Premium and Atrium using EcoStruxure™ Control Expert

Axis Control Modules for Servomotors User Manual

(Original Document)

12/2018

35006220.09

www.schneider-electric.com

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.

About the Book

At a Glance

Document Scope

This manual describes the software installation of axis control tasks for Premium and Atrium PLCs from 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
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)
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)

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
Control of Servo Drive Axes in Premium PLCs

Subject of this Part

This part gives a concise introduction to the servo drive axis control range and describes the methodology for setting up the independent axes or interpolated axes.

What Is in This Part?

This part contains the following chapters:

Chapter 1
General Introduction

Subject of this Section

This section gives a concise introduction to the servo drive axis control range for Premium PLCs, as well as the functions provided by the TSX CAY modules.

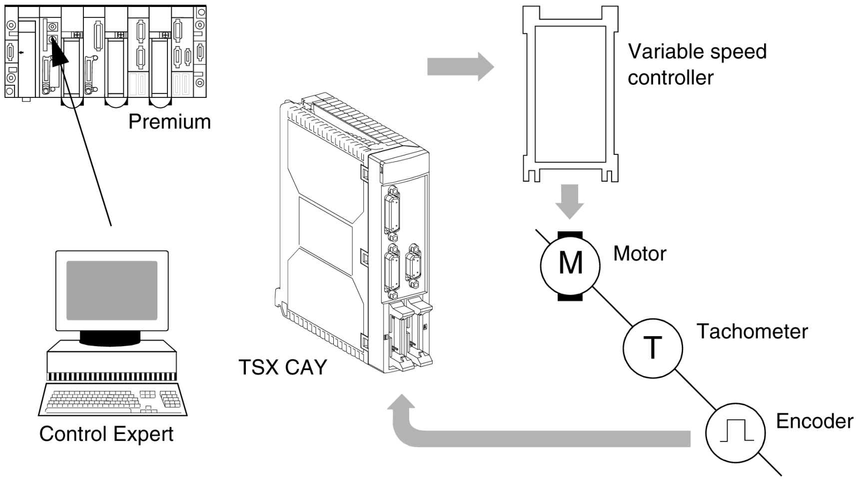
What Is in This Chapter?

This chapter contains the following topics:

Introduction to the Motor Operator Axis Control Range

At a Glance

Below is an overview of a motor operator axis control architecture:



Axis Control Range

The motor operator axis control range for Premium PLCs is made up of 5 modules: TSX CAY 21 / 41 / 22 / 42 / 33:

- TSX CAY 21 (2 axes with limited run-time),
- TSX CAY 41 (4 axes with limited run-time),
- TSX CAY 22 (2 axes with unlimited run-time),
- TSX CAY 42 (4 axes with unlimited run-time),
- TSX CAY 33 (3 axes with limited or interpolated run-time),

The Control Expert software includes as standard the application-specific movement function for programming these axis control modules.

The basic movements are controlled via the machine's principal sequential command program, but are made and checked by the TSX CAY modules, which ensure the position of the moving part is controlled.

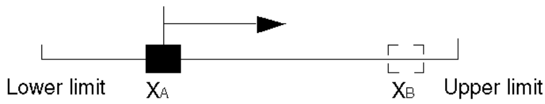
The position of each channel is measured by either an incremental encoder or by an absolute encoder. The analog output is used to control a variable speed controller.

TSX Modules CAY 21 / 41

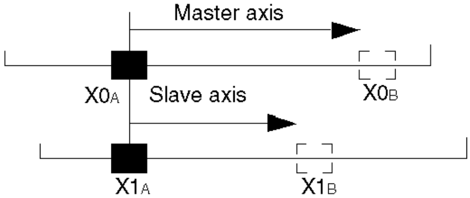
These modules (2 and 4 axes respectively) are used to control the movement of independent axes on machines with limited run-time. They are also used for master / slave applications.

At a Glance

The following diagram illustrates the movement of an axis between XA and XB on a machine with limited run-time (lower and upper limits):



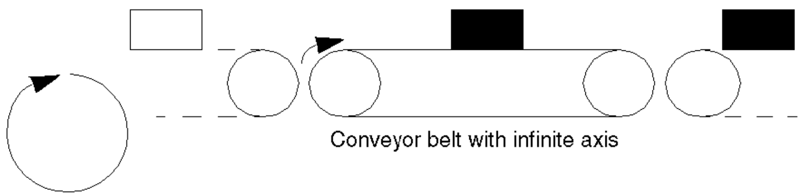
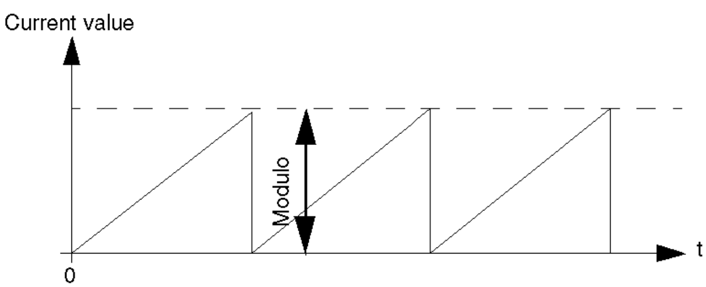
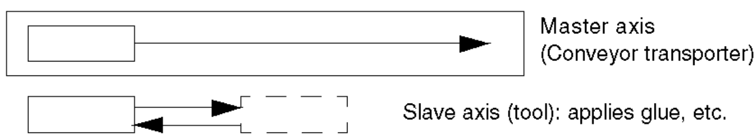
The following diagram illustrates the movements of a master axis between X0A and X0B and those of a slave axis between X1A and X1B:



Master position / slave position = Constant

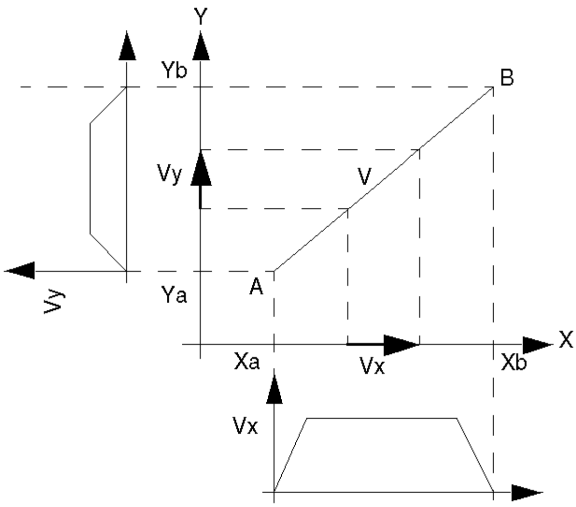
TSX Modules CAY 22 / 42

These modules (2 and 4 axes respectively) are used to control the movement of independent axes on machines with infinite run-time (usually rotational axes or similar). This type of application creates a measurement variation area called the "Modulo". These modules are also used to perform master-slave object tracking applications.

At a Glance
The following diagram illustrates the movement of an axis on a machine with unlimited run-time: 
The following diagram provides an outline of the Modulo: 
The following diagram illustrates a master-slave object tracking application: 

TSX Module CAY 33

This module (3 axes) is used to control the movement of axes, which are linearly interpolated on Cartesian machines with limited run-time. It is used to follow trajectories, either on a plane (2 axes), or in space (3 axes).

At a Glance
The following diagram provides an outline of an interpolation of 2 axes: 

The TSX CAY 33 module can be used in the following configurations:

- 3 interpolated axes,
- 2 interpolated axes and one independent axis,
- 3 independent axes (in the event of use without interpolation).

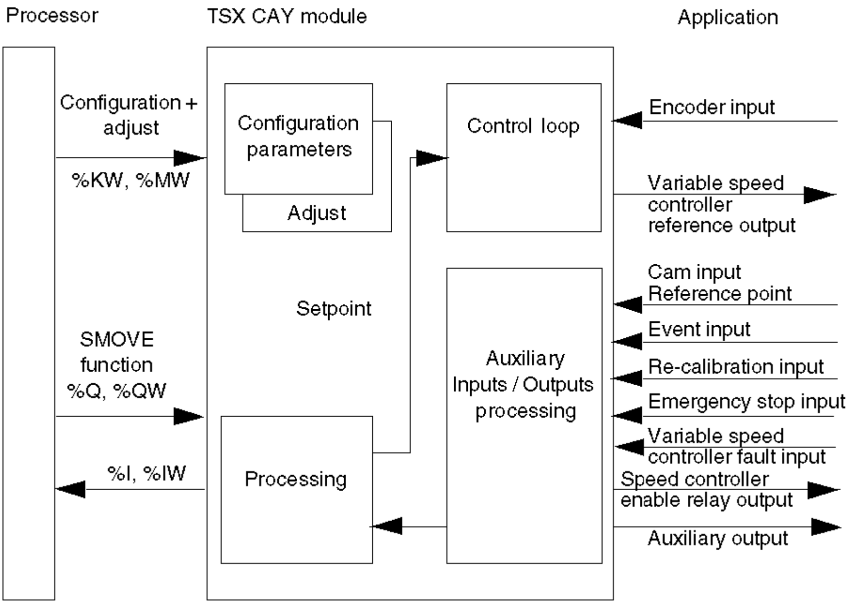
This module does not provide the necessary circular interpolation to process bypass applications.

Functions Provided by the Axis Control Modules

General

The axis control modules provide application inputs and outputs for each of the axes, which are used to implement the different functions.

The following diagram shows the inputs/outputs associated with one channel:



Application Inputs/Outputs

For each of the axes, the axis control modules offer:

In inputs:

- One input to acquire position measurements:
  - either via type RS 485 incremental encoder (maximum frequency 500 KHz without multiplication or 1 MHz multiplied by 4). The module provides the choice of either multiplying by one or by four,
  - or by absolute encoder, for up to 25 bits of data, with serial link and transmission according to SSI protocol (200 KHz clock frequency).
- One input serving as a reference point cam (if an incremental encoder is selected).
- One event input.
- One emergency stop input.
- One re-calibration on-the-fly input.
- One drive fault input.

In outputs:

- One 10 V analog output isolated from the logic part of the module. This 13 bit + sign resolution output is used to control a variable speed controller, associated with a continuous current motor, on autosynchronous or asynchronous auto-pilot.
- A relay output to enable the variable speed controller.
- A static auxiliary output.

Programming a Movement

In Control Expert language, each independent axis movement is described by a SMOVE movement command function. Movements of interpolated axes are described by an XMOVE command (linear interpolation of TSX CAY 33 module). From this SMOVE or XMOVE command and from the position of the moving part, module TSX CAY calculates the position/speed setpoint.

Configuration and Adjustment Parameters

These parameters are used to define the usage characteristics, limits, resolution, control dimensions, etc.

Position Control Loop

The loop controller is of proportional type with feed forward, so as to reduce following errors.

For each axis, the user can choose between 3 types of acceleration law: rectangular, trapezoid or triangular.

TSX CAY Module Functions

The functions offered by the axis control modules are as follows:

- **Position slave movement of another axis**: one or more axes can be controlled by a master axis. The movements of the slave axis thus follow all the movements of the master axis.
- **Slave movement of a periodic setpoint**: periodically, the position setpoint can be sent directly by the PLC processor.
- **Re-calibration on-the-fly**: this function (used with an incremental encoder) is used to monitor the position of the moving part and to recalibrate the measurement when the re-calibration input is activated. This function can be used for movements with slip, so that the position measurement is periodically recalibrated.
- **Event processing**: events detected by the module can be used to activate an event task in the sequential program.
- **Feed hold**: this function is used to momentarily stop a movement in progress (e.g. to synchronize the axes).
- **Deferred pause**: this function is used to momentarily stop a machine cycle without causing disturbance to it.
- **Step by step mode**: this mode is used to carry out a sequence of movements stopping after each elementary instruction.
- **Movement monitoring**: this function is used to detect any abnormal process operations (such as a faulty encoder), which do not change the position value even when the moving part moves.
- **Infinite axis function**: (only available on the TSX CAY •2 modules), this function is used to process unlimited-type axes ("conveyor belt" applications). The axis defines a movement, which is always performed in the same direction.
  It is also used for two other types of application:
  - the drift function,
  - synchronized movement of infinite axes.
- **Linear interpolation function**: (only available on the TSX CAY 33 module). This function is used to associate 2 or 3 axes with the application-specific Interpolation function. It is used to access certain functional characteristics of numerical command machines.

NOTE: The 3 physical axes (X, Y, Z) use the zones 0, 1 and 2. Zone 3 is dedicated to the linear interpolation function.

Summary of TSX CAY Module Functions

The TSX CAY axis control modules perform the following functions:

Modules	CAY 21	CAY 41	CAY 22	CAY 42	CAY 33
2/3 axes interpolation	No	No	No	No	Yes
Limited axes	Yes	Yes	Yes	Yes	Yes
Infinite axes	No	No	Yes	Yes	Yes
Slave axes (static ratio)	Yes	Yes	No	No	No
Slave axes (dynamic ratio)	No	No	Yes	Yes	No
Integral gain (correction of offsets in the kinematic string)	No	No	Yes	Yes	Yes

Chapter 2
Introduction to the TSX CAY Modules

Subject of this Chapter

This chapter provides an overview of the different TSX CAY axis command modules.

What Is in This Chapter?

This chapter contains the following topics:

General

Introduction

The axis command and controlled placement offer for Premium PLCs is designed for machines, which require a simultaneous performing movement command and a sequential command by programmable controller.

The following modules: **TSX CAY 21** (2 axes) and **TSX CAY 41** (4 axes) make controlled placement possible on independent, linear and limited axes.

The modules **TSX CAY 22** (2 axes) and **TSX CAY 42** (4 axes) make controlled placement possible on independent, circular and infinite axes.

The module **TSX CAY 33** (3 axes) makes a placement on 2 or 3 synchronized axes (linear interpolation) possible.

Terminology

- the term TSX CAY covers everything on the axis command offer,
- the reference TSX CAY 2 regroups the TSX CAY 21 and 22 modules,
- the reference TSX CAY 4• corresponds to the TSX CAY 41 and 42 modules.

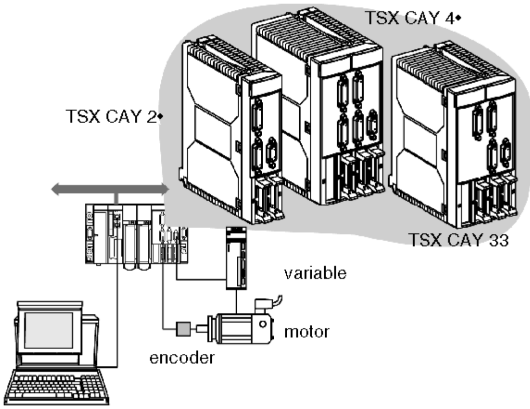
These modules in standard format (TSX CAY 2•) or double format (TSX CAY 4• and TSX CAY 33) can be installed in all the available slots of a PLC configuration (TSX, or PCX).

To ensure position measurement, an encoder (which may be a different type) is wired onto each of the channels:

- RS 422/485 incremental encoder,
- 5 V Totem pole incremental encoder,
- SSI serial absolute encoder,
- parallel output absolute encoder (with ABE-7CPA11 interface).

Illustration

This diagram illustrates different types of TSX CAY modules:



Physical Description

Illustration

This diagram illustrates different TSX CAY modules:

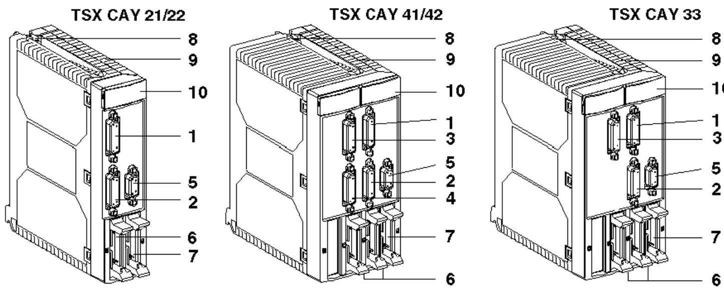


Table of Numbers

The following table describes the above diagrams using numbers:

Number	Description
1	15-pin SUB-D connector for connecting an axis 0 encoder.
2	15-pin SUB-D connector for connecting an axis 1 encoder.
3	15-pin SUB-D connector for connecting an axis 2 encoder.
4	15-pin SUB-D connector for connecting an axis 3 encoder.
5	9-pin SUB-D connector for connecting speed references.
6	HE10 connector(s) for connecting: ● auxiliary inputs: ❍ cam reference point, ❍ emergency stop, ❍ recalibration, ● of auxiliary outputs, ● of external supplies (encoders and sensors).
7	HE10 connector for connecting variable controller inputs/outputs.
8	Screw for fixing module in place.
9	Rigid body, which functions as the module captor in the slot.
10	Module diagnostic LEDs: ● module level diagnostics: ❍ green LED RUN: indicates the operating mode of the module, ❍ red LED ERR: indicates an internal error, ❍ red LED I/O: indicates an external error or application fault, ● module channel level diagnostics: ❍ CHx green LEDs: indicates of channel diagnostics.

Chapter 3
Functions

Subject of this Chapter

This Chapter introduces the various functions of the TSX CAY modules.

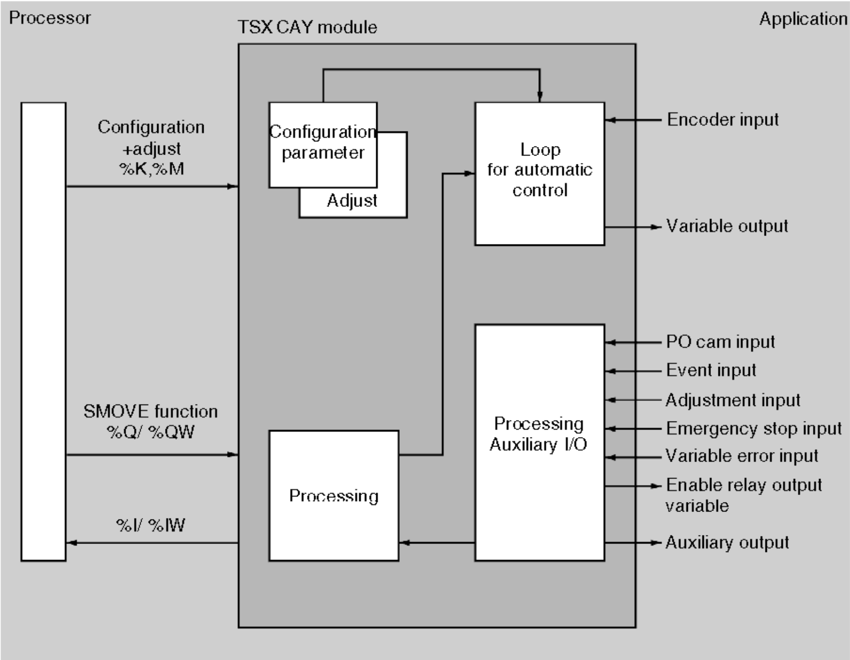
What Is in This Chapter?

This chapter contains the following topics:

Circuit Diagram of an Axis Command

Illustration

Process diagram:



Functions Provided by the Axis Command Modules

The axis command modules provide the following functions for each axis:

- Inputs:
  - one input for the acquisition of position measurements:
    RS 485 incremental encoder or 5 V totem pole, 16 to 25-data bit SSI serial absolute encoder,
  - one machine reference point input,
  - one event input,
  - one variable default input,
  - one recalibration input,
  - one emergency stop input.
- Outputs:
  - one +/- 10 V, +sign 13-bit resolution analog output, for the speed controller command,
  - one relay output for validating the controller,
  - one auxiliary static output.

Command Processing

At a Glance

Each movement, controlled from the PLC sequential program, is described by a SMOVE movement command function in the Control Expert language. From this SMOVE command, the TSX CAY modules work out a position/speed trajectory.

The Control Expert screens make it possible to easily achieve the configuration, adjustment, and setting of the axes.

Axis Configuration

The configuration screen enables the required parameters to be entered, in order to adapt the operation of the module to the characteristics of the machine. These are: the encoder type, position limits, maximum speed, etc. These parameters cannot be modified by a program. There is no default configuration.

Axis Adjustment

The parameters offered by the adjustment screen are linked to axis operation. The parameters are adjusted when on or offline.

The operating parameters are:

- corrected resolution,
- movement control: errors of following, adjustment, overspeed, etc.,
- stop control: delay, speed, debug window,
- position loop: position gain, speed anticipation coefficient, offset,
- command: soft stops, acceleration, acceleration profile,
- manual mode parameters: speed, reference point value etc.

NOTE: These parameters can be changed by program.

Debugging

The debug screen can only be accessed in online mode. This makes it possible to control and observe the performance of the axis.

Information and commands differ according to the operation mode chosen:

- automatic mode,
- manual mode,
- loop control disabled mode,
- measurement mode (off).

The top part of the screen indicates the operating state and diagnostics of the module. The lower part accesses the commands and indications on the operation of movement, inputs/outputs, errors, etc.

Compatibility of the Absolute Encoders with the TSX CAY Modules

General

All absolute SSI encoders, 16 ≤ Number of data bits ≤ 25, Gray or binary code are compatible with the TSX CAY modules. For example:

- IVO trademark
  - GM 400 0 10 11 01
    24 Volts, Gray code, 0 header bits, 25 data bits, 0 status bits, without parity,
  - GM 401 1 30 R20 00
    24 Volts, Gray code, 0 header bits, 25 data bits, 1 status bit, with even parity.
- Hengstler trademark
  - RA58-M/1212
    24 Volts, Gray code, 0 header bits, 24 data bits, 1 status bit, without parity.
- Stegmann trademark
  - AG 661 01
    24 Volts, Gray code, 0 header bits, 25 data bits, 0 status bits, without parity.
- IDEACOD trademark
  - SHM506S 428R / 4096 / 8192 / 26
    11-30 Volts, Gray code, 0 header bits, 25 data bits, 0 status bits, without parity,

Chapter 4
Set-up Methodology

Subject of this Section

This section describes the overall methodology for setting up an independent axis movement or interpolated axes movements.

What Is in This Chapter?

This chapter contains the following topics:

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 37)* 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.

Interpolation Implementation Method

At a Glance

The method for implementing interpolation follows the general implementation *(see page 36)* principle, with additional phases for interpolation.

The following order of installation phases is recommended, but the order of some phases may be altered (for example, start with the configuration phase).

In order to fully understand these different phases, please also refer to the introductory example *(see page 41)* which accompanies this table.

Installation Principle with a Processor

The following table presents the different installation phases with the processor:

Phase	Description	Mode
Declaration of variables	Declaration of IODDT type variables for axis command module and application variables.	Local (1)
Programming	Programming the application. Programming movements: SMOVE : independent axis/es, XMOVE : interpolating channel.	Local (1)
Configuration	Declaration of the module.	Local
	Individual axis configuration.	
	Interpolating channel configuration.	
	Input of configuration parameters.	
Association	Association of IODDTs with the configured module (variable editor).	Local (1)
Generation	Generation (analyze and edit links) of the application.	Local
Transfer	Transferring the application into the PLC.	Online
Adjust/Debug	Adjusting independent axis parameters.	Online
	Adjusting interpolation parameters.	
	Debugging independent axes.	
	Debug of interpolation channel.	
	Debugging the application using the debug screens and animation tables.	
	Modifying the program.	
Documentation	Building documentation file and printing the different data concerning the application.	Online (1)

Phase	Description	Mode
Operation/Diagnostics	View of the different data needed to carry out the application.	Online
	Diagnostics of the application and modules.	
Key		
(1)	These different phases may also be performed in the other mode.	

Chapter 5
Introductory Example

Subject of this Section

This section provides an example of how to set up a TSX CAY axis control application. This example is instructive and covers all the phases necessary for setting up independent or interpolated axes.

What Is in This Chapter?

This chapter contains the following topics:

Description of the Example

Introduction

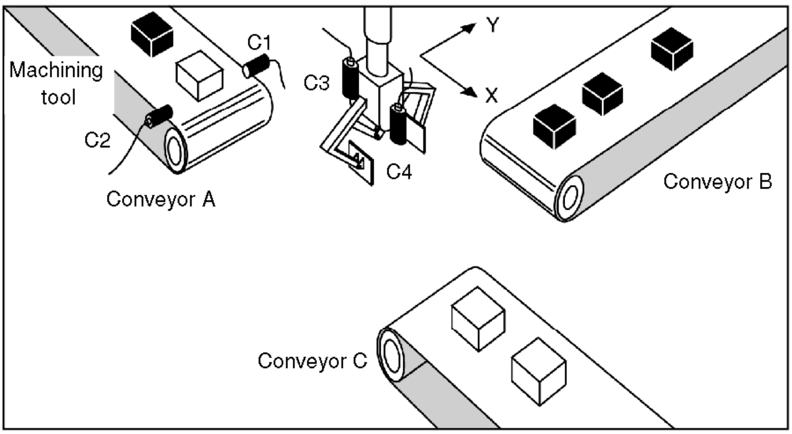
The following example covers all the phases for setting up a TSX CAY axes control application. It complements the set-up methodologies.

Transfer Device

A transfer device evacuates the parts during machining output. This device consists of a grab which can move through the air along a plane (X, Y axes) parallel to the ground.

As soon as a part appears on evacuation conveyor A, the grab automatically retrieves it and deposits it on conveyor B or C depending on the type of part. The grab then returns to the waiting position until a new machined part needing to be picked up is detected.

The following figure illustrates this transfer device:



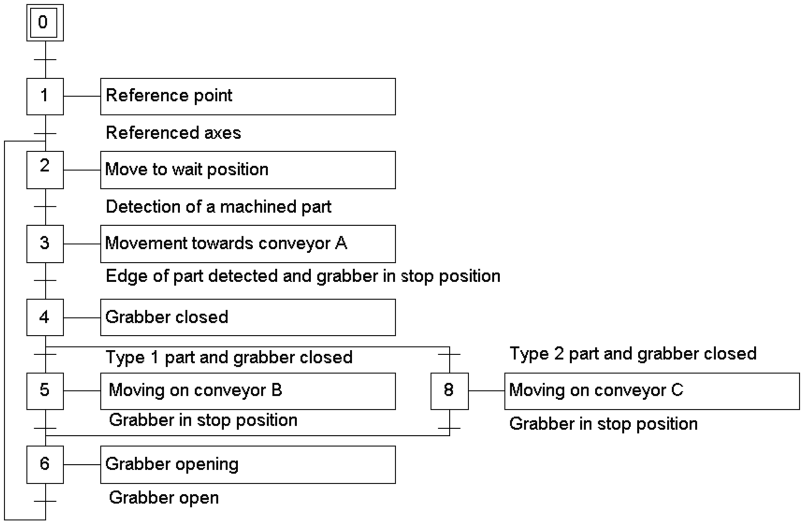
Inputs/Outputs

The inputs / outputs are as follows:

I/O	Description
C1	Machined part detection cell
C2	Part type identification sensor.
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	Grab Open / Close control.

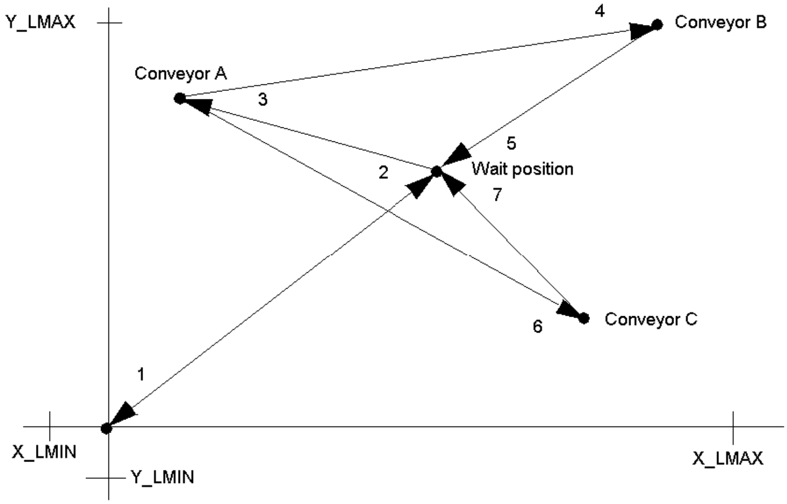
Application Grafcet

The application Grafcet is as follows:



Trajectory Description

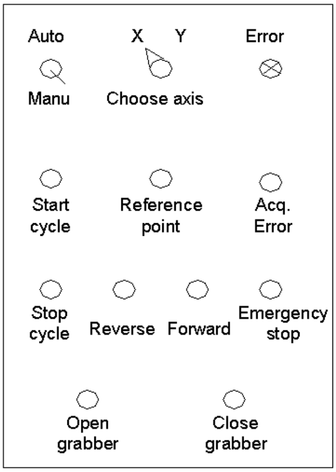
The following diagram shows the trajectory of the grab:



1 Reference point at speed Vp0,
2 Movement at speed Vret to waiting position (Xatt, Yatt) with stop,
3 Movement towards conveyor A (XA, YA) at speed VA, until the machined part is detected,
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.

Operator Dialogue Front Panel

The following controls grouped together on a front panel, are used to control the moving part manually when the installation is faulty. The controls and the LEDs are managed by one discrete input module and one discrete output module.



Auto / Manu: Operating mode selection switch.
Start cycle: Executes the automatic cycle.
Stop cycle: Stops the automatic cycle.
X / Y axis selection: For selection of the axis to be controlled in manual mode.
Reference point: Manual reference point on the selected axis.
Forward / Reverse: Control for manual movement of the selected axis in a positive or negative direction.
Error: LED for all hardware and application errors.
Acq. Error: 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 and Methodology

Prerequisites

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

- Control Expert software is installed,
- The hardware has been installed: the modules, variable speed controllers and encoders controlling the 2 axes are connected.

Set-up Using Independent Axes

In order to set up this application, the following operations must be carried out:

- Enter and declare project variables,
- Program the project,
- Enter the configuration parameters of the axes,
- Adjust the axis control parameters,
- Debug the program,
- Save the project.

Set-up Using Interpolated Axes

With a TSX CAY 33 module, it is possible to use 2 interpolated axes to control the grab's movement through the plane (X,Y). In order to set up this application the same operations must be carried out as for 2 independent axes, with the addition of some operations specific to interpolation:

- Configure the interpolator (channel 3), in addition to the independent axes,
- Enter the symbols linked with interpolation,
- Program the application by using the XMOVE function (not SMOVE)

Declaration of Variables Used in the Example

Access to the Declaration of Values

Access to the entry of variables takes place by clicking the **Variables** directory of the project browser.

