Premium and Atrium using EcoStruxure™ Control Expert

Motion Control Modules for Step by Step Motor

User Manual

(Original Document)

12/2018

35006224.11

www.schneider-electric.com

Schneider Electric

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

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

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.

Table of Contents

Safety Information

Important Information

NOTICE

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

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

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

⚠ DANGER

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

⚠ WARNING

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

⚠ CAUTION

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

NOTICE

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

PLEASE NOTE

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

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

BEFORE YOU BEGIN

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

⚠ WARNING

UNGUARDED EQUIPMENT

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

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

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

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

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

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

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

START-UP AND TEST

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

⚠ WARNING

EQUIPMENT OPERATION HAZARD

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

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

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

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

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

Before energizing equipment:

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

OPERATION AND ADJUSTMENTS

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

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

About the Book

At a Glance

Document Scope

This manual describes the software implementation of the motion control application for step by step motors in Premium PLC's, using Control Expert software.

Validity Note

This documentation is valid for EcoStruxure™ Control Expert 14.0 or later.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

Related Documents

Title of documentation	Reference number
Premium and Atrium using EcoStruxure™ Control Expert, Processors, racks and power supply modules, Implementation Manual	35010524 (English), 35010525 (French), 35006162 (German), 35012772 (Italian), 35006163 (Spanish), 35012773 (Chinese)
Premium and Atrium using EcoStruxure™ Control Expert, Axis Control Modules for Servomotors, User Manual	35006220 (English), 35006221 (French), 35006222 (German), 35014004 (Italian), 35006223 (Spanish), 35014005 (Chinese)
EcoStruxure™ Control Expert, Operating Modes	33003101 (English), 33003102 (French), 33003103 (German), 33003104 (Spanish), 33003696 (Italian), 33003697 (Chinese)
EcoStruxure™ Control Expert, I/O Management, Block Library	33002531 (English), 33002532 (French), 33002533 (German), 33003684 (Italian), 33002534 (Spanish), 33003685 (Chinese)

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

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.

Part I
The servo drive axis control in Premium PLCs

Aim of the part

This part offers an overview of the step by step axis control range and describes the methodology for the installation of the axes.

What Is in This Part?

This part contains the following chapters:

Chapter 1
General on step by step axes control

Subject of this chapter

This chapter introduces step by step axes control, describes it and the associated application-specific functions.

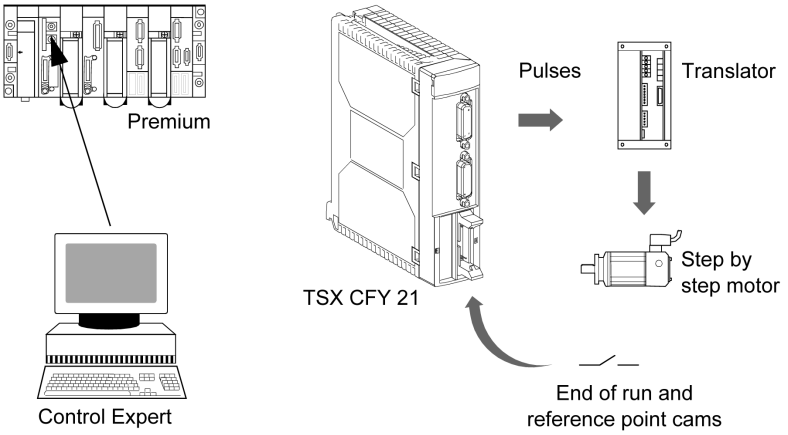
What Is in This Chapter?

This chapter contains the following topics:

Presentation of the step by step axes control range

Introduction

The diagram below shows the control architecture for step by step axes:



Step by step axes control environment

The step by step axes control environment for Premium PLCs is made up of 2 modules:

- TSX CFY 11 (1 limited linear axis),
- TSX CFY 21 (2 independently limited linear axes).

The Control Expert software includes the basic application-specific step by step movement function, used to program these step by step axes control modules.

Basic movements are driven from the machine's principal sequential control program, but carried out and controlled by the TSX CFY modules.

These modules control the rotation speed of a step by step motor, as well as its accelerations and decelerations, by delivering a frequency control to a translator (fmax = 187 KHz). The translator transforms each pulse into a basic movement of the step by step motor.

Control of the step by step motor is carried out in open loop. End of run, reference point and event inputs allow the module to control the movements of the moving part on the axis.

Some translators include a step failure mechanism: this information is made available to the user program, which can create a new reference point.

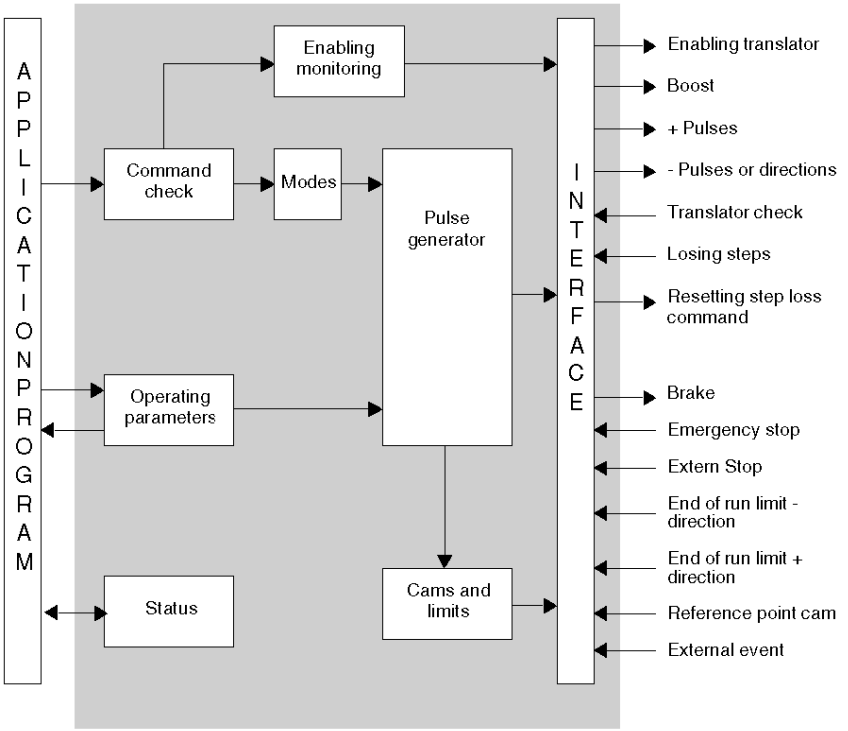
The step by step axes control environment also includes the TSX CXP 611 cable to link the TSX CFY11/21 modules directly to the MSD and MS translators from the manufacturer Phyton Lektronik GmbH.

Functionality offered by the axes control modules

General

The axes control modules offer application inputs and outputs for each axis which allow you to implement the different functions.

The following structural breakdown presents the inputs/outputs associated to a channel:



Application inputs/outputs

The step by step axes control modules offer for each of the axes:

for auxiliary inputs/outputs:

- one cam reference point input,
- two end of run inputs,
- one event input,
- one emergency stop input,
- one external stop input,
- one static output for the axis brake.

for translator inputs/inputs:

- one translator check input,
- one step failure control input,
- one differential translator validation output,
- two differential pulse outputs, one positive, one negative,
- one differential step by step motor boost output,
- one differential step failure reset output.

Programming a movement

In Control Expert language, each movement is described by a SMOVE movement control function. From this SMOVE command and the position of the moving part, the TSX CFY module creates the position/speed setpoint and generates the pulses for this movement.

Configuration and control parameters

These parameters are used to define the specifications of use, limits, etc.

Specific functions of the TSX CFY modules

The functions offered by the step by step axes control modules are as follows:

- **Event processing**: the events detected by the module can be used to activate an event task in the sequential program.
- **Boost command**: this function allows you to boost the step by step motor during acceleration or deceleration phases.
- **Brake command**: this function allows you to control the brake of the step by step motor when starting and stopping the movement.
- **Feed hold**: this function allows you to stop the movement in progress momentarily.
- **Deferred pause**: this function allows you to stop a machine cycle momentarily without disrupting it.
- **End of run limits**: overshooting these limits triggers the stopping of the movement. Having overshot an end of run limit, only return movements between the limits are accepted.
- **External stop**: activation of the external stop input causes the movement to stop.
- **Step failure input and step failure control reset output**: these functions allow you, via the application, to manage step failure information originating from the translator. For the module, the activation of step failure input does not constitute a stop condition, nor an error condition.

General on the step by step axes control

Introduction to specific functions

The specific functions of step by step axes control apply to the whole control system made up of:

- the command
- the translator
- the step by step motor.

The important ideas are as follows:

- the start stop frequency SS_FREQ
- the boost
- the brake output

Diagram of the step by step axes control system

This diagram describes a typical step by step axes control system.

Description

Block	Description
Command	The command function is carried out by a channel of a TSX CFY 11 or 21 module. The principal function of this channel is to provide a series of pulses of constantly controlled frequency, in order to carry out the required functions.
Translator	The translator's essential function is to transform each received pulse into a step (elementary rotation) of the motor, by circulating suitable currents in its windings.
Step by step motor	Step by step motors are built according to different technologies. For example permanent magnet motors, variable reluctance motors and hybrid motors which combine the two techniques. Furthermore, different winding solutions exist on the market: two, four or five phase motors. Each type of motor is thus associated with a specific type of translator optimized for its technology.

Start stop frequency

The control of different step by step systems must generally obey a common restraint, due to the response of the inertial system (motor and axis) to a pulsed command. The common constraint is the start stop frequency.

The start stop frequency is the frequency at which the motor can stop or start without ramp and without losing step. Its maximum threshold depends upon the external inertia related to the motor axis. Its average value is 400 Hz in a 1/2 step (1 turn/sec) and can be critical beyond 600/800 Hz (1.5 to 2 turns/sec) (typical values for Phytron Elektronik tranlators/200 step motors/lathes).

This constraint exists at the stopping as at the starting of all movements, whence its name: **start stop frequency**, SS_FREQ. The TSX CFY modules allow you to adjust this value.

NOTE: in this manual, the terms **frequency** and **speed are used interchangeably.** In the same way, the units of pulse position and pulse are the same. It is the same for the units of speed **Hertz** and **Pulses/s** and the acceleration units **Hertz/s** and **Pulses/s** ^{2}.

Boost

Some translators have a boost input. This function consists in increasing the current in the motor windings.

The boost output from a channel of a TSX CFY module controls this input to the translator. It is thus possible to control the intensity of the motor current in synchronization with the movement. Notably automatic control mode for this output supports its activation during acceleration and deceleration phases.

Brake output

When a brake is present on the axis, this static output supports its control in synchronization with the movement or at the request of the user.

This function is useful when you want to interrupt the supply to the motor in load carrying applications.

NOTE: when the channel is in the safety position, this output makes the brake active (generally the brake is active when voltage is absent).

Chapter 2
Introduction to step by step axis command

Subject of this chapter

This Chapter introduces the step by step axis command.

What Is in This Chapter?

This chapter contains the following topics:

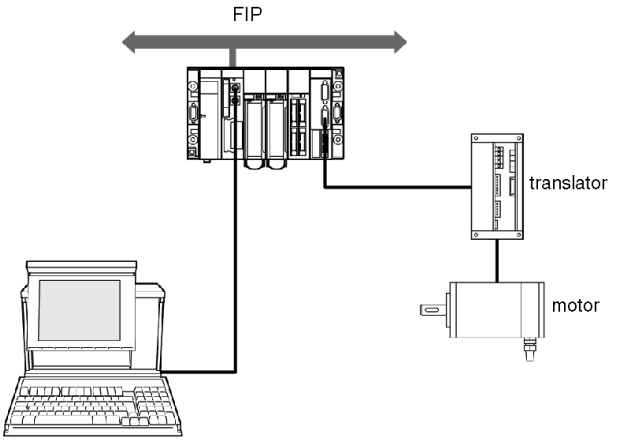
General

Step by step axis command offer

The TSX CFY 11/21 step by step axis command for Premium PLCs is offered to satisfy the demands of the machine manufacturers.

It is designed for machines, which require a step by step movement command by a motor linked with a sequential command by a programmable PLC.

Illustration:

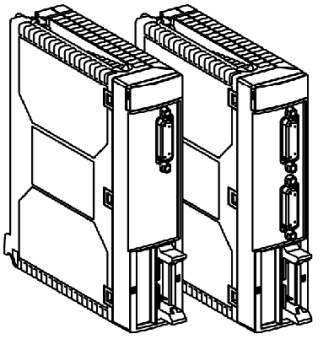


At a Glance

Two modules are available:

- **TSX CFY 11** module: an axis with a command output with one translator,
- **TSX CFY 21** module: two axes with two command outputs with two translators.

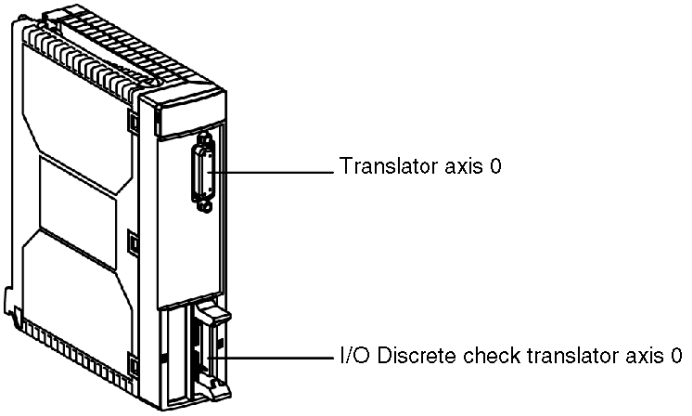
Illustration:



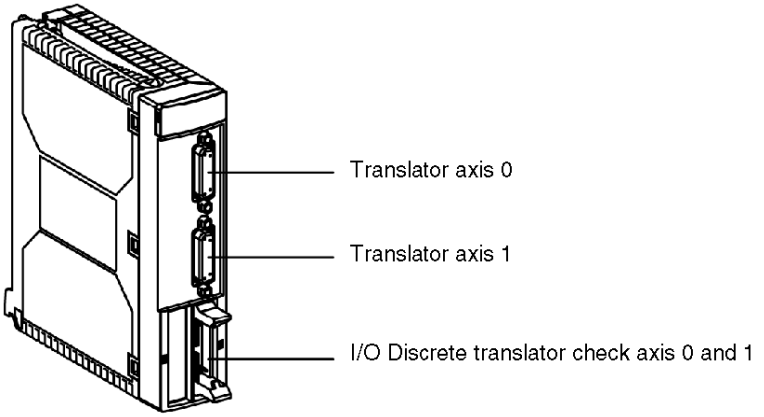
Physical description

Description of step-by-step axis command modules.

TSX CFY 11 module:



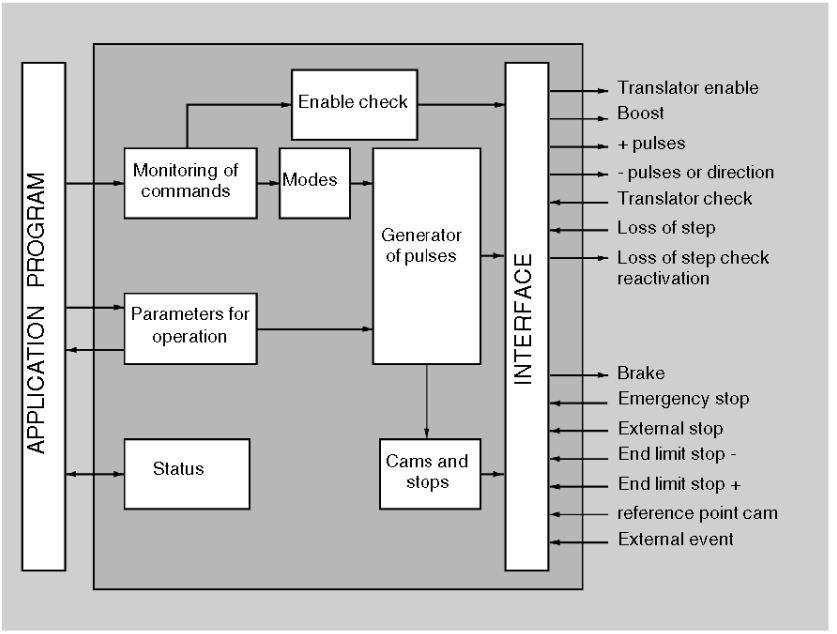
TSX CFY 21:



Standard functions

Illustration

Summary of step-by-step axis command module:



Features of TSX CFY 11/21 step-by-step axis command modules

Each axis of TSX CFM 11/21 axis command modules has:

- inputs
  - a translator check input,
  - an control input for loss of step,
  - a + end limit input,
  - a - end limit input,
  - a reference point cam input,
  - an event input,
  - an emergency stop input,
  - an external stop input,
- outputs
  - brake output,
  - pulse+ output,
  - pulse - output or direction,
  - loss of step control reactivation output,
  - boost output,
  - translator enable output.

Chapter 3
Implementation methodology

Installation Phase Overview

Introduction

The software installation of the application-specific modules is carried out from the various Control Expert editors:

- in offline mode
- in online mode

If you do not have a processor to connect to, Control Expert allows you to carry out an initial test using the simulator. In this case the installation *(see page 28)* 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.

Chapter 4
Introductory example

Subject of this chapter

This chapter describes the implementation of an axes control application with the help of a TSX CFY module. This instructional example allows you to follow all the phases necessary to the implementation of step by step axis control.

What Is in This Chapter?

This chapter contains the following topics:

Description of the example

Introduction

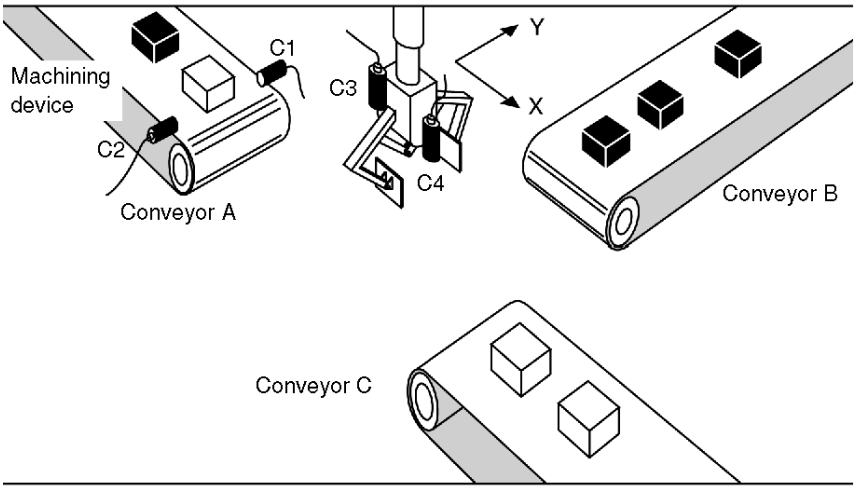
The following example allows you to follow all the implementation phases of a TSX CFY axes control application. It complements the set-up methodologies.

Transfer Device

A transfer device removes parts coming out of machining. This device is made up of a claw that can move in space (X, Y axes) on a plane parallel to the ground.

As soon as a part appears on output conveyor belt A, the claw automatically goes to fetch it and put it on belt B or belt C according to the part type. Then the claw returns to the stand-by position and will move again as soon as a new machined part is detected.

The following figure illustrates this transfer device:



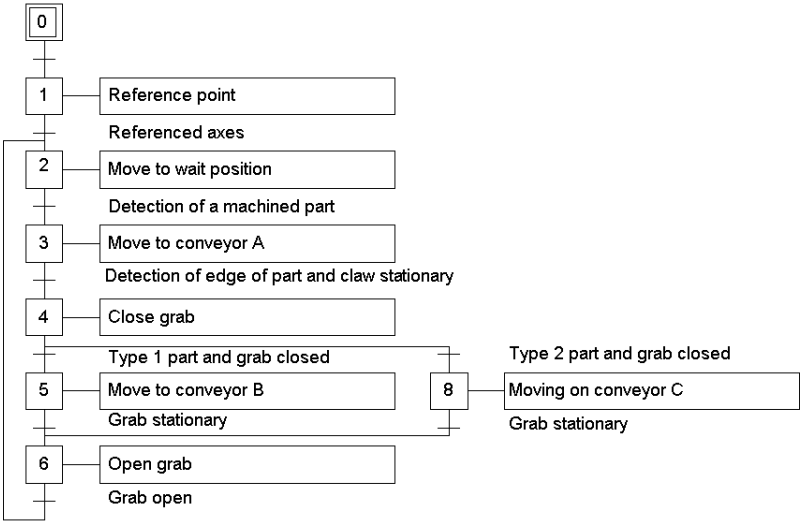
Inputs/outputs

The inputs/outputs are as follows:

I/O	Description
C1	Machined part detection cell
C2	Sensor to identify the part type.
C3	Grab open / grab closed detection sensor.
C4	Part edge detection cell (located in the grab), connected to the module's event input.
ENC0	Incremental encoder in the X axis position.
ENC1	Incremental encoder in the Y axis position.
O/C grab	Open / close claw command.

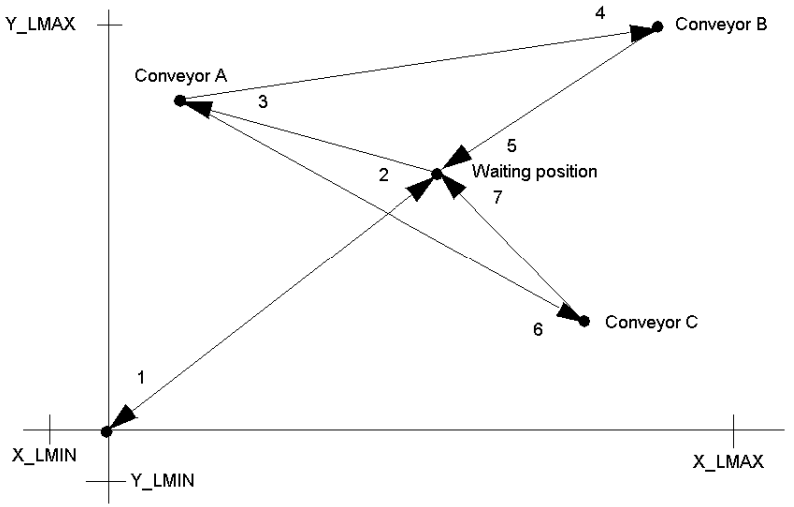
SFC of the application

The sequential graph of the application is as follows:



Trajectory description

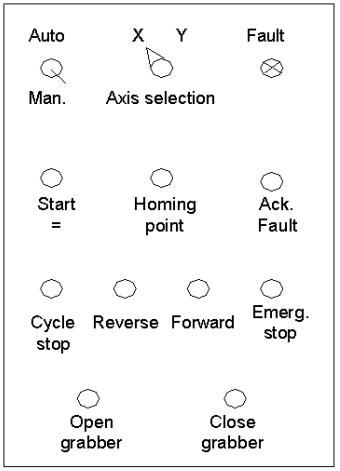
The following diagram shows the trajectory of the grab:



1 Reference point at speed Vp0,
2 Movement at speed Vret to the stand-by position (Xatt, Yatt) with stop,
3 Movement towards belt A (XA, YA) at speed VA, as far as detection of the machined part,
4 Movement towards conveyor B (XB, YB) at speed VB, with stop,
6 Movement towards conveyor C (XC, YC) at speed VC, with stop,
5, 7 Movement to waiting position (Xatt, Yatt) at speed Vret, with stop.

Front Panel of the Human-Machine Interface

The following controls grouped on a panel, allow you to drive the moving part manually when the installation has failed. The controls and the indicators are managed by a discrete input module and a discrete output module.



Auto/Man. Operating mode selection switch.
Starting cycle Executes the automatic cycle.
Stop cycle Stops the automatic cycle.
Select axis X / Y Selects axis to be driven in manual mode.
Reference point Manual reference point on the selected axis.
Forward / Back Control for manual movement of the selected axis in a positive or negative direction.
Fault Indicator of all hardware faults and application errors.
Ack. Fault Fault acknowledgement control.
Emergency stop Immediate stop of the moving part whatever the selected mode.
Open grab Control for opening grab.
Close grab Control for closing grab.

Prerequisites

Prerequisites

In order that only functions specific to axis control are described, it is assumed that the following operations have been carried out:

- Control Expert software is installed,
- The hardware installation has been carried out: modules, translators driving the 2 axes have been connected.

Configuring the TSX CFY module

Software declaration in the PLC configuration

Start the Control Expert software, select the command **File → New**, select a Premium processor.

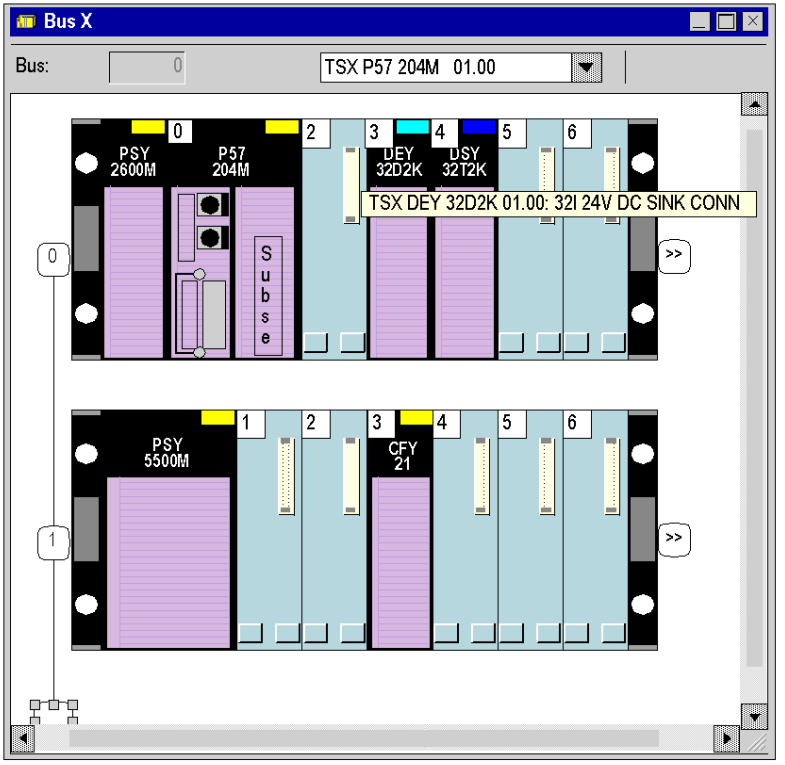
From within the **Project Browser**, access the configuration editor in the following way:

Step	Action
1	Open the folder **Station** (double click on the icon or click on its attachment).
2	Open the folder **Configuration** (double click on the icon or click on its attachment).
3	Double click on the icon **Bus X**.

You must then select each constituent of the PLC configuration. The following selections have been made in this application:

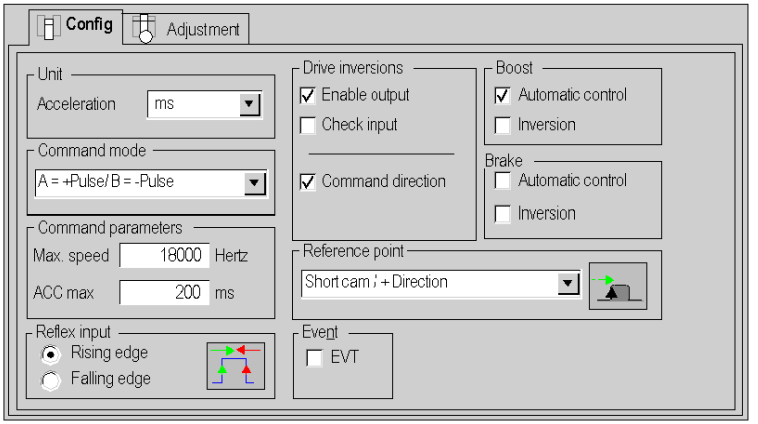
- rack 0 and rack 1: TSX RKY 8EX,
- processor: TSX P57 204,
- supply modules: TSX PSY 2600 for rack 0 and TSX PSY 5500 for rack 1,
- 32 inputs module: TSX DEY 32D2K in position 3 of rack 0,
- 32 outputs module: TSX DSY 32T2K in position 4 of rack 0,
- axes control module: TSX CFY 21 in position 3 in rack 1,

Example of a module configuration screen



Enter the axes configuration parameters.

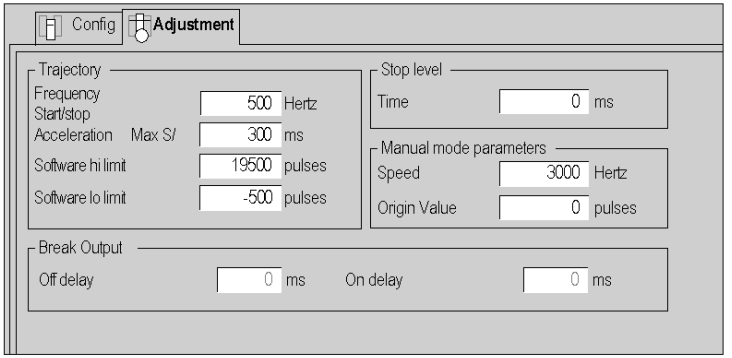
For each axis, enter the configuration parameters in the following way:

Step	Action
1	Select position 3 in rack 1 then execute the command **Edit → Open module** (or double click on the selected module).
2	Configure the parameters of channel 0. To do that: ● select channel 0, ● select the function **Positioning**, ● select the task **MAST**, ● enter the parameters, in accordance with the following screen: Configuration screen for channel 0 
3	Confirm your entries using the command **Edition → Confirm** or by clicking on the confirm icon.
4	Carry out the same configuration for channel 1 of the module by repeating the procedure from step 2 for channel 1.

Adjusting the TSX CFY module

Inputting axes adjustment parameters

For each axis, enter the configuration parameters in the following way:

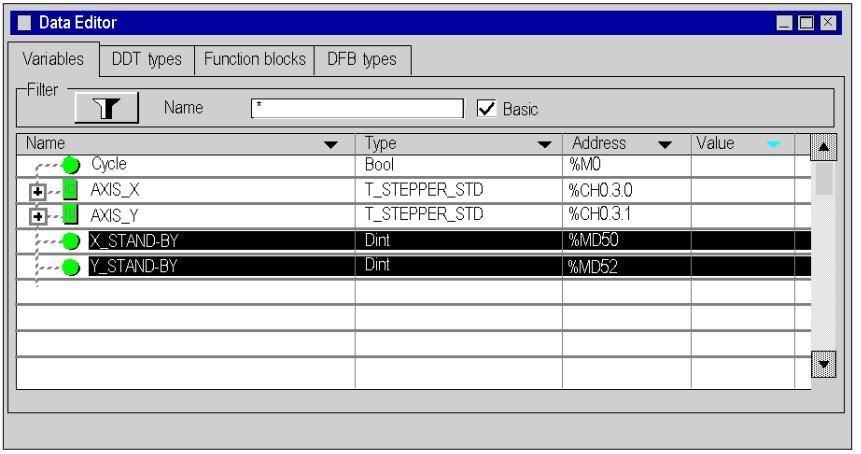
Step	Action
1	Select position 3 in rack 1 then execute the command **Edit → Open module** (or double click on the selected module).
2	Click on the **Adjustment** tab.
3	Configure the adjustment parameters for channel 0. To do this: ● select channel 0, ● enter the adjustment parameters, in line with the following screen: Adjustment screen for channel 0 
4	Confirm your entries using the command **Edition → Confirm** or by clicking on the confirm icon.
5	Carry out the same adjustments for channel 1 of the module by repeating the procedure from step 3 for channel 1
6	Then confirm the global configuration using the command **Edit → Confirm** or by clicking on the confirm icon

Symbolization of variables in the example

Entering variables

Access to symbols entry is by means of a double-click in the **Application browser**, then on the icons **Variables** and **Elementary variables** for example to obtain the following screen which permits the input of all the variables described in the following paragraphs.

