopsis describes event management:

Process		TSX CCY 1128 module	TSX 57 processor
Event source	**Enabling events**		**In the EVT task**
Recalibrating the angle and number of cycles	evt_preset_enable %Qr.m.0.10	event buffer memory  	(* Recalibration *) ! IF evt_preset ... %IWr.m.0.12.2
Switch to angle rollover	evt_ang_enable %Qr.m.0.8		(* ANGLE rollover *) ! IF evt_ang ... %IWr.m.0.12.0
Switch to cycle rollover	evt_turn_enable %Qr.m.0.9		(* Cycle rollover *) ! IF evt_turn ... %IWr.m.0.12.1
Detection of cam	evt_cam_enable %Qr.m.0.13	**Note:** Saturation of the buffer memory is signaled by bit OVERRUN_EVT. This bit must be read in the event task. **(1)** x ms for 1 ms, normally if system event management is not saturated.	(*CAMS*) ! IF evt_cam ... %IWr.m.0.12.5
Capture 0	evt_capt0_enable %Qr.m.0.11		(* Capture 0 *) ! IF evt_capt0 ... %IWr.m.0.12.3
Capture1	evt_capt1_enable %Qr.m.0.12		(* Capture 1 *) ! IF evt_capt1 ... %IWr.m.0.12.4
parts counter full	evt_pieces_full_enable %Qr.m.0.14		(* Parts counter full *) ! IF evt_pieces_full ... %IWr.m.0.12.6

Language interface

At a Glance

The language interface defines all the data which can be read or modified by the application program. The data is implicit and periodic (%Q, %QW, %I, %IW) when it is updated automatically by the PLC task. It is explicit and aperiodic (%MW) when it is updated by the program after execution of the READ_STS or READ / WRITE_PARAM functions.

It is possible to associate with each word or bit a symbol which can be used by the application program (see Unity Pro Reference Manual). A table of symbols has been predefined. They can be assigned to the module data (see Unity Pro Reference Manual).

Processor and module exchanges

At a Glance

The module is used to modify parameters using the following exchanges:

Role	Name	Special Features
Transferring initial configuration and adjustment data	system exchanges	
Transferring current parameters	WRITE_PARAM	
	READ_PARAM	
Transferring initial parameters	RESTORE_PARAM	
	SAVE_PARAM	
Adjusting the axis	MOD_PARAM	These exchanges are specific to the cam module
Adjusting a track	MOD_TRACK	
Adjusting a cam	MOD_CAM	

System exchanges

Reminder on exchanges

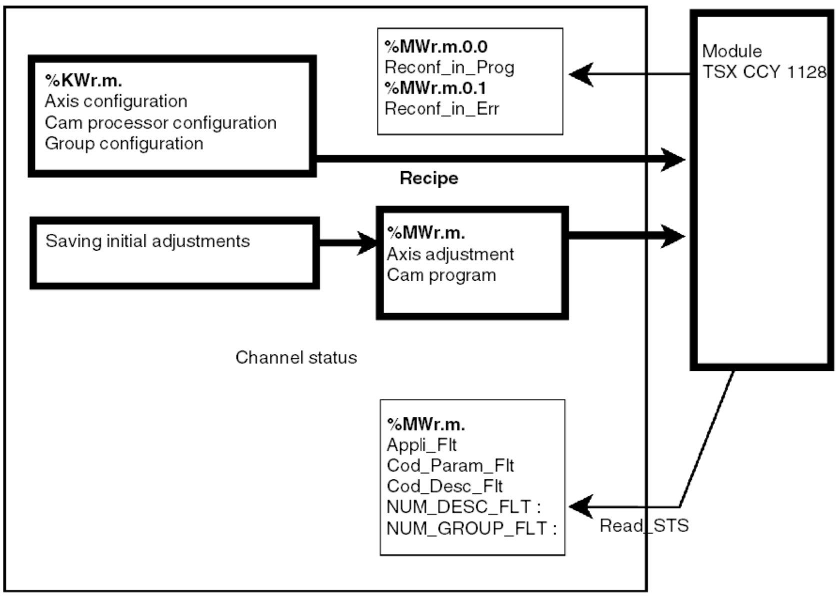
The exchanges comply with the exchange standard of the Premium range.

Transferring initial configuration and adjustment data

Data is transferred on a warm restart, cold restart, or reconfiguration request from a programming terminal in online mode.

The module stops after each transfer.

Processor memory



Transfer report

During the transfer, %MWr.m.0.0:15 `Reconf_In_Prog` is at 1.

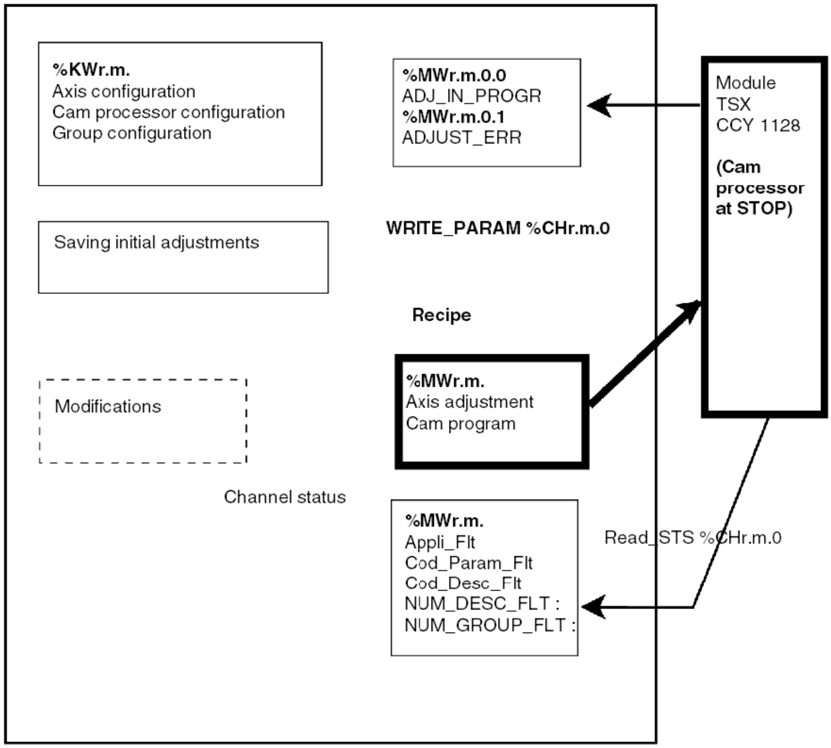
At the end of the exchange, the `Recnf_err` bit is set to 1 if the exchange has not been successful. The `READ_STS` function is used to update the channel status.

It contains the following data:
- `Appli_Flt` : The module does not have the configuration and adjustment data necessary for its operation.
- `Cod_Param_Flt` : error code found by the module in a configuration or adjustment data item in the axis. part.
- `Cod_Desc_Flt`: error code found by the module in a configuration or adjustment data item in the trackor cam.descriptor part.
- `Num_Desc_Flt` : codes the track or cam number containing a description error.
- `Num_Group_Flt` : codes the group number which contains the track or cam containing a description error.

WRITE_PARAM : Transfer of current recipe parameters

Reloading a modified recipe in the module

The `WRITE_PARAM` instruction is used to load the parameters of a modified recipe



All the recipe adjustment data are transmitted to the module using the `WRITE_PARAM` function. The cam processor is set to STOP.

If the exchange is faulty the module remains at STOP.

Transfer report

During the transfer, the bit %MWr.m.0.0.2 `Adj_In_ Prog` is set to 1.

At the end of the exchange, the `Adjust_err` bit (bit %MWr.m.0.1.2) is set to 1 if the exchange has not been successful. The `READ_STS` function is used to update the channel status.

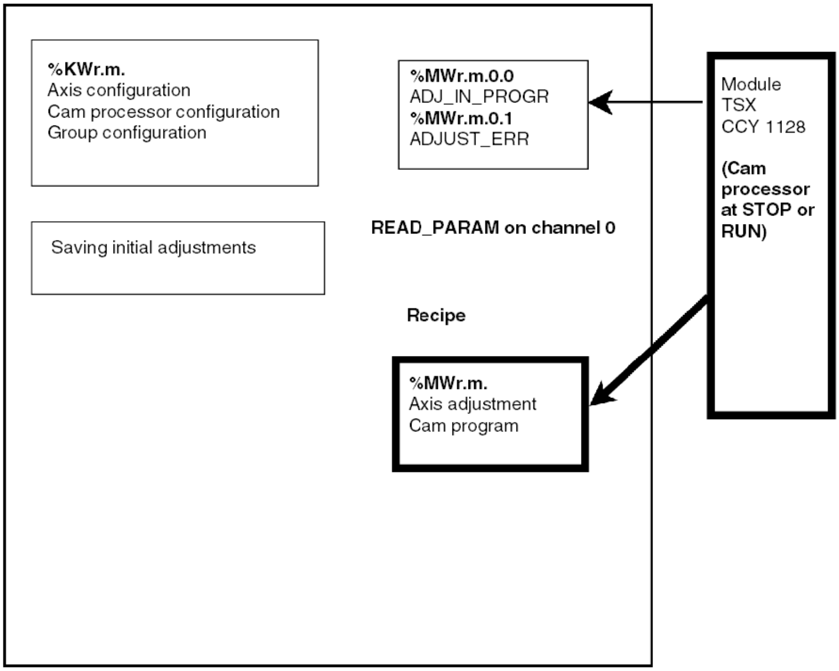
It contains the following data:

- `Appli_Flt` : the module refused the new recipe. The cam function cannot be activated. However, the old parameters contained in the module can be saved using the `READ_PARAM` function.
- `Cod_Param_Flt` : error code found by the module in a configuration or adjustment data item in the axis. part.
- `Cod_Desc_Flt`: error code found by the module in a configuration or adjustment data item in the trackor cam descriptor part.
- `Num_Desc_Flt` : codes the track or cam number containing a description error.
- `Num_Group_Flt` : codes the group number which contains the track or cam containing a description error.

READ_PARAM : Transfer of current recipe parameters

Returning current recipe parameters

The READ_PARAM instruction is used to return the parameters of the current recipe contained in this module..

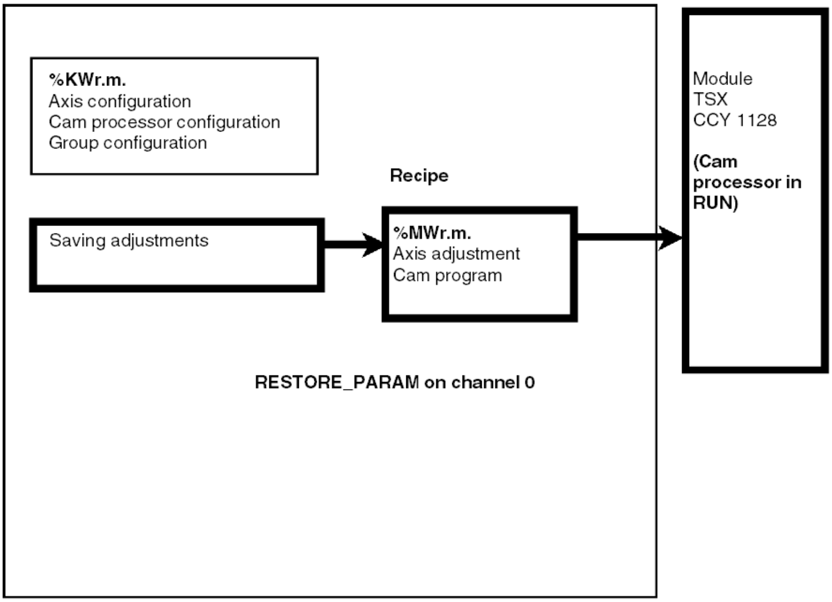


During the transfer, the ADJ_IN_PROGR bit is set to 1. The READ_PARAM instruction does not force the cam processor to STOP.

RESTORE_PARAM : Transferring initial parameters

Loading the original recipe

In an application, it may be necessary to reload the initial parameters of a recipe.

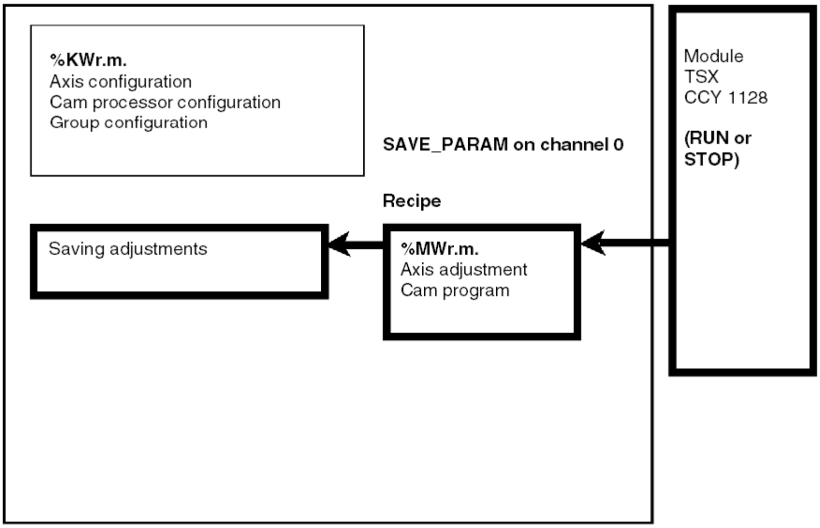


The operating mode is identical to that of `WRITE_PARAM`.

SAVE_PARAM : Transferring initial parameters

Saving recipe adjustments and modifications

In an application, after the adjustment and debugging phases have finished, it is necessary to save the newaxis parameters.



Saving the parameters can be done with the module and cam processor at RUN.

MOD_PARAM: Axis adjustment

General

The `MOD_PARAM` function is used to adjust an axis dynamically. In this case, transferring modified data does not mean that the cam processor must be switched to STOP. If the transfer is successful, the new parameters are taken into account by the module. If the transfer is not successful, the module uses the old axis adjustment values.

This function applies to a channel and therefore a IODDT type variable. We will use the variable `Channel_0` of type `T_CCY_GROUP0`.

`MOD_PARAM` assigns the following parameters to each exchange:
- `PRESET_ANG_VALUE`: angle recalibration value
- `PRESET_TURN_VALUE`: recalibration value of the number of cycles
- `SLACK_VALUE`: axis backlash value
- `MAX_PIECES`: parts counter limit value

Parameter buffer

The exchange zone used by the `MOD_PARAM` function is a buffer made up of reserved words: %MWr.m..0.16 to %MWr.m.0.19.

Loading the exchange zone

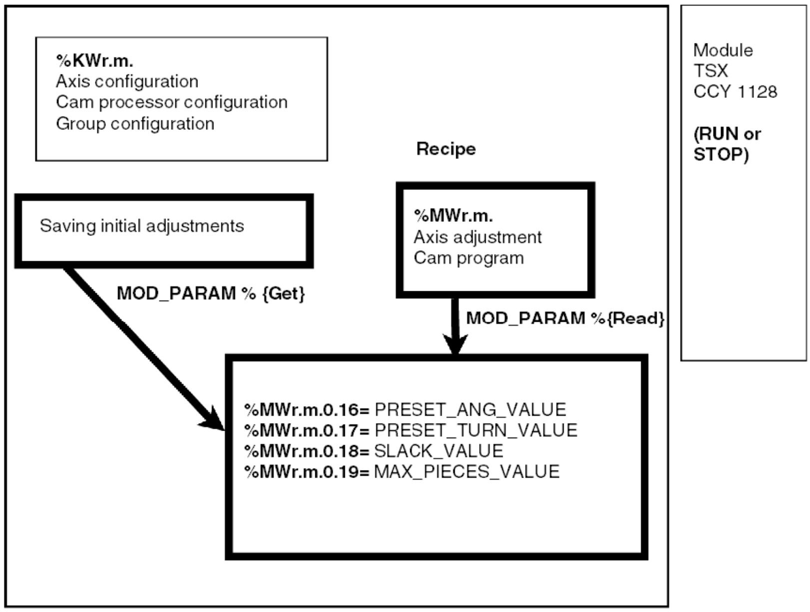
The exchange zone (buffer) can be preloaded:
- with the initial adjustment parameters `MOD_PARAM (Channel_0,0,0,0,0)` action {Get}
- with the current adjustment parameters `MOD_PARAM (Channel_0,1,0,0,0)` action {Read}

Sending the new values to the module

After modification in the buffer, the `MOD_PARAM (Channel_0,2,0,0,0)` function, action {Send} sends the new values to the module and updates the current parameters zone.

Initializing the parameter buffer

{Get} action or {Read} action



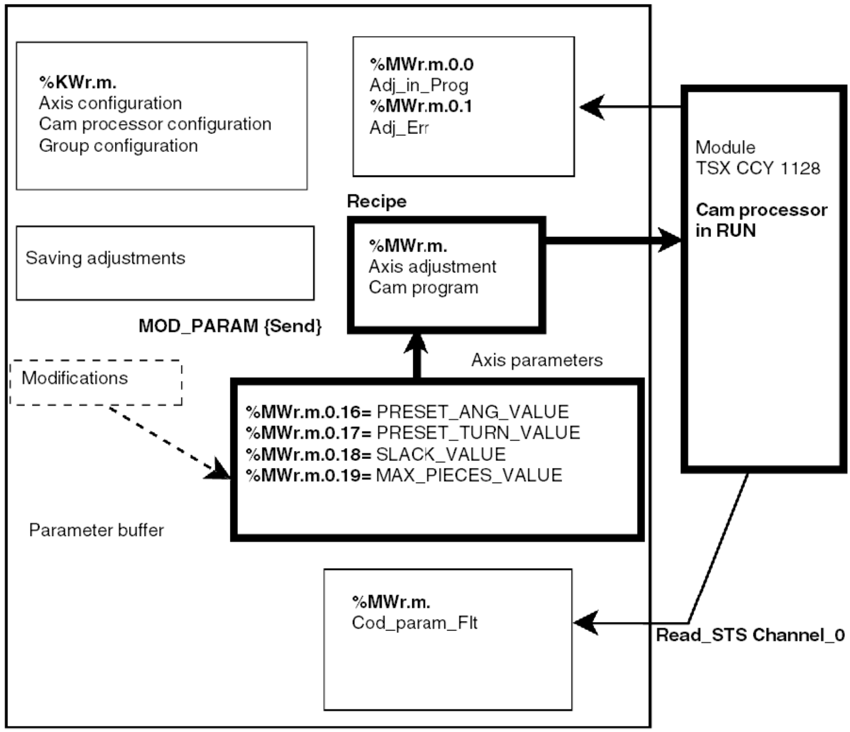
Example

{Get} `MOD_PARAM (Channel_0,0,0,0,0);`

{Read} `MOD_PARAM (Channel_0,1,0,0,0);`

Sending new parameters

{Send} action



Example

{Send} `MOD_PARAM (Channel_0,2,0,0,0)`

Checking the exchange

During the transfer, the bit (x15) `Adj_In_Prog` is at 1. At the end of the exchange, the `Adjust_err` bit is set to 1 if the exchange was not successful. The `READ_STS` function is used to update the channel status.

It contains the following data:

- `Cod_Param_Flt`: error code found by the module in a data item for the configuration or adjustment of the axis part.

MOD_TRACK: Adjusting a track

General

The MOD_TRACK function is used to adjust a track dynamically. Transferring new data does not require the cam processor to switch to STOP. If the transfer is successful, the new parameters are taken into account by the module. If the transfer is not successful, the cam processor remains in RUN with the old values.

The MOD_TRACK function only affects the feedforward value of a track.

This function applies to a channel and therefore a variable type IODDT. We will use the variable Channel_0 of type T_CCY_GROUP0.

Parameter buffer

The exchange zone used by the MOD_TRACK function is a buffer made up of the reserved word: %MWr.m.0.16.

Loading the exchange zone

The exchange zone (buffer) can be preloaded:

- with the initial feedforward values:
  `MOD_TRACK(Channel_0,0,Group,Track)` action {Get}
- with the current feedforward values:
  `MOD_TRACK(Channel_0,1,Group,Track)` action {READ}

Sending the new values to the module

After modification in the buffer:

- the function
  `MOD_TRACK(Channel_0,2,Group,Track)` action {Send}
  updates the track feedforward value in the module and in the current parameters zone.

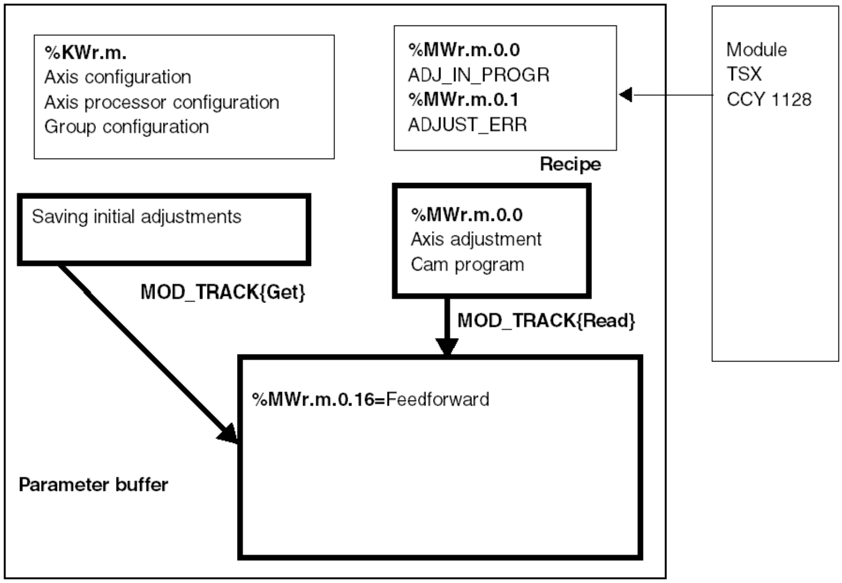
Key

Group: identifies the group number

Track: identifies the track number

Initializing the parameter buffer

{Get} action or {Read} action



Example

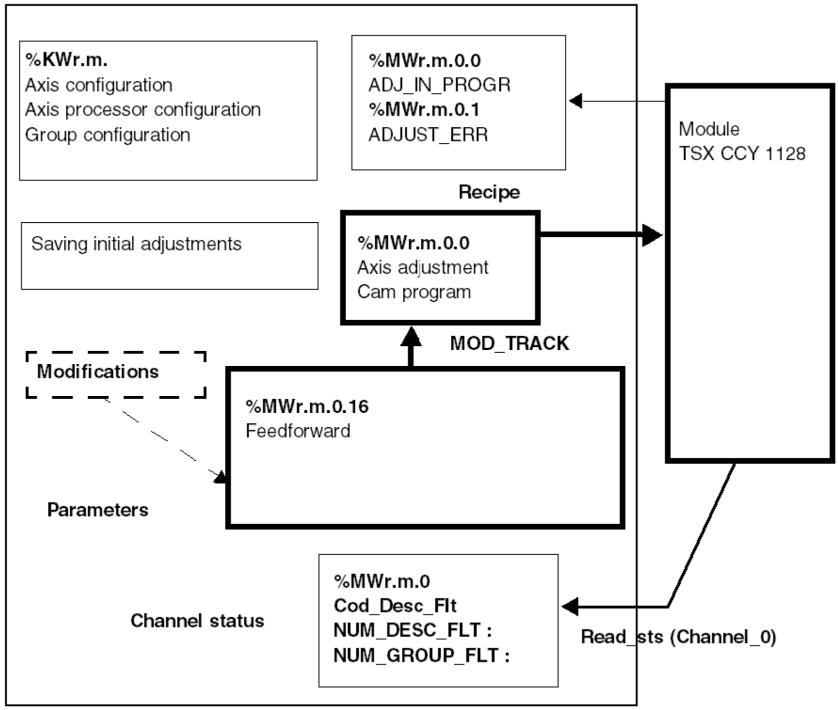
For Group 0 Track 4

- {Read}: `MOD_TRACK(Channel_0,1,0,4)`
- {Get}: `MOD_TRACK(Channel_0,0,0,4)`

The execution of the MOD_TRACK function can be monitored via the channel status level.

Sending the new

{Send} action



Example

For Group 0 Track 4:
- {Send}: MOD_TRACK (Channel_0,2,0,4)

Checking the exchange

During the transfer, the bit (x15) `Adj_In_ Prog` is set to 1.