Internal Variables

The following internal variables are declared located:

Variable	Address	Comment
Cycle	%M0	Condition of the machine in operating mode
X_attente	%MD50	Waiting position (X axis)
y_attente	%MD52	Waiting position (Y axis)
X_b	%MD54	Position of Conveyor B (X axis)
y_b	%MD56	Position of Conveyor B (Y axis)
X_c	%MD58	Position of Conveyor C (X axis)
Y_c	%MD60	Position of Conveyor C (Y axis)

Variables Associated with the Discrete Input Module

The discrete input module is positioned in slot 3 of Rack 0. The variables associated are as follows:

Variable	Address	Comment
Capteur_1	%I0.3.0	Machined part detection cell
Capteur_2	%I0.3.1	Part type identification sensor (0 = type 2, 1 = type 1)
Capteur_3	%I0.3.2	Grab open / grab closed detection sensor
Auto_man	%I0.3.3	Mode selection switch (0 = Auto, 1 = Manual)
Depart_cycle	%I0.3.4	Automatic cycle start button
Arret_cycle	%I0.3.5	Automatic cycle stop button
Selection_x_y	I%I0.3.6	Selection of the axis to be controlled in manual mode (1 = X, 0 = Y)
Po_man	%I0.3.7	Manual reference point
Avant	%I0.3.8	Moving the moving part in a positive direction
Arriere	%I0.3.9	Moving the moving part in a negative direction
Acq_defauts	%I0.3.10	Fault acknowledgement
Arret_urgence	%I0.3.12	Emergency stop
Ouv_pince	%I0.3.13	Grab opening button
Ferm_pince	%I0.3.14	Grab closing button

Variables Associated with the Discrete Output Module

The discrete output module is positioned in slot 4 of Rack 0. The variables associated with it are as follows:

Variable	Address	Comment
Pince	%Q0.40.0	Grab open/close control (0 = Open, 1 = Close)
Defaut	%Q0.4.1	Fault signaling

IODDT of the Axis Control Module:

The axes control module is positioned in slot 3 of Rack 1. IODDT type variables that are associated with it are declared as type`T_AXIS_STD`, and there are 2 of them:

IODDT	Address	Symbol	Address
Axe_x	%CH1.03.0	Axe_y	%CH1.03.1

IODDT Connected with Interpolation

If you use a TSX CAY 33 axis control module with interpolated axes in your application, it will be of the type `T_INTERPO_STD`:

Variable	Address	Comment
interpo	%CH0.03.2	Third channel of the TSX CAY 33 module

Internal Constants

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

Variable	Address	Value	Comment
Vitesse_p_o_x	%KD0	1000	Reference point speed following the X axis
Vitesse_x_attente	%KD4	1200	Speed towards X axis waiting position
Vitesse_y_attente	%KD6	1200	Speed towards Y axis waiting position
Vitesse_pos_a_x	%KD8	1500	Speed towards X axis conveyor A position
Vitesse_pos_a_y	%KD10	1500	Speed towards Y axis conveyor A position
Vitesse_pos_b_x	%KD12	1200	Speed towards X axis conveyor B position
Vitesse_pos_b_y	%KD14	1200	Speed towards Y axis conveyor B position
Vitesse_pos_c_x	%KD16	1800	Speed towards X axis conveyor C position
Vitesse_pos_c_y	%KD18	1800	Speed towards Y axis conveyor C position

Where there are 2 interpolated axes, the symbols and values of these constants are as follows:

Variable	Address	Value	Comment
Vitesse_p_o_x	%KD0	1000	Reference point speed following the X axis
Vitesse_attente	%KD4	1200	Speed towards waiting position
Vitesse_pos_a	%KD8	1500	Speed towards conveyor A position
Vitesse_pos_b	%KD12	1200	Speed towards conveyor B position
Vitesse_pos_c	%KD16	1800	Speed towards conveyor C position

Programming the Preliminary Processing

At a Glance

As implied in its name, preliminary processing is processing that is carried out first. A section is created at the beginning of the project to manage operating modes:

On a blocking fault:

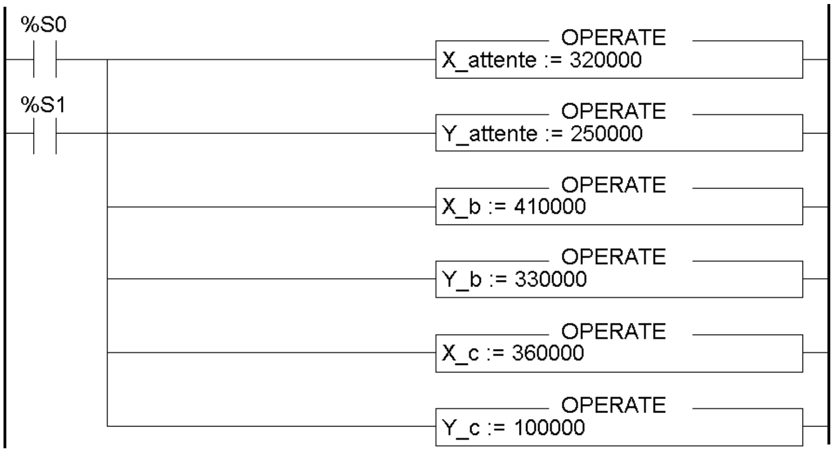
- The chart is frozen,
- The moving part can then be controlled in manual mode, and the fault can be corrected and acknowledged from the front panel,
- The chart is reinitialized when the fault has been corrected and acknowledged.

When switching to manual mode:

- The chart is frozen,
- The chart is reinitialized when automatic mode is reselected.

Program in Ladder Language

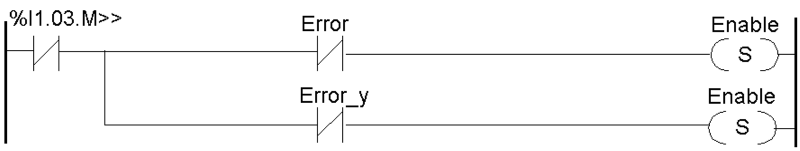
Position initialization



Start cycle

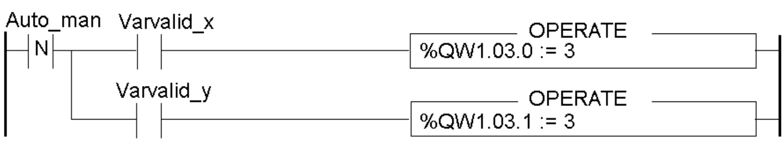
%I1.03.M>> = %I1.03.MOD.ERR
Start_>> = Start_cycle
Stop_>> = Stop_cycle

Enable speed controller

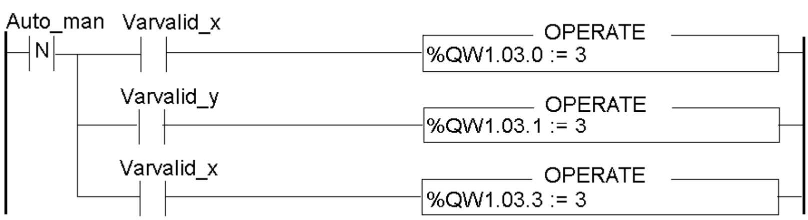


%I1.03.M>> = %I1.03.MOD.ERR

Automatic mode selection (when there are 2 independent axes)



Automatic mode selection (when there are 2 interpolated axes)



Manual mode selection

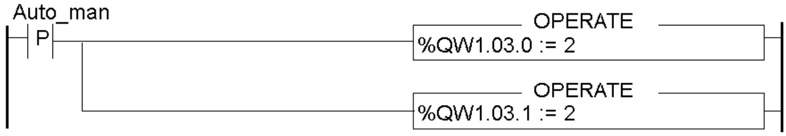
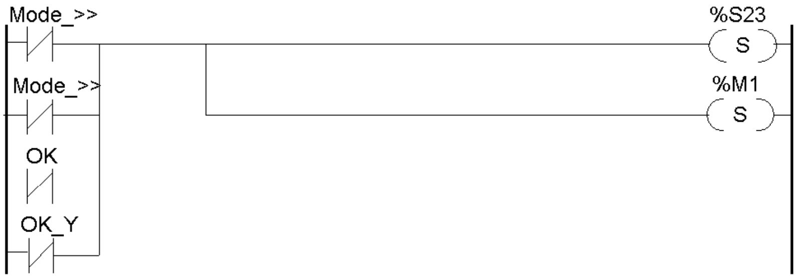
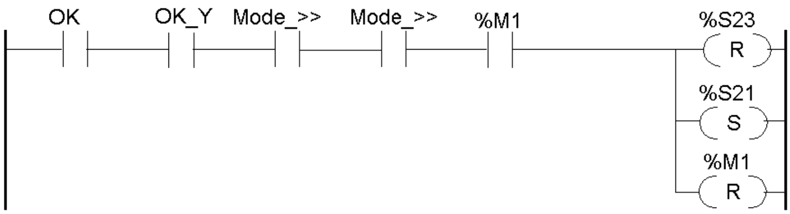


Chart freezing on a fault or when switching to manual mode



Mode_>> = Mode_auto
Mode_>> = Mode_auto_y
%M1 = Grafcet frozen

Chart reset



Mode_>> = Mode_auto
Mode_>> = Mode_auto_y
%M1 = Grafcet frozen

Fault signalling

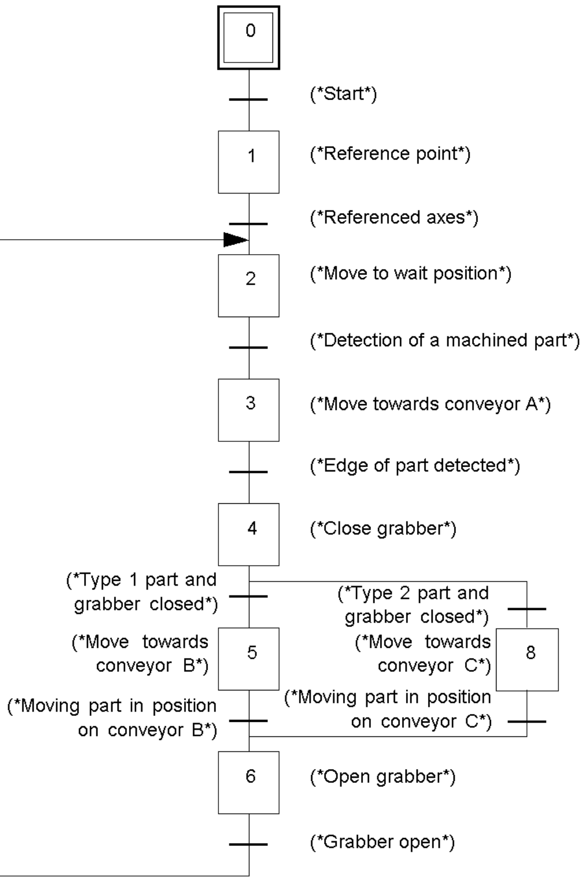
Programming the SFC

At a Glance

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

Sequential Processing

Chart description:



Transition Programming

At a Glance

The transitions associated with steps 2, 3, 5 and 8 are different depending on whether the SMOVE (for independent axes) or XMOVE (for interpolated axes) command is used.

Step 0 -> 1

! (*Channel X not faulty, open grab, Auto_man switch set to Auto, start cycle, channel Y not faulty and automatic mode active*)

```
NOT Error AND NOT Capteur_3 AND NOT Auto_man AND Cycle AND NOT Error_y
AND Mode_Auto
```

Step 1 -> 2

! (*Test: X axis done and calibrated*)

```
Done AND Calib
```

Step 2 -> 3

For 2 independent axes

! (*Moving part in waiting position and part detected on conveyor A*)

```
Capteur_1 AND Cycle AND Axe_x.Next AND Axe_y.Next
```

For 2 interpolated axes

! (*Moving part in waiting position and part detected on conveyor A*)

```
Capteur_1 AND Cycle AND Next_INT
```

Step 3 -> 4

For 2 independent axes

! (*Moving part in detected part retrieval position on conveyor A*)

```
Axe_x.At_point AND Axe_x.Next AND Axe_y.Next AND Axe_y.At_point
```

For 2 interpolated axes

! (*Moving part in detected part retrieval position on conveyor A*)

```
interpo.At_point AND interpo.Next
```

Step 4 -> 5

! (*Type 1 part and closed grab*)

```
Capteur_2 AND Capteur_3
```

Step 4 -> 8

! (*Type 2 part and closed grab*)

```
NOT Capteur_2 AND Capteur_3
```

Step 5 -> 6

For 2 independent axes

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

```
Axe_x.At_point AND Axe_x.Next AND Axe_y.Next AND Axe_y.At_point
```

For 2 interpolated axes

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

```
interpo.At_point AND interpo.Next
```

Step 8 -> 6

For 2 independent axes

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

```
Axe_x.At_point AND Axe_x.Next AND Axe_y.Next AND Axe_y.At_point
```

For 2 interpolated axes

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

```
interpo.At_point AND interpo.Next
```

Step 6 -> 2

! (*Grabber open*)

```
NOT Capteur_3 AND Cycle
```

Action Programming

At a Glance

To move the independent axes X and Y (step 2, 3, 5 and 8), the SMOVE command must be used and applied to each of the X and Y axes. To simultaneously move axes X and Y (for interpolated axes) the XMOVE command must be used (associated with channel 3).

Step 1: How to Activate It

! (*Reference point following the X axis*)

```
SMOVE (Axe_x, 1, 90, 14, 0, Vitesse_p_o_x, 16#0000);
```

Step 2: How to Activate It

For 2 independent axes

! (*Movement into waiting position (Xatt, Yatt)*

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

For 2 interpolated axes

(*Movement into waiting position (Xatt, Yatt)*

```
XMOVE (INTERPO, 2, 90, 9, 0, X_attente, Y_attente, Vitesse_attente,
16#0000);
```

Step 3: How to Activate It

For 2 independent axes

! (*Movement towards conveyor A)*

```
SMOVE (Axe_x, 3, 90, 10, 150000, Vitesse_pos_a_x, 16#0000);
SMOVE (Axe_y, 3, 90, 10, 280000, Vitesse_pos_a_y, 16#0000);
```

For 2 interpolated axes

(*Movement towards conveyor A*)

```
XMOVE (INTERPO, 3, 90, 10, 0, 150000, 280000, 0, Vitesse_pos_a, 16#0000);
```

Step 4: Continuous Action

! (*Closing the grabber *)

```
SET Pince;
```

Step 5: How to Activate It

For 2 independent axes

! (*Movement towards conveyor B*)

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

For 2 interpolated axes

! (*Movement towards conveyor B*)

```
XMOVE (INTERPO, 4, 90, 9, 0, X_b, Y_b, 0, Vitesse_pos_b, 16#0000);
```

Step 8: How to Activate It

For 2 independent axes

! (*Movement towards conveyor C*)

```
SMOVE (Axe_x, 5, 90, 9, X_c, Vitesse_pos_c_x, 16#0000);
SMOVE (Axe_y, 5, 90, 9, Y_c, Vitesse_pos_c_y, 16#0000);
```

For 2 interpolated axes

! (*Movement towards conveyor C*)

```
SMOVE (INTERPO, 5, 90, 9, 0, X_c, Y_c, 0, Vitesse_pos_c, 16#0000);
```

Step 6: Continuous Action

! (*Opening the grabber *)

```
RESET Pince;
```

Programming the Post-processing

At a Glance

Post-processing is performed at the end of a task. Post-processing is located at the end of the project and will allow you to program management of manual mode.

End Section

! (*Test of selected mode*)

```
IF Axe_x.Mode_auto AND Axe_y.Mode_auto AND Axe_x.Config AND Axe_y.Config
THEN JUMP %L200;
END_IF;
```

! (*Selection of axis to be controlled*)

```
%L100: IF NOT Selection_x_y
THEN JUMP %L200;
END_IF;
```

! (*X axis manual reference point*)

```
IF RE Po_man
THEN Axe_x.Posrp := 0; SET Axe_x.Setrp; Fmanu_x := 1000; WRITE_PARAM
(Axe_x);
END_IF;
IF NOT Axe_x.Po_man
THEN RESET Axe_x.Setrp;
END_IF;
```

! (*Moving the moving part in X axis + direction*)

```
Axe_x.Jog_p := Forward;
```

! (*Moving the moving part in - X axis direction*)

```
Axe_x.Jog_m := Reverse;
%L200: IF selection_x_y
THEN JUMP %L300;
END_IF;
```

! (*Moving the moving part in Y axis + direction*)

```
Axe_y.Jog_p := Forward;
```

! (*Moving the moving part in Y axis - direction*)

```
Axe_Y.Jog_m := Reverse;
```

! (*Open grabber*)

```
%L300: IF Auto_man AND Ouv_pince
THEN RESET Grabber;
END_IF;
```

! (Closing the grabber *)

```
IF Auto_man AND Ferm_pince
THEN SET Grabber;
END_IF;
```

! (*Fault acknowledgement*)

```
Axe_x.Ack_def := Axe_y.Ack_def := Acq_defauts;
%L999:
```

TSX CAY Module Configuration

Software Declaration of the PLC Configuration

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

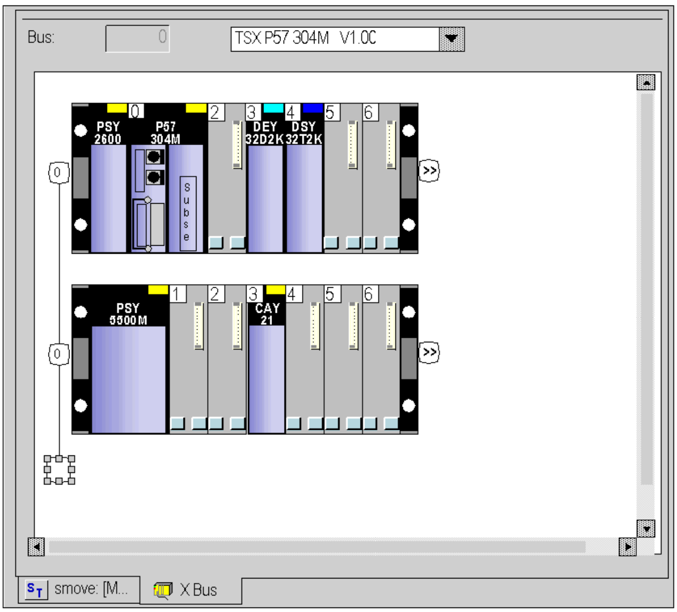
From the **Project Browser**, access the hardware configuration editor in the following manner:

Step	Action
1	Open the **Station** file (either by double clicking on the icon or clicking on its link).
2	Open the **Configuration** file (either by double clicking on the icon or clicking on its link).
2	Open the **Bus X** file (either by double clicking on the icon or clicking on its link).

Each component part of the PLC configuration must then be selected. The following choices have been made in this application:

- rack 0 and rack 1: TSX RKY 8E,
- processor: TSX P57 304,
- power supply modules: TSX PSY 2600 for Rack 0 and TSX PSY 5500 for Rack 1,
- 32 input module: TSX DEY 32D2K in position 3 of Rack 0,
- 32 output module: TSX DSY 32T2K in position 4 of Rack 0,
- axis control module: TSX CAY 21 in position 3 of Rack 1.

Module configuration screen



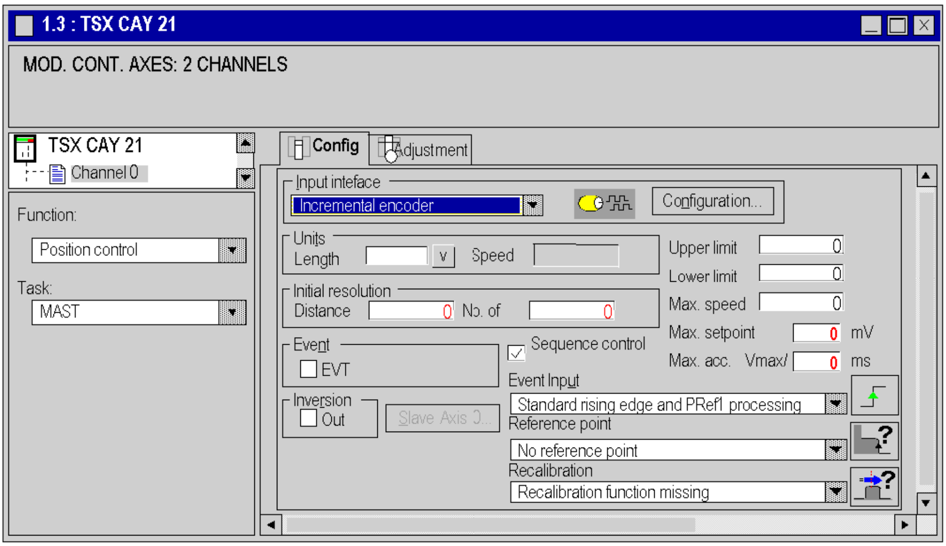
Entry of the Axis Configuration Parameters

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

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Select position 3 of Rack 1, then execute the Edit → Open module command (or double click on the selected module).</td></tr>
<tr><td>2</td><td>Configure the parameters of channel 0. To do this:

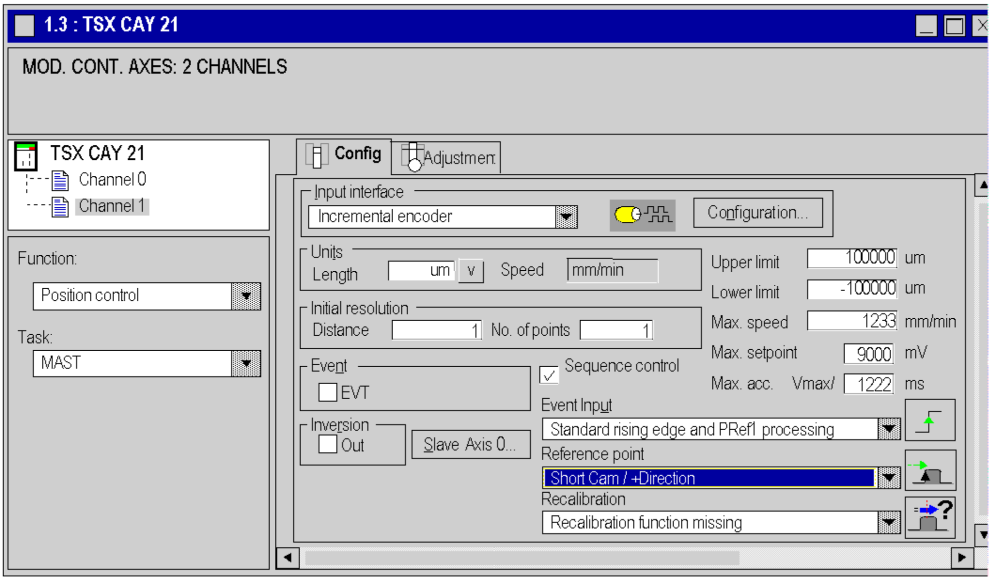
select the Position control function,
select the MAST task,
enter the parameters as shown below:

Channel 0 configuration screen



</td></tr>
<tr><td>3</td><td>Set the adjustment parameters according to the configuration.
Mandatory fields are in red</td></tr>
<tr><td>4</td><td>Confirm your entries with the Edit → Confirm command or by clicking on the icon


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

Step	Action
5	Configure the parameters of channel 1 then validate your entries, by following the same procedure as for channel 0: Channel 1 configuration screen 
6	Set the adjustment parameters according to the configuration. Mandatory fields are in red
7	In the main screen of the configuration editor, validate the configuration using the **Edit** → **Confirm** command or by clicking on the icon 

Channel 0 Configuration Parameters

The following table provides a list of the parameters which must be entered for channel 0:

Parameter	Designation	Value	Comment
Units	Physical length unit	mm	
Units	Physical speed unit	mm/min	automatically deduced
Initial resolution	Distance	4000	
Initial resolution	No. of points	4000	
Encoder type		Absolute	
Encoder type	Offset	Direct	
Encoder type	Code	Binary	
Encoder type	No. of header bits	0	
Encoder type	No. of data bits	24	
Encoder type	No. of status bits	0	
Encoder type	Parity	Odd	
Max. setpoint		9000 mV	
Speed	Maximum speed of the moving part	5400	
Max. acceleration		300 ms	
Upper limit	Upper axis limit	900000	
Lower limit	Lower axis limit	0	
Event		Rising edge and PREF1	
Reference point		No reference point	

Channel 1 Configuration Parameters

The following table provides a list of the parameters which must be entered for channel 1:

Parameter	Designation	Value	Comment
Units	Physical length unit	mm	
Units	Physical speed unit	mm/min	Automatically deduced
Initial resolution	Distance	2000	
Initial resolution	No. of points	500	
Encoder type		Incremental	Default choice
Encoder type		x 1	Default choice
Encoder type	Code	Binary	
Max. setpoint		9000 mV	
Speed	Maximum speed of the moving part	5400	
Max. acceleration		200 ms	
Upper limit	Upper axis limit	500000	
Lower limit	Lower axis limit	-5000	
Event		Rising edge and PREF1	
Reference point		Long cam Top Z - Direction	

Interpolator Configuration

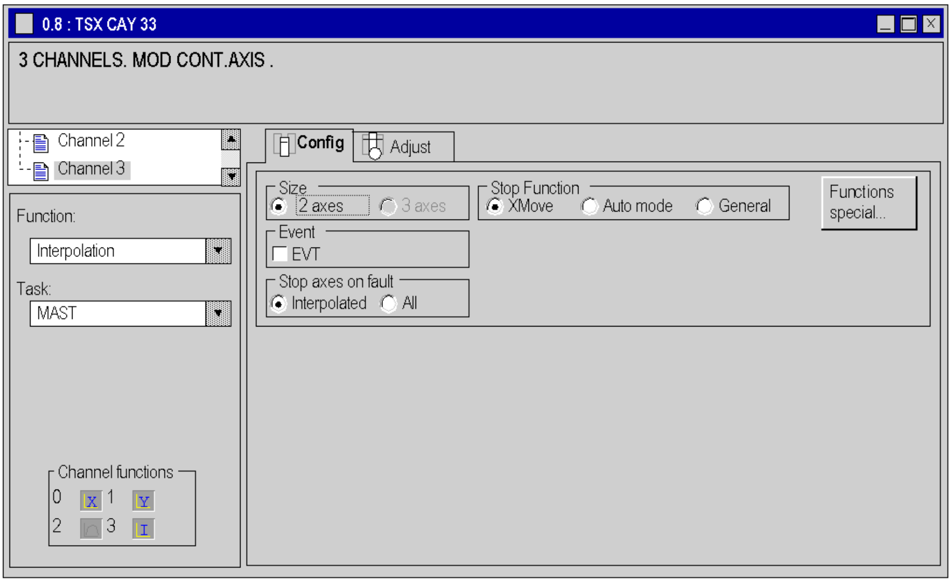
At a Glance

2 interpolated axes can be used to control the grab. In this case:

- Use a TSX CAY 33 module as the axis control module,
- Configure the 0 and 1 axes the same way as with independent axes,
- Configure channel 3, which does not correspond to any physical axis, but which is used to interpolate between axes 0 and 1.

Enter the Configuration Parameters for Axis 3

Enter the configuration parameters for axis 3 as follows:

Step	Action
1	Configure the parameters of channel 3. To do this: ● select the **Interpolation** function, ● select the **MAST** task, ● enter the parameters as shown below: Channel 3 configuration screen 

Step	Action
2	Confirm your entries with the **Edit** → **Confirm** command or by clicking on the icon 
3	In the main screen of the configuration editor, validate the configuration using the **Edit** → **Confirm** command or by clicking on the icon 

Channel 3 Configuration Parameters

The following table provides a list of the parameters which must be entered for channel 3:

Parameter	Designation	Value
Dimension	Number of interpolated axes	2
Stop Function	Effect of the STOP command	XMOVE
Stop axes on fault	Consequence of a fault	All

Parameter Adjustment

Preliminary Operations

Before beginning parameter adjustment, you must first backup the project on the hard drive and transfer it to the PLC.

Adjustment Procedure

The following operations must be performed to adjust the parameters:

Step	Action
1	Set the PLC to RUN.
2	Open the configuration editor for bus X from the project browser.
3	Double click on the module which needs adjusting. Position 3 of Rack 1 can also be selected and the **Edit** → **Open module** command executed.
4	Activate the **View** → **Adjustment** command to access the parameters adjustment screen:
5	Enter the new parameter values for channel 0: ● Target window = 320 micron ● Speed (manual mode) = 5400 mm/min ● PO value = 0 micron
6	Confirm the entries with the **Edit** → **Confirm** command or click on the icon 
7	Select channel 1 in the channel zone then select the new values for this channel: ● Offset encoder = 8388607 ● Following error 1 and 2 = 8000 micron ● Target window = 8000 micron ● Speed (manual mode) = 5400 mm/min
8	Confirm the entries with the **Edit** → **Confirm** command or click on the icon 
9	Save the new values in the PLC processor, using the **Services** → **Save parameters** command.

Using Manual Mode

Accessing Manual Mode

If you would like to move the moving part without first going into the programming phase, use manual mode. In order to do this, access the debug screen, in online mode:

Step	Action
1	Open the configuration editor for bus X from the project browser.
2	Select the TSX CAY module to be opened.
3	Execute the **Services** → **Open module** command (or double click on the module to be opened).
4	Click on the corresponding tab to choose the debugging screen.

Moving in Manual Mode

The following operations must be performed to move the moving part in manual mode:

Step	Action
1	In online mode, set the PLC to RUN, using the **PLC** → **Run** command or by clicking on the icon 
2	Select the axis to be controlled: channel 0 (X axis) or channel 1 (Y axis).
3	Select manual mode by positioning the mode switch to **Manu**.
4	Enable the safety relay of the variable speed controller by clicking on the **Confirm** button in the **Axis** zone.
5	Acknowledge the faults by clicking on the **Ack** button in the **Faults** field.
6	Set a reference point: ● either by using the **Manual reference point** command, ● or by using the **Forced reference point** command, In this case, first enter in the **Param** field, the position value of the moving part in relation to the source.
7	Move the moving part: ● in a positive direction using the **JOG+** command, ● in a negative direction using the **JOG-** command. The moving part's position is displayed in the **X** field and speed is displayed in the **F** field from the **Movement / Speed** zone.

Debugging

Debugging Procedure

The program can be debugged in the following way:

Step	Action
1	Set the PLC to RUN.
2	Display the TSX CAY module Debugging screen.
3	View the Grafcet screen at the same time in order to follow the progress of sequential processing.
4	Start the program by pressing the **Start_cycle** button on the front panel.

Saving

Saving Procedure

When debugging is complete, you may save your project. To do this:

Step	Action
1	If parameters have been modified during debugging, activate the **Services** → **Save parameters** command.
2	Transfer the PLC processor application on to the hard drive: ● activate the **PLC** → **Transfer project to PLC** command, ● activate the **File** → **Save as** command, ● name the application, ● confirm.

Part II
TSX CAY Axis Command Modules

Subject of this Part

This part provides an overview of the TSX CAY axis command modules, their functionality and how to implement them.

What Is in This Part?

This part contains the following chapters:

Chapter 6
Implementing

Subject of this Chapter

This Chapter describes the implementation of TSX CAY axis command modules.

What Is in This Chapter?

This chapter contains the following sections:

Section 6.1
General

Subject of this Section

This Section introduces general instructions for the installation of TSX CAY axis command modules.

What Is in This Section?

This section contains the following topics:

Standard Configuration Required

General

The servo drive axis control modules can be installed in all the available slots in a Premium or Atrium PLC configuration.

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 off the rack supply voltage. The design of the modules allows this action to be carried out with the power on, in order to ensure that a device is available.

⚠ CAUTION
POSSIBLE DAMAGE TO ENCORDERS
Do not connect or disconnect encoder connectors with encoders 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 obtain good electrical contacts, thus guaranteeing effective resistance to electrostatic and electromagnetic interference.

General Precautions for Wiring

General

The supplies to sensors and actuators need non-delay fuses against overload or overvoltage.

When wiring, use wires of a satisfactory size to avoid on-line drops in voltage and overheating,

Keep sensor and actuator cables away from any source of radiation resulting from high-power electric circuit switches.

All cables which link the incremental or absolute encoders must be shielded. The shielding should be good quality and linked to the protective ground connection on the side of the module and the side of the encoder. Continuity must be ensured throughout connections. Do not introduce any other signals than those of the encoders in the cable.

For reasons of performance, the auxiliary inputs of the module have a short response time. You must therefore make sure that the supply autonomy of these inputs is sufficient to ensure the module continues to operate correctly in the event of short power breaks. It is recommended that you use regulated supplies to ensure more reliable response times from the actuators and sensors. The 0 V supply must be linked to the protective ground connection as near to the supply output as possible.