Variables entry screen



Internal variables symbols

The following internal variables are symbolized:

Address	Symbol	Comment
%M0	CYCLE	Condition of the machine in work mode
%MD50	X_STAND-BY	Stand-by position (X axis)
%MD52	Y_STAND-BY	Stand-by position (Y axis)
%MD54	X_B	Position of belt B (X axis)
%MD56	Y_B	Position of belt B (Y axis)
%MD58	X_C	Position of belt C (X axis)
%MD60	Y_C	Position of belt C (Y axis)

Symbols for the discrete inputs module

The discrete inputs module is positioned in slot 3 in rack 0. Its symbols are as follows:

Address	Symbol	Comment
%I0.3.0	SENSOR_1	Detector to detect presence of machined part
%I0.3.1	SENSOR_2	Sensor to identify the part type (0 = type 2, 1 = type 1)
%I0.3.2	SENSOR_3	Sensor to detect claw open / claw closed
%I0.3.3	AUTO_MAN	Switch to select mode (0 = Auto, 1 = Manual)
%I0.3.4	START_CYCLE	Push button to execute automatic cycle
%I0.3.5	STOP_CYCLE	Push button to stop automatic cycle
I%I0.3.6	SELECTION_X_Y	Select axis to drive in Manual mode (1 = X, 0 = Y)
%I0.3.7	PO_MAN	Manual reference point
%I0.3.8	Forward	Move the moving part in the positive direction
%I0.3.9	BACK	Move the moving part in the negative direction
%I0.3.10	ACK_ERROR	Fault acknowledgment
%I0.3.12	EMERGENCY_STOP	Emergency stop
%I0.3.13	OPEN_CLAW	Push button to open claw
%I0.3.14	CLOSE_CLAW	Push button to close claw

Symbols for the discrete outputs module

The discrete outputs module is positioned in slot 4 in rack 0. Its symbols are as follows:

Address	Symbol	Comment
%Q0.40.0	CLAW	Command to open or close the claw (0 = Open, 1 = Close)
%Q0.4.1	Error	Signals an error

Internal constants

The speed of the moving part along the different axes is contained in the internal constants. In the case of 2 independent axes, the symbols and the values of these constants are as follows:

Address	Symbol	Value	Comment
%KD0	SPEED_O_C	5000	Speed from the reference point along the axis of the X and Y axes
%KD4	SPEED_X_WAIT	10000	Speed towards the X axis stand-by position
%KD6	SPEED_Y_WAIT	10000	Speed towards the Y axis stand-by position
%KD8	SPEED_POS_A_X	15000	Speed towards the X axis belt A position
%KD10	SPEED_POS_A_Y	15000	Speed towards the Y axis belt A position
%KD12	SPEED_POS_B_X	15000	Speed towards the X axis belt B position

Address	Symbol	Value	Comment
%KD14	SPEED_POS_B_Y	15000	Speed towards the Y axis belt B position
%KD16	SPEED_POS_C_X	12000	Speed towards the X axis belt C position
%KD18	SPEED_POS_C_Y	12000	Speed towards the Y axis belt C position

Symbols for the axes control module

The axes control module is positioned in slot 3 in rack 1. Its symbols are as follows:

Address	Symbol	Address	Symbol
%CH1.3.0	AXIS_X	%CH1.3.1	AXIS_Y
%I1.3.0.9	AT_POINT	%I103.1.9	AT_POINT_Y

Symbols of the axes control module linked to IODDT de type T_STEPPER_STD

The table below shows the T_STEPPER_STD type IODDT objects which are used for the two channels in the programming example:

Address	Standard symbol
%Ir.m.c.0	NEXT
%Ir.m.c.1	DONE
%Ir.m.c.2	AX_FLT
%Ir.m.c.3	AX_OK
%Ir.m.c.4	HD_ERR
%Ir.m.c.5	AX_ERR
%Ir.m.c.6	CMD_NOK
%Ir.m.c.11	CONF_OK
%Ir.m.c.12	REF_OK
%Ir.m.c.16	IN_DROFF
%Ir.m.c.17	IN_DIRDR
%Ir.m.c.18	IN_MANU
%Ir.m.c.19	IN_AUTO
%Ir.m.c.35	ST_DRIVE
%Qr.m.c.0	DIRDRV
%Qr.m.c.1	JOG_P
%Qr.m.c.2	JOG_M
%Qr.m.c.3	INC_P
%Qr.m.c.4	INC_M
%Qr.m.c.5	SET_RP

Address	Standard symbol
%Qr.m.c.6	RP_HERE
%Qr.m.c.9	ACK_FLT
%Qr.m.c.10	ENABLE
%Qr.m.c.11	EXT_EVT
%MDr.m.c.22	RP_POS

Programming preliminary processing

Introduction

Preliminary processing is a section at the beginning of the program that manages the operating modes:

With a blocking error:

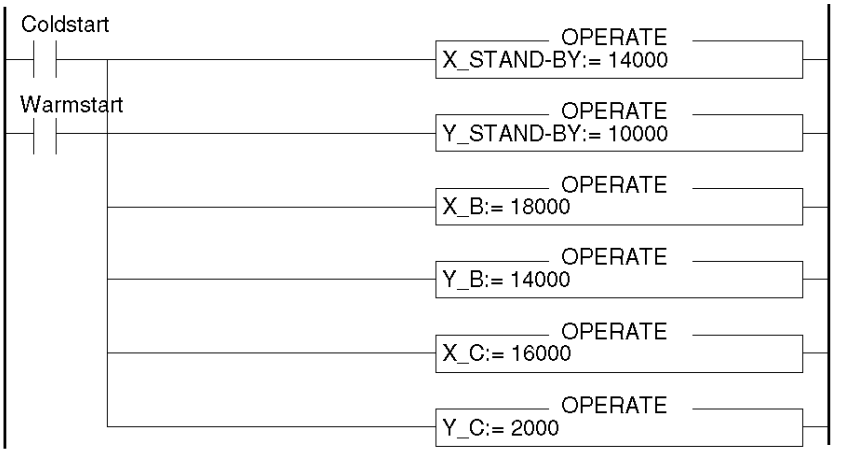
- The chart is disabled.
- You can then drive the moving part in Manual mode, fix the error and acknowledge it from the panel.
- The chart is reinitialized when the error is fixed and acknowledged.

On switching to Manual mode:

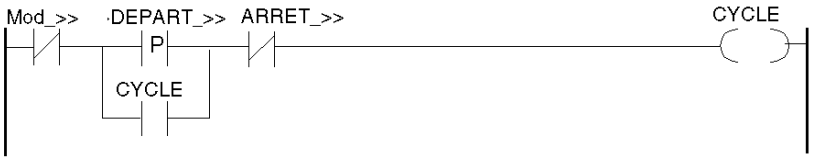
- The chart is disabled.
- The chart is reinitialized when Automatic mode is selected again.

Program in Ladder language

Initializing positions

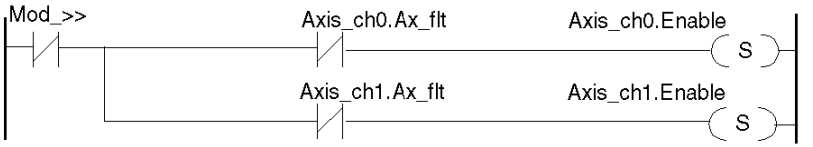


Starting cycle



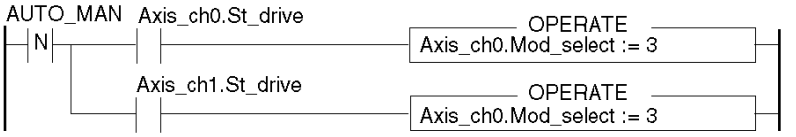
Mod_>> = Mod_error
DEPART_>> = DEPART_CYCLE
ARRET_>> = ARRET_CYCLE

Confirming controller



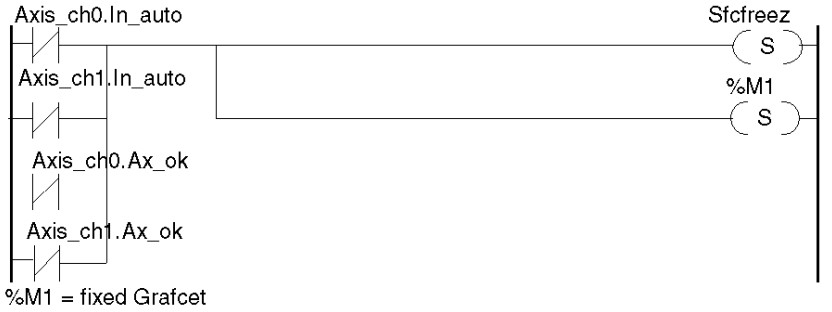
Mod_>> = Mod_error

Selecting automatic mode

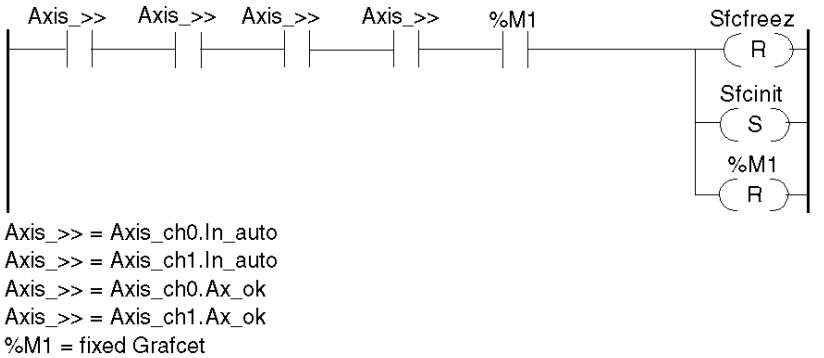


Selecting manual mode

Disabling chart on error or when switching to manual mode



Reinitializing chart



Signaling errors

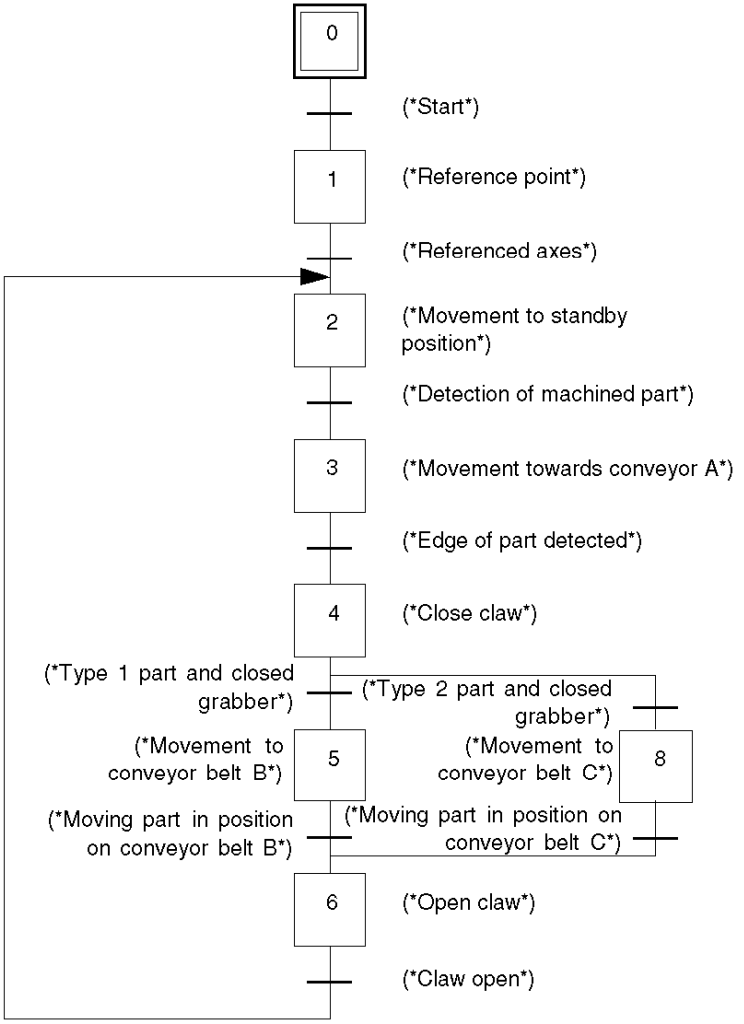
SFC Programming

At a Glance

The SFC allows you to program the sequential handling of the application: automatic cycle processing.

Sequential processing

Representation of sequential processing:



Programming transitions

At a Glance

Transitions drawn in the Grafcet are programmed in the following way:

Step 0 -> 1

! (*X channel not in error, claw open, Auto_man switch set to Auto, Y channel not in error and automatic mode activated*)

```
NOT Axis_ch0.Ax_flt AND NOT Capteur_3 AND NOT Auto_man AND Cycle AND NOT
Axis_ch1.Ax_flt AND Mode_Auto
```

Step 1 -> 2

! (*Test: axes ready and referenced*)

```
Axis_ch0.Done AND Axis_ch0.Ref_OK AND Axis_ch1.Done AND Axis_ch1.Ref_OK
```

Step 2 -> 3

! (*Moving part in stand-by position and part detected on belt A*)

```
Capteur_1 AND Cycle AND Axis_ch0.Next AND Axis_ch1.Next
```

Step 3 -> 4

! (*Moving part in position to take part detected on belt A*)

```
Axis_ch0.At_point AND Axis_ch0.Next AND Axis_ch1.Next AND
Axis_ch1.At_point
```

Step 4 -> 5

! (*Type 1 part and claw closed*)

```
Capteur_2 AND Capteur_3
```

Step 4 -> 8

! (*Type 2 part and claw closed*)

```
NOT Capteur_2 AND Capteur_3
```

Step 5 -> 6

! (*Moving part in position on conveyor belt B*)

```
Axis_ch0.At_point AND Axis_ch0.Next AND Axis_ch1.Next AND
Axis_ch1.At_point
```

Step 8 -> 6

! (*Moving part in position on conveyor belt C*)

```
Axis_ch0.At_point AND Axis_ch0.Next AND Axis_ch1.Next AND
Axis_ch1.At_point
```

Step 6 -> 2

! (*Claw open*)

```
NOT Capteur_3 AND Cycle
```

Programming actions

At a Glance

In Grafcet it is possible to program actions for each step. Three types of action are possible:

- on activation
- continuous
- on deactivation

When for a given step the type of action is not described, this means that it has not been programmed.

Step 1: Action on activation

! (*Reference point taken along the X and Y axes*)

```
SMOVE (Axis_ch0, 1, 90, 14, 0, Vitesse_p_o, 16#0000);
SMOVE (Axis_ch1, 1, 90, 14, 0, Vitesse_p_o, 16#0000);
```

Step 2: Action on activation

! (*Movement to stand-by position (Xatt, Yatt*)

```
SMOVE (Axis_ch0, 2, 90, 9, X_attente, Vitesse_x_attente, 16#0000);
SMOVE (Axis_ch1, 2, 90, 9, Y_attente, Vitesse_y_attente, 16#0000);
```

Step 3: Action on activation

! (*Movement towards belt A*)

```
SMOVE (Axis_ch0, 3, 90, 10, 19500, Vitesse_pos_a_x, 16#0000);
SMOVE (Axis_ch1, 3, 90, 10, 19500, Vitesse_pos_a_y, 16#0000);
```

Step 4: Continuous action

! (*Close claw*)

```
SET(Claw);
```

Step 5: Action on activation

! (*Movement to conveyor belt B*)

```
SMOVE (Axis_ch0, 4, 90, 9, X_b, Vitesse_pos_b_x, 16#0000);
SMOVE (Axis_ch1, 4, 90, 9, Y_b, Vitesse_pos_b_y, 16#0000);
```

Step 8: Action on activation

```
! (*Movement to conveyor belt C*)
SMOVE (Axis_ch0, 5, 90, 9, X_c, Vitesse_pos_c_x, 16#0000);
SMOVE (Axis_ch1, 5, 90, 9, Y_c, Vitesse_pos_c_y, 16#0000);
```

Step 6: Continuous action

```
! (*Open claw*)
RESET (Claw);
```

Programming subsequent processing

Introduction

Subsequent processing is a section at the end of the program that manages the manual operating mode.

MAST - POST

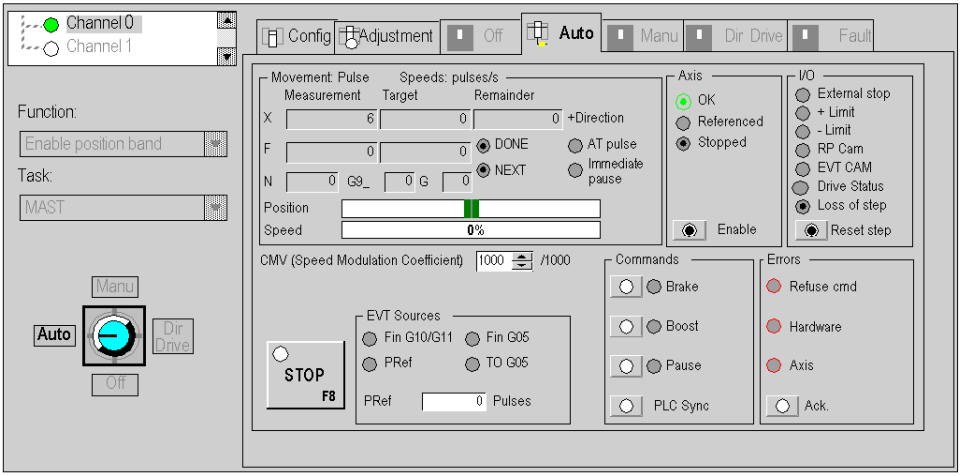
```
! (*Test selected mode*)
IF Mode_auto AND Mode_auto_y AND Axis_ch0.Conf_ok AND Axis_ch1.Conf_ok
THEN JUMP %L200;
END_IF;
! (*Selecting the axis to be driven*)
%L100: IF NOT Selection_x_y
THEN JUMP %L200;
END_IF;
! (*Reference point taken manually on X axis*)
IF RE(Po_man)
THEN SET(Axis_ch0.Set_rp);
END_IF;
IF NOT Po_man
THEN RESET(Axis_ch0.Set_rp);
END_IF;
! (*Movement of the moving part in the + direction along the X axis*)
Jog_p := Front;
! (*Movement of the moving part in the - direction along the X axis*)
Jog_m := Back;
%L200: IF selection_x_y
THEN JUMP %L300;
END_IF;
! (*Reference point taken manually on Y axis*)
IF RE(Po_man)
THEN SET(Axis_ch1.Set_rp);
END_IF;
IF NOT Po_man
```

```
THEN RESET(Axis_ch1.Set_rp);
END_IF;
! (*Movement of the moving part in the + direction along the Y axis*)
Axis_ch1.Jog_p := Front;
! (*Movement of the moving part in the - direction along the Y axis*)
Axis_ch1.Jog_p := Back;
! (*Open claw*)
%L300: IF Auto_man AND Ouv_pince
THEN RESET(Claw);
END_IF;
! (*Closure of the claw*)
IF Auto_man AND Ferm_pince
THEN SET(Claw);
END_IF;
! (*Error acknowledgement*)
Axis_ch0.Ack_def := Axis_ch1.Ack_def := Acq_defauts;
%L999:
```

Taking over in manual mode

Accessing manual mode

If you want to move the moving part without first going through the programming phase, use Manual mode. To do that, access the debug screen, in connected mode:

Step	Action
1	Activate the command **Tools → Configuration**.
2	Select the TSX CFY module to be opened.
3	Execute the command **Services → Open the module** (or double click on the module to be opened).
4	The following debug screen is then displayed: 

Movements in manual mode

To move the moving part in Manual mode, you must carry out the following operations:

Step	Action
1	Put the PLC into RUN mode, using the command **AP → Run** or by clicking on the icon 
2	Select the axis to drive: channel 0 (X axis) or channel 1 (Y axis).
3	Select Manual mode by turning the switch to **Man.** mode.
4	Confirm the safety relays of the speed controller by clicking on the button **Confirmation** in the **Axis** box.
5	Acknowledge the errors by clicking on the button **Ack** in the **Errors** box.
6	Take a manual origin: • either by selecting the command **Manual reference point**, • or by selecting the command **Forced reference point**. In this case, first enter in the **Param** field, the value of the position of the moving part in relation to the origin.
7	Move the moving part: • in the positive direction by using the command **JOG+**, • or in the negative direction by using the command **JOG-**. The position of the moving part is displayed in the field **X** and the speed is displayed in the field **F** in the **Movement / Speed** box.

Debugging

Debugging procedure

You can debug the program by proceeding in the following manner:

Step	Action
1	Put the PLC into RUN mode.
2	Display the debugging screen for the TSX CFY module.
3	Simultaneously display the Grafcet screen in order to follow the evolution of the sequential processing.
4	Start the program by pressing the **Start_cycle** button on the panel.

Part II
TSX CFY axis control modules

Aim of this part

This part presents the TSX CFY step by step axis command modules, their functionalities and their installation.

What Is in This Part?

This part contains the following chapters:

Chapter 5
Installation

Subject of this chapter

This Chapter deals with the installation of step-by-step axis command modules.

What Is in This Chapter?

This chapter contains the following sections:

Section 5.1
General

Subject of this section

This Section introduces general points for installing TSX CFY modules.

What Is in This Section?

This section contains the following topics:

Base configuration necessary

Introduction

The step by step axes control modules can be installed in any of the slots on a TSX RKY•• rack. The power of the supply to the rack must be selected according to the number of modules fitted.

Maximum number of TSX CFY •1 modules per station

Each step by step control module comprises:

- 1 application specific channel for the TSX CFY 11 module,
- 2 application specific channels for the TSX CFY 21 module.

Given that the maximum number of application specific channels managed by a PLC station depends on the processor type installed, the maximum number of TSX CFY •1 modules in a PLC station will depend on:

- the processor type installed,
- the number of application specific channels already in use other than step by step application specific control channels.

As a result, a global assessment must be made at PLC station level to find out the number of application specific channels already in use, in order to define the number of TSX CFY •1 modules that can be used.

Number of "application-specific" channels supported:

- Premium *(see Premium and Atrium using EcoStruxure™ Control Expert, Processors, racks and power supply modules, Implementation Manual)*
- Atrium *(see Premium and Atrium using EcoStruxure™ Control Expert, Processors, racks and power supply modules, Implementation Manual)*

Installation procedure

General

The module can be installed or removed without cutting the rack power supply, to ensure that a device is available.

⚠ CAUTION
POSSIBLE DAMAGE TO TRANSLATORS Do not connect or disconnect Translators connectors with Translators powered on. It is not recommended, though allowed, to disconnect Auxiliary modules input/output connectors with modules powered on. **Failure to follow these instructions can result in injury or equipment damage.**

The module fixing screws and connectors must be correctly screwed in place in order to guarantee effective resistance to electrostatic and electromagnetic interference.

General precautions for wiring

General

The power supply to sensors and actuators need fast-blow fuses against overload or overvoltage.

- when wiring use wires of sufficient size to avoid on-line voltage falls and overheating,
- keep sensor and actuator cables away from any source of radiation resulting from high-power electric circuit switching,
- all cables connecting the translators must be shielded, the shielding must be good quality and connected to the protective ground both for the module and the translator. Continuity must be ensured throughout connections. Do not transmit any other signals in the cable than those for the translators.

For reasons of performance the auxiliary inputs of the module have a short response time. You must therefore make sure that there is enough self-sufficient supply to these inputs to ensure the module continues to operate correctly in the event of a short power break. It is recommended that you use a regulated supply to ensure more reliable response times from the actuators and sensors. The 0 V supply must be connected to the protective ground nearest to the supply module output.

Section 5.2
Connection of translator signals

Subject of this section

This Section deals with the connection of translator signals.

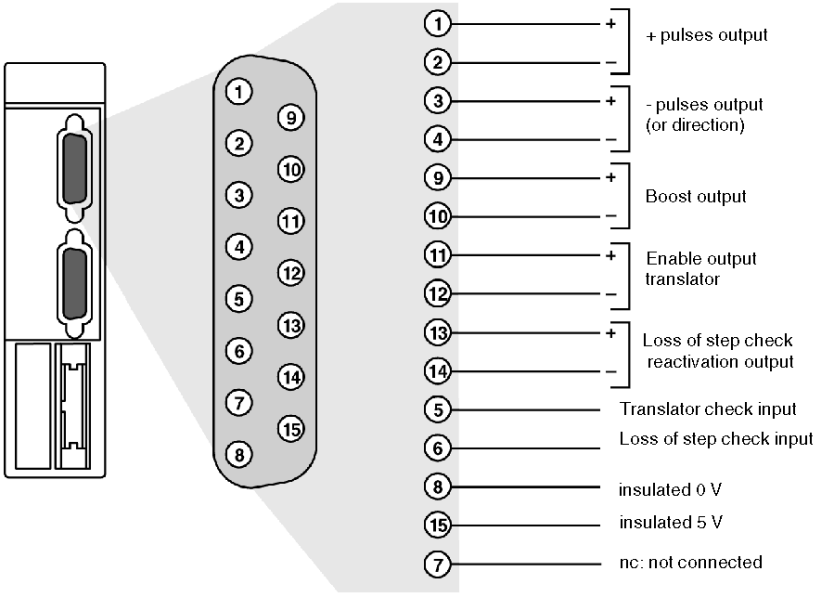
What Is in This Section?

This section contains the following topics:

Signal labeling

Diagram of the principle

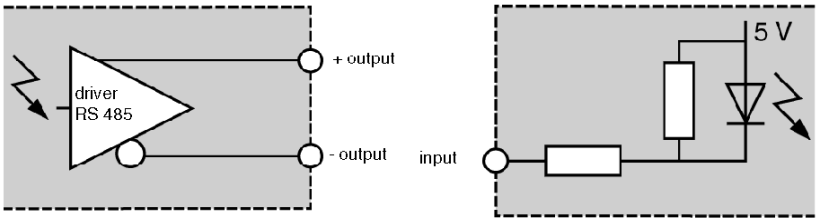
This diagram illustrates the principles for labeling:



Description

Each module output signal is RS 485, for each output there is therefore a direct signal (+) and its complement (-). The outputs are TTL type current extraction compatible. The 5 V insulated voltage is only available, if necessary, to supply translator input and output. The 0 V is common to inputs and outputs. 5 V must only be used with translators with open collector outputs and TTL type inputs (5 V insulated not provided by the translator).

Illustration:



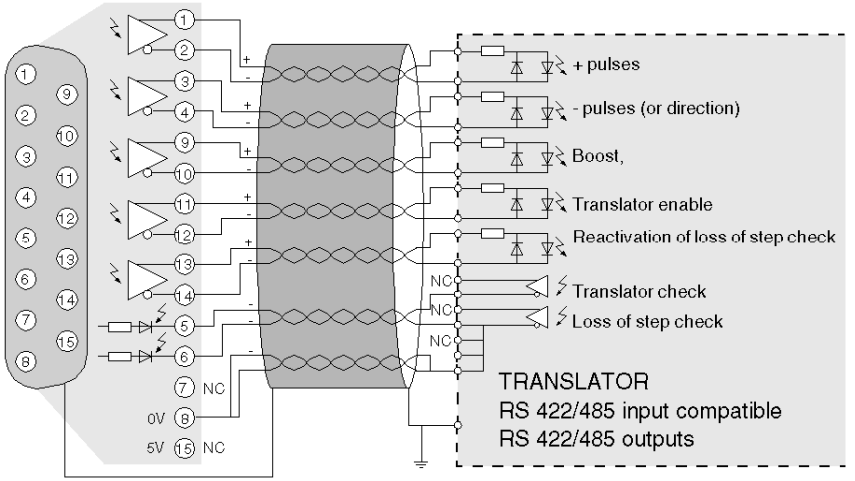
The proposed type of connection is direct wiring by soldering onto the connector: the TSX CAP S15 *(see page 68)* kit comprising a SUB-D connector and its protective cover.

Connection to a translator with RS 422/485 interface

Diagram of the principle

It is recommended that you use a shielded cable containing 7 twisted pairs. The + and – wires of each module output signal must be connected in the same pair.

This diagram illustrates the principle of connection:

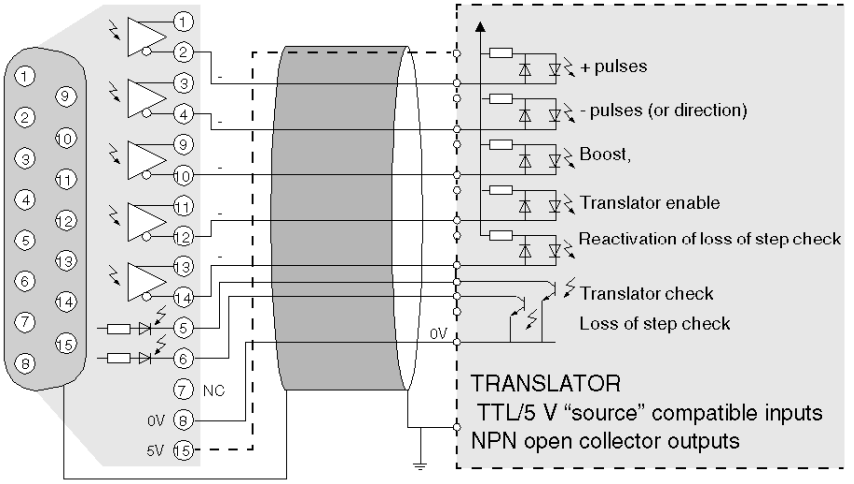


Connecting to a translator with NPN open collector interface

Diagram of the principle

Only one wire is used per input/output signal. If the translator does not provide 5 V insulated voltage do not forget to supply the interface from the insulated 5 V provided by the module.

This diagram illustrates the principle of connection:



Overview of the TSX TAP S15xx wiring accessories

General

The TSX TAP S15•• wiring accessories make it possible to connect an incremental encoder to the counting module, by using a specific cable (supplied by the encoder manufacturer):

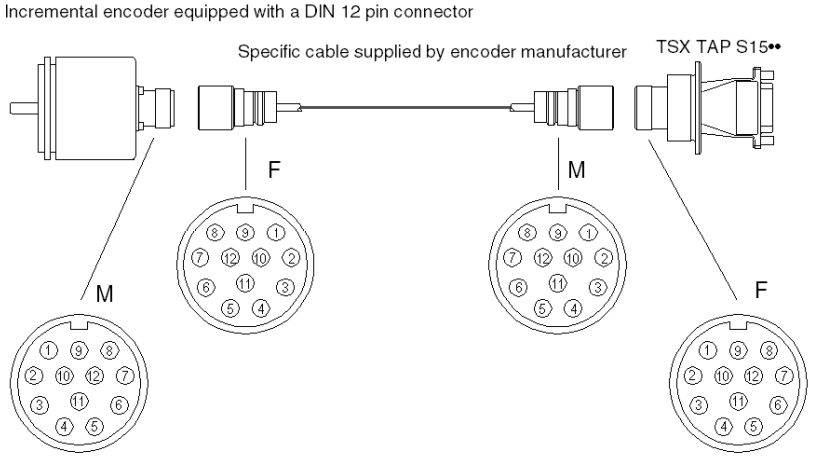
- TSX TAP S15 05: makes it possible to connect an incremental encoder with a 5 VDC supply: encoder with RS 422 line issuer outputs,
- TSX TAP S15 24: makes it possible to connect an incremental encoder with a 24 VDC supply: encoder with totem pole outputs or open collector PNP outputs.

The TSX TAP S15•• has 2 connectors:

- a female 12-pin DIN base, labeled in an anti-clockwise direction. This connector makes it possible to connect the encoder, via a cable supplied by the manufacturer of the encoder,
- a standard 15-pin SUB-D connector making it possible to connect the module counting inputs to the SUB-D connector, using a standard TSX CCP S15 cable.

The TSX TAP S15•• product can be fixed onto a DIN rail using a bracket supplied with the accessories, or it can be fixed to a cabinet lead-in with a gasket, which is supplied with the product.

Illustration:



Section 5.3
Connecting sensors/pre-actuators and supply modules

Subject of this section

This Section deals with the connection of sensors/pre-actuators and supply modules.

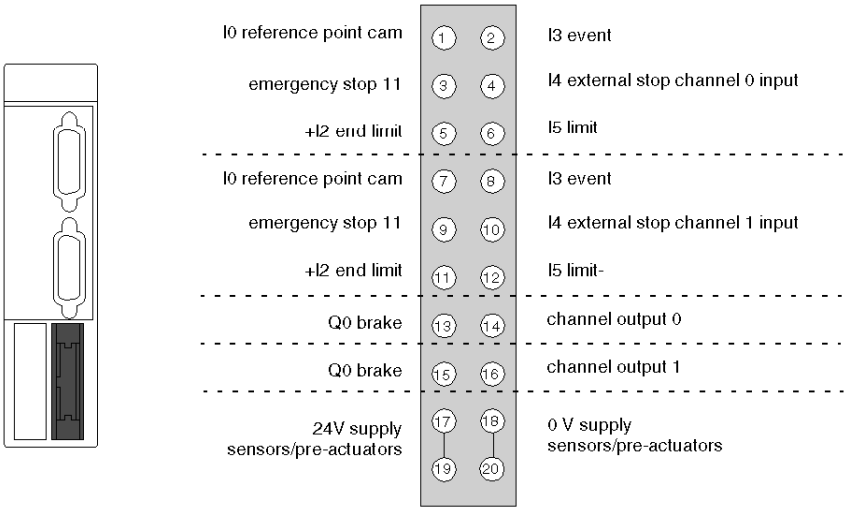
What Is in This Section?

This section contains the following topics:

Signal labeling

Diagram of the principle

This diagram illustrates the principles for labeling signals:



The 0 V of sensors/pre-actuators is connected in the module to the protective ground by an R/C network with the value: R = 100MΩ / C = 4.7 nF.

Connections

General

There are several possible options for connecting the sensors/pre-actuators of the TSX CFY 11 / 21 module. They can be connected directly by the TSX CDP 301 / 501 *(see Premium and Atrium using EcoStruxure™ Control Expert, Axis Control Modules for Servomotors, User Manual)* strip or via the Discrete TELEFAST pre-wiring system.

Connecting auxiliary inputs and outputs to processor.

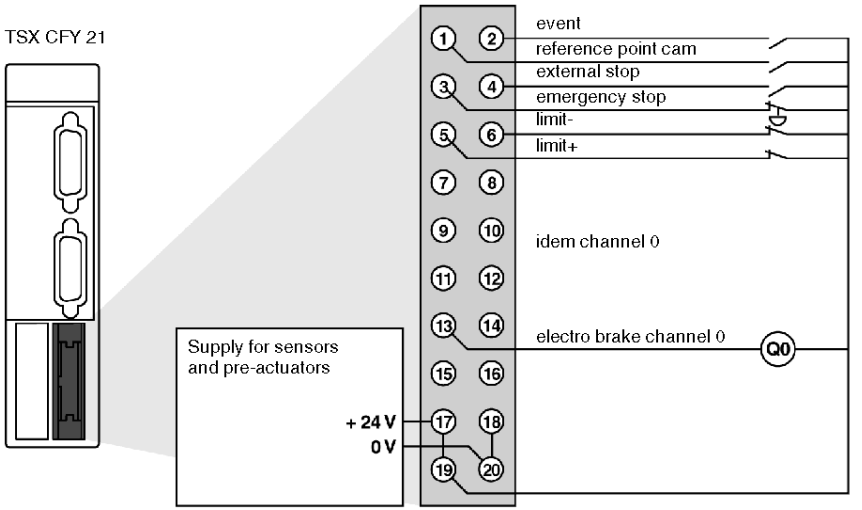
General

To ensure optimum operation, the event and reference point inputs have a weak immunity. It is recommended that you use contacts without bounce (proximity sensor for example).