At the end of the exchange, the `Adjust_err` bit is set to 1 if the exchange has not been successful. The `READ_STS` function is used to update the channel status.

It contains the following data:
- `Cod_Desc_Flt`: error code found by the module in a track adjustment data item.
- `NUM_DESC_FLT`: codes the track number containing a description error.
- `NUM_GROUP_FLT`: codes the group number which contains the track containing a description error.
- `COD_LOCAL_FLT`: indicates an error in the group number or track number.

MOD_CAM: Adjusting a cam

General

The `MOD_CAM` function is used to adjust a cam dynamically. Transferring new data does not require the cam processor to switch to STOP. If the transfer is successful, the new parameters are taken into account by the module. If the transfer is not successful, the cam processor remains in RUN with the old values.

The `MOD_CAM` function assigns the following parameters:
- `X1`: Low threshold
- `X2`: high threshold
- `TIME_SWITCH_OFF`: timer

This function applies to a channel and therefore a variable type IODDT. We will use the variable `Channel_0` of type `T_CCY_GROUP0`.

Parameter buffer

The exchange zone used by the `MOD_CAM` function is a buffer made up of reserved words: `%MWr.m.0.16` to `%MWr.m.0.18`.

Loading the exchange zone

The exchange zone (buffer) can be preloaded:
- with the initial adjustment values:
  `MOD_CAM(Channel_0,0,Group,Cam)` action {Get}
- with the current adjustment values:
  `MOD_CAM(Channel_0,1,Group,Cam)` action {Read}

Sending the new values to the module

After modification in the buffer, the function:
- `MOD_CAM(Channel_0,2,Group,Cam)` action {Send}
  sends the new values to the module and updates the current parameters zone.

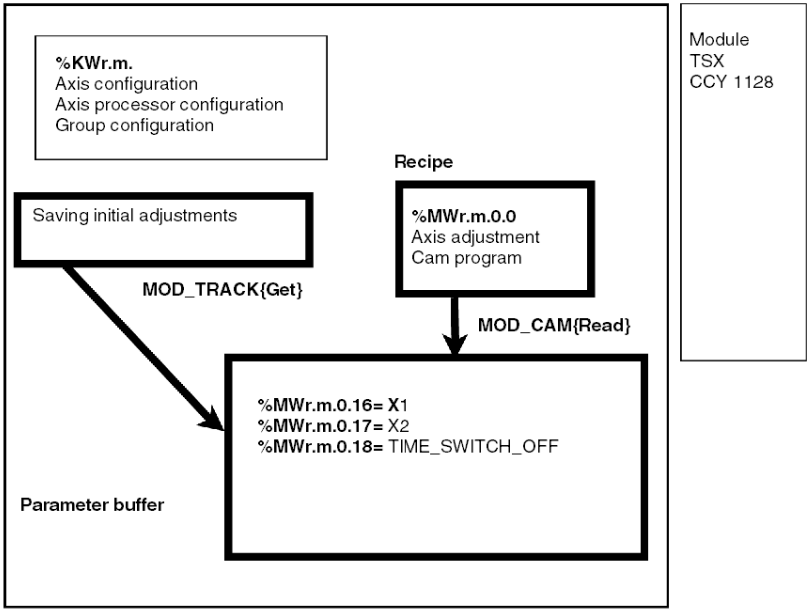
Key

Group: identifies the group number

Cam: identifies the tracknumber

Initializing the parameter buffer

{Get} action or {Read} action



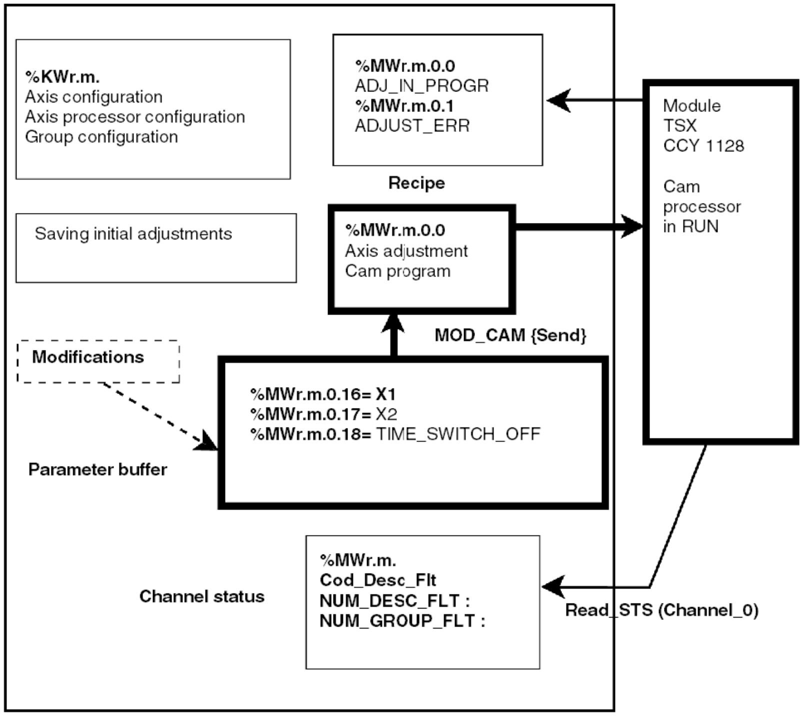
Initializing the parameter buffer

For cam 9 in group 2:

- {Read}action: `MOD_CAM(Channel_0,1,2,9)`
- {Get} action: `MOD_CAM(Channel_0,0,2,9)`

Sending the new

{Send} action



Example

For cam 9 in group 2:
- {Send} action: `MOD_CAM(Channel_0,2,2,9)`

Checking the exchange

At the end of the exchange, the Adjust_err bit is set to 1 if the exchange has not been successful. The `READ_STS` function is used to update the channel status.

It contains the following data:
- `COD_DESC_FLT`: error code found by the module in an adjustment data item in the cam.
- `NUM_DESC_FLT`: codes the cam number containing a description error.
- `NUM_GROUP_FLT`: codes the group number which contains the cam containing a description error.
- `COM_LOCAL_FLT`: indicates an error in the group number or track number.

TRF_RECIPE: Recipe transfer functions

General

An application program can use several recipes. These are contained in several memory zones. The TRF_RECIPE instruction is used to:

- transfer the contents of the current recipe to a memory zone.
- transfer a recipe from a memory zone to the %MW zone containing the current recipe and transfer it to the module. In this case the cam processor switches to STOP mode as for the WRITE_PARAM instruction.

NOTE: It is possible to save (Restore) one or more recipes to the memory zone in a paged PCMCIA using the WRITE_PCMCIA (READ_PCMCIA) instructions.

TRF_RECIPE: Functions for storing recipes

%MW for the recipe

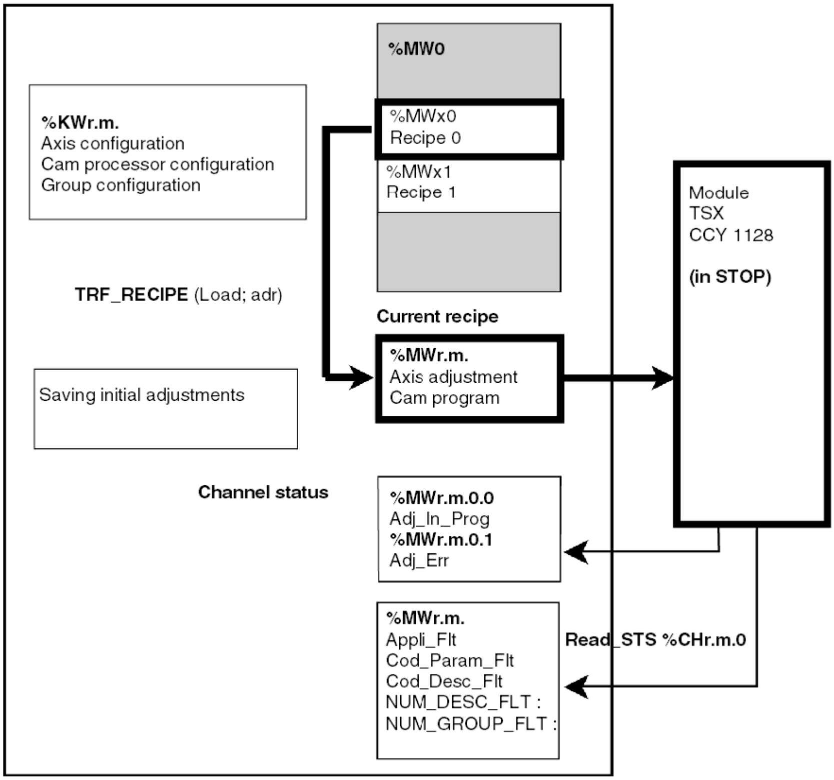
A recipe is stored in %MW words in this way (n is the value of the "ADR" parameter coded in the instruction TRF_RECIPE):

Word addresses	**Content**	**Number of words**	**Details**	**Group**
%MWn to %MWn+11	Adjusting the axis	12 words		
%MWn+12 to %MWn+27	Descriptor for tracks 0 to 7	16 words	• SPECIF-TRACK track 0 • ANTICIP-FACTOR track 0 • ... • SPECIF-TRACK track 7 • ANTICIP-FACTOR track 7	Group 0
%MWn+28 to %MWn+187	Descriptor for cams 0 to 31	160 words	**Cam 0** • SPECIF-CAM_0 • SPECIF_CAM_1 • TIME_SWITCH_OFF **Cam...** • ... **Cam 31** • SPECIF-CAM_0 • SPECIF_CAM_1 • TIME_SWITCH_OFF	
%MWn+188 to %MWn+203	Descriptor for tracks 0 to7	16 words	see Group 0	Group 1
%MWn+204 to %MWn+363	Descriptor for cams 0 to 31	160 words	see Group 0	
%MWn+364 to %MWn+379	Descriptor for tracks 0 to 7	16 words	see Group 0	Group 2
%MWn+380 to %MWn+539	Descriptor for cams 0 to 31	160 words	see Group 0	
%MWn+540 to %MWn+555	Descriptor for tracks 0 to 7	16 words	see Group 0	Group 3
%MWn+556 to %MWn+715	Descriptor for cams 0 to 31	160 words	see Group 0	

TRF_RECIPE: Loading a new recipe

Overview of loading

Loading is triggered by a call from the application program.



Example

Loading defines word %MW800. We will use a variable `Channel_0` of type IODDT `T_CCY_GROUP0`

Action {Load} `TRF_RECIPE (Channel_0,0,800)` or

%MW0:= 800; `TRF_RECIPE (Channel_0,0,%MW0)`

Checking the exchange

During the transfer, the bit (x15) Adj_In_ Prog is set to 1.

At the end of the exchange, the Adjust_err bit is set to 1 if the exchange has not been successful. The READ_STS function is used to update the channel status.

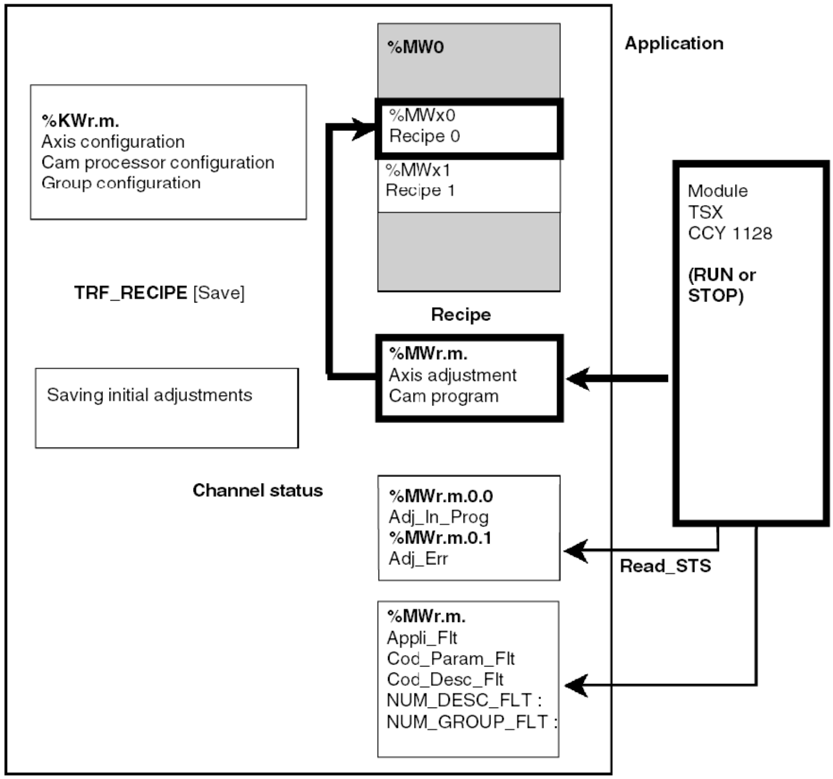
It contains the following data:

- Appli_FLT : the module refused the new recipe. The cam function cannot be activated. However, the old parameters contained in the module can be retrieved using the READ_PARAM instruction.
- Cod_Param_Flt: error code found by the module in a configuration or adjustment data item in the axis. part.
- Cod_Desc_Flt: error code found by the module in a configuration or adjustment data item in the trackor cam descriptor part.
- NUM_DESC_FLT: codes the track or cam number containing a description error.
- NUM_GROUP_FLT: codes the group number which contains the track or cam containing a description error.

TRF_RECIPE: Saving a new

Saving overview

Saving a recipe:



Example

Transferring the recipe used in the module to the %MW800 table. We will use a variable `Channel_0` of type IODDT `T_CCY_GROUP0`.

(* Action {Save} *) `TRF_RECIPE (Channel_0,1,800);` or

%MW0:= 800; `TRF_RECIPE (Channel_0,1,%MW0);`

Checking the exchange

During the transfer, the bit (x15) Adj_In_ Prog is set to 1.

At the end of the exchange, the Adj_err bit is set to 1 if the exchange has not been successful. The READ_STS function is used to update the channel status.

It contains the following data:

- Appli_FLT : the module refused the new recipe. The cam function cannot be activated. However, the old parameters contained in the module can be retrieved using the READ_PARAM instruction.
- Cod_Param_Flt: error code found by the module in a configuration or adjustment data item in the axis. part.
- Cod_Desc_Flt: error code found by the module in a configuration or adjustment data item in the trackor cam descriptor part.
- NUM_DESC_FLT: codes the track or cam number containing a description error.
- NUM_GROUP_FLT: codes the group number which contains the track or cam containing a description error.

DETAIL_OBJECT : Operator dialogue interface

General

The DETAIL_OBJECT function facilitates the management and creation of a recipe using operator dialog. It gives the application program use of all the descriptive information for a track or a cam in a %MW memory zone selected by the programmer. If we associate the variable Channel_0 of type IODDT T_CCY_GROUP0 to channel 0 of the module to obtain:

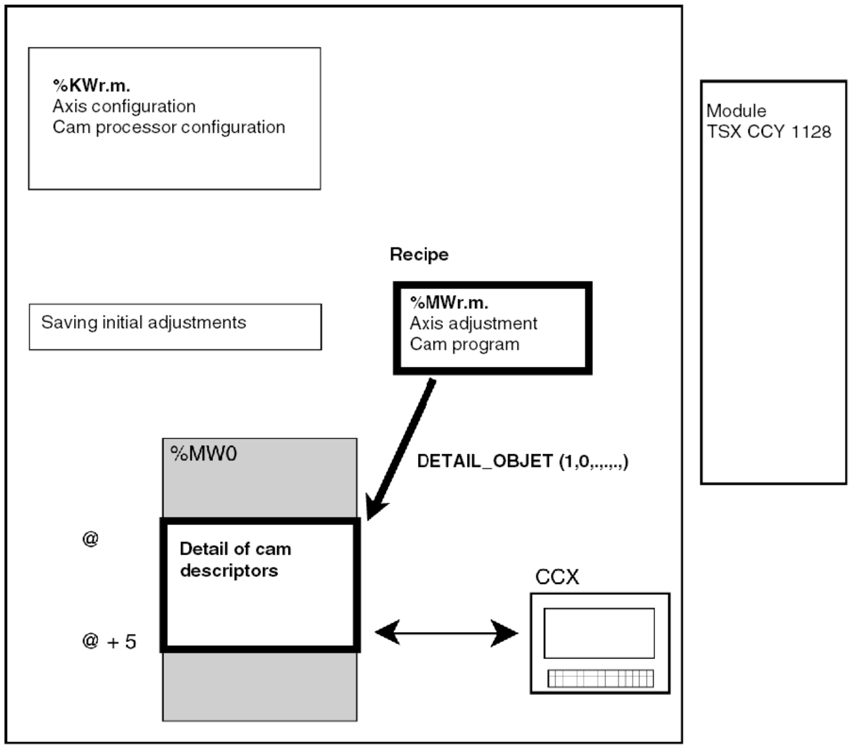
Function parameters

- DETAIL_OBJECT (Channel_0,Action, type_object, num_group, num_object, adr)
- Action = 1 : Ext is used to write the cam or track descriptor in a memory zone
- Action = 0 : Inc is used to write the cam or track descriptor with data in the memory zone
- type of object = 0: cam.
- type of object = 1: track.
- NUM_GROUP = number of the group to which the cam or track belongs.
- NUM_OBJET = number of the cam or track in the group.
- ADR = address of the first object in the memory zone.

DETAIL_OBJECT: Transferring cam details

Overview

This diagram describes transferring cam details



Example

Breakdown of parameters for cam 9 in group 2 from address %MW100.

This function applies to a channel and therefore a variable type IODDT. In this example, we will use the variable `Channel_0` of type `T_CCY_GROUP0`.

```
DETAIL_OBJECT (Channel_0,1,0,2,9,100);
```

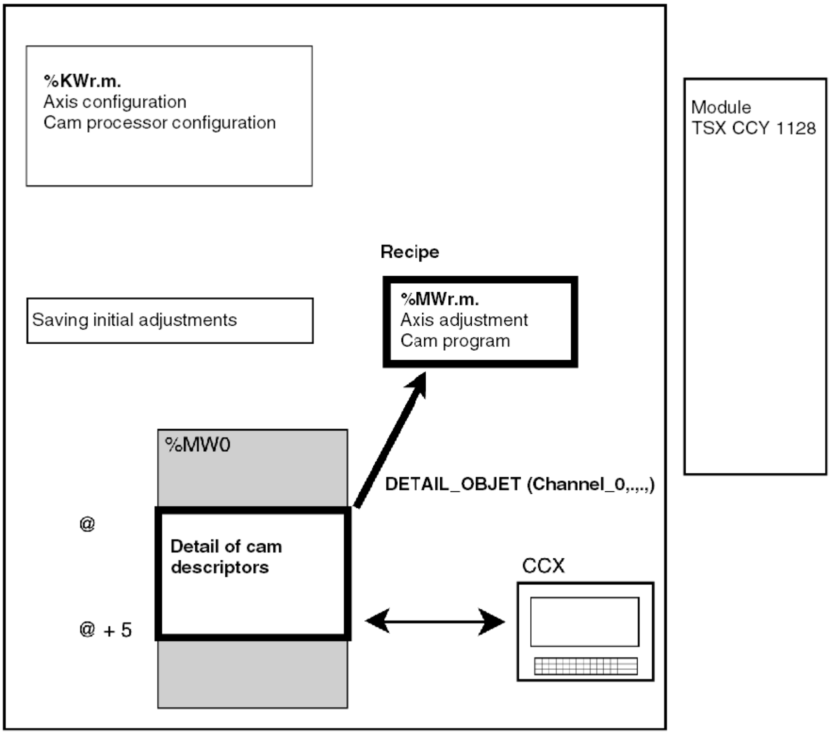
Order in which cam parameters are stored

The parameters are stored in a %MW memory zone:

Rank	Meaning
0	bit 0 = 1 cam used
1	type of cam
2	bit 0 =1 valid in forward direction, bit 1 =1 valid in reverse direction
3	reserved
4	reserved
5	reserved
6	track number
7	reserved
8	cam monitoring selection code ● 0 : always valid ● 1 : enable bit condition ● 2 : counter full condition
9	Cam monitoring bit number
10	reserved
11	reserved
12	X1: low threshold
13	X2: high threshold
14	time value on opening
15	reserved

Loading details of a cam

The parameters are transferred into the recipe buffer from the memory zone



⚠ CAUTION

DATA INCONSISTENCY

Before transferring the parameters, do check their consistency with cam and/or recipe.

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

Example

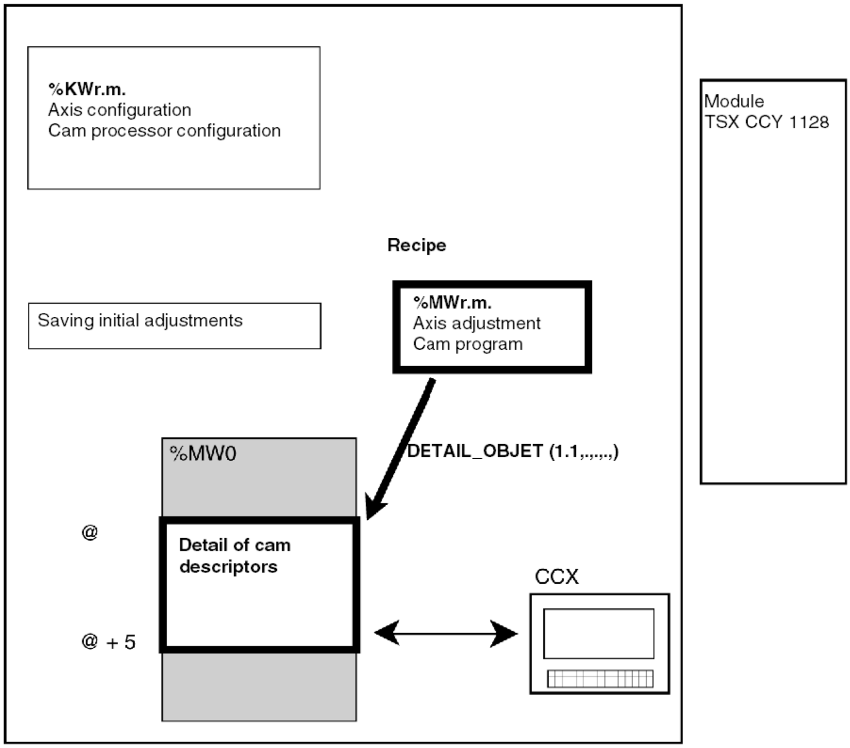
Loading the details of cam 3, group 2, into the current recipe

```
DETAIL_OBJECT (Channel_0,0,0,2,9,100);
```

DETAIL_OBJECT: Transferring track details

Overview

This diagram describes transferring a track



Example

Breakdown of parameters for track 5 group 1 from address %MW200. Loading details in the current recipe of cam 3 of group 2. We will use a variable `Channel_0` of type IODDT `T_CCY_GROUP0`

```
DETAIL_OBJECT (Channel_0,1,1,1,5,100);
```

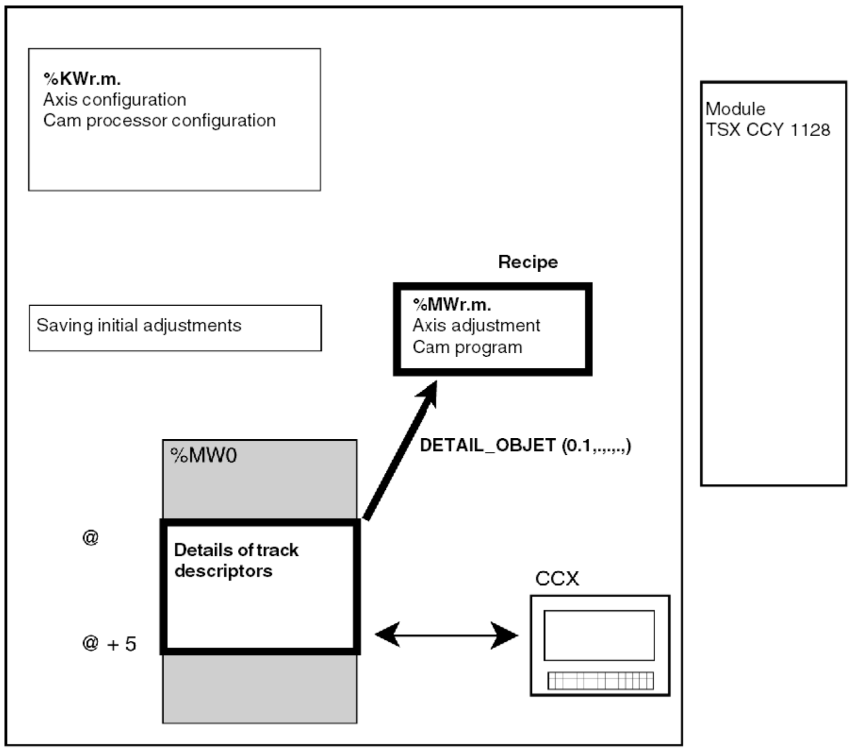
Order in which track parameters are stored

The parameters are stored in a %MW memory zone:

Rank	Meaning
0	bit 0 = 1 track used
1	Output, bit 0: INVERT_TRACK, bit 1: ADD_TRACK
2	Action code for switching to 1 for parts counter in forward direction
3	Action code for switching to 0 for parts counter in reverse direction
4	= 1 track declared as event
5	value offeedforward factor.