Section 6.2
Select an Encoder

Choice of Encoders

Output Snterface

The output interfaces of incremental encoders or pulse generators are:

- RS 422/485 standard output, two push-pull outputs, complemented by the signal,
- 5 V Totem pole output, two complementary push-pull outputs.

Absolute SSI serial encoders have a standardized RS 485 interface for clock and data signals.

We recommend an encoder with opto type "CLOCK" signal input stage. Different types of encoders can be connected onto the same module. For example, an incremental encoder on channel 0 and an absolute SII encoder on channel 1.

Encoder supply

The module is designed to supply encoders with 5 V or 24 V. Mixing supply voltages is possible on all module channels.

Incremental encoders usually have a 5 V supply.

Absolute SSI encoders have a 24 V (10/30 V) supply.

5 V encoder supply: maximum drop in voltage.

In this case there is reason for taking the on-line voltage drop into account. This drop depends on cable length and encoder consumption for a given wire gauge.

Example for a 100m-long cable:

Section of the wire	Drop in voltage for a 100m-long cable			
Encoder consumption	50 mA	100 mA	150 mA	200 mA
Gauge 28 = 0.08mm^2	1.1 V	2.2 V	3.3 V	4.4 V
Gauge 22 = 0.34mm^2	0.25 V	0.5 V	0.75 V	1 V
0.5 mm^2	0.17 V	0.34 V	0.51 V	0.68 V
1 mm^2	0.09 V	0.17 V	0.24 V	0.34 V

24 V encoder supply.

This type of encoder is recommended, because it does not need a precise supply (10 V/30 V). When there is a 24 V supply, these encoders make it possible to have a very large cable, which makes the voltage drop in the cable rather insignificant. This is the case for SSI serial link encoders.

NOTE: If a 24 V absolute SSI serial encoder is used, it is not necessary to connect the 5 V supply.

Shielding

To ensure good working order in the case of interference, an encoder, whose metal casing is grounded by the connected device, must be chosen. The encoder must ground the connection cable shielding.

Section 6.3
Connecting Speed Reference Signals

Subject of this Section

This section deals with the connection of speed reference signals.

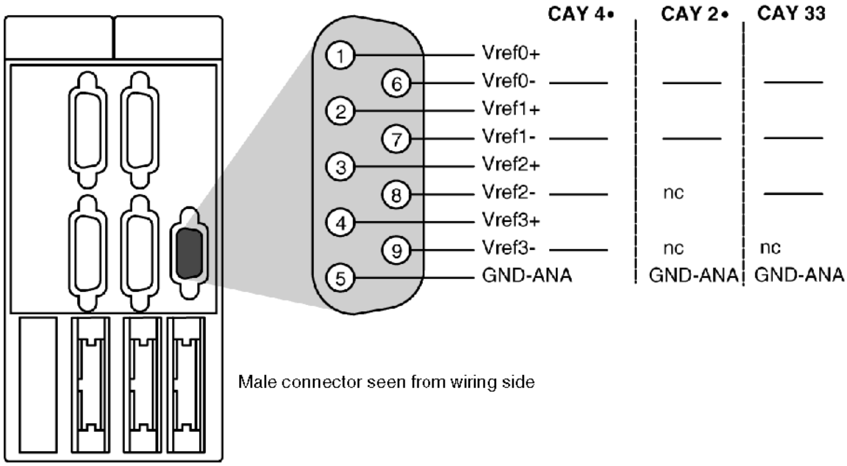
What Is in This Section?

This section contains the following topics:

Signal Labeling

Process Diagram

This diagram illustrates the principles for labeling signals:



Connecting the Speed References

Four types of connection are offered:

- wiring with TSX CAP S9 connector and cover,
- using the TSX CDP 611 strip,
- wiring with output on terminals with TELEFAST ABE-7CPA01,
- wiring with output on TAP MAS (exploding device).

Connection Using TSX CAP S9

General

The connection is made manually by soldering onto the 9 pins SUB-D connector, as labeled in the preceding principle diagram. However, checks must be carried out to ensure that the shielding is properly connected to the cable, which must be correctly clamped to the cover of the connector.

Connection Using TSX CDP 611 Strips

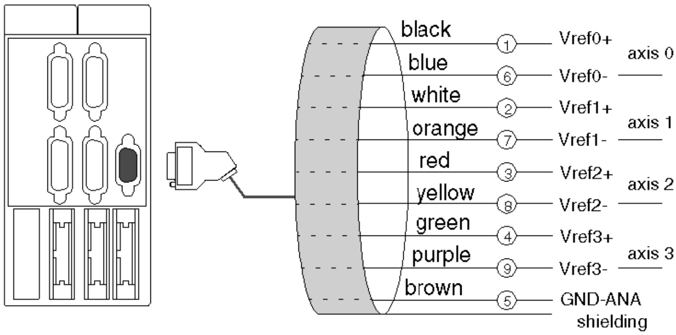
General

This pre-wired cable is made up of a SUB-D 9-pin connector at one end, to connect to the TSX CAY module, and free wires at the other end. With a length of 6m, it is made up of 24 gage wires, corresponding to the SUB-D connector pins; It enables direct connection of the equipment to the module. The different signals are labeled using a color code.

NOTE: It is imperative to connect the shielding to the protective ground of the connected equipment.

Diagram of the Principle

This diagram illustrates the principle for connection using TSX CDP 611 strips:



NOTE: The TSX CDP 611 cable is 6m in length.

Connection of Terminals with the TELEFAST Pre-wiring System

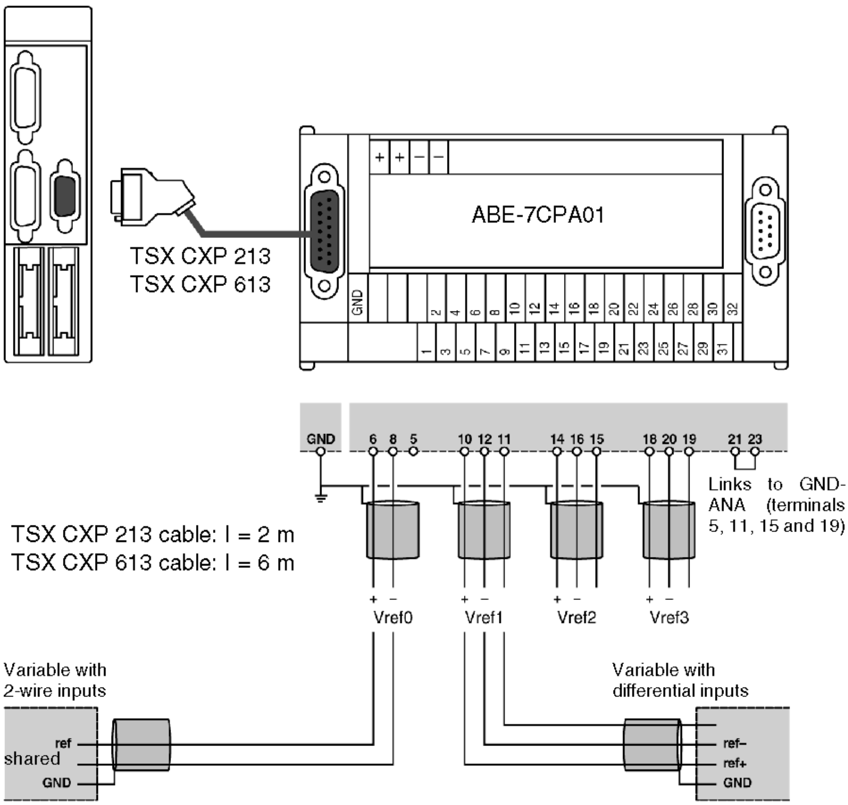
General

The TELEFAST 2 system is a collection of products, which allow rapid connection of the modules from the Micro and Premium range. It acts as a substitute for screw terminal blocks, by realigning the single wire connection.

The connection on speed reference terminals is necessary when the variable speed controllers are not close to each other. The TELEFAST pre-wiring system facilitates installation by allowing access to signals via the screw terminal blocks. Connection to the module with the TELEFAST reference: ABE-7CPA01 assists a cable equipped with a 9-pin SUB-D connector on the module side and a 15 pin SUB-D connector on the TELEFAST side. This cable can be: TSX CXP 213 or TSX CXP 613.

Diagram of the Principle

This diagram illustrates the principle for connection with the TELEFAST pre-wiring system:



Correspondence Between the SUB-D Connector Pins and the TELEFAST Terminals

General

This table shows the correspondence between the SUB-D connector pins and the TELEFAST terminals:

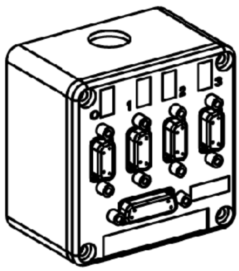
TELEFAST screw terminal block (Terminal No.)	Standard SUB-D 15-pin connector (Pin No.)	TSX CAY module SUB-D 9-pin connector	Kind of signal
2	1		
4	2		
5			
6	10	1	Vref0+
8	3	6	Vref0-
10	11	2	vref1+
11			
12	4	7	Vref1-
14	12	3	Vref2+
15			
16	5	8	Vref2-
18	13	4	Vref3+
19			
20	6	9	Vref3-
21			link to terminal 23
22	nc		
23	14	5	GND-ANA
24	nc		
26	nc		
28	nc		
30	nc		

TAP MAS Connection Device

General

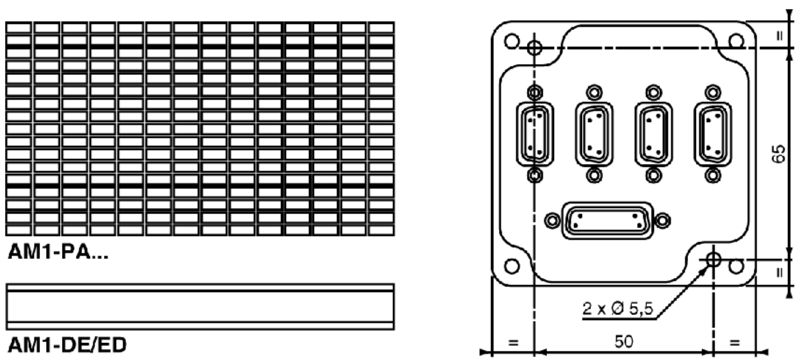
The connection device enables the speed references of each variable speed controller to start again at the same time. This allows the simple connection of several variables, while maintaining good ground connection continuity.

Illustration of the connection device:



Dimensions and Fixing

The TSX TAP MAS device is installed either on an AM1 PA… type perforated board or on a DIN rail with an LA9 D09976 fixation board with two M3x8 or M3x10 screws:



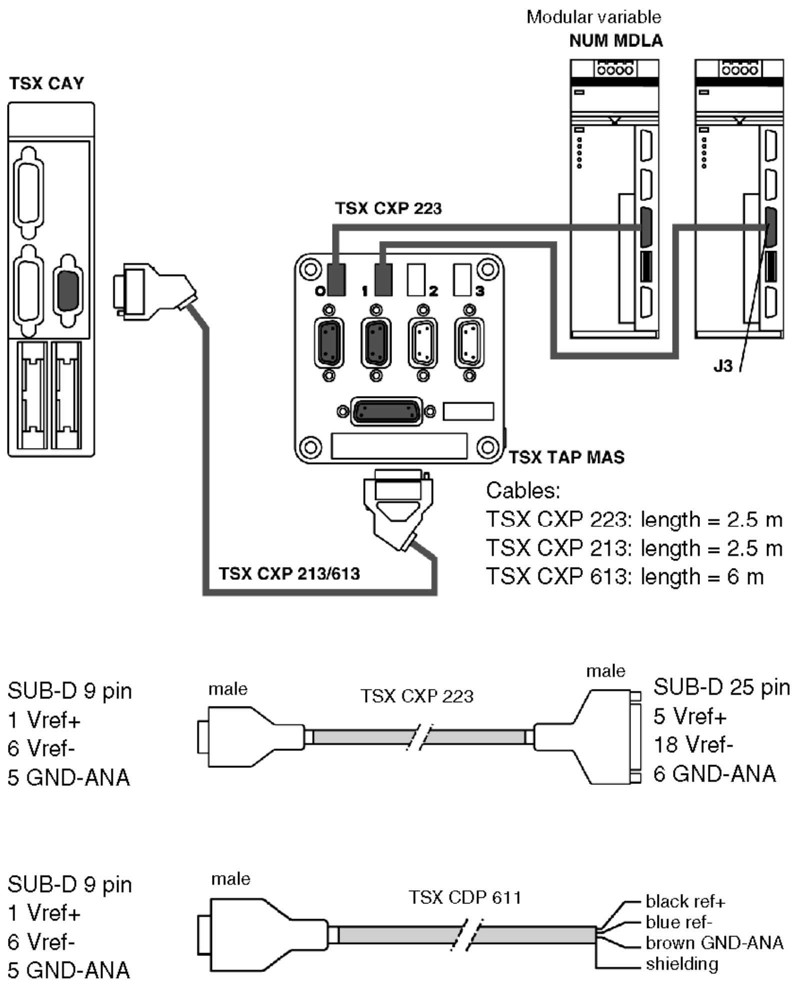
Connecting the Variable Using the TAP MAS Device

General

The NUM MDLA modular variable speed controllers can be connected to the TSX CAY module using the TSX TAP MAS connection device. Installation is simplified by using predefined cables and the connection device, which simply directs the voltage references to the different axes.

Illustration

This diagram illustrates the principle for connection using the TAP MAS connection device:



Section 6.4
Connecting the Counting Signals

Subject of this Section

This section deals with the connection of counting signals.

What Is in This Section?

This section contains the following topics:

Connecting Counting Signals

Introduction

To ensure position measurement, the TSX CAY modules are equipped with connectors allowing direct connection of an incremental or absolute SSI encoder on each channel. Each of these channels can be equipped with a different type of encoder.

Signal Labeling

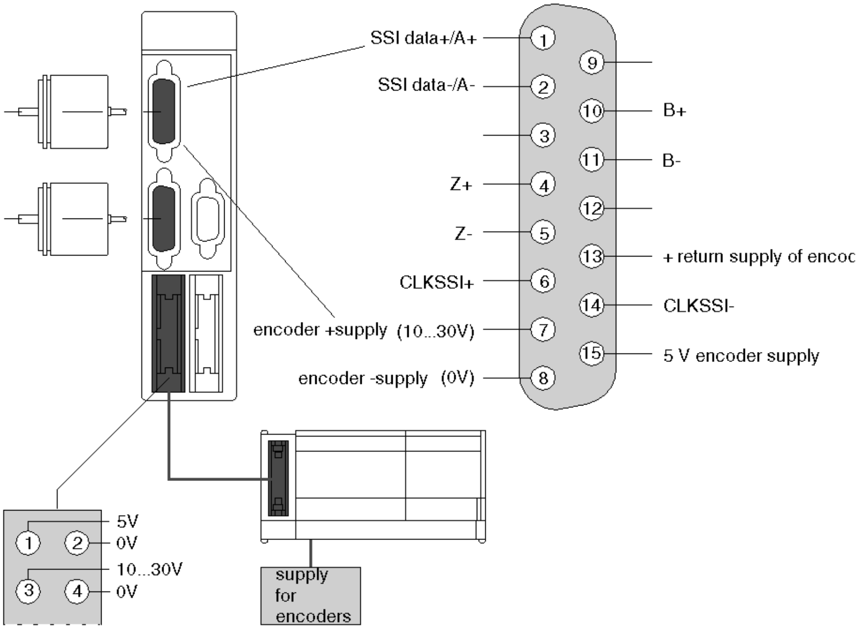
TSX CAY modules can be connected either to incremental encoders, or to SSI type encoders with serial links. In configuration mode, the available functions are as follows:

- Two types of interface are possible for the incremental encoders:
  - RS 422/RS 485 outputs with two outputs complemented by a signal,
  - 5 V Totem Pole outputs.
- Absolute SSI encoder, standard RS 485 interface.

A 15-pin SUB-D connector is assigned to each channel. This also allows the encoder supply. These supplies are elaborated from the +supply discrete HE10 connector. Signal: +return supply encoder, from the encoder allows monitoring for accidental disconnection of the encoder.

Illustration

This diagram illustrates the principles for labeling signals:



Branching

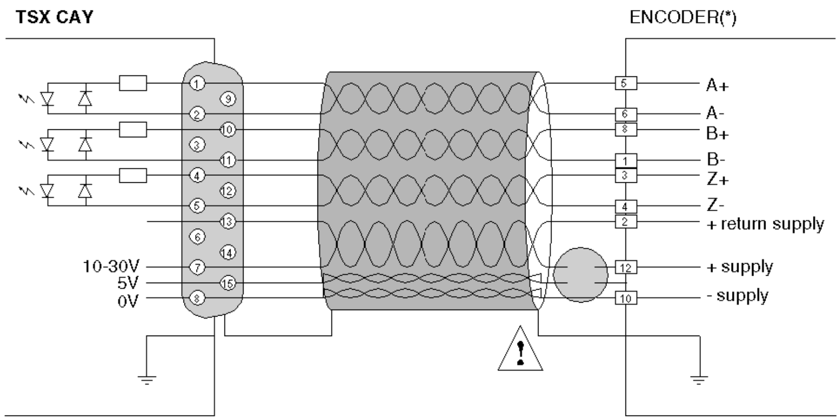
Branching table:

Element	Designation	Terminal
Incremental encoder	input A+	1
	input A-	2
	input Z+	4
	input Z-	5
	input B+	10
	input B-	11
	return supply of encoder	13
Absolute SSI encoder:	+ SSI Data	1
	- SSI data	2
	CLKSSI+	6
	CLKSSI-	14
5 V encoder supply	+supply (5 V)	15
	- supply (0 V)	8
Encoder supply (10-30 V)	+supply (10-30 V)	7
	- supply (0 V)	8

Connecting an Incremental Encoder

Connection Diagram

The type of interface is either RS 422 / RS 485 or totem pole:



(*) standard pinouts for an encoder equipped with a 12-pin DIN connector.

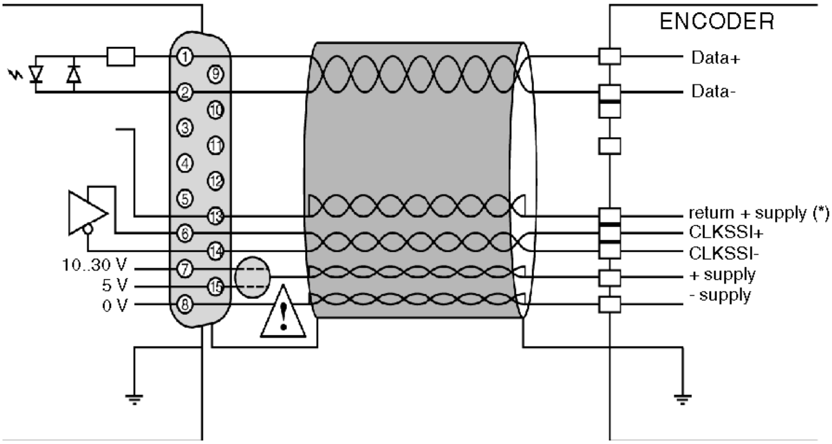
Each signal (A+, A- for example) should be connected by a twisted pair. To reduce on-line voltage falls, it is recommended to connect each supply point using a pair. Cable shielding should be connected at each end to the protective ground.

⚠ CAUTION
IMPROPER POWER SUPPLY Connect the +supply encoder input of the DIN connector to a 10-30 V supply wire or a 5 V wire, according to the type of encoder used. **Failure to follow these instructions can result in injury or equipment damage.**

Connecting an Absolute SSI Encoder

Connection Diagram

illustration:



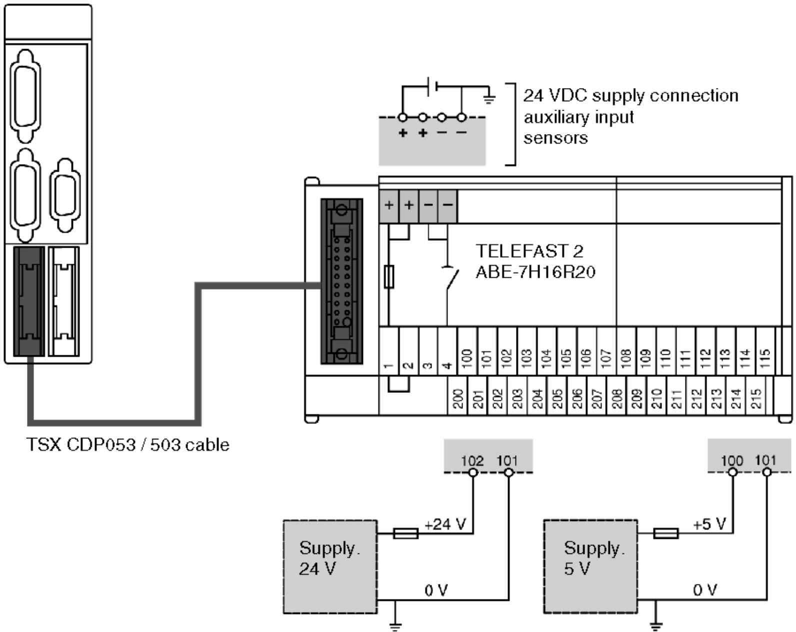
⚠ WARNING
Connecting the encoder supply
Connect the encoder supply to pin 15 or 7 of the SUB-D connector, according to the encoder supply voltage.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

(*) + return supply: encoder output, which returns the supply voltage to the module, therefore allowing the module to monitor the presence of the encoder.

Connecting the Encoder Supply

Diagram of the Principle

This diagram illustrates the connection of the encoder supply:



Cable length:

Cable	Length
TSX CDP 053	0.5 m
TSX CDP 103	1 m
TSX CDP 203	2 m
TSX CDP 303	3 m
TSX CDP 503	5 m

NOTE: The maximum length of the wire between the supply outputs and the connection points on the TELEFAST should be less than 0.5 m.
Only one supply is required if the encoders on the two channels are of the same type.

Fuses

This module integrates several basic protection systems against wiring errors and accidental short circuits on the cable:

- polarity inversions of the supplies,
- inversion of 5 V supplies <--> 10/30 V,
- 10/30 V short circuit on the CLOCK signal of the serial link.

The module cannot tolerate them for very long time, it should therefore have very fast blow fuses. The fuses should therefore be "rapid" and of 1A caliber maximum. Supplies should have a limitation current, such that the blow of the fuse can be correctly executed.

Section 6.5
Wiring Accessories

Subject of this section

This section introduces the wiring accessories for the TSX CAY modules.

What Is in This Section?

This section contains the following topics:

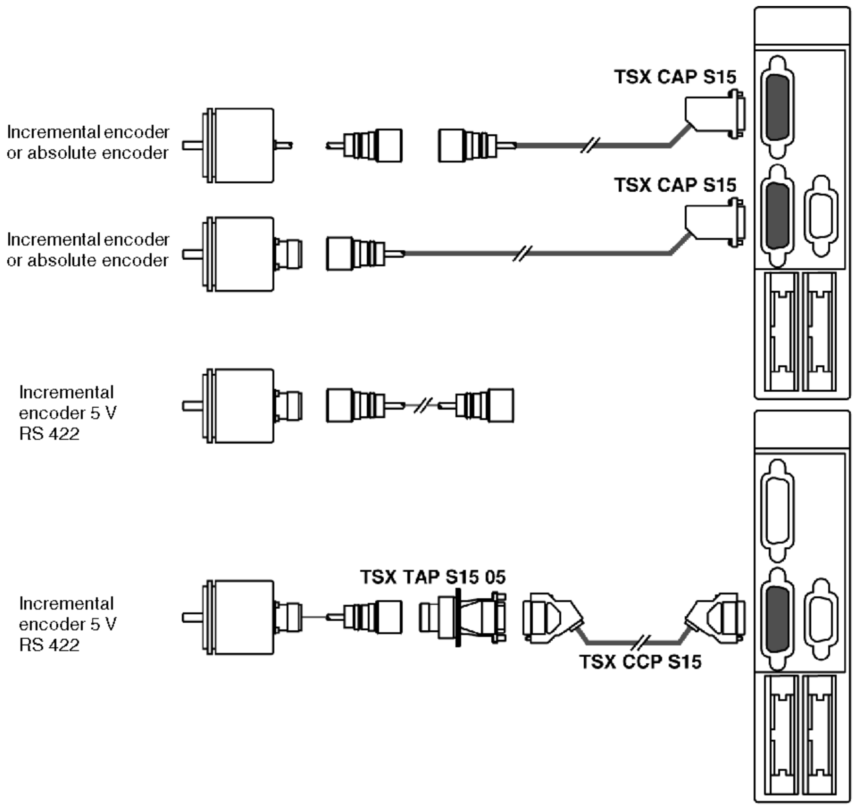
Encoder Connection Accessories

General

A number of accessories are available to facilitate implementation and installation. These accessories are used to pre-wire the installation. A direct link with the installation can be established using cover kits containing the 15 pin SUB-D connector, TSX CAP S15. To facilitate installation, the TSX TAP S15 05 is used as an interface between the SUB-D and 12 pin DIN connector. Using a fixing hook, this accessory can be mounted on a DIN rail or on a cabinet lead-in with a gasket and adjusting nut. Connection to the module is via a 2.5m long TSX CCP S15 cable.

Examples

Illustration:



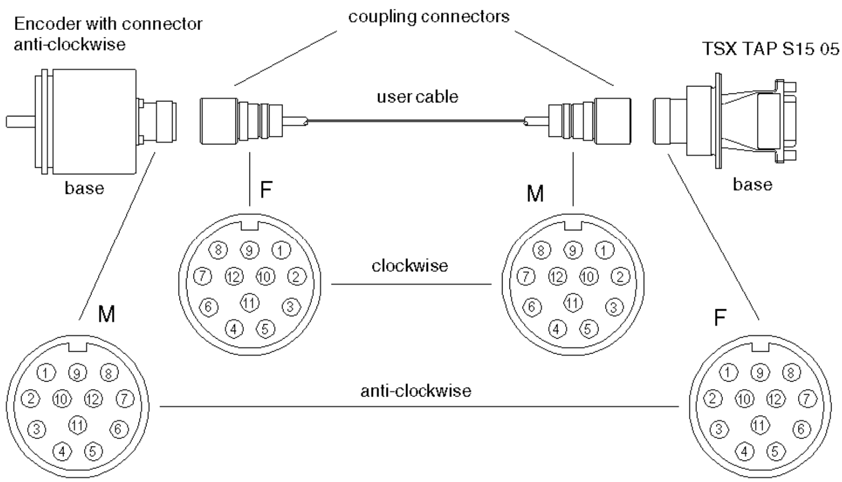
NOTE: Good signal and shielding continuity can be ensured in difficult conditions thanks to these accessories. Encoder connection cables can generally be obtained from encoder suppliers.

Information on FRB Type 12-Pin connectors

General

Number labeling of pins in these connectors is performed in two different ways. Most encoders have a built-in 12-pin base and are labeled anti-clockwise. The TSX TAP S15 has a 12-pin female base labeled anti-clockwise. All user cables must be equipped with connecting plugs labeled clockwise, so that the pin numbers correspond to one another when wired.

Illustration:



Labeling of the DIN and 15-pin SUB-Connector of the TSX TAP S15 05

Table of numbers:

DIN Pin	Signal	SUB_D Pin
1	B-	11
2	Supp return	13
3	Z+	4
4	Z-	5
5	A+	1
6	A-	2
7	nc	
8	B+	10
9	nc	
10	0 V	8
11	nc	
12	5 V	15

Shielding should be continuous along the connections which should be linked to the mechanical ground connection on both sides.

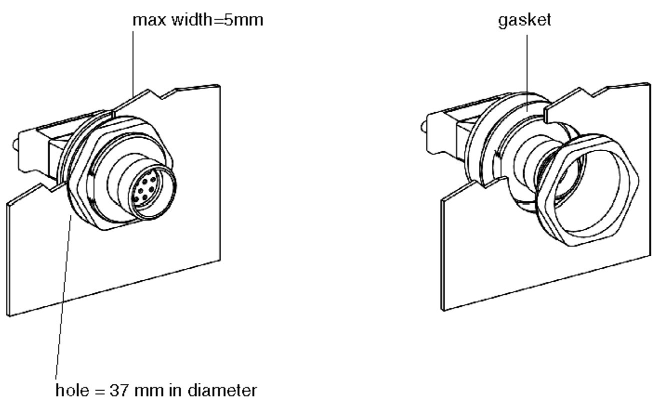
TSX TAP S15 05 Mounting and Dimensions

Mounting onto a Telequick Board

The TSX TAP S15 05 can be attached to an AM1-PA••• type perforated board or any other support using the bracket supplied.

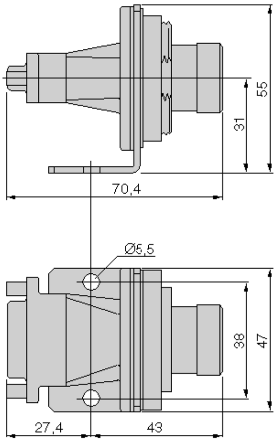
Mounting Through a Cabinet

The TSX TAP S15 05 can be mounted through a cabinet as it has a fixing nut. Its joint creates an impervious seal between the interior and exterior.



Dimensions

Illustration:



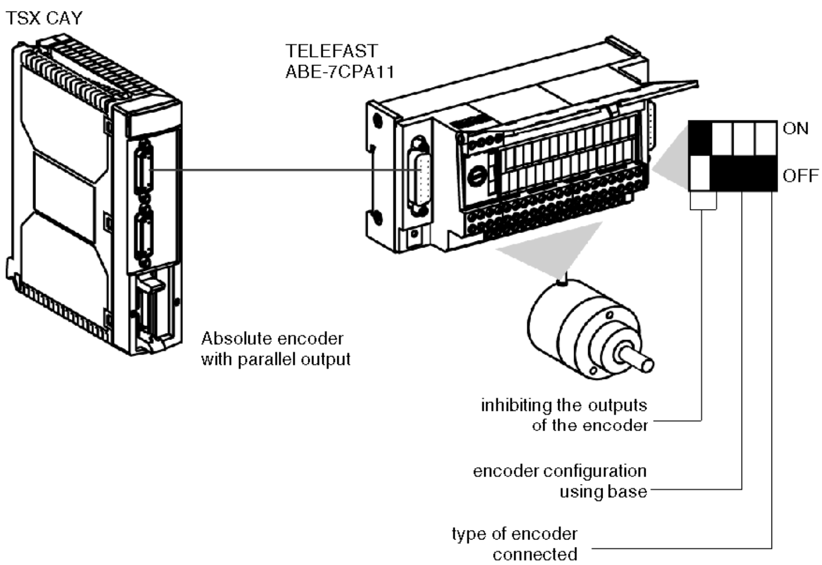
Connecting Absolute Encoder // via a TELEFAST with ABE-7CPA11 Adaptation

General

- the multiplexing function must not be used: each channel uses a base, to which only one absolute encoder with parallel outputs is connected,
- the encoder frame should be configured as follows:
  - code: binary or Gray (according to the encoder type),
  - header bits: 0,
  - data bits: 24 (irrespective of the number of encoder data bits),
  - status bits: 3,
  - rank of the error bit: 1 (optional),
  - parity: even.