Principle of connecting I/O channel 0

Diagram of the principle

This diagram illustrates the principles for connecting I/O channel 0:

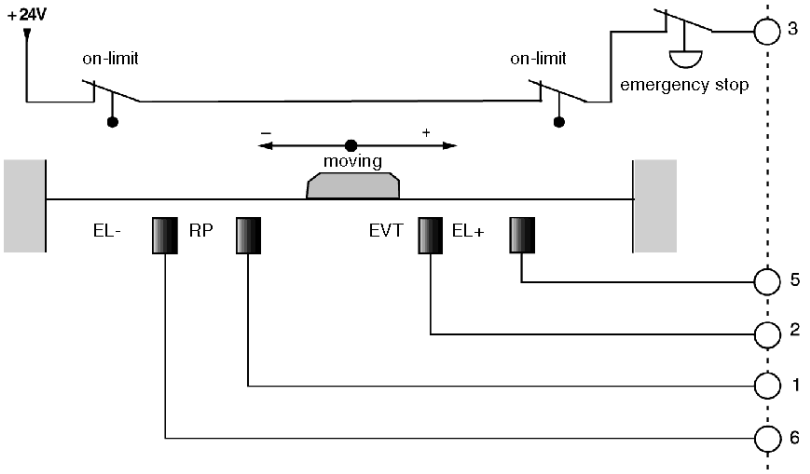


Description

The emergency stop or end limit switch contacts are open.

End limit switch contacts are not contacts which have to be wired in series with the emergency input. End limit switch contacts are used to command movement stop with deceleration. The end limit switch (ELS+) stops movement in the +direction, the end limit switch (ELS-) stops movement in the- direction. It is therefore important to positions them at the correct end of the axis (see diagram below).

Illustration:



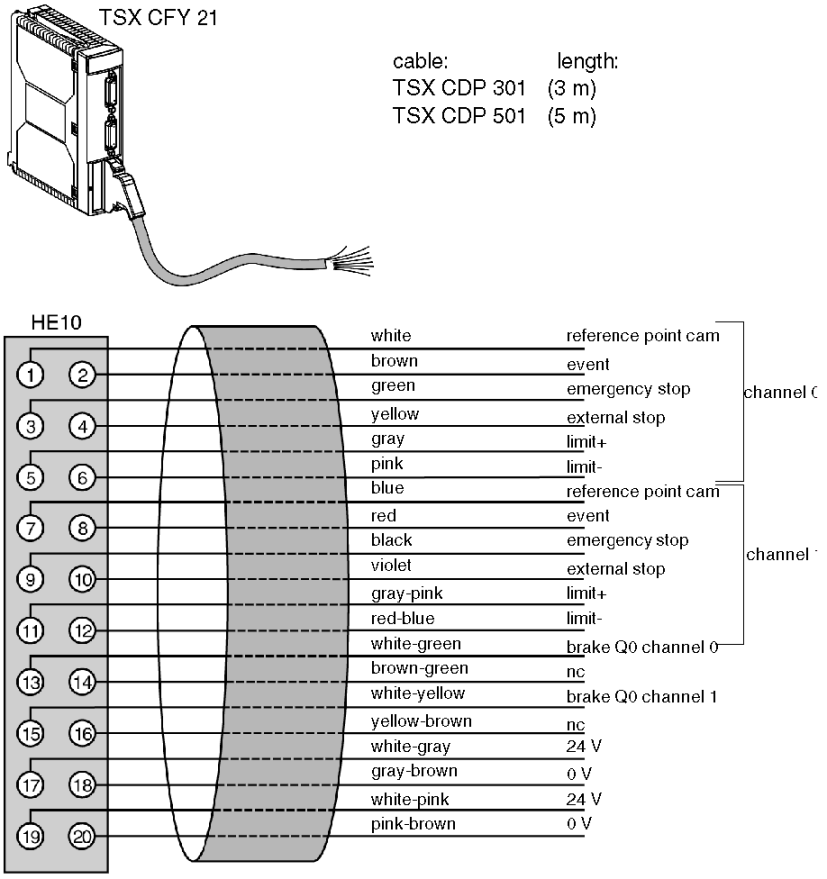
Connection using a TSX CDP 301/501 pre-wired strand

General

Connection using a pre-wired strand enables direct contact to the actuators, pre-actuators or any terminal system. This strand comprises 20 gage 22 wires (0.34 mm^2) with a connector at one end and free wires at the other end, labeled using a color code.

Illustration

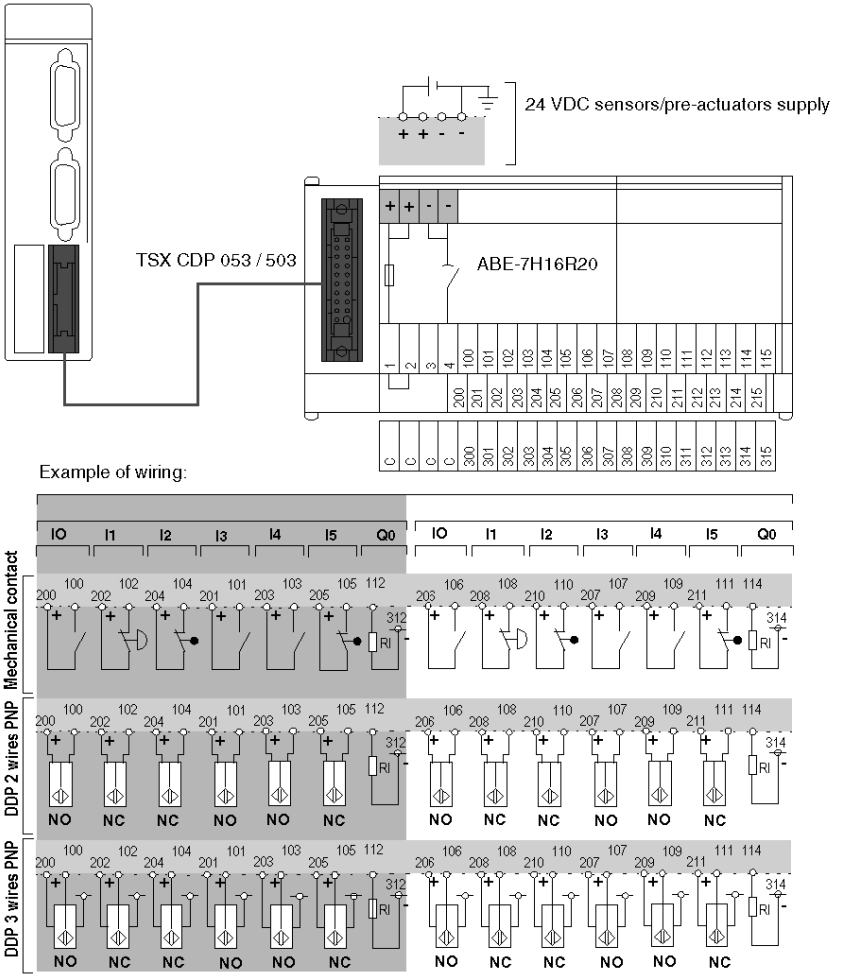
This diagram shows the color code:



Connection with TELEFAST pre-wiring system

Diagram of the principle

This connection is made using a TELEFAST 2 base: ABE-7H16R20.



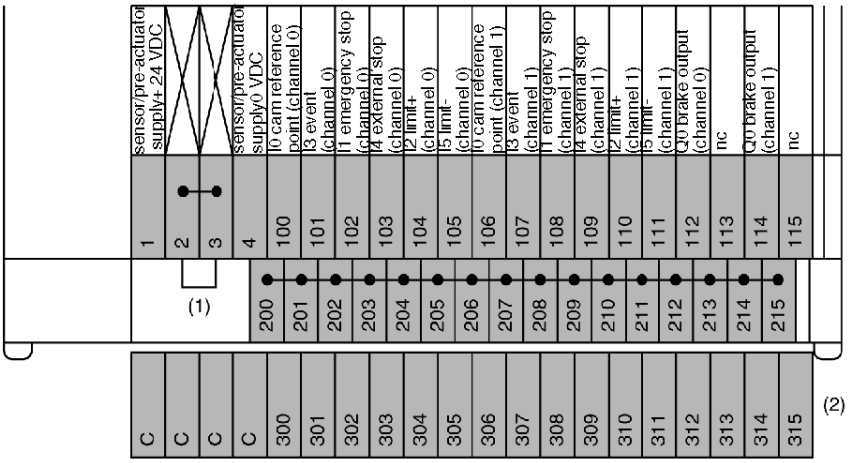
NO: normally open,

NC: normally conductor.

Availability of signals on TELEFAST

Illustration

This diagram shows the availability of signals on TELEFAST:



(1) At the ABE-7H16R20 base, 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 + polarity,
- jumper wire in position 3 or 4: terminals 200 to 215 have - polarity,

(2) At the ABE-7H16R20 base, it is possible to add an optional ABE-7BV20 strip to create a second shared sensor (+ or - according to user's choice).

Correspondence between TELEFAST terminals and HE10 connector

General

This table shows the correspondence between TELEFAST terminals and the module's HE10 connector:

TELEFAST screw terminal block (Terminal No.)	HE10 20-pin connector (Pin No.)	Nature of signal	
100	1	I0 cam reference point	channel 0
101	2	I3 event	
102	3	I1 emergency stop	
103	4	I4 external stop	
104	5	I2 limit	
105	6	I5 limit	
106	7	I0 cam reference point	channel 1
107	8	I3 event	
108	9	I1 emergency stop	
109	10	I4 external stop	
110	11	I2 limit+	
111	12	I5 limit-	
112	13	Q0 brake output	channel 0
113	14	nc	
114	15	Q0 brake output	channel 1
115	16	nc (1)	
+ 24 VDC	17	Auxiliary input sensor supply	
- 0 VDC	18		
+ 24 VDC	19		
- 0 VDC	20		
1		Terminals 200 to 215 at + 24 VDC	
2			
3		Terminals 200 to 215 at - 0 VDC	
4			

TELEFAST screw terminal block (Terminal No.)	HE10 20-pin connector (Pin No.)	Nature of signal
200...215		Connecting shared sensors to: ● + 24 VDC if terminals 1 and 2 are connected, ● - 0 VDC if terminals 3 and 4 are connected
300...315		On the optional ABE-7BV20 bar, the terminals that can be used as a shared sensor must be connected by a wire to the shared voltage.

(1) nc = not connected

For a TSX CFY 11 module, the signals corresponding to the channel are not connected.

Wiring precautions

General

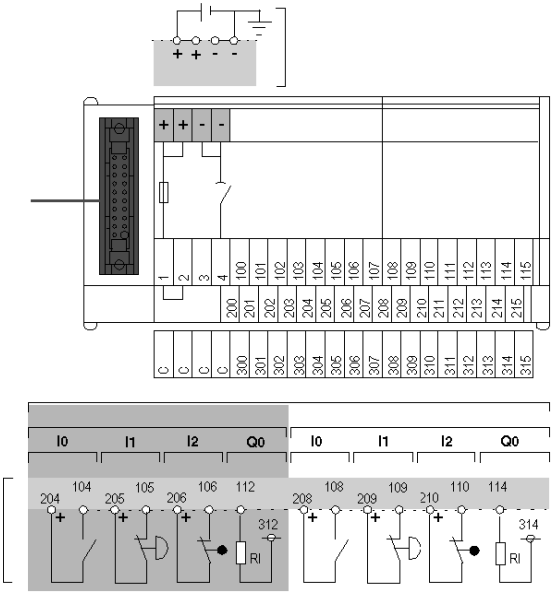
To ensure the best performance, inputs I0 to I5 are rapid inputs. If the actuator is a dry contact, the inputs must be connected by a twisted pair, or by a shielded cable if the sensor is a two or three-wire proximity detector.

The module includes as standard basic protection against short circuits or voltage inversions. However, the module cannot remain operational for long with an error. You must therefore ensure that the fuses in series with the supply carry out their protective function. These are 1A maximum fast-blow fuses, the supply energy must be sufficient to ensure their fusion.

Important note: wiring of Q0 static outputs

The actuator connected to the Q0 brake output has its shared pin connected to supply 0 V. If for any reason there is a 0 V outage of the output amplifier supply (e.g. poor contact or accidental unplugging), when the 0 V of the actuators remains connected to the 0 V supply, there may be enough mA output current from the amplifier to keep low-power actuators triggered.

Illustration:



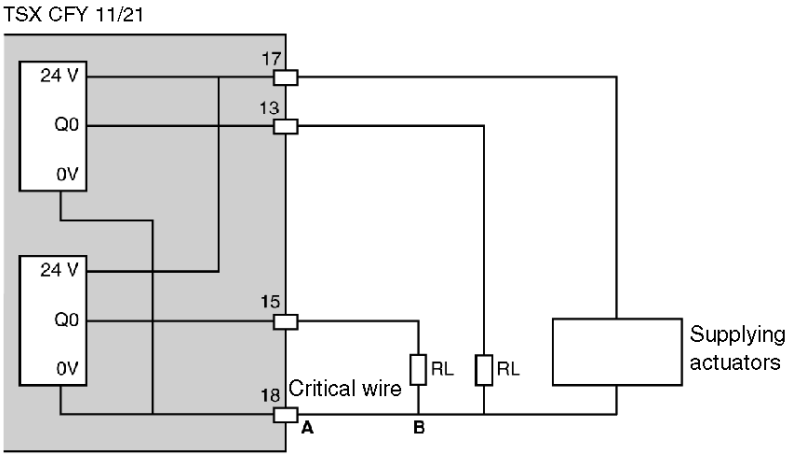
Connection via TELEFAST

If the shared actuators are connected to the bar for shared points 200 to 215 (jumper wire in position 1-2), there can be no outage of the shared module without an outage of the shared actuators.

Connection using a TSX CDP 301 / 501 pre-wired strand

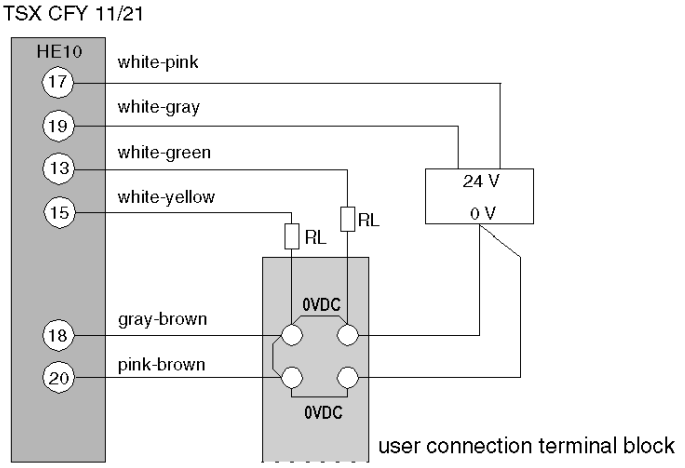
This kind of connection must be carried out with the greatest care and attention. It is recommended that you take special care in wiring this cable, for example using the cable ferules on screw terminals. It may be necessary to double the connections in order to ensure permanent contacts. When the actuator supply is a long distance away from the modules and close to the shared actuators, there may be an accidental break of the link between the latter and the 0 V terminal of the module(s).

Illustration:



If there is a break of the supply section between A and B, there is a risk that the RL actuators may not remain operational. You must, if possible, double connections of 0 V supply to the modules.

Connection using a TSX CDP 301 / 501 pre-wired strand:



Chapter 6
Characteristics and maintenance of the TSX CFY

Aim of this part

This part presents the various electrical characteristics of TSX CFY modules and describes the maintenance actions to be performed to ensure that the module operates correctly.

What Is in This Chapter?

This chapter contains the following topics:

General characteristics

Table of characteristics

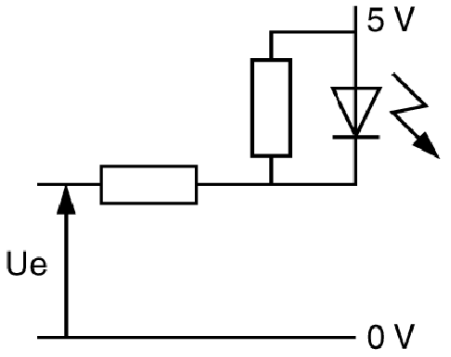
This table introduces the general characteristics of TSX CFY modules:

Maximum frequency of pulses		187.316 KHz
Current consumed on internal 5 V	**Module**	**Value**
	TSX CFY 11 TSX CFY 21	510 mA 650 mA
Current consumed by the module on sensor/pre-actuator 24V with no sensor/pre-actuator current	TSX CFY 11 TSX CFY 21	50 mA 100 mA
Power dissipated in the module	TSX CFY 11 TSX CFY 21	3.8 W 5.6 W
Insulation resistance	> 10 MΩ under 500 VDC	
Dielectric rigidity between I/O "translator" and protective ground or PLC logic	1000 Veff 50 / 60 Hz for 1 mn	
Operating temperature	0 to 60 °C	
Storage temperature	-25 °C to 70 °C	
Hygrometry (without condensation)	5% to 95%	
Operating altitude	< 2000 m	

Characteristics of translator inputs (SUB-D connector)

Diagram

These inputs have positive logic current extraction:



Characteristics

The following table shows the characteristics of translator inputs:

Characteristics	Symbol	Value	Unit
Nominal current (Ue = 0 V)	Ie	4,5	mA
Voltage for ON state	Uon	2	V
Voltage for OFF state	Uoff	3,6	V
Immunity of loss of step input		15 to 30	µs
Immunity of translator error input		3 to 10	ms

Characteristics of translator outputs (SUB-D connector)

Table of characteristics

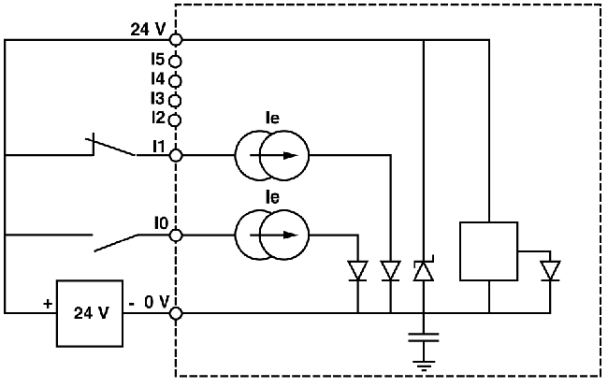
These outputs are insulated RS 422/485. There are two complemented outputs per signal.

Characteristics	Values	Units
Differential voltage output on R load ≤ 100Ω	+/- 2	V
Short-circuit current	< 150	mA
Permitted shared mode voltage	≤ 7	V
Permitted differential voltage	≤ 12	V

Characteristics of auxiliary inputs (HE10 connector)

Illustration

Diagram:



Characteristics

Table of characteristics for auxiliary inputs:

Electrical characteristics	Symbol	Value	Unit
Nominal voltage	Un	24	V
Nominal voltage limits (ripple included)	U1 Utime (1)	19 to 30 34	V
Nominal current	In	7	mA
Input impedance (at Unom)	Re	3,4	kΩ
Voltage for "On" state	Uon	≥11	V
Current at Uon (11 V)	Ion	>6	mA
Voltage for "Off" state	Uoff	<5	V
Current at "Off" state	Ioff	<2	mA
Input immunity: **Cam reference point input and event** **Other inputs**	 ton/toff (2) ton/toff	 < 250 3 to 10	 µs ms
IEC 1131 compatibility with sensors	type 2		
Compatibility with 2 and 3-wire sensors	all proximity sensors supplied at 24 VDC		
Type of input	current ducts		
Logic type	Positive (sink)		

Electrical characteristics		Symbol	Value	Unit
Pre-actuator voltage check	supply threshold OK		> 18	V
	supply threshold error		< 14	V
Time of supply detection	supply OK		< 30	ms
	supply error		> 1	ms

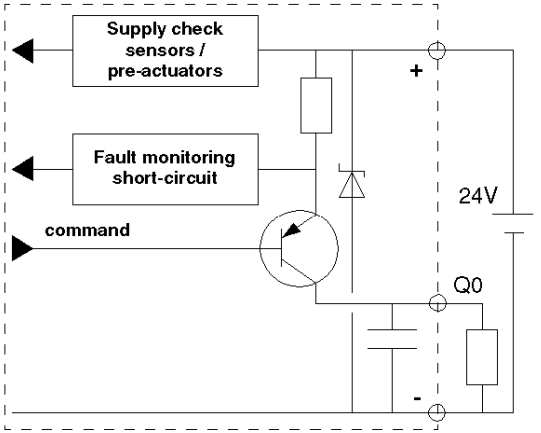
(1) Utime: maximum permitted voltage for 1 hour in every 24 hours.

(2) Inputs: reference point cam and events are fast inputs (response time < 250 µs) compliant with maximum frequency of 187.316 KHz of translator command outputs.

Characteristics of Q0 brake output

Illustration

Brake output:



Characteristics

Table of characteristics:

Electrical characteristics	Value	Unit
Nominal voltage	24	V
Voltage limits Temporary voltage	19 to 30 34 (1)	V V
Nominal current	500	mA
Max voltage fall "On"	< 1	V
Leakage current at "OFF" state	< 0,3	mA
Load impedance	80<Zon<1500	Ω
Max current to 30 V and to 34 V	625	mA
Communication time	< 250	µs
Electro unload time	< L/R	s
Max switching frequency (on inductive load)	F<0.6 / (LI^2)	Hz
Compatibility with inductive inputs	Any input whose Re is less than 15 kΩ and has positive logic	
IEC 1131 compatibility	Yes	

Electrical characteristics	Value	Unit
Protection from overloads and short-circuits	by current limiter and circuit breaker	
Monitoring short-circuits of each channel	thermal, signaling: 1 bit per channel	
Reset ● via application program ● automatic	One bit per module	
Protection against channel overvoltage	Zener (55 V) between outputs and +24 V	
Protection against polarity inversions	Using a reverse diode on the supply	
Power of a filament lamp	8	W
Pre-actuator voltage check	OK if supply > 18 (ascending) not OK if supply < 14 (decreasing)	V V
Reaction time of voltage check	NOK --> OK<30 OK --> NOK>1	ms ms

(1) maximum permitted voltage for 1 hour in every 24 hours of operation.

Chapter 7
Programming step by step axes control

Subject of this chapter

This chapter describes the programming principles for the different operating modes: it describes the principal instructions and the operating modes.

What Is in This Chapter?

This chapter contains the following topics:

Principles of programming a step by step axis

Introduction

Each channel (axis) of the axes control module is programmed using:

- the **SMOVE function** for movements in automatic mode,
- the **bit objects** (%I and %Q) and **words** (%IW, %QW and %MW), (See *Application-specific step by step axes control language objects, page 209* associated with the module to define:
  - the selection of operating modes,
  - the control of movements, except for automatic mode,
  - the monitoring of the operating status of the axis and the module.

Bit objects and words

The bit objects and words can be retrieved using their address or their symbol. The symbols are defined in the variables editor which offers as a default a symbol name for each of the objects.

Operating modes

At a Glance

You can use each axes control channel in 4 operating modes:

Operating mode	Description
Automatic (AUTO)	This mode supports the execution of movement commands driven by the SMOVE functions.
Manual (MANU)	This mode supports the visual driving of the moving part, from a front panel or an operator dialog desk. The commands are enabled via output bits %Q.
Direct (DIRDRIVE)	In this mode the output behaves like a digital / frequency converter. This mode controls the movement according to the movement setpoint indicated in the variable PARAM.
Stop (OFF)	In this mode the channel does not monitor the moving part; it only reports on current position and speed. This mode is forced at start up, if the axis is configured and error free.

Selecting the mode

The mode is selected by means of the word MOD_SELECT (%QWr.m.c.0)

The following table indicates the selected mode, according to the value of the word MOD_SELECT:

Value	Selected mode	Description
0	OFF	Switch to stopped movement
1	DIRDRIVE	Order to move in direct mode.
2	MANU	Order to move in manual mode.
3	AUTO	Order to move in automatic mode.

For any other value of MOD_SELECT, the OFF mode is selected.

Changing mode during a movement

Changing operating mode while a movement is in progress (DONE bit set to 1: %Ir.m.c.1) causes the moving part to stop. When the moving part is stopped (NO_MOTION bit set to 1: %Ir.m.c.7), the new operating mode is then activated.

NOTE: Only the commands concerning the current mode are examined. The other commands are ignored: for example, the channel in MANU mode (IN_MANU set to 1: %Ir.m.c.18), if the DIRDRV (%Qr.m.c.0) command is activated, it is ignored. It is necessary to change firstly to DIRDRIVE mode.

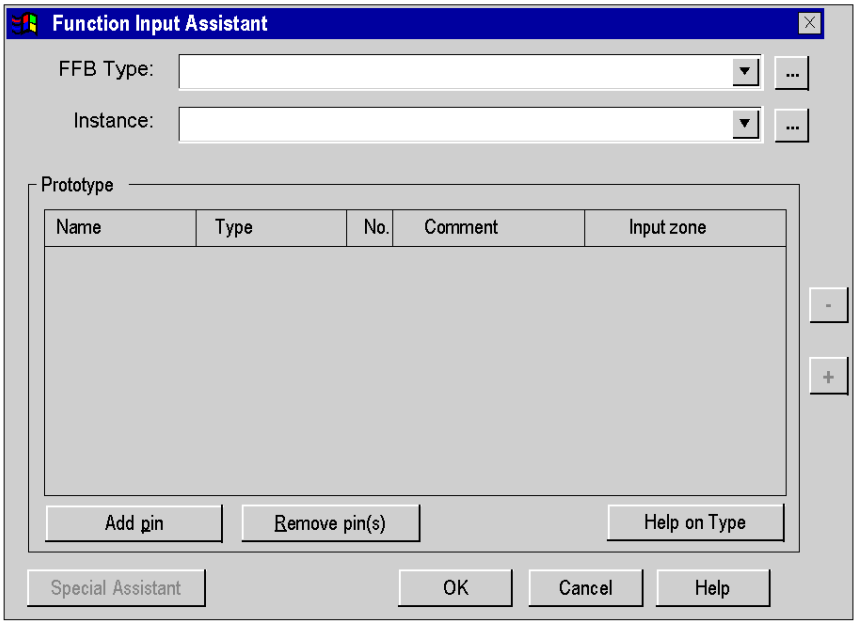
Programming the SMOVE function (in automatic mode)

At a Glance

You can program an SMOVE function in any programming module in ladder language (by means of an operation block), in instruction list language (between square brackets) or in structured text language. In all cases the syntax remains the same.

Assisted input screen

You can enter the SMOVE function or through the entry help screen use:



Assisted input

In the editor of the ST program for example, proceed in the following way:

Step	Action
1	Right-click on the place you wish to insert the SMOVE function and select the **FFB input Assistant...**.
2	Enter SMOVE in the field **Type FFB**. **Result**: The entry help window of the SMOVE function appears automatically and allows you to enter the parameters or to enter the details screen.
3	Press the **Special Assistant** button and complete the different fields offered *(see page 97)*. You can also enter the function variables directly into the parameter input area.
4	Confirm with **OK**. The function is then displayed.

Inputting SMOVE function parameters

Introduction

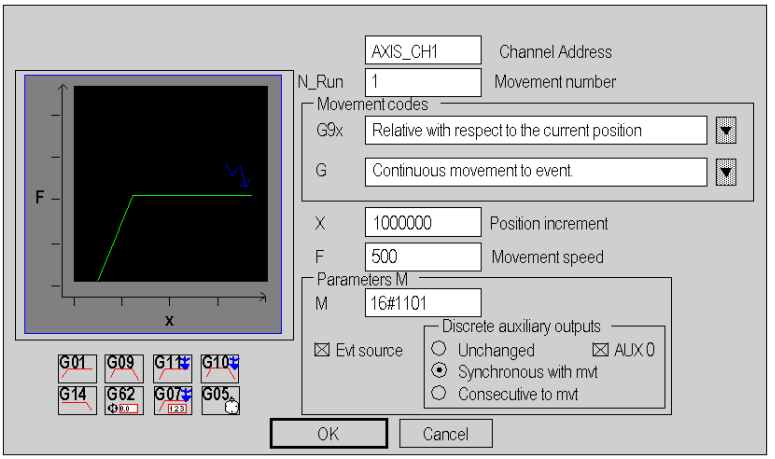
A movement command is programmed via a SMOVE function, with the following syntax:

```
SMOVE (Axis_ch1,N_Run,G9x,G,X,F,M)
```

The **Details** screen allows you to enter each of the parameters with help.

Details screen for the SMOVE function

The Details screen for the SMOVE function is as follows:



The entry fields (parameters for the SMOVE function) are as follows:

Parameter	Description
Axis_ch1	IODDT type variable corresponding to channel 1 on which the function must be carried out. Example: `AXIS_CH1` is of the type `T_STEPPER_STD`
N_Run	Movement number.
G9x	Movement type.
G	Instruction code.
X	Coordinate of the position to be reached.
F	Speed of movement of the moving part.
M	Event processing, auxiliary discrete outputs associated with the channel.

Description of the SMOVE function parameters

At a Glance

You must enter the following parameters in order to program a movement function:

```
SMOVE (Axis_ch1,N_Run,G9_,G,X,F,M)
```

IODDT

`AXIS_CH1` is a variable of type IODDT *(see EcoStruxure™ Control Expert, Operating Modes)* corresponding to a channel 1 of the axes control module on which the function is to be performed. `AXIS_CH1` is of the type `T_STEPPER_STD`.

Movement number

N_Run defines the movement number (between 0 and 32767). This number identifies the movement carried out by the SMOVE function.

In debug mode, this number tells you which movement is in progress.

Movement type.

G9_ defines the movement type:

Code	Movement type.
90	**Absolute** movement.
91	Movement **relative to the current position**.
98	Movement **relative to the stored PREF1 position**. The storing of the PREF1 position is achieved using the instruction code G07.

To select the movement type, use the browse button situated to the right of the **G9_** field or enter the code directly when inputting directly (without going to the **Details** screen).

Instruction code

G defines the instruction code *(see page 100)* for the SMOVE function

Coordinate of the position to be reached

X defines the coordinate of the position to be reached or towards which the moving part must move (in the case of a continuous movement). This position can be:

- immediate,
- coded in a double internal word %MDi or internal constant %KDi (this word can be indexed).

This value is expressed in the unit defined by the configuration parameter **Length Units** (e.g. micron).

NOTE: In the case of instructions G14, G21 and G62, this parameter represents the value of the reference point.

Speed of movement of the moving part

F defines the speed of movement of the moving part. This speed can be:

- immediate,
- coded in a double internal word %MDi or internal constant %KDi (this word can be indexed).

The unit of speed is Hertz.

NOTE: The speed can be modulated in the course of the movement by means of the CMV (Speed Modulation Coefficient). Actual F = programmed F x CMV/1000. This parameter, initialized by default at 1000, can be in between [02000], the resulting speed must always be more than SS_FREQ. The value 0 signifies the moving part has been stopped.

Parameter M

M defines a word which is coded in 4-bit bytes (in hexadecimal):

- the activation or non activation of the trigger for application event processing, for instructions G10, G11, G05 and G07:
  - M = 16#1000: activation of the associated event task,
  - M = 16#0000: non activation of the event task when the SMOVE command is executed.

For example:

Byte	3	2	1	0
16#				

NOTE: Coding is automatically performed in the **M** field in the **Details** screen, when you make selections using the check boxes and radio buttons available in this screen.

Instruction codes for the SMOVE function

Introduction

The **G** defines the instruction code.

To select the instruction code you can use the browse button situated to the right of the **G** field, click on the icon corresponding to the movement or enter the code directly when inputting directly (without going to the **Details screen**).

List of instruction codes

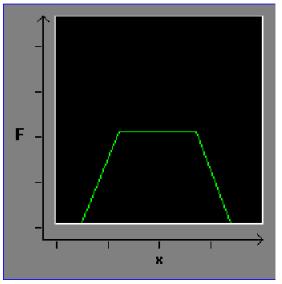
The instruction codes that you can select in the **Details** screen are as follows:

Instruction code	Meaning	Icon
09	Movement to position with stop.	
01	Continuous movement to position.	
10	Movement to event with stop.	
11	Continuous movement to event.	
14	Reference point.	
62	Forced reference point.	
05	Await event.	
07	Storage of position on event.	

Details screen gaphics

The **Details** screen also display a graphic which represents the selected movement.

For example, code G09:



Description of basic movements with the SMOVE function

At a Glance

Some instructions of the SMOVE function allow you to carry out basic movements.

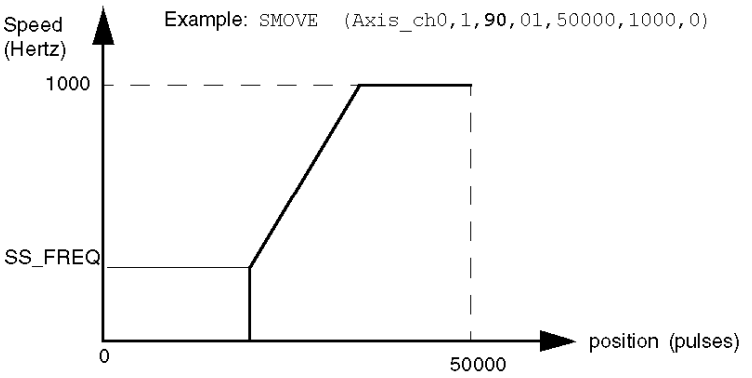
When programming these movements, the user defines the position to be reached and the speed. The acceleration parameter (constant, trapezoidal law of speed) is defined by this adjustable parameter.