Loading details of a track

The parameters are transferred into the recipe buffer from the memory zone



⚠ CAUTION

DATA INCONSISTENCY

Before transferring the parameters, do check their consistency with track and/or recipe.

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

Example

Changing the details of track 5, group 1, in the current recipe from word %MW100. We will use a variable `Channel_0` of type IODDT `T_CCY_GROUP0`

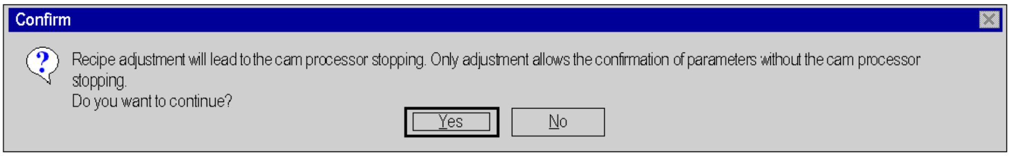
```
DETAIL_OBJECT (Channel_0,0,1,1,5,100);
```

Reconfiguration in connected mode

Procedure

When all the configuration parameters are modified, confirm these parameters using the **Edit/Confirm** command or click on the confirmation icon.

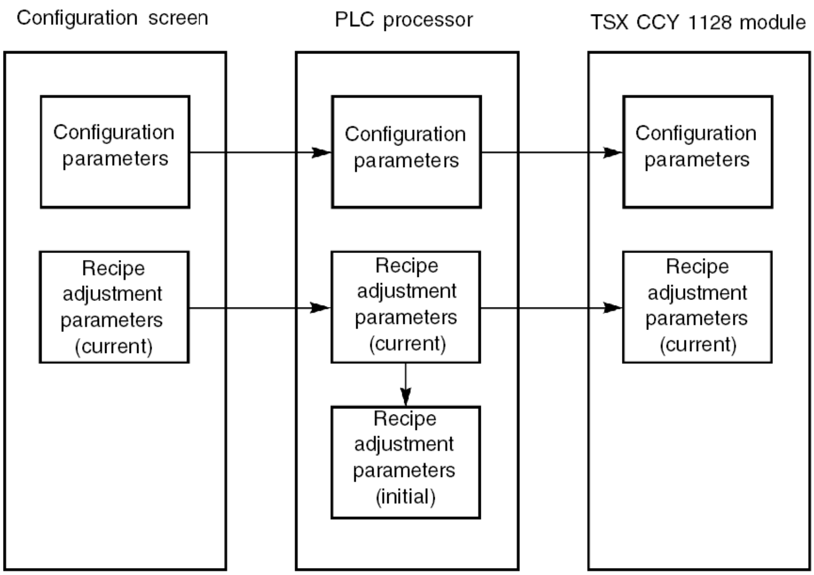
Only the parameters that are not grayed out can be modified in connected, the other parameters, (task, event, masking faults, encoder and encoder configuration) must be modified in local mode.



Any reconfiguration in connected mode causes the cam processor to stop its operations.

Illustration

The following diagram describes the reconfiguration process.



Entering recipe adjustment parameters for the electronic cam module

13

Subject of this chapter

This chapter describes the operations for entering the recipe adjustment parameters for the TSX CCY 1128 electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Accessing the recipe adjustment parameters for the module	220
Entering acquisition parameters for an incremental encoder	221
Entering the acquisition parameters for an absolute encoder	223
Parametering for the parts counter	225
Activating/deactivating the tracks	226
Parametering for the tracks	227
Creating cams	229
Parametering the cams	230
Cam in position	231
Monostable cam	234
Breaking cam	235
Parametrizing the confirmation condition associated with a cam	236
Confirming the recipe adjustment parameters	237
Saving the recipe adjustment parameters	238
Restoring the recipe adjustment parameters	239

Accessing the recipe adjustment parameters for the module

Role

This operation is used to access the recipe adjustment parameters for the TSX CCY 1128 module.

Adjust Recipe mode provides access to all the recipe parameters.

Confirming modifications online causes the cam function to stop (Cam processor to Stop). **Adjust** mode can be used to access certain parameters (values in blue in the associated fields) without stopping the cam processor.

NOTE: The values are updated when the window is opened.

Preliminary conditions

The TSX CCY 1128 module must already be configured.

Procedure

The following table describes the operations to be performed:

Step	Action
1	Access the configuration screen of the module.
2	Select the **Adjust** tab.
3	Confirm the configuration entered if the confirmation request dialog box is displayed.
4	In the browser, click on the recipe element to be entered: • Acquisition • Parts counter • Connector 0 or 1

Entering acquisition parameters for an incremental encoder

Role

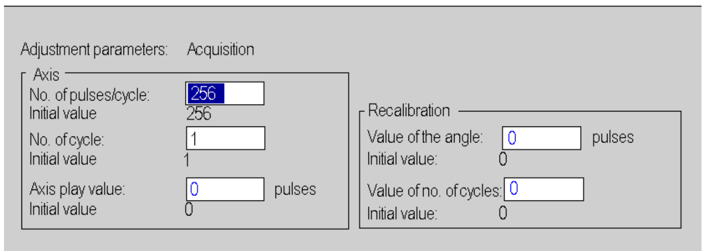
This operation defines the recipe adjustment parameters linked with the incremental encoder.

Preliminary conditions

This operation requires an incremental encoder to have been defined during configuration.

Procedure

The following table describes the operations to be performed:

Step	Action
1	In the browser, double-click on **Acquisition**. **Result**: 
2	Enter the Axis parameters. The parameters that are displayed depend on the reading format chosen in the configuration: ● Type 1: number of pulses per cycle (minimum value 256 and maximum value 32767). ● Type 2: number of pulses per cycle (minimum value 256 and maximum value 32767) and number of cycles (minimum value 1 and maximum value 32767) ● Type 3: number of pulses per cycle = 32 767. The value that is displayed cannot be modified. It indicates the counting capacity.

Step	Action
3	Enter the axis play value. This value is used to compensate for the position error caused by changing the direction of rotation, if there is mechanical play between the drive and the axis (encoder). If recalibration of the axis is done in the direction of rotation: • **forward**, declare a negative correction value. The correction will be made on reverse movements, • **reverse**, declare a positive correction value. The correction will be made on forward movements. A value of 0 corresponds to no correction. The values are between -(No. of pulses/cycle) /2 and +(No. of pulses/cycle) /2. The correction value cannot exceed 1023 pulses. **Note**: this correction modifies the switching position of all the cams according to the declared value. Whatever the value of the play entered, the position and captures values sent to the PLC are current values (no corrections).
4	Enter the recalibration value: • value of the angle, • value of the number of cycles (for type 2 reading format only). The recalibration value is loaded in the position counter on the recalibration command. The values are between 0 and + No. of pulses/cycle

Entering the acquisition parameters for an absolute encoder

Role

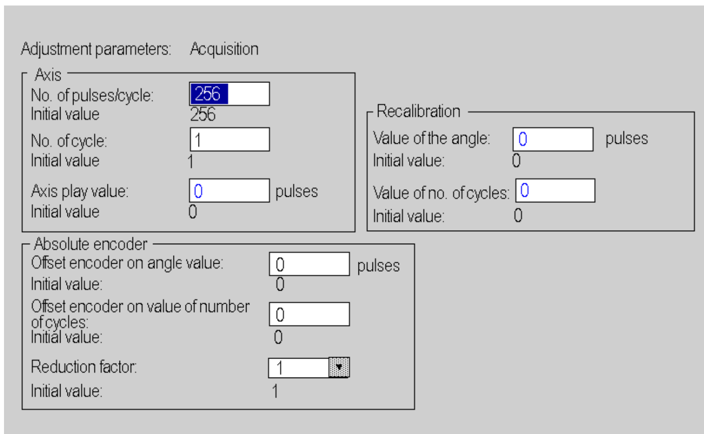
This operation allows you to define an absolute encoder's recipe adjustment parameters .

Preliminary conditions

This operation requires that an absolute encoder has already been configured.

Procedure

The following table describes the operations to be performed:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the browser, double-click on Acquisition.
Result:

</td></tr>
<tr><td>2</td><td>Enter the Axis parameters. The parameters that are displayed depend on the reading format chosen in the configuration:
● Type 1: number of pulses per cycle.
● Type 2: number of pulses per cycle and number of cycles.

Note:
Choose a value to the power of 2, for the number of pulses per cycle.
Min. value: number of pulses per cycle = 256 and number of cycles = 1
Max. value: 2^{14} (see the formula in step 5 of the table).</td></tr>
</table>

Step	Action
3	Enter the encoder offset value (angle and cycle) to obtain a position value equal to 0 when the axis arrives at the machine 0. This value is the gross value provided by the encoder when the axis is on the machine 0. The values consist of: • 0 to No of pulses/cycle for the encoder offset on the angle, • 0 to No of cycles for the encoder offset on number of cycles.
4	Enter the axis play value. This value is used to compensate for the position error caused by changing the direction of rotation, if there is mechanical play between the drive and the (encoder) axis. • **forward**, declare a negative correction value. The correction will be made on reverse movements, • **reverse**, declare a positive correction value. The correction will be made on forward movements. A value of 0 corresponds to no correction. The values are between -(No of pulses/cycle) /2 and +(No of pulses/cycle) /2. The correction value cannot exceed 1023 pulses. **Note**: this correction modifies the switching position of all the cams according to the declared value. Whatever the value of the backlash entered, the position and captures values sent to the PLC are current values (no corrections).
5	Enter the reduction factor. This factor reduces the encoder resolution. The position given by the encoder is divided by the reduction factor. This enables the cam program to be effective over a narrower range of pulses than that given by the machine. Possible reduction factor values: 1 (the cam program reacts in relation to the actual position of the encoder, no reduction), 2 (the position given by the encoder is divided by 2, 4, 8, 16 or 32). In addition the following relationship must be checked: $2^{\text{no of data bits}}$ ≥ no of pulses/cycles x no of cycles x reduction factor (number of pulses less than or equal to 2^{14})
6	Enter the recalibration value (angle and cycle). This recalibration value is systematically added to the position value after correction of the encoder offset. It moves the axis in relation to the machine reference point. The values consist of: • 0 to No of pulses/cycles for the recalibration on the angle, • 0 to No of cycles for recalibration on the number of cycles. **Note:** The position value (angle and cycle) after offset correction and recalibration application is: Position value = Gross encoder value - Offset + Recalibration

Parametering for the parts counter

Role

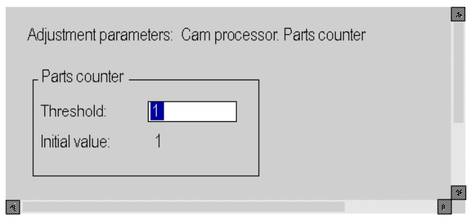
The parts counter is used to:
- indicate the number of parts processed,
- confirm the action of a cam when a certain number of parts (limit value) has been completed.

This operation is used to set the limit value of the parts counter.

The parts counter is incremented, decremented or reset to 0 when changing to 1 on a track according to the cam program.

Procedure

The following table describes the operations to be performed:

Step	Action
1	In the browser, double-click on **Parts counter**. **Result** 
2	Enter the limit value for the parts counter (value from 1 to 32767)

Activating/deactivating the tracks

Role

The activation operation defines the tracks to be used.

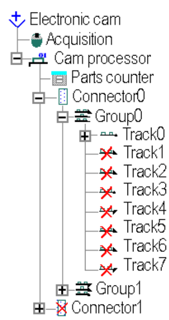
Deactivation frees the unused tracks. The cams associated with these tracks are then destroyed and can be reused for the other tracks in the same group.

Preliminary conditions

The connectors that these tracks are associated to must first be unlocked (in Configuration mode).

Procedure

The following table describes the operations to be performed:

Step	Action
1	In the browser double click on **Connector 0 or 1** then double click in the browser on **Group 0, 1, 2 or 3** to access the track(s) to be activated. **Result**: The browser displays all the tracks that are associated with this connector and this group. Tracks that are crossed out are inactive.
2	Click with the right mouse button on the track to be activated and select the **Activate** command (**Deactivate** for the reverse operation). The cross disappears from the activated track. Example: track 0 of connector 0 group 0 is active and tracks 1 to 7 are inactive. 

Parametering for the tracks

Role

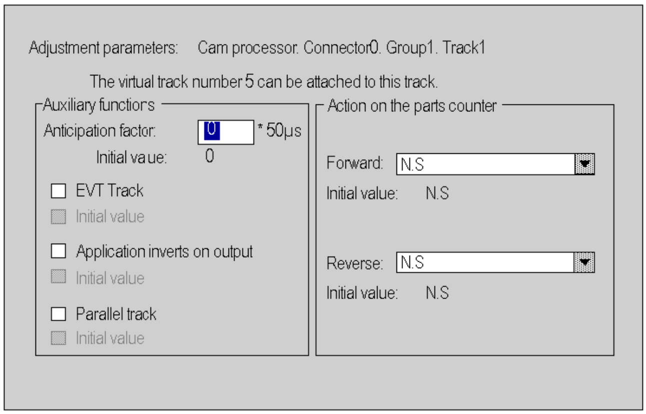
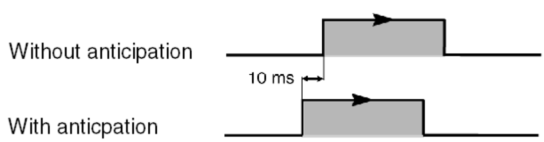
This operation is used to define the tracks parameters.

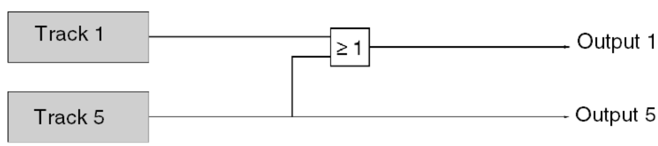
Preliminary conditions

These tracks must first be activated.

Procedure

The following table describes the operations to be performed:

Step	Action
1	Double-click on the track to be parameterized. **Result**: 
2	Enter the value of the anticipation factor from 0 to 32767 x 50 micro s. This factor makes it possible to anticipate all switches of track in order to compensate for the delay caused by the actuators of the machine. **Example**: anticipation with a factor of 200 
3	Select **EVT** Track to trigger an event at each change of track.

Step	Action
4	Check **Invert application on output** to invert the status of the track when the cam processor is in "Run" mode. This inversion has no effect when the processor is in "Stop". The track remains at 0. If the track does not have a cam, this function does not apply.
5	If the track is to perform operations on the parts counter, select the following types of action: • **No action** • **Inc**: increments the parts counter each time a cam on the track changes to 1 • **Dec**: decrements the parts counter each time a cam on the track changes to 1 • **Reset**: resets the parts counter to zero each time a cam on the track changes to 1 in Forward and/or Reverse direction.
6	Check **Parallel track** to put a second predefined track (track n+4) from the same group in parallel on the output of track n. This option allows you to: • associate tracks 4 to 7 with the outputs of tracks 0 to 3 in the same group. • have 2 tracks with different characteristics (different anticipation factor depending on the direction of movement) on a single output. If the track does not have a cam, this function does not apply. **Example**: if the option is selected on track 1, the value of output 1 will be the "OR logic" of tracks 1 and 5.  **Note**: This choice only appears on tracks 0 to 3 of each group.

Creating cams

Role

This operation associates one (or more) cams to one track.

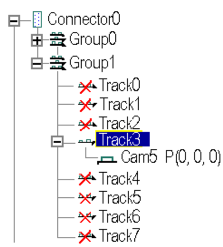
It is possible to associate up to 32 cams with one track.

Preliminary conditions

The tracks that you wish to associate the cams to must first be activated (in Configuration mode).

Procedure

The following table describes the operations to be performed:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>In the browser, click with the right mouse button on the track concerned, and select the Create cam command…</td></tr>
<tr><td>2</td><td>In the dialog box select the number of the cam (0 to 31) and confirm.
The number of a cam already being used in the group cannot be entered.
Result:
The browser displays the cam that has been created under the track to which it belongs.
Example: creation of cam 5 under track 3 of group 1 of connector 0

</td></tr>
</table>

Parametering the cams

Role

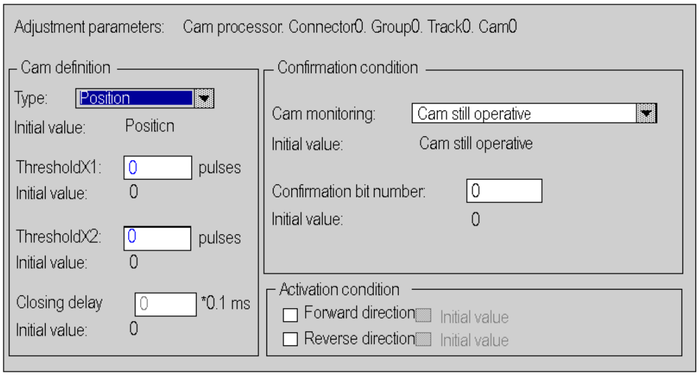
This operation allows you to define the cam adjustment parameters.

Preliminary conditions

These cams must first be created.

Procedure

The following table describes the operations to be performed:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Double-click on the cam to be parameterized.
Result

</td></tr>
<tr><td>2</td><td>Enter the value of each parameter:
type of came: Position, Monostable or Braking,
values of associated thresholds, and time delay (for the Monostable cam),
cam confirmation condition with the confirmation bit number,
direction for which the cam is active.</td></tr>
</table>

Cam in position

Role

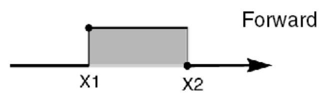
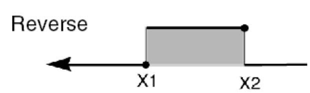
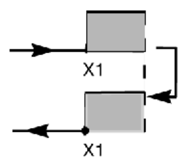
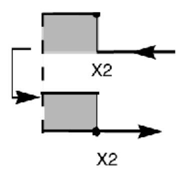
A Position type cam is at state 1 when the position of the axis is between the 2 thresholds. The 2 thresholds, low threshold X1 and high threshold X2 must be defined (X1 and X2 both between 0 and the number of pulses/cycles). X1 can be greater than the value of X2. The cam is then active between two cycles.

One type of activation from the 3 suggested: forward/reverse, forward and reverse, must be selected.

NOTE: The confirmation condition must be at state 1 for the cam to be able to change to state 1.

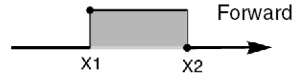
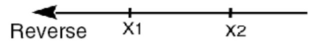
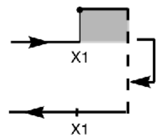
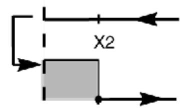
Operation of a cam which is active in forward and reverse directions

This table describes the performance of a cam that is active in forward and reverse directions.

Description	Illustration
If the movement is forward, the cam changes to 1 when threshold X1 is crossed and returns to 0 when threshold X2 is crossed.	
If the movement is reversed, the cam changes to 1 when threshold X2 is crossed and returns to 0 when threshold X1 is crossed.	
If the movement is: ● forward (without crossing threshold X2), the cam changes to 1 when threshold X1 is crossed. ● then reverse, the cam returns to 0 when threshold X1 is crossed	
If the movement is: ● reverse (without crossing threshold X1), the cam changes to 1 when threshold X2 is crossed ● then forward, the cam returns to 0 when threshold X2 is crossed	

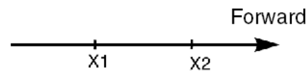
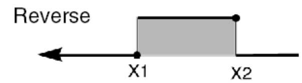
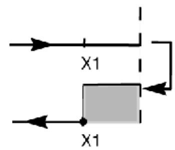
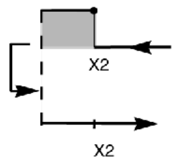
Operation of a cam which is active in the forward direction

This table describes the performance of a cam that is active in the forward direction.

Description	Illustration
If the movement is forward, the cam changes to 1 when threshold X1 is crossed and returns to 0 when threshold X2 is crossed	
If the movement is reverse, the cam remains at 0.	
If the movement is: • forward (without crossing threshold X2), the cam changes to 1 when threshold X1 is crossed • then reverse, the cam returns to 0 as soon as the direction of movement changes.	
If the movement is: • reverse, the cam remains at 0 • then forward, the cam returns to 1 when threshold X2 is crossed.	

Operation of a cam that is active in the reverse direction

This table describes the performance of a cam that is active in the reverse directions.

Description	Illustration
If the movement is forward, the cam remains at 0.	
If the movement is reverse, the cam changes to 1 when threshold X2 is crossed and returns to 0 when threshold X1 is crossed	
If the movement is: ● forward, the cam remains at 0 ● then reverse, the cam returns to 1 until threshold X1 is crossed.	
If the movement is: ● reverse, the cam changes to 1 when threshold X2 is crossed ● then forward, the cam returns to 0 as soon as the direction of movement changes.	

Conditions for angle 0

Threshold X1 can have a value greater than X2 if the cam is active either side of value 0, between:

- X1 and 0
- 0 and X2

Monostable cam

Definition

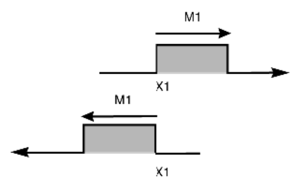
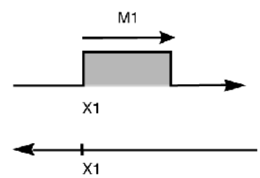
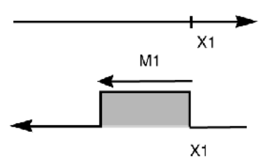
A Monostable type cam is one that changes to 1 when a threshold is crossed and returns to 0 after a time delay.

It is characterized by:
- a value of threshold X1 expressed in a number of pulses (X1 is between 0 and the number of pulses/cycles)
- An M1 time delay expressed in 1/10 ms (0 to 16383, i.e. 1.6383 s max.)
- An activation type: forward/reverse, forward or reverse.

NOTE:

- The confirmation condition must be at state 1 for the time delay to be set when the threshold is crossed. If the confirmation condition changes to 0, the time delay in progress ends normally.
- If the time delay is in progress at a new crossing of threshold X1, the time delay is reset with the setpoint value. The output remains at 1.

Operation

Activation type	Description	Illustration
Forward and reverse	The monostable is set or reset in both directions.	
Forward	The monostable is only set in forward movement.	
Reverse	The monostable is only set in reverse.	

Breaking cam

Definition

A Braking type cam is one that changes to 1 when a threshold is crossed and returns to 0 when the same threshold is crossed in the opposite direction.

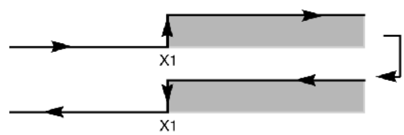
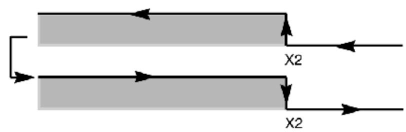
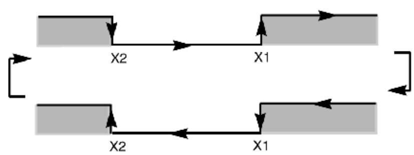
It is characterized by:
- a forward threshold value X1 (value of the angle which sets the brake when the threshold is crossed in the forward direction)
- a reverse threshold value X2 (value of the angle which sets the brake when the threshold is crossed in the reverse direction)
- an activation type: forward/reverse, forward or reverse activation type.