Illustration

This diagram shows the connection between a TSX CAY and a TELEFAST ABE-7CPA11:



Connecting to a NUM MDLA Variable Speed Controller

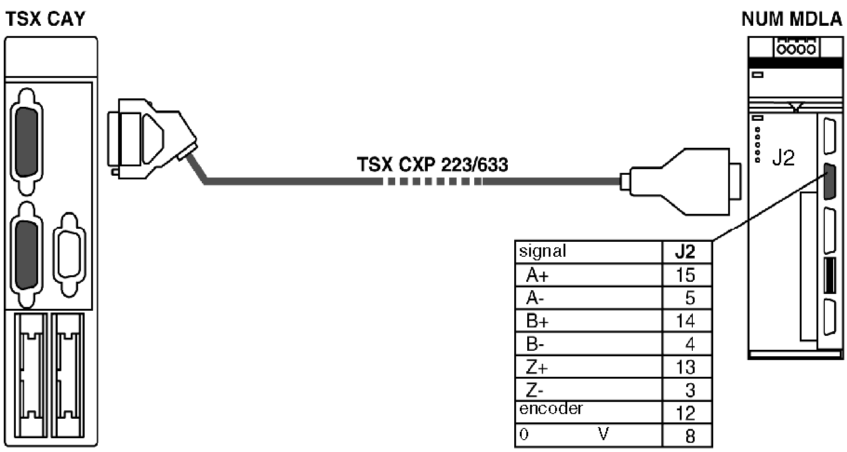
General

The NUM 400 V controller contains all the elements necessary to functioning.

It offers an output whose signals simulate the functioning of an incremental encoder as a position report. Direct connection is possible using the 2.5 cm or 6m long TSX CXP 233 / 633 cable accessory.

Illustration

Connection to a variable speed controller:



Cable length:

Cable	Length
TSX CXP 213	2.5 cm
TSX CXP 633	6 m

NOTE: Here, it is not necessary to have an encoder supply.

Section 6.6
Connection of Sensors/Pre-actuator and Supply Modules, Without Variable Speed Controller

Subject of this section

This section deals with the connection of sensors/pre-actuators and supply modules without a variable speed controller.

What Is in This Section?

This section contains the following topics:

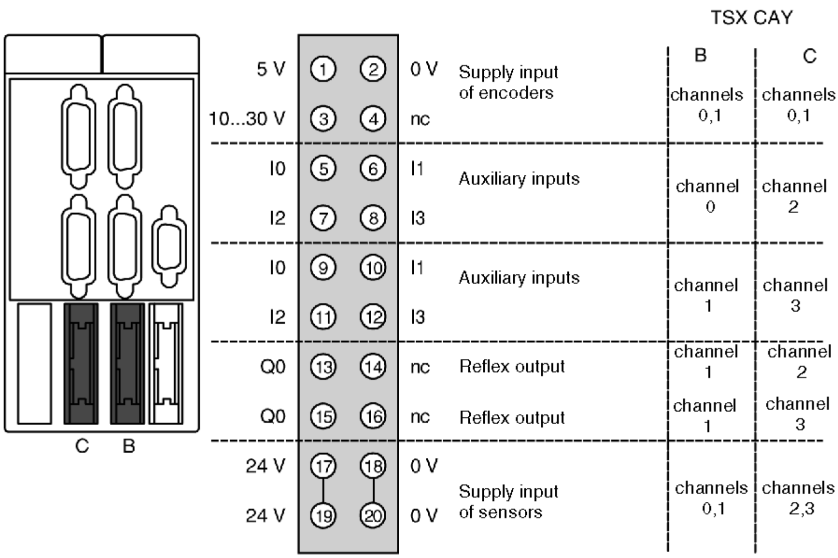
General

Introduction

The TSX CAY modules integrate basic inputs/outputs, which ensure complete functioning of the movement command, as well as ensuring the encoder supply.

Signal Labeling

The connector is a high density HE10:



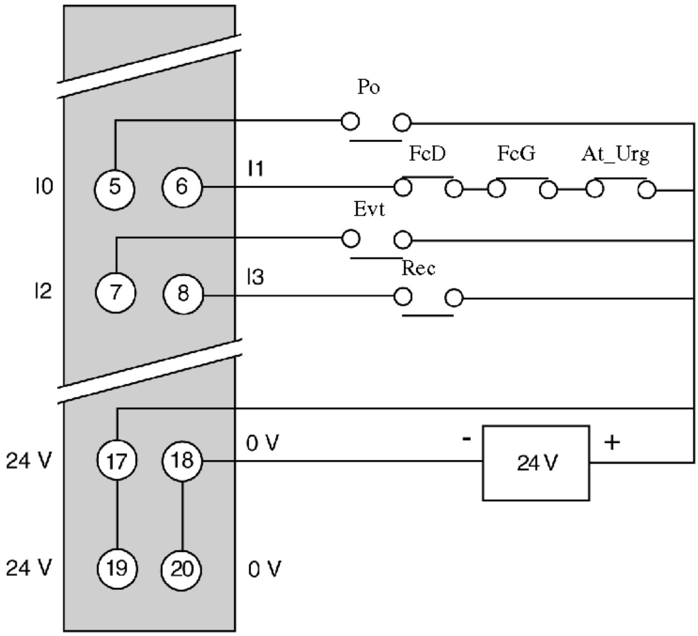
TSX CAY 2• module: Channels 0 and 1
TSX CAY 4• module: Channels 0,1,2 and 3
TSX CAY 33• module: Channels 0,1 and 2

The auxiliary inputs/outputs are allocated the following functions:

- I0 = cam reference point input,
- I1 =emergency stop input (stop if there is no current in the input),
- I2 = adjusting input,
- I3 = adjustment input,
- Q0 = reflex output (static output),
- 0 V = shared auxiliary inputs and reflex outputs.

Principle for Connecting the I/O Associated with Channel 0

Illustration:



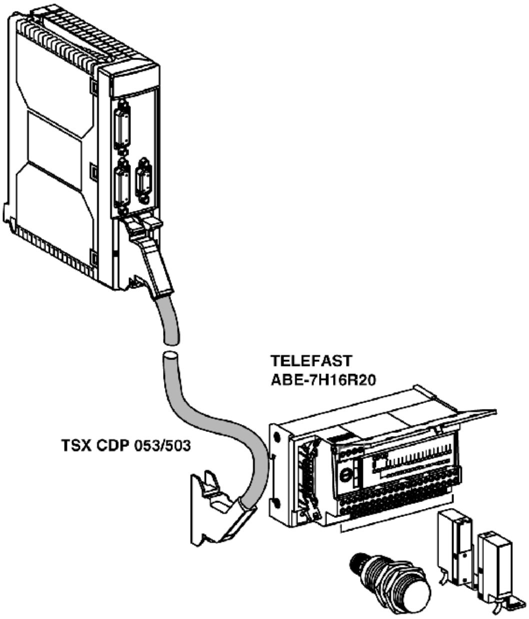
TELEFAST Connection and Wiring Accessories

General

When connecting this high density connector, it is recommended that you use the discrete TELEFAST ABE-7H16R20 pre-wiring accessory and the TSX CDP 053/503 cable or a 3m long strip of the 20-wire TSX CDP 301 or a 5m strip of the TSX CDP 501, which contains a HE10 connector at one end and free wires at the other.

Illustration

Discrete TELEFAST wiring:



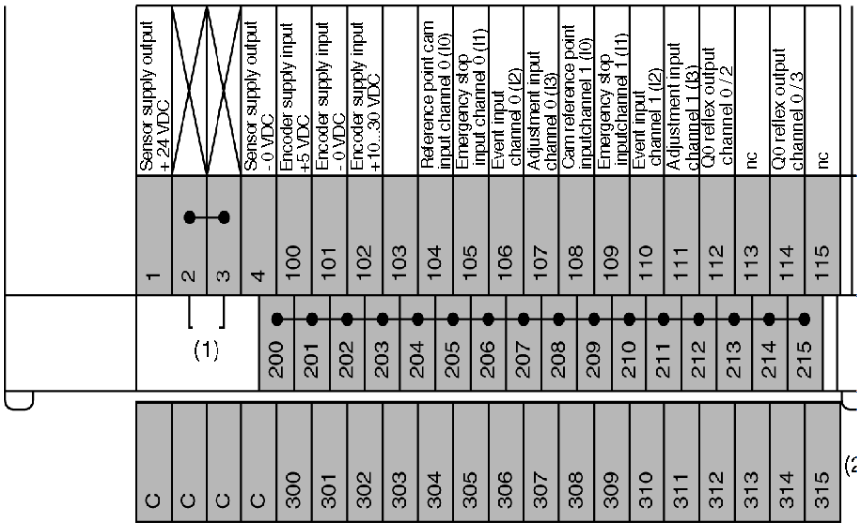
Cable length:

Cable	Length
TSX CDP 053	0.5 m
TSX CDP 103	1 m
TSX CDP 203	2 m
TSX CDP 303	3 m
TSX CDP 503	5 m

Availability of Signals on TELEFAST

Illustration

The terminal below represents the terminal of the ABE-7H16R20 base. The signals are represented using TSX CDP 053 / 503 cable:



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

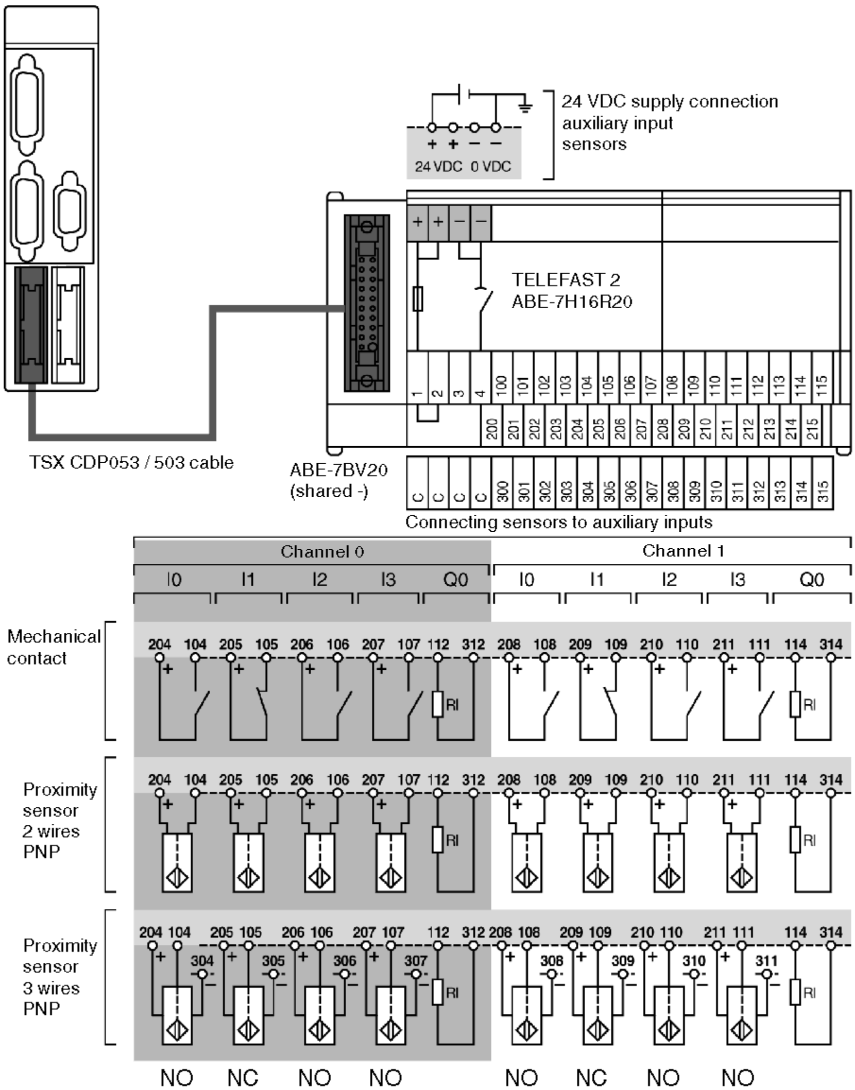
- jumper wire in position 1 or 2: terminals 200 to 215 have + polarity,
- jumper wire in position 3 or 4: terminals 200 to 215 have - polarity,

(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).

Example of Connecting Sensors to the Auxiliary Inputs and Their Supply.

Illustration

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



NO: Normally Open.

NC: Normally Closed (Conductor).

Correspondence Between TELEFAST Terminal Blocks and Module 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.)	Kind of signal	
100	1	+5 VDC	Encoder supply
101	2	- 0 VDC	
102	3	+10...30 VDC	
103	4	nc	
104	5	Reference point cam input I0 (channel 0)	Channel 0 auxiliary inputs:
105	6	Emergency stop input I1 (channel 0)	
106	7	Event input I2 (channel 0)	
107	8	Recalibration input I3 (channel 0)	
108	9	Reference point cam input I0 (channel 1)	Channel 1 auxiliary inputs
109	10	Emergency stop input I1 (channel 1)	
110	11	Event input I2 (channel 1)	
111	12	Recalibration input I3 (channel 1)	
112	13	Q0 reflex output (channel 0)	
113	14	nc	
114	15	Q0 reflex 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			

TELEFAST screw terminal block (Terminal No.)	HE10 20-pin connector (Pin No.)	Kind of signal
3		Terminals 200 to 215 at –0 VDC
4		
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

The same wiring applies to the TSX CAY 4• modules for channels 2 and 3, as well as for channel 2 of the TSX CAY 33 module.

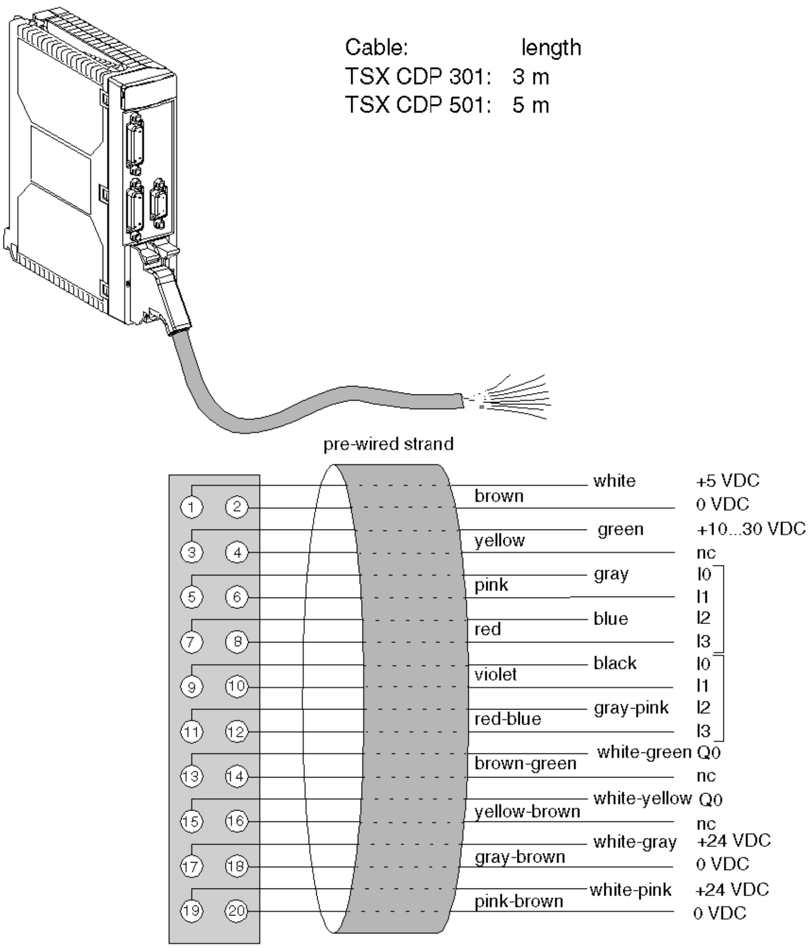
Connection Using TSX CDP 301 or 501 Strips

Introduction

Connection using strips allows a direct connection to actuators, pre-actuators or terminals. This strand comprises 20 gage 22 wires (0.34 mm^2) with a HE10 connector at one end and free wires at the other end, each identified using a color code.

Illustration

This diagram shows the relation between the color of the wires and the pin number of the HE10 connector:



Wiring precautions

General

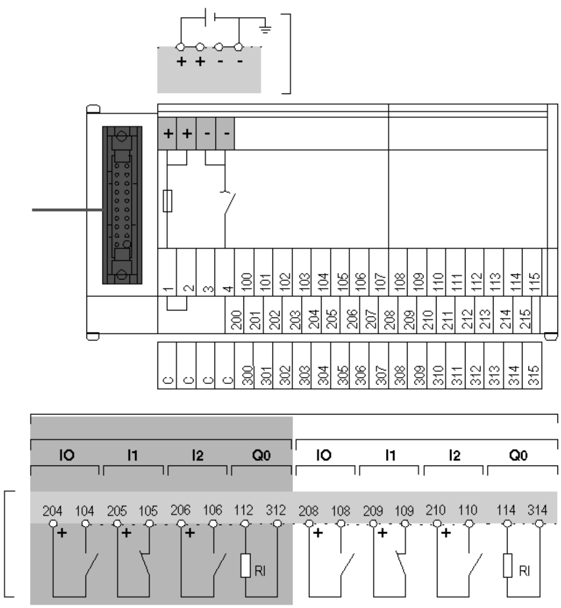
The I0, I1 and I3 inputs are rapid inputs, which should be connected to the sensor using either a twisted wire if it is a dry contact, or using shielded cables if it is a 2 or 3-wire proximity sensor.

The module integrates 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 non-delay fuses, the supply energy must be sufficient to ensure their fusion.

Important Note: Wiring of Q0 Static Outputs

The actuator connected to the Q0 output has its shared point at 0 V of the supply. If for any reason (poor contact or accidental unplugging) there is a 0 V outage of the output amplifier supply, 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 locked.

Illustration:



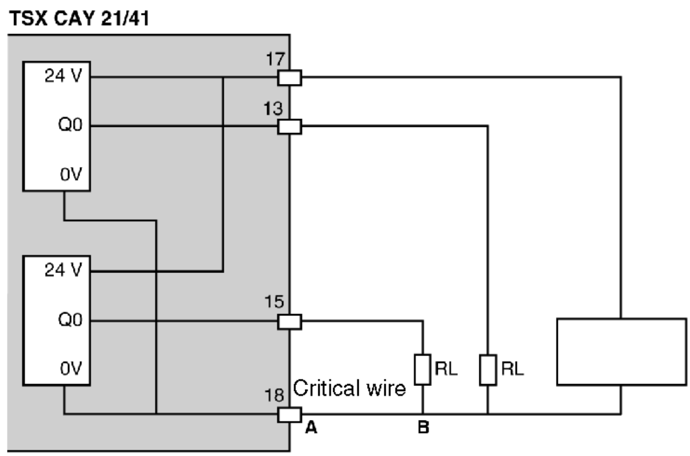
Connection via TELEFAST

This kind of connection provides the most guarantees, on condition that the shared actuators are connected to the bar for shared points 200 to 215 (jumper wire in position 1-2). In this case there can be no outage of the shared module without an outage of the shared actuators.

Connection Using Strips

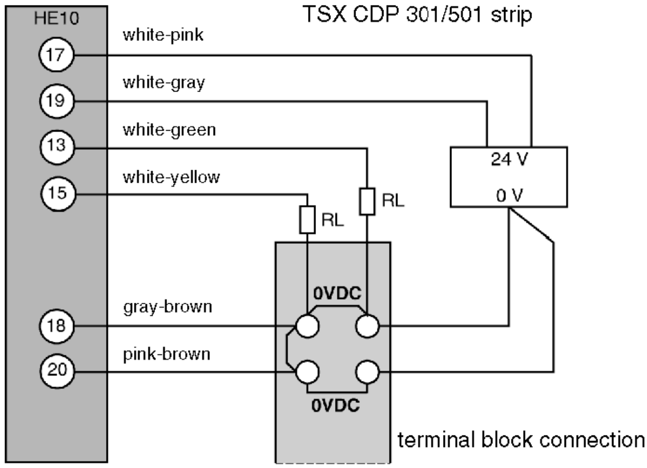
This kind of connection must be carried out with the highest care and attention. It is recommended that you take special care in wiring this cable, for example using cable markers on screw terminals. It may be necessary to double the connections in order to ensure permanent contacts. When the actuator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

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.

Using TSX CDP 301/501 strips:



Section 6.7
Connecting the Variable Speed Controller Signals

Subject of this section

This section deals with the connection of variable speed controller signals.

What Is in This Section?

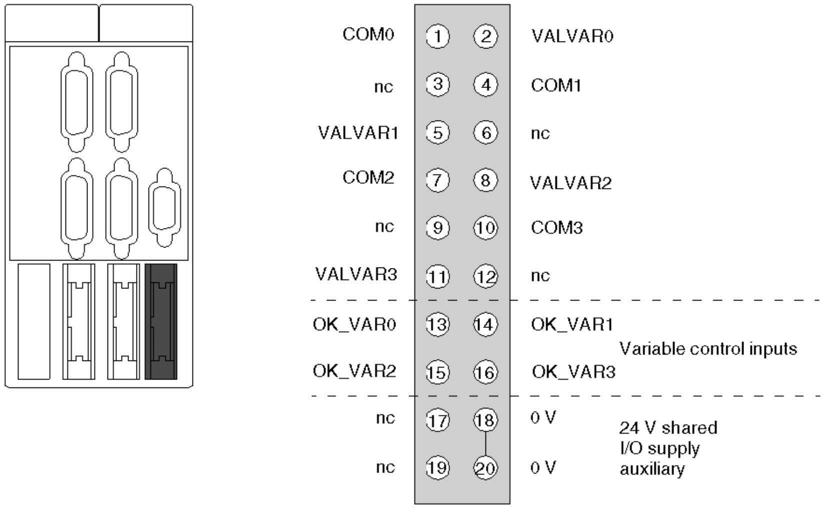
This section contains the following topics:

Signal Labeling

General

The TSX CAY modules implement basic management of the signals necessary for correct operation of the variable speed controllers. There is only one connector, regardless of the number of TSX CAY module channels.

Illustration:



COMx – VALVARx: potential free contact to validate variable speed controller

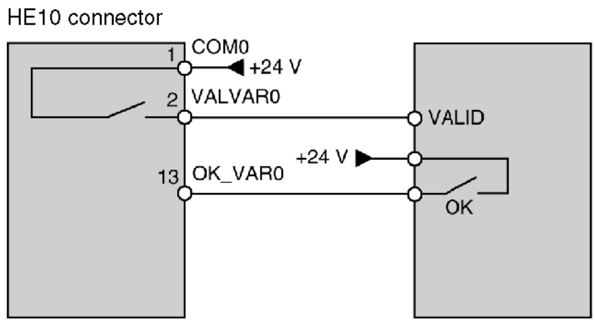
OK_VARx: variable speed controller input check

24 V – 0 V sensor supply

NOTE: Each channel uses a potential free closing contact.

Principle for Connecting the Variable Speed Controller I/O Associated with Channel 0

Illustration:

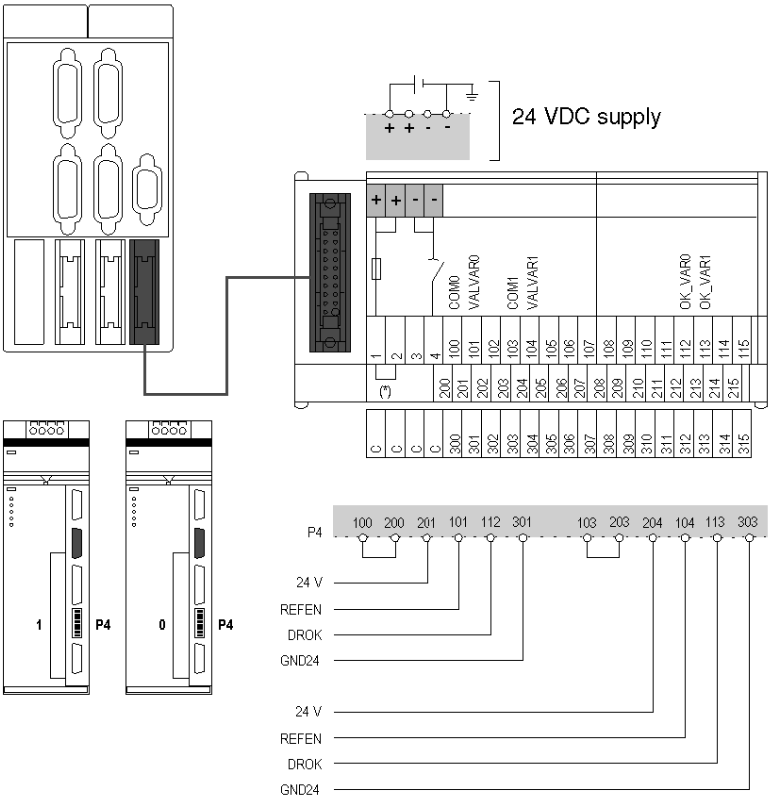


To connect this HE10 connector, use the discrete ABE-7H16R20 TELEFAST wiring accessories and the TSX CDP 303 or TSX CDP 503 cable.

Connection Using the TELEFAST Pre-wiring System

Diagram of the Principle

This diagram illustrates the principles for connection:



To connect directly, use the TSX CDP 301 or 501 strip *(see page 113)*.

(*) Strap between 1 and 2: terminals 200 to 215 are +24 VDC.

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.)	Kind of signal	
100	1	COM0	closed contact = variable speed controller confirmation
101	2	VALR0	
102	3	nc	
103	4	COM1	
104	5	VALR1	
105	6	nv	
106	7	COM2	
107	8	VALR2	
108	9	nc	
109	10	COM3	
110	11	VALR3	
111	12	nc	
112	13	OK_VAR0	VARiable OK = voltage presence of the encoder supply
113	14	OK_VAR1	
114	15	OK_VAR2	
115	16	OK_VAR3	
+ 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.)	Kind 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.

Chapter 7
Characteristics and Maintenance of TSX CAY

Aim of this Part

This part introduces the different electrical characteristics of the TSX CAY modules, and describes the maintenance actions to be carried out to guarantee the correct operation of the module.

What Is in This Chapter?

This chapter contains the following sections:

Section 7.1
Electrical Characteristics of Modules

Subject of this Section

This section introduces the different characteristics of the TSX CAY axis command modules.

What Is in This Section?

This section contains the following topics:

General Characteristics

Table of Characteristics

This table shows the general characteristics of TSX CAY modules:

Maximum frequency of counting: absolute SSI encoder: transmission CLK frequency incremental encoder		200 kHz 500 kHz x 1 250 kHz x 4	
Current used on internal 5 V (ventilator in operation)	**Module**	**Typical**	**Max.**
	CAY 2• CAY 4•/33	1.1 A 1.5 A	1.4 A 1.8 A
Current used on the 24V sensor/ pre-sensor, outputs OFF	CAY 2• CAY 4•/33	15 mA 30 mA	18 mA 36 mA
Current consumed by the module on the 10/30 V encoder at 24V (1)	CAY 2• CAY 4•/33	11 mA 22 mA	20 mA 40 mA
Power dissipated in the module	CAY 2• CAY 4•/33	7.2 W (2) 10 W (2)	11.5 W (3) 17 W (3)
Insulation resistance	> 10 MΩ under 500 VDC		
Dielectric rigidity with ground connection or 0 V logical PLC	1000 Veff 50/60 Hz per min		
Operating temperature	0 to 60 °C		
Storage temperature	-25 °C to 70 °C		
Hygrometry (without condensation)	5% to 95%		
Operating altitude	< 2000 m		

Note (1): absolute encoder and supply used exclusively in 24V.

Note (2): normal conditions of use: one active auxiliary input per channel (under 24V).

Note (3): "worst" case and extreme conditions: all auxiliary inputs active (under 30 V).

This module has a mini internal ventilator, which enables good working order in all temperatures. The ventilator is started up when necessary by the module’s internal temperature sensor (triggered at an external temp. of 45 °C).

It is possible to use external ventilation blocks (TSX FAN••), if the conditions around the module surpass the above parameters.

Characteristics of the Analog Outputs

Table of Characteristics

This table shows the characteristics of the analog inputs:

Parameters	Value	Units
Range	+/- 10,24	V
Real dynamic	+/- 10,24	V
Resolution	13 bits + signs	
LSB value	1,25	m V
Max. current supplied by an output	1,5	m A
fallback value	max +/- 1	LSB
Monotony	100	%
Differential linearity	+ /- 2	LSB
Accuracy	0,5	% P.E.
Dielectric rigidity between the channels and the protective ground	1000 VAC	

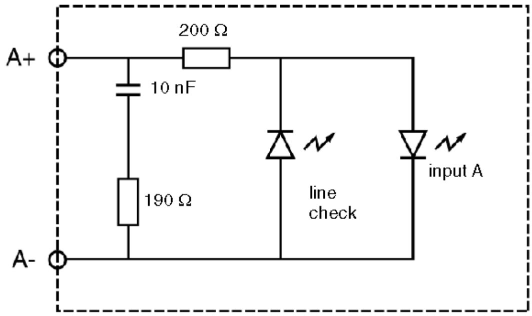
Each output is protected against short circuits or overloads. In case of error, a signal is sent to the CPU using a status word. A short circuit of these outputs is not harmful to the module.

There is no check for an absent connector on the analog output.

Characteristics of the Counting Inputs

Diagram

Example of input A:



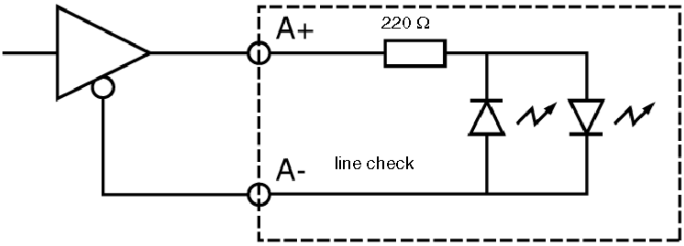
Characteristics

This table shows the characteristics of the counting inputs:

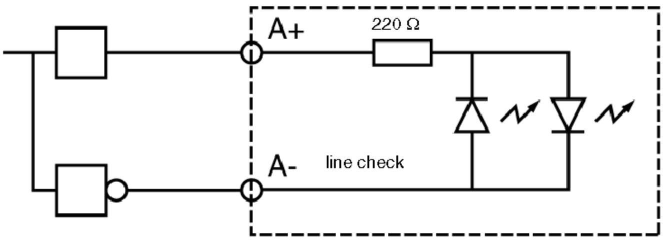
Electrical characteristics	Symbol	Value	Units
Nominal voltage	One	+/- 5	V
Voltage limit	U1	+/- 5,5	V
Nominal current	In	+/- 18	mA
Input impedance (under 5 V)	Re	270	Ohms
Voltage for "On" state	Uon	>= +2,4	V
Current at "On" state	Ion	> +3,7	mA
Voltage for "Off" state	Uoff	<1,2	V
Current at "Off" state	Ioff	<1	mA
Encoder/sensor voltage feedback check	Presence check		

Compatibility of A, B, Z Inputs

RS 422 / RS 485 line transmitter outputs, 7 mA current loop Differential line monitor on each input:



Outputs complemented by 5 V totem pole supply. Differential line monitor on each input:



Characteristics of the Return +supply Encoder Inputs

Illustration:

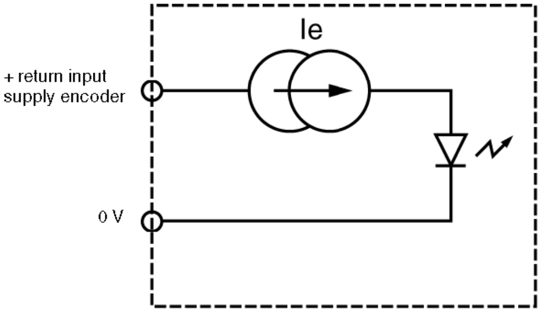


Table of characteristics:

Characteristics	Symbol	Value	Units
Voltage for ON state (OK)	Uok	> 2,5	V
Voltage limits	Umax	30	V
Input current (2.5 < Uok < 30)	Imax	3	mA

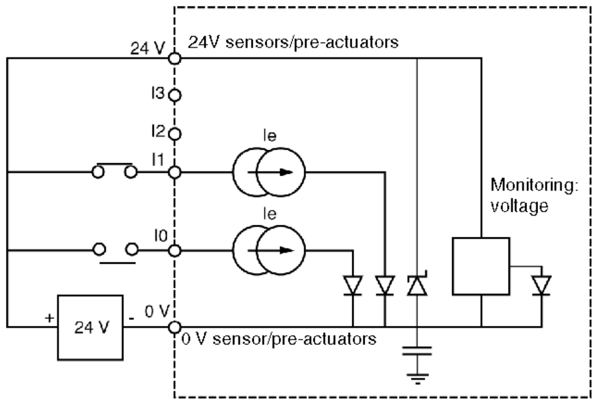
As long as the input is active, the presence of the encoder is detected.