The movements can be:

- absolute in relation to the machine origin **90**
- relative in relation to the current position **91**
- relative in relation to the stored position PREF **98**

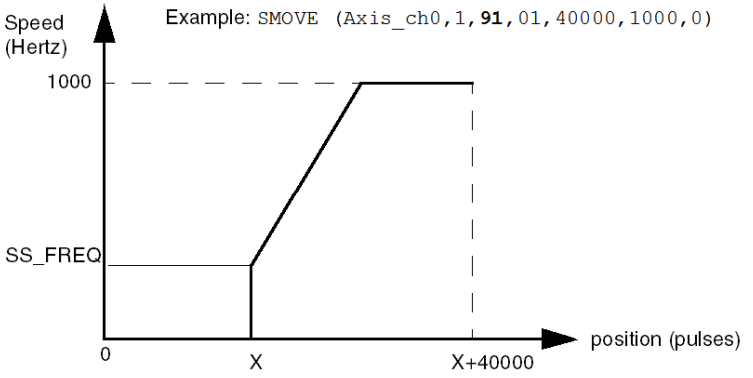
Absolute movement in relation to the machine origin

Example of an absolute movement in relation to the machine origin **90**.



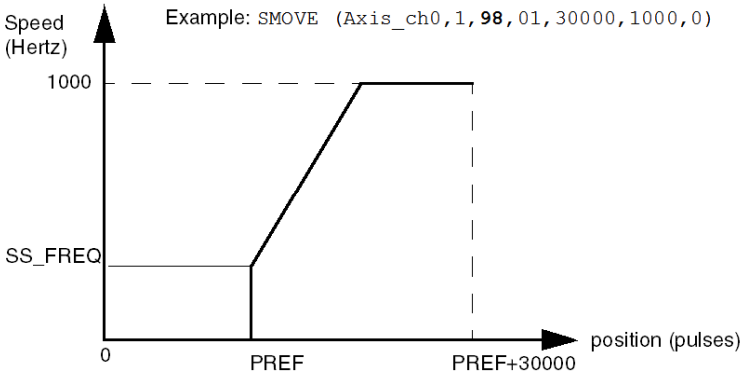
Relative movement in relation to the current position

Example of a relative movement in relation to the current position **91**.



Relative movement in relation to the stored position

Example of a relative movement in relation to the stored position PREF **98**.



Description of SMOVE instruction codes

At a Glance

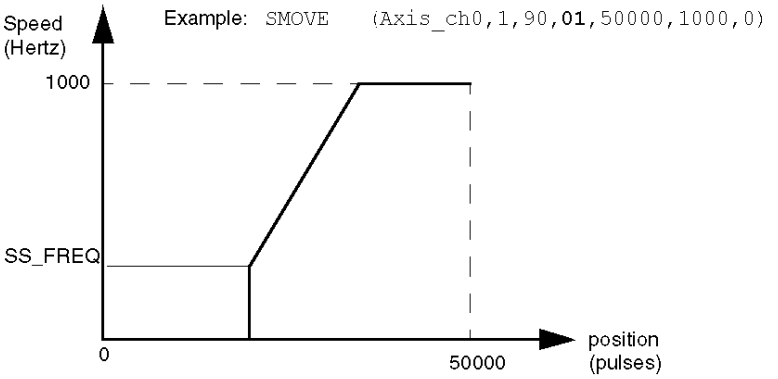
Three classes of movement can be programmed:

- movements to a position (instruction codes 01 and 09)
- movements until the detection of an event (instruction codes 11 and 10)
- reference point (instruction 14)

To find out about the execution conditions for instructions, see *Diagnostics and maintenance, page 195*.

Continuous movements to a position

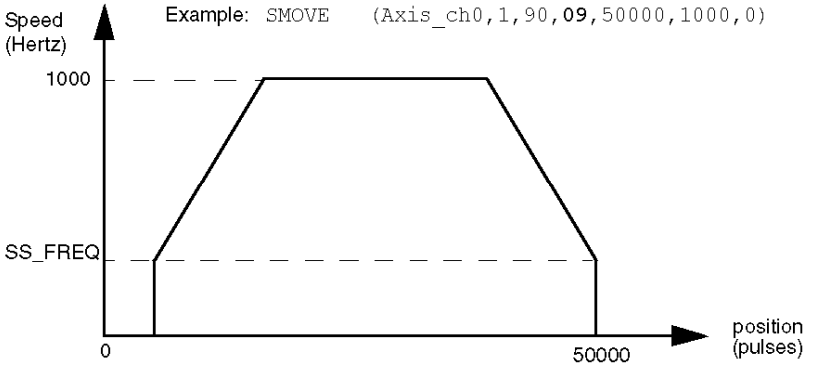
Example of a continuous movement to a position: instruction code **01**.



NOTE: If the instruction 01 is not followed by any movement instruction, the moving part continues the movement until it reaches the soft stops (after passing the position to be reached, the CMV (Speed Modulation Coefficient) is no longer interpreted).

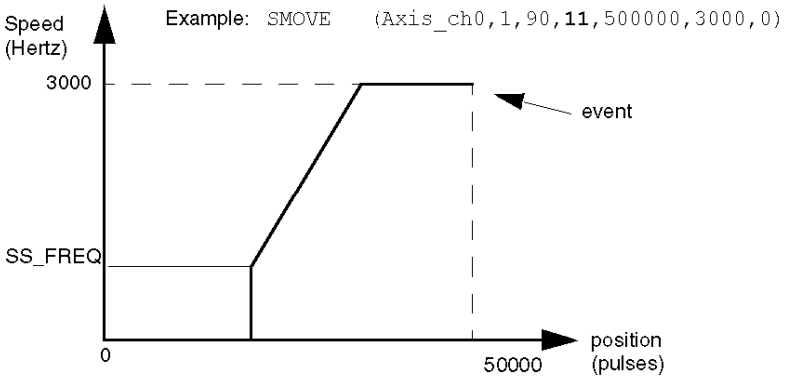
Movements to a position with stop

Example of a movement to a position with stop: instruction code **09**.



Continuous movement until an event

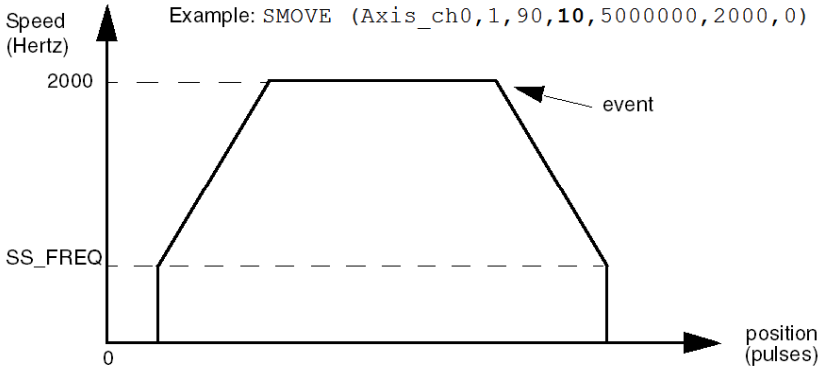
Example of a continuous movement until an event: instruction code **11**.



NOTE: The event can be a rising or falling edge on the dedicated event cam input, or a rising edge on the EXT_EVT bit (%Qr.m.c.11) by program.
It is essential that the position parameter be defined. If the event is not detected, the instruction is terminated when the requested target position is reached.
These instructions 11 and 12 can activate the event task when the event is detected if M is equal to 16#1000.

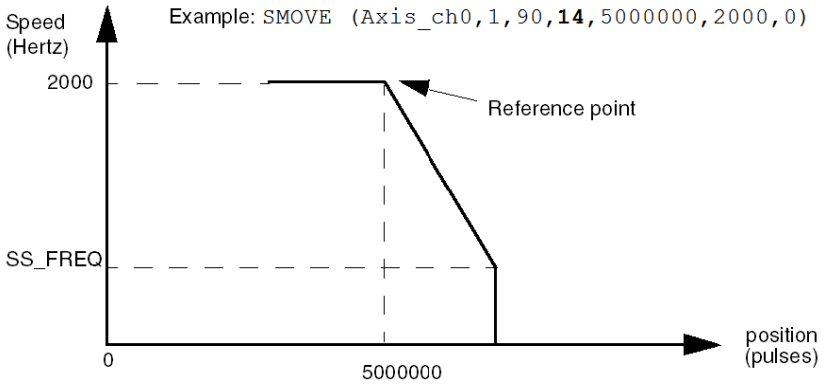
Movement until an event with stop

Example of a movement until an event with stop: instruction code **10**.



Reference point

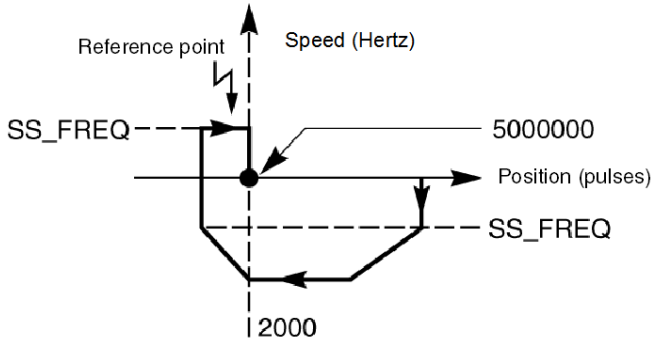
Example of setting a reference point: instruction code **14**. Reference point configured towards short cam in the + direction. At the start, the moving part moves away from the cam.



NOTE: This instruction triggers a reference point sequence according to the selection made in configuration. The value provided in the X parameter corresponds to the coordinate to be loaded with the current value when the reference point is detected.

Example of setting a reference point: instruction code **14**. Reference point configured towards long cam in the + direction. At the start, the moving part moves towards the cam.

Example: `SMOVE (Axis_ch0,1,90,14,5000000,2000,0)`



NOTE: This command is only accepted if the moving part is at rest: NO_MOTION bit = 1 (%Ir.m.c.7).

Forced reference point

This command performs a forced reference point (without movement of the part), the instruction code is **62**. The current value of the reference point is forced to the value entered in the position parameter **X**.

Example: `SMOVE (Axis_ch0,1, 90, 62, 100000, 100, 0).`

When this instruction is executed, the position of the moving part is forced to 100000.

NOTE: Regardless of the status of the axis, referenced or not, this command is accepted and it has the action of referencing the axis when the execution is completed. This command is only accepted if the moving part is at rest, NO_MOTION bit = 1 (%Ir.m.c.7).

Await event

This command, instruction code **05**, makes the channel wait for an event which can be:

- a change of state of the reflex input (rising or falling edge according to the selection made at configuration),
- a rising edge for the EVT_EXT bit (%Qr.m.c.11)

In the context of this instruction, the **F** parameter specifies the time envelope with a resolution of 10 ms. If the event is not triggered at the end of the time envelope, the command is deactivated. If **F** = 0, the wait carries on indefinitely.

Example: `SMOVE (Axis_ch0,1, 90, 05, 500, 100, 0).`

It is possible to associate event processing *(see page 117)*, to do that you must program M to 16#1000.

NOTE: When executing this instruction, the T_SPEED object (%MDr.m.c.10) does not contain the parameter F for the wait time. On the other hand, it is advised to systematically associate an event process to this command as the bit TO_G05 (%Ir.m.c.39), which allows the application to distinguish whether the command is terminated by detecting an event or by the time-out of the time envelope, is only refreshed if this process is activated.

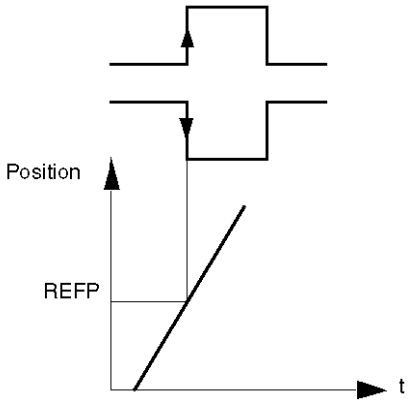
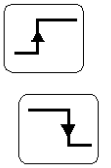
Storage of the current position on event.

After the execution of this instruction, code **07**, when the trigger input is entered the event defined in configuration is produced and the current position is stored in the PREF register.

NOTE: The parameter of position X must be equal to 1.

Example: `SMOVE (Axis_ch0,1, 90, 07, 1, 0, 0).`

Descriptive table for storing the current position on event.

Type of event on event input	Timing diagram	Configuration selection
Rising edge Falling edge Behavior		

NOTE: This instruction is not blocking, the program carries on straightaway to the next instruction. The stored value of the current position can be accessed in the PREF register (%IWr.m.c.7) only if activation of event processing is requested (M=16#10000).

NOTE: When this instruction is carried out, the object T_XPOS (%MDr.m.c.8) does not contain the parameter X=1.

Example of using an indexed position (repetitive movements)

At a Glance

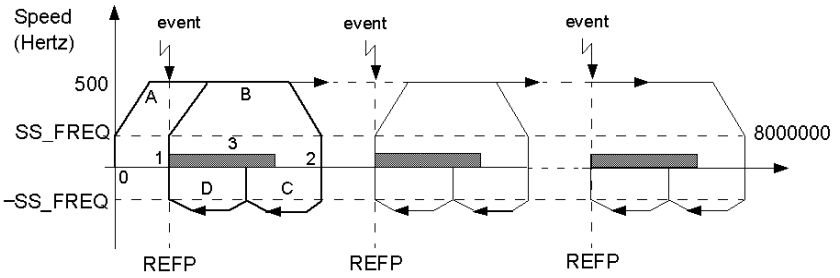
We want to carry out the sequence of the following basic movements 9 times:

- movement **A** until the detection of the edge of part 1,
- movement **B** as far as position 2 = +20000 in relation to the edge of part 1,
- movement **C** as far as position 3 = +10000 in relation to the edge of part 1,
- movement **D** as far as the edge of part 1.

In this example, the reference point is assumed to be taken and the moving part is at the reference point. We use an `AXIS_CH0` IODDT type variable associated with channel 0 of the axis command module on which the function is to be applied. `AXIS_CH0` is of the type `T_STEPPER_STD`.

Illustration

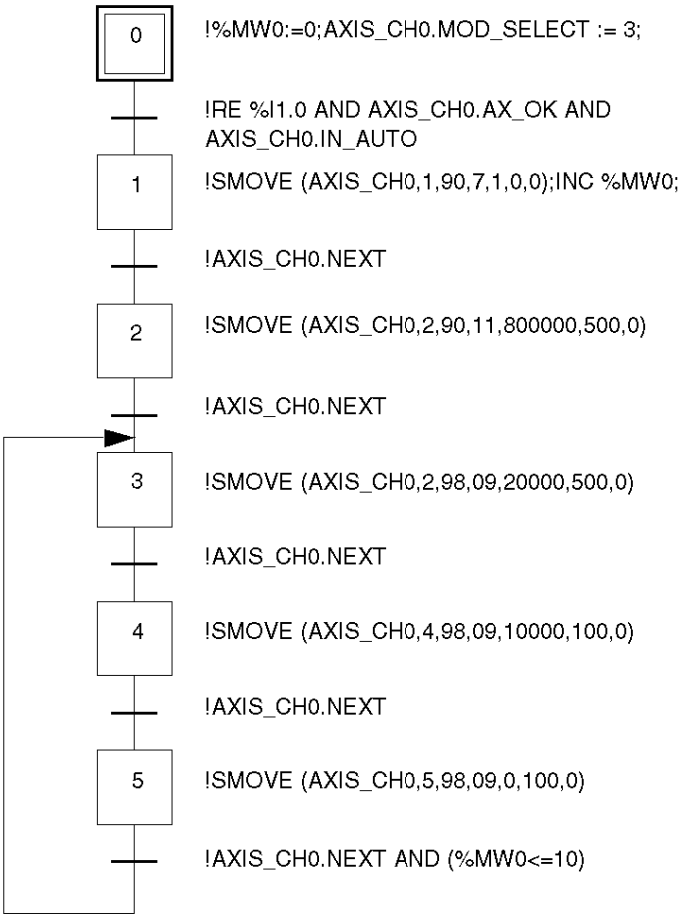
Position diagram.



NOTE: the sequence of basic movements is represented in bold on the curve. The numbers given correspond to the program step numbers included in the SMOVE function.

Description of the program

Grafcet for the operation of repetitive movements.



NOTE: all the actions must be programmed on activation.

Movement command sequencing

Creating a trajectory

The creation of a trajectory is carried out by programming a series of basic movement instructions (SMOVE function). This function applies to an IODDT type variable `T_STEPPER_STD`. In the example shown, we declare the`AXIS_0` variable of type `T_STEPPER_STD`.

Each basic command to execute an SMOVE function must only be done once. You must program the execution either:

- in Grafcet: in a step programmed for activation or deactivation,
- in structured text or Ladder language, in one bit rising edge.

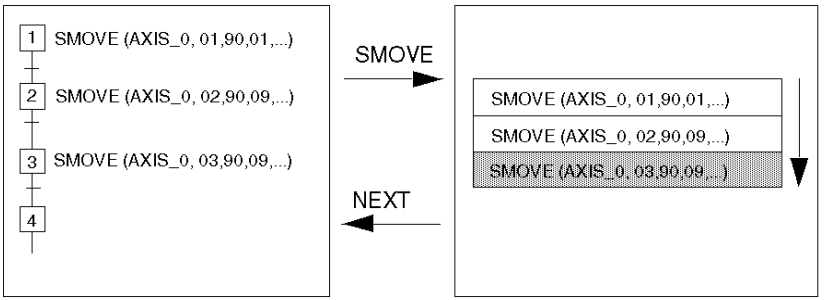
The report on the execution of the function is supplied by the module, via the bits NEXT and DONE.

Memory buffer

The TSX CFY module has a mechanism that supports the sequencing of movement commands.

Each axis of the TSX CFY module includes a memory buffer that allows it to receive 2 movement commands, in addition to that which it is executing. Thus, after finishing executing the current command, it carries on immediately to the first command present in the buffer memory.

Command sequencing:



Sequencing between 2 commands

The sequencing between 2 movement commands is carried out in the following way:

- instantaneously if the first movement is continuous,
- as soon as the moving part has stopped, if the first movement is with stop.

For sequencing to be instantaneous, the execution time of the instruction in progress must be greater than the period of the master task.

NOTE: A new command must only be transmitted to the module if the memory buffer associated with the axis to be controlled is not full.

Bits associated with the sequencing mechanism

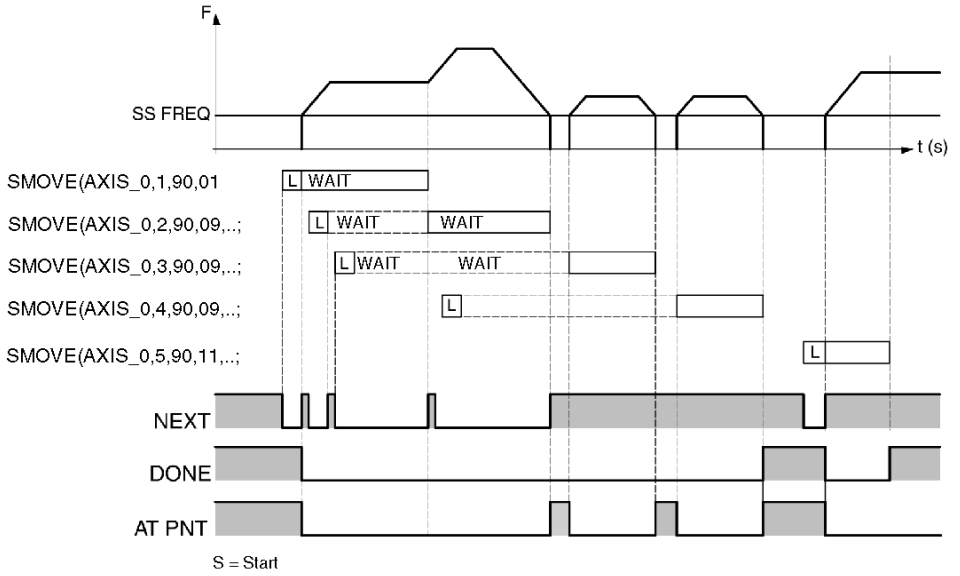
The bits associated with the sequencing mechanism are as follows:

Addressing	Description
NEXT (%Ir.m.c.0)	Indicates to the program user that the module is ready to receive the next movement command.
DONE (%Ir.m.c.1)	Indicates the end of the execution of the current command and the absence of a new command in the memory buffer.
AT_PNT (%Ir.m.c.8)	Indicates that the moving part has reached the point aimed for: • for a continuous movement, remains at 0, • for a movement with stop, is equivalent to NO_MOTION.

NOTE: The program must always test either the NEXT bit or the DONE bit before executing an SMOVE command.

Example

The following diagram represents the timing diagram of a sequence:



For a movement with stop: DONE switches to 1 when NO_MOTION (%Ir.m.c.7) switches to 1 and when the memory buffer is available.

For a continuous movement: DONE switches to 1 when the target position is exceeded and when the buffer memory is empty.

Deferred PAUSE function

At a Glance

The command PAUSE (%Qr.m.c.12) allows you to suspend the sequence of movements. It only becomes active when the moving part is stopped, that is, at the end of a G09 or G10 instruction.

The next movement starts as soon as the PAUSE command is reset to 0.

The ON_PAUSE bit (%Ir.m.c.26) indicates when set to 1 that the axis is in PAUSE status.

This function has 2 possible uses:

- the block by block execution of the movement program,
- synchronization of the axes by the same step by step axes control module.

Block by block execution of the movement program

If the instruction in progress is with stop, the activation of the **PAUSE** command in the debug screen in automatic mode or the setting of the PAUSE bit (%Qr.m.c.12) to 1, causes a switch to standby after the execution of the instruction in progress: this stops the movement sequencing.

It is thus possible by successively activating and deactivating the PAUSE command to execute the movements block by block with the aim of facilitating the debug.

Synchronization of several axes

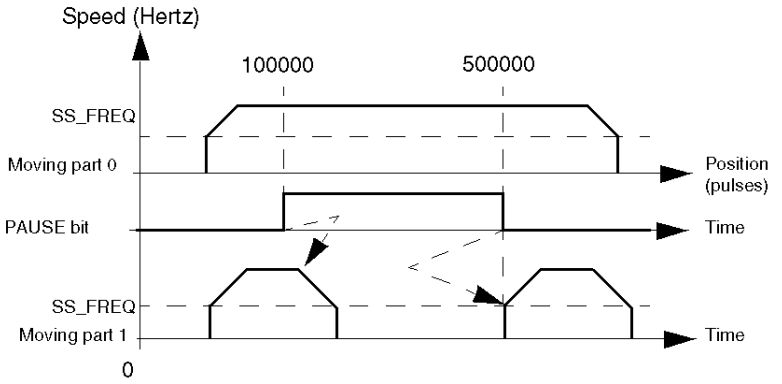
For each axis, the setting of the PAUSE bit (%Qr.m.c.12) to 1 by the program causes a switch to standby after the execution of the instruction in progress.

When the PAUSE bit is reset to 0, the module continues the execution of the instructions.

Example

The execution of the movement of the moving part 1 is stopped when moving part 0 reaches position 100000. The movement is activated again when the moving part 0 reaches dimension 500000. We use `AXIS_0` of type `T_STEPPER_STD` as an IODDT variable associated with the channel

```
IF (AXIS_0.POS >= 100000) THEN SET AXIS_0.PAUSE;
.............................
IF (AXIS_0.POS >= 500000) THEN RESET AXIS_0.PAUSE;
```



NOTE: The PAUSE command is only processed when AUTO mode is active.

Feed hold function

At a Glance

This function allows you, in automatic mode, to cause the moving part to stop, while assuring, at the time of the command to restart the movement, the continuation of the programmed trajectory (with no risk of commands being refused).

Activating the function

The Feed Hold function is activated by assigning the value 0 to the CMV (Speed Modulation Coefficient) word (%QWr.m.c.1).

It causes the moving part to stop in accordance with the programmed deceleration.

The status report on the pause is indicated by the IM_PAUSE bit (%Ir.m.c.27).

Deactivating the function

The Feed Hold function is deactivated by reassigning the initial value (> 0) to the CMV (Speed Modulation Coefficient) word.

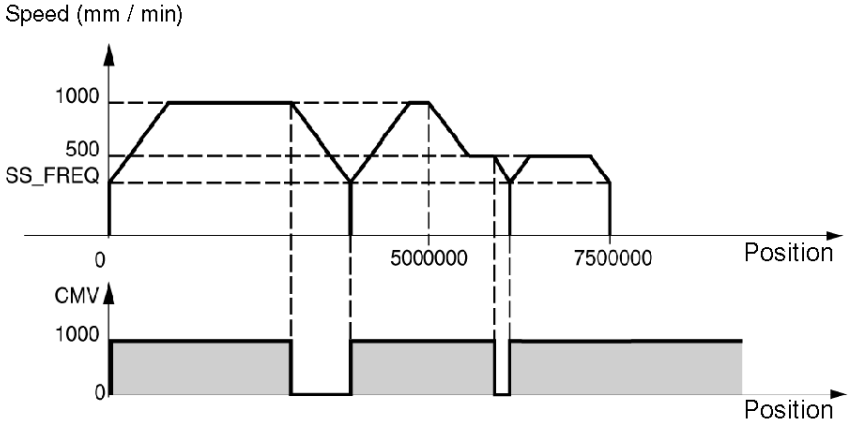
It causes the interrupted movement to restart at the speed corresponding to:

F x CMV / 1000.

Example

Activation / deactivation of the Feed Hold function applied to the channel 0 of a module located in slot 2 of a standard rack to which the variable `Axis_0` of type `T_STEPPER_STD` has been added:

```
SMOVE (Axis_0,1,90,10,5000000,1000,0);
SMOVE (Axis_0,2,90,09,7500000,500,0);
.............................
IF RE %M10 THEN %MW100 := Axis_0.SMC; Axis_0.SMC := 0;
IF RE %M10 THEN Axis_0.SMC := %MW100;
```



NOTE: Upon a STOP order or a blocking error, this command is deactivated.

NOTE: when the position aimed for is exceeded, when there is a stop following a Feed Hold command, the movement in progress is considered terminated. In this case the trajectory restarts with the movement which was on standby in the memory buffer.

Event processing

At a Glance

The channels of the TSX CFY modules can activate an event task. To do that you must have enabled the functionality in the configuration screen, by associating an event processing number to the channel *(see page 151)*.

Activating an event task

The following instructions trigger the sending of an event that activates the event task:

- Movement until an event, codes **10** and **11**: the event processing application is activated, when the event is detected.
- Await event, code **05**: the event processing application is activated, at the end of the instruction.
- Storing the current position when the event appears, code **07**: the event processing application is activated at the end of storing the PREF position.

The event processing application is activated, if bit 12 in the M parameter of the SMOVE function associated to the instruction is set to 1 (M equals 16#1000).

Variables which can be used by the event task

- If several event sources are selected, the following bits allow you to determine what caused the event processing application to be triggered:
  - EVT_G1X (%Ir.m.c.40): end of G10 or G11 on event,
  - EVT_G05 (%Ir.m.c.38): end of G05 on event,
  - TO_G05 (%Ir.m.c.39): G05 timing elapsed,
  - EVT_G07 (%Ir.m.c.37): storage of position,
- The OVR_EVT bit (%Ir.m.c.36) allows you to detect a delay in the sending of the event or a loss of event.
- Value of the stored position PREF (%IWr.m.c.7).

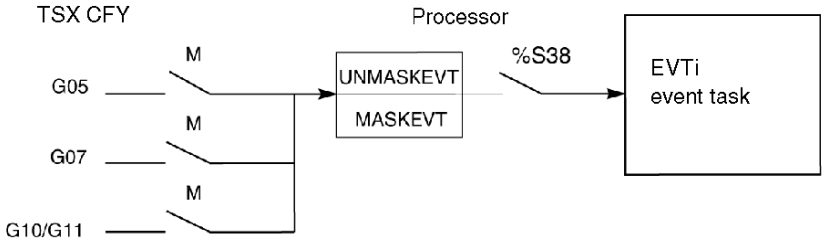
NOTE: The bits and the words described above are the only values refreshed in the event task, and are only updated into the PLC when the task is activated.

Masking events

The programming language offers 2 methods of masking events:

- Instruction for the global masking of events: MASKEVT() (the instruction UNMASKEVT() is used for unmasking).
- ACTIVEVT bit = 0 (%S38) global disabling of events. The ACTIVEVT bit is normally set to 1.

Summary diagram:



Managing operating modes

Powering up the module

When powering up the module or when plugging it in, the TSX CFY module carries out auto-tests with the outputs in safety position (outputs at 0).

At the end of the auto-tests:

If the auto-tests	Then the module
have not detected any errors.	tests the configuration with the outputs in safety position. If the configuration is correct, the module switches to disabled mode (OFF).
have detected an error or the configuration is incorrect.	signals an error and retains the outputs in safety position.

PLC in RUN mode

All the operating modes of the configured channels are usable.

Switching the PLC from RUN to STOP mode

When switching the PLC from RUN mode to STOP mode or when there is a loss of communication between the processor and the module, the moving part decelerates and stops and the module switches to STOP mode (OFF).

NOTE: The IRSTSCANRUN bit (%S13) allows you to detect when the PLC switches to STOP mode. It is set to 1 during the first cycle after the PLC switches to RUN mode.

Changing configuration (reconfiguration)

- The moving part decelerates and stops.
- The channel becomes unconfigured.
- The channel tests the new configuration with the outputs in safety position.
- If the new configuration is correct, the channel switches to STOP mode (OFF).
- If the configuration is incorrect, the module signals an error and retains the outputs in safety position.

Power outage and return

When there is a power outage, the moving part stops.

At the time of a cold start or a warm restart, the configuration of the channels is automatically transmitted by the processor to the module The latter switches to STOP mode (OFF).

Managing faults

At a Glance

The monitoring of faults is essential in the area of position control due to the inherent risks with moving parts.

The checks are carried out internally and automatically by the module.

Types of fault

The module detects 4 types of faults:

- **Module faults**. These are the hardware faults internal to the module. All the axes driven by the module are thus affected by the appearance of this type of fault. They can be detected during auto-tests (when reinitializing the module) or during normal operation (I/O fault).
- **Hardware channel faults external to the module** (for example, brake output short-circuit).
- **Application channel fault** linked to the axes (for example, exceeding soft stop).
  The monitoring of faults at axis level is permanently active when the axis is configured.
- **Commands refused channel faults**. These are the faults which can appear when executing a movement, configuration transfer, adjustment parameters transfer or change of operating modes command.

NOTE: The monitoring of some faults at axis level can be enabled or disabled by the axes control parameters. These control parameters can be adjusted in the adjustment screen.

In STOP mode (OFF), the monitoring of application faults is disabled

Levels of gravity

Faults are classified into 2 levels of gravity:

- **Critical or blocking faults** which cause the moving part (in the case of an axis fault) or the moving parts managed by the module (in the case of a module fault) to stop. They cause the following processes:
  - signaling of the fault,
  - deceleration of the moving part until it stops,
  - disabling of the translator, activation of the brake,
  - clearing of all the stored commands,
  - the wait for an acknowledgement.

  The fault must have disappeared and been acknowledged for you to be able to restart the application.
- **Non critical faults** which cause a signaling of the fault without stopping the moving part. You must program the action to be taken with this type of fault in Control Expert.
  The fault message disappears when the fault has gone and has been acknowledged (the acknowledgement is not stored and is only effective if the fault has gone).

NOTE: in the case of opening the emergency stop input, or of disabling the translator ENABLE = 0(%Qr.m.c.10), the deceleration phase is not carried out and the stop is immediate. However, the appearance of step failure information is not considered to be a blocking fault, and is simply signaled to the application.

Programming faults

Faults can be displayed, fixed and acknowledged from the debug screen, but it can be useful in operation to be able to drive the moving part and fix faults from a console. For this end, the application has all the necessary information and commands.

Signaling faults

The module offers a lot of information in the form of status bits and words, which can be accessed via the Control Expert program. These bits allow faults to be handled hierarchically:

- to act on the principle program,
- to simply signal the fault.

Levels of signaling

2 levels of signaling are provided:

First level: general information

Bit	Fault
MOD_ERROR (%Ir.m.c.ERR)	Channel fault
AX_OK (%Ir.m.c.3)	No blocking fault (with stop of moving part) is detected
AX_FLT (%Ir.m.c.2)	Fault (groups all the faults together)
HD_ERR (%Ir.m.c.4)	External hardware fault
AX_ERR (%Ir.m.c.5)	Application fault
CMD_NOK (%Ir.m.c.6)	Command refused

Second level: detailed information

Module and axis fault status words CH_FLT(%MWr.m.c.2) and AX_STS(%MWr.m.c.3). These words are obtained via explicit exchange requests described in language objects *(see page 209)*.

NOTE: On encountering a blocking fault, we advise you to stop the development of the sequential processing to which the axis is associated and to fix the fault by driving the moving part in manual mode. The correction of the fault must be followed by an acknowledgement of the fault.

Fault acknowledgment

When a fault appears:

- The fault bits AX_FLT, HD_ERR, AX_ERR and the extract bits of the status words concerned by the fault are set to 1.
- If the fault is blocking the AX_OK bit is set to 0.