Threshold X1 can be greater than X2 (it is thus possible to set the brake cam either in the cycle or between 2 cycles).

NOTE: The confirmation condition must be at state 1 so that the cam changes to 1 when the threshold is crossed. If the confirmation condition changes to 0, the brake is engaged.

Operation

This table describes the performance of a braking cam.

Activation type	Description	Illustration
Forward	The brake is activated when threshold X1 is crossed in the forward direction. The brake is deactivated when threshold X1 is crossed in the reverse direction.	
Reverse	The brake is activated when threshold X2 is crossed in the reverse direction. The brake is deactivated when threshold X2 is crossed in the forward direction.	
Forward and reverse	The brake is activated when: • Threshold X1 is crossed in the forward direction. • Threshold X2 is crossed in the reverse direction. The brake is deactivated when these thresholds are crossed in the opposite direction	

Parametrizing the confirmation condition associated with a cam

Role

This is used to associate an enable condition with a cam, using an enable bit or via the parts counter.

Enable Bit

A group has 8 control bits which are contained in words ENAB_GROUPi_BITS (where i = group no. 0 to 3) and can be accessed by the PLC program. When an enable bit is associated with a cam, if this bit is at:
- 1, the cam is enabled,
- 0, the logic state of the cam remains at 0 whatever the angular position.

Examples of use:
- controlling the cams in one group by assigning them the same enable bit.
- managing the operating modes

Parts counter

The cam can also be controlled by the state of the parts counter. The cam is only enabled when the parts counter reaches the limit value.

Procedure

The following table describes the operations to be performed:

Step	Action
1	Access the recipe screen for the cam which has to have its parameters entered.
2	Select the enable type in the **Enable condition** box: • **Cam continuously enabled**: no condition is assigned, and the cam is always enabled. • **Conditioned by an enable bit**: a bit is associated with the cam: choose the bit number (from 0 to 7) in the **Enable bit number** field. Example: if bit 5 is chosen for a cam in group 0, it is bit ENAB_GROUP0_BITS.5 which enables the cam when it is at 1. • **When the parts counter is full**.

Confirming the recipe adjustment parameters

Procedure

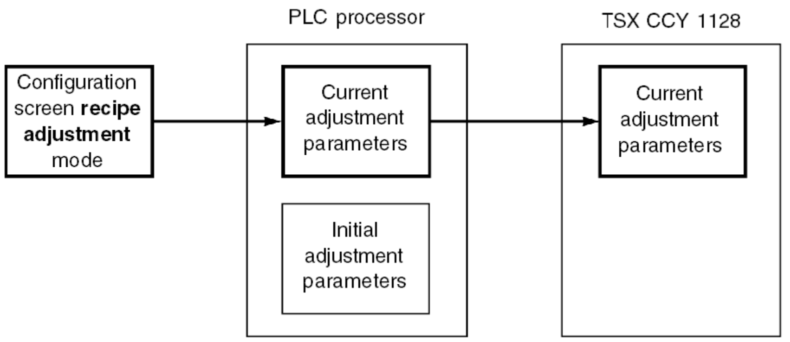
When recipe adjustment parameters are entered, confirm these parameters using the Edit/Confirm command or click on the confirmation icon.

If one or more parameter values are outside the permitted limits, an error message appears indicating the parameter concerned.

Correct the faulty parameter or parameters then confirm.

In connected mode:

The modified parameters are the current parameters (the initial parameters remain unchanged).



WARNING

UNEXPECTED APPLICATION BEHAVIOR - PARAMETERS RESET

On a cold restart the current recipe adjustment parameters are replaced by the initial parameters.

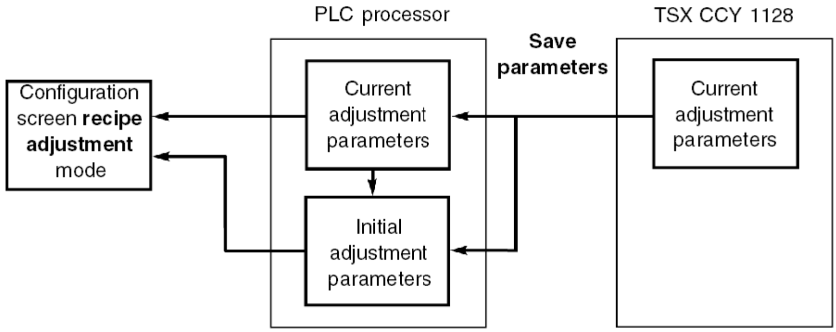
Failure to follow these instructions can result in death, serious injury, or equipment damage.

The initial parameters can be updated by the save command or via a reconfiguration operation.

Saving the recipe adjustment parameters

Procedure

To save the current parameters (updating the initial parameters) select the **Services/Save Parameters command.**



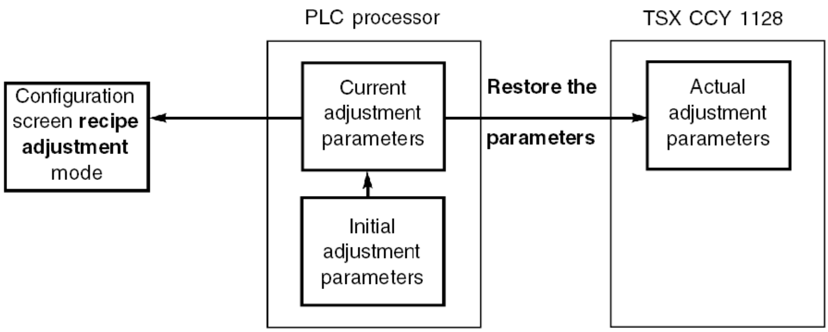
NOTE: The SAVE_PARAM%CHXY.0 instruction enables the application to perform this save operation.

Restoring the recipe adjustment parameters

Procedure

The Services/Restore parameters command replaces the current parameters with the initial values.

This operation causes the cam processor to stop.



NOTE:

- The RESTORE_PARAM %CHXY.0 instruction enables the application to perform this restoration operation.
- This operation is also performed automatically on a cold restart.

Debugging and adjusting

14

Subject of this chapter

This chapter describes the electronic cam module various adjust and debug screens.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Description of the debug screen	242
Description of the main debugging zone	243
Description of the debugging zone: "Acquisition"	244
Description of the debugging zone: "Parts counter"	246
Description of the adjustment zone	247
Description of the debug zone: "Group x"	250
Description of the adjustment screen	252

Description of the debug screen

Accessing the debug screen

On the module screen, select the debug mode by clicking on the corresponding tab.

Debug screen

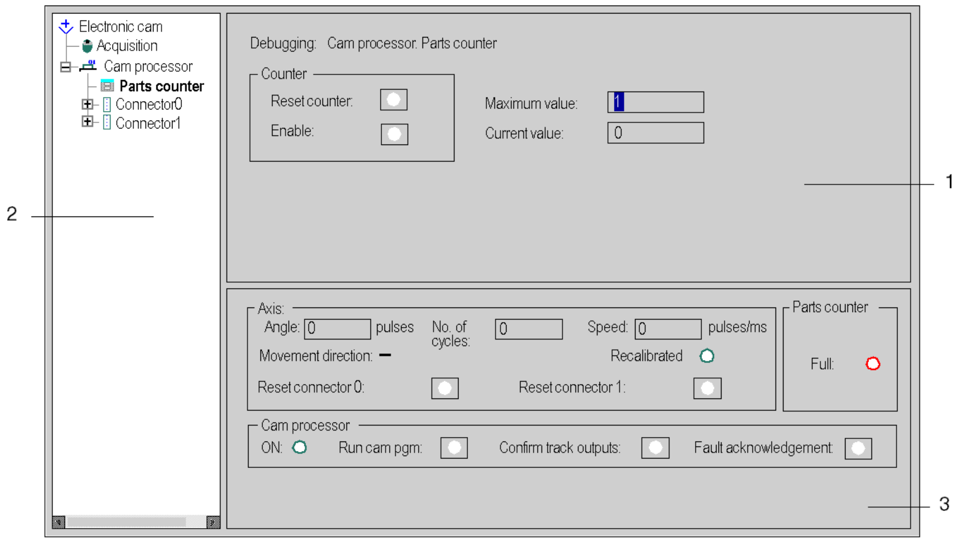
This screen is used to find out:

- the module status, by displaying the various errors which may have arisen
- the present status of the application-specific function,
- forcing the track commands, performing recalibrations, setting the module to RUN or to STOP, and confirming or locking the track action on the outputs.
- the status of the cam processors, locking the outputs,

It can be used to reactivate protected outputs and acknowledge errors.

Description

The part specific to the CCY 1128 of the debug screen is composed of 3 parts:



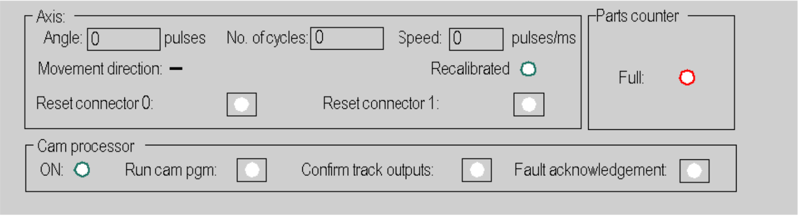
This table describes the different parts:

Address	Description
1	Debug zone for the selected function.
2	Browser zone: used to select a function.
3	Main debug zone: used to access the main objects

Description of the main debugging zone

At a glance

This zone is permanent in the debugging screen. It is used to display the main functions of the axis, and the current status of the cam processor.



Description

The current values of the axis parameters are provided in the axis section, and cannot be modified. Only two buttons are accessible. These are used to reset short-circuit faults occurring on each of the module connectors.

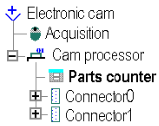
In the cam processor section, it is possible to set the processor to RUN or STOP, to enable/lock track outputs, or to acknowledge faults occurring on these outputs.

Description of the debugging zone: "Acquisition"

Accessing the screen

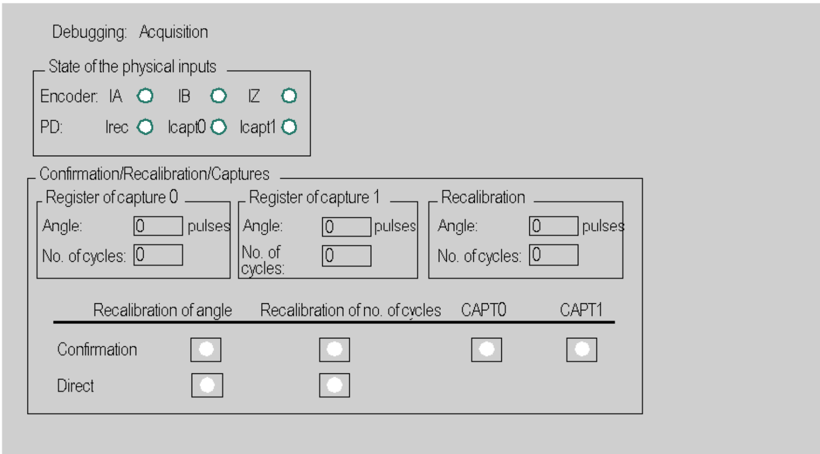
This zone is variable and depends on the selection made in the browser.

Selection of acquisition in the browser:



At a glance

This acquisition zone is used to display the state and activity of the encoder inputs and the recalibration and capture sensors.

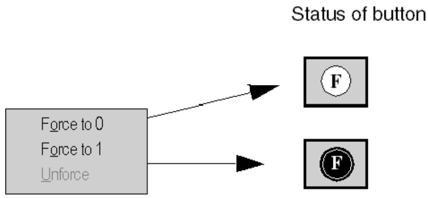


The Enable/Recalibrate/Captures zone indicates the current status of the capture and recalibration registers. The buttons are used to confirm the angle and cycle recalibration conditions. The direct recalibration buttons are used to recalibrate the angle and the number of cycles on the rising edge of the action on the button. The capture confirmation buttons are used to confirm the angle and number of cycles capture conditions.

Operation of the Buttons

"**Left click**", the action is a push/pull type of operation on the %Q object unless it is controlled by the application program.

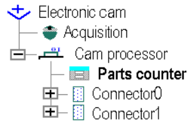
"**Right click**", opens a pull-down menu which is used to force the function to 0 or 1. The Unforce option is used to remove the forcing in progress. The status of the button indicates forcing in progress.



Description of the debugging zone: "Parts counter"

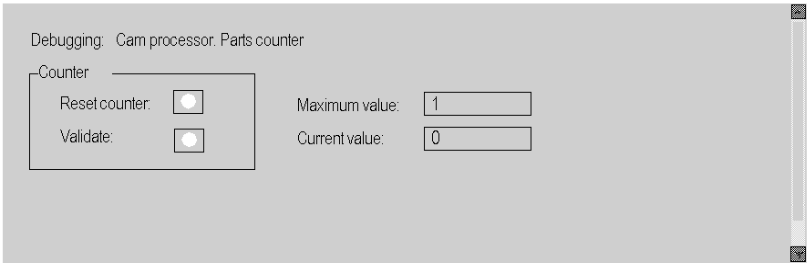
Accessing the screen

The "Parts counter" function is selected in the browser.



At a Glance

"Parts counter" screen zone



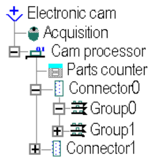
The current value of the parts counter is displayed, together with the maximum value given in the recipe.

An enabling button: **Reset counter** enables the counter to be reset to 0 on the rising edge. An enabling button: **Confirm** is used to confirm the conditions for parts counting. This button confirms conditions permanently when it is pressed (black).

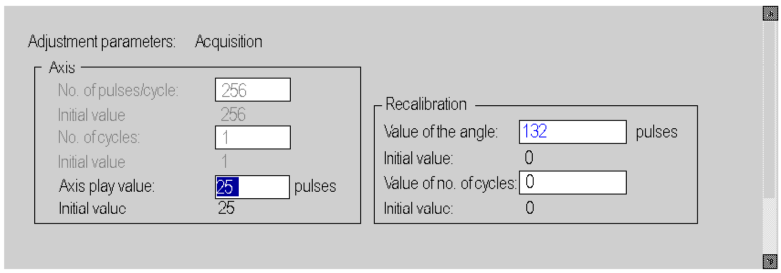
Description of the adjustment zone

Adjustment zone: "Acquisition"

The "Acquisition" function is selected in the browser.

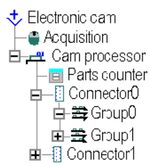


The only values that can be modified are the values for the play of the axis, the value of the angle of recalibration and the number of cycles. In these screens: whatever is displayed in blue can be modified.

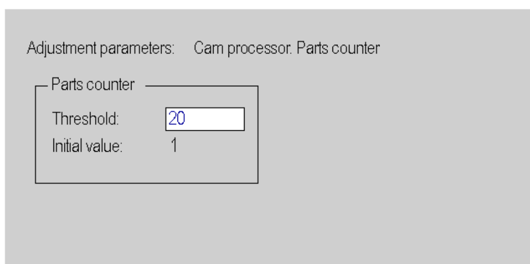


Adjustment zone: "parts counter"

Opening via the browser:

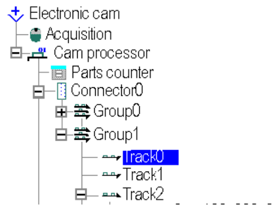


In this screen the maximum value of the parts counter can be modified.

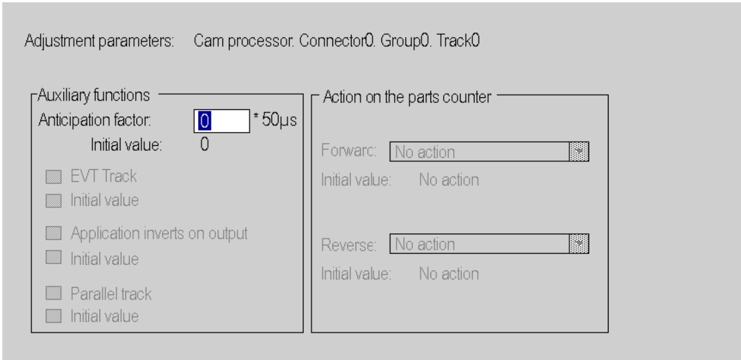


Adjustment zone: ""

Opening via the browser:

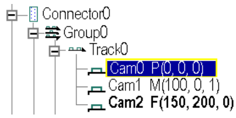


For each selected track, it is possible to assign to it, (modify), a factor of anticipation. The value will be a number of steps of 50 micro seconds.



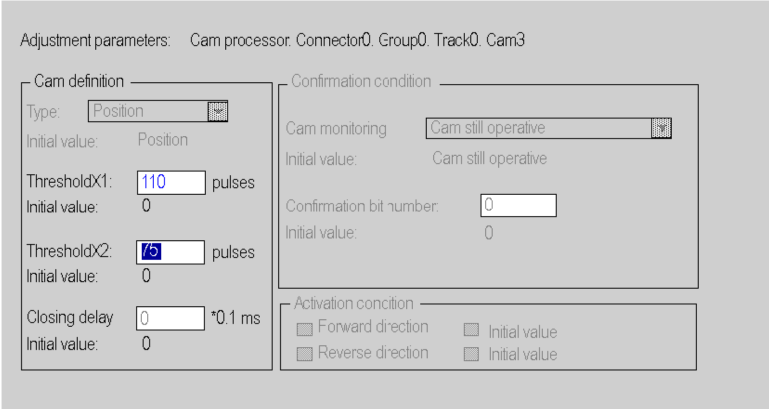
Adjustment zone: ""

Opening via the browser:



It is only possible to open configured cams.

Thresholds X1, X2 and the time delay value can be modified. The initial value is given for information and allows you to return easily to the previous conditions.



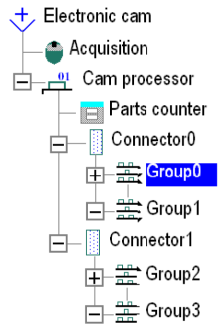
Confirming new parameters

After modifying parameter values during debugging, the user should press the confirm button. A request is then sent to the module. The module acknowledges the modification without stopping the cam processor. The modification is in the current module parameters. Switch to "Adjust Recipe" mode to save them as initial parameters.

Description of the debug zone: "Group x"

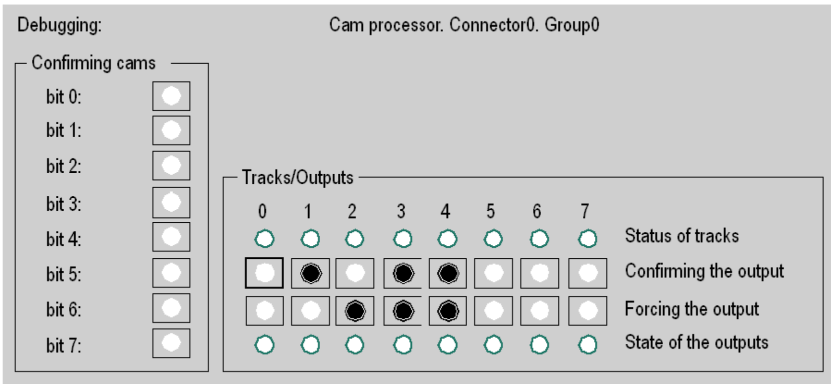
Accessing the screen

The Group x function is selected in the browser.



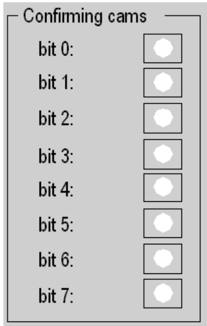
At a Glance

This screen zone is divided into two sections.



Confirming

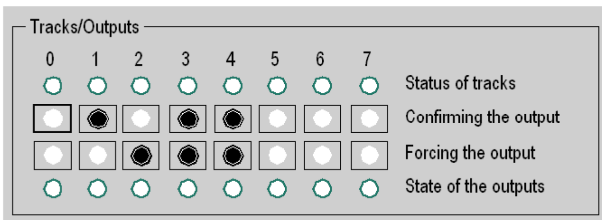
Confirming the cams:



Each cam in a group can be conditioned on the status of one of the group's 8 cam confirmation bits. Periodic command objects can be accessed from this screen. (It is possible to set them to 1 as long as the application program is not controlling them). There is no forcing.

Confirming and outputs

Confirming tracks and outputs:



Correspondence between the status of the tracks and the status of the outputs

Address	Description
1	As long as output confirmation is not set to 1 the output remains at 0.
2	If output confirmation is set to 1, the output takes the status of the track.
3	As long as the forcing bit is set to 1 the output is set to 1.
4	Effect of output inversion linked to configuration.

This zone shows the current status of the tracks. Two rows of buttons are used either to confirm each output individually, or to force each output individually (the button turns black).

The lower row is used to display the actual state of the outputs.

Description of the adjustment screen

At a Glance

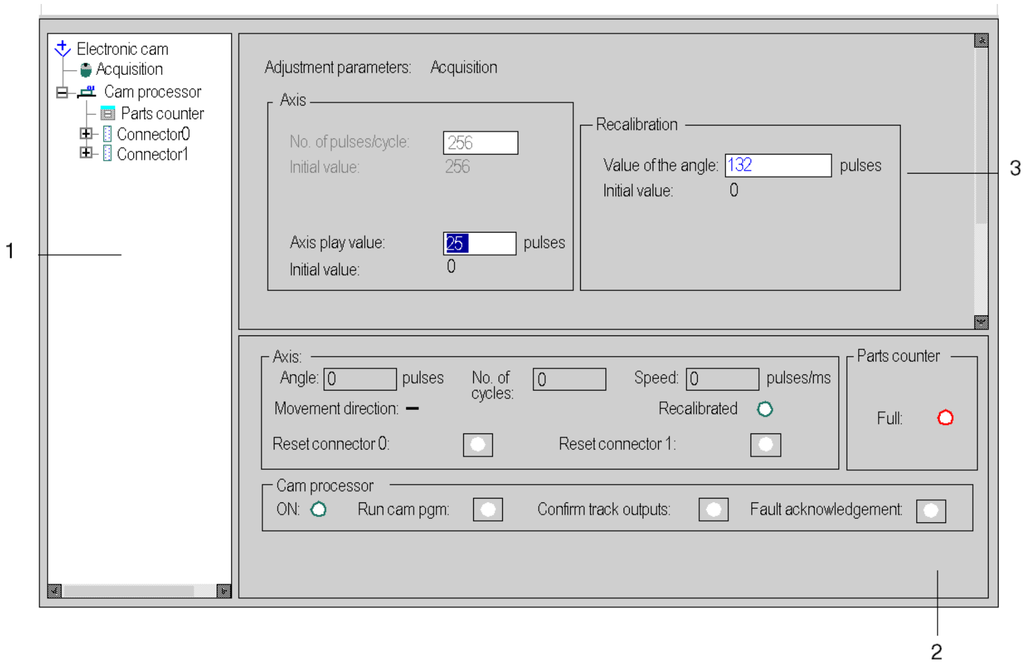
This screen is used to enter and/or modify the parameters of the axis without stopping the processor cam treatment. This mode is used to access a certain number of recipe adjustment parameters.

The layout of this recipe adjustment zone is identical to the "adjust recipe" screen. Only one function can be modified at a time.

The debug zone remains simultaneously in these modes.

Description

Adjustment Screen



This table describes the different zones:

Address	Description
1	Selection zone
2	Main debug zone
3	Adjustment zone (e.g.: acquisition)

Confirming new parameters

After modifying parameter values during debugging, the user should press the confirm button. A request is then sent to the module. The module acknowledges the modification without stopping the cam processor. The modification is in the current module parameters. **Switch to Adjust Recipe mode to save them as initial parameters**.