Characteristics of Auxiliary Inputs

Illustration

The inputs use 24 V from a supply provided via the connector.

Diagram:



Characteristics

Table of characteristics for auxiliary inputs:

Electrical characteristics	Symbol	Value	Unit
Nominal voltage	Un	24	V
Voltage limits (1) (ripple included)	U1 Utime (*)	19 to 30 34	V
Nominal current	In	8	mA
Input impedance (at Unom)	Re	3	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
Immunity Off -->On (for I0, I2 and I3) (for I1)	ton	0.1 to 0.2 1 to 4	ms ms
EVT input (on G07)	incremental encoder: 1μs absolute encoder: ≤ 400 μs		
Dielectric rigidity with the ground connection	1500 Veff 50 / 60 Hz for 1 mn		

Electrical characteristics	Symbol	Value	Unit
IEC compatibility with sensors	type 2		
2-/3-wire proximity sensor compatibility	all proximity sensors function at 24 VDC		
Type of input	current ducts		
Logic type	Positive (sink)		

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

Characteristics of the Q0 Reflex Outputs

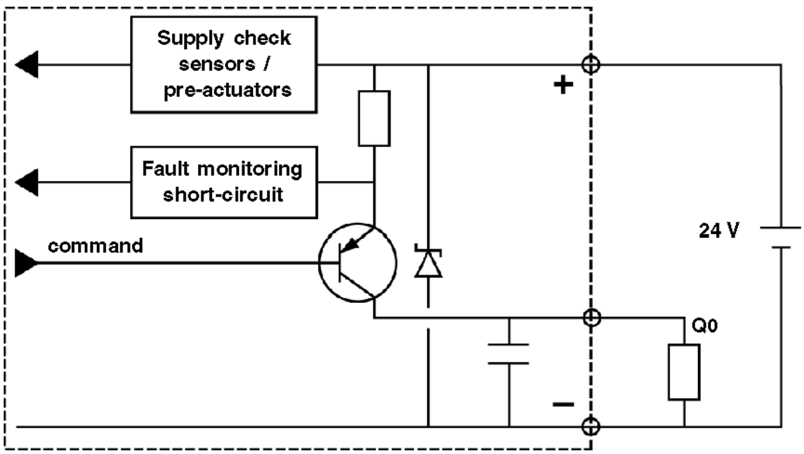
General

Each positioning channel has an output controlled by the processor and which allows the integrated command from an ordered axis function to be performed. For example, a brake command between two shifts, safety etc. This output is static, the shared load is at 0 V of the sensor/pre-sensor voltage.

The output is protected against overloads and short circuits and in case of fault, information is made available to the processor about it.

Illustration

Reflex output:



Characteristics

Table of characteristics:

Electrical characteristics	Value	Units
Nominal voltage	24	V
Voltage limits max for 1 hour in 24 hours (Utime)*	19 to 30 34	V V
Nominal current	500	mA
Max voltage fall "On"	< 1	V
Leakage current	< 0,3	mA
Max current to 30 V and to 34 V	625	mA
Communication time	< 500	µs
Dielectric rigidity with the ground connection	1500 Veff 50/60 Hz per min	
Compatibility with direct current inputs	All positive logic inputs whose input resistance is less than 15 kΩ	
IEC 1131 compatibility	Yes	
Monitoring short-circuits of each channel	One signaling bit per channel	
Reset ● via application program ● automatic	One bit per channel in write mode via program	
Protection against overloads and short-circuits	Using current limiter and thermal circuit breaker (0.7A< id < 2 A)	
Protection against overvoltage of the channels	Zener (breakdown) between outputs and +24V	
Protection against polarity inversions	Using a reverse diode on the supply	
Power of a lamp with filament	10 W (max)	

(*) Utime is the maximum voltage applicable to the module for 1 hour in a 24 hour period of operation.

Monitoring Sensor/Pre-sensor Voltage

General

The supply for the actuators / pre-actuators is monitored by the module to signal to the processor any malfunction, which could lead to incorrect working order.

Table of characteristics:

Electrical characteristics	Symbol	Value	Units
Voltage for OK state	Uok	> 18	V
Voltage for faulty state	Udef	< 14	V
Immunity OK --> Error	Im.off	> 1	ms
Immunity error --> OK	Im.on	> 1	ms
Inclusion of error	Toff	< 10	ms
Inclusion of non-error	Ton	< 10	ms

Characteristics of the Variable Speed Controller Inputs

General

The auxiliary inputs of the variable speed controller are supplied by the same supply as the auxiliary input/outputs. This is not monitored by the module, but any loss in voltage less than 5 V on a CTRL_VAR input can signal to the processor a fault with the variable speed controller.

Illustration:

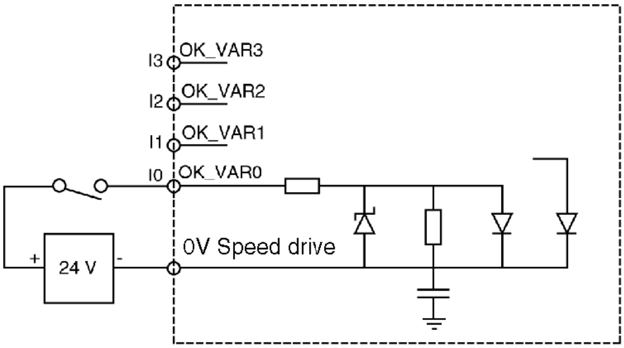


Table of Characteristics

Table of electrical characteristics:

Electrical characteristics	Symbol	Value	Units
Nominal voltage	One	24	V
Voltage limits (1) (ripple included)	U1 Utime (*)	19 to 30 34	V V
Nominal current	In	8	mA
Input impedance (at Un)	Re	3	kΩ
Voltage for "OK" state	Uon	≥11	V
Current at Uon (11 V)	Ion	> 3,5	mA
Voltage for "Error" state	Uoff	< 5	V
Current at "Error" state	Ioff	< 1,5	mA
Immunity OK --> Error	toff	1 to 4	ms
Immunity at Error --> OK	ton	1 to 4	ms
Dielectric rigidity with the ground connection	1500 Veff 50/60 Hz per min		
IEC 1131 compatibility with sensors	Type 1		
Logic type	Positive (sink)		

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

Characteristics of the Relay Outputs

Illustration

Each channel has a relay output.

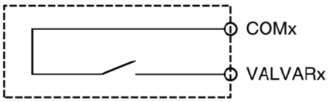


Table of Characteristics

This table describes the electrical characteristics:

Electrical characteristics	Value	Units
Direct voltage used	5 to 30	V
Switched current permitted in direct 30 V on resistive load	200	mA
Minimum permitted load	1V/1mA	
Switching time	< 5	ms
Dielectric rigidity: ● between contacts and between channels ● between contacts and ground connection	300 VAC per min 1000 VAC per min	

Section 7.2
View of the module status

Module Display

General

The TSX CAY 2•/4• and 33 modules are provided with LEDs, used to display the state of the modules and channels.

- Module state LEDs (RUN, ERR, I/O)
  Three LEDs located on the front panel of the module provide information about the module’s operation through their state (LED off, blinking or lit):
  - RUN LED: indicates the operating state of the module,
  - ERR LED: indicates an internal module error,
  - I/O LED: indicates an external error.
- Channel state LEDs (CH.)
  The TSX CAY 2•/4• and 33 modules have 2, 3 or 4 LEDs, which are used to display and diagnose the state of each channel. These LEDs are green.

Diagnostic Table

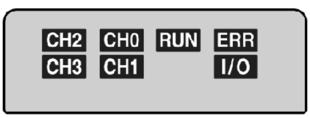
This table shows the diagnostics of the module according to the state of the LEDs:

	Lit ●	Blinking ⊗	Off ○
RUN	Module normal	/	Module switched off or experiencing a fault
ERR	Internal module error: module has broken down.	Communication error Application missing, invalid or experiencing a fault during execution	No error.
I/O	External module error: ● wiring fault ● Encoder supply and 10/30 V supply fault ● absolute encoder error (*)	/	No error.
CH **TSX CAY 2•** CH0 and CH1 **TSX CTY 4•/33** CH0, CH1, CH2, CH3.	The channel is operational.	The channel is not functioning correctly due to: ● an external fault ● a communication error ● a processing error	Channel inoperative. The channel is not configured, or is badly configured.

(*) application fault:

- configuration declined,
- SMOVE function declined.

Illustration of module LEDs:



Part III
Independent Axes

Subject of this Part

This part introduces the TSX CAY modules and describes how to set up servo drive axis control with these modules.

What Is in This Part?

This part contains the following chapters:

Chapter 8
Programming Axis Control

Subject of this Section

This section describes the programming principle for the different operating modes: description of the main instructions and operating modes.

What Is in This Chapter?

This chapter contains the following topics:

Programming an Independent Axis

Introduction

Each axis control module channel (axis) is programmed by using:

- the **SMOVE function** for movements in automatic mode,
- the **bit objects** (%I and %Q) and **words** (%IW, %QW and %MW), associated with the module to be defined:
  - selection of operating modes,
  - movement commands, except for automatic mode,
  - monitoring of the axis and module operating state.

Bit Objects and Words

The bit objects and words can be accessed by their address or their symbol.The symbols are defined in the variable editor, which proposes a symbol name for each object by default.

Operating Modes

At a Glance

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

Operating mode	Description
Automatic (AUTO)	This mode is used for the execution of movement commands controlled by the SMOVE functions.
Manual (MANU)	This mode is used to control the moving part visually from a front panel or human machine interface terminal. The commands can be accessed by the %Q output bits.
Loop control disabled (DIRDRIVE)	The output behaves like a digital / analog converter in this mode. The control loop is inoperative. During adjustment this mode is used to analyze the behavior of the axis independently of the control loop.
Measurement (OFF)	In this mode, the channel does not check the moving part. It only returns information on position and current speeds. This mode is forced at start-up if the axis is configured and not faulty.

Mode Selection

Modes are selected:

- using the MOD_SELECT (%QWr.m.c.0) word, or
- by the debugging screen selector.

The following table indicates the selected mode, according to the %QWr.m.c.0 word value:

Value	Selected mode	Description
0	OFF	Measurement mode, analog output inhibition.
1	DIRDRIVE	Loop control disabled mode.
2	MANU	Manual mode.
3	AUTO	Automatic mode.

For all other values of MOD-SELECT, the OFF mode is selected.

Changing Mode During a Movement

Changing mode while a movement is in progress (DONE bit: % Ir.m.c.1 at 1) stops the moving part. When the moving part is actually stopped (NOMOTION bit: % Ir.m.c.8 at 1) the new operating mode is then activated.

NOTE: Only commands concerning the current mode are examined. The other commands are ignored (except when an SMOVE function is performed in manual mode).

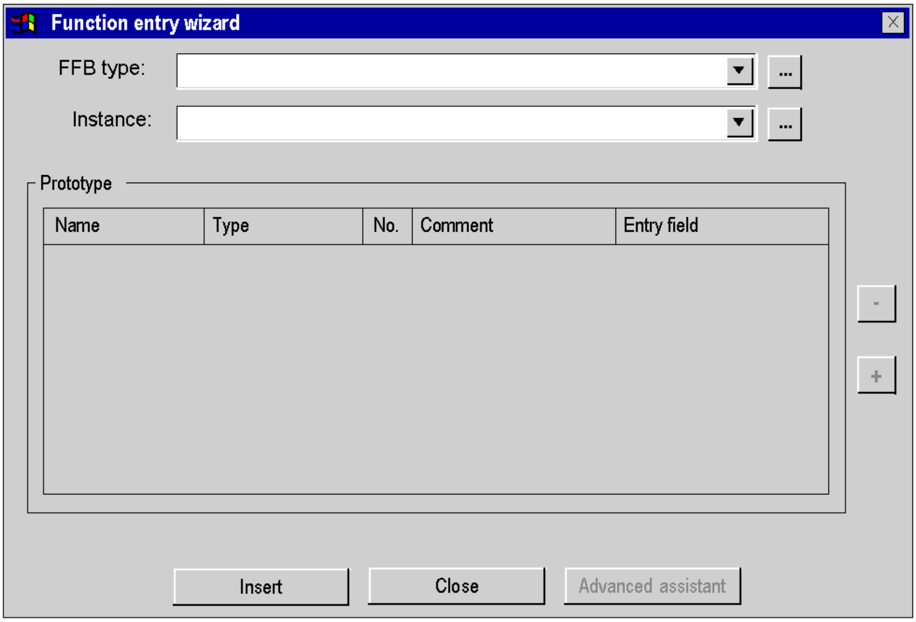
Programming the SMOVE Function (in Automatic Mode)

At a Glance

A SMOVE function can be programmed in all programming modules in ladder language (using an operate block), instruction list language (between square brackets) or in structured text language. In all cases, the syntax stays the same.

Assisted Entry Screen

You can enter the SMOVE function directly or using the entry help screen:



Assisted Entry

From within the selected program editor, proceed as follows:

Step	Action
1	Right click in the editor where you want to enter the function and select **FFB entry assistant...**. The entry window appears.
2	Enter **SMOVE**.
3	Press the **Details** button and fill in the different fields that are offered. Function variables can also be entered directly into the parameters entry zone.
4	Confirm with **OK** or **Enter**. The function is then displayed.

Entering SMOVE Function Parameters

At a Glance

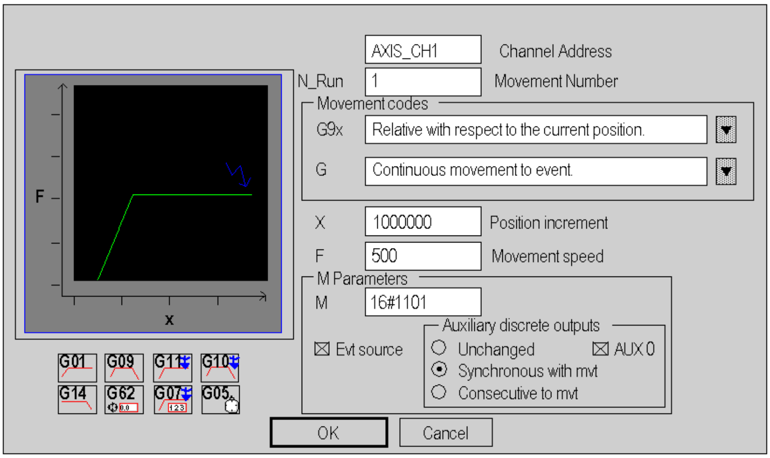
A movement command is programmed by an SMOVE function using the following syntax:

`SMOVE (AXIS_CH1,N_Run,G9x,G,X,F,M)`

The **Details** screen will assist you in entering each parameter.

SMOVE Function Details Screen

The SMOVE function details screen is as follows:



Entry fields (for SMOVE function parameters) are as follows:

Parameter	Description
`AXIS_CH1`	IODDT type variable corresponding to channel 1 on which the function must operate. Example: `AXIS_CH1` of type `T_AXIS_STD`
`N_Run`	Movement number.
`G9x`	Movement type.
`G`	Instruction code.
`X`	Coordinate of target position.
`F`	Speed of moving part.
`M`	Event processing, auxiliary discrete output associated with channel.

Description of SMOVE Function Parameters

At a Glance

The following parameters must be entered to program a movement function:

`SMOVE (AXIS_CH1,N_Run,G9x,G,X,F,M)`

IODDT

`AXIS_CH1` is an IODDT type variable corresponding to channel 1 of the axis control module on which the function must be applied. `AXIS_CH1` can be, for example, an IODDT of type `T_AXIS_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 debugging mode, this number is used to determine the current movement.

Movement Type

G9x defines the movement type:

Code	Movement type
90	**Absolute** movement.
91	Relative movement **with respect to the current position**.
98	Relative movement **with respect to stored PREF1 position**. Instruction code G07 is used to store PREF1 position.
60	**Absolute** movement for set direction (infinite type machine only).
68	Relative movement **with respect to PREF** in set direction (infinite type machine only).

To choose the movement type, use the scroll button on the right of the **G9x** field, or enter the code directly with a "direct entry" (without going to the **Details** screen).

Instruction Code

G defines the instruction code *(see page 150)* for the SMOVE function.

Coordinates for Reach Position

X defines the coordinates of the reach position or where the moving part must move (in the case of non-stop movement). This position can be:

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

This value is expressed as a unit defined by the **Length Units** configuration parameter (for example; micron).

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

Moving Part Movement Speed

F defines the speed at which the moving part travels. This speed can be:

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

The speed unit is deduced from the selected length unit:

`Speed = u x 1000 / min` where u = length unit chosen.

For example, if the micron is chosen as the length unit, the speed unit will be:

`micron x 1000 / min -> mm / min`

M Parameter

M defines a word which codes nibbles (in hexadecimal):

- activation or inactivation of the event processing application trigger for G10, G11, G05 and G07 instructions:
  - bit 12 set to 1: activation,
  - bit 12 set to 0: inactivation.
- setting to 0 or 1 of auxiliary discrete output associated with channel:
  - Nibble 2: activating moment
    0 = unchanged (no modification of output),
    1 = synchronous with movement (assignment of output at beginning of execution of instruction),
    2 = consecutive to movement (assignment of output at end of execution of instruction)
  - Nibble 0: auxiliary output status during execution of G01, G09, G10 and G11 instructions
    0 = output set to 0 (AUX0 box not checked),
    1 = output set to 1 (AUX0 box checked).
- event type expected by instruction G05:
  - Bit 13
    0 = wait for time out or event,
    1 = wait modulo crossing number.

For example:

Byte	3	2	1	0
16#				

- 16#0101 = trigger for event processing application not activated, and auxiliary output set to 1 when SMOVE command is executed.
- 16#1200 = trigger for event processing application is activated, and auxiliary output set to 0 when SMOVE command has finished being executed.

NOTE: Coding is automatically completed in the **M** field on the **Details** screen, when the choices have been made using the check-boxes and buttons offered by the screen.

Instruction Codes for SMOVE Function

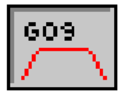
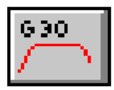
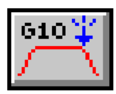
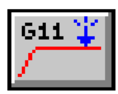
At a Glance

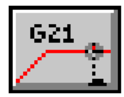
The **G** parameter defines the instruction code.

To choose the instruction code, either use the scroll button on the right of the **G** field, or press on the icon which corresponds to the movement. You can also enter the code directly with a "direct entry" (without going via the **Details** screen).

Instruction Codes List

The instruction codes which can be chosen on the **Details** screen are as follows:

Instruction code	Meaning	Icon
09	Movement on position with stop *(see page 158)*	
01	Movement on position without stop *(see page 157)*	
32	Preparation of machining command *(see page 161)*	
30	Simple machining *(see page 161)*	
10	Movement as far as event with stop *(see page 159)*	
11	Movement as far as event without stop *(see page 159)*	
14	Reference point *(see page 164)*	

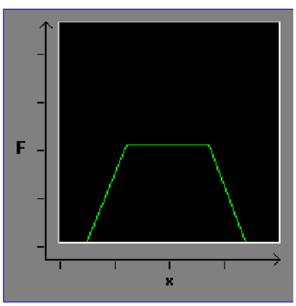
Instruction code	Meaning	Icon
62	Forced reference point *(see page 169)*	
05	Await event *(see page 170)*	
07	Storing position on event *(see page 171)*	
21 (1)	Unlimited movement with reference point on the fly *(see page 166)*	
04 (1)	Movement stop *(see page 168)*	

(1) With TSX CAY 22 / 42 or TSX CAY 33 module

Imaging on Details Screen

The **Details** screen also displays an image which represents the selected movement.

For example, code G09:



Description of Elementary Movements Using a Limited Machine

At a Glance

3 types of movement category can be programmed:

- movements on a position (instruction codes 01 and 09),
- movements until event detection (instruction codes 10 and 11),
- reference points (instruction code 14).

The reach position and speed must be set while programming. Acceleration parameters (e.g. rectangular, trapezoidal or triangular) are set in the configuration.

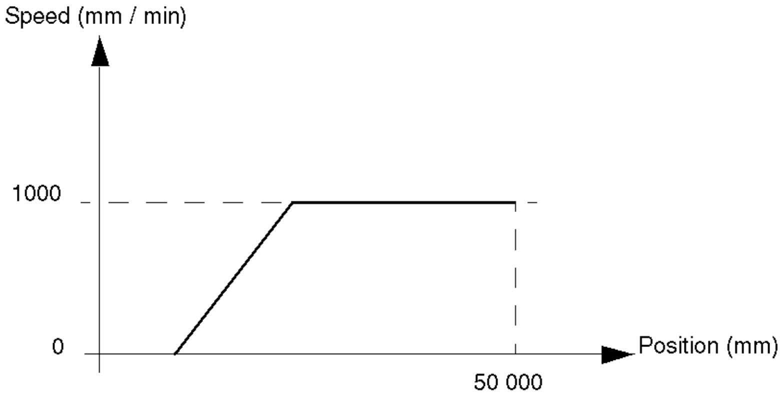
Types of Movement

With a limited machine, types of movement are as follows:

- Absolute in relation to the machine homing point (code 90).

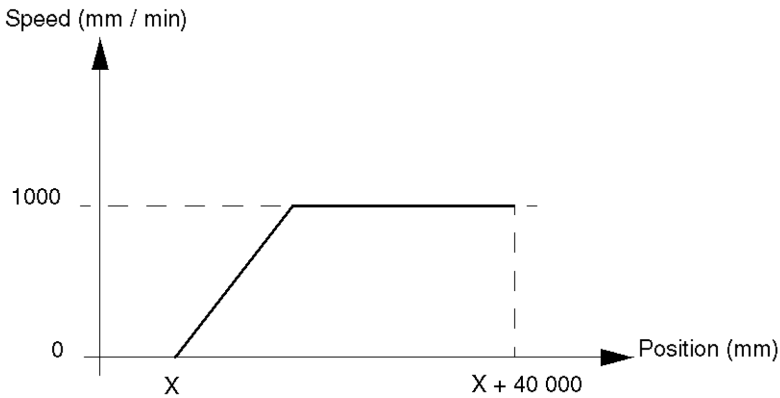
Example: `SMOVE (AXIS_CH0,1,90,01,50000,1000,0)`

`AXIS_CH0` of type `T_AXIS_STD`

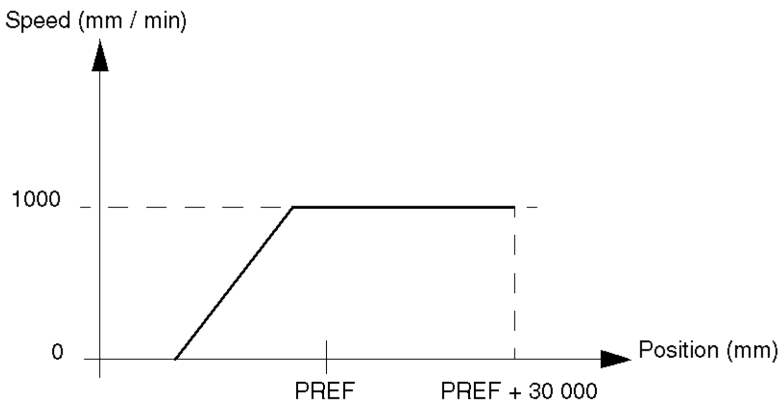


- Relative in relation to current position (code 91).

  Example: `SMOVE (AXIS_CH0,1,91,01,40000,1000,0)`
  `AXIS_CH0` of type `T_AXIS_STD`



- Relative in relation to stored position PREF1 (code 98).



Description of Elementary Movements Using an Infinite Machine

At a Glance

3 types of movement categories can be programmed:

- movements on a position (instruction codes 01 and 09),
- movements until event detection (instruction codes 10 and 11),
- reference points (instruction code 14).

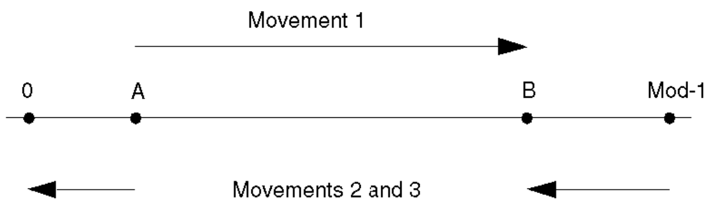
The reach position and speed must be set while programming. Acceleration parameters (e.g. rectangular, trapezoidal or triangular) are set in the configuration.

Types of Movement

Whatever the current position and target may be, it is always possible to reach the objective position in + direction as well as – direction. There are 3 possible ways to go from point A to point B:

- ascending position movement (movement 1),
- descending position movement (movement 2),
- for the shortest movement: the module decides on the direction (movement 3)

The required direction of movement is specified by the speed sign.

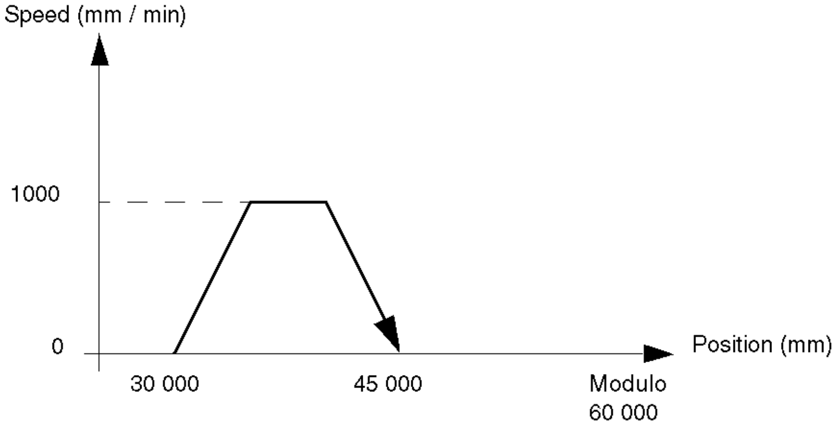


With an infinite machine, movements can be of the following type:

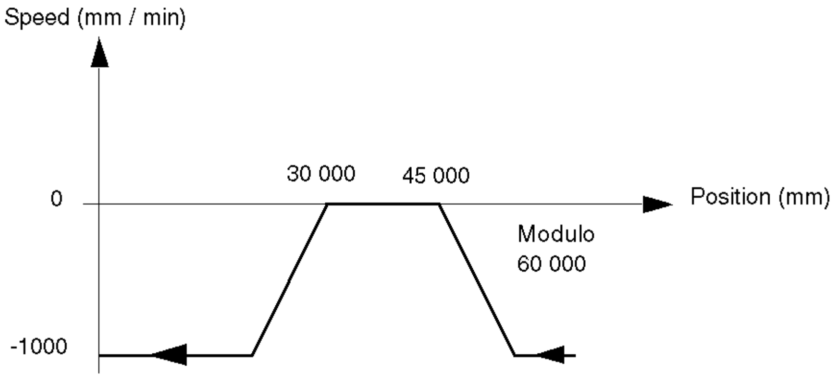
- Shortest movement in relation to the machine homing point (code 90). In this case, the shortest path determines the direction of movement.

  Example: `SMOVE (AXIS_CH0,1,90,09,45000,1000,0)`

  `AXIS_CH0` of type `T_AXIS_STD`



- Set direction movement in relation to the machine homing point (code 60). In this case, the speed sign determines the direction of movement.

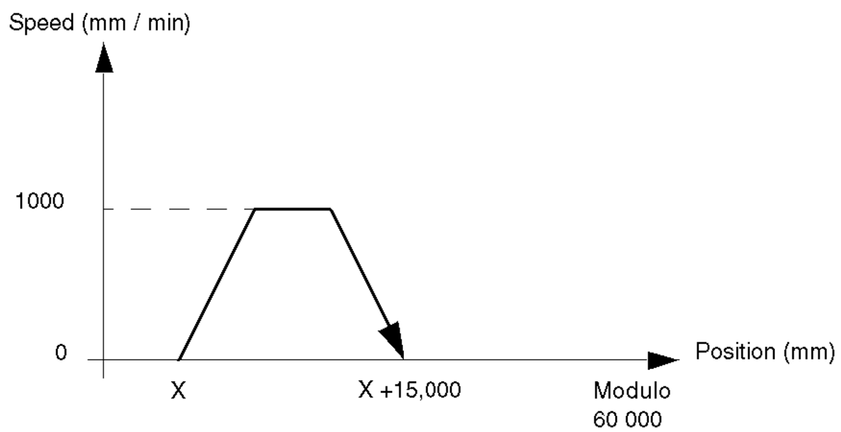


- Shortest movement in relation to stored PREF1 position (code 98).
  For example, SMOVE (AXIS_CH0,1,98,09,45000,1000,0) targets (45000 + PREF1).

- In opposite direction in relation to stored PREF1 position (code 68).
  For example, SMOVE (AXIS_CH0,1,68,09,45000,-1000,0) targets (45000 + PREF1) value while moving in – direction.
  For example, SMOVE (AXIS_CH0,1,68,09,45000,1000,0) targets (45000 + PREF1) value while moving in + direction.
- Relative in relation to current position (code 91). In this case, direction of movement is determined by the X parameter sign (position increment).

Example: `SMOVE (AXIS_CH0,1,91,09,15000,1000,0)`

`AXIS_CH0` of type `T_AXIS_STD`

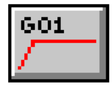


NOTE: The value targeted by G68 or G91 is calculated in relation to the modulo. `(1000 + PREF1) Mod ModuloValue` and `(X + 15000) Mod ModuloValue` are targeted in the examples. `Mod` is the Modulo mathematical operator.
For example, if PREF1 = 40000 and modulo = 60000: 45000 + PREF1 corresponds to 25000.

Programming a Move to Non-stop Position

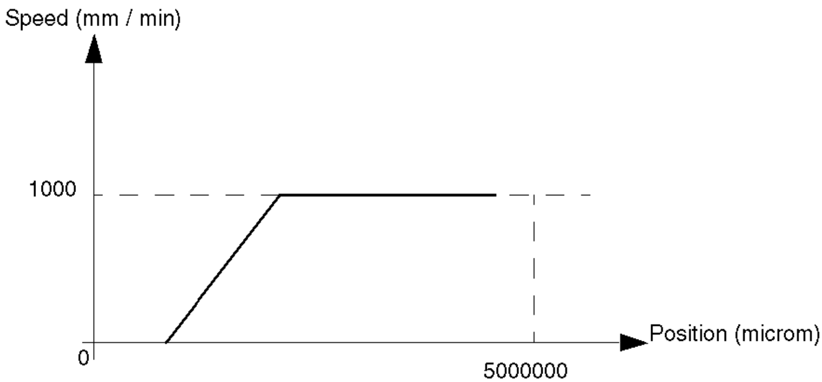
Instruction

The move to non-stop position instruction is as follows:

Instruction	Instruction code	Icon
Moving to non-stop position	01	

Example

`SMOVE (AXIS_CH0,1,90,01,5000000,1000,0) AXIS_CH0` of type `T_AXIS_STD`

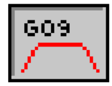


NOTE: If instruction G01 is not followed by another instruction, its behavior depends on the **Sequence control** parameter defined in configuration.