When the fault disappears, all the fault bits remain in their status. The fault is stored until the acknowledgement is obtained, by setting the ACK_FLT bit (%Qr.m.c.9) to 1 (or re-initializing the module). The acknowledgement must be carried out after the fault has gone (except for soft stop faults)

If several faults are detected, the acknowledgement order only works on the faults that have effectively disappeared. The faults still present must be acknowledged again after they have disappeared.

NOTE: The acknowledgement of a fault can also be carried out on initialization of the PLC, or when a correct new command is accepted in the case of a command refused fault.

Summary table for the different types of faults

The following table summarizes the different types of fault and the associated bits:

<table>
<tr><td rowspan="2">Channel fault (MOD_ERROR bit: %Ir.m.c.ERR)</td><td colspan="2">Process faults (AX_FLT bit: %Ir.m.c.2)</td><td rowspan="2">Command refused (CMD_NOK bit: %Ir.m.c.6)</td></tr>
<tr><td colspan="2">AX_OK: %Ir.m.c.3 (No blocking fault detected)</td></tr>
<tr><td>• Internal
• Communication
• Configuration
• Configuration or adjustment</td><td>• Emergency stop
• Translator
• 24 Volt supply
• Brake output short circuit</td><td>• Soft stops</td><td>Coding of the fault in the word CMD_FLT: %MWr.m.c.7</td></tr>
</table>

(*) These faults are non blocking faults and have no influence on the AX_OK bit.

Description of the channel faults

The MOD_ERROR bit groups all the faults at channel level:

- Internal fault MOD_FLT (%MWr.m.c.2.4): module absent, out of service or in auto-test.
- Communication fault COM_FLT (%MWr.m.c.2.6): communication fault with the processor.
- Configuration fault COM_FLT (%MWr.m.c.2.6): difference between the declaration of the position of the module in the configuration and its actual position.

NOTE: The words %MW require a READ_STS command in order to be updated.

Description of external hardware faults

At a Glance

These faults are signaled by the bit **HD_ERR** (%Ir.m.c.4). These faults are blocking faults and cannot be deactivated.

Emergency stop

The following table shows the cause, the signal and the remedy to be applied in the case of a **Emergency stop** fault:

Cause	Circuit open between the 24V and the **Emergency stop** input on the front panel of the module
Parameters	None
Consequence	The moving part is forced to stop
Signal	EMG_STOP (%Ir.m.c.29) and EMG_STP (%MWr.m.c.3.5) bits
Remedy	Re-establish the connection between the input and the 24V then acknowledge the fault.

24V supply

The following table shows the cause, the signal and the remedy to be applied in the case of a **24V supply** fault:

Cause	24V supply fault
Parameters	None
Consequence	The axis is not referenced, the moving part is forced to stop
Signal	AUX_SUP bit (%MWr.m.c.3.6)
Remedy	Re-establish the connection then acknowledge the fault

Brake output short circuit

The following table shows the cause, signal and the remedy to be applied in the case of a **Brake output short circuit** fault:

Cause	Short circuit detected on the module’s brake output
Parameters	None
Consequence	The axis is not referenced, the moving part is forced to stop
Signal	BRAKE_FLT bit (%MWr.m.c.3.1)
Remedy	Remove the short circuit then acknowledge the fault

Translator

The following table shows the cause, the signal and the remedy to be applied in the case of a **Translator** fault:

Cause	The translator check input does not receive the level **translator OK** defined in the channel configuration
Parameters	None
Consequence	The axis is not referenced, the moving part is forced to stop
Signal	DRV_FLT bit (%MWr.m.c.3.2)
Remedy	Remove the translator fault then acknowledge the fault

Description of application faults

At a Glance

These faults are signaled by the AX_ERR bit (%Ir.m.c.5). The parameters can be accessed via the Configuration editor adjustment screen.

Soft stops

The following table shows the cause, the signal and the remedy to be applied in the case of a **Soft stop** fault: This fault is blocking and cannot be deactivated.

Cause	The moving part is no longer situated between the 2 thresholds: software Lo limit and software Hi Limit (this monitoring is activated as soon as the axis is referenced)
Parameters	Software Hi Limit: SL_MAX (%MDr.m.c.14) Software Lo Limit: SL_MIN (%MDr.m.c.16)
Consequence	The moving part is forced to stop
Signal	SLMAX bit (%MWr.m.c.3.3): Software Hi Limit passed SLMIN bit (%MWr.m.c.3.4): Software Lo Limit exceeded
Remedy	Acknowledge the fault and, in manual mode, remove the moving part exceeding the soft stops to within the valid measurement zone. To do that, you must check: ● that no movement is in progress, ● that manual mode is selected, ● that the STOP command is set to 0, ● that the axis on which the command is carried out is referenced, ● that there is no other fault with stop on the axis. The moving part can either be brought back manually or by means of the commands JOG+ and JOG-.

Description of the command refused faults

At a Glance

A command refused fault is generated each time that a command cannot be executed. This command is not compatible with the status of the axis or the current mode or at least one of the parameters is not valid.

These faults are signaled by the indicator **Command Refused** in the debug screens. The **DIAG** key at the channel level tells you the source of the command refused. This information can also be accessed by program via the CMD_NOK bit (%Ir.m.c.6) and the CMD_FLT word (%MWr.m.c.7) *(see page 226)*.

Command refused

The following table shows the cause, the signal and the remedy to be applied in the case of a **Command refused** fault.

Cause	Unauthorized movement command Transfer of incorrect configuration or parameters
Parameters	None
Consequence	Immediate stop of the movement in progress Reset to 0 of the memory buffer receiving the movement commands in automatic mode
Signal	CMD_NOK bit (%Ir.m.c.6): movement command refused CMD_FLT word (%MWr.m.c.7): type of fault detected ● Least significant byte: executable commands, ● Most significant byte: adjustment configuration and parameters.
Remedy	Acknowledgement is implicit upon receipt of a new accepted command Acknowledgement is equally possible via the command ACK_FLT (%Qr.m.c.9)

NOTE: In the case of movement sequencing in automatic mode, we advise you to condition the execution of each movement with the end of the execution of the preceding movement and with the AX_FLT bit (%Ir.m.c.2). This means that the following command is not sequenced, when the current command has experienced a command refusal.

Management of manual mode

At a Glance

You can select and control the manual mode from the debug screen, but also via an application program, from a panel or an operator or supervisor dialog desk.

In this case, the dialog is programmed in Ladder, Instruction List or Structured Text language, using basic commands (movements, reference points, etc.)

Selecting manual mode

This is done by assigning the value 2 to the word MODE_SEL (%QWr.m.c.0).

The switch from the mode in use to manual mode forces the moving part to stop if a movement is in progress. Manual mode is effective as soon as the moving part has stopped.

When the command to switch into manual mode has been recognized, the IN_MANU bit (%Ir.m.c.22) bit is set to 1.

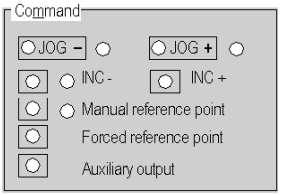
Executing manual commands

The basic commands associated with manual mode and accessible via command bits %Qr.m.c.j are as follows:

- Visual movement in the + direction JOG_P (%Qr.m.c.1).
- Visual movement in the - direction JOG_M (%Qr.m.c.2).
- Incremental movement in the + direction INC_P (%Qr.m.c.3).
- Incremental movement in – direction INC_M (%Qr.m.c.4).
- Manual reference point SET_RP (%Qr.m.c.5).
- Forced reference point RP_HERE (%Qr.m.c.6).

These commands are equivalent to those accessible from the debug screen of the TSX CFY module.

Manual commands:



General conditions for executing commands in manual mode

The following conditions must be fulfilled in order to execute commands in manual mode:

- Target position within in the software limits.
- Axis without blocking fault (AX_OK bit = 1: %Ir.m.c.3).
- No command being executed (DONE bit = 1: %Ir.m.c.1).
- STOP command (%Qr.m.c.8) inactive and confirmation bit of the translator relay ENABLE (%Qxy.i.10) set to 1.

NOTE: Except, in the case of a software limit fault, for commands JOG_P and JOG_M and after fault acknowledgement.

Stopping a movement

A movement may be stopped by:

- The appearance of the STOP command (%Qr.m.c.8) or the setting to 0 of the ENABLE bit (%Qr.m.c.10) or the STOP input.
- The appearance of a blocking fault.
- Changing the operating mode.
- Receiving a configuration.
- The switch to plus (or minus) end of run limit during a movement in the plus (or minus) direction.

Visual movement commands

At a Glance

To carry out a visual movement, you must use the manual commands JOG_P and JOG_M.

The bits JOG_P (%Qr.m.c.1) and JOG_M (%Qr.m.c.2) control the movement of the moving part in the positive or negative direction. The operator must visually follow the position of the moving part. The movement continues for as long as the command is present and it is not disabled by a STOP command or a fault.

The commands JOG_P and JOG_M are taken into account on edge and are kept active on state, whether the axis is referenced or not.

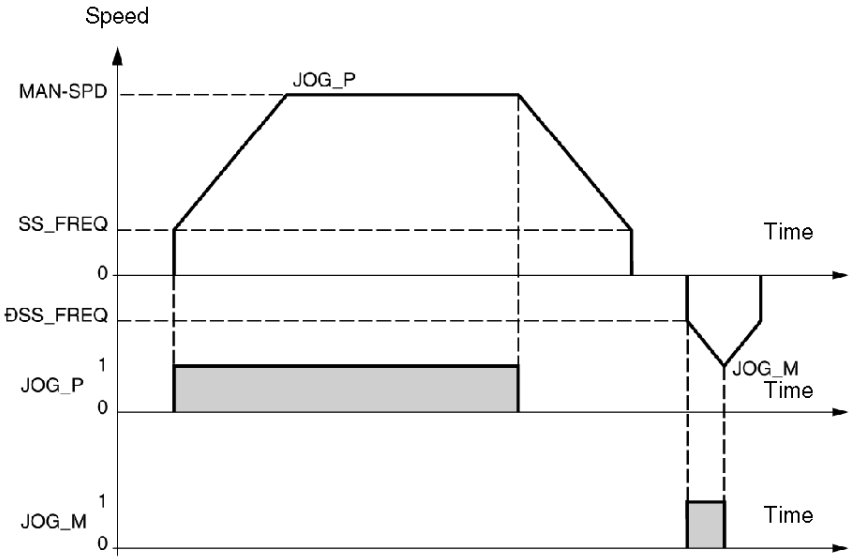
Movement speed

The movement is carried out at the manual mode speed MAN_SPD, defined in the adjustment screen (or in the double word MAN_SPD (%MDr.m.c.20)).

The speed can be modulated during movement by means of the CMV (Speed Modulation Coefficient) (%QWr.m.c.1).

Any movement speed higher than FMAX (maximum speed of the axis defined at configuration) is limited to the FMAX value.

Movement speed of the moving part:



Notes on the JOG_P and JOG_M commands

- The commands JOG_P and JOG_M are used to release the moving part when a software limit fault is detected. This is done after first acknowledging the fault.
- If the JOG_P or JOG_M bit is set to 1 when switching to manual mode, this command is not recognized. It will only be recognized when the bit has been cleared and then reset to 1 (detection of rising edge).

Incremental movement commands

At a Glance

To carry out an incremental movement, you must use the manual commands INC_P and INC_M.

The bits INC_P (%Qr.m.c.3) and INC_M (%Qr.m.c.4) control the movement to increment the position of the moving part in the positive or negative direction.

The value of the position increment PARAM is entered in the double word PARAM (%QDr.m.c.2) or in the debug screen of the TSX CFY module.

Further to the general conditions of execution in manual mode, the commands INC_P and INC_M are active on rising edge when:

- The axis is referenced.
- The target position is between the software limits.

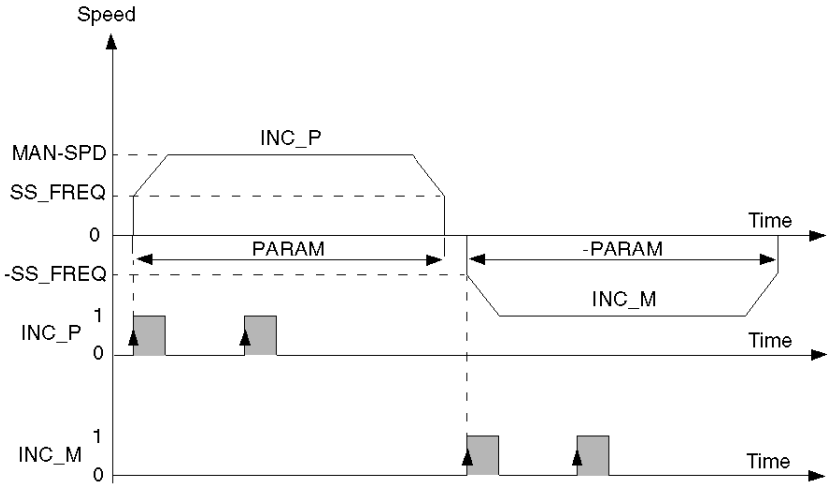
Movement speed

The movement is carried out at the manual mode speed, defined in the adjustment screen or in the double word MAN_SPD (%MDr.m.c.20)).

The speed can be modulated during movement by means of the SMC coefficient (%QWr.m.c.1).

Any movement speed higher than FMAX (maximum speed of the axis defined at configuration) is limited to the FMAX value.

Movement speed of the moving part:



Reference point command

At a Glance

You can take a reference point by using the command SET_RP.

The bit SET_RP (%Qr.m.c.5) takes a manual reference point with movement.

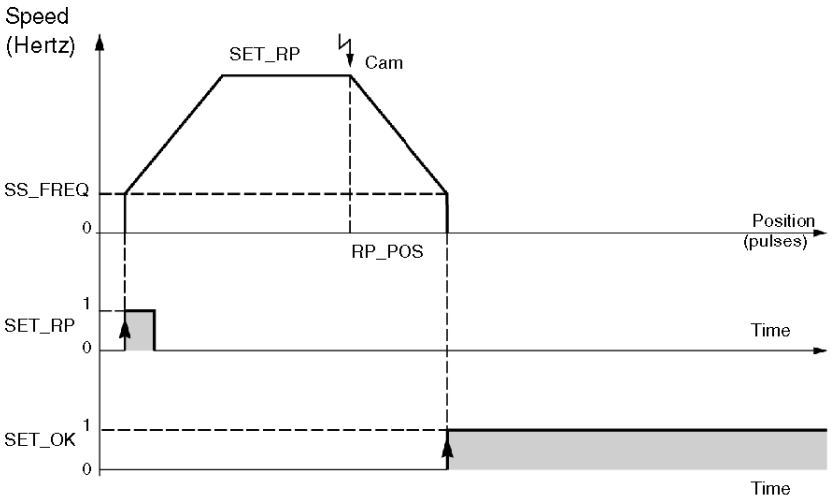
The type and the direction of the reference point are defined at configuration in the Reference point *(see page 152)* parameter. The value of the source is defined in the adjustment screen by the parameter RP Value or the double word RP_POS: (%MDr.m.c.22).

Approach speed

The approach speed is the manual speed defined in the adjustment screen or in the double word MAN_SPD (%MDr.m.c.20) multiplied by the CMV (Speed Modulation Coefficient). The reference point speed varies according to the type of reference point selected.

Any movement speed higher than FMAX (maximum speed of the axis defined at configuration) is limited to the FMAX value.

Example: short cam only and + direction



Forced reference point command

At a Glance

You can take a forced reference point by using the command RP_HERE.

The bit RP_HERE (%Qr.m.c.6) takes a forced reference point without moving, at the value defined in the PARAM parameter. This value is entered in the double word PARAM (%QDr.m.c.2) or in the debug screen of the TSX CFY 11/21 module.

The forced reference point command is used to reference the axis without carrying out a movement.

NOTE: The command RP_HERE does not modify the value of the RP_POS parameter.
The value of the PARAM parameter must be between the software limits.
No blocking fault is tolerated during execution of this command.

Management of direct mode (DIRDRIVE)

At a Glance

DIRDRIVE mode is used to simulate the axes control without operating the functional part; all the returned information is calculated.

The behavior of the axis can therefore be analyzed independently from the functional part.

Selecting direct mode

Direct mode is selected by assigning value 1 to the word MODE_SEL (%QWr.m.c.0.).

On a request to change mode, the moving part is stopped, then the mode is changed. When the command for switching into direct mode has been recognized, the bit IN_DIRDR (%Ir.m.c.17) is set to 1.

Execution of commands in direct mode

Direct mode includes the movement command DIRDRV (%Qr.m.c.0).

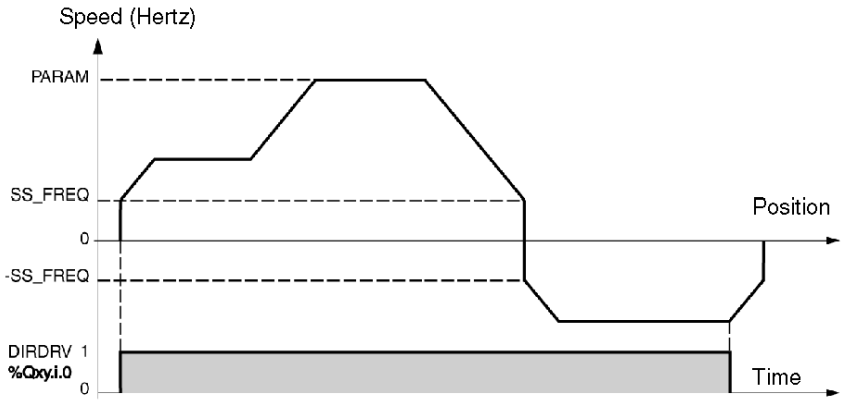
The speed setpoint is transmitted periodically by the variable PARAM (%QDr.m.c.2). The sign of this variable gives the movement direction.

The speed of the translator is controlled between SS_FREQ and FMAX. These values are defined in the configuration screen (FMAX) and in the adjustment screen (SS_FREQ).

The bit ST_DIRDR (%Ir.m.c.20) indicates that a movement is in progress in DIRDRIVE mode.

Speed law

When the setpoint is changed, the output achieves the new setpoint according to a trapezoid speed law, respecting the parameterized acceleration .



Execution of the DIRDRIVE command

The general conditions for execution of the DIRDRIVE command are as follows:

- Axis without blocking fault bit AX_OK = : (%Ir.m.c.3).
- STOP command (%Qr.m.c.8) inactive and confirmation bit of the translator relay ENABLE (%Qxy.i.10) set to 1.
- Parameter PARAM (%QDr.m.c.2) between - FMAX and -SS_FREQ or between SS_FREQ and FMAX of the selected axis.

Stopping a movement

A movement may be stopped by:

- The appearance of the STOP command or the setting to 0 of the confirmation bit of the translator relay ENABLE (%Qr.m.c.10).
- The appearance of a blocking error or a software limit error.
- Changing the operating mode.
- Receiving a configuration.
- The switch to the end of run plus limit (or minus) during a movement in the plus (or minus) direction.

NOTE: Software limit monitoring remains active is the axis is referenced. To disable this monitoring, cause the axis reference to be lost by temporarily diasabling ENABLE (%Qr.m.c.10) to 0 and confirming by setting ENABLE to 1 or by pressing the confirmation button.

Management of stop mode (OFF)

At a Glance

This mode is mainly used in debug from within the configuration editor. It can however be program-driven. In this mode the module remains passive, but continues to update the current position information POS (%IDr.m.c.0) and the current speed information SPEED (%IDr.m.c.2).

Selecting stop mode

Stop mode is selected by assigning value 0 to the word MODE_SEL (%QWr.m.c.0.).

Stop mode is also selected by the module when the PLC is in STOP mode. It is selected by default following channel configuration.

Execution of commands in stop mode

The OFF mode has no associated movement command.

The movement of the moving part is not monitored and the monitoring of software errors is disabled (with the exception of the software limit monitoring).

The translator enable output continues to be monitored by the command ENABLE (%Qr.m.c.10).

Chapter 8
Configuring step by step axes control

Subject of this chapter

This chapter describes the configuration screens in the TSX CFY modules.

What Is in This Chapter?

This chapter contains the following topics:

Description of the configuration screen of an axis control module

General

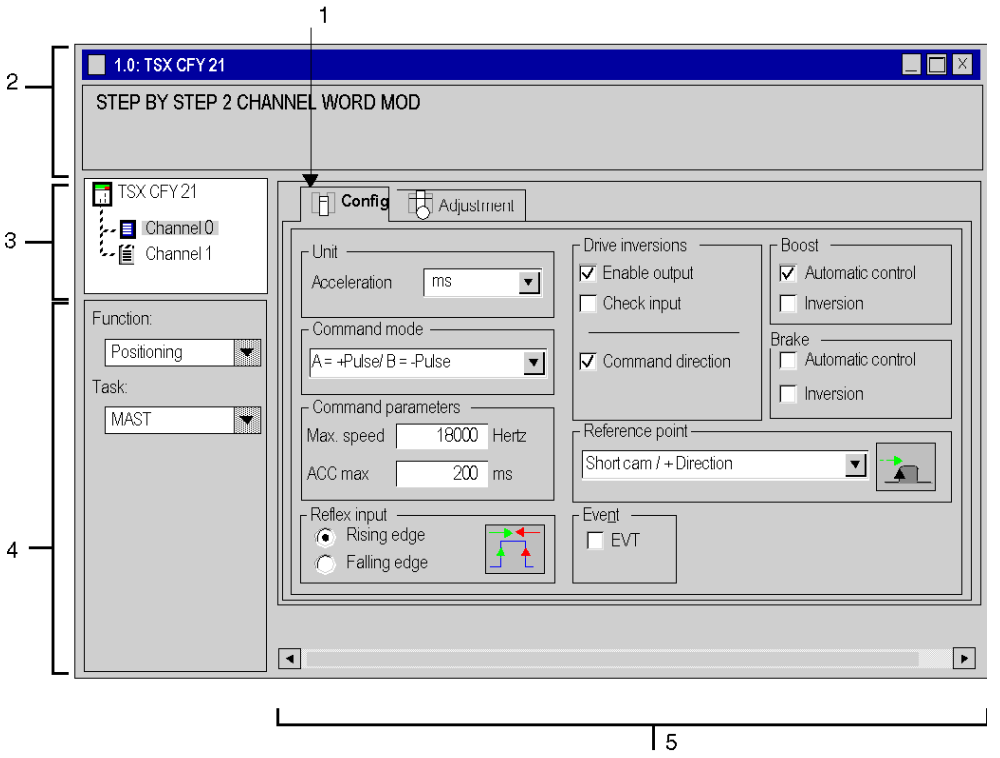
The configuration screen is a graphics tool designed to configure *(see EcoStruxure™ Control Expert, Operating Modes)* a selected module in a rack. It displays the parameters associated to channels of this module, and makes it possible to modify them in local and connected mode.

It also gives access to debug and adjustment screens (the latter only in connected mode).

NOTE: it is not possible to configure a module by program directly using %KW language objects, these words are accessible in read only.

Illustration

The figure below represents a configuration screen.



Description

The following table shows the different configuration screen elements and their functions.

Address	Element	Function
1	Tabs	The tab in the foreground shows the current mode (**Configuration** for this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Configuration,** ● **Adjustment,** **Note** : in online mode, other tabs appear that allow you to command the module and debug the program.
2	**Module** zone	Summary of the abbreviated heading of the module.
3	**Channel** zone	Is used: ● By clicking on the reference number, to display the tabs: ❍ **Description** which gives the characteristics of the device. ❍ **I/O Objects** *(see EcoStruxure™ Control Expert, Operating Modes)* which is used to presymbolize the input/output objects. ❍ **Fault** which shows the device faults (in online mode). ● To select the channel, ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** zone	Makes it possible to choose the axis control function and the task associated to the channel : ● **Function : Positioning**. By default, **No** function is configured. ● **Task:** defines the task (**MAST** or **FAST**) in which the implicit exchange objects or the channel will be exchanged.
5	**Configuration**zone	Makes it possible to configure the channel parameters. This zone includes different headings, displayed according to the function selected. Some choices may be disabled and appear grayed out. For each parameter, the limits are displayed in the status bar.

Access to the parameters configuration screen

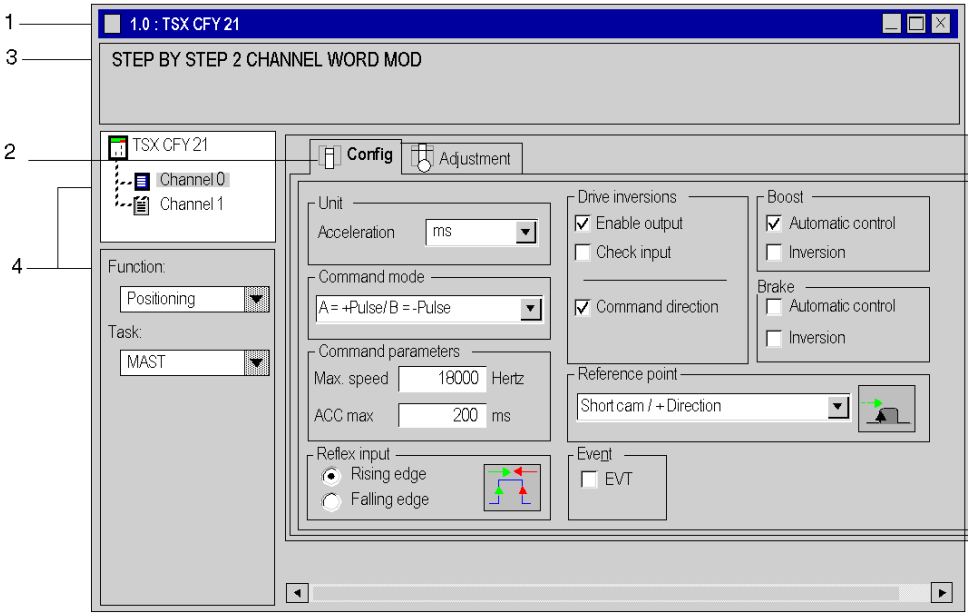
Accessing module parameterization

To access module parameterization, double click on its graphical representation in the rack or:

- select the module (by clicking on it),
- activate the command **Open module** from the **Edit** menu.

Parametering screen

The following screen allows you to parameterize the module:

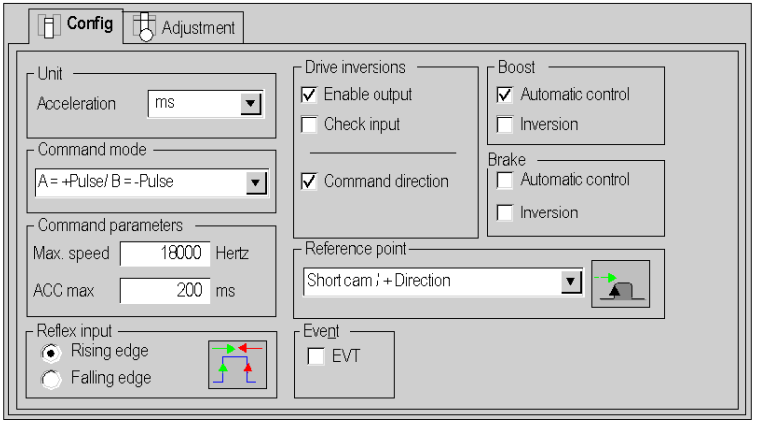


This screen is made up of 4 areas for information or the selection of parameters.

Area	Description
1	This header is a reminder of the catalog reference of the module and its geographical address in the PLC (rack number and position in the rack).
2	This command field indicates the current mode: Configuration.
3	This **module** level area contains the short title of the module.
4	This **channel** level field allows you to select the channel to be configured, the associated function: **Position** and the task in which the implicit exchange objects are exchanged: **MAST** or **FAST**.

Channel configuration parameters input area

The lower right part of the screen gives access to parameter entry.



NOTE: For each parameter, the limits are displayed in the status bar.

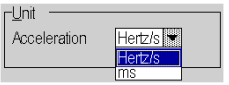
Configuring user units

Introduction

Movements and positions are always expressed in numbers of pulses or increments. Speeds are always expressed in pulses per second (Hertz).

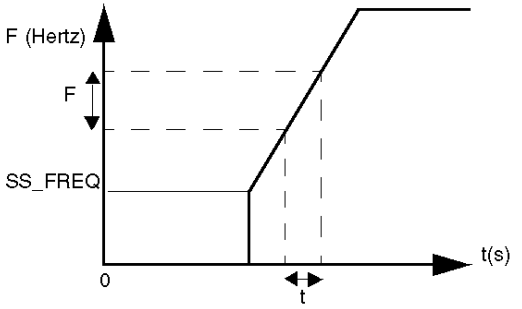
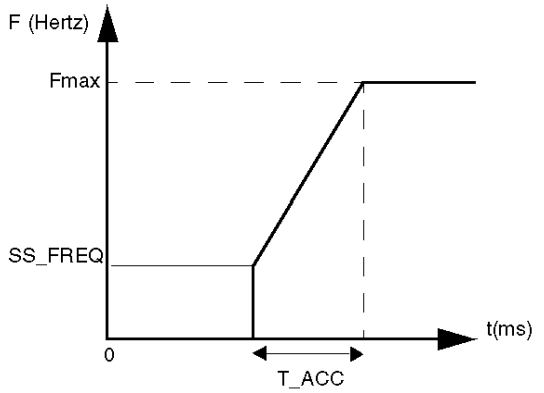
Units pick list

The units pick list is as follows:



Description

Two selections are possible.

Unit	Meaning
Hertz/s	When this selection is enabled, we will speak of the moving part's gradient of acceleration and deceleration. Acceleration in Hertz/s: is equal to the speed gradient, dF/dt 
ms	When this selection is enabled, we will speak of the duration of acceleration and deceleration of the moving part in miliseconds. Acceleration in ms: is equal to the acceleration time for the speed to go from SS_FREQ to maximum speed 

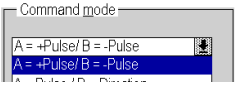
Configuring the translator's command mode

Introduction

The speed setpoint is sent to the translator in order to control the step by step motor. This menu allows you to determine the manner in which this information is transmitted.

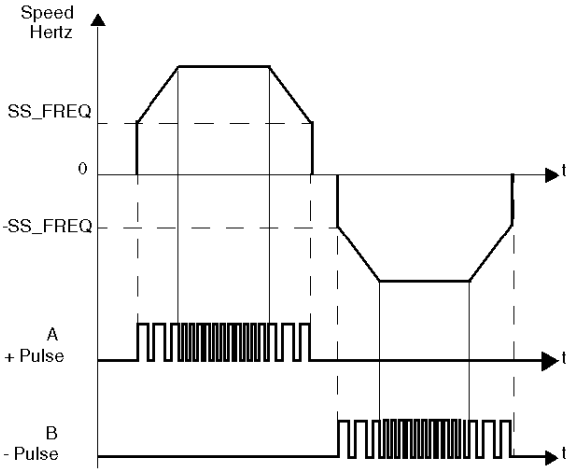
Command mode pick list

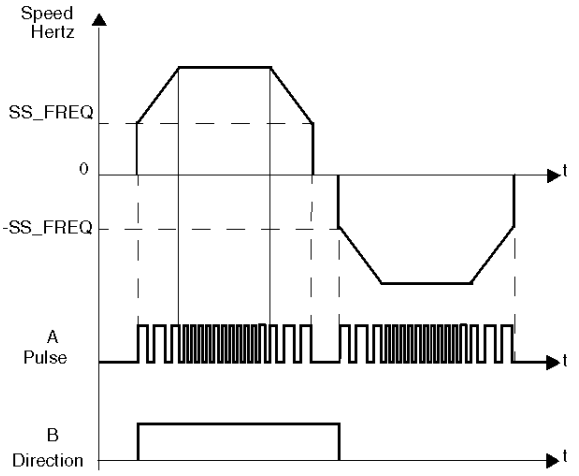
The command mode for the translator is as follows:



Description

Two selections are possible.

Selection	Meaning
A = + Pulse B = - Pulse	A pulse on A is a movement command (a step) in the positive direction along the axis, a pulse on B is a movement command in the negative direction along the axis. + pulse, - pulse mode 

Selection	Meaning
A = Pulse B = Direction	In this pulse mode, A is a step movement command, the direction of the movement is indicated by B: ● if B is set to 1 the movement is in the positive direction, ● if B is set to 0 the movement is in the negative direction. Pulse / direction mode 

Configuring control parameters

Introduction

The control parameters fields allow you to define the maximum speed and the maximum acceleration of the axes control.

NOTE: the terms speed and frequency are used interchangeably to characterize the ideas of speed.

Control parameters selection screen

The control parameters selection screen for the translator is as follows:

Description

Two fields are to be completed.