Diagnostics

15

Object of this chapter

This chapter describes the error codes and relevant status words of the electronic cam

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Module status level	256
Channel status level	257
Error codes	259
Controls	262
Checking the integrity of the module	263
Monitoring the encoder	264
Checking auxiliary inputs	266
Monitoring track outputs	267
Questions and Answers	269

Module status level

Module status objects

These objects are common to all the modules in the range.

Object	Symbol	Meaning
%Ir.m.MOD.ERR	MOD_FAULT	module fault bit
%MWr.m.MOD.2	FAULTY_MOD FAULTY_CH MOD_CNF_FLT MISSING_MOD	bit 0 = 1 internal fault: module failure bit 1 = 1 channel operating fault (see channel status) bit 2 to bit 4 reserved bit 5 = 1configuration fault: difference between the value configured and that read bit 6 = 1 module missing or off bit 7 = reserved

They give information about the module status.

The value `%MWr.m.MOD.1` is updated by an explicit command:

`READ_STS` applied to the module.

Method

If `%Ir.m.Mod.Err` = 1 (implicit), the `READ_STS` command must be executed in order to obtain additional information.

Channel status level

Aperiodic status

This data is used for module diagnostics. Faults may be internal or external to the module.

Object	Symbol	Meaning
%Ir.m.0.ERR	Track_FAULT	group 0 fault bit
%MWr.m.0.0	**EX_STS** STS_IN_PROG ADJ_IN_PROG RECNF_IN_PROG	exchange status management bit 0 = 1 channel status reading in progress bit 1 reserved bit 2 = 1 function: WRITE_PARAM, MOD_PARAM, MOD_TRACK, MOD_CAM, TRF_RECEIPT in progress. bit 3 to bit 14 reserved bit 15 = 1 reconfiguration in progress
%MWr.m.0.1	**EX_RPT** ADJUST_ERR RECONF_ERR	exchange report status bit 0 to bit 1 reserved bit 2 = 1 error on communication function: WRITE_PARAM, MOD_PARAM, MOD_TRACK, MOD_CAM, TRANF_RECEIPT. bit 3 to bit 14 reserved bit 15 = 1 fault in reconfiguration
%MWr.m.0.2	**CH_STS** ENC_FLT AUXIL_FLT TRACK_FLT INTERNAL_FLT CONF_FLT COMMUNIC_FLT APPLI_FLT CH_LED ENC_SUPPLY_FLT ENC_WIRE_FLT ENC_TRANSMIT_FLT AUX_SUPPLY_FLT C0_SUPPLY_FLT C1_SUPPLY_FLT	channel status bit 0 = 1 external fault: Encoder supply fault bit 1 = 1 external fault: auxiliary input power supply fault bit 2 reserved bit 3 = 1 external fault: Track outputs fault bit 4 = 1 internal fault: internal module fault or module performing self-tests bit 5 = 1 module hardware configuration fault or software configuration different from expected bit 6 = 1 communication fault bit 7 = 1 error on a recipe data item bit 8 and bit 9 state of the channel LED bit 10 = 1 encoder power supply fault bit 11 =1 encoder line fault bit 12 = 1 SSI frame transmission fault (parity or format) bit 13 = 1 auxiliary input power supply fault bit 14 = 1 connector 0 power supply fault bit 15 = 1 connector 1 power supply fault
%MWr.m.0.3	**EXTEN0_FLT** COD_PARAM_FLT COD_DESC_FLT ENC_ALARM	channel status (specific) bit 0 to 5 error code for axisconfiguration or adjustment fault. bit 6 to bit 11 error code for track or cam description fault bit 12 = 1 fault sent by the SSI encoder bit 13 to bit 15 reserved

Object	Symbol	Meaning
%MWr.m.0.4	**EXTEN1_FLT** NUM_DESC_FLT NUM_GROUP_FLT COD_LOCAL_FLT	channel status (specific) bit 0 to bit 4 order number of the faulty track or cam bit 5 to bit 6 group number of the faulty track or cam bit 7 to bit 8 reserved bit 9 to bit 15 error code on transfer of recipe data: 16#01 = the address specified does not exist 16#02 = the group specified does not exist 16#03 = the track specified does not exist 16#04 = the cam specified does not exist 16#05 = the action code specified does not exist 16#06 = the type of object does not exist
%MWr.m.0.5		reserved
%MWr.m.0.6	**C0_SHORT_CIRCUIT**	short-circuit fault on connector 0 16#0000 no fault 16#0001 short-circuit on group 0 tracks 16#0100 short-circuit on group 1 tracks 16#0101 short-circuit on group 0 and group 1 tracks
%MWr.m.0.7	**C1_SHORT_CIRCUIT**	short-circuit fault on connector 1 16#0000 no fault 16#0001 short-circuit on group 2 tracks 16#0100 short-circuit on group 3 tracks 16#0101 short-circuit on group 2 and group 3 tracks
%MWr.m.0.8 to %MWr.m.0.11		reserved

The value of the %MW objects is updated by the READ_STS %CHr.m.0 command.

Method

If %Ir.m.Mod.Err = 1 (implicit), perform the READ_STS command.

Error codes

Error codes

In the channel periodic status word %MWr.m.0.3, bits 0 to 5 are used to code the configuration or adjustment errors for the axis, and bits 6 to 11 code the description errors for a track or a cam. A symbol is associated with each error code.

COD_PARAM_FLT: error code for axis configuration or adjustment error

Code	Faulty parameters
0	No error
1	EVT_ENABLE is not 0,1,or 255
2	EVT_NUM is not between 0 and 63
3	INPUT_MOD is not 0 (inc) or 1 (abs)
4	Reserved field is not at 0
5	FORMAT_MEAS is not 0,1 or 2
6	PRESET_MOD is not 0,1,2, or 3
7	CAPTS_MOD is not 0,1,2, or 4
8	Conf: Reserved field is not at 0
9	Conf: Reserved field is not at 0
10	Conf: Reserved field is not at 0
11	WITH_ERR is not zero - in incremental encoder
12	ABS_ENC_ERROR_RANGE is not zero - in incremental encoder
13	ABS_ENC_READ_PERIOD is not zero - in incremental encoder
14	CAPTS_MOD is not zero - in absolute encoder
15	ABS_ENC_READ_RANGE is greater than or the same as the number of status bits
16	ABS_ENC_READ_RANGE is at 0
17	ABS_ENC_READ_PERIOD is not 0,1 or 2 - in absolute encoder
18	ABS_ENC_READ_PERIOD - 50 micro – incompatible with frame length
19	ABS_ENC_READ_PERIOD incompatible with frame length
20	ABS_ENC_READ_EXTRA_NB - no. of heading bits too big (0.4 allowed)
21	ABS_ENC_READ_EXTRA_NB - no. of status bits too large (0.3 allowed)
22	ABS_ENC_EXTRA_NB is at 0 – incompatible with WITH_ERR
23	ABS_ENC_DATA_NB - number of data bits is more than 25
24	ABS_ENC_DATA_NB + ABS_ENC_READ_EXTRA_NB + WITH_PAR is more than 32
25	PRESET_ANG_VALUE is more than RESOL_ANG
26	PRESET_TURN_VALUE is more than RESOL_TURN

Code	Faulty parameters
27	SLACK_VALUE less than -1023
28	SLACK_VALUE more than 1023
29	SLACK_VALUE more than (RESOL_ANGL * RESOL_TURN)
30	SLACK_VALUE more than RESOL_ANG/2
31	ABS_REDUC is at 0
32	ABS_REDUC is not 1,2,4,8,16 or 32
33	(ABS_REDUC*RESOL_ANGL*RESOL_TURN) more than ABS_ENC_DATA_NB
34	ABS_OFFSET_ANG more than RESOL_ANGL
35	ABS_OFFSET_TURN more than RESOL_TURN
36	RESOL_ANGL is not a power of 2 - in absolute encoder
37	Param: Reserved field is not at 0
38	PRESET_ANG_VALUE(adjusted) is more than RES_ANG
39	PRESET_TURN_VALUE(adjusted) is more than RES_TURN
40	SLACK_VALUE (adjusted) less than -1023
41	SLACK_VALUE (adjusted) more than 1023
42	SLACK_VALUE (adjusted) more than (RESOL_ANGL * RESOL_TURN)
43	SLACK_VALUE (adjusted) more than RESOL_ANG/2
44	ABS_ENC_DATA_NB less than 8
45	RESOL_ANGL less than 256
46	INPUT_MOD (encoder) incompatible with FORMAT_MEAS
47	MAX_PIECES is less than 1
48	MAX_PIECES is more than 32767
49	MAX_PIECES (adjusted) is less than 1
50	MAX_PIECES (adjusted) is more than 32767

List of track or cam error codes

COD_DESC_FLT: error code for track or cam description fault

Code	Faulty parameters
0	No error
1	TYP_PROFIL unknown cam code
2	TYP_PROFIL unknown cam code
4	Cam: Reserved field is not at 0
5	Cam: Reserved field is not at 0
6	TRACK_NUM is more than 7
7	COND_ENABLE is not at 0,1,2
8	BIT_NUM_ENABLE impossible
9	Cam: Reserved field is not at 0
10	X1 is more than RESOL_ANG
11	X2 is more than RESOL_ANG
12	TIME_SWICH_OFF is not at 0
13	X2 is not at 0
14	TIME_SWICH_OFF is more than 16383
15	Cam: Reserved field is not at 0
16	Cam: Reserved field is not at 0
32	Track: Reserved field is not at 0
33	Track: Reserved field is not at 0
34	ADD_TRACK on track 4 to 7
35	ANTICIP_FACTOR more than 32767
36	ANTICIP_FACTOR is not at 0 – logical track
48	USED_CAM (adjustment) the cam has not been declared
49	X1(adjusted) is more than RESOL_ANG
50	X2 (adjusted) is more than RESOL_ANG
51	TIME_SWITCH_OFF(adjustment) is not at 0
52	X2 (adjustment) is not at 0
53	TIME_SWICH_OFF (adjusted) is more than 16383
58	USED_TRACK (adjustment) the track has not been declared
59	ANTICIP_FACTOR (adjusted) more than 32767
60	ANTICIP_FACTOR (adjustment) is not at 0 – logical track

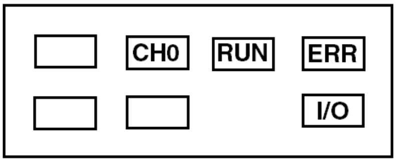
Controls

General

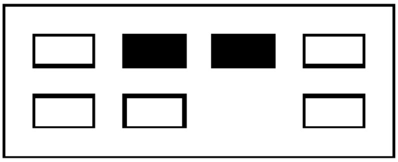
- The system checks that a module really is in good working order and is capable of performing the designated function.
- The module tests its main components.
- Checks correct operation of the internal self-tests performed on cold or warm restart of the internal software.
- Checks communication between the cams module and the processing unit.
- Checks supply and connection of the encoder.
- Checks auxiliary inputs supply.
- Checks track outputs.

Display on module front panel

The display on the module front panel shows the module operating status. The information is laid out in the following arrangement:



On power-up, if CH0 and RUN light up green, the module has not detected any errors and is ready to operate.



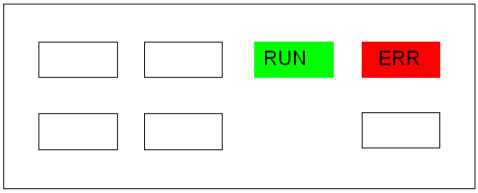
The CH0 light is lit (green)

The RUN light is lit (green)

Checking the integrity of the module

Internal fault

Example



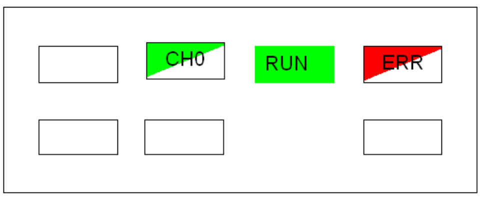
The ERR (red) LED is lit

When the bit `%Ir.m.MOD.ERR` = 1, the `READ_STS` instruction applied to the module is used to refresh the status, and you will find: FAULTY_MOD = 1

The outputs are guaranteed at 0V.

Communication fault with the module

For example, due to a break in the X bus linking the extension rack in which the module is located.



The ERR light flashes (red)

The RUN (green) LED is lit

The CH0 (green) LED flashes

When the bit `%Ir.m.MOD.ERR` = 1, the `READ_STS` instruction applied to the module is used to refresh the status, and you will find: `FAULTY_MOD` = 1

When the bit `%Ir.m.0.ERR` = 1, the `READ_STS` instruction applied to channel 0 is used to refresh the status, and you will find: `COMMUNIC_FLT` = 1 or `INTERNAL_FLT` =1.

The status of the outputs depends on the configuration selected (cam processor configuration).

Monitoring the encoder

Introduction

Conitinous monitoring of the configured encoder is performed.

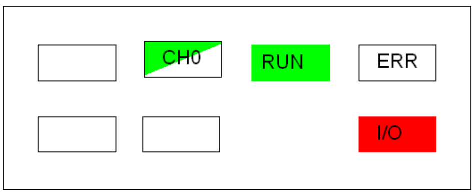
Encoder power supply check: the real voltage applied to the encoder is measured.

Line check: line breaks and short-circuits, if requested during configuration, are detected by measuring the differential voltage on the encoder connection lines.

Transmission check: two checks are made on the SSI link with the encoder. A parity check is performed if requested during configuration. An answer present check is also performed.

Encoder alarm: certain SSI encoders bring up fault data in the series link frame. The module transmits the data, if requested during configuration, to the application software.

Signalling:



The ERR (red) LED is off

The RUN (green) LED is lit

The CH0 (green) LED flashes

The I/O (red) LED is on

Consequences of an encoder fault

If the encoder power supply voltage is too low, if there is a line fault or if a transmission fault is detected:

- Position reading is no longer performed: data item `Ang_Ok` = 0
- The cam processor is set to STOP. data item `Pcam_On` = 0

For an `Alarm_Encoder` fault, the module proceeds with processing as normal, the processor remains in RUN.

The bit `%Ir.m.MOD.ERR` = 1. The `READ_STS` instruction applied to the module is used to refresh the status, and you will find:

- `FAULTY_CH` =1

The bit `%Ir.m.0.ERR` = 1. The `READ_STS` instruction applied to channel 0 is used to refresh the status, and you will find:

- `ENC_FLT` = 1 and
- `Enc_Alarm` = 1 if the data is detected in the SSI frame
- `Enc_Transmit_Flt` = 1 if an SSI frame fault is detected
- `Enc_Wire_Flt` = 1 if a line fault is detected
- `Enc_Supply_Flt` = 1 if the encoder is incorrectly supplied

If encoder power supply fault masking has been configured (supply enc_MSK = 1) then `%IWr.m.MOD.ERR`, `%IWxy.0.ERR`and `Enc_Flt` will not change to 1 on an encoder power supply fault.

Checking auxiliary inputs

Introduction

The 24 V voltage on the auxiliary I/O connector is monitored. A fault occurs if the voltage is less than 19V.

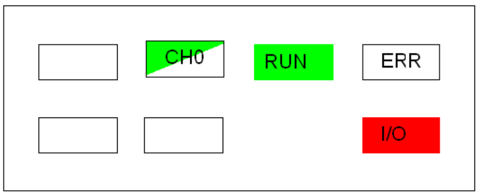
Consequences of the power supply fault

If the 24V power supply is too low (<19V):

- Position reading is no longer performed: data item `Ang_Ok` = 0
- The cam processor is set to STOP. data item `Pcam_on` = 0

Auxiliary input power supply fault

Signalling:



- The ERR (red) LED is off
- The RUN (green) LED is lit
- The CH0 (green) LED flashes
- The I/O (red) LED is on

The bit `%Ir.m.MOD.ERR` = 1. The `READ_STS` instruction applied to the module is used to refresh the status, and you will find:

- `FAULTY_CH` =1

The bit `%Ir.m.0.ERR` = 1. The `READ_STS` instruction applied to channel 0 is used to refresh the status, and you will find:

- `Aux_FLT` = 1 and
- `Aux_Supply` = 1 if the encoder is incorrectly supplied

If auxiliary input power supply fault masking has been configured (`Supply_Aux_MSK`) then: `%IWr.m.MOD.ERR`, `%IWr.m.0.ERR` and `Aux_Flt` will not be set to 1 on an encoder power supply fault.

Monitoring track outputs

Introduction

The 24V voltage is checked on each of the track output connectors. A fault occurs if the voltage is less than 19V. The check is only made if the connector is unlocked.

Each output has a current limiting system (from 0.7 to 2 A). In the event of prolonged overcurrent, thermal tripping occurs.

Consequences of the fault

For a power supply fault on one of the connectors, the cam processoris set to STOP.

For a short-circuit fault on one of the track outputs, all the connector outputs are set to 0V.

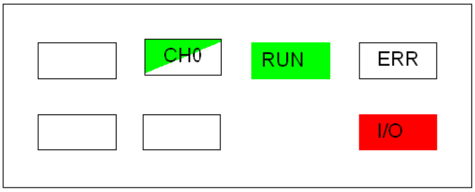
Depending on the chosenconfiguration of the "cam processor" :

- If the processor ignores short-circuit faults (`Cp_ign_sc` = 1) then the processor remains in RUN.
- Otherwise the processor is set to STOP.

If automatic reactivation is required (`Réarm_Mod` =1), reactivation of the faulty connector is automatic after 10s or else the fault must be acknowledged by the command `C0_REARM` or `C1_REARM` according to the connector.

Track output faults

Signalling:



- The ERR light (red) is off.
- The RUN light (green) is on.
- The CH0 light (green) flashes.
- The I/O light (red) is on.

The bit `%Ir.m.MOD.ERR` = 1. The `READ_STS` instruction applied to the module is used to refresh the status, and you will find:

- `FAULTY_CH` =1

The bit `%Ir.m.0.ERR` = 1. The `READ_STS` instruction applied to channel 0 is used to refresh the status, and you will find:

- `Track_FLT` = 1 and
- `C0_Supply_Flt` = 1 if connector 0 is incorrectly supplied
- `C1_Supply_Flt` = 1 if connector 1 is incorrectly supplied
- `C0_Short_Circuit` = 1 if an output in group 0 has short-circuited
- = 256 if an output in group 1 has short-circuited
- `C0_Short_Circuit` = 1 if an output in group 0 has short-circuited
- = 256 if an output in group 1 has short-circuited

If track output supply fault masking has been configured (`Supply_Track_Msk` = 1) then `%IWr.m.MOD.ERR`, `%IWr.m.0.ERR` and `Track_Flt` will not change to 1 on a connector supply fault.

Questions and Answers

List

Table of malfunctions

Malfunction	Possible causes
The cam processor does not switch to RUN mode	• The module has not been recalibrated (bit %Ixy.0.0.i=0). • There is no encoder supply • Lack of auxiliary input supply **NOTE:** If the faults have been masked, a fault can prevent the switch to RUN mode without appearing!
The cam processor changes to stop mode without modifying bit %Qxy.0.5 (PCAM_START_STOP)	Loading a recipe using the commands • WRITE_PARAM • RESTORE_PARAM • TRF_RECIPE
No action on the outputs	• The outputs are already valid OUTS_ENABLE • Parallelization of a track without configured cam • Forced outputs • Inverted outputs
Loss of modifications to the current recipe	When switching to local mode, if the save has not been done when adjusting the recipe
Following modification to the configuration, the browser turns red and it is not possible to confirm the configuration	A parameter is no longer compatible with the new configuration • Switch to recipe adjustment • Click on No when requested to confirm • Modify the parameter • Confirm

Performance and limitations

16

Object of this chapter

This chapter describes the performances and limits of the electronic cam module.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Global precision in controlling the actuators	272
Actuator commands	275
General time-division performance characteristics	279
Functional limitations	281

Global precision in controlling the actuators

General

Precision in controlling actuators in relation to the target mechanical position is linked to three factors: the resolution of the encoder, the speed and the reaction time of the module.

The resolution of the encoder

For a target precision of 0.1mm, the encoder should produce at least one or two pulses for 0.1mm.

NOTE:

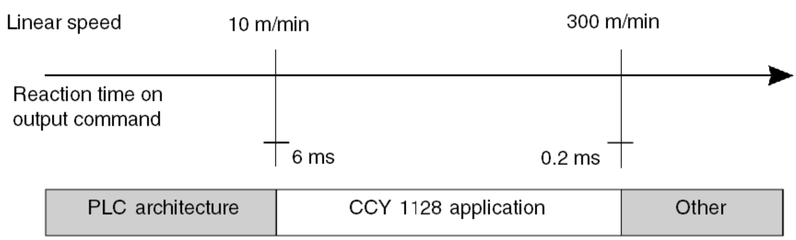
Depending on the configuration selected, the resolution must be taken into account:
- after multiplying by 4 for an incremental encoder
- after reducing the resolution for an absolute encoder

Speed

For the required position, and depending on the approach speed, the system should have a maximum reaction time.

Example: reaction time required for a target precision of 1mm.

Using the cam module in an application is justified for speeds between 10 and 300m/min.

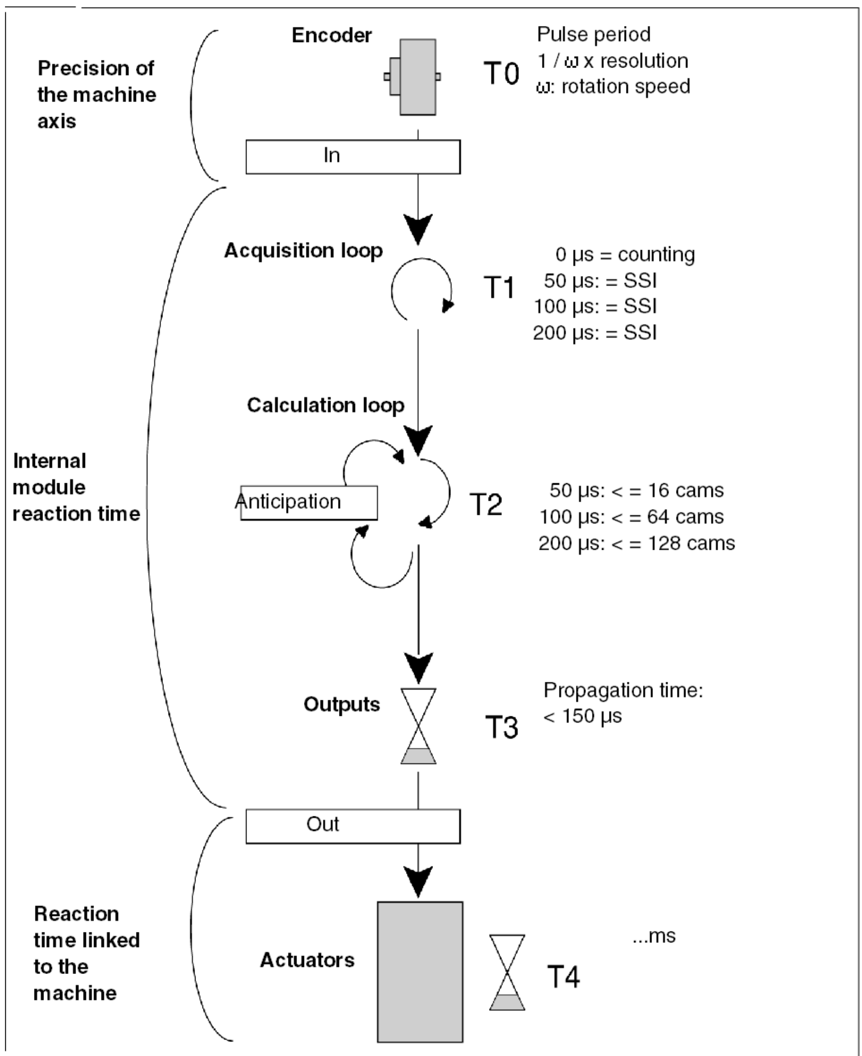


For a movement speed of<10 m/min, the entire application could be handled by a TSX PLC.

The minimum reaction time of the CCY 1128 module is 0.2ms, the maximum movement speed is therefore 300m/min.