Programming a Move to Position with Stop

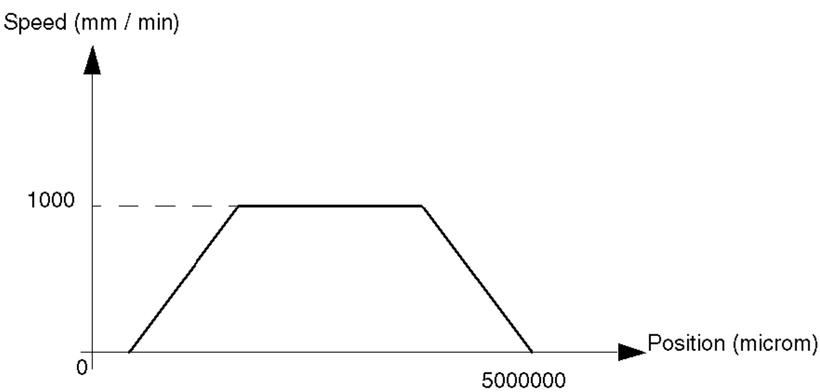
Instruction

The move to position with stop instruction is as follows:

Instruction	Instruction code	Icon
Moving to position with stop	09	

Example

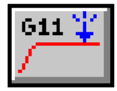
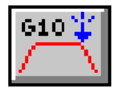
`SMOVE (AXIS_CH0,1,90,09,5000000,1000,0` `AXIS_CH0` of type `T_AXIS_STD`



Programming a Movement Until Event Detection

Instruction

The instruction for movement until event detection is as follows:

Instruction	Instruction code	Icon
Movement until event detection without stop	11	
Movement until event detection with stop	10	

Instructions 11 and 10 are similar to instructions 01 and 09, with command end on detection of an event (or command end on position entered if event has not been detected).

Event

The awaited event can be:

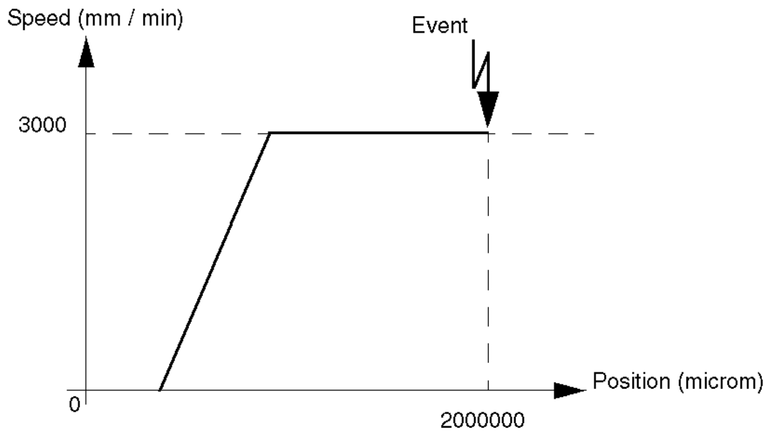
- a rising or falling edge (depending on the choice made in the **Event** field in the configuration screen) on dedicated event input, associated with the channel which controls the axis,
- a rising edge of the EXT_EVT bit (%Qr.m.c.10) generated by the program.

It is mandatory to define the **Position** parameter. If the event is not been detected, the command finishes when this position has been reached.

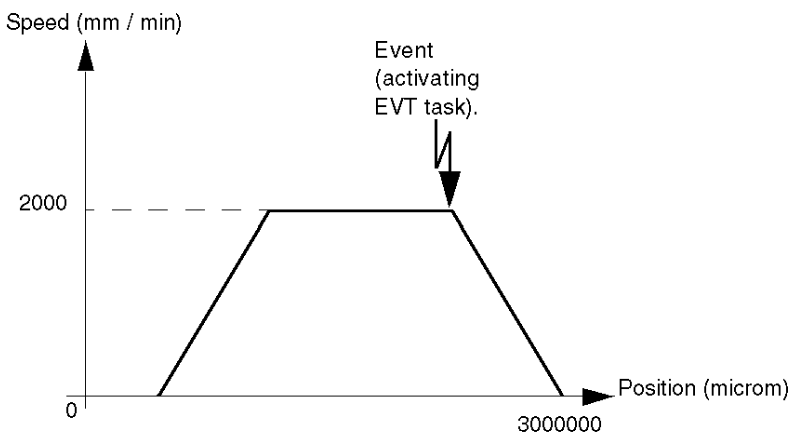
Instructions 11 and 10 can activate an event task on detection of an event, if bit 12 of parameter **M** is set to 1.

Examples

Example 1: `SMOVE (AXIS_CH0,1,90,11,2000000,3000,0) AXIS_CH0 of type T_AXIS_STD`



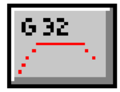
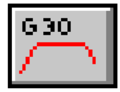
Example 2: `SMOVE (AXIS_CH0,1,90,10,3000000,2000,16#1000) AXIS_CH0 of type T_AXIS_STD`



Programming a Simple Machining Command

Instruction

A simple machining command instruction is as follows:

Instruction	Instruction code	Icon
Preparing simple machining	32	
Executing simple machining	30	

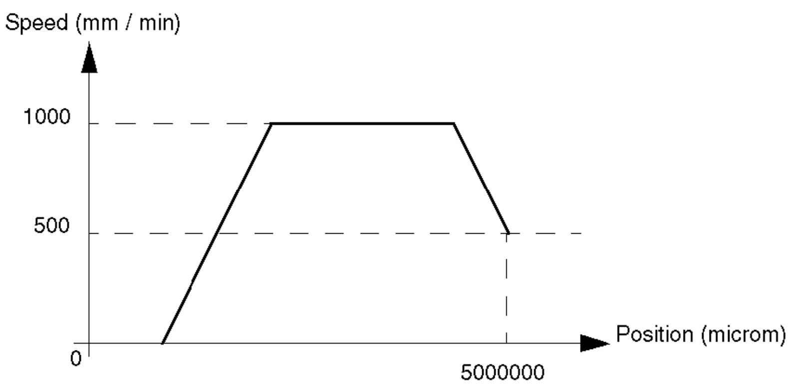
Instruction 32 and 30 are used to create a simple machining profile consisting of:

- an approach speed defined in instruction G32,
- a machining speed and a target position defined in instruction G30.

Example

`SMOVE (AXIS_CH0,1,90,32,0,1000,0) AXIS_CH0` of type `T_AXIS_STD`

`SMOVE (AXIS_CH0,2,90,30,5000000,500,0) AXIS_CH0` of type `T_AXIS_STD`



Programming

- Command G32 is a preparation command. It can be activated in the same PLC cycle as a G30 command, without monitoring the NEXT and DONE bits.
- The reactivating command G32 does not have to be reactivated when the approach speed remains unchanged, as the speed is stored. On the other hand, it is mandatory to send at least one G32 command before executing a G30 command.
- If it is not followed by a movement command, instruction G30 triggers a movement without stop whose behavior is identical to instruction G01. If instruction G30 is not followed by a movement command and if a sequence control is requested, it will cause a command refusal.
- If the moving part is in motion, instruction G30 must not change the direction of movement.
- Instruction G30 is normally followed by a G09 instruction (figure 1). If this sequence causes a change in direction, the process stops and inverts in order to obtain the value G09 (figure 2).

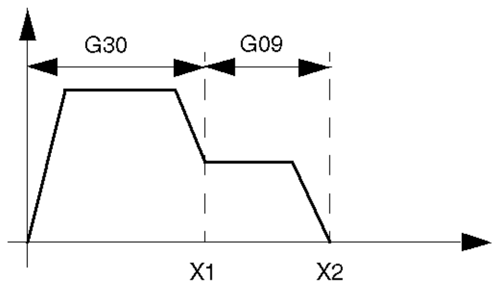


Figure 1

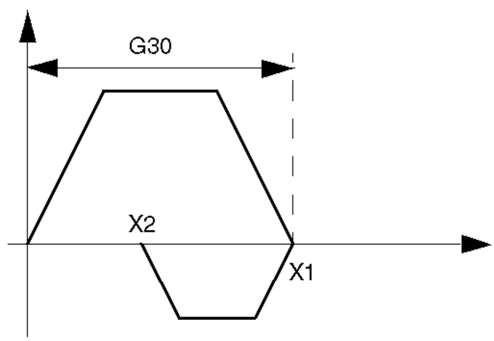
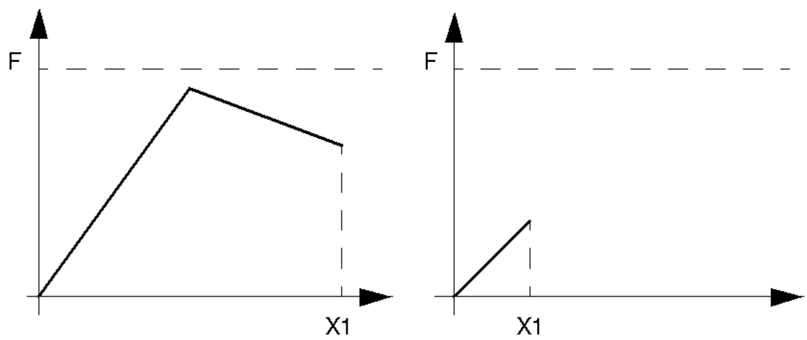


Figure 2

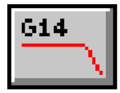
- If the distance to be covered by instruction G30 does not allow the specified speed to be reached, the movement then takes one of the following trajectories:



Programming a Reference Point

Instruction

The instruction for setting a reference point is as follows:

Instruction	Instruction code	Icon
Reference point	14	

The displayed position corresponds to the coordinate to be loaded as the current value when the source is detected.

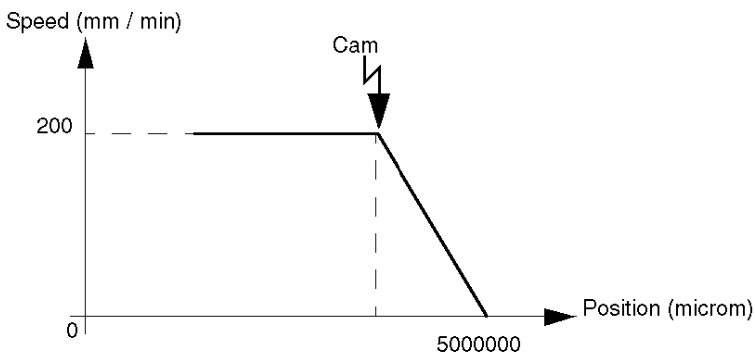
According to the type of reference point chosen, the reference point event is detected either during cam input or during cam and Zero Marker inputs, associated with the controlled axis.

The type of reference point and the direction of movement are defined in the configuration.

Examples

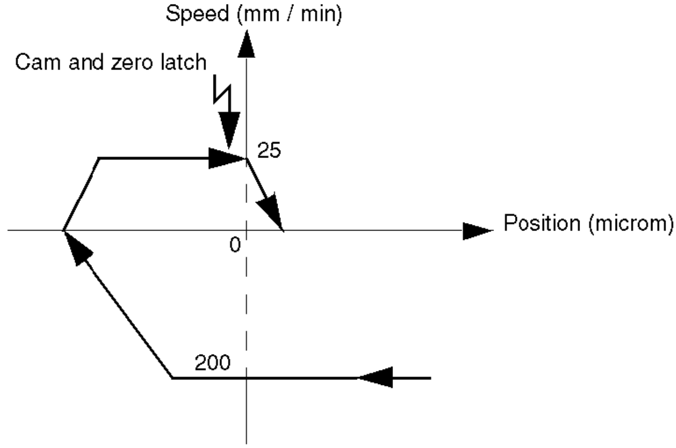
Example 1: `SMOVE (AXIS_CH0,1,90,14,5000000,200,0) AXIS_CH0` of type `T_AXIS_STD`

Type of reference point = short cam, - direction



Example 2: `SMOVE (AXIS_CH0,1,90,14,0,200,0)` `AXIS_CH0` of type `T_AXIS_STD`

Type of reference point = short cam, zero latch, - direction

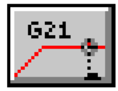


NOTE: The axis is referenced at the start of the execution of the instruction.
The movement must always be an absolute movement (code 90).

Programming a Reference Point on the Fly on Event

Instruction

The instruction for setting a reference point on the fly on event is as follows:

Instruction	Instruction code	Icon
Setting a reference point on the fly on event	21	

The position supplied by the X parameter corresponds to the coordinate to be loaded as the current value when the source is detected.

The direction of movement is defined by the speed sign (the direction defined by the type of source is not taken into account).

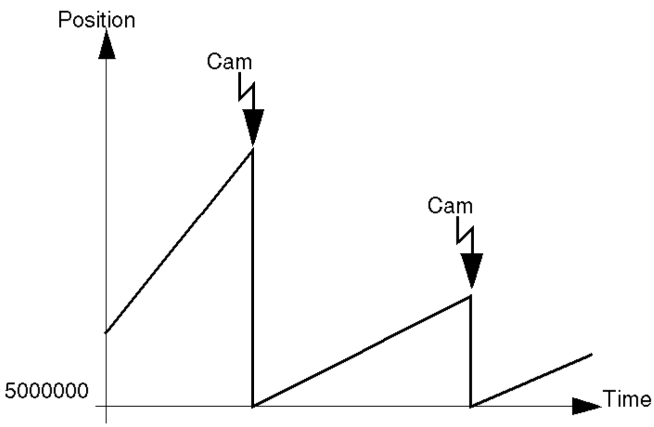
The G21 instruction never finishes of its own accord. A STOP command (%Qr.m.c.15) must be sent to terminate this instruction.

According to the type of reference point chosen, the reference point event is detected either during cam input or during cam and Zero Marker inputs, associated with the controlled axis.

The type of reference point and the direction of movement are defined in the configuration.

Example

`SMOVE (AXIS_CH0,1,60,21,5000000,200,0) AXIS_CH0` of type `T_AXIS_STD`



Conditions of Execution

The execution conditions are as follows:

- the encoder must be an incremental encoder,
- the recalibration function is inactive,
- the type of movement uses a G60 instruction code.

How to Program a Movement Stop

Instruction

The instruction for a movement stop is as follows:

Instruction	Instruction code	Icon
Movement stop	04	

This instruction is used to stop G01, G30 and G11 non-stop movements as quickly as possible. It is equivalent to a STOP order.

There are no parameters associated with instruction G04.

Example

Stopping a G01 movement after a time period of 10s:

```
SMOVE (AXIS_CH0,1,91,01,100000,1500,16#0000)
SMOVE (AXIS_CH0,2,90,05,0,10000,16#0000)
SMOVE (AXIS_CH0,3,90,04,0,0,16#0000)
```

`AXIS_CHO` of type `T_AXIS_STD`

Comment:

Unlike the STOP instruction, a movement stop using the instruction code 04 does not empty the buffers.

Programming a Forced Reference Point

Instruction

The instruction for setting a forced reference point is as follows:

Instruction	Instruction code	Icon
Forced reference point	62	

This command sets a forced reference point, without moving the moving part.

The current position value is forced to the value entered in the position X parameter.

Example

`SMOVE (AXIS_CH0,1,90,62,100000,0,0)` `AXIS_CH0` of type `T_AXIS_STD`

When this instruction is carried out, the current position is forced to 100000.

Notes

- Whatever the state of the axis (referenced or not referenced), if the G62 command is accepted, it references the axis once the command is executed.
- The G62 command is only accepted if the moving part is stationary: NOMOTION bit (%Ir.m.c.8) set to 1.

Programming an Await Event

Instruction

The await event instruction is as follows:

Instruction	Instruction code	Icon
Await event	05	

This instruction is used to await an event within a Time Out (in ms), defined in the parameter F. If an event does not appear within this Time Out period, the await command is then deactivated. If F parameter is defined at 0, the wait is without a time limit.

For an Infinite Machine

With an infinite machine, the G05 instruction is also used to await the crossing of a number of modulo.

The choice is determined by the bit 13 value of parameter M:

- bit 13 = 0, await event,
- bit 13 = 1, await modulo number.

Event Associated with the Command

The event associated with command G05 can be:

- a rising or falling edge (depending on the choice made in the **Event** field in the configuration screen) on dedicated event input, associated with the channel which controls the axis,
- a rising edge of the EXT_EVT bit (%Qr.m.c.10) generated by the program,
- a number of modulo crossings (for an infinite machine). For example, awaiting 10 modulo crossings with activation of the event task:
  `SMOVE (AXIS_CH0,1,90,05,0,10,16#2000) AXIS_CH0` of type `T_AXIS_STD`

Event Task

The G05 instruction can activate an event task on detection of an event, if bit 12 of parameter M is set at 1.

The TO_G05 bit is set to 1 when the Time Out has elapsed without an event being detected. For example, an event waits with a Time Out period of 1.5 s and with activation of the event task:

`SMOVE (AXIS_CH0,1,90,05,0,1500,16#1000) AXIS_CH0` of type `T_AXIS_STD`

Programming Storage of Current Position on Event

Instruction

The instruction for storage of current position on event is as follows:

Instruction	Instruction code	Icon
Storage of current position on event	07	

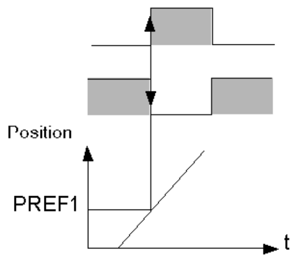
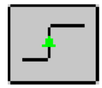
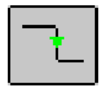
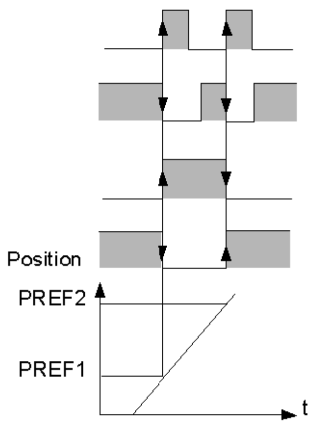
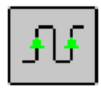
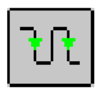
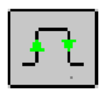
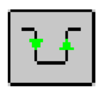
After execution of this instruction, a change of state awaited on the event input of the axis control module causes the current position to be stored.

During configuration and in the position parameter X, you can choose to store one or two positions (PREF1 and PREF2):

- If the **without measurement** option is chosen during configuration, only PREF1 is stored (parameter X must be equal to 1).
- If the **with measurement** option is chosen during configuration:
  - if X = 1, event processing will be activated on storage of position PREF1,
  - if X = 2, event processing will be activated after storage of position PREF1, then PREF2.

Storage of Current Position

The following table illustrates storage of current position, according to the choices made during configuration:

Event type on the EVT input	Diagram	Choices made during configuration	
Rising edge			Rising edge and PREF1
Falling edge			Falling edge and PREF1
Rising edge			Rising edge and PREF1, then rising edge and PREF2
Falling edge			Falling edge and PREF1, then falling edge and PREF2
Rising and falling edge			Rising edge and PREF1, then falling edge and PREF2
Falling and rising edge			Falling edge and PREF1, then rising edge and PREF2

Event processing is activated on detection of an event, if bit 12 of parameter M is set at 1.

The program passes directly on to the following instruction. The PREF1 (%IDr.m.c.9) and PREF2 (%IDr.m.c.11) words are only refreshed if an event task is triggered by the awaited event.

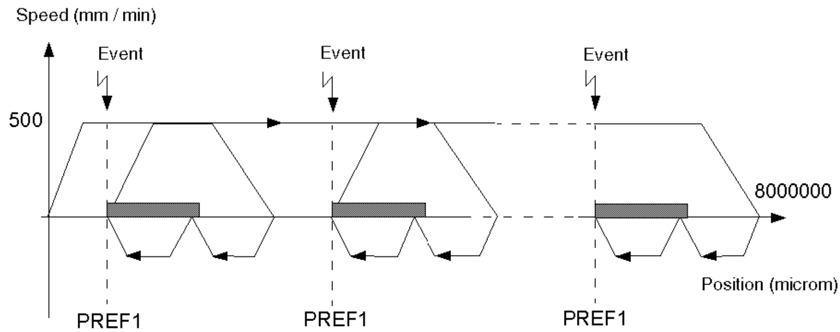
The performance of the G07 instruction, or in other words the measurement / event delay is **immediate** for an incremental encoder and **less than or equal to 400 microseconds** for an absolute encoder.

Example of How to Use an Indexed Position

An indexed position is used to resolve repetitive movements. For example, let us assume that the elementary movement sequence below has to be performed 9 times:

- movement until the edge of the part is detected (2),
- movement up to position 2000 in relation to the edge of the part (3),
- movement up to position 1000 in relation to the edge of the part (4),
- movement up to the edge of the part (5).

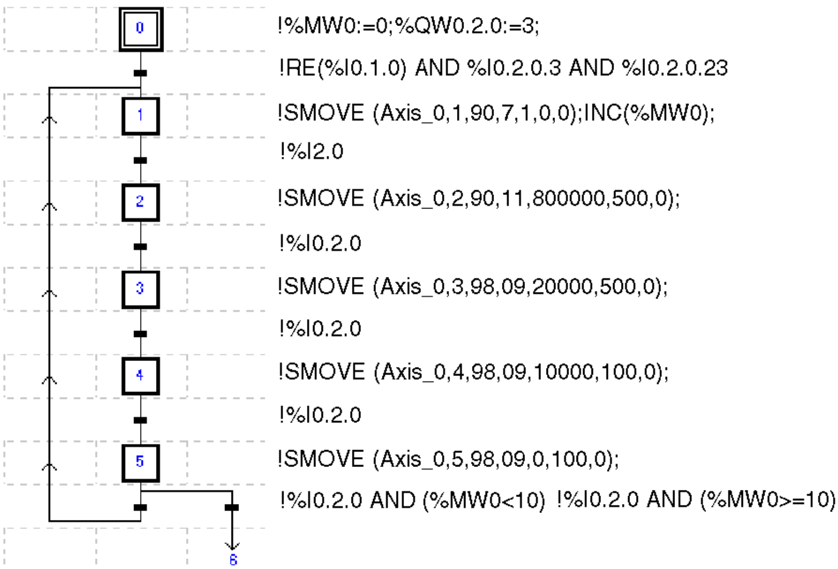
In this example, it is assumed that the reference point is already set and the moving part is in the source position.



NOTE: The sequence of elementary movements is shown in bold on the curve. The numbers indicated correspond to the numbers of program steps included in the SMOVE function.

Program

The elementary movement sequence program is as follows:



NOTE: All the actions have to have been programmed on activation.

Sequencing Movement Commands

Producing a Trajectory

A trajectory is produced by programming a succession of elementary movement instructions (SMOVE function).

Each elementary SMOVE command must only be performed once. It must be programmed:

- in Grafcet: in one programmed step, on activation or deactivation,
- in structured text or ladder language, on the rising edge of a bit.

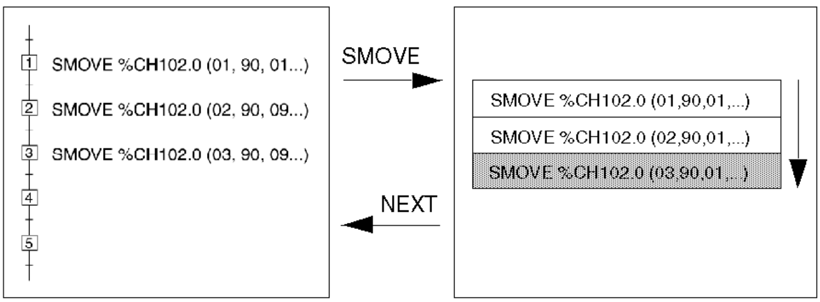
A report on function execution is provided by the module using the NEXT and DONE bits.

Buffer Memory

The TSX CAY module features a mechanism which is used to sequence movement commands.

Each axis of the TSX CAY module has a buffer memory, which can receive 2 movement commands, in addition to the one it is in the process of executing. Thus when a movement in progress has finished, it proceeds directly to the first command present in the buffer memory.

Command sequence:



Sequencing 2 Commands

Sequencing between 2 movement commands is as follows :

- instantaneously if the first movement is without stop,
- as soon as the moving part is in the target window or after time delay TSTOP has elapsed (which is defined in the stop control on parameter adjustment screen) if the first movement is with a stop.

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

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

Bits Associated with a Sequencing Mechanism

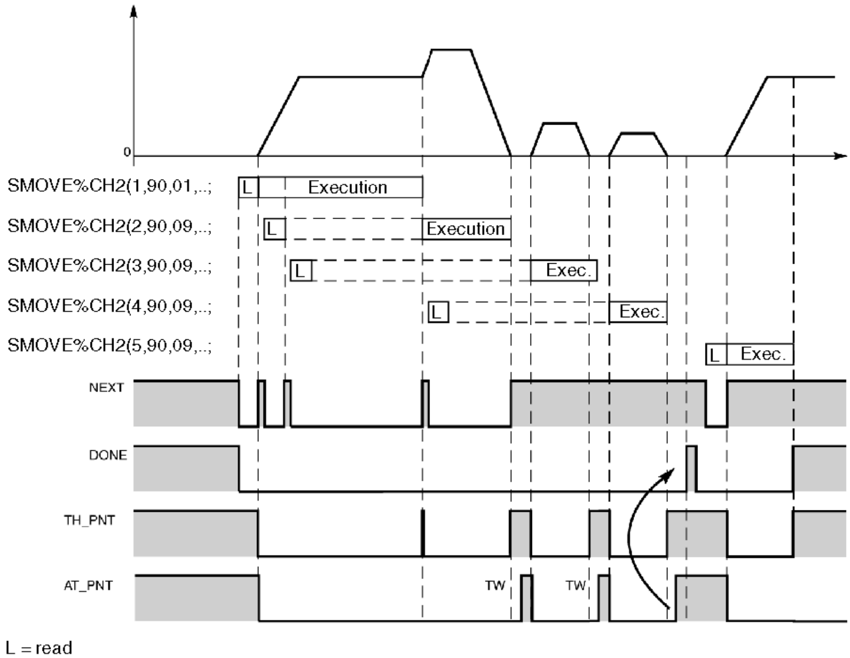
The bits associated with the sequencing mechanism are as follows:

Addressing	Description
NEXT (%Ir.m.c.0)	Indicates to the user program that the module is ready to receive the next movement command.
DONE (%Ir.m.c.1)	Indicates the command in progress has finished being executed and there are no new commands in the buffer memory.
TH_PNT (%Ir.m.c.10)	Indicates that the position target value has been reached.
AT_PNT (%Ir.m.c.9)	Indicates that the moving part has reached the target point: ● by an INC command in manual mode, ● by a movement command with stop in automatic mode, For a movement with stop, this bit is set to 1 as soon as the moving part enters the target window. This bit is not set to 1, following a JOG command, reference point or STOP during a movement.

NOTE: The program must always test the NEXT bit, or the DONE bit before executing an SMOVE command, except in the case of a G32 command which can be immediately followed by another command.
The SYNC_N_RUN (%IWr.m.c.8) word periodically supplies information on the number of steps in progress, in order to carry out movement sequencing.

Example

The diagram below is a timing diagram for a sequence:



For a movement with stop: DONE is set to 1 when NOMOTION is set to 1 and when there is available buffer memory.

For a movement without stop: DONE is set to 1 when TH_PNT is set to 1 and when there is available buffer memory.

NOTE: This simplified diagram does not take following error into account.

Programming the Recalibration on the Fly Function

At a Glance

This function, available with an incremental encoder, updates the current position of the moving part each time the recalibration on the fly input detects a rising edge in a positive direction or a falling edge in a negative direction.

It has been specifically adapted for axes where the moving part is susceptible to slip, or in other words where the position value no longer reflects the true position.

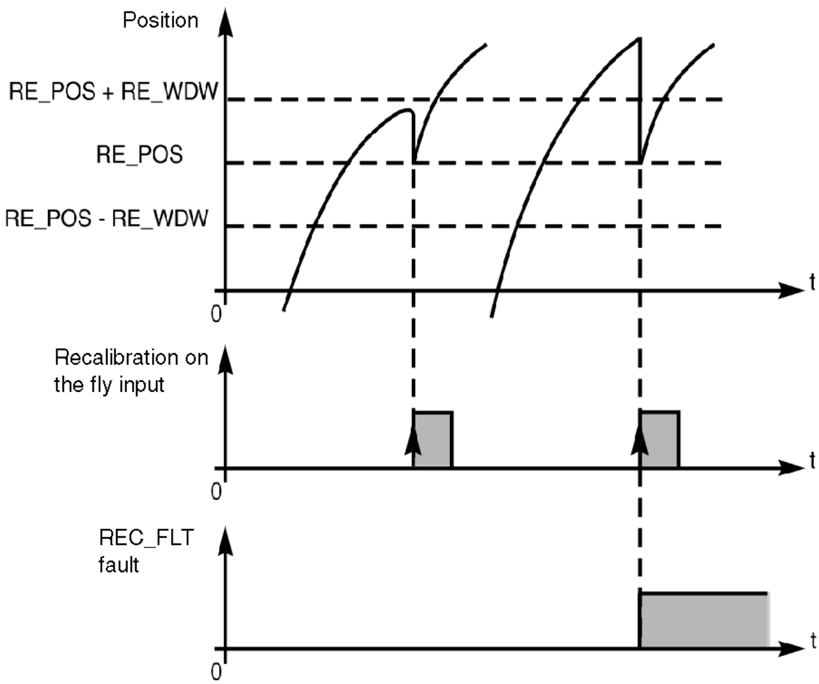
This function is confirmed in the configuration screen.

Recalibration on the Fly Function

When the event takes place, the axis control module preselects the current value to the RE_POS value and compares the current position with the recalibration value RE_POS (%MDr.m.c.43), defined in the recalibration screen (or by the program).

- If the comparison indicates that the current value is outside of the tolerances defined by RE_WDW (%MDr.m.c.51), a fault is signaled: REC_FLT bit (%MWr.m.c.3.12).

The moving part is not stopped:



Execution Condition

Recalibration on the fly function is enabled:

- referenced axis,
- in automatic, manual or loop control disabled mode,
- in Drv_Off mode, with a TSX CAY 22 / 42 module or TSX CAY 33 module.

NOTE: The value of the RE_WDW parameter must be much lower than the DMAX1 following error fault threshold.

Movement Slaved to Another TSX CAYx1 Axis

At a Glance

This function is used to slave the position of an axis (called the slave axis) to the position of another axis from the same module (called the master axis).

The master axis is always the axis 0. A 2 axis module can have one master axis and one slave axis. A 4 axis module can have one master axis and up to 3 slave axes.

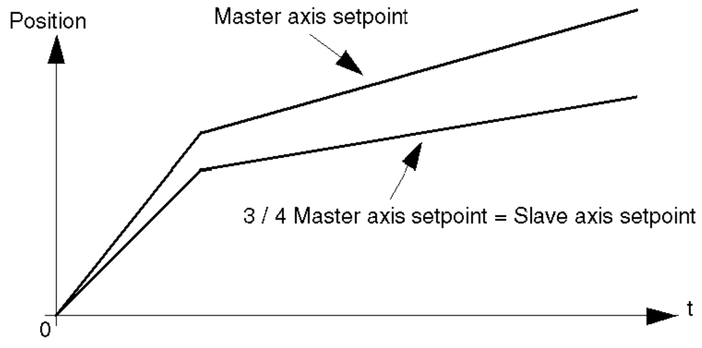
The position slave function is confirmed in the configuration screen. At programming level, the slave axis is slaved to the master axis when the SLAVE bit (%Qr.m.c.17) of this axis is set to 1. The IN_SLAVE bit (%Ir.m.c.36) indicates that the slave axis is operating in tracking mode.