Selection	Meaning
Max. speed	The maximum speed (frequency) depends on the translator – motor – moving part as a whole. The pulse generation circuit has a resolution of 1024 pulses on the frequency dynamic (zero frequency included). The selection of a maximum speed has an influence on the frequency resolution of the channel. The following list gives the frequency resolution (Minimum Frequency) for a given max. frequency interval: ● [1 Hz …936 Hz] mini frequency 0.92 Hz ● [937 Hz …1873 Hz] mini frequency 1.83 Hz ● [1874 Hz …4682 Hz] mini frequency 4.58 Hz ● [4683 Hz …9365 Hz] mini frequency 9.16 Hz ● [9366 Hz …46829 Hz] mini frequence 45.78 Hz ● [46830 Hz …93658 Hz] mini frequence 91.55 Hz ● [93659 Hz … 187316 Hz] mini frequency 183.11 Hz **Example**: for a max. frequency of 20 KHz, the resolution (Min. Frequency) is 45.78 Hz.
Max. acc.	The effective acceleration of the axis that is defined in adjustment, must always be less than or equal to the maximum acceleration defined in the configuration. The TSX CFY 11 and 21 modules are capable of modifying the rate of acceleration or deceleration every 5 ms. The dynamic resolution is 63 point, which means that when the selected unit of acceleration is the Hertz/s, in a given interval of maximum speed the acceleration can be between 1 and 63 times the minimum acceleration. The following list gives the authorized minimum accelerations for a given speed interval: ● [1 Hz …936 Hz] minimum acceleration 183 Hz/s ● [937 Hz …1873 Hz] minimum acceleration 366 Hz/s ● [1874 Hz …4682 Hz] minimum acceleration 916 Hz/s ● [4683 Hz,9365 Hz] minimum acceleration 1831 Hz/s ● [9366 Hz,46829 Hz] minimum acceleration 9155 Hz/s ● [46830 Hz …93658 Hz] minimum acceleration 18311 Hz/s ● [93659 Hz …187316 Hz] minimum acceleration 36621 Hz/s When acceleration is given in **ms**, the **maximum acceleration** corresponds to the minimum time taken to reach **maximum speed** when accelerating from the start stop frequency (SS_FREQ).

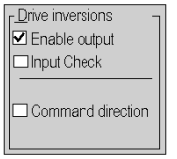
Configuring translator inversion

Introduction

The translator is controlled by the channel of the TSX CFY 11 or 21 module. It is possible to configure the logic state of the translator **enable output** and the translator **monitor input** as well as the movement **direction of the control** of the signals **A** and **B**.

Configuration screen for translator inversions

The configuration screen for translator inversions is as follows:



Description

Three selections are envisaged.

Field	Meaning
Enable Output	For translators having input **validation**, when this box is not selected, output validation is set to 1, when the translator is enabled. If not, it is set to 0. For translators having input **disabling**, when this box is selected, output validation is set to 0, when the translator is enabled. If not, it is set to 1.
Input Monitoring	When this box is not selected, when the input monitor is set to 1, the translator is unavailable. If not, it is available (configuration case for Phytron MSD/SD translator) When this box is selected, when the input monitor is set to 1, the translator is available. If not, it is unavailable.
Command direction	When this case is not selected, the direction of the signals **A** and **B** is that specified in the section Configuring the command mode *(see page 144)*. When this box is selected, the logic of the command is inverted. The selection **A=+ Pulse / B=-Pulse** becomes **A=-Pulse / B=+ Pulse** and the selection **A=Pulse / B=Direction** is such that **B** set to 1 commands a negative direction along the axis and **B** set to 0 commands a positive direction along the axis.

Configuring translator boost

At a Glance

Some translators have a boost input that can be configured on the TSX CFY 11 and 21.

Boost configuration screen

The configuration screen for translator boost is as follows:

Description

Two selections are conceivable:

Field	Meaning
Automatic control	For translators with a **boost** input, when this box is not checked, the translator boost is controlled by the object %Qr.m.c.14 BOOST *(see page 209)*. **NOTE:** The BOOST command remains active in automatic control mode. If the Automatic control option is checked, it is important to refrain from using this command to avoid any conflicts. For translators with a **boost** input, when this box is checked, the translator boost is automatically activated in the acceleration or deceleration phase of the moving part.
Inversion	When this box is not checked, the translator boost is active when the boost output is set to 1. When this box is selected, the translator boost is active when the boost output is set to 0.

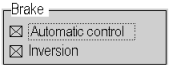
Configuring the step by step motor's brake

At a Glance

Within the context of load carrying applications, it is possible to use a brake on the step by step motor.

Brake configuration screen

The brake configuration screen is as follows.



Description

Two selections are conceivable.

Field	Meaning
Automatic control	When this box is not selected, the brake is controlled by the object %Qr.m.c.13 BRAKE *(see page 209)*. **NOTE:** The BRAKE command remains active in automatic control mode. If the Automatic control option is checked, it is important to refrain from using this command to avoid any conflicts. When this box is checked, the control of the step by step motor's brake is automatically activated when the moving part is stopped and deactivated when it starts.
Inversion	When this box is not selected, the brake output is set to 0 when the brake command is active, if not it is set to 1 (24V) in order to deactivate the brake. When this box is checked, the brake output is set to 1 when the brake command is active, if not it is set to 0.

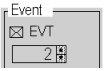
Configuring event tasks

At a Glance

When you want to carry out a complementary process that uses the reflex entry, it is necessary to configure an event task associated to the axes control channel.

Event configuration screen

The configuration screen for the event task is as follows:



Description

Two fields are to be completed.

Field	Meaning
Evt	When this box is selected, this signifies that you want to associate an event task to the axes control channel.
Task number	This number indicates the number of the event task that will be attached to the axes control channel. This number is between 0 and 31 for the TSX P57 1••, between 0 and 63 for the TSX P57 2••, TSX P57 3•• and TSX P57 4•• and between 0 and 127 for the TSX P57 5••.

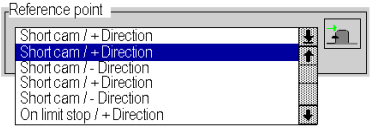
Reference point configuration

At a Glance

In order for a movement to be transformed into a position, it is necessary to assign a known dimension (generally selected as equal to 0) to a specific point on the axis. This operation is called setting the reference point. An axis on which a reference point has been set is known as "referenced".

Illustration of the Reference point field

The reference point pick list is as follows.



Description

The Reference point field defines the type and direction of the reference point.

The types **short cam** and **long cam** are linked by connecting a reference point detector on the **cam reference point** input. The types **end of run limit** assume that end of run detectors have been installed.

Possibilities	Approach speed (1)	Reference Point Speed	Icon
Short cam, + direction	F	F	(2) 
Short cam, - direction	F	SS_FREQ	(2) 
Long cam, + direction	F	SS_FREQ	(2) 

Possibilities	Approach speed (1)	Reference Point Speed	Icon
Long cam, - direction	F	SS_FREQ	(2) 
End of run limit, + direction	F	SS_FREQ	(2) 
End of run limit, - direction	F	SS_FREQ	(2) 

(1) F is the speed programmed in the instruction in automatic mode or the FMANU speed (defined in the adjustment screen) in manual mode. This speed can be modulated by the CMV (Speed Modulation Coefficient).

(2) The icon illustrates the reference point.

Reference point command

The reference point command is carried out:

- in automatic mode, using instruction code 14: set reference point,
- in manual mode, using the command SET_RP: set manual reference point.

If SS_FREQ is zero, and if the reference point speed is SS_FREQ, then the actual reference point speed is the smallest that the module can generate in the selected range.

NOTE: SS_FREQ = the start stop frequency.

Forced reference point

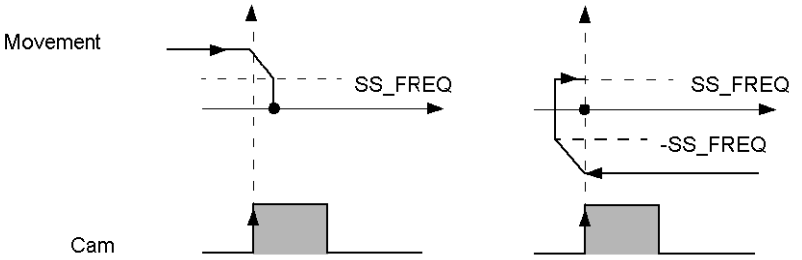
There is also a forced reference point mechanism:

- command G62 in Auto mode,
- command RP_HERE in Manu mode.

This setting of the reference point consists in forcing the position to a specified value. This operation does not entail any movement and therefore does not take account of the type of RP selected.

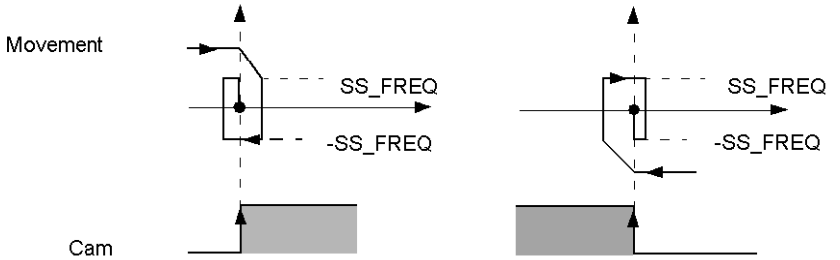
Short cam reference point

The following table gives a detailed description of short cam reference points

Type	Short cam	
Direction	+Direction	-Direction
Icon		
		

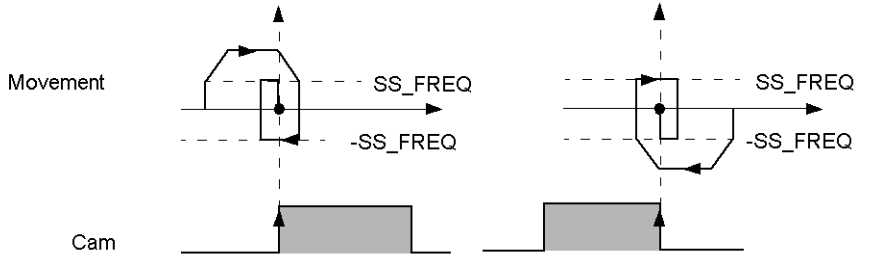
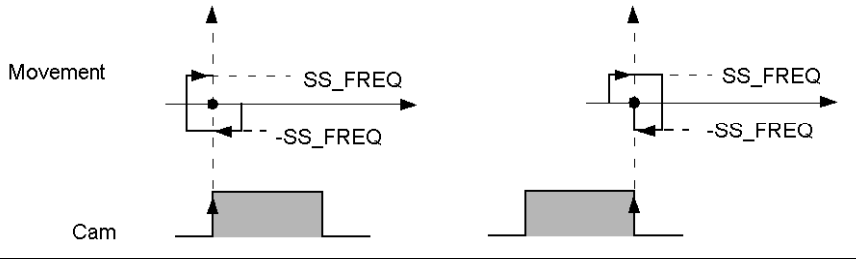
Long cam reference point

The following table gives a detailed description of long cam reference points

Type	Long cam	
Direction	+ Direction, start away from the cam	- Direction, start away from the cam
Icon		
		
Direction	+ Direction, start towards the cam	- Direction, start towards the cam
Icon		
		

Reference point end of run limit

The following table gives a detailed description of end of run limit reference points

Type	End of run limit	
Direction	+ Direction, start away from the cam	- Direction, start away from the cam
Icon		
		
Direction	+ Direction, start towards the cam	- Direction, start towards the cam
Icon		
		

Validating the configuration parameters

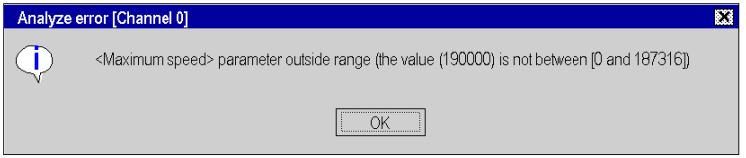
At a Glance

When all the configuration parameters have been defined, the configuration must be confirmed using the **Edit → Confirm** command or by activating the associated icon:

Invalid configuration parameters

If one or several parameter values are outside the permitted limits, an error message is displayed to signal the invalid parameter.

For example the **Maximum speed** value is invalid:



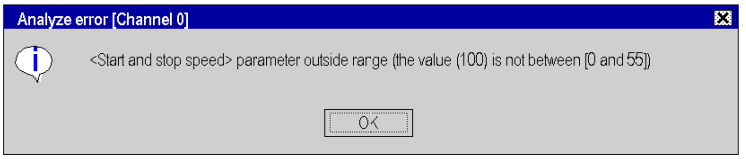
The invalid parameters must be corrected before your configuration can be confirmed.

NOTE: In the configuration screens, invalid parameters are shown in red. The grayed out parameters cannot be modified because they depend on faulty parameters.

Invalid adjustment parameters

The first time the configuration is confirmed, the adjustment parameters are initialized. If subsequent modifications made to the configuration values result in the adjustment parameters being incorrect, an error message is displayed to signal the parameter in question.

For example, the speeds are outside the limits:



You must access the adjustment screen, correct the invalid parameter, then confirm.

Acknowledgment of confirmation

Your configuration has been acknowledged when:

- all the configuration parameters are correct,
- all the adjustment parameters are correct,
- you have confirmed everything from the main screen of the configuration editor.

Chapter 9
Adjusting step by step axes control

Subject of this chapter

This chapter describes the principles of parameter adjustment: access to the screens, description of parameters and the adjustment procedure.

What Is in This Chapter?

This chapter contains the following topics:

Preliminary operations before adjustment

Preliminary conditions

- TSX CFY module(s) installed on the PLC,
- Axes control application(s) connected to the TSX CFY module(s),
- Terminal connected to the PLC via the terminal port or via network,
- Axes control configuration and program completed and transferred to the PLC processor,
- PLC in RUN mode. You are advised to disable the movement command application program (by using, for example, a program execution condition bit), in order to facilitate the adjustment operations.

Preliminary checks

- Check the cabling,
- Check that the movements can take place safely,
- Check that the mechanical stops are cabled conforming to the safety rules (generally they act directly on the power supply sequence of the translator),
- Check and adjust the translator according to manufacturer’s instructions,

Accessing adjustment parameters

At a Glance

To access adjustment parameters, use the command **Adjustment** from the **View** menu in the configuration screen of the TSX CFY. You can also select **Adjustment** in the module zone in the configuration or debugging screens.

Accessing parameters

The adjustment screen allows you to select the channel to be adjusted and gives you access to the current or initial parameters:

Command	Function
Choose axis	Select channel 0 for example.
	This button allows you to display either the current parameters or the initial parameters.

Initial parameters

The initial parameters are:

- The parameters entered (or defined by default) in the configuration screen in local mode. These parameters have been enabled at configuration and transferred to the PLC.
- The parameters taken into account at the time of the last reconfiguration in connected mode.

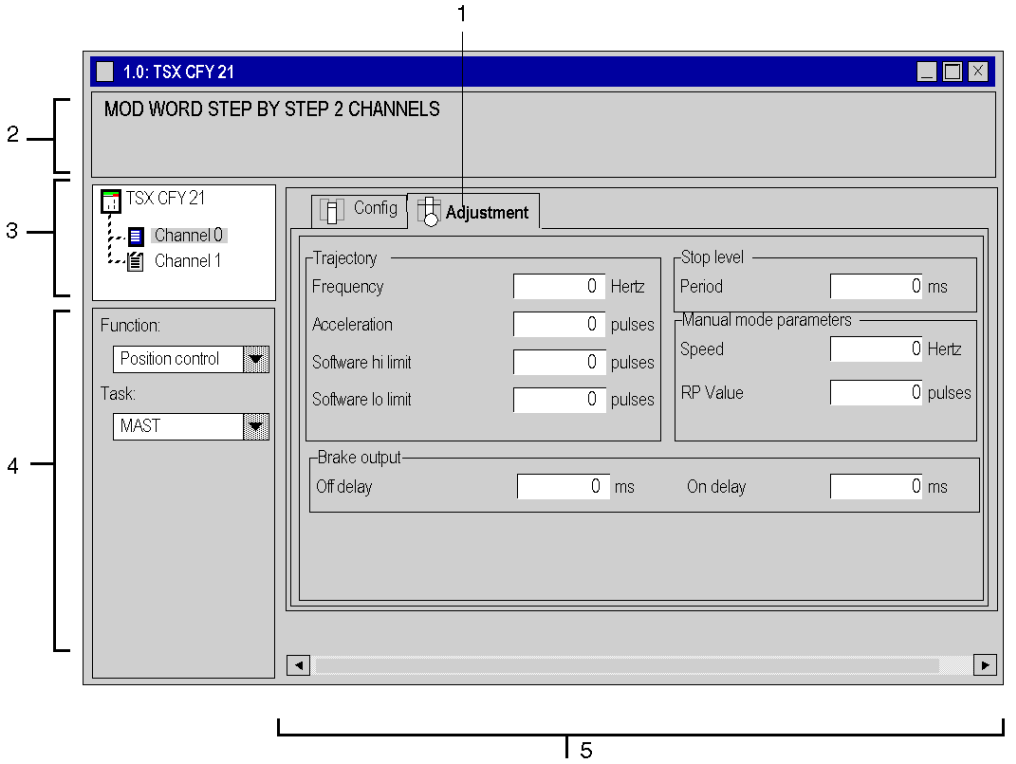
Current parameters

The current parameters are those which have been modified and enabled from the adjustment screen in connected mode (or by the program through an explicit exchange). These parameters are replaced by the initial parameters at the time of a cold restart.

NOTE: It is imperative to follow a session of determining adjustment parameters with an operation to save parameters.

Illustration

The figure below shows an adjustment screen.



Description

The table below presents the various elements of the adjustment screen and their function.

Number	Element	Function
1	Tabs	The tab in the foreground indicates the current mode (**Adjustment** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Adjustment** ● **Configuration** ● **Debug** (or diagnostics), accessible in online mode only
2	**module** zone	Summary of the abbreviated heading of the module.
3	**Channel** zone	Is used: ● By clicking on the reference number, to display the tabs: ❍ **Description** which gives the characteristics of the device. ❍ **I/O Objects** *(see EcoStruxure™ Control Expert, Operating Modes)* which is used to presymbolize the input/output objects. ❍ **Fault** which shows the device faults (in online mode). ● To select the channel, ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** zone	Allows you to choose the axis control function and task associated with the channel : ● **Function : Position control**. By default, no function is configured (**None**). ● **Task:** defines the task (**MAST** or **FAST** or **AUX0/1**) in which the objects (implicit exchange) of the channel will be exchanged.
5	**Adjustment** zone	This zone contains the various values of the adjustment parameters.

Trajectory adjustment

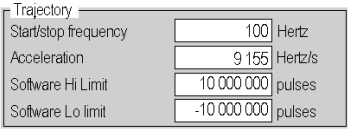
At a Glance

The adjustment screen supports characterization of the trajectory on the axis:

- start and stop frequency,
- acceleration,
- software hi limit,
- software lo limit.

Illustration

The input area for trajectory characteristics is as follows.



Description

The table below describes the dialog box for inputting trajectory characteristics.

Field	Description
Start/stop frequency	Called **SS_FREQ**, this is the minimum speed of the movement of the moving part. If **FMAX**, the maximum speed defined in configuration, is lower than 4 KHz, **SS_FREQ** must be contained between **0** and **FMAX**. Otherwise, **SS_FREQ** must be contained between **0** and **4KHz**. When **SS_FREQ** is left at zero, the start and stop frequency is the smallest frequency in the range *(see page 146)*.
Acceleration	Called **ACC**, this is the moving part's acceleration and deceleration gradient or the period of acceleration needed to go from the speed **SS_FREQ** to **FMAX** (See *Configuring control parameters, page 146*). When the user-defined unit is **Hertz/s**, this parameter must be between the lower acceleration limit for the range of **max speed** and the acceleration entered in configuration (See *Description, page 147*). When the user-defined unit is **ms**, this parameter must be between the value of the **max acceleration** entered at configuration and **5000 ms**.
Software Hi Limit	Called **SLMAX** and expressed by number of pulses, this is the maximum position of the movement of the moving part in the positive direction.
Software Lo limit	Called **SLMIN** and expressed by number of pulses, this is the minimum position of the movement of the moving part in the negative direction.
The software limits must respect the following differences: • **SLMIN** less than or equal to **SLMAX** • **SLMIN** and **SLMAX** between 16 777 216 and 16 777 215 When the two software limits **SLMIN** and **SLMAX** are zero, the monitoring of these software limits is not activated. Movements can be executed on the whole of the counting range from -16 777 216 to + 16 777 215 without, however, exceeding one of these limits.	

Adjusting brake output

At a Glance

The adjustment screen is used to parameterize the brake output, when automatic brake management has been selected at configuration. The delay can be set:

- on activation,
- on deactivation.

Illustration

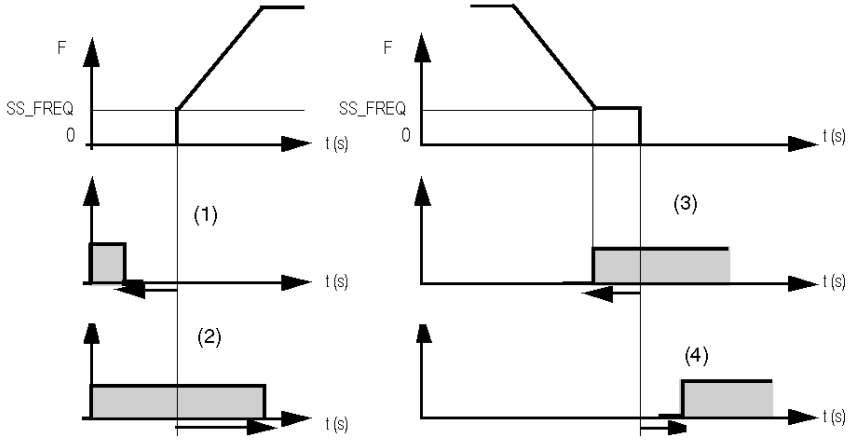
The zone for inputting delays on activation and deactivation is as follows.

Description

The table below describes the dialog box for entering delays on activation and deactivation.

Field	Description
Delay on activation	This parameter is between –1000 and 1000 milliseconds. A negative value indicates an anticipation of the end of a movement. A positive value indicates a delay.
Delay on deactivation	This parameter is between –1000 and 1000 milliseconds. A negative value indicates an anticipation of the beginning of a movement. A positive value indicates a delay.

Diagram showing operation of the delay on output **Brake**.



1 Anticipation, delay on deactivation, negative value.
2 Delay, delay on deactivation, positive value.
3 Anticipation, delay on activation, negative value.
4 Delay, delay on activation, positive value.

Adjusting the stop stage

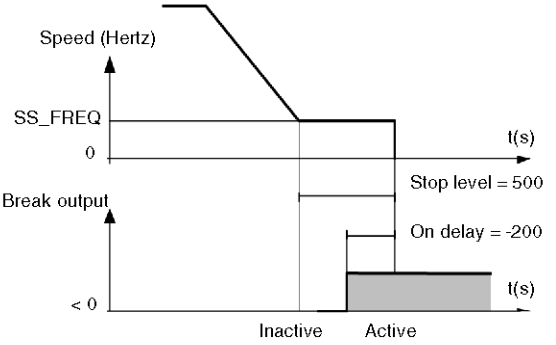
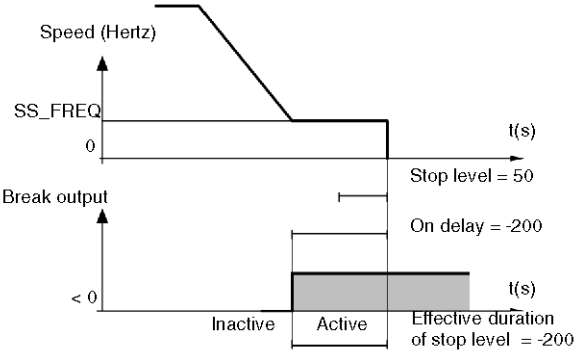
At a Glance

This field is the period of the **stop stage** order when the speed is equal to the start and stop speed, **FDA**. This period must be between **0** and **1000 ms**.

There is a relation between the duration of the **stop stage** and the brake activation delay (when this delay value is negative) in cases when **automatic management** *(see page 150)* of the brake is configured.

Operation

How axes control operates with regard to the duration of the stop stage.

If the duration of the stop stage is	Then
greater than the brake activation delay	the brake is activated as soon as the speed has reached start/stop speed, SS_FREQ. The duration of the stop stage is that of the adjustment parameter. 
less than the brake activation delay	the duration of the stop stage is forced to the duration of the brake activation, regardless of the value fixed in adjustment mode. The brake is activated as soon as the speed of the moving part has reached start/stop speed, FDA. 

Adjusting manual mode parameters

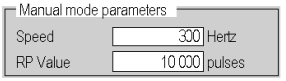
At a Glance

Adjustment of manual mode parameters defines the action of the moving part in manual mode *(see page 127)*. There are two parameters:

- speed,
- the value of the reference point.

Illustration

The manual mode parameter input zone is as follows.



Description

The table below describes the dialog zone for inputting manual mode parameters.

Field	Description
Speed	This is the movement speed, **MAN_SPD**, of the moving part in manual mode. The value of the field determines the speed of the moving part in manual mode, when it is controlled by instructions **JOG+, JOG-, INC+, INC-** and the approach and stopping speed in **SET_RP**, etc. The value of this field must be between the start/stop speed **SS_FREQ** and the **maximum speed FMAX** set in the configuration *(see page 146)*. Just as in automatic mode, the actual movement speed is modulated by the **Speed Modulation Coefficient** (CMV).
RP Value	This is the value loaded into the current position when taking a reference point in manual mode. The value of the Reference point field, **RP_POS**, is transferred to the immediate position, **X_POS** when taking a reference point in manual mode with axes control in manual mode. Generally speaking, the value of this field must be between **SLMIN** and **SLMAX**. In the special case, where **SLMIN=SLMAX=0**, the value of this field must be between -16 777 216 and 16 777 215.

Confirmation of adjustment parameters

Introduction

When you have entered the adjustment parameters, you must confirm these parameters using the command **Edit/Confirm** or by clicking the icon. 

Parameters outside the limits

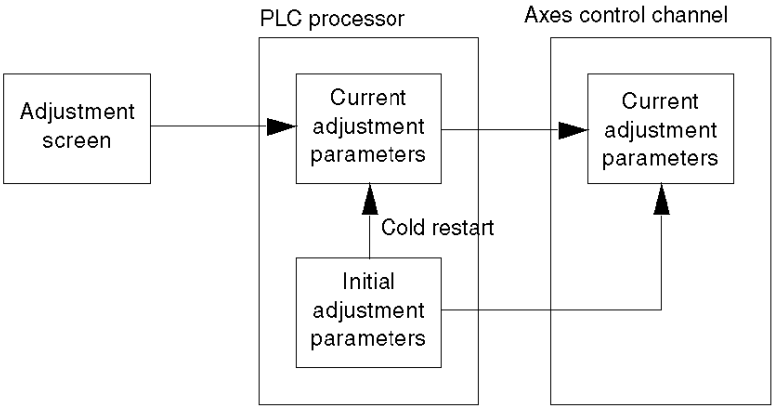
If one or several parameter values are not within the authorized limits, an error message appears referencing the parameter concerned.

You must fix the parameter(s) that are at fault, then execute a further confirmation.

No modification of the configuration parameters

If you have not modified the configuration parameters, the modification of the adjustment parameters does not interrupt the working of the axis but changes its behavior.

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

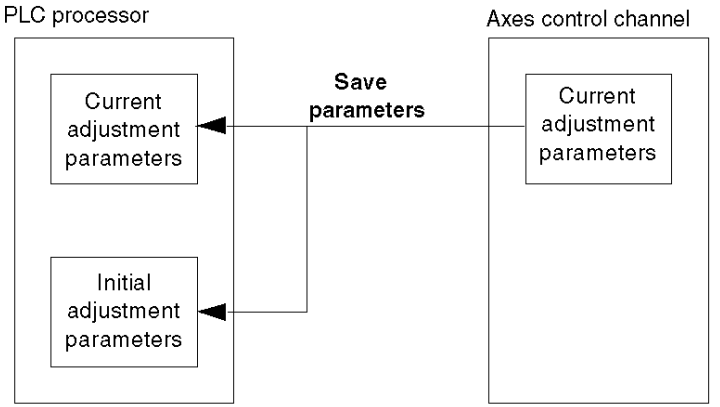


NOTE: During a cold start the current parameters are replaced by the initial parameters.
The initial parameters can be updated by the save command or by a reconfiguration operation.

Saving / Restoring adjustment parameters

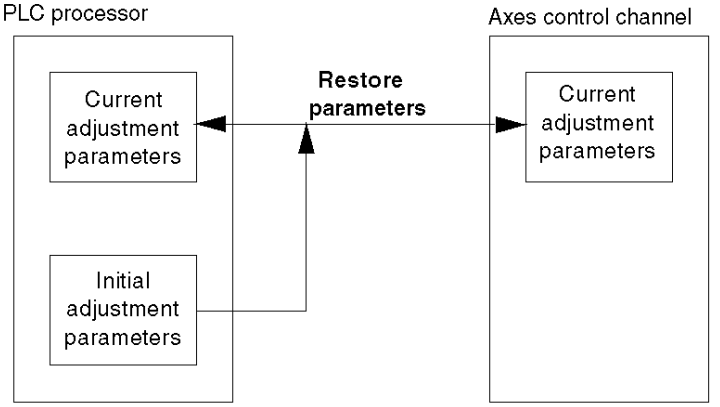
Saving parameters

To save current parameters (updating initial parameters), activate the command **Services → Save parameters** .



Restoring parameters

To replace the current parameters by the initial parameters, activate the command **Services → Restore parameters**.



NOTE: The instruction RESTORE_PARAM allows the application to carry out this restoration operation. The restoration is also carried out automatically at the time of a cold start.

Reconfiguration in connected mode

Introduction

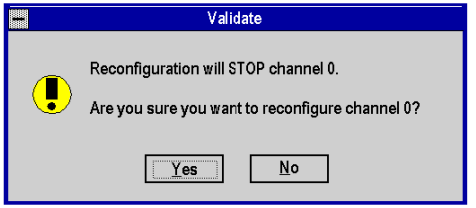
When configuration parameters are modified, you must confirm these parameters using the command **Edit → Confirm** or by clicking on the icon. 

Parameters which can be modified in connected mode

Only parameters that are not grayed out can be modified in connected mode. Other parameters like activation of an event task must be modified in local mode. Every time that you reconfigure, the corrected resolution becomes the initial resolution.

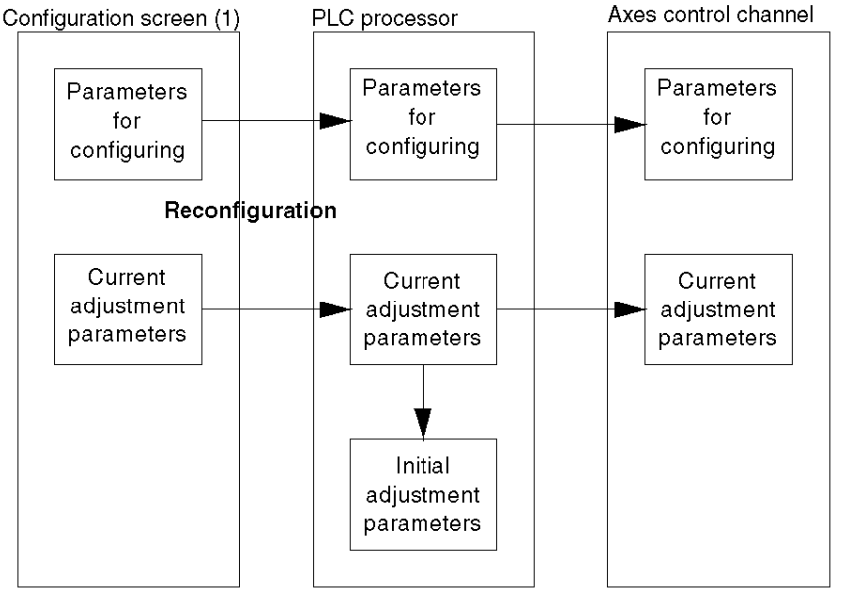
Stopping the movement in progress

All configuration in connected mode brings about a stop in the operation of the channel concerned and thus the stopping of the movement in progress. This is signaled by a dialog box:



Parameter exchange when reconfiguring

The following diagram presents the parameter exchanges when reconfiguring in connected mode:



(1) or adjustment screen if a configuration parameter has first been modified in the configuration screen.

Chapter 10
Debugging a step by step axes control program

Subject of this chapter

This chapter describes the debugging functions for an axes control channel, in the different modes: stop, direct, manual, automatic. It also describes the diagnostics screen which gives you access to possible errors.

What Is in This Chapter?

This chapter contains the following topics:

Debugging principles

At a Glance

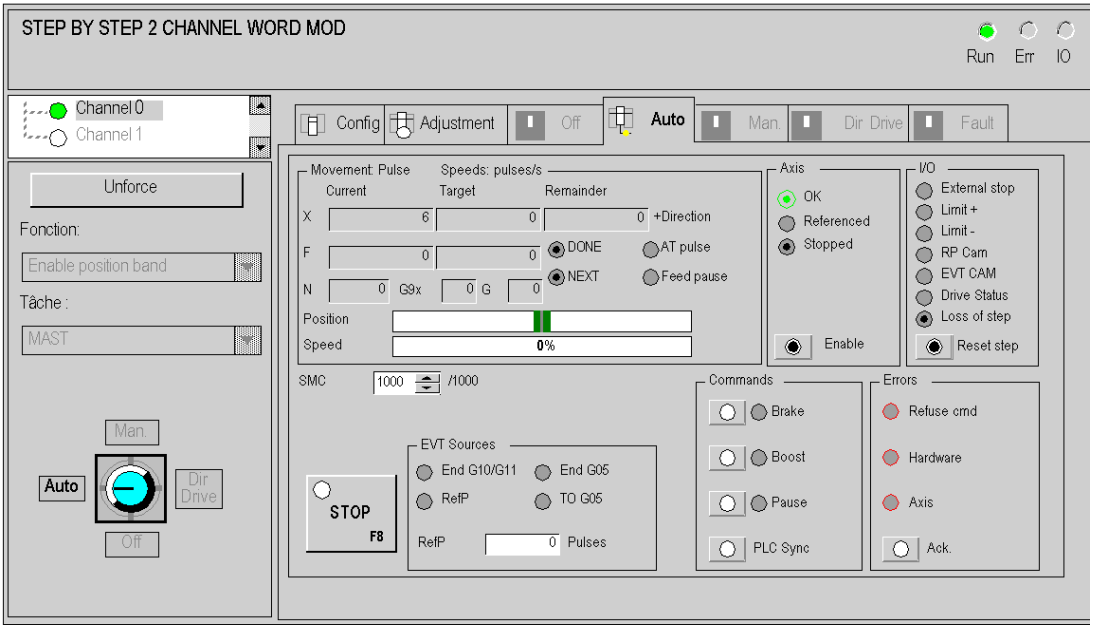
Axes control, being integrated into the Control Expert program, uses the Control Expert debug functions.