Structural diagram of cam processing

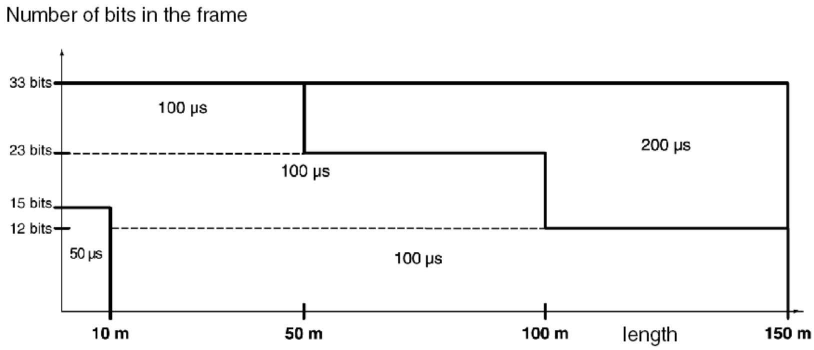
The TSX CCY 1128 cam module optimizes its reaction time for controlling outputs depending on the parameters configured for the application.



loop

The duration of the acquisition loop corresponds to the cycle for reading the absolute encoder via its SSI link. It is always synchronous with the calculation loop. The reading period is configurable (50 micro seconds, 100 micro seconds, 200 micro seconds).

The value selected must be dependent on the length of the SSI frame (number of bits), and the distance (l) between the encoder and the module.



The output refresh period is fixed by the calculation loop:

With the incremental encoder, it is automatically fixed by the module depending on the number of cams used by the cam program (50 micro seconds up to 16 cams, 100 micro seconds up to 32 cams, 200 micro seconds up to 128 cams).

Calculation loop

For an SSI encoder, the calculation period is optimized according to the number of cams configured, but cannot be less than the reading period.

For example: if T1 = 100 micro seconds then T2 = 100 micro seconds for a configuration of between 1 and 32 cams, T2 = 200 micro seconds for a configuration of between 33 and 128 cams.

function

This is used to compensate for the fixed delay introduced by the actuator and its associated mechanics. The calculation loop ensures that the anticipation value for switching each output is updated.

Actuator commands

Estimating precision for controlling actuators

The global communication delay of an output in relation to the crossing of a mechanical threshold depends on the application. It may be divided into two parts:

- **Minimum delay:** is a constant part corresponding to the time necessary to set the outputs.
- **Max/min delay:** a variable part corresponding to the "jitter" brought by the periodicity of the refreshing of outputs. Internal module synchronization is such that the variable part is reduced to the influence of T0 (period between two encoder pulses) and of T2 (calculation period).

Minimum delay (1)

in incremental encoder	= T2+T3
in absolute encoder	= T1 + T2 + T3

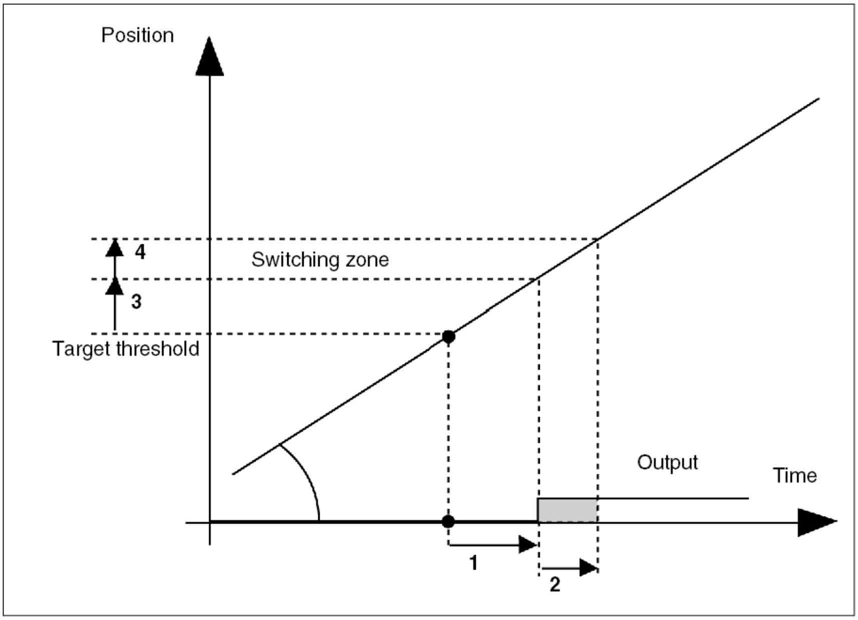
Maximum / minimum delta (2)

without anticipation	= T0 + T2
with anticipation of the SSI	= 2 x (T0 = T2)
with anticipation of the incremental	= 2 x T0 = T2

Precision in output control

The switching delay on outputs causes an error in relation to the target threshold.

Illustration of the use of the without anticipation function

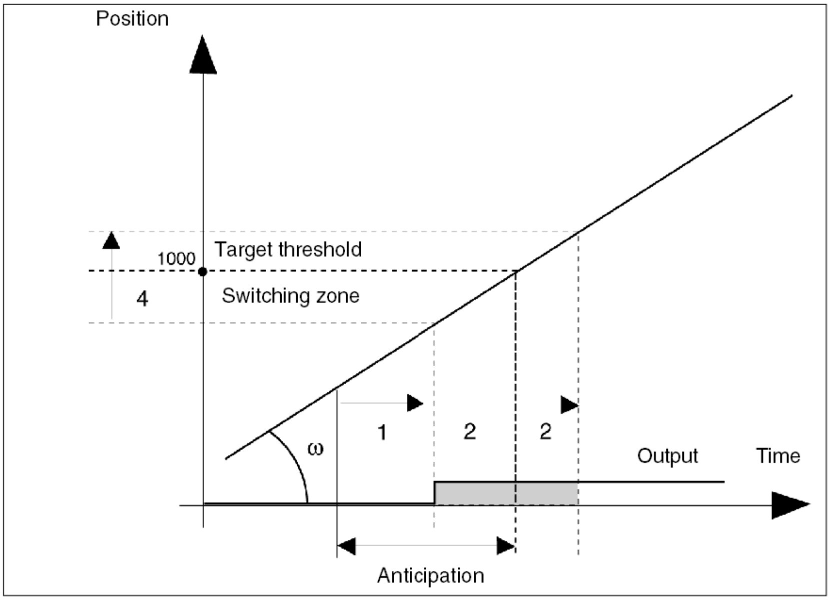


The switching zone deviates from the target threshold proportionally to the speed.

Switching can be reset around the threshold using the anticipation system. To do this, simply add to the anticipation value = minimum delay (**1** + 1/2 delta delay (**2**).

1	Minimum delay
2	Maximum / Minimum delay
3	Fixed deviation
4	Delta switching

Illustration of the use of the with anticipation function



NOTE: To make a valid correction to the delay caused by the module, the encoder resolution should be 2 to 5 times more precise than that which was estimated at the first attempt.

Calculation chart

The tables below show the foreseeable errors depending on the application type and for a reference speed of 120m/min. Values for other speeds can be obtained by a simple rule of three.

Fixed deviation for a speed of 120 m/mn (encoder resolution = 0.1 mm)

Calculation	No. of cams	incremental	SSI 50 (1)	SSI 100 (1)	SSI 200 (1)
50 (1)	up to 16	0.4 mm	0.5 mm	-	-
100 (1)	up to 32	0.5 mm	0.6 mm	0.7 mm	-
200 (1)	up to 128	0.7 mm	0.8 mm	0.9 mm	1.1 mm

Delta switching for a speed of 120 m/min without anticipation (encoder resolution = 0.1 mm)

Calculation	No. of cams	incremental	SSI 50 (1)	SSI 100 (1)	SSI 200 (1)
50 (1)	up to 16	0.2 mm	0.2 mm	-	-
100 (1)	up to 32	0.3 mm	0.3 mm	0.3 mm	-
200 (1)	up to 128	0.5 mm	0.5 mm	0.5 mm	0.5 mm

1 Values are expressed in micro seconds.

⚠ WARNING

UNEXPECTED APPLICATION BEHAVIOR - DELTA SWITCHING VARIATION

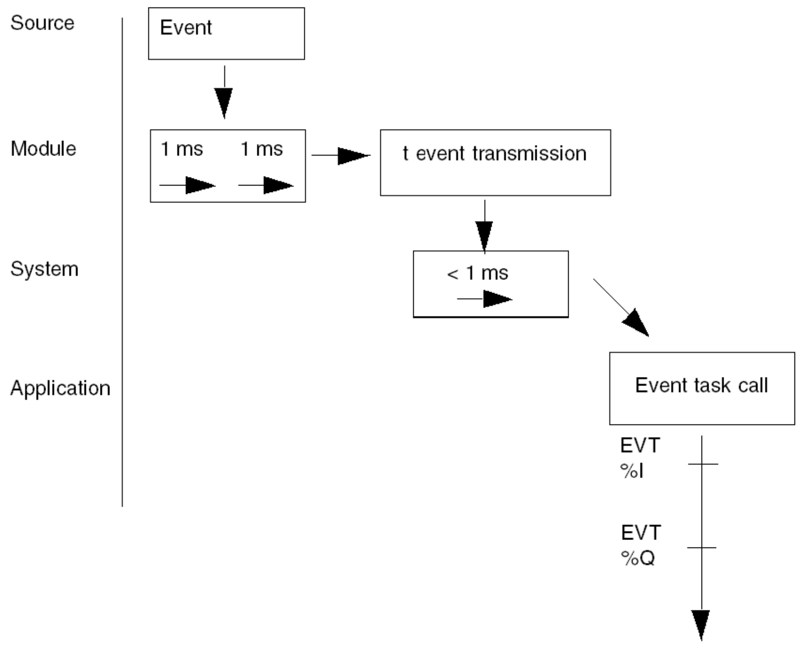
The variation in communication (Delta switching) is increased if the anticipation system is used.

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

General time-division performance characteristics

Calling an event task

Diagram of an event task call



The module includes 7 event sources (capture, cycle switching, cam ....).

The calling of event tasks is limited to a maximum of 1 event per ms.

Only one type of event is sent at a time to the system.

In the case of one event

The start of the event task is executed no more than 3 ms after the real event (e.g.: switch to cycle modulo).

In the case of several simultaneous events

The module integrates a buffer that allows the storing up to 7 events awaiting transmission to the system. The events will be sent in their order of arrival (1 per ms). This, therefore, increases the reaction time.

Table

Description of the functions

Function	Comments	Value
Counter	Permitted frequency	500Khz in x1 250Khz in x4
	Activation of the event task on switching of cycle	< 3 ms
Recalibration	Recalibration of counter on Z marker pulse	< 1 micro second
	Recalibration of counter on Irec	< 50 micro seconds
	Event task activation	< 3 ms
Cam function	Refreshing output	50 micro seconds up to 16 cams 100 micro seconds up to 32 cams 200 micro seconds up to 128 cams
	Updating the correction values (anticipation)	< 4ms
	Activating the event task (cam, parts counter)	< 3 ms
Implicit interface	Influence of the module on the processor cycle time	
	Refreshing %I and %IW	< 1 ms
	Recognizing %Q and %QW	< 1 ms
Explicit interface	WRITE_PARAM	300 ms
	SAVE_PARAM	300 ms
	RESTORE	300 ms
	READ_STS	immediate (1)
	MOD_PARAM	send: 20 ms (2)
	MOD_CAM	send: 20 ms (2)
	MOD_TRACK	send: 20 ms (2)
	TRF_RECIPE	300 ms
	DETAIL_OBJECT	immediate (1)
Additional cycle time	The cycle time has no effect on the output reaction time	1 ms

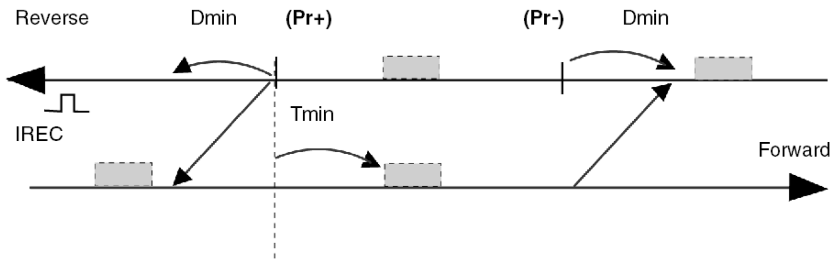
(1) There is no access to the module / Execution time is included in the task execution.

(2) For GET and READ it is immediate as in (1).

Functional limitations

Correction of axis play

Example



The application for correcting axis play allows you to to obtain a positioning of the cams on a mechanical position whatever the direction.

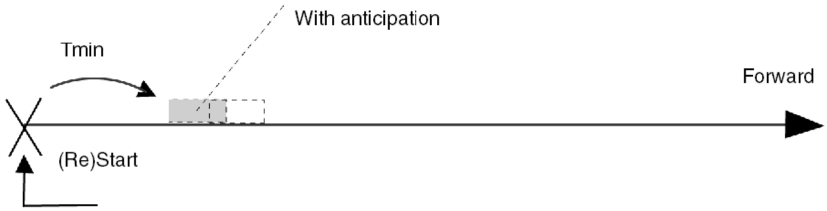
In the example, recalibration is done in a backwards direction. The actual mechanical position is the value delivered by the encoder in the backward direction and the corrected value in the forward direction (end).

A safety distance (Dmin) is necessary between the position of the cames and the points of return. (Pr+ and Pr-). This distance corresponds to the value of axis play that is given when the adjusting the track.

In addition, the actual application of the correction (in forward direction in the example) is effective 4 ms after the change in direction. Also the cams must be positioned in such a way that the direction of development is correctly established: 4 ms (Tmin) before the switch to the first cam.

Anticipation

Example with anticipation



Cam switching is anticipated (in time) in relation to the actual crossing of thresholds. The value is fixed by the anticipation factor (T anticip = n x 50 microseconds). Following a (re)start or a change of direction, there is a delay (Tmin) until anticipation is applied. For correct operation, the first cam switch should not be expected before this delay.

T min = 2 x (T anticip + 4 ms)

The delay required for the axis to reach speed can be added to t min to increase precision.

The language objects of the independent axes application

17

Aim of the section

This section describes the language objects associated with the axis application as well as the various ways of using them.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Description of the language objects of the axis application-specific function	284
Implicit Exchange Language Objects Associated with the Application-Specific Function	285
Explicit Exchange Language Objects Associated with the Application-Specific Function	286
Management of Exchanges and Reports with Explicit Objects	288
Details of the Language Objects of the T_GEN_MOD-Type IODDT	292
Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP0	293
Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP0	297
Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3	306
Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3	307
Configuration constants	311

Description of the language objects of the axis application-specific function

General

The IODDT are predefined by the manufacturer; they contain input/output language objects belonging to the channel of an application-specific module.

The counting modules have various associated IODDTs.

There are two IODDT types for the electronic cam application:

- `T_CCY_GROUP0` which applies to channel 0 of the TSX CCY 1128 module
- `T_CCY_GROUP1_2_3` which applies to channels 1, 2 or 3 of the TSX CCY 1128

NOTE: IODDT variables can be created in two different ways:

- Using the **I/O objects** *(see Unity Pro, Operating Modes)* tab,
- Data Editor *(see Unity Pro, Operating Modes)*.

Language object types

Each IODDT contains a set of language objects enabling the IODDTs to be controlled and their operation checked.

There are two types of language object:

- **implicit exchange objects**, which are automatically exchanged on each cycle turn of the task associated with the module,
- **explicit exchange objects**, which are exchanged at the request of the application, using the explicit exchange instructions.

Implicit exchanges concern the inputs/outputs of the module: measurement results, information and commands.

Explicit exchanges are used to set parameters for the module and for diagnostic purposes.

Implicit Exchange Language Objects Associated with the Application-Specific Function

At a Glance

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

Reminders

The module inputs (`%I` and `%IW`) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

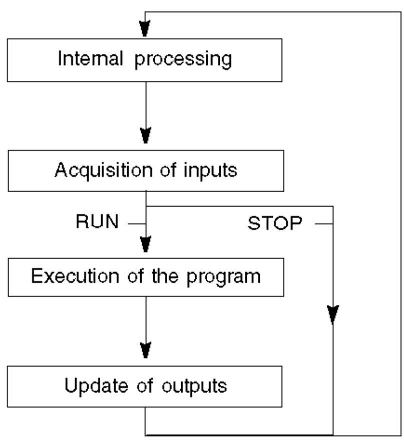
The outputs (`%Q` and `%QW`) are updated at the end of the task, only when the PLC is in RUN mode.

NOTE: When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- outputs are set to fallback position (fallback mode)
- outputs are maintained at their last value (maintain mode)

Figure

The following diagram shows the operating cycle of a PLC task (cyclical execution).



Explicit Exchange Language Objects Associated with the Application-Specific Function

Introduction

Explicit exchanges are performed at the user program's request using these instructions:

- READ_STS *(see Unity Pro, I/O Management, Block Library)* (read status words)
- WRITE_CMD *(see Unity Pro, I/O Management, Block Library)* (write command words)
- WRITE_PARAM *(see Unity Pro, I/O Management, Block Library)* (write adjustment parameters)
- READ_PARAM *(see Unity Pro, I/O Management, Block Library)* (read adjustment parameters)
- SAVE_PARAM *(see Unity Pro, I/O Management, Block Library)* (save adjustment parameters)
- RESTORE_PARAM *(see Unity Pro, I/O Management, Block Library)* (restore adjustment parameters)

These exchanges apply to a set of %MW objects of the same type (status, commands or parameters) that belong to a channel.

These objects can:

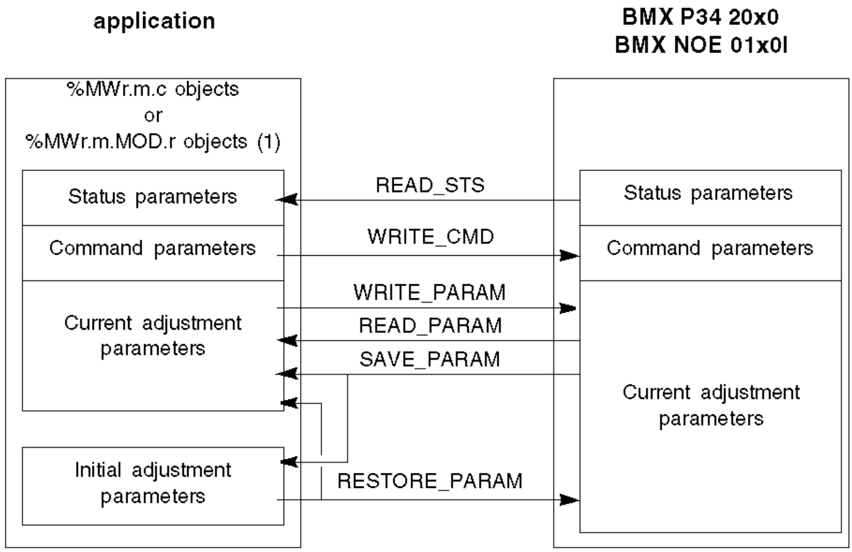
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

NOTE: To avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

NOTE: Explicit Exchanges are not supported when Modicon M340 Analog and Digital I/O modules are configured behind a M340 Ethernet Remote I/O adapter module in a Quantum EIO Ethernet Configuration. As a consequence, it is not possible to setup a module's parameters from the PLC application during operation.

General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the application and module.



(1) Only with READ_STS and WRITE_CMD instructions.

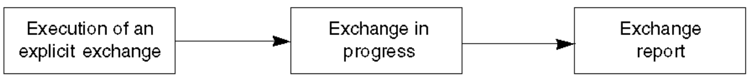
Managing Exchanges

During an explicit exchange, check performance to see that the data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress *(see page 290)*
- the exchange report *(see page 291)*

The following diagram describes the management principle for an exchange.



NOTE: In order to avoid several simultaneous explicit exchanges for the same channel, it is necessary to test the value of the word EXCH_STS (`%MWr.m.c.0`) of the IODDT associated to the channel before calling any EF addressing this channel.

Management of Exchanges and Reports with Explicit Objects

At a Glance

When data is exchanged between the PLC memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

- `EXCH_STS` (`%MWr.m.c.0`): exchange in progress
- `EXCH_RPT` (`%MWr.m.c.1`): report

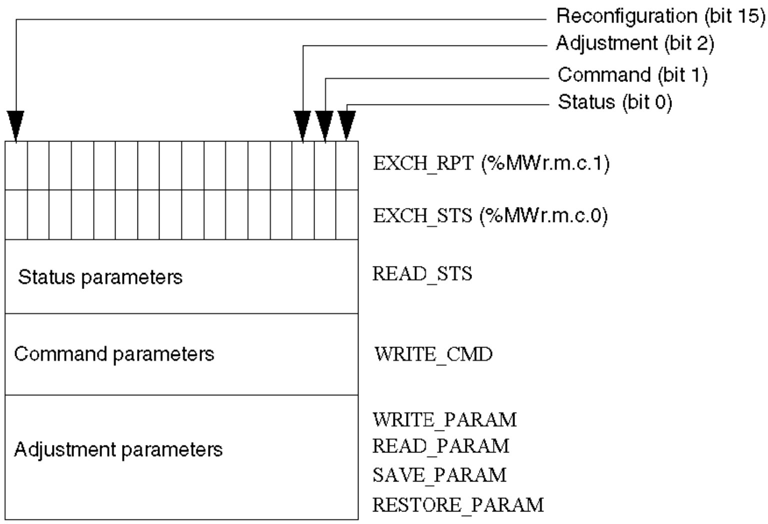
NOTE:

Depending on the localization of the module, the management of the explicit exchanges (`%MW0.0.MOD.0.0` for example) will not be detected by the application:

- For in-rack modules, explicit exchanges are done immediately on the local PLC Bus and are finished before the end of the execution task. So, the `READ_STS`, for example, is always finished when the `%MW0.0.mod.0.0` bit is checked by the application.
- For remote bus (Fipio for example), explicit exchanges are not synchronous with the execution task, so the detection is possible by the application.

Illustration

The illustration below shows the different significant bits for managing exchanges:



Description of Significant Bits

Each bit of the words EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) is associated with a type of parameter:

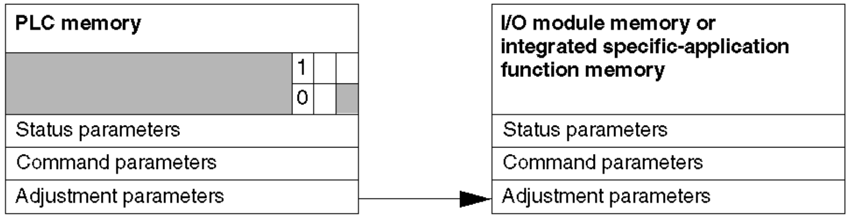
- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - The CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel.
  - The CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - The ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM).
  - The ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- Rank 15 bits indicate a reconfiguration on channel **c** of the module from the console (modification of the configuration parameters + cold start-up of the channel).
- The *r*, *m* and *c* bits indicates the following elements:
  - the **r** bit represents the rack number.
  - The **m** bit represents the position of the module in the rack.
  - The **c** bit represents the channel number in the module.

NOTE: **r** represents the rack number, **m** the position of the module in the rack, while **c** represents the channel number in the module.

NOTE: Exchange and report words also exist at module level EXCH_STS (%MWr.m.MOD) and EXCH_RPT (%MWr.m.MOD.1) as per IODDT type T_GEN_MOD.

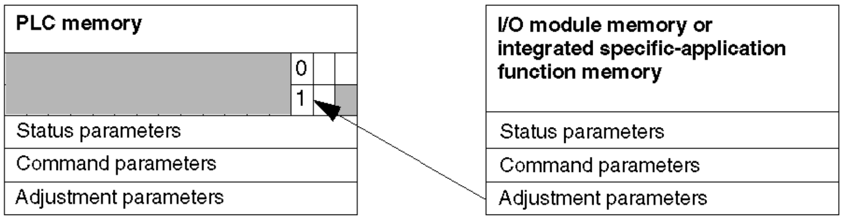
Example

Phase 1: Sending data by using the `WRITE_PARAM` instruction



When the instruction is scanned by the PLC processor, the **Exchange in progress** bit is set to 1 in `%MWr.m.c`.

Phase 2: Analysis of the data by the I/O module and report.



When the data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the `ADJ_ERR` bit (`%MWr.m.c.1.2`).