Slaving

An axis is slaved either to the measured position or to the master axis position setpoint (choice defined in configuration). A RATIO1 / RATIO2 ratio is applied in order to obtain the final setpoint. These 2 parameters are defined during configuration.

In order to indicate that the slave axis is tracking the master axis correctly, the AT_PNT (%Ir.m.c.9) bit of the slave axis is at 1 when the tracking error of the latter is lower than DMAX2.

For example, Ratio = 3/4, with slaving to the master axis setpoint.



Execution Condition

- The master axis is configured.
- The slave axis is referenced.
- No blocking fault is detected.
- The slave axis is in automatic mode.
- The master axis is in automatic or manual mode.

If the calculated setpoint overshoots the axis soft stops, the moving part stops and the command is refused.

Important

- For the calculated slave axis position setpoint to be valid, you must make sure that the slave axis is already at the Master x RATIO position before switching this axis to slave mode.
- Reference point commands on the master axis must be avoided when there are slave axes (risk of tracking error fault on the slave axis).
- In Tracking mode:
  - the TH_PNT, NEXT information bits are not managed,
  - the PAUSE command is not active,
  - CMV modifications are not taken into account (CMV = 1000).

Movement Slaved to Another TSX CAYx2 Axis

At a Glance

This function is used to slave the position of an axis (called the slave axis) to the position of another axis from the same module (called the master axis).

The master axis is always the axis 0. A 2 axis module can have one master axis and one slave axis. A 4 axis module can have one master axis and up to 3 slave axes.

This position slave function is confirmed in the configuration screen. At programming level, the slave axis is slaved to the master axis when the SLAVE bit (%Qr.m.c.17) of this axis is set at 1. The IN_SLAVE bit (%Ir.m.c.36) indicates that the slave axis is operating in tracking mode.

Slaving

An axis is slaved either to the measured position or to the master axis position setpoint (choice defined in the configuration screen). A RATIO1 / RATIO2 ratio and an Offset are applied to obtain the final setpoint. These 3 parameters are defined in the adjustment screen.

The slave axis is related to the master axis in the following way:

SlavePositionSetpoint = MasterPosition x (Ratio1 / Ratio2) + SlaveOffset

In order to indicate that the slave axis is tracking the master axis correctly, the AT_PNT (%Ir.m.c.9) bit from the slave axis is set at 1 when the slave has caught up with the master and remains close to it (tracking error of the latter is lower than DMAX2) for more than TSTOP ms

Differences from the TSX CAY •1 Module

The tracking function of a TSX CAY•2 module differs from the TSX CAY •1 in the following ways:

- the ratio can be modified from the application or by the P_Unit in recalibration mode (for the TSX CAY •1 module the ratio is set during configuration,
- the offset enables to slave axis to slave itself to the master axis, whatever the master's position. This allows object tracking applications to be set up where a tool-carrying axis has to be slaved to a permanently moving axis (conveyor) for the transportation of objects (glue application, etc.).

The offset value can be modified from the application or from the P_Unit software in recalibration mode. The TSX CAY •2 module provides an alignment or locking device in order to avoid a 'surprise' during the switch to slave mode.

The function calculates the offset as follows:

SlavePosition = MasterPosition X Ratio + Offset

The configuration parameter **Automatic offset** is used to select the operating mode. The signed ratio value advances from 0.01 and 100.

Parameters Associated with the Master Mode - Slave

The parameters of the master – slave mode are as follows:

- Ratio1 (%MWr.m.c.29) and Ratio2 (%MWr.m.c.30) which determine the value of the master – slave ratio,
- Slave_Off (%MDr.m.c.55): offset value when the Automatic offset option has not been selected during configuration,
- InternalSlaveOffset: offset value (calculated by the module and not accessible by the user) when the Automatic offset option has been selected during configuration,

Execution Condition

- The master axis is configured in automatic or manual mode in the framework of a setpoint follower.
- The slave axis is referenced in automatic mode.
- No blocking fault is detected.
- The master axis must be referenced within the framework of a measurement follower (the master being situated indifferently in one of the 4 modes).

If the calculated setpoint overshoots the axis soft stops, the moving part stops and the command is refused.

SlavePositionSetpoint = MasterPosition (1) x Ratio1 / Ratio2 + Slave_Off (or **InternalSlaveOffset**)

(1) according to configuration.

For the DRIVE_OFF Mode

This mode is used to slave a slave axis to a master axis. In this case, the slave is declared a measurement follower.

Meaning of the AT_PNT Bit and DMAX2 Parameter

The DMAX2 parameter defines the precision threshold.

This value is particularly useful in the object tracking applications where the slave axis goes into a recovery phase before satisfying the condition:

(MasterPosition x Ratio + Offset) - DMAX2 < or = (SlavePosition) < or = MasterPosition x Ratio + Offset) + DMAX2

As soon as the condition is satisfied over a period at least equal to the value of the T_STOP parameter, the AT_PNT bit takes the value 1, to signify that the slave axis has caught up.

Important

- For the calculated slave axis position setpoint to be valid, you must make sure that the slave axis is already at the Master x RATIO position before switching this axis to slave mode.
- Reference point commands on the master axis must be avoided when there are slave axes (risk of tracking error fault on the slave axis).
- In Tracking mode:
  - the TH_PNT, NEXT information bits are not managed,
  - the PAUSE command is not active,
  - CMV modifications are not taken into account (CMV = 1000).
- When a slave axis is slaved to a master axis, the slave tries to catch up with the master by the shortest possible route. For this reason, the tracking error must always be less than half the slave modulo value.
- During configuration you can specify that the link between slave and master must be made with no movement.
  In this case, the slave does not take the SlaveOffset parameter into account, and calculates an InternalSlaveOffset parameter (which is not communicated to the application), in order to inhibit all slave movements, at the point when it becomes a slave, while the master is immobile.

Specifications

These applications are made up of:

- an infinite axis or "master axis" which moves continuously,
- an axis which is sometimes controlled by a master, and sometimes independent.

Movement Slaved to an External Setpoint

At a Glance

This function is used to slave the position of an axis to a position written by an application program in the double word PARAM (%QDr.m.c.2).

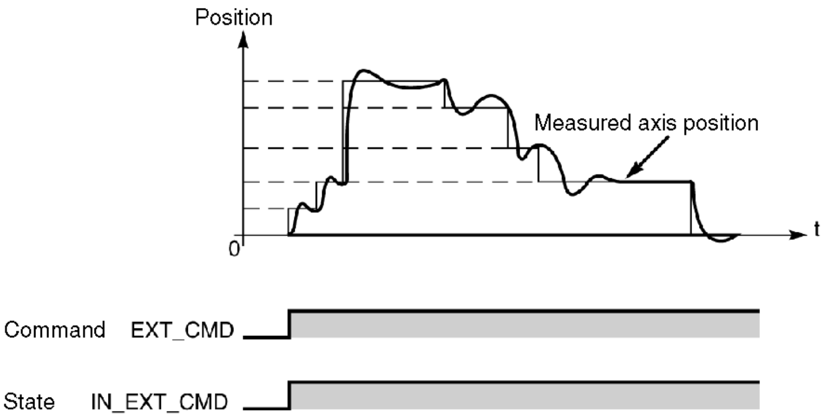
It provides a means of slaving an axis to a pre-programmed trajectory.

It can also be used to control a module axis so that the refresh rate of the slave axis setpoint is equal to the period of the task in which the modules are managed.

Setting the EXT_CMD (%Qr.m.c.18) bit to 1 enables this function. The IN_EXT_CMD (%Ir.m.c.37) status bit indicates that the slave axis is effectively performing the tracking function.

Example

In this mode with TSX CAY •2 and TSX CAY 33 modules, you can control the discrete event output:



Execution Condition

This function is enabled if:

- the axis is referenced,
- no blocking fault is detected,
- the PARAM position is within the soft stops.

NOTE: You must check that:
- the axis is already in the PARAM position before switching this axis to tracking mode,
- PARAM is moving in a continuous and coherent fashion (risk of tracking error fault on the axis).

Deferred PAUSE Function

At a Glance

The PAUSE command (%Qr.m.c.16) is used to suspend movement sequencing. It only becomes active when the moving part has come to a stop, or in other words at the end of a G09 or G10 instruction.

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

When at 1, the ON_PAUSE (%Ir.m.c.33) bit signals that the axis is in PAUSE state.

This function has 2 possible uses:

- block to block execution of the movement program,
- synchronizing axes from the same axis control module.

Block to Block Execution of the Movement Program

If the instruction in progress is an instruction with stop, activating the **PAUSE** command in the debugging screen in automatic mode, or setting the PAUSE (%Qr.m.c.16) bit to 1 puts the module in waiting status after the instruction in progress is finished: movement sequencing stopped.

The movements without stop are stopped when they are terminated by reaching the soft stop.

By successively activating and deactivating the PAUSE command it is thus possible to execute block to block movements in order to facilitate debugging.

Synchronizing Several Axes

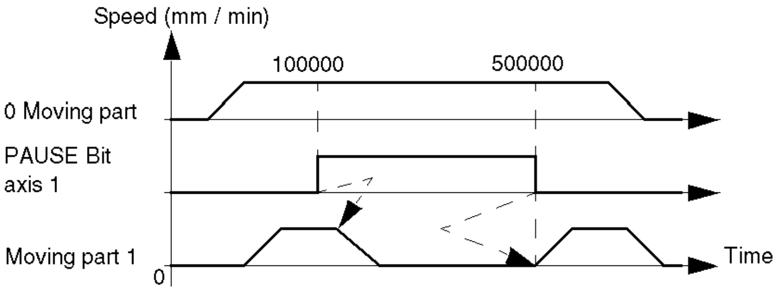
Using the program to set the PAUSE bit (%Qr.m.c.16) for each axis at 1 after the instruction in progress is complete puts the module in waiting status.

When the PAUSE bit is reset to 0, the module continues carrying out the instructions.

Example

Execution of the movement of moving part 1 is stopped when moving part 0 reaches position 100000. Movement is reactivated when moving part 0 reaches 500000.

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



NOTE: The PAUSE command is only processed when AUTO mode is active and when the position tracking functions are inactive.

Step by Step Mode

At a Glance

This mode is used to execute a sequence of movements by stopping after each elementary instruction (step).

Movements without stop are thus transformed into movements with stop at the same value and speed (except for instruction G21, which never stops). For the G30 command, the speed used is the approach speed.

Activating Step by Step Mode

Step by step mode is activated by setting the MOD_STEP (%Qr.m.c.19) bit to 1.

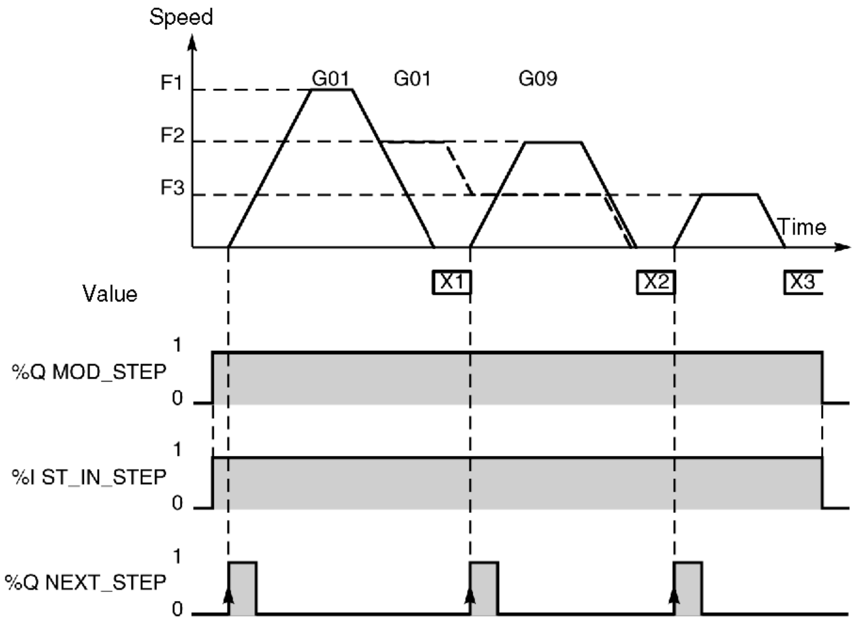
The ST_IN_STEP (%Ir.m.c.39) bit indicates that the mode is active, or in other words that the command in progress has been modified to be executed in step by step mode.

A rising edge on the NEXT_STEP bit (%Qr.m.c.22) is used to start the next step.

Example 1

Step by step execution of the following profile:

```
SMOVE (1,90,01,X1,F1,M)
SMOVE (2,90,01,X2,F2,M)
SMOVE (3,900.09,X3,F3,M)
```



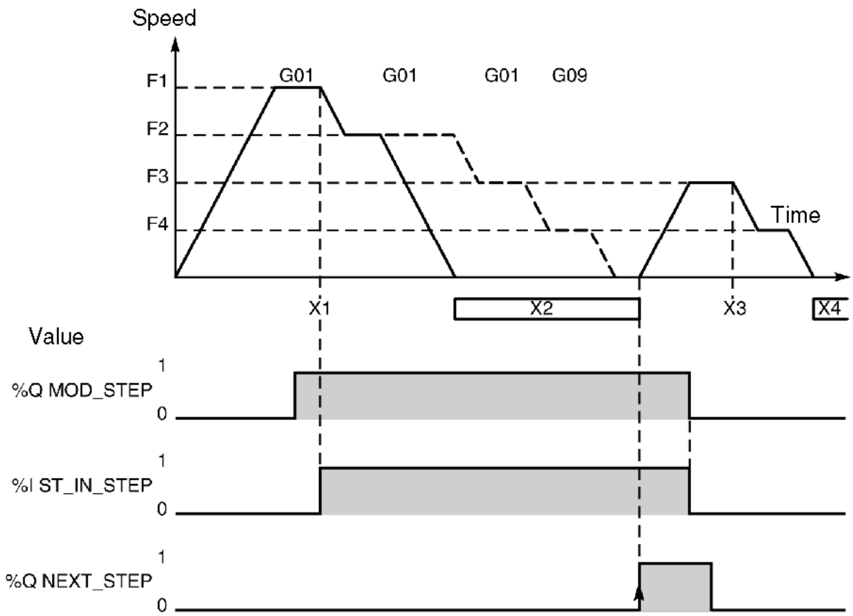
If a movement is in progress on request to go into step by step mode, the mode is entered at the start of the next movement.

However, the mode is exited immediately even if a movement is in progress.

Example 2

Step by step execution of the following profile:

```
SMOVE (1,90,01,X1,F1,M)
SMOVE (2,90,01,X2,F2,M)
SMOVE (3,90,01,X3,F3,M)
SMOVE (4,90,09,X4,F4,M)
```



However, if the exit mode request is made during a deceleration, which corresponds to a transformed movement without stop, the mode is only exited on movement completion.

NOTE: The G05, G07 and G62 commands are performed in step by step,
The G32 command is not considered to be a step.

Feed HOLD Function

At a Glance

This function is used to stop the moving part in automatic mode, while ensuring that, on a resume movement command, it follows the programmed trajectory (with no risk of the command being refused).

Activating the Function

The feed HOLD function is activated:

- by the program: by assigning a value of 0 to the CMV word (%QWr.m.c.1), speed modulation coefficient,
- by the debugging screen: by assigning a value of 0 to the speed modulation coefficient parameter CMV.

It stops the moving part according to the programmed deceleration.

The feedhold status report is indicated by the IM_PAUSE bit (%Ir.m.c.34).

Deactivating the Function

The feed HOLD function is deactivated:

- by the program: by reassigning the initial value (>0) to the CMV word, speed modulation coefficient,
- By the debugging screen: by reassigning the initial value (>0) to the speed modulation coefficient parameter CMV.

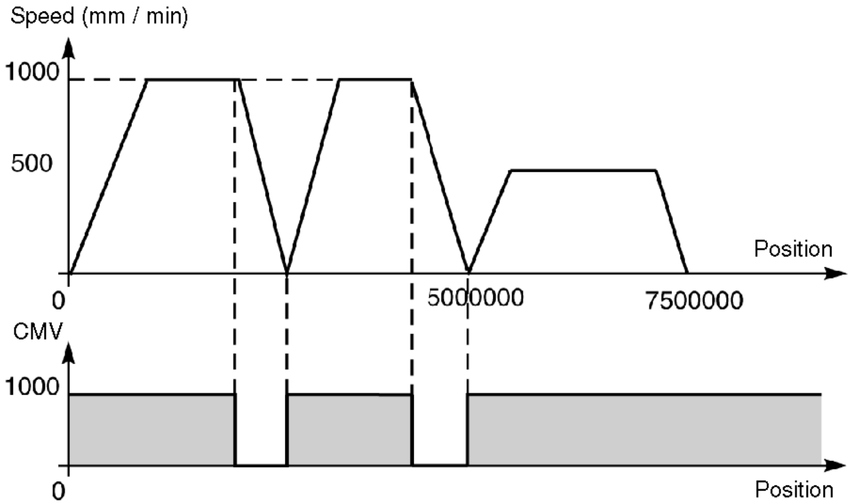
It reinitializes the interrupted movement at a speed corresponding to:

F x CMV / 1000.

Example

Activating / deactivating the feed HOLD function:

```
SMOVE (AXIS_CH0,1,90,10,5000000,1000,0);
SMOVE (AXIS_CH0,2,90,09,7500000,500,0);
................................
IF RE %M10 THEN %MW100 := SMC; SMC := 0;
IF RE %M10 THEN SMC := %MW100;
AXIS_CH0 of type T_AXIS_STD
```



NOTE: This command is deactivated on a STOP order or blocking fault.
For a movement without stop instruction, when, during a stop following a feed HOLD command, the target position is overrun, the respective movement in progress is terminated. In this case, the trajectory is reset with the movement, which was queued in the stack.
The feed hold function is not activated when the movement in progress is a movement slaved to a position (slave axis or PARAM position follower).

Event Processing with an Independent Axis

At a Glance

The TSX CAY module channels are able to activate an event task. To do this, the functionality must have been enabled on the configuration screen by associating an event processing number to the channel.

Activating an Event Task

The following instructions trigger an event transmission, which activates the event task:

- Movement until the event, codes **10** and **11**: the event processing application is activated, on event detection.
- Await event, code **05**: the event processing application is activated, on instruction completion.
- Storage of current position on event occurrence, code **07** : the event processing application is activated on completion of storage of the position PREF1 or PREF2.
- Modulo crossing for an unlimited axis: the event processing application is activated on every modulo crossing during a movement. Event processing activation must be enabled by setting the VALIDEVTMOD parameter (%MWr.m.c.62.0) to 1.

The event processing application is activated if bit 12 of parameter M of the SMOVE function associated with the instruction is set at 1.

Usable Variables for the Event Task

- If several event sources are chosen, the following bits are used to determine the source to trigger event processing application:
  - EVT_G1 (%Ir.m.c.50): G10 or G11 end on event,
  - EVT_G05 (%Ir.m.c.48): G05 end on event,
  - TO_G05 (%Ir.m.c.49): G05 time-out elapsed,
  - EVT_G07 (%Ir.m.c.47): Storage of position,
  - EVT_MOD (%Ir.m.c.51): Modulo crossing.
- The OVR_EVT bit (%Ir.m.c.46) is used to detect a delay in event transmission or an event loss.
- Value of the stored positions PREF1 (%IDr.m.c.9) and PREF2 (%IDr.m.c.11).

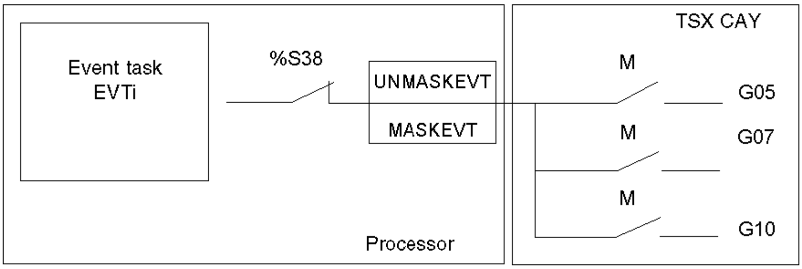
NOTE: The words and bits described below are the only values, which are refreshed during execution of an event task. They are only updated in the PLC on activation of the task.

Event Masking

Control Expert language offers 2 ways to mask events:

- Instruction for global event masking : MASKEVT (UNMASKEVT is used for unmasking):
- Bit %S38 = 0 (global event inhibition). %S38 bit is normally at 1.

Summary diagram



Managing the Operating Modes

On Module Power-up

On module power-up or during run through, the TSX CAY module performs self-tests with the outputs in safety position (outputs to 0).

On completion of the self-tests:

Self-tests	Module
If the self-tests have not detected an error.	The module tests the configuration with the outputs in the safety position. If the configuration is correct, the module goes into measurement mode (OFF).
If the self-tests have detected an error or if the configuration is incorrect.	The module signals a fault and retains the outputs in the safety position.

PLC in RUN

All the configured channel operating modes are useable.

Switching the PLC From RUN to STOP

On switching the PLC from RUN to STOP or on loss of processor / module communication, the moving part decelerates and stops and the module goes into measurement mode (OFF).

NOTE: The `1RTSSCANRUN` bit (%S13) is used to detect the PLC switching to STOP. It is set to 1 during the first cycle after the PLC enters RUN mode.

Change of Configuration (Reconfiguration)

- The moving part decelerates and stops.
- The channel deconfigures.
- The module tests the new configuration with the outputs in the safety position.
- If the new configuration is correct, the module goes into measurement mode (OFF).
- If the configuration is incorrect, the module signals a fault and retains the outputs in the safety position.

Power Outage and Restart

On power outage, the moving part stops.

On a cold start or warm restart, the channel configuration is automatically sent to the module by the processor. The module goes into measurement mode (OFF) then into the mode requested by the program.

Fault Management

At a Glance

Fault monitoring is essential for position control, due to the inherent risks associated with active moving parts.

The module automatically carries out checks internally.

Fault Types

The module detects 4 fault types:

- **The module faults**. These are internal hardware faults within the module. All the axes controlled by the module are thus affected when this type of fault occurs. They can be detected during self-tests (on module reset) or during normal operation (I/O fault).
- **Hardware channel faults external to the module** (for example, encoder break).
- **Application channel faults** associated with the axes (for example, tracking error). Faults are constantly checked at axis level once the axis is configured.
- **Command refused channel faults**. These are the faults, which may occur during execution of a movement command, a configuration transfer, an adjustment parameter transfer or an operating mode change command.

NOTE: The axis monitoring parameters may enable or inhibit the checking of some faults. These monitoring parameters can be adjusted in the adjustment screen.
In loop control disabled mode (DIRDV), checking of application faults is inhibited,
In measurement mode (OFF), checking of application faults except soft stop faults, is inhibited

Levels of Severity

Faults are classed according to 2 levels of severity:

- **The blocking or critical faults** which cause the moving part to stop in the case of an axis fault or moving parts managed by the module in the case of a module fault. The following processes then occur:
  - the fault is indicated,
  - the moving part slows down until analog output is zero,
  - deactivation of the speed drive enable relay,
  - clearing of all memorized commands,
  - wait for acknowledgement.

  The fault must have disappeared and been acknowledged before the application can be re-started.
- **Non-critical faults** which cause a fault to be signaled without stopping the moving part. The Control Expert must be programmed with what action to take in the event of this type of fault. The fault signal disappears when the fault has disappeared and has been acknowledged (the acknowledgment is not stored and only comes into effect if the error has disappeared).

Fault Programming

Faults can be viewed, corrected, and acknowledged from the debugging screen. However it can be useful to be able to control the moving part and correct faults from a terminal during operation. For this purpose, all the information and commands necessary are available within the application.

Fault Indication

The module supports a wide range of information in the form of bits and status words, all accessible through the Control Expert program. These bits are used to process faults in hierarchical order :

- to act on the main program,
- to simply indicate the fault.

Level of Indicating

2 indicating levels are provided:

First level: general information

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

Second level: detailed information

Module fault status words and axes CH_FLT(%MWr.m.c.2) and AX_STS(%MWr.m.c.3)

NOTE: With a blocking fault, it is advisable to stop the sequential processing which is associated with the axis, and correct the fault by controlling the moving part in manual mode. Correction of the fault must be followed by a fault acknowledgment.

Fault Acknowledgement

When a fault appears:

- The fault bits AX_FLT, HD_ERR, AX_ERR and the status word extract bits affected by the fault are put in position 1.
- If it is a blocking fault, the AX_OK bit is set at 0.

When the fault disappears, all fault bits retain their status. A fault is stored until acknowledgement is obtained by setting the ACK_DEF %Qr.m.c.8 bit to 1 (or by resetting the module). The acknowledgment has to be made after the fault has disappeared (except for soft stop errors)

If several faults are detected, the acknowledgement order will only act upon faults which have effectively disappeared. Persisting faults must by acknowledged again after their disappearance.

NOTE: Faults can also be acknowledged on PLC initialization, or when a new correct command is accepted in the case of a command refused fault

Summary Table of the Different Fault Types

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

Channel fault (Bit CH_ERROR: %Ir.m.c.ERR)	Process faults (Bit AX_FLT: %Ir.m.c.2)		
	AX_OK: %Ir.m.c.3 (No blocking fault detected)		Command refused (CMD_NOK bit: %Ir.m.c.6)
	External hardware (HD_ERR bit: %Ir.m.c.4)	Application (AX_ERR Bit: %Ir.m.c.5)	
• Internal • Communication • Configuration • External hardware • Configuration or adjustment	• Emergency stop • Drive • Encoder break • Analog output short circuit • Auxiliary output short-circuit • Encoder supply • Absolute encoder frame	• Soft stops • Overspeed • Recalibration (*) • Tracking error MAX_F1 • Tracking error MAX_F2 (*) • Stopping fault (*) • Target window (*)	Fault coding in the CMD_FLT word: %MWr.m.c.7

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

Description of Channel Faults

The CH_ERROR (%Ir.m.c.ERR) bit covers all faults at channel level:

- Internal fault MOD_FLT (%MWr.m.c.2.4): module absent, inoperative, or in self-test mode.
- Communication fault COM_FLT (%MWr.m.c.2.6): processor communication fault.
- Communication fault CONF_FLT (%MWr.m.c.2.5): difference between the module position declared in the configuration and the current position.

NOTE: In order to be updated, the %MW words require a READ_STS command.

Description of External Hardware Faults

At a Glance

These faults are signaled by the **HD_ERR** bit (%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 solution if an **Emergency Stop** fault should occur:

Cause	Open circuit between 24 V and the "Emergency Stop" input on the front panel of the module
Parameter	None
Result	Moving part is forced to stop
Indication	EMG_STP bit (%MWr.m.c.3.5) (1)
Remedy	Reestablish the input connection at 24 V then acknowledge the fault.

Drive

The following table shows the cause, the signal and the solution if a **Drive** fault should occur:

Cause	Open circuit between 24 V and the "Drive fault" input on the front panel of the module
Parameter	None
Result	Moving part is forced to stop
Indication	DRV_FLT bit (%MWr.m.c.3.2) (1)
Remedy	Eliminate the drive fault then acknowledge the fault

Encoder Break

The following table shows the cause, the signal and the solution if an **Encoder break** fault should occur:

Cause	Discrepancies in information from the encoder
Parameter	None
Result	The axis stops being referenced (in the case of an incremental encoder). Moving part is forced to stop
Indication	ENC_BRK bit (%MWr.m.c.3.4) (1)
Remedy	Reestablish the affected encoder link then acknowledge the fault

NOTE: When there is an encoder link fault, the module stops taking measurements. With an absolute encoder, pulse sequences stop being sent out on the CLK line, until the fault disappears and is acknowledged.

Analog Output Short-circuit

The following table shows the cause, the signal and the solution if an **Analog output short circuit** fault should occur:

Cause	Short circuit detected on one of the analog inputs of the module
Parameter	None
Result	Moving part is forced to stop
Indication	ANA_FLT bit (%MWr.m.c.3.0) (1)
Remedy	Eliminate the short circuit then acknowledge the fault

Auxiliary Output Short-circuit

The following table shows the cause, the signal and the solution if an **Auxiliary output short-circuit** fault should occur:

Cause	Short circuit detected on one of the auxiliary outputs of the module
Parameter	None
Result	Moving part is forced to stop
Indication	AUX_FLT bit (%MWr.m.c.3.1) (1)
Remedy	Eliminate the short circuit then acknowledge the fault.

Encoder Supply

The following table shows the cause, the signal and the solution if an **Encoder supply** fault should occur:

Cause	There is no longer any power supplied to the encoder
Parameter	None
Result	The axis stops being referenced (in the case of an incremental encoder). Moving part is forced to stop
Indication	ENC_SUP bit (%MWr.m.c.3.3) (1)
Remedy	Reestablish the connection then acknowledge the fault

Absolute Encoder Frame

The following table shows the cause, the signal and the solution if an **Absolute encoder frame** fault should occur:

Cause	SSI Frame fault: parity or error bit
Parameter	None
Result	Moving part is forced to stop
Indication	ENC_FLT bit (%MWr.m.c.3.7) (1)
Remedy	Eliminate the fault then acknowledge the fault

24 V Supply

The following table shows the cause, the signal and the solution if an **24 V supply** fault should occur:

Cause	24 V supply fault
Parameter	None
Result	Moving part is forced to stop
Indication	AUX_SUP bit (%MWr.m.c.3.6) (1)
Remedy	Reestablish the connection then acknowledge the fault

Note (1)

In order to be updated, the %MW words require a READ_STS command.

Description of Application Faults

At a Glance

These faults are indicated by the AX_ERR bit (%Ir.m.c.5). The parameters can be accessed by the Adjustment screen of the configuration editor. There **is no** fault checking associated with the soft stops for the **unlimited axes** (modulo).

Soft Stops

The following table shows the cause, indication and the solution if a **Soft stops** fault should occur. This is a blocking fault and cannot be deactivated.

Cause	The moving part is no longer situated between the 2 thresholds: lo and hi software limits (this check is activated as soon as the axis is referenced)
Parameter	Software hi limit: SL_MAX (%MDr.m.c.31) Software lo limit: SL_MIN (%MDr.m.c.33)
Result	Moving part is forced to stop
Indication	SLMAX bit (%MWr.m.c.3.8): Software hi limit exceeded SLMIN bit (%MWr.m.c.3.9): Software lo limit exceeded
Remedy	Acknowledge the fault and in manual mode release the moving part outside of the soft stops within the valid measurement space. To do this you must check: ● that there is no movement in progress, ● that manual mode is selected, ● that the STOP command is at 0, ● that the axis on which this command is carried out is referenced, ● that there is no other fault with stop on the axis. The moving part can either be repositioned manually or by using the JOG+ and JOG- commands.

Overspeed

The following table shows the cause, the signal and the solution if an **Overspeed** fault should occur. This is a blocking fault and can be deactivated.

Cause	On one of the axes, the speed of the moving part has exceeded the increased max. speed of the overspeed threshold: VMAX (1 + OVR_SPD)
Parameter	Overspeed threshold OVR_SPD (%MWr.m.c.23). If this parameter is equal to 0, the monitoring is inhibited.
Result	The moving part is stopped
Indication	SPD_FLT bit (%MWr.m.c.3.10)
Remedy	Acknowledge the fault

Stop

The following table shows the cause, the signal and the solution if a **Stop** fault should occur. This is a non-blocking fault and can be deactivated.