Recap of the possibilities offered by Control Expert

- Real time display and animation of the program.
  For example, in Grafcet language, if you program each movement in a step, it will be easy to know which movement is in progress.
- Inserting program breakpoints and execution points: cycle by cycle, network by network or statement by statement.
- Accessing animation tables. This allows you to display the status words and bits and control the command bits for the SMOVE. You can also force object bits and block the Grafcet development.

Application specific debug screen

The Control Expert software also offers you an application specific debug screen specific to the TSX CFY modules, which gives you access to all necessary information and commands:



User interface of the debug screen

Accessing the debug screen

You can only access the debug screen if the terminal is in **connected mode**.

If this is the case, access the debug screen in the following way:

- Select the Configuration editor,
- Select and confirm (or double click on) the rack position that contains the axes control module,
- By default in connected mode, the debug screen is displayed.

Command buttons

The operation of the command buttons  is as follows:

- For the on state commands (except JOG commands):
  Pressing then releasing the button activates the associated command. The button's internal light is lit when this command is recognized (the corresponding command bit %Q is set to 1).
  Pressing then releasing the button a second time deactivates the command. The button's internal light is extinguished when this command is recognized (the corresponding command bit %Q is set to 0).
- For on edge commands:
  The command is activated as soon as the button is pressed then released. The button's internal light lights up then is automatically extinguished.

The light situated next to the button signals the recognition of the command by the module.

Input field

All values entered into an input field must be confirmed using the  button.

Using the keyboard

You can use the keyboard to browse in the screens or activate a command:

Keys	Action
Shift F2	Allows you to move from one zone to another
Tab	In the same zone, allows you to move from one group of commands to another
Arrow keys	In a group of commands, allows you to move from one command to another
Space Bar	Allows you to activate or deactivate a command

Conflicts with the program

There can be conflicts between the Control Expert program that carries out the commands or writes the variables and the commands executed from the debug screen. In all cases, it is the last recognized command that will be active.

Animation

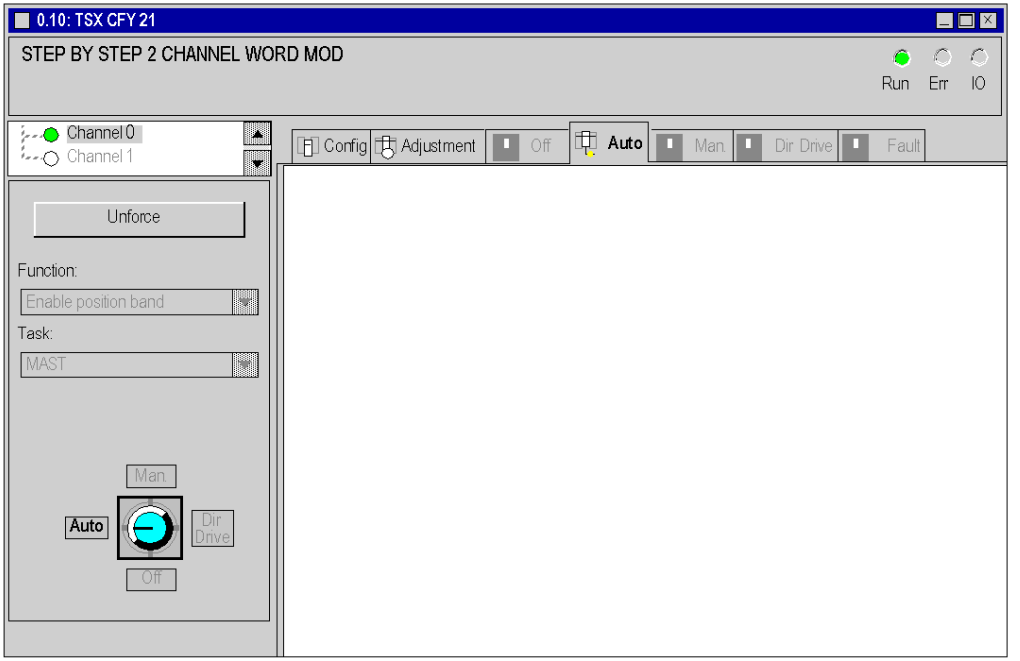
You can stop animation in the display fields:

- The command **Services → Stop animation** stops animation in the display fields and disables the command buttons. For this function you can also use the icon 
- The command **Services → Animate** reactivates animation. You can also use the icon 

Description of the debug screens

At a Glance

The debug screens have a common header, made up of a module zone and a channel/function zone.



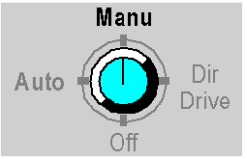
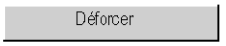
Module zone

This table describes the module zone:

Light	State	Meaning
RUN	On	Module is operating
ERR	On Flashing	Module is out of service Communication error with the processor
IO	On	Process error (AX_FLT %Ir.m.c.2 bit) Module failed. In choosing the **Default** tab associated with this button, a diagnostics window appears in order to clarify the source of the error (See *Diagnostics and maintenance, page 195*).

Channel/function zone

In addition to **Channel** and **Function** options (common to all screens), this zone also includes a mode selector button and an unforcing button:

Command	Function
	Button to select the operating mode If you want to change the operating mode, click on the name of the new mode to be selected (or click as many times as necessary on the button). Using the keyboard, select the button using the **Tab** key then press as many times as necessary on the **Space** bar. You can also access the operating modes via the **View** menu. When the selected mode is actually recognized by the module, the monitoring zone for movements in the selected mode is displayed. **Caution**: although selected, it is possible for the selected mode not to be recognized by the module channel (for example, if the PLC is in STOP mode).
	The unforcing button enables global unforcing of all forced objects.

Detailed Information on the Debug Screen

At a Glance

The debug screen is different according to the switch position selected. There are four possible choices:

- *Stop Mode (Off), page 186*
- *Direct Mode (Dir Drive), page 187*
- *Manual mode (Man.), page 188*
- *Automatic mode (Auto), page 189*

The fields and buttons that appear in the four screens are detailed below.

Description of the Movement / Speed Field

This table describes the display zones of the Movement / Speed field:

Display zone	Description
X Current	Displays the position of the moving part in number of pulses.
X Target	Displays the position setpoint of the moving part (position to be reached)
X Remainder	Displays the number of pulses left to run
F Current	Displays the speed of the moving part in number of pulses
F Target	Displays the speed setpoint of the moving part: speed to be reached (manual speed modulated by the SMC (Speed Modulation Coefficient))
N G G9	In auto mode, displays the instruction currently being executed: ● N for the number of steps ● G9 for the type of movement ● G for the instruction code
Position	This slider shows the evolution of the moving part between the limits defined in the configuration screen. The color of the slider is green and becomes red if the limits are exceeded.
Speed	This slider displays as a % the speed of the moving part in relation to the maximum speed. The color of the slider is green and becomes red if **VMAX** is exceeded.

This table describes the indicators in the Movement / Speed field:

Indicator	Status	Meaning
+Direction -Direction	/	Indicates a movement of the moving part in the positive direction Indicates a movement of the moving part in the negative direction
AT Point	On	Indicates that the movement in progress has finished and that the moving part has reached the target point
NEXT	On	Indicates that the moving part is ready to receive a movement command

Indicator	Status	Meaning
DONE	On	Indicates that the movement in progress has finished
Immediate pause	On	Indicates that the immediate pause function is activated SMC (Speed Modulation Coefficient) set to 1). At this moment the target position contains the immediate pause stop position.

Description of Axis Field

This table describes the display and command zones of the Axis field:

Indicator / Button	Status	Meaning
OK	On	Axis in operational state (no blocking fault)
Referenced	On	Referenced axis:
Stopped	On	Moving part stationary
Enable	/	This button is used to control the variable speed controller activation relay

Description of I/O Field

This table describes the display zones of the I/O field:

Indicator	Meaning
External stop	Signal state (0 or 1) on **External stop**input. The indicator is on when the **external stop** is activated, with presence of 24V on the input.
End of run +/-	Activity of **+/- limit switch** function. The indicator is on when the moving part is at the limit switch stop, with absence of 24V on the input.
RP Cam	Signal state (0 or 1) on **Reference point** input. The indicator is on when the moving part is on the cam, with presence of 24V.
Event Cam	Signal state (0 or 1) on **Event** input. The indicator is on when the moving part is on the **event cam**, with presence of 24V.
Ctrl trans	The indicator is on when the translator does not deliver the signal **ready**. The indicator is off when the translator gives the signal **OK**. The levels depend on the selection made at configuration.
Loss of step	Signal state (0 or 1) on**Loss of step monitoring** input, signal delivered by the translator. The indicator is on when the input is set to 1(cable disconnected), otherwise it is off.
Reset step	This button controls the resetting of the detection system which detects loss of translator step.

1 = indicator on, 0 = indicator off

Description of Commands

This table describes the command zone:

Command	Description
STOP	Causes the moving part to stop according to the deceleration defined on configuration
Param	Is used to enter the value of an incremental movement (INC+ or INC- command) or the forced reference point
SMC (Speed Modulation Coefficient)	Is used to enter a value from 0 to 2000 which determines the speed multiplier coefficient (0.000 to 2.000 in steps of 1/1000)

Descriptions of EVT Sources

This table describes the EVT sources zone:

Command	Description
RefP (indicator)	Indicates positioning of the event cam (reflex input) in response to SMOVE request (G07)
RefP (field)	Displays the stored PREF position
End G10/G11	Indicates that an event has occurred during execution of the G10 or G11 instruction
End G05	Indicates the end of execution of the G05 instruction
TO G05	Indicates that the Time Out, defined in the G05 instruction, has elapsed

Description of Commands Field

This table describes the buttons of the Commands field:

Command	Description
JOG-	Visual movement command in the negative direction (1)
JOG+	Visual movement command in the positive direction (1)
INC-	Incremental movement command in the negative direction, for a distance defined in the **Param** field
INC+	Incremental movement command in the positive direction, for a distance defined in the **Param** field
Manual reference point	Order to search and take manual reference point. The current position takes the value **RP Value**, defined in the adjustment screen, having found the reference point conforming to the type defined on configuration.
Forced reference point	With an incremental encoder, forced reference point order. The current position is forced to the value defined in the **Param** field This type of reference point does not cause movement of the moving part

Command	Description
Brake	Manual command for the activation or deactivation of the brake output In the case where automatic brake management is configured, it is the last activation or deactivation command edge between this manual BRAKE command (%Qr.m.c.13) and the automatic command that is taken into account.
Boost	Manual command for the activation or deactivation of the boost output In the case where automatic boost management is configured, it is the last activation or deactivation command edge between this manual BOOST command (%Qr.m.c.14) and the automatic command that is taken into account.
Pause	Stop the movement sequence at the end of the next movement with stop
PLC Sync	Order to trigger an event from the processor

(1) These commands remain active as long as the button is pressed. They are used to release the moving part outside the soft stops (after fault acknowledgment).

Description of Faults Field

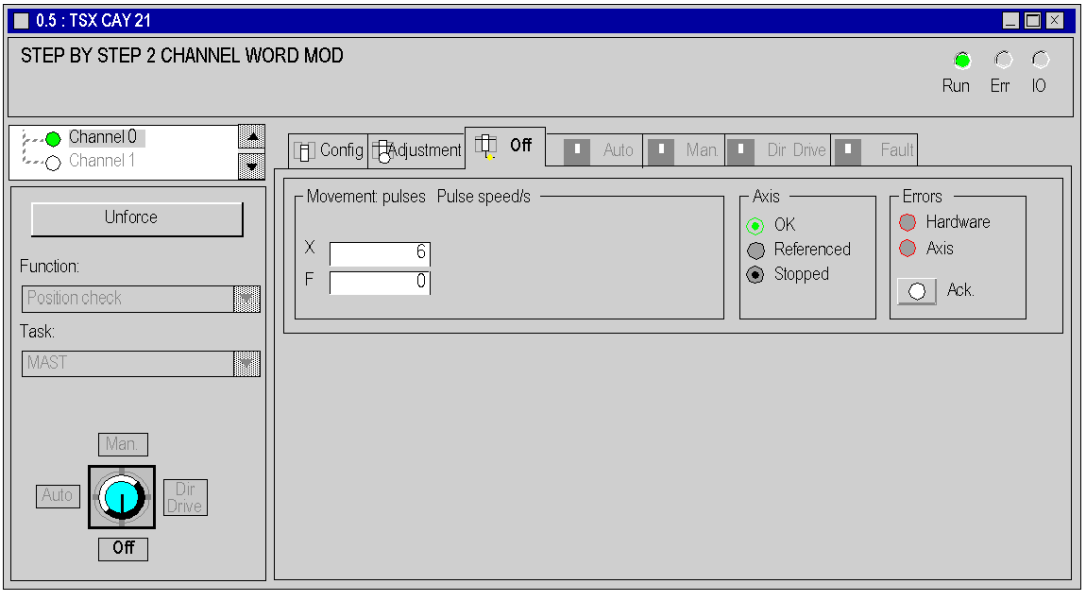
This table describes the display and command zones of the Faults field:

Indicator/ Button	Status	Meaning
Cmd refused	On	Refusal of last command
Hardware	On	External hardware fault (encoder, variable speed controller, outputs, etc.)
Axis	On	Application fault (following error, soft stops, etc.)
Ack.	/	Fault acknowledgment button. Pressing this button acknowledges all the faults that have disappeared

Stop Mode (Off)

At a Glance

In this mode, the axis control channel only reports on position and speed. The module movement is not monitored by the channel. The translator confirmation output continues to be monitored by the ENABLE (%Qr.m.c.10). command.

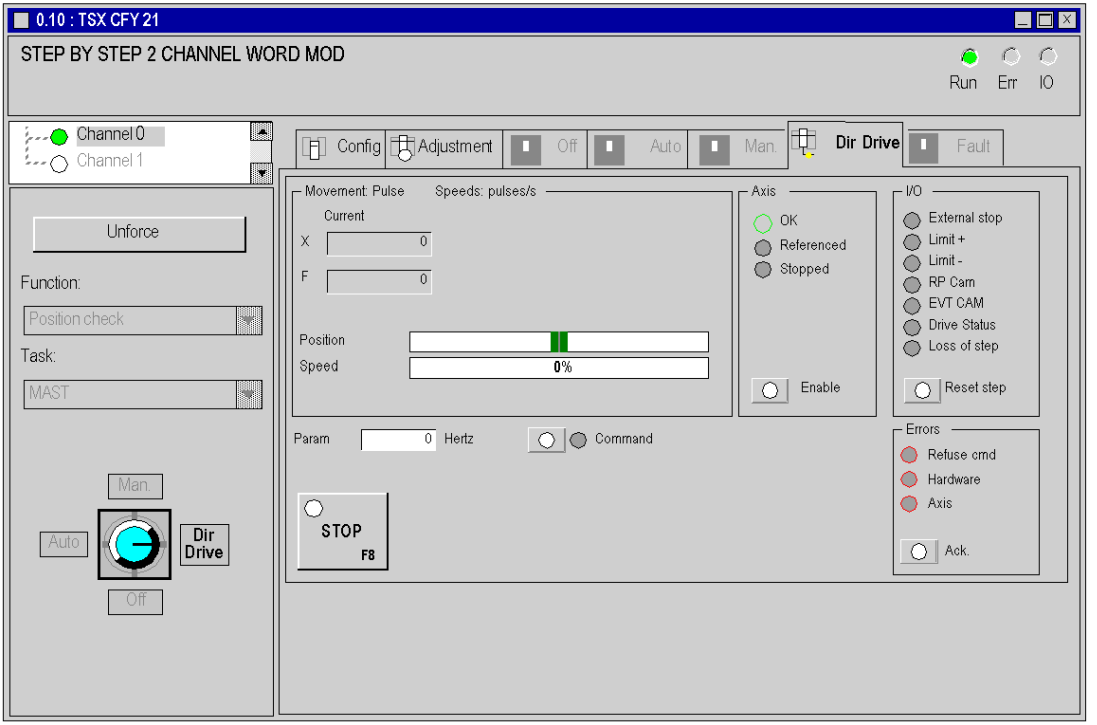


For details on the fields and buttons in this screen, see *Detailed Information on the Debug Screen, page 182*.

Direct Mode (Dir Drive)

At a Glance

Direct mode allows you to directly control the movement of the moving part, according to the movement setpoint indicated in the PARAM variable.

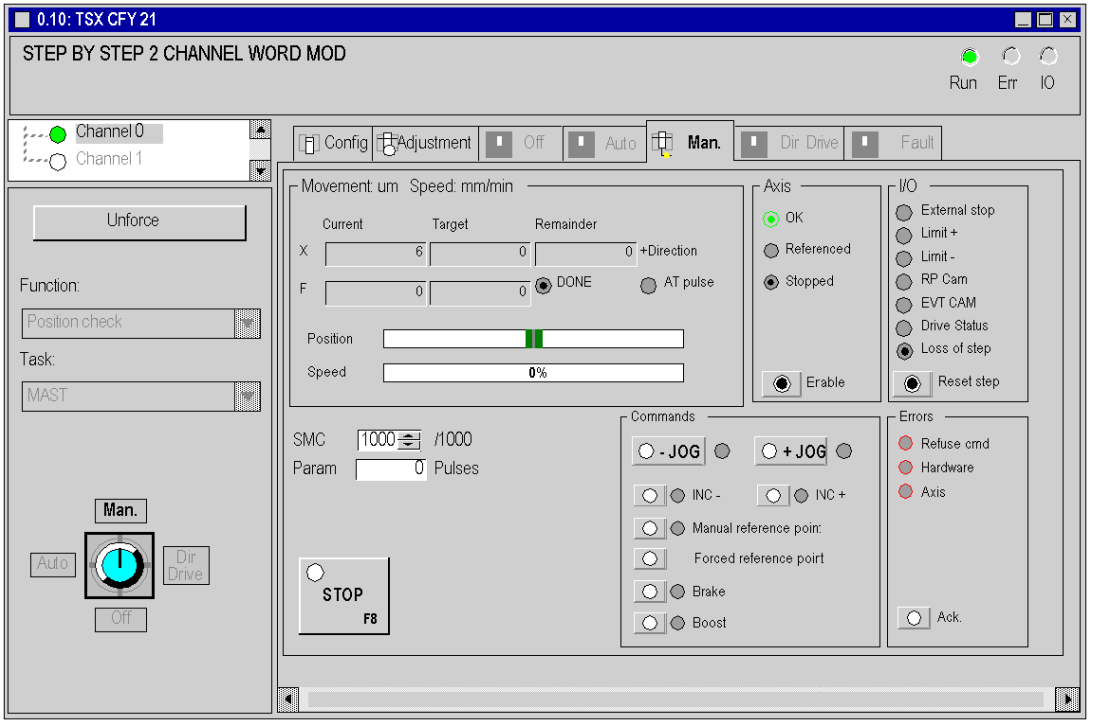


For details on the fields and buttons in this screen, see *Detailed Information on the Debug Screen, page 182*.

Manual mode (Man.)

At a Glance

Manual mode allows you to directly control the movement of the moving part, from the debug screen. To do this you can use the commands JOG+, JOG-, INC+,

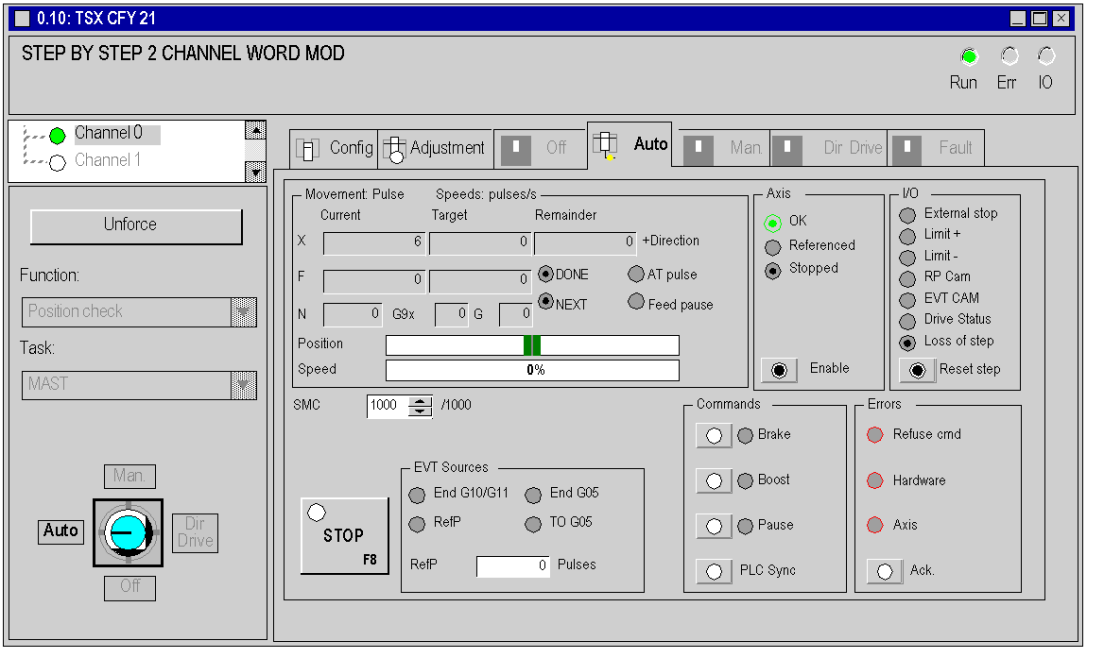


For details on the fields and buttons in this screen, see *Detailed Information on the Debug Screen, page 182*.

Automatic mode (Auto)

At a Glance

Automatic mode is the mode in which the SMOVE functions are executed.



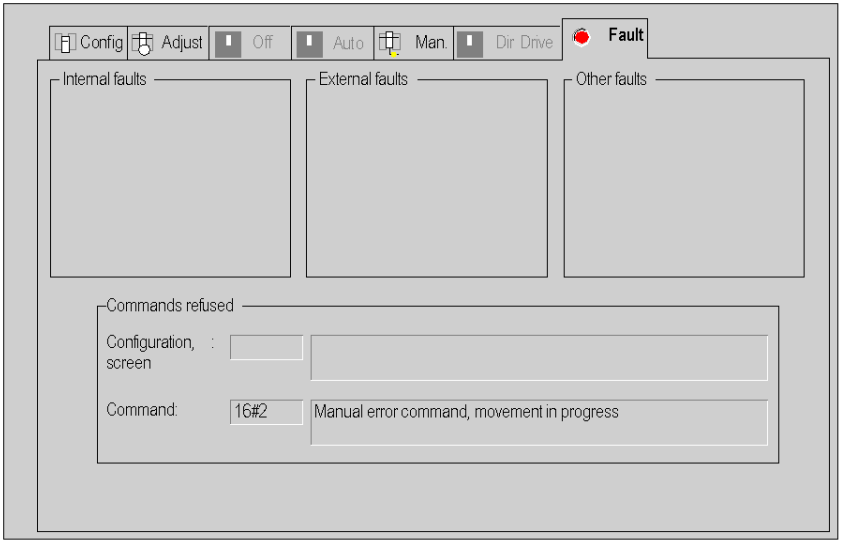
For details on the fields and buttons in this screen, see *Detailed Information on the Debug Screen, page 182*.

Channel diagnostics

At a Glance

The module screens propose a **Default** tab in online mode that provides access to details of faults detected on the module and channel.

Example of channel diagnostics.



Description of different fields

The **Default** screen offers the following fields:

Field	Description
Internal faults	Internal errors in the module that generally necessitate the replacement of the module
External faults	Errors coming from the operating part *(see page 123)*
Other faults	Faults from the application *(see page 125)*
Commands refused	Indicate the cause and the message number of a command refused *(see page 126)*

Storage, document and simulation

Storage

When you have debugged your program in connected mode, you must carry out the following save operations:

- Save the adjustment parameters if these have been modified. To do this, select the adjustment screen and use the command **Services → Save parameters**,
- Save the application on to disk using the command **File → Save**.

Documentation

The documentation for the axes control application is included in the complete documentation for the Control Expert application. This documentation allows you to group in a file:

- The program,
- The saved **Configuration** and **Adjustment** parameters.

Simulation

To operate the channels of the TSX CFY module, all you need is the Telefast discrete simulator strip, reference ABE-6TES160, supplied by the 24 volts available on the rack supply, and to connect it directly to the TSX CFY's auxiliary I/O HE10 connector via a ribbon cable.

For channel 0, add a level 1 to inputs 2, 4 and 5 (emergency stops and end of runs).

For channel 1 (only TSX CFY 21), inputs 8, 10 and 11. Leave the level 0 everywhere else.

When configuring the axis control channel, check the box **drive inversionsmonitor input**. This supports operation in the absence of any connection to the SUB D translator.

Confirm

How to carry out simulation with a TSC CFY

To operate the channels of the TSX CFY module, all you need is the Telefast discrete simulator strip, reference ABE-6TES160, supplied by the 24 volts available on the rack supply, and to connect it directly to the TSX CFY's auxiliary I/O HE10 connector via a ribbon cable.

Step	Action
1	For channel 0, add a level 1 to inputs 2, 4 and 5 (emergency stops and end of runs).
2	For channel 1 (only TSX CFY 21), add a level 1 to inputs 8, 10 and 11. Leave the level 0 everywhere else.
3	When configuring the axis control channel, check the box **drive inversionsmonitor input**. This supports operation in the absence of any connection to the SUB D translator.
4	Confirm the channel in the implementation screen in manual mode.
5	Use the buttons **JOG+** or **JOG-** to simulate movements of the moving part.

Chapter 11
Operation

Designing an operator dialog

Buttons box

To design a simple or a complex button box, you have at your disposal elementary commands and information, in the form of command and status bits and words *(see page 209)*.

Chapter 12
Diagnostics and maintenance

Subject of this chapter

This chapter describes what to do in certain maintenance situations (symptoms, diagnostics and course of action to undertake).

What Is in This Chapter?

This chapter contains the following topics:

Fault monitoring and command execution commands

Fault monitoring

You have several methods of detecting a possible fault:

- the LEDs on the front panel of the module,
- The diagnostics screens which can be accessed by means of the **DIAG** key in connected mode, from all the application-specific screens *(see page 190)* in the axes control module,
- The debug screens *(see page 180)*,
- The fault bits and the status words *(see page 209)*.

Movement commands

In order for the movement commands (in automatic or manual mode) to be executable, the following conditions must be met:

- The axis is configured and free of blocking errors,
- The variable speed controller command is active: ENABLE (%Qr.m.c.10) and the STOP command is inactive (%Qr.m.c.8)
- Automatic or manual mode is selected,
- For commands using absolute position, the position is between the limits SL_MIN and SL_MAX,
- For commands using relative position, the target calculated from the current relative position is between the limits SL_MIN and SL_MAX,
- The axes are referenced, except for reference point and JOG commands,
- The speed F must be less than or equal to FMAX,
- If the moving part is outside the end of run limits, the movement direction requested must be that which returns it to within limits.

Modifying the CMV (Speed Modulation Coefficient) parameter

If a modification of the CMV (Speed Modulation Coefficient) parameter involves a speed higher than FMAX, then this is limited to FMAX.

Sequence control

If you have not selected the option **Sequence control** at configuration, a continuous movement followed by any sequence control will continue until the soft stops.

Diagnostic help

At a Glance

You might be faced with situations that you need to resolve. The following procedure helps you to diagnose these situations and informs you of what to do.

Procedure to follow in various situations

New parameters are not recognized

Symptom	The TSX CFY module does not seem to have recognized the new parameters written by WRITE_PARAM
Diagnostics	Program a READ_PARAM instruction into your application so that you can find out the values actually used by the module. A WRITE_PARAM activated while another adjustment exchange is in progress is ignored.
What to do	Test the ADJ_IN_PROGR bit (%MWr.m.c.0.2) before any adjustment exchange

Event processing

Symptom	The event processing associated with the axes control channel is not executed
Diagnostics	Check that the whole event trace string is valid ● Event number declared in the configuration is identical to that in the event processing, ● The event source is unmasked (code M of the SMOVE command), ● Events are authorized at system level ACTIVEVT = 1 (%S38), ● Events are unmasked at system level (UNMASKEVT)
What to do	Refer to the use of events

Adjustments are lost

Symptom	Your adjustments have been lost
Diagnostics	A cold start causes the loss of the current adjustments carried out via the screen or a WRITE_PARAM instruction
What to do	Save the current adjustments using the command **Service → Save parameters** or using the instruction SAVE_PARAM

Inconsistent status words

Symptom	The status words EXCH_RPT (%MWr.m.c.1) and CH_FLT (%MWr.m.c.2) are not consistent with the status of the axis control channel.
Diagnostics	These words are only updated on the explicit request READ_STS
What to do	Program a READ_STS instruction in your application

Ineffectual commands

Symptom	The commands from the debug screen are not having any effect
Diagnostics	The application or the task is in STOP mode
What to do	Switch the application or the task into RUN mode

Non modifiable commands

Symptom	Certain commands from the debug screen are not modifiable
Diagnostics	These bits are written by the application
What to do	Use bit forcing for objects of type %Qr.m.c.d or rewrite your application so that it does not write these bits systematically (modification on transition and not on state)

Impossible to enter characters

Symptom	It is impossible to enter more than 3 characters in the numeric fields of the adjustment and configuration screens
Diagnostics	In the Windows configuration panel, you have not selected the thousand separator
What to do	In the Windows configuration panel, select the icon **International** in the field **Number format**. Activate the **Modify** command and select a thousand separator

Command refused

Symptom	In DIRDRIVE mode, after a stop after exceeding the soft stops, commands are refused.
Diagnostics	The DIRDRIVE mode is activated after a session in MANU or AUTO mode during which a reference point has been taken. The axis is referenced. Monitoring of soft stops is active. Exceeding one of these limits causes a stop with an error. No further movement in DIRDRIVE mode is accepted.
What to do	Two types of actions are possible to restart movements: ● Cause the loss of the axis reference, after the moving part has completely stopped: ❍ unset the channel, ENABLE = 0 (%Qr.m.c.10) ❍ reset the channel, ENABLE = 1 (%Qr.m.c.10) ❍ acknowledge the error (rising edge on the command ACK_FLT (%Qr.m.c.9)) ● Force the position of the moving part between the soft stops: ❍ switch for a moment to MANU mode ❍ acknowledge the error: ACK_FLT (%Qr.m.c.9) ❍ take a forced reference point at a position situated between the soft stops ❍ switch back to DIRDRIVE mode.

Bad recognition of commands in AUTO mode

Symptom	In AUTO mode, after overshooting the soft stops of the end of run, movement commands are not carried out correctly.
Diagnostics	After overshooting an end of run limit, the only commands accepted are movement commands going back in the return direction between the end of run limits.
What to do	Check whether the movement requested and incorrectly carried out was aiming to bring the moving part back between the end of run limits.

Chapter 13
Complementary functions

Dimension learning

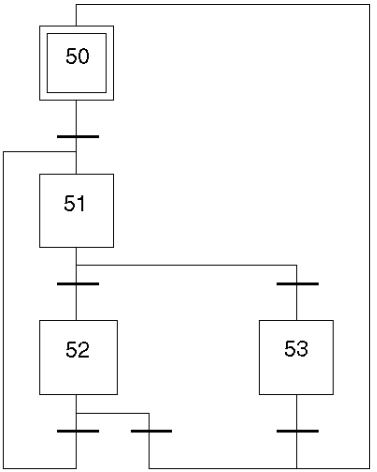
At a Glance

The following example of a Control Expert program supports learning and the use of 16 dimensions.

In the use part we will have first declared an AXIS_0 variable of the T_STEPPER_STD_ type

Dimension learning

This chart supports the programming of learning 16 dimensions.