This bit makes the following reports:
- **0:** correct exchange
- **1:** faulty exchange)

NOTE: There is no adjustment parameter at module level.

Execution Indicators for an Explicit Exchange: EXCH_STS

The table below shows the control bits of the explicit exchanges: `EXCH_STS` (`%MWr.m.c.0`)

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1

Standard symbol	Type	Access	Meaning	Address
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration of the module in progress	%MWr.m.c.0.15

NOTE: If the module is not present or is disconnected, explicit exchange objects (`READ_STS` for example) are not sent to the module (`STS_IN_PROG` (%MWr.m.c.0.0) = 0), but the words are refreshed.

Explicit Exchange Report: EXCH_RPT

The table below shows the report bits: `EXCH_RPT` (`%MWr.m.c.1`)

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error reading channel status words (1 = failure)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error during a command parameter exchange (1 = failure)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error during an adjust parameter exchange (1 = failure)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error during reconfiguration of the channel (1 = failure)	%MWr.m.c.1.15

Counting Module Use

The following table describes the steps realised between a Couting Module and the system after a power-on.

Step	Action
1	Power on.
2	The system sends the configuration parameters.
3	The system sends the adjust parameters by WRITE_PARAM method. **Note:** When the operation is finished, the bit %MWr.m.c.0.2 switches to 0.

If, in the begining of your application, you use a WRITE_PARAM command, you must wait until the bit %MWr.m.c.0.2 switches to 0.

Details of the Language Objects of the T_GEN_MOD-Type IODDT

At a Glance

All the modules of Premium PLCs have an associated IODDT of type `T_GEN_MOD`.

Observations

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

List of Objects

The table below presents the objects of the IODDT:

Standard symbol	Type	Access	Meaning	Address
MOD_ERROR	BOOL	R	Module error bit	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Module exchange control word.	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Reading of status words of the module in progress.	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Exchange report word.	%MWr.m.MOD.1
STS_ERR	BOOL	R	Fault when reading module status words.	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Internal error word of the module.	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Internal error, module failure.	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Faulty channel(s).	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block fault.	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration fault.	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module missing or inoperative.	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Internal error word of the module (Fipio extension only).	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Internal fault, module unserviceable (Fipio extension only).	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Faulty channel(s) (Fipio extension only).	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block fault (Fipio extension only).	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration fault (Fipio extension only).	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module missing or inoperative (Fipio extension only).	%MWr.m.MOD.2.14

Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP0

At a Glance

The IODDT of type `T_CCY_GROUP0` has implicit exchange objects, which are described below. This type of IODDT applies to the `TSX CCY 1128` module for channel 0.

Example of declaring a variable: `IODDT_VAR1` of type `T_CCY_GROUP0`

Generally speaking, the meaning of the bits is given for state 1 of this bit. In specific cases, the two states of the bit are explained.

List of implicit exchange discrete input objects

The table below shows the implicit exchange discrete input objects of the IODDT of type `T_CCY_GROUP0` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Access	Meaning	Number
TRACK_FAULT	EBOOL	R	Channel error bit.	%Ir.m.c.ERR
ANG_OK	EBOOL	R	=1 : Valid angle reading	%Ir.m.0.0
DIRECTION	EBOOL	R	= 0 : reverse motion (-) = 1 : forward motion (+)	%Ir.m.0.2
PCAM_ON	EBOOL	R	= 0 : cam processor on STOP = 1 : cam processor on RUN	%Ir.m.0.3
PIECES_FULL	EBOOL	R	= 1 : parts counter at the limit value	%Ir.m.0.4
IREC_STAT	EBOOL	R	state of physical input IREC	%Ir.m.0.8
ICAPT0_STAT	EBOOL	R	state of physical input ICAPT0	%Ir.m.0.10
ICAPT1_STAT	EBOOL	R	state of physical input ICAPT1	%Ir.m.0.11
IA_STAT	EBOOL	R	state of encoder input IA	%Ir.m.0.12
IB_STAT	EBOOL	R	state of encoder input IB	%Ir.m.0.13
IZ_STAT	EBOOL	R	state of encoder input IZ	%Ir.m.0.14

List of implicit exchange discrete output objects

The table below shows the implicit exchange discrete output objects of the IODDT of type `T_CCY_GROUP0` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Ac-cess	Meaning	Number
PRESET_ANG_ENABLE	EBOOL	RW	= 1 : enabling of the recalibration function on the angle value only	%Qr.m.0.0
PRESET_ALL_ENABLE	EBOOL	RW	= 1 : enabling of the recalibration function on the angle and cycle value	%Qr.m.0.1
CAPT0_ENABLE	EBOOL	RW	= 1 : enabling capture 0	%Qr.m.0.2
CAPT1_ENABLE	EBOOL	RW	= 1 : enabling capture 1	%Qr.m.0.3
PCAM_START_STOP	EBOOL	RW	start command for cam processor active on rising edge stop command for cam processor active on falling edge	%Qr.m.0.5
PIECES_ENABLE	EBOOL	RW	= 1 : enabling the parts counter function	%Qr.m.0.7
EVT_ANG_ENABLE	EBOOL	RW	= 1 : event source when switching to enabled angle rollover	%Qr.m.0.8
EVT_TURN_ENABLE	EBOOL	RW	= 1 : event source when switching to enabled cycle rollover	%Qr.m.0.9
EVT_PRESET_ENABLE	EBOOL	RW	= 1 : event source for enabled preselection	%Qr.m.0.10
EVT_CAPT0_ENABLE	EBOOL	RW	= 1 : capture 0 enabled event source	%Qr.m.0.11
EVT_CAPT1_ENABLE	EBOOL	RW	= 1 : capture 1 enabled event source	%Qr.m.0.12
EVT_CAM_ENABLE	EBOOL	RW	= 1 : event source for enabled cam program	%Qr.m.0.13
EVT_PIECES_FULL_ENABLE	EBOOL	RW	= 1 : event source for enabled parts counter limit value reached	%Qr.m.0.14
ACK_FLT	EBOOL	RW	= 1 : acknowledgement of faults present	%Qr.m.0.15
PRESET_ANG_FORCE	EBOOL	RW	= 1 : (re)calibrating the angle value	%Qr.m.0.21
PRESET_ANG_TURN_FOR	EBOOL	RW	=1 : (re)calibrating the angle and cycle values	%Qr.m.0.22
PIECES_RESET	EBOOL	RW	= 1 : parts counter reset to zero	%Qr.m.0.23
OUTS_ENABLE	EBOOL	RW	= 0 : the outputs are kept at rest (according to the configuration of the outputs) bit 25 = 1 global enabling of outputs	%Qr.m.0.25
C0_REARM	EBOOL	RW	resetting of connector 0 of groups 0 and 1 active on rising edge	%Qr.m.0.32
C1_REARM	EBOOL	RW	resetting of connector 1 of groups 2 and 3 active on rising edge	%Qr.m.0.33

List of implicit exchange %IW input objects

The table below shows the implicit exchange %IW input objects of the IODDT of type `T_CCY_GROUP0` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Access	Meaning	Number
GROUP0_TRACKS	INT	R	bit 0 to bit 7: state of group 0 tracks	%IWr.m.0.0
ANG_VALUE	INT	R	bit 0 to bit 15: actual value of the position angle	%IWr.m.0.1
TURN_VALUE	INT	R	bit 0 to bit 15: actual value of number of cycles	%IWr.m.0.2
SPEED	INT	R	bit 0 to bit 15: speed value	%IWr.m.0.3
PIECES_VALUE	INT	R	bit 0 to bit 15: value of the parts counter	%IWr.m.0.4
CAPT0_ANG	INT	R	bit 0 to bit 15: value of capture 0 register (angle)	%IWr.m.0.5
CAPT0_TURN	INT	R	bit 0 to bit 15: value of capture 0 register (cycle)	%IWr.m.0.6
CAPT1_ANG	INT	R	bit 0 to bit 15: value of capture 1 register (angle)	%IWr.m.0.7
CAPT1_TURN	INT	R	bit 0 to bit 15: value of capture 1 register (cycle)	%IWr.m.0.8
OUTS_C0	INT	R	bit 0 to bit 15: state of connector 0 outputs	%IWr.m.0.9
OUTS_C1	INT	R	bit 0 to bit 15: state of connector 1 outputs	%IWr.m.0.10
EVENTS EVT_ANG EVT_TURN EVT_PRESET EVT_CAPT0 EVT_CAPT1 EVT_CAM EVT_PIECES_FULL DIRECTION_EVT OVERRUN_EVT	INT	R	event register bit 0 = 1: event sent out at each rollover of the angle value bit 1 = 1: event sent out at each rollover of the cycle value bit 2 = 1: event sent out on recalibration bit 3 = 1: event sent out on capture 0 bit 4 = 1: event sent out on capture 1 bit 5 = 1: event sent out by the cam program bit 6 = 1: event sent out when the parts counter reaches the limit value bit 7 to bit 13 reserved bit 14 = 1: forward direction of rotation on event: EVT_TURN or EVT_ANG bit 14 = 0: reverse direction of rotation on event: bit 15 = 1: events overrun	%IWr.m.0.12
CAME_EVT	INT	R	bit 0 to bit 4 = cam number bit 5 to bit 6 = group number bit 7 to bit 15 reserved	%IWr.m.0.13
CAPT_ANG_EVT	INT	R	bit 0 to bit 15 captured value of the angle	%IWr.m.0.14
CAPT_TURN_EVT	INT	R	bit 0 to bit 15 captured value of the number of cycles	%IWr.m.0.15

List of implicit exchange %QW output objects

The table below shows the implicit exchange %QW output objects of the IODDT of type `T_CCY_GROUP0` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Access	Meaning	Number
ENAB_GROUP_BITS	INT	RW	bit 0 to bit 7 = 1: validation of cams by track from 0 to 7, of group	%QWr.m.0.0
GROUP_AND_BITS	INT	RW	bit 0 to bit 7 = 1: assignment of tracks to group outputs	%QWr.m.0.1
GROUP_OR_BITS	INT	RW	bit 0 to bit 7 = 1: forcing group outputs (to 1)	%QWr.m.0.2

Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP0

At a Glance

This part shows the explicit exchange objects of the IODDT of type `T_CCY_GROUP0` which applies to the TSX CCY 1128 module for channel 0. It includes word objects whose bits have a special meaning. These objects are described in detail below.

Some of these objects are updated by specific functions:

- `MOD_PARAM` *(see page 195)*
- `MOD_TRACK` *(see page 198)*
- `MOD_CAM` *(see page 201)*

Example of declaring a variable: `IODDT_VAR1` of type `T_CCY_GROUP0`

Notes

- Generally speaking, the meaning of the bits is given for state 1 of this bit. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

Explicit exchange execution indicator: EXCH_STS

The table below shows the explicit exchange control bits : `EXCH_STS` (%MWr.m.c.0). These variables are automatically updated by the system on each explicit exchange.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading current channel status words	%MWr.m.c.0.0
ADJ_IN_PROGR	BOOL	R	Adjust parameters exchange in progress	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration of the module in progress	%MWr.m.c.0.15

Explicit exchange report: EXCH_RPT

The table below shows the report bits : `EXCH_RPT` (%MWr.m.c.1). These variables are automatically updated by the system on each explicit exchange.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error in reading status words of the channel (1 = failure)	%MWr.m.c.1.0
ADJ_ERR	BOOL	R	Error when exchanging adjustment parameters (1 = failure)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Fault when reconfiguring the channel (1 = failure)	%MWr.m.c.1.15

Channel fault word

The table below shows the channel fault bits: `CH_FLT`. These variables are updated by `READ_STS` (`IODDT_VAR1`).

Standard symbol	Type	Access	Meaning	Address
ENC_FLT	BOOL	R	Encoder reading fault	%MWr.m.c.2.0
AUXIL_FLT	BOOL	R	Auxiliary inputs fault	%MWr.m.c.2.1
TRACKS_FLT	BOOL	R	Track outputs fault	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	R	Internal fault, channel inoperative	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configuration fault: different hardware and software configurations	%MWr.m.c.2.5
COM_FLT	BOOL	R	Communication fault	%MWr.m.c.2.6
APPLI_FLT	BOOL	R	Application fault: configuration, adjustment or command fault	%MWr.m.c.2.7
CH_LED0	BOOL	R	Channel LED	%MWr.m.c.2.8
CH_LED1	BOOL	R	Channel LED	%MWr.m.c.2.9
ENC_SUPPLY_FLT	BOOL	R	Encoder supply fault	%MWr.m.c.2.10
ENC_WIRE_FLT	BOOL	R	Connection fault on A, B or Z	%MWr.m.c.2.11
ENC_TRANSMIT_FLT	BOOL	R	SSI frame transmission fault (parity, format)	%MWr.m.c.2.12
AUX_SUPPLY_FLT	BOOL	R	Auxiliary I/O supply fault	%MWr.m.c.2.13
C0_SUPPLY_FLT	BOOL	R	Connector 0 supply fault	%MWr.m.c.2.14
C1_SUPPLY_FLT	BOOL	R	Connector 1 supply fault	%MWr.m.c.2.15

Extended channel fault word

The table below shows the extended channel fault words and bits: `EXTEN0_FLT`, `EXTEN1_FLT`, `C0_SHORT_CIRCUIT`, `C1_SHORT_CIRCUIT`. These variables are updated by `READ_STS` (`IODDT_VAR1`).

Standard symbol	Type	Access	Meaning	Address
EXTEN0_FLT	INT	R	Channel state	%MWr.m.c.3
ENC_ALARM	BOOL	R	SSI encoder alarm	%MWr.m.c.3.2
EXTEN1_FLT	INT	R	Channel state	%MWr.m.c.4
C0_SHORT_CIRCUIT	INT	R	Short-circuit fault on connector 0	%MWr.m.c.6
C1_SHORT_CIRCUIT	INT	R	Short-circuit fault on connector 1	%MWr.m.c.7

Measurement function adjustment parameters

The table below shows the measurement adjustment objects accessible using the `READ_PARAM`, `WRITE_PARAM`, `SAVE_PARAM`, `RESTORE_PARAM`, `MOD_PARAM` functions. To find out the objects affected by each function, refer to the corresponding description.

Standard symbol	Type	Access	Meaning	Address
PRESET_ANG_VALUE	INT	RW	angle recalibration value	%MWr.m.0.20
PRESET_TURN_VALUE	INT	RW	recalibration value of the number of cycles	%MWr.m.0.21
SLACK_VALUE	INT	RW	value of axis backlash adjustment inverted : hysteresis between -1023 and + 1023	%MWr.m.0.22
MAX_PIECES	INT	RW	maximum value of number of parts	%MWr.m.0.23
ABS_OFFSET_ANG	INT	RW	value of the offset angle of the absolute encoder in relation to the machine 0	%MWr.m.0.24
ABS_OFFSET_TURN	INT	RW	angle recalibration value	%MWr.m.0.25
ABS_REDUC	INT	RW	recalibration value of the number of cycles	%MWr.m.0.26
RESOL_ANG	INT	RW	machine resolution in number of pulses per cycle	%MWr.m.0.27
RESOL_TURN	INT	RW	machine resolution in number of cycles	%MWr.m.0.28

Track adjustment parameters

The table below shows the track adjustment objects accessible using the `READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM, MOD_TRACK` functions. To find out the objects affected by each function, refer to the corresponding description

Standard symbol	Type	Access	Meaning	Address
SPECIF_TRACK_T0	INT	RW	Details of bits below	%MWr.m.0.32
-	BOOL	RW	Action on parts counter in forward direction. bit 0 =0, bit 1 = 0: no action bit 0 =1, bit 1 = 0: increment every time track set to 1 bit 0 =0, bit 1 = 1: decrement every time track set to 1 bit 0 =1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.32.0 %MWr.m.0.32.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 =0, bit 3 = 0: no action bit 2 =1, bit 3 = 0: increment every time track set to 1 bit 2 =0, bit 3 = 1: decrement every time track set to 1 bit 2 =1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.32.2 %MWr.m.0.32.3
ADD_TRACK_T0	BOOL	RW	bit =1 paralleling of tracks on an output, OUT 0 = track 0 + track 4	%MWr.m.0.32.11
INVERT_TRACK_T0	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.32.12
EVT_TRACK_T0	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.32.13
USE_TRACK_T0	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.32.15
ANTICIP_FACTOR_T0	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.33
SPECIF_TRACK_T1	INT	RW	Details of bits below	%MWr.m.0.34
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.34.0 %MWr.m.0.34.1

Standard symbol	Type	Access	Meaning	Address
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.34.2 %MWr.m.0.34.3
ADD_TRACK_T1	BOOL	RW	bit = 1 paralleling of tracks on an output, OUT 1 = track 1 + track 5	%MWr.m.0.34.11
INVERT_TRACK_T1	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.34.12
EVT_TRACK_T1	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.34.13
USE_TRACK_T1	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.34.15
ANTICIP_FACTOR_T1	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.35
SPECIF_TRACK_T2	INT	RW	Details of bits below	%MWr.m.0.36
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.36.0 %MWr.m.0.36.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.36.2 %MWr.m.0.36.3
ADD_TRACK_T2	BOOL	RW	bit = 1 paralleling of tracks on an output, OUT 2 = track 2 + track 6	%MWr.m.0.36.11
INVERT_TRACK_T2	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.36.12
EVT_TRACK_T2	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.36.13
USE_TRACK_T2	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.36.15
ANTICIP_FACTOR_T2	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.37
SPECIF_TRACK_T3	INT	RW	Details of bits below	%MWr.m.0.38

Standard symbol	Type	Access	Meaning	Address
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.38.0 %MWr.m.0.38.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.38.2 %MWr.m.0.38.3
ADD_TRACK_T3	BOOL	RW	bit = 1 paralleling of tracks on an output, OUT 3 = track 3 + track 7	%MWr.m.0.38.11
INVERT_TRACK_T3	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.38.12
EVT_TRACK_T3	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.38.13
USE_TRACK_T3	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.38.15
ANTICIP_FACTOR_T3	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.39
SPECIF_TRACK_T4	INT	RW	Details of bits below	%MWr.m.0.40
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.40.0 %MWr.m.0.40.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.40.2 %MWr.m.0.40.3
INVERT_TRACK_T4	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.40.12
EVT_TRACK_T4	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.40.13
USE_TRACK_T4	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.40.15

Standard symbol	Type	Access	Meaning	Address
ANTICIP_FACTOR_T4	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.41
SPECIF_TRACK_T5	INT	RW	Details of bits below	%MWr.m.0.42
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.42.0 %MWr.m.0.42.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.42.2 %MWr.m.0.42.3
INVERT_TRACK_T5	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.42.12
EVT_TRACK_T5	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.42.13
USE_TRACK_T5	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.42.15
ANTICIP_FACTOR_T5	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.43
SPECIF_TRACK_T6	INT	RW	Details of bits below	%MWr.m.0.44
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.44.0 %MWr.m.0.44.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.44.2 %MWr.m.0.44.3
INVERT_TRACK_T6	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.44.12
EVT_TRACK_T6	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.44.13

Standard symbol	Type	Access	Meaning	Address
USE_TRACK_T6	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.44.15
ANTICIP_FACTOR_T6	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.45
SPECIF_TRACK_T7	INT	RW	Details of bits below	%MWr.m.0.46
-	BOOL	RW	Action on parts counter in forward direction. bit 0 = 0, bit 1 = 0: no action bit 0 = 1, bit 1 = 0: increment every time track set to 1 bit 0 = 0, bit 1 = 1: decrement every time track set to 1 bit 0 = 1, bit 1 = 1: reset every time track set to 1	%MWr.m.0.46.0 %MWr.m.0.46.1
-	BOOL	RW	Action on parts counter in reverse direction. bit 2 = 0, bit 3 = 0: no action bit 2 = 1, bit 3 = 0: increment every time track set to 1 bit 2 = 0, bit 3 = 1: decrement every time track set to 1 bit 2 = 1, bit 3 = 1: reset every time track set to 1	%MWr.m.0.46.2 %MWr.m.0.46.3
INVERT_TRACK_T7	BOOL	RW	bit = 1 inversion of the track state	%MWr.m.0.46.12
EVT_TRACK_T7	BOOL	RW	bit = 1 event-triggered track	%MWr.m.0.46.13
USE_TRACK_T7	BOOL	RW	bit = 0 the track does not participate in processing bit = 1 the track participates in processing	%MWr.m.0.46.15
ANTICIP_FACTOR_T7	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.47

Cam adjustment parameters

The table below shows the cam adjustment objects accessible using the `READ_PARAM`, `WRITE_PARAM`, `SAVE_PARAM`, `RESTORE_PARAM`, `MOD_CAM` functions. To find out the objects affected by each function, refer to the corresponding description.

Standard symbol	Type	Access	Meaning	Address
SPECIF_CAM_C0	INT	RW	specifying the first cam profile	%MWr.m.c.48
FORW_ENABLE_C0	BOOL	RW	bit = 1 cam enabled in forward direction	%MWr.m.c.48.3
BACK_ENABLE_C0	BOOL	RW	bit = 1 cam enabled in reverse direction	%MWr.m.c.48.4
USE_CAM_C0	BOOL	RW	bit = 0 the cam is not processed by the program bit = 1 the cam is declared and processed by the program	%MWr.m.c.48.15

Standard symbol	Type	Access	Meaning	Address
SPECIF_CAM_1_C0	BOOL	R	specifying the second cam profile	%MWr.m.c.49
X1_C0	INT	R	value of cam threshold X1	%MWr.m.c.50
X2_C0	INT	R	value of cam threshold X2	%MWr.m.c.51
TIME_SWITCH_OFF_C0	INT	R	value of time delay at opening in steps of 100 microseconds from 0 to 16383	%MWr.m.c.52
...	...	...	...	...
SPECIF_CAM_C31	INT	RW	specifying the first cam profile	%MWr.m.c.203
FORW_ENABLE_C31	BOOL	RW	bit = 1 cam enabled in forward direction	%MWr.m.c.203.3
BACK_ENABLE_C31	BOOL	RW	bit = 1 cam enabled in reverse direction	%MWr.m.c.203.4
USE_CAM_C31	BOOL	RW	bit = 0 the cam is not processed by the program bit = 1 the cam is declared and processed by the program	%MWr.m.c.203.15
SPECIF_CAM_1_C31	BOOL	R	specifying the second cam profile	%MWr.m.c.204
X1_C31	INT	R	value of cam threshold X1	%MWr.m.c.205
X2_C31	INT	R	value of cam threshold X2	%MWr.m.c.206
TIME_SWITCH_OFF_C31	INT	R	value of time delay at opening in steps of 100 microseconds from 0 to 16383	%MWr.m.c.207

The table above describes the parameters of cams 0 and 31. The write rule is given in the following example.

For example, X2_C12 corresponds to the word %MWr.m.c.111 (111 = 51+5*12). The general rule is as follows:

- `SPECIF_CAM_Ci` corresponds to %MWr.m.c.(48+5 times i)
- `FORW_ENABLE_Ci` corresponds to %MWr.m.c.(48+5 times i).3
- `BACK_ENABLE_Ci` corresponds to %MWr.m.c.(48+5 times i).4
- `USE_CAM_Ci` corresponds to %MWr.m.c.(48+5 times i).15
- `SPECIF_CAM_1_Ci` corresponds to %MWr.m.c.(49+5 times i)
- `X1_Ci` corresponds to %MWr.m.c.(50+5 times i)
- `X2_Ci` corresponds to %MWr.m.c.(51+5 times i)
- `TIME_SWITCH_OFF_Ci` corresponds to %MWr.m.c.(52+5 times i).

Details of the implicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3

At a Glance

The IODDT of type `T_CCY_GROUP1_2_3` has implicit exchange objects, which are described below. This type of IODDT applies to the `TSX CCY 1128` module for channels 1, 2 and 3.

Example of declaring a variable: `IODDT_VAR1` of type `T_CCY_GROUP1_2_3`

Generally speaking, the meaning of the bits is given for state 1 of this bit. In specific cases, the two states of the bit are explained.

List of implicit exchange %IW input objects

The table below shows the implicit exchange %IW input objects of the IODDT of type `T_CCY_GROUP1_2_3` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Access	Meaning	Number
GROUP0_TRACKS	INT	R	bit 0 to bit 7: state of group 0 tracks	%IWr.m.0.0

List of implicit exchange %QW output objects

The table below shows the implicit exchange %QW output objects of the IODDT of type `T_CCY_GROUP1_2_3` which applies to channel 0 of the TSX CCY 1128 module.

Standard symbol	Type	Access	Meaning	Number
ENAB_GROUP_BITS	INT	RW	bit 0 to bit 7 = 1: validation of cams by track from 0 to 7, of group	%QWr.m.0.0
GROUP_AND_BITS	INT	RW	bit 0 to bit 7 = 1: assignment of tracks to group outputs	%QWr.m.0.1
GROUP_OR_BITS	INT	RW	bit 0 to bit 7 = 1: forcing group outputs (to 1)	%QWr.m.0.2

Details of the explicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3

At a Glance

This part shows the explicit exchange objects of the IODDT of type T_CCY_GROUP1_2_3 which applies to the TSX CCY 1128 module for channels 1, 2 and 3. It includes word objects whose bits have a special meaning. These objects are shown in detail below.

Some of these objects are updated by specific functions:

- MOD_PARAM *(see page 195)*
- MOD_TRACK *(see page 198)*
- MOD_CAM *(see page 201)*

Example of declaring a variable: IODDT_VAR1 of type T_CCY_GROUP1_2_3

Notes

- Generally speaking, the meaning of the bits is given for state 1 of this bit. In specific cases, each bit state is explained.
- Not all bits are used.

Track adjustment parameters

The table below shows the track adjustment objects accessible using the READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM, MOD_TRACK functions. To find out the objects affected by each function, refer to the corresponding description

Standard symbol	Type	Access	Meaning	Number
SPECIF_TRACK_T0	INT	RW	Details of bits below	%MWr.m.0.32
ADD_TRACK_T0	BOOL	RW	bit =1 paralleling of tracks on an output, OUT 0 = track 0 + track 4	%MWr.m.0.32.11
INVERT_TRACK_T0	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.32.12
EVT_TRACK_T0	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.32.13
USE_TRACK_T0	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.32.15
ANTICIP_FACTOR_T0	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.33
SPECIF_TRACK_T1	INT	RW	Details of bits below	%MWr.m.0.34
ADD_TRACK_T1	BOOL	RW	bit =1 paralleling of tracks on an output, OUT 1 = track 1 + track 5	%MWr.m.0.34.11
INVERT_TRACK_T1	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.34.12

Standard symbol	Type	Access	Meaning	Number
EVT_TRACK_T1	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.34.13
USE_TRACK_T1	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.34.15
ANTICIP_FACTOR_T1	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.35
SPECIF_TRACK_T2	INT	RW	Details of bits below	%MWr.m.0.36
ADD_TRACK_T2	BOOL	RW	bit =1 paralleling of tracks on an output, OUT 2 = track 2 + track 6	%MWr.m.0.36.11
INVERT_TRACK_T2	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.36.12
EVT_TRACK_T2	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.36.13
USE_TRACK_T2	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.36.15
ANTICIP_FACTOR_T2	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.37
SPECIF_TRACK_T3	INT	RW	Details of bits below	%MWr.m.0.38
ADD_TRACK_T3	BOOL	RW	bit =1 paralleling of tracks on an output, OUT 3 = track 3 + track 7	%MWr.m.0.38.11
INVERT_TRACK_T3	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.38.12
EVT_TRACK_T3	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.38.13
USE_TRACK_T3	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.38.15
ANTICIP_FACTOR_T3	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.39
SPECIF_TRACK_T4	INT	RW	Details of bits below	%MWr.m.0.40
INVERT_TRACK_T4	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.40.12
EVT_TRACK_T4	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.40.13
USE_TRACK_T4	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.40.15
ANTICIP_FACTOR_T4	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.41
SPECIF_TRACK_T5	INT	RW	Details of bits below	%MWr.m.0.42
INVERT_TRACK_T5	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.42.12
EVT_TRACK_T5	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.42.13
USE_TRACK_T5	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.42.15
ANTICIP_FACTOR_T5	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.43
SPECIF_TRACK_T6	INT	RW	Details of bits below	%MWr.m.0.44

Standard symbol	Type	Access	Meaning	Number
INVERT_TRACK_T6	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.44.12
EVT_TRACK_T6	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.44.13
USE_TRACK_T6	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.44.15
ANTICIP_FACTOR_T6	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.45
SPECIF_TRACK_T7	INT	RW	Details of bits below	%MWr.m.0.46
INVERT_TRACK_T7	BOOL	RW	bit =1 inversion of the track state	%MWr.m.0.46.12
EVT_TRACK_T7	BOOL	RW	bit =1 event-triggered track	%MWr.m.0.46.13
USE_TRACK_T7	BOOL	RW	bit =0 the track does not participate in processing bit =1 the track participates in processing	%MWr.m.0.46.15
ANTICIP_FACTOR_T7	INT	RW	anticipation factor in steps of 50 microseconds value between 0 and 32767	%MWr.m.0.47

Cam adjustment parameters

The table below shows the cam adjustment objects accessible using the `READ_PARAM`, `WRITE_PARAM`, `SAVE_PARAM`, `RESTORE_PARAM`, `MOD_CAM` functions. To find out the objects affected by each function, refer to the corresponding description

Standard symbol	Type	Access	Meaning	Number
SPECIF_CAM_C0	INT	RW	specifying the first cam profile	%MWr.m.c.48
FORW_ENABLE_C0	BOOL	RW	bit =1 cam enabled in forward direction	%MWr.m.c.48.3
BACK_ENABLE_C0	BOOL	RW	bit =1 cam enabled in reverse direction	%MWr.m.c.48.4
USE_CAM_C0	BOOL	RW	bit =0 the cam is not processed by the program bit =1 the cam is declared and processed by the program	%MWr.m.c.48.15
SPECIF_CAM_1_C0	BOOL	R	specifying the second cam profile	%MWr.m.c.49
X1_C0	INT	R	value of cam threshold X1	%MWr.m.c.50
X2_C0	INT	R	value of cam threshold X2	%MWr.m.c.51
TIME_SWITCH_OFF_C0	INT	R	value of time delay at opening in steps of 100 microseconds from 0 to 16383	%MWr.m.c.52
...	...	...	...	...

Standard symbol	Type	Access	Meaning	Number
SPECIF_CAM_C31	INT	RW	specifying the first cam profile	%MWr.m.c.203
FORW_ENABLE_C31	BOOL	RW	bit =1 cam enabled in forward direction	%MWr.m.c.203.3
BACK_ENABLE_C31	BOOL	RW	bit =1 cam enabled in reverse direction	%MWr.m.c.203.4
USE_CAM_C31	BOOL	RW	bit =0 the cam is not processed by the program bit =1 the cam is declared and processed by the program	%MWr.m.c.203.15
SPECIF_CAM_1_C31	BOOL	R	specifying the second cam profile	%MWr.m.c.204
X1_C31	INT	R	value of cam threshold X1	%MWr.m.c.205
X2_C31	INT	R	value of cam threshold X2	%MWr.m.c.206
TIME_SWITCH_OFF_C31	INT	R	value of time delay at opening in steps of 100 microseconds from 0 to 16383	%MWr.m.c.207

The table above describes the parameters of cams 0 and 31, the write rule is given in the following example.

For example, X2_C12 corresponds to the word %MWr.m.c.111 (111 = 51+5*12). The general rule is as follows:

- `SPECIF_CAM_Ci` corresponds to %MWr.m.c.(48+5 times i)
- `FORW_ENABLE_Ci` corresponds to %MWr.m.c.(48+5 times i).3
- `BACK_ENABLE_Ci` corresponds to %MWr.m.c.(48+5 times i).4
- `USE_CAM_Ci` corresponds to %MWr.m.c.(48+5 times i).15
- `SPECIF_CAM_1_Ci` corresponds to %MWr.m.c.(49+5 times i)
- `X1_Ci` corresponds to %MWr.m.c.(50+5 times i)
- `X2_Ci` corresponds to %MWr.m.c.(51+5 times i)
- `TIME_SWITCH_OFF_Ci` corresponds to %MWr.m.c.(52+5 times i).

Configuration constants

Module level constants

%KW common to the module:

Object	Symbol	Meaning
%KWr.m.0.0	**EVT_CONF** EVT_ENABLE EVT_NUM	Event configuration Byte 0: masking the event 16#FF no event task 16#00 event task priority 0 16#01 event task priority 1 Byte 1: event task number 16#FF no event task
%KWr.m.0.1	reserved	
%KWr.m.0.2	**INPUT_CONF** INPUT_MOD LINE_FILT DIRECTION_INV MULT4_RESOL LINE_CTRL FORMAT_MEAS_0 FORMAT_MEAS_1 PRESET_MOD_0 PRESET_MOD_1 CAPTS_MOD_0 CAPTS_MOD_1 CAPTS_MOD_2	Configuration of the'axis bit 0, bit 1: Type of encoder = 16#00 : Incremental encoder = 16#01 : SSI link absolute encoder = 16#02 : parallel absolute encoder and TSX ABE7CPA11 bit 2: reserved bit 3 =1 filtering of inputs (disconnection frequency125KHz) bit 4 =1 the axis is rotating in the opposite direction to the encoder bit 5: reserved bit 6 =1 multiplication by 4 of the encoder resolution bit 7 =1 encoder line check bit 8, bit 9 measurement type =16#00 Type 1 (angle) =16#01 Type 2 (angle + cycles) =16#02 Type 3 (linear) bit 10, bit 11 recalibration type =16#00 no recalibration =16#01 recalibration on rising edge of Irec =16#02 recalibration on rising edge of Irec recalibration on falling edge of Irec in REV =16#03 recalibration on rising edge of Z FORWARD and Irec=1 recalibration on falling edge of Z in REV and Irec =1 bit 12, bit 13, bit 14 capture types capture 0: on the rising edge Icapt0 in every case capture 1: =16#00 on falling edge Icapt0 =16#01 on rising edge Icapt1 =16#02 number of points per cycle =16#03 angle value before recalibration =16#04 angle value on rising edge of Z bit 15: reserved

Object	Symbol	Meaning
%KWr.m.0.3		reserved
%KWr.m.0.4	**ABS_ENC_CONF_0** GRAY WITH_PAR EVEN_PAR WITH_ERR ERROR_LEV	SSI absolute encoder configuration Choice of binary / Gray code bit 0 =0 binary bit 0 =1 Gray bit 1 =1 parity bit present in the frame bit 2 =1 even parity bit 3, bit 4 reserved bit 5 =1 error bit present in the frame bit 6 = 1 logical level of error bit (default for logic 1) bit 7 to bit 15 reserved
%KWr.m.0.5	**ABS_ENC_ERROR_RANGE**	rank of error bit in the status bits value: 16#00, 16#01, 16#02, 16#03, or 16#04
%KWr.m.0.6	**ABS_ENC_READ_PERIOD**	encoder read period: 16#00 = 50 microseconds 16#01 = 100 microseconds 16#02 = 200 microseconds
%KWr.m.0.7	**ABS_ENC_EXTRA_NB** BEGIN_NB STAT_NB	number of additional bits supplied by the encoder: Byte 0: number of header bits length of header field before MSB = 4 max Byte 1: number of status bits length of status field before LSB = 4 max
%KWr.m.0.8	**ABS_ENC_DATA_NB**	Total number of encoder bits length of data field = 25 max
%KWr.m.0.9 to %KWr.m.0.14		reserved

Object	Symbol	Meaning
%KWr.m.0.15	CONTROLES_CONF C0_LOCK C0_LOCK PCAM_STAND_ALONE PCAM_IGN_SC OUTS_MAINT SUPPLY_AUX_MSK SUPPLY_ENC_MSK SUPPLY_C0C1_MSK REARM_MOD	Fault configuration bit 0: locking connector 0 bit 0 =1 no power supply monitoring on connector 0 bit 1: locking connector 0 bit 1 =1 no power supply monitoring on connector 1 bit 2 to bit 7 reserved bit 8: Option on switching to PLC communication fault: bit 8 = 0 the cam processor goes to STOP bit 8 = 1 cam processor remains in RUN bit 9: Option to switch to short-circuit fault bit 9 = 0 the cam processor goes to STOP bit 9 = 1 cam processor remains in RUN bit 10: Option on switching to PLC communication fault: bit 10 =0 outputs are set to 0V bit 10 =1 direct commands on maintained outputs bit 11: Option on auxiliary input power supply fault bit 11 =0 sets bit: %Ir.m.0.ERR bit 11 =1 no effect on %Ir.m.0.ERR bit 12: Option on encoder power supply fault bit 12 =0 sets bit: %Ir.m.0.ERR bit 12 =1 no effect on %Ir.m.0.ERR bit 13: Option on CNX0 and CNX1 connectors power supply fault bit 13 =0 sets bit: %Ir.m.0.ERR bit 13 =1 no effect on %Ir.m.0.ERR bit 14: Reactivation of output option bit 14 =0 reactivation on explicit command bit 14 =1 reactivation 10 seconds after tripping bit 15: reserved
%KWr.m.0.16	SPEED_FORMAT	expression of speed 16#00 : speed expressed in pts/ms 16#04 : speed expressed in pts/s
%KWr.m.0.17		reserved

Group level constants

%KWr.m

Object	Symbol	Meaning
%KWr.m.c.18	INVERT_OUTi INVERT_OUTi_0 INVERT_OUTi_1 INVERT_OUTi_7	electric level of track outputs of group "i" Inversion of track output i.0: bit 0 = 0 output i.0 = 24V for a logic state 1 bit 0 = 1 output i.0 = 24V for a logic state 0 Inversion of track output i.1: bit 1 = 0 output i.0 = 24V for a logic state 1 bit 1 = 1 output i.0 = 24V for a logic state 0 the same right up to: Inversion of track output i.7: bit 7 = 0 output i.0 = 24V for a logic state 1 bit 7 = 1 output i.0 = 24V for a logic state 0

Glossary

A

Absolute encoder

This type of encoder directly delivers the numeric value of the axis position. The position reading is kept in case of power outage.

Acquisition

Module functions that are used to calculate the position reading of the machine.

Adjusting the Recipe

Adjusting the recipe is a PL7 facility that allows modification of the parameters of a recipe element (axis, cam or track) in connected mode. Modifications done under recipe adjustment do not stop the cam processor.

Alternating movement

This is a typical movement for hydraulic presses and transfer machines. The axis describes a to and fro movement in a pulses domain equal to or less than the value of the cycle.

The "reading format" is type 1. The arrival of parts is synchronized by the machine (Machine Synchro).

Angle reading

This is the instant position value of the axis in the cycle. This value is expressed as a number of pulses.

Anticipation

Cam processor function that is used to compensate for the delay caused by the machine's actuators.

The anticipation value is applied to all changes of track status. It is specified by the "anticipation factor" defined in the recipe parameters (value between 0 and 32 767 x 50 micro seconds).

Axis

These are all the external elements that control the movements of the machine (reducer, encoder…).

Axis play

This is the position error caused by the axis according to the direction of movement. The cam processor can compensate for this error. The value of axis play is supplied in a configuration parameter.

C

Cam

Logic state which switches to 1 when an angular value in the cycle is overshot and which resets to 0 according to the type of cam.

A cam is systematically associated to one and only one track. Several cams can be associated to one track.

Cam descriptor

Part of the cam program that characterizes a cam (number of associated track, thresholds, type, confirmation condition, etc.).

Cam processor

This is the part of the module that directly controls the outputs according to the angle reading and according to the cam program supplied with the module.

Cam program

This is all the internal data that defines the activation of the outputs according to the position reading of the axis. The cam program represents the most important part of the recipe.

Capture

A module function that is used to sample the position value of the axis on detection of a precise event (input Icapt0 and/or Icapt1). The capture has no impact on the values of the axis nor on the cam processor.

The implementation of this function allows the application to manage the process better, for example: in monitoring the number of pulses sent by the encoder, the size of the parts, the axis slip, the angle of arrival of the parts.

Cold restart

The module is initialized with current configuration and adjustment parameters.

This happens after the PLC supply has been cut or the supply has been reset or the module has been disconnected.

The module will execute the recipe in operation before the Reset.

Communication fault

Fault detected by the module when periodic exchanges with the PLC processor are no longer made.

Configuration

The configuration gathers together the data which characterizes the machine (invariant) and which is necessary for the TSX CCY 1128 module to operate.

All this information is stored in the PLC %KW constants zone. The PLC application cannot modify them.

Continuous movement

This is the movement typical of conveyor belts. Theoretically the cycle is infinite. In fact, the limit for this module is 32768 pulses. The axis must be recalibrated when each part arrives (Part Synchro).

The "reading format" is type 3.

Cycle

Area in which the action of the outputs can be programmed.

NOTE: A complete cycle of the machine (machine cycle) must be a whole number of cycles.

Cyclical movement

This is the typical movement for packaging machines. The axis describes several cycles to carry out all the operations on a part. The direction of movement is usually constant.

The "reading format" is type 2. The arrival of parts is synchronized by the machine (Machine Synchro).

D

Debugging

Debugging is a PL7 facility that allows direct monitoring of the module in connected mode.

Deviation angle

This is the axis position reading, taken each time it passes machine zero. This function for measuring axis slip can be accessed via the capture function.

E

Encoder

Position sensor accepted by the module It can be incremental, SSI absolute or parallel (via Telefast ABE 7CPA11).

F

Filtering

Filtering allows better environmental handling (for difficult environments). It is a filter that limits the bandwidth of counting signals which come from an incremental encoder.

With filtering the permitted frequency (before multiplying by 4) is 125 KHz, as against 250 KHz without filtering.

G

Gray code

A binary code known as reflected, in which the change from term n to term n+1 is carried out by changing only a single digit, and so the code can be read unambiguously.

I

Incremental encoder

Pulse generator with 2 signals offset by 90°. These are produced according to the movement of the axis and are counted by the module.

L

Line monitoring

System of monitoring the connection lines with the encoder. It detects breaks and short circuits between signals in the cable.

M

Machine cycle

All of the cycles necessary to effect a complete operation of the machine.

Machine synchro

Synchronization is carried out on a physical reference of the machine called "Machine zero".

In this case the mechanism must control the arrival of the parts in the cycle. Each part will have to arrive for the same angle value, but you will always be able to find several parts on the machine at the same time.

Machine zero

This is the mechanical reference position of the machine.

Measurement format

Defines the format for the axis position reading calculated by the module. It depends on the type of machine.

O

Offset

This is the rough value delivered by an absolute encoder to the machine zero.

By completing the adjustment parameter "Offset encoder", it is possible to set the angle value of the axis to 0 on the machine zero position.

Operating mode

These are all the rules that govern the performance of the module during transitory phases or when an error appears.

Output fallback

This is the behavior of the outputs under different errors:

- If an electrical fault is detected on a connector (short circuit or supply under voltage).
  - All faulty connector outputs switch to 0V.
  - Direct commands continue to be applied on the outputs of the other connector. The cam processor can be set to Stop (according to the configuration).
- If communication between the module and the central unit is defective.
  - The cam processor can be set to Stop (according to the configuration)
  - Direct commands continue to be applied or not (according to the configuration)
- If the cam processor is in Stop mode.
  - The tracks are set to logical 0.
  - The outputs can no longer be conditioned except by direct commands and INV information set at configuration.
- If the module is not configured (RUN light is out), the outputs are at 0V.

P

Part synchro

Synchronization is carried out when each part arrives.

In this case parts can arrive at random on the machine, but the tool box can only handle one part at a time.

Pulses per cycle

This is the number of pulses delivered by the axis per cycle.

In relation to the encoder resolution, this adjustment parameter must be considered:
- after multiplying by 4 for an incremental encoder
- after reducing the resolution for an absolute encoder.

R

Reading inversion

This function allows the module to adapt itself to the type of mechanical mounting of the encoder on the axis.

Recalibration

This is the module function which is used to adjust the axis in relation to the machine zero or to synchronize the axis in relation to the arrival of a part.

Recalibration forces the position reading to a value which is predefined by the "recalibration value" parameter (included in the pulse field of the cycle).

The module allows recalibration to be carried out systematically for each cycle or for only one cycle. This recalibration is always conditioned to the IREC input.

Recipe

The recipe groups together the data required for the module to control the machine for a series of parts. The recipe can be modified or completely changed by the PLC application. All this information is contained in the PLC %MW words in the memory zone reserved for the module.

Reducing the resolution

This function is used to divide the position value delivered by an absolute encoder via the "resolution reduction factor" by 2, 4, 8, 16 or 32.

Resetting

Module function that allows a return to normal for the outputs after the outputs have been tripped.

The reset mode can be configured as "Manual" or "Automatic":
- In "Manual", resetting is conditioned by a command bit controlled by the application.
- In "Automatic", resetting is carried out 10 seconds after the trip.

Resolution

this is the smallest variation of input information which gives a measurable piece of output information.

Rotating movement

This is a typical movement for mechanical presses and punching machines. The axis describes a complete cycle to carry out all the operations on a part. The direction of rotation is constant.

The "reading format" is type 1. The arrival of parts is synchronized by the machine (Machine Synchro).

S

Short circuit fault

The module includes a thermal circuit break system for the module's 24 V outputs.

This system shows up a "short circuit fault".

Before a break, the short circuit current is limited to 1.5 A. The 24 V supply must be able to cope with this overload without a drop in voltage so that the rest of the application is not disrupted if there is an output fault.

Slipping

This is the error of pulse loss in a cycle. This can be caused by the mechanical transmission of the axis. The module allows measurement of this slip (see: "deviation angle").

SSI encoder

Synchronous Series Interface link This is the standard connection protocol for absolute encoders that is used by the module.

The transmission frequency is set by the module according to the following configuration parameters:
- Number of bits making up the frame,
- Reading period (50, 100 or 200 micro seconds).

Synchro

"Part synchro", "Machine synchro": these are the two main modes of synchronization of the axis (which brings the parts in) in relation to the tool box. One or other of them is used according to the type of application.

Synchronization is necessary with an incremental encoder.

T

Top dead center

With mechanical presses, you find a zone in the cycle which is called TDC. It is in this zone that the machine can and must be stopped. The "brake" cam is specially designed to deal with this problem.

Track

This is the logic state which can be applied to the physical output. The maximum number of tracks is 32.

Track descriptor

Part of the cam program that sets the parameters for the functions associated with a track (anticipation factor, generation of an event, evolution of the parts counter).

Turn pulse marker

A pulse supplied by a rotary incremental encoder, detected on each complete turn of the axis.

Type of cam

Essential feature of a cam. Defines the type of cam calculation according to the angle value (position, monostable or brake command).

Type of movement

This is the feature of the machine that sets the speed cycles on the axis.

W

Warm restart

The module is initialized with the initial configuration and adjustment parameters.

This happens after a Reset of the CPU.

The module executes the initial recipe. Modifications done in connected mode, are not recognized if a "Save parameters" has not been carried out before the Reset.

Index

A

ABE7-CPA11, *135*
absolute encoder
  offset , *223*
adjusting, *241*
  recipes, *219*

B

Braking cam, *235*

C

channel data structure for electronic cam
  T_CCY_GROUP0, *283*
  T_CCY_GROUP1_2_3, *283*
  T_GEN_MOD, *283*
configuring, *161*
connecting, *113*
connecting absolute encoder, *91*
connecting incremental encoders, *91*

D

debugging, *241*
DETAIL_OBJECT, *177*
diagnosing, *255*
diagnostics, *255*

E

error codes, *259*
event management, *185*
Event processing, *184*

F

FAQs, *269*

I

Incremental encoder
  Electronic cam module parameters, *221*
installing, *79*

M

MOD_CAM, *201*
MOD_PARAM, *195*
MOD_TRACK, *198*
monostable cam, *234*
movement types, *41*

P

parameter settings, *283*
performances, *271*
programming, *177*

Q

quick start, *55*

R

READ_PARAM, *192*
RESTORE_PARAM, *193*

S

SAVE_PARAM, *194*

T

T_CCY_GROUP0, *283*
T_CCY_GROUP1_2_3, *283*
T_GEN_MOD, *283*
TELEFAST2
    encoders, *133*
TRF_RECIPE, *204*
TSXCAPS15, *107*
TSXCCPS15x, *108*
TSXCCY1128, *149*
TSXTAPS15xx, *108*

W

WRITE_PARAM, *190*