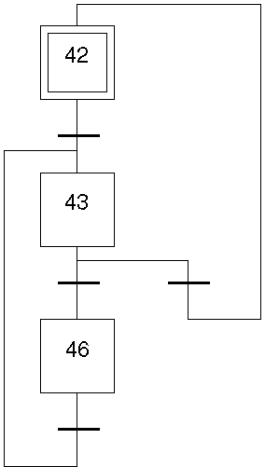



```
STEP 50 ACTION ON ACTIVATION
<stores %MW99 with a view to use it as a limit
! %MW98 := %MW99;
<Initializes the index during the learning phase
!  %MW99 := -1;
TRANSITION: X50 -> X51
! RE AXIS_0.NEXT
```

```
STEP 51 ACTION ON ACTIVATION
<updates the index
! %MW99 :=%MW99+1;
<learning of the positions
! %MD200[%MW99] := AXIS_0.POS;
TRANSITION: X51 -> X52
! %MW99 <= 16
TRANSITION: X51 -> X53
! %MW99 > 16
TRANSITION: X53 -> X50
! RE AXIS_0.DONE
TRANSITION: X52 -> X51
! RE AXIS_0.NEXT
TRANSITION: X52 -> X50
! RE AXIS_0.DONE
```

Using dimensions

This chart supports the programming of using dimensions




```
STEP 42 ACTION ON ACTIVATION
<initializes %MW97 as execution index
! %MW97 := -1;
TRANSITION: X42 -> X43
! RE AXIS_0.AX_FLT
STEP 43 ACTION ON ACTIVATION
<increments execution index
! %MW97 := %MW97+1;
<executes the next segment
! SMOVE (AXIS_0,%MW97,%KW8,%KW1,%MD200[%MW97],150000,0);
```

%KW8 : 90 movement to absolute value

%KW1 : 09 go to breakpoint

```
TRANSITION: X43 -> X46
! AXIS80.NEXT AND %MW97 < %MW98) AND NOT AXIS_0.AX_FLT
TRANSITION: X43 -> X42
! AXIS_0.DONE AND (%MW97 >= %MW98) OR AXIS_0.AX_FLT
TRANSITION: X46 -> X43
! TRUE
```

Chapter 14
Characteristics and performance

Performance characteristics and limitations

At a Glance

This section describes the performance and characteristics of the step by step control functions:

- the memory size used by a SMOVE function,
- the time taken for the execution of step by step control functions,
- the module cycle time,
- the characteristics of the low-order value movements,

Size of an SMOVE function

This table indicates the memory zones used during an SMOVE instruction as well as the corresponding size in number of 16 bit words.

	Bit memory	Data zone	Program zone
TSX CFY 11	29	390	170
TSX CFY 21	58	780	220
Additional cost of first configured channel	0	0	2290

Execution time

This table details the execution time for functions associated with step by step axes control.

Function description	Execution time
Acquisition of inputs/outputs from TSX CFY	95 micro-seconds
SMOVE function	840 micro-seconds
READ_STATUS	540 micro-seconds
READ_PARAM	460 micro-seconds
WRITE_PARAM	760 micro-seconds
SAVE_PARAM	500 micro-seconds
RESTORE_PARAM	780 micro-seconds
Recognition of an adjustment (following a WRITE_PARAM instruction)	60 ms for the TSX CFY 11 210 ms for the TSX CFY 21
Recognition of the reconfiguration of a channel	1.5 s

NOTE: the module cycle time is 10 ms

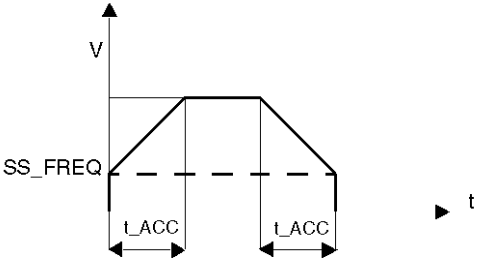
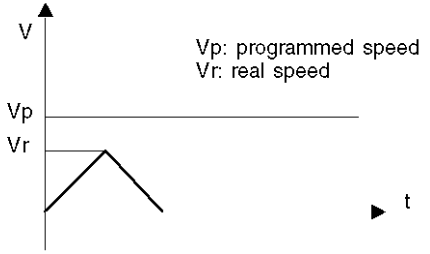
Small-sized movement characteristics

A low-order value movement corresponds to a movement that does not enable the speed specified in the instruction to be reached. The speed law resembles a triangle instead of a trapezoid.

Example of the instruction `SMOVE (Axis_ch1,1, 90, 09, X1, V ,0),`

- `Axis_ch1` is an IODDT type variable (T_STEPPER_STD),
- `X1` is the position to be reached,
- `V` the **cruising** speed at which the movement must be made.
- note `X0`, the start position of the moving part.

This table describes the possible scenarios.

If the distance to run is X0 to X1	Then
is sufficient to reach the specified speed **V**	the movement is executed according to a trapezoidal trajectory. This trajectory reveals the acceleration and deceleration periods equal to **t_ACC** 
is insufficient to reach the specified speed **V**	the movement is executed according to a triangular trajectory, the periods of the acceleration and deceleration phases are reduced in proportion to the speed. 

Chapter 15
Application-specific step by step axes control language objects

Subject of this chapter

This chapter describes the language objects associated to application-specific axes control as well as the different ways of using them.

What Is in This Chapter?

This chapter contains the following topics:

Introduction to the language objects of the application-specific step by step axes control function

General

The IODDTs are predefined by the manufacturer. They contain input/output language objects belonging to a channel of an application-specific module.

Step by step axis modules have an associated IODDT.

There is an IODDT type for the application-specific axis:

- `T_STEPPER_STD` which applies to the 2 modules TSX CFY 11/21

NOTE: IODDT variables can be created in two different ways:
- Using the **I/O objects** *(see EcoStruxure™ Control Expert, Operating Modes)* tab,
- Data Editor *(see EcoStruxure™ Control Expert, Operating Modes)*.

Language object types

This IODDT contains a set of language objects that enable their operation to be controlled and verified.

There are two types of language object:

- **implicit exchange objects**, which are automatically exchanged on each cycle revolution of the task associated to the module,
- **explicit exchange objects**, which are exchanged at the request of the application, using explicit exchange instructions.

The explicit exchanges concern the inputs/outputs of the module: measurement results, information and commands.

The explicit exchanges are used for the parametering and diagnostics of the module.

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.

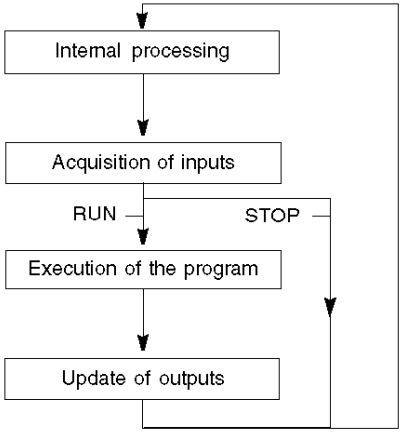
The outputs (`%Q` and `%QW`) are updated at the end of the task, only when the PLC is in RUN mode.

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 (read status words)
- WRITE_CMD (write command words)
- WRITE_PARAM (write adjustment parameters)
- READ_PARAM (read adjustment parameters)
- SAVE_PARAM (save adjustment parameters)
- RESTORE_PARAM (restore adjustment parameters)

For more details about instructions, refer to *EcoStruxure™ Control Expert, I/O Management, Block Library*.

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:

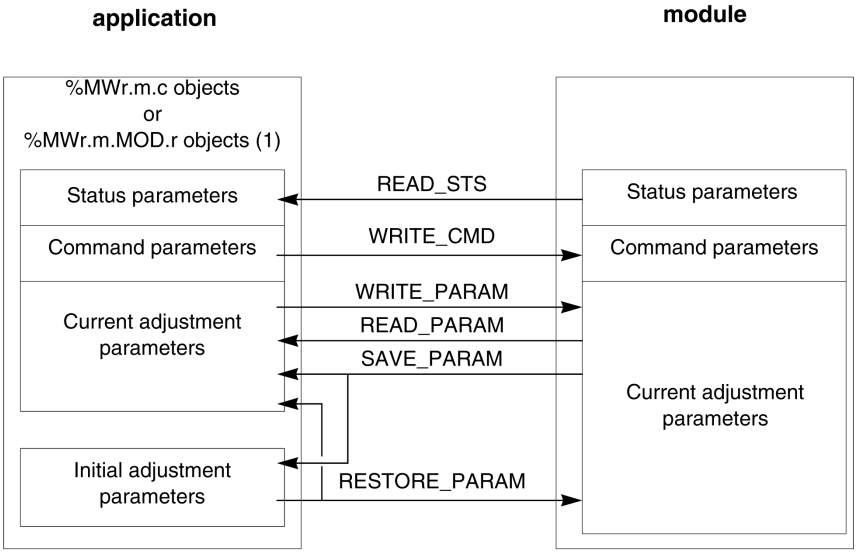
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

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 X80 analog and digital I/O modules are configured through an eX80 adapter module (BMECRA31210) in a Quantum EIO configuration. You cannot set up 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.

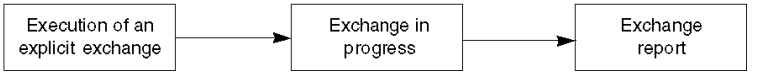
Managing Exchanges

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 217)*
- the exchange report *(see page 217)*

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. IODDTs use two words to manage exchanges:

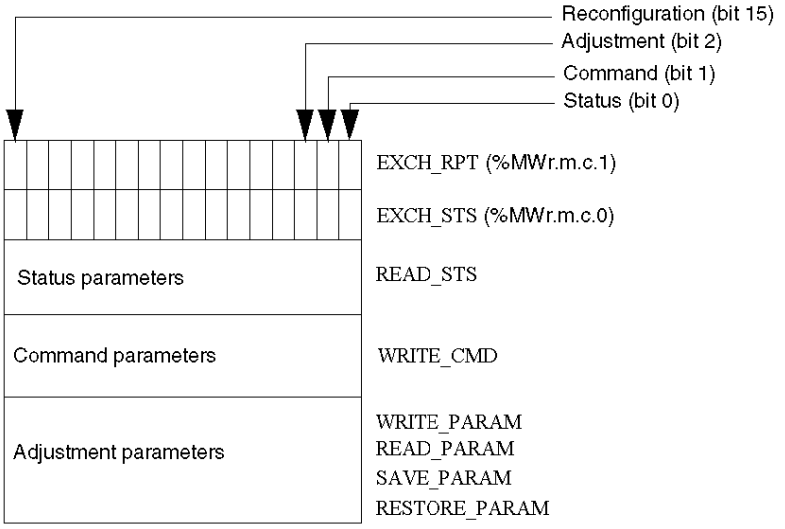
- `EXCH_STS` (`%MWr.m.c.0`): exchange in progress
- `EXCH_RPT` (`%MWr.m.c.1`): report

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 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:

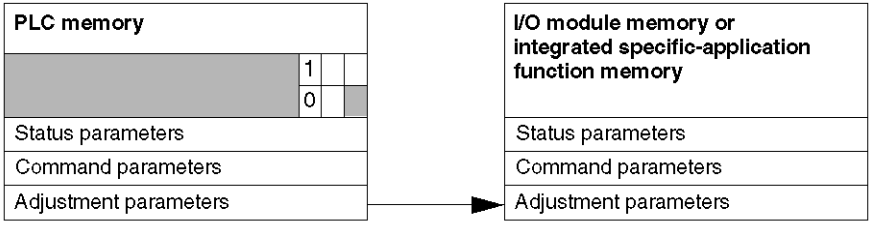
- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits 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.

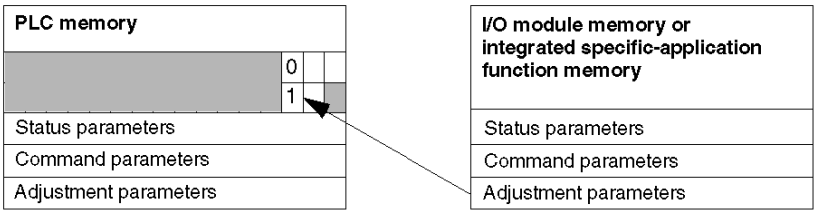
Example

Phase 1: Sending data by using the `WRITE_PARAM` instruction



When the instruction is scanned by the PLC, 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:** incorrect 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
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 detected while reading channel status words (1 = detected error)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error detected during a command parameter exchange (1 = detected error)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error dectected during an adjust parameter exchange (1 = detected error)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error detected during reconfiguration of the channel (1 = detected error)	%MWr.m.c.1.15

Counting Module Use

The following table describes the steps realized 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, wait until the bit %MWr.m.c.0.2 switches to 0.

Internal control objects (implicit exchange) of the IODDT of type T_STEPPER_STD

List of objects (implicit exchange)

The table below presents the internal status objects (implicit exchange) of the IODDT of type T_STEPPER_STD

Standard symbol	Type	Access	Description	Address
CH_ERROR	EBOOL	R	Channel fault	%Ir.m.c.ERR
NEXT	EBOOL	R	Ready to receive a new movement command (in AUTO mode)	%Ir.m.c.0
DONE	EBOOL	R	All instructions have been carried out: no instructions in the stack	%Ir.m.c.1
AX_FLT	EBOOL	R	Error present on the axis	%Ir.m.c.2
AX_OK	EBOOL	R	No errors to stop the moving part	%Ir.m.c.3
HD_ERR	EBOOL	R	Hardware fault present	%Ir.m.c.4
AX_ERR	EBOOL	R	Application fault present	%Ir.m.c.5
CMD_NOK	EBOOL	R	Command refused	%Ir.m.c.6
NO_MOTION	EBOOL	R	Moving part stationary	%Ir.m.c.7
AT_PNT	EBOOL	R	Position of the moving part on target (in the window at the point, on instruction with stop)	%Ir.m.c.8
CONF_OK	EBOOL	R	Axis configured	%Ir.m.c.11
REF_OK	EBOOL	R	Reference point taken (axis referenced)	%Ir.m.c.12
AX_EVT	EBOOL	R	Copy event physical inputs	%Ir.m.c.13
HOME	EBOOL	R	Copy the CAME physical input of the module’s reference point	%Ir.m.c.3
DIRECT	EBOOL	R	Indicates the movement direction.	%Ir.m.c.15
IN_DROFF	EBOOL	R	ARRET mode active	%Ir.m.c.16
IN_DIRDR	EBOOL	R	Direct mode active	%Ir.m.c.17
IN_MANU	EBOOL	R	manuel mode active	%Ir.m.c.18
IN_AUTO	EBOOL	R	automatique mode active	%Ir.m.c.19
ST_DIRDR	EBOOL	R	Movement in direct mode in progress	%Ir.m.c.3
ST_JOG_P	EBOOL	R	Unlimited movement in the + direction in progress	%Ir.m.c.21
ST_JOG_M	EBOOL	R	Unlimited movement in the - direction in progress	%Ir.m.c.22
ST_INC_P	EBOOL	R	Unlimited incremental movement in the + direction in progress	%Ir.m.c.23
ST_INC_M	EBOOL	R	Unlimited incremental movement in the - direction in progress	%Ir.m.c.24
ST_SETRP	EBOOL	R	Manual reference point in progress	%Ir.m.c.25

Standard symbol	Type	Access	Description	Address
ON_PAUSE	EBOOL	R	Movement sequencing suspended	%Ir.m.c.26
IM_PAUSE	EBOOL	R	Movement suspended (immediate PAUSE)	%Ir.m.c.27
STEP_FLT	EBOOL	R	Loss of step input status	%Ir.m.c.28
EMG_STOP	EBOOL	R	Emergency stop input status	%Ir.m.c.29
EXT_STOP	EBOOL	R	External stop input status	%Ir.m.c.30
HD_LMAX	EBOOL	R	End of run + limit status	%Ir.m.c.31
HD_LMIN	EBOOL	R	End of run – limit status	%Ir.m.c.32
ST_BRAKE	EBOOL	R	Image of the step by step motor brake output	%Ir.m.c.33
ST_BOOST	EBOOL	R	Image of the BOOST output activity	%Ir.m.c.34
ST_DRIVE	EBOOL	R	Translator status	%Ir.m.c.35
OVR_EVT	EBOOL	R	Event overrun	%Ir.m.c.36
EVT_G07	EBOOL	R	Event source: save position	%Ir.m.c.37
EVT_G05	EBOOL	R	Event source: end of G05 on event	%Ir.m.c.38
TO_G05	EBOOL	R	Event source: G05 timing elapsed	%Ir.m.c.39
EVT_G1X	EBOOL	R	Event source: end of G10 or G11 on event	%Ir.m.c.40
POS	DINT	R	position measured	%IDr.m.c.0
SPEED	DINT	R	speed measured	%IDr.m.c.2
REMAIN	DINT	R	number of pulses left to run	%IDr.m.c.4
SYNC_N_RUN	INT	R	number of steps in progress	%IWr.m.c.6
PREF	DINT	R	value of PREF register (only refreshed on activating event processing).	%IDr.m.c.7

Internal control objects (implicit exchange) of the IODDT of type T_STEPPER_STD

List of objects (implicit exchange)

The table below presents the internal control objects (implicit exchange) of the IODDT of type T_STEPPER_STD

Standard symbol	Type	Access	Active on	Description	Address
DIRDRV	EBOOL	R/W	Edge	Command to move in direct mode.	%Qr.m.c.0
JOG_P	EBOOL	R/W	Edge	Unlimited manual movement in the + direction	%Qr.m.c.1
JOG_M	EBOOL	R/W	Edge	Unlimited manual movement in the - direction	%Qr.m.c.2
INC_P	EBOOL	R/W	Edge	Order for incremental movement (PARAM) in the + direction	%Qr.m.c.3
INC_M	EBOOL	R/W	Edge	Order for incremental movement (PARAM) in the - direction	%Qr.m.c.4
SET_RP	EBOOL	R/W	Edge	Manual reference point (RP_POS = source value) or switch to the non-referenced state	%Qr.m.c.5
RP_HERE	EBOOL	R/W	Edge	Forced reference point of a defined value in PARAM or switch to the referenced state / calculation offset	%Qr.m.c.6
STOP	EBOOL	R/W	State	Immediate stop command (stopping of the moving part)	%Qr.m.c.8
ACK_FLT	EBOOL	R/W	Edge	Fault acknowledgment	%Qr.m.c.9
ENABLE	EBOOL	R/W	State	Confirmation of the axis controller safety relay	%Qr.m.c.10
EXT_EVT	EBOOL	R/W	Edge	Order to trigger an event from the processor	%Qr.m.c.11
PAUSE	EBOOL	R/W	State	Command to suspend movements at end of the movement in progress	%Qr.m.c.12
BRAKE	EBOOL	R/W	Edge	Command to apply brake on the step by step motor	%Qr.m.c.13
BOOST	EBOOL	R/W	Edge	Boost translator	%Qr.m.c.14
ACK_STEPFLT	EBOOL	R/W	State	Command to reset translator step monitoring	%Qr.m.c.15
MOD_SELECT	INT	R/W		mode selector	%QWr.m.c.0
SMC	INT	R/W		speed modulation Value = speed modulation setpoint value. This setpoint is between 0 and 2, in 1/1000 increments.	%QWr.m.c.1
PARAM	DINT	R/W		value of movement increment	%QDr.m.c.2

Mode selector

MOD_SELECT : mode selector

Value	Mode	Description
0	DRV_OFF	mesure mode: disables CNA output
1	DIRDRIVE	Loop control disabled mode: direct control in operation
2	MANU	manuel mode
3	AUTO	automatique mode

Internal control objects (explicit exchange) of the IODDT of type T_STEPPER_STD

At a Glance

This part presents the internal status objects (explicit exchange) of the IODDT the T_STEPPER_STD type which applies to the TSX CFY11/21 modules. It groups the word type objects, the bits of which have a special significance. These objects are presented in detail below.

Notes

- Generally, the significance of the bits is given for status 1 of the bit. In some cases each bit status is explained.
- Not all the bits are used.

Management of the exchanges: EXCH_STS

The table below shows the meaning of channel exchange check bits EXCH_STS (%MWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Exchange in progress of status parameters (STATUS)	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	Adjustment 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

Exchange report: EXCH_RPT

The table below shows the meaning of report bits EXCH_RPT (%MWr.m.c.1)..

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	status parameters (STATUS) exchange report	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Command parameters exchange report	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Adjustment parameters exchange report	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Configuration fault	%MWr.m.c.1.15

Channel operation status: CH_FLT

The table below shows the meanings of the bits of the CH_FLT (%MWr.m.c.1) status word.

Standard symbol	Type	Access	Meaning	Address
EXT_FLT	BOOL	R	External error (idem HD_ERR bit)	%MWr.m.c.2.0
MOD_FLT	BOOL	R	Internal error: module absent, out of order or in self-test mode	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Hardware or software configuration error	%MWr.m.c.2.5

Standard symbol	Type	Access	Meaning	Address
COM_FLT	BOOL	R	Communication error with the processor	%MWr.m.c.2.6
APP_FLT	BOOL	R	Application fault (invalid configuration) or command fault	%MWr.m.c.2.7
CH_LED_LOW	BOOL	R	Status of the channel light, there are three scenarios: ● bit 8 = bit 9 = 0 channel LED off ● bit 8 = bit 9 = 0 channel LED flashing ● bit 8 = bit 9 = 1 channel LED lit	%MWr.m.c.2.7
CH_LED_HIGH	BOOL	R		%MWr.m.c.2.9

Axis operation status: AX_STS

The table below shows the meanings of the bits of the `AX_STS` (%MWr.m.c.3) status word.

Standard symbol	Type	Access	Meaning	Address
Hardware faults: HD_ERR (%Ir.m.c.4) (groups together the faults below)				
BRAKE_FLT	BOOL	R	Short-circuit fault on the brake output	%MWr.m.c.3.1
DRV_FLT	BOOL	R	Drive fault	%MWr.m.c.3.2
EMG_STP	BOOL	R	Emergency stop fault	%MWr.m.c.3.5
AUX_SUP	BOOL	R	24V supply fault	%MWr.m.c.3.6
Application faults: AX_ERR (%Ir.m.c.5) (groups the faults below)				
SLMAX	BOOL	R	Maximum soft stop overshoot	%MWr.m.c.3.3
SLMIN	BOOL	R	Minimum soft stop overshoot	%MWr.m.c.3.4

Other status data

The table below presents the meanings of the other status data.

Standard symbol	Type	Access	Meaning	Address
N_RUN	INT	R	number of steps in progress	%MWr.m.c.4
G9_COD	INT	R	type of movement in progress	%MWr.m.c.5
G_COD	INT	R	code of instruction in progress	%MWr.m.c.6
CMD_FLT	INT	R	refusal report	%MWr.m.c.7
T_XPOS	DINT	R	target of the position to be reached	%MDr.m.c.8
T_SPEED	DINT	R	speed to be reached	%MDr.m.c.10

NOTE: all this internal status data is updated on execution of the instruction READ_STS.

Objects of the adjustment parameters (explicit exchanges) of the IODDT of the T_STEPPER_STD type

Adjustment parameters

%MWr.m.c.d or **%MDr.m.c.d**

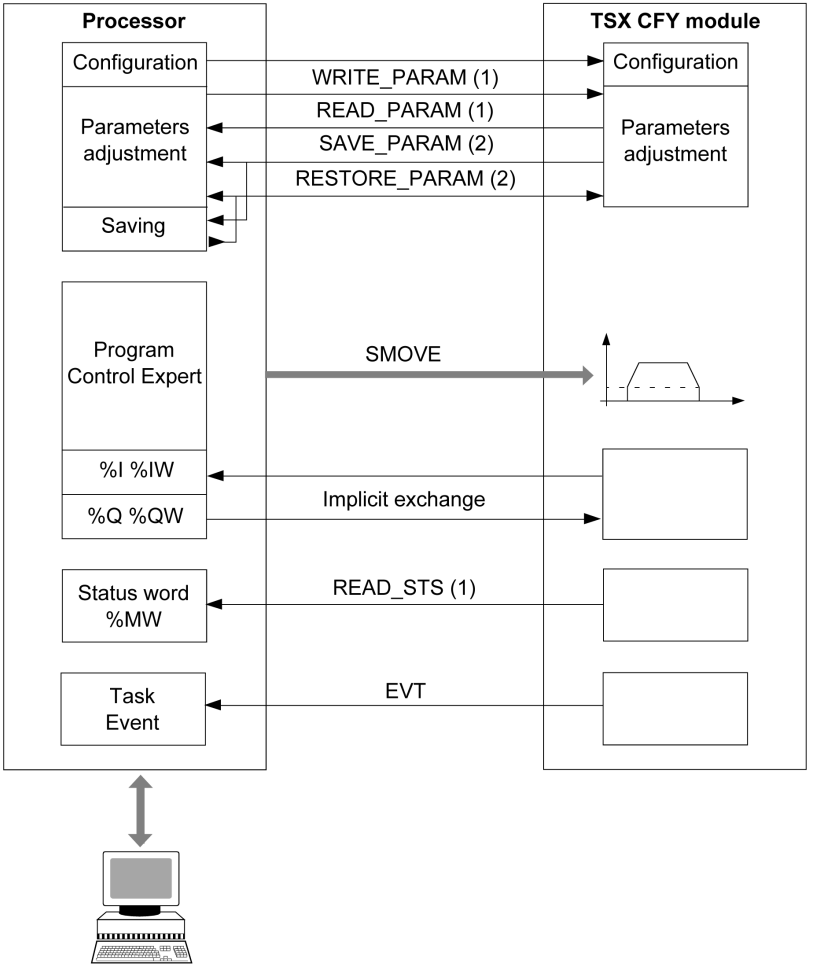
Standard symbol	Type	Access	Description	Address
ACC	DINT	R/W	Acceleration value, dependent on the user-defined unit	%MDr.m.c.12
SL_MAX	DINT	R/W	High soft stop: SLMIN to LMAX for limited axis Modulo in points for infinite axis	%MDr.m.c.14
SL_MIN	DINT	R/W	Lower soft stop: LMIN to SLMAX for limited axis Modulo value in user units for infinite axis	%MDr.m.c.16
SS_FREQ	DINT	R/W	Start and stop speed: 0 to FMAX	%MDr.m.c.18
MAN_SPD	DINT	R/W	Manual mode speed: 10 to VMAX	%MDr.m.c.20
RP_POS	DINT	R/W	Reference point value in manual mode: SLMIN to SLMAX	%MDr.m.c.22
BRK_DLY1	INT	R/W	Shift register to brake deactivation: -1000 to 1000	%MWr.m.c.24
BRK_DLY2	INT	R/W	Shift register to brake activation: -1000 to 1000	%MWr.m.c.25
STOP_DLY	INT	R/W	The duration of the stop stage at the start and stop speed: 0 to 1000	%MWr.m.c.26

NOTE: these adjustment parameters are updated on the execution of a `READ_PARAM` function.

Exchanges between the processor and the axes control module

Diagram showing the exchanges

The different exchanges between the processor and the axis control module are as follows:



(1) Read or write from the adjustment screen or from the application, using explicit exchange instructions.

(2) Saving or restoring using commands **Save parameters** or **Restore parameters** from the Control Expert **Services** menu or using the instructions SAVE_PARAM or RESTORE_PARAM.

List of error codes CMD_FLT

At a Glance

The command refused word CMD_FLT (%MWr.m.c.7) is read by explicit change. The messages are also available in non-encrypted format in the diagnostics dialog box, which can be accessed via the command **DIAG**.

Each byte of the word CMD_FLT is associated to a type of error:

- The most significant byte signals an error in the configuration and adjustment parameters (XX00).
- The least significant byte signals a refusal to execute the movement command (00XX).

For example: CMD_FLT = 0004 (the least significant byte signals an error in the command JOG+).

Mot CMD_FLT

Configuration and adjustment parameters	Motion control
Most significant byte	Least significant byte

Configuration

These errors are signaled in the most significant byte of the word CMD_FLT. The numbers in brackets indicate the hexadecimal value of the code.

Value	Meaning
2 (2)	Reference point configuration error
3 (3)	Event priority configuration error
4 (4)	Maximum frequency configuration error
5 (5)	Maximum acceleration configuration error

Adjustment parameter

These errors are signaled in the most significant byte of the word CMD_FLT. The numbers in brackets indicate the hexadecimal value of the code.

Value	Meaning
7 (07)	Acceleration profile parameter error
8 (08)	Software Hi Limit parameter error
9 (09)	Software Low limit parameter error
10 (0A)	Start and stop frequency parameter error
11 (0B)	Frequency parameter error in manual mode

Value	Meaning
12 (0C)	Reference point value parameter error
13 (0D)	Delay on deactivation of brake parameter error
14 (0E)	Delay on activation of brake parameter error
15 (0F)	Stop stage parameter error
32 (20)	Parameter error, more than one WRITE_PARAM during movement

Movement command refused

These errors are signaled in the least significant byte of the word CMD_FLT. The numbers in brackets indicate the hexadecimal value of the code.

Value	Message
1 (1)	MAN. command error insufficient conditions (Mode, Value, etc.)
2 (2)	MAN. command error MAN. movement in progress
3 (3)	MAN. command error simultaneous commands
4 (4)	MAN. command error JogP
5 (5)	MAN. command error JogM
6 (6)	MAN. command error IncP
7 (7)	MAN. command error IncM
8 (8)	MAN. command error IncP parameter
9 (9)	MAN. command error IncM parameter
10 (0A)	MAN. command error manual RP
11 (0B)	MAN. command error forced RP
12 (0C)	AUTO command error insufficient conditions (parameters)
13 (0D)	AUTO command error AUTO movement in progress
14 (0E)	SMOVE command error insufficient conditions (Mode)
15 (0F)	SMOVE G01 command error (1)
16 (10)	SMOVE G09 command error (1)
17 (11)	SMOVE G10 command error (1)
18 (12)	SMOVE G11 command error (1)
21 (15)	SMOVE G14 command error (1)
22 (16)	SMOVE G05 command error (1)
23 (17)	SMOVE G07 command error (1)
24 (18)	SMOVE G62 command error (1)
25 (19)	SMOVE execution command error
26 (1A)	AUTO command error movement in progress

Value	Message
27 (1B)	AUTO Command error, stack full
48 (30)	DIRDRIVE command error, insufficient command
49 (31)	DIRDRIVE command error, with mode change in progress
50 (32)	DIRDRIVE command error, with moving axis
51 (33)	DIRDRIVE command error, with stopped axis
52 (34)	DIRDRIVE command error, with disabled axis
53 (35)	DIRDRIVE command error, with blocking fault
54 (36)	DIRDRIVE command error, with frequency less than SS_FREQ
55 (37)	DIRDRIVE command error, with frequency more than FMAX
56 (38)	DIRDRIVE command error, with axis at the end of run + limit
57 (39)	DIRDRIVE command error, with axis at the end of run - limit
58 (3A)	DIRDRIVE command error, with axis outside the end of run + limit
59 (3B)	DIRDRIVE command error, with axis outside the end of run - limit
60 (3C)	DIRDRIVE command error, with axis outside the Software Hi Limit
61 (3D)	DIRDRIVE command error, with axis outside the Software Low Limit
96 (60)	MAN. JogP command error in Software Hi Limit
97 (61)	MAN. JogP command error, with stopped axis
101 (65)	MAN. JogP command error, JogM movement in progress
102 (66)	MAN. JogP command error at the end of run + limit
103 (67)	MAN. JogP command error position above the end of run + limit
108 (6C)	MAN. JogP command error blocking fault other than software limit
109 (6D)	MAN. JogP command error unacknowledged software limit blocking fault
110 (6E)	MAN. JogP command error, disabled axis
113 (71)	MAN. JogM command error, stopped axis
116 (74)	MAN. JogM command error, JogP movement in progress
118 (76)	MAN. JogM command error at the end of run - limit
119 (77)	MAN. JogM command error position above the end of run - limit
124 (7C)	MAN. JogM command error blocking fault other than software limit
125 (7D)	MAN. JogM command error unacknowledged software limit blocking fault
126 (7E)	MAN. JogM command error, disabled axis
127 (7F)	MAN. JogM command error in Software Lo Limit
130 (82)	MAN. IncP command error position less than the Software Lo Limit
131 (83)	MAN. IncP command error position more than the Software Hi Limit
132 (84)	MAN. IncP command error, JogP movement in progress

Value	Message
133 (85)	MAN. IncP command error, JogM movement in progress
134 (86)	MAN. IncP command error at the end of run - limit
135 (87)	MAN. IncP command error position above the end of run + limit
136 (88)	MAN. IncP command error, axis not referenced
137 (89)	MAN. IncP command error causes movement of the Software Lo Limit
138 (8A)	MAN. Inc P command error stop condition
141 (8D)	MAN. IncP command error, disabled axis
146 (92)	MAN. IncM command error position less than the Software Lo Limit
147 (93)	MAN. IncM command error position above the Software Hi Limit
148 (94)	MAN. IncM command error, JogP movement in progress
149 (95)	MAN. IncM command error, JogM movement in progress
150 (96)	MAN. IncM command error at the end of run - limit
151 (97)	MAN. IncM command error position above the end of run + limit
152 (98)	MAN. IncM command error, axis not referenced
154 (9A)	MAN. command error IncM stop condition
155 (9B)	MAN. IncM command error causes movement of the Software Hi Limit
158 (9E)	MAN. IncM command error, disabled axis
164 (A4)	MAN. manual RP manual IncP command error, JogP movement in progress
165 (A5)	MAN. manual PO manual IncM command error, JogM movement in progress
170 (AA)	MAN. error command manual PO stop condition
174 (AE)	MAN. command error, manual PO disabled axis
178 (B2)	MAN. command error forced RP position less than the Software Lo Limit
179 (B3)	MAN. command error forced RP position above the Software Hi Limit
180 (B4)	MAN. command error, forced RP JogP movement in progress
181 (B5)	MAN. command error, forced RP JogM movement in progress
189 (BD)	MAN. command error forced RP with unacknowledged software limit error
190 (BE)	MAN. command error, forced PO disabled axis

(1) Indicates that one of the parameters of the SMOVE function does not conform. Examples: invalid movement type code, position outside the software limits, speed greater than FMAX, etc.

Details of the Language Objects of the T_GEN_MOD-Type IODDT

Introduction

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	Error detected while 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, inoperable module	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Channel error detected	%MWr.m.MOD.2.1
BLK	BOOL	R	Terminal block error	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Hardware or software configuration mismatch	%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	Module is unserviceable (Fipio extension only)	%MWr.m.MOD.2.8
CH_FLT_EXT	BOOL	R	Channel error detected (Fipio extension only)	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Terminal block error detected (Fipio extension only)	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Hardware or software configuration mismatch (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

Index

W