Cause	As soon as the speed setpoint value calculated by the module becomes equal to 0, the module activates a T_STOP time-out: ● If this parameter is equal to 0, the fault monitoring is inhibited. ● If this parameter is different from 0, when the time-out has elapsed, the module compares the moving part's measured speed with the stop speed S_STOP. If the measured speed is greater than S_STOP, the module indicates a stop fault.
Parameter	T_STOP (%MWr.m.c.25): maximum delay for detecting a stop S_STOP (%MWr.m.c.24): speed at which moving part is considered to be at a stop
Result	the fault is signaled
Indication	STP_FLT bit (%MWr.m.c.3.14)
Remedy	Eliminate the drive fault or perform further adjustments then acknowledge the fault

Target Window

The following table shows the cause, the signal and the solution if a **Target window** fault should occur. This is a non-blocking fault and can be deactivated.

Cause	When a movement to a position with stop is requested, the module checks that the position reached corresponds to the requested position, according to the theoretical stop, using a tolerance that you defined in the parameter TW (Setpoint – TW <= measurement <= Setpoint + TW) If this parameter is equal to 0, the check is inhibited
Parameter	TW (%MDr.m.c.49): target window
Result	If the moving part is not in the target window, the fault is signaled
Indication	TW_FLT bit (%MWr.m.c.3.13): target window fault
Remedy	Check the control loop then acknowledge the fault

Recalibration

The following table shows the cause, the signal and the solution if a **Recalibration** fault should occur. This is a non-blocking fault and can be deactivated.

Cause	During a recalibration event, the error between the current position and the recalibration reference value is higher than the recalibration threshold The check is inhibited if you have chosen the Recalibration function missing configuration parameter
Parameter	RE_WDW (%MDr.m.c.51): recalibration deviation threshold RE_POS (%MDr.m.c.43): recalibration reference value
Result	If the deviation exceeds the threshold, the fault is signaled
Indication	REC_FLT bit (%MWr.m.c.3.12): recalibration fault
Remedy	Check the control loop then acknowledge the fault

Zero Latch Presence Monitoring

The following table shows the cause, the signal and the solution if a **Zero Pulse Presence Monitoring** fault should occur.

Cause	During a reference point short cam with zero pulse
Parameter	None
Result	The axis stops
Indication	CMD_NOK bit (%Ir.m.c.6) CMD_FLT word (%MWr.m.c.7) = 16#0015
Remedy	Mechanically adjust the cam then restart the operation

Movement Monitoring

The following table shows the cause, the signal and the solution if a **Movement monitoring** fault should occur: This is a blocking fault and can be deactivated.

Cause	When the analog output from a channel exceeds a VLIM limit (in absolute value), a time delay is activated When T is reached, a fault is signaled if the position value is the same as that of the internal cycle of the module
Parameter	Limit analog output: VLIM (%MWr.m.c.27) The time-out T is programmed to TACC / 2. TACC (%MWr.m.c.26) is the acceleration adjustment parameter
Result	If the fault is detected the moving part is stopped (analog output set to 0 and speed control authorization relay open) The check is only enabled if VLIM > 0
Indication	FE1_FLT bit (%MWr.m.c.3.11): MAX_F1 deviation exceeded
Remedy	Check the control loop then acknowledge the fault

NOTE: Movement monitoring is active in direct control, manual and automatic modes.

Following Error

The following table shows the cause, the signal and the solution if a **Tracking error** fault should occur. The MAX_F1 fault is a blocking fault and can be deactivated. The MAX_F2 fault is a non-blocking fault and can be deactivated.

Cause	During a movement, the module compares the measured position of the moving part. A fault is signaled when the tracking error becomes higher than the maximum authorized error that you defined
Parameter	Abnormal non-critical following error MAX_F2 (%MDr.m.c.47) Abnormal critical following error MAX_F1 (%MDr.m.c.45) If these parameters are at 0, monitoring is inhibited.
Result	If the MAX_F2 error is exceeded, the fault is signaled If the MAX_F1 error is exceeded, the moving part is stopped. This fault is only taken into account if MAX_F1 is different from 0
Indication	FE2_FLT bit (%MWr.m.c.3.15): MAX_F2 deviation exceeded FE1_FLT bit (%MWr.m.c.3.11): MAX_F1 deviation exceeded
Remedy	Check the control loop then acknowledge the fault

Description of Command Refused Faults

At a Glance

A command refused fault is generated each time a command cannot be executed. This occurs when a command is not compatible with the axis state, with the mode in progress or where at least one of the parameters is not valid.

These faults are indicated by the Refus Cde LED on the debugging screens. At channel level, the DIAG key can be used to identify the source of the command refused fault. This information can also be accessed by the program with the CMD_NOK (%Ir.m.c.6) bit and CMD_FLT (%MWr.m.c.7) word.

Command Refused

The following table shows the cause, indication and the remedy in the event of a **Command Refused** fault.

Cause	Unauthorized movement command Transfer of faulty configurations or parameters
Parameter	None
Result	Immediate stop of movement in progress Buffer memory, which receives movement commands in automatic mode, is reset to 0.
Indication	CMD_NOK bit (%Ir.m.c.6): Movement command refused CMD_FLT mot (%MWr.m.c.7): type of fault detected ● Least significant byte : movement commands, ● Most significant byte : configuration and parameter adjust.
Remedy	When a new command is received and accepted, acknowledgement is implicit. Acknowledgment is also possible via the ACK_DEF (%Qr.m.c.8) command

NOTE: For movement sequences in automatic mode, it is advisable to make the execution of each movement conditional upon the end of the previous movement, and the AX_FLT (%Ir.m.c.2) bit. This will prevent the program moving on to the following command should the present command be refused..

Managing Manual Mode

At a Glance

Manual mode can be selected and controlled from the debugging screen, but also via the application program from the front panel or human-machine interface/monitoring terminal.

In this case, the dialog is programmed in ladder, instruction list or structured text language, with the help of elementary commands (movements, reference points etc.).

Manual Mode Selection

Manual mode is selected by assigning the value of 2 to the MOD_SEL (%QWr.m.c.0) word.

Switching from the current to manual mode forces the moving part to stop of there is a movement in progress. Manual mode is engaged as soon as the moving part has stopped.

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

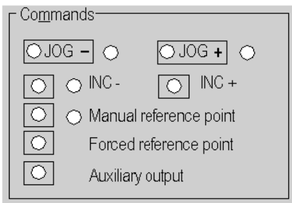
Execution of Manual Commands

Elementary commands associated with manual mode and accessed via command bits %Qr.m.c.d are as follows:

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

These commands are the same as those that can be accessed from the referenced axis TSX CAY module debugging screen.

Manual commands:



General Execution Conditions for Commands in Manual Mode

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

- Target position within soft stops.
- Axis without fault blocking: AX_OK bit (%Ir.m.c.3) = 1.
- No command in progress: DONE bit (%Ir.m.c.1) = 1.
- STOP (%Qr.m.c.15) command inactive and ENABLE (%Qr.m.c.9) bit for variable speed controller safety relay set to 1.

NOTE: Except for, in the case of soft stop fault, JOG_P and JOG_M commands and after fault acknowledgment.

Movement Stop

A movement stop can be caused by:

- Appearance of a STOP (%Qr.m.c.15) command or the ENABLE (%Qr.m.c.9) bit being set to 0.
- Appearance of a blocking fault.
- Change in operating mode.
- Receiving a configuration.

Visual Movement Commands

At a Glance

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

The JOG_P (%Qr.m.c.1) and JOG_M (%Qr.m.c.2) bits give the command for the moving part to move in a positive or negative direction. The operator must visually follow the position of the moving part. Movement occurs as long as the command is present, and a STOP command or a fault does not inhibit the command.

For limited axes, the JOG_P and JOG_M commands cause an automatic stop at the latest at a distance from the soft stops equal to the target window distance.

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

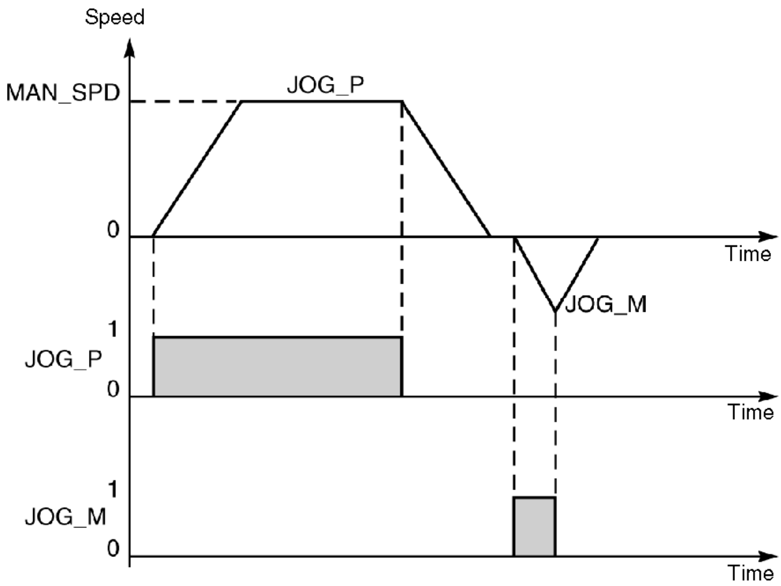
Movement Speed

Movement occurs at the speed of the manual mode MAN_SPD (%MDr.m.c.35) defined in the adjustment screen.

The speed can be modulated during a movement via the CMV coefficient (%QWr.m.c.1).

Any movement speed greater than VMAX (maximum axial speed defined in configuration) is limited to the value of VMAX.

Speed of moving part:



Notes on JOG_P and JOG_M Commands

- JOG_P and JOG_M commands are used to release the moving part when a soft stop fault is detected. This happens after prior acknowledgment of the fault.
- If the JOG_P or JOG_M bit is set to 1 during operation in manual mode, this command is ignored. It will only be taken into account after the bit has been set at 0 then reset to 1.

Incremental Movement Commands

At a Glance

INC_P and INC_M manual commands must be used to carry out an incremental movement.

INC_P (%Qr.m.c.3) and INC_M (%Qr.m.c.4) bits give the command for movement by incrementation of the position of a moving part in a positive or negative direction.

The value of the PARAM position increment is entered either in the %QDr.m.c.2 double word or in the debugging screen of the TSX CAY module.

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

- the axis is referenced for limited run-time machines.
- The target position is between the soft stops.

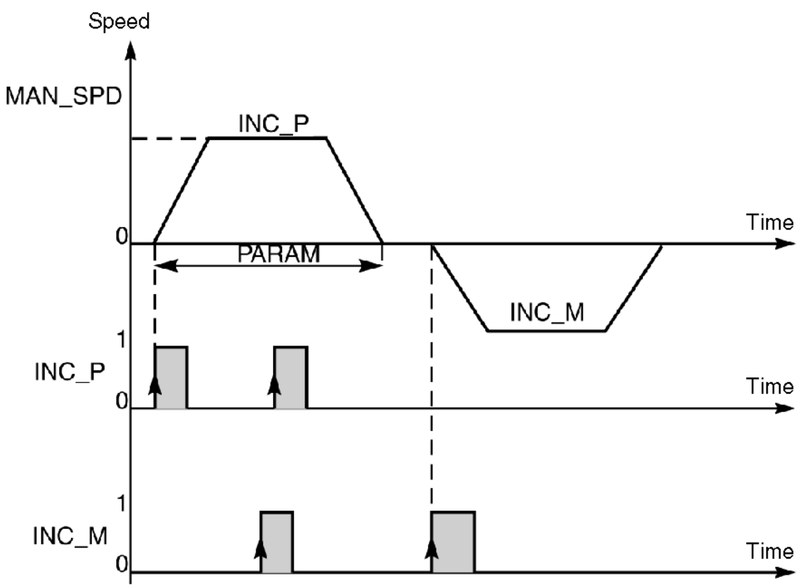
Movement Speed

Movement occurs at the speed of the manual mode MAN_SPD, defined in the adjustment screen (or in the double word %MDr.m.c.35).

The speed can be modulated during a movement via the CMV coefficient (%QWr.m.c.1).

Any movement speed greater than VMAX (maximum axial speed defined in configuration) is limited to the value of VMAX.

Speed of moving part:



Reference Point Command

At a Glance

If an **incremental encoder** is being used, a reference point can be set with the SET_RP command.

Bit SET_RP (%Qr.m.c.5) carries out a manual reference point with movement.

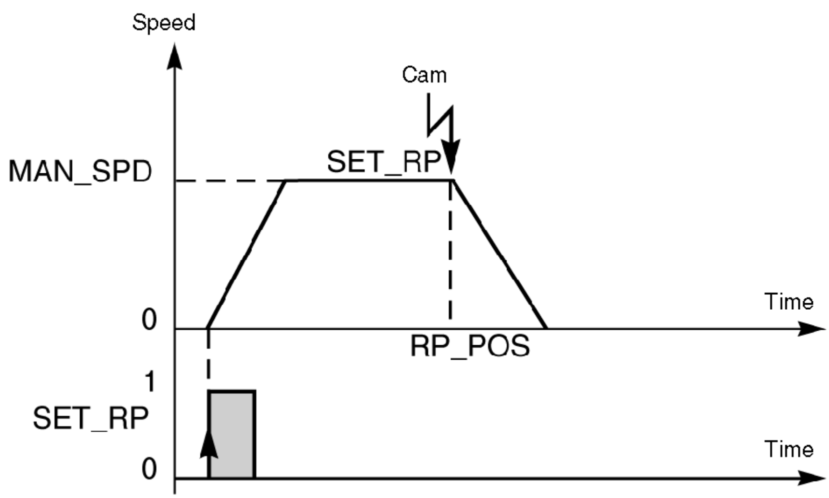
The type and direction of the reference point are set in the Reference Point parameter on configuration. The source value is set on the adjustment screen by the PO Value parameter (or by the RP_POS double word) %MDr.m.c.41).

Approach Speed

Approach speed is the manual speed MAN_SPD set on the adjustment screen (or by %MDr.m.c.35 double word) multiplied by the CMV speed modulation coefficient. The speed of the reference point varies according to the type of reference point chosen.

Any movement speed greater than VMAX (maximum axial speed defined in configuration) is limited to the value of VMAX.

Example: short cam only and + direction



Forced Reference Point Command

At a Glance

If an **incremental encoder** is being used, a forced reference point can be set with the RP_HERE command.

The RP_HERE (%Qr.m.c.6) forces a reference point without movement to the value set in the PARAM parameter. This value is entered either in the %QDr.m.c.2 double word or in the debugging window for the TSX CAY module.

The forced reference point command references the axis without generating a movement.

NOTE: The RP_HERE command does not modify the value of the RP_POS parameter.
The value of the PARAM parameter must be between the soft stops.
All blocking faults are tolerated while this command is being carried out (except for an encoder break fault).

Cancel References Command

At a Glance

If an **absolute encoder** is being used, a reference point can be canceled using the SET_RP command.

Cancellation of references needs to be carried out before a reference command can be given.

An edge on the SET_RP (%Qr.m.c.5) bit is used to change the axis to a non-referenced state, in order to be able to shift the moving part without provoking a soft stop fault. However it is not possible to overshoot, in either direction, to a position outside the measurement area of the absolute encoder.

Parameter ABS_OFF (%MDr.m.c.53) is forced to 0

Referencing and Offset Calculation Command

At a Glance

If an **assisted offset absolute encoder** is being used, referencing and offset calculations can be set using the RP_HERE command.

An edge on the RP_HERE (%Qr.m.c.6) bit is used to change the axis to a referenced state.

Assisted Offset

If the encoder has been declared in assisted offset mode (any reference must be cancelled), the offset is recalculated at the current point as being at position defined in the PARAM parameter. This position is entered either in the %QDr.m.c.2 double word or in the debugging screen of the TSX CAY module.

In this case, it is essential to force adjustment parameters to be saved so as not to lose them on restart:

- Either use **Save parameters** from the adjustment screen.
- Or via the application by calling the SAVE_PARAM function.

NOTE: The value of PARAM must be between the soft stops.
The offset calculation is refused if an adjustment is in progress or if the axis is referenced.
If resolution is modified, the offset must be recalculated.

Managing Loop Control Disabled Mode (DIRDRIVE)

At a Glance

DIRDRIVE (loop control disabled) mode is used to free the axis from slave mode. The axis acts as a digital/analog converter (D/A converter) and the position loop is inoperative.

Axis behavior can be analyzed independently of the process control loop.

Selection of Loop Control Disabled Mode

Loop control disabled mode is selected by assigning the value of 1 to the MOD_SEL (%QWr.m.c.0) word.

When a change of mode is requested, the moving part first stops then the mode is actually changed. When the command to change to loop control disabled mode is taken into account, the IN_DIRDR (%Ir.m.c.21) bit is set to 1.

Execution of Commands in Loop Control Disabled Mode

The loop control disabled mode features the movement command DIRDRV (%Qr.m.c.0).

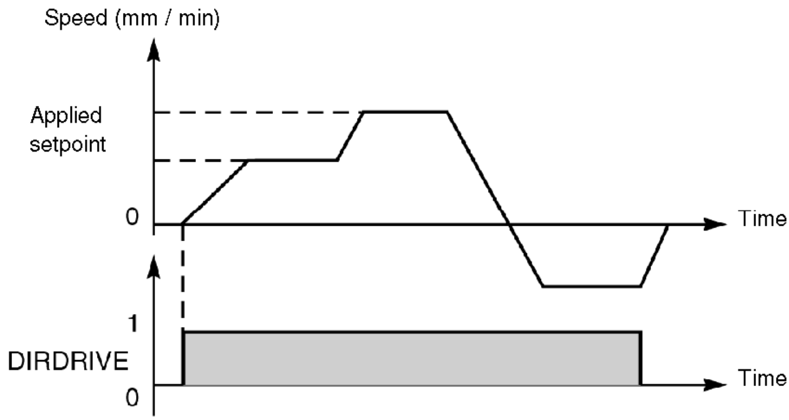
The drive is controlled at a voltage of between -UMAX and +UMAX (the value UMAX is defined in the configuration screen). It is expressed in mV. Application of this setpoint is rounded off into multiples of 1.25 mV (For example, if 1004 mV is requested, the setpoint will be rounded down to 1003.75 mV, and the display screen will show 1003 mV).

The voltage setpoint is periodically sent by the variable PARAM (%QDr.m.c.2). The variable sign gives the direction of movement. The software fault checks are inhibited (excepted for soft stop checks if the axis is referenced).

In order to take in account the mechanics during a setpoint change, the acceleration/deceleration value is respected when switching to a new value.

Speed Law

On a setpoint change, the output reaches the new setpoint according to a trapezoidal speed law, while respecting the parametered acceleration.



Execution of DIRDRIVE Command

The general execution conditions for a DIRDRIVE function are as follows:

- Axis without fault blocking: AX_OK bit (%Ir.m.c.3) = 1.
- STOP (%Qr.m.c.15) command inactive and ENABLE (%Qr.m.c.9) bit for variable speed controller safety relay set to 1.
- Voltage parameter PARAM (%QDr.m.c.2) between – UMAX and + UMAX for selected axis.

Movement Stop

A movement stop can be caused by:

- Appearance of a STOP command or ENABLE (Qr.m.c.9) bit for variable speed controller safety relay set to 0.
- Appearance of a blocking fault or a soft stop fault.
- Change in operating mode.
- Receiving a configuration.

Managing Measurement (OFF) Mode

At a Glance

The measurement mode must be used each time the moving part has to pass outside module control (moving part moved by hand or controlled by an exterior device). In this mode the module remains passive, but continues to update the position (%IDr.m.c.0) and current speed (%IDr.m.c.2) information bits.

Selection of Measurement Mode

Measurement mode is selected by assigning the value of 0 to the MOD_SEL (%QWr.m.c.0) word.

The module also selects measurement mode while the PLC is in STOP mode. By default, it is selected following channel configuration.

Execution of Commands in Measurement Mode

OFF mode is not associated with any movement commands.

Movement of the moving part is not monitored and software fault checks are inhibited (except soft stop checks). The position loop is inoperative.

The variable speed controller enable relay is unlocked whatever the status of the ENABLE (%Qr.m.c.9) bit.

Modules TSX CAY 22/42 and TSX CAY 33

- The AUX_OUT (%Qr.m.c.11) command is used to control the auxiliary output.
- The RP_HERE command can be executed in measurement mode.

Chapter 9
Axis Control Configuration

Subject of this Section

This section describes the TSX CAY module configuration screens as well as the parameters, which must be defined for the independent axes.

What Is in This Chapter?

This chapter contains the following topics:

Description of the Axis Control Module Configuration Screen

General

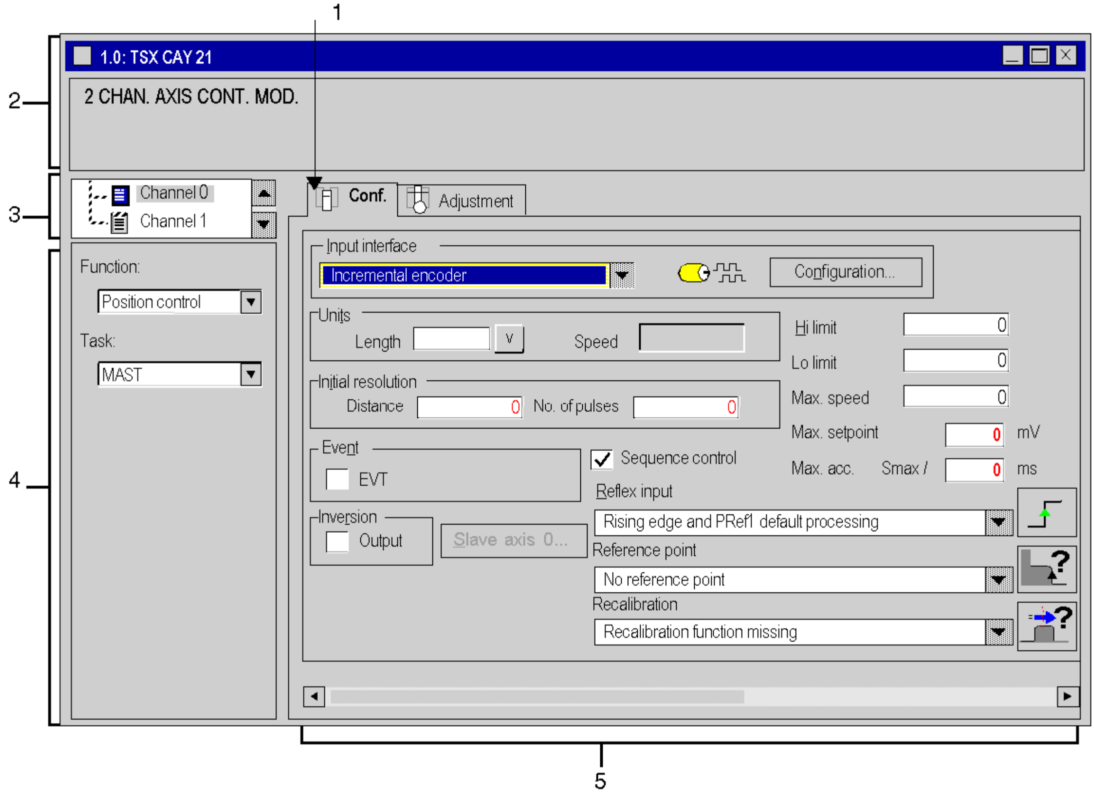
The configuration screen is a graphic tool intended for configuring *(see EcoStruxure™ Control Expert, Operating Modes)* a module selected in a rack. It displays the parameters associated with the channels of this module, and allows these to be modified in offline and online modes.

It also allows access to the adjustment and debugging screens (the latter in online mode only).

NOTE: It is impossible to configure a program module using %KW language objects directly, as these words have read-only access.

Illustration

The diagram below shows a configuration screen.



Description

The following tables presents the various element of the configuration screen and their functions.

Number	Element	Function
1	Tabs	The tab in the foreground indicates the current mode (**Configuration** in this example). Each mode can be selected by the corresponding tab. The modes available are: ● **Configuration,** ● **Adjustment,** ● **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 command function and the task associated with the channel: ● **Function :** command function for axis from among those available for the modules concerned. Depending on this choice, the headings of the configuration zone may differ. By default, no function is configured (**None**). ● **Task:** defines the (**MAST** or **FAST**) task in which the explicit exchange objects of the channel will be exchanged.
5	**Configuration** field	Allows you to configure the channel parameters. This field consists of a number of headings, which are displayed according to the function selected. Some selections may be set and appear dimmed. For each parameter, the limits are displayed in the status bar. In this example, the **Input interfaces** heading comprises a button giving access to a sub-menu which must be filled.

Axis Type

Introduction

The **Machine** zone is only displayed with a TSX CAY •2 or TSX CAY 33 module. It is used to choose the type of axis to be managed by the channel.

Machine Type

Two selection buttons are used to choose your machine type:

Parameter	Description
Limited	For a limited machine, the position measurement advances between two values defined by soft stops
Infinite	For an infinite machine, the position measurement advances between the values 0 and Modulo.

Encoder Type

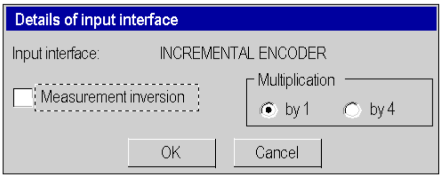
At a Glance

The **Input interface** field allows you to choose the encoder type: incremental encoder or absolute SSI encoder.

To access the input interface parameters screen, press the **Configuration** button: **Input interface details**.

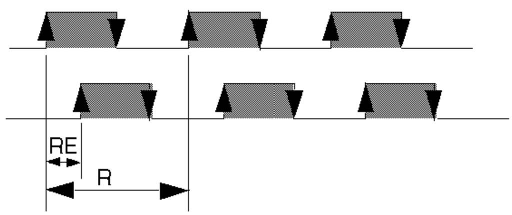
Incremental Encoder Parameters Screen

In the event of an incremental encoder, the configuration screen of the input interface is as follows:



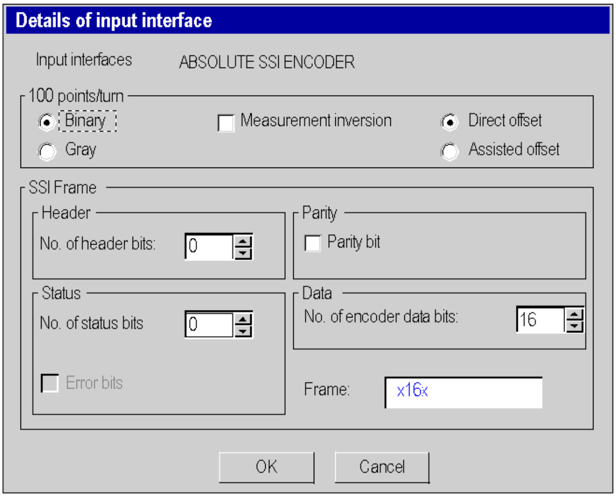
Incremental Encoder Parameters

The parameters of an incremental encoder are as follows:

Parameter	Description
Measurement inversion	If you check this box, the direction in which the measurement changes will be reversed.
Multiplication by 1 or multiplication by 4	Multiplying by 4 increases the precision of the encoder: ● for any given encoder, the precision will be 4 times greater, ● to obtain a given resolution, you can use a resolution encoder, which is 4 times smaller. The following diagram illustrates multiplication by 4  RE, corresponding to the resolution obtained by multiplication by 4, is also called equivalent resolution.

Absolute Encoder Parameters Screen

The input interface configuration screen is used to define the characteristics of the SSI frame used by the encoder.



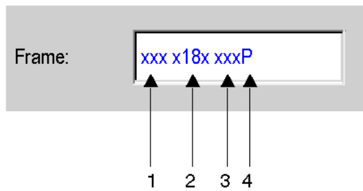
Encoder Field Description

The Encoder field is used to define the type of coding:

Parameter	Description
Direct offset or assisted offset	With a direct offset, you must specify the offset value in encoder points. With assisted offset, the offset is calculated by the module, using a position value supplied by the user. By default, direct offset is selected.
Binary or Gray	These selection buttons are used to define the code used by the encoder: binary code or Gray code (binary code by default).
Measurement inversion	This parameter defines the measurement inversion; in other words it defines direction in which the measurement changes for a given encoder rotation direction. This parameter is inaccessible from an infinite machine.

Description of SSI Frame Field

The SSI frame field is used define the characteristics of the frame:

Parameter	Description
No. of header bits	Number of frame header bits (not significant): 0 to 4 (0 by default).
No. of encoder data bits	Number of frame data bits for the modules: TSX CAY 21/41:16 to 25 (16 by default), TSX CAY •2/33:12 to 25 (12 by default).
No. of status bits	Number of frame status bits: 0 to 3 (0 by default). If you choose a number of bits other than 0 it will give access to the error bit and its position (Position 1 to 3) in the status bits zone.
Presence of parity bit	Presence or absence of parity bit (absence by default). If this box is checked, you can define the type of parity: even or odd. If you select odd parity, the module will no longer perform the parity check and the parity bit will be managed as a status bit.
Frame	Redisplays the characteristics defined for the SSI frame.  Addresses: 1 : number of header bits 2 : number of data bits 3 : number of status bits 4 : presence of parity bit P: parity, I: Odd

Absolute Encoder with Parallel Outputs

It is possible to connect an absolute encoder with parallel outputs, using an ABE-7CPA11 conversion interface. In this case, you must enter the configuration of an absolute SSI encoder.

Initial Resolution

At a Glance

The initial resolution corresponds to an encoder increment. As this is usually not an integer, it is expressed in the form of the following ratio:

RESOL = Distance / No. of points where:

- Distance = distance covered by the moving part
- No. of points = number of encoder points corresponding to the distance covered. The threshold is 1 to 1,000,000.

The resolution is then deduced from these 2 parameters (Distance and Nbre points) in a ratio of 0.5 to 1,000.

Resolution Calculation Example

Either an incremental encoder of 512 points per turn. The distance covered by 1 encoder turn is 10,000 microm (microm = chosen length unit).

You must enter:

- Distance = 10 000,
- Number of points = 512

The resolution is then 19.5 microm:

$$RESOL = 10000 / 512 = 19,5$$

NOTE: You can correct this resolution in the adjustment screen. It is therefore called the initial resolution.
In the case of an incremental encoder with multiplication by 4, enter the distance corresponding to RE *(see page 223)*

Measurement Units

At a Glance

This field is used to choose the physical units in which position and speed measurements are expressed.

Proposed Measurement Units

The following table shows a list of proposed measurement units. To view this list, press the "down arrow" button situated at the right of the entry field.

Position unit (Length)	Speed unit
microm	mm/min
mm	m/min
in.e^{-2} (10^{-2} inch)	in.e^{+1}/min (10 inch/min)
in.e^{-5} (10^{-5} inch)	in.e^{-2}/min (10^{-2} inch/min)

Customized Measurement Units

You can choose your own measurement units, with the Length field allowing a maximum of 5 characters to be entered. For example, degrees.

You must, however, choose the position unit to be such that the resolution value (Distance / No. of points) is between 0.5 and 1,000. The speed unit is calculated using the formula:

Speed unit = Position unit * 1000 / min

NOTE: You cannot choose the speed unit which results from the formula. You can however modify the text. For example,
Either an incremental encoder capable of 500 points per turn. The distance corresponding to 1 turn is 2mm, or 2000 microm. The resolution is expressed by the ratio of 2000 / 500 (thus in microm). The resulting speed unit will be mm/min (see table below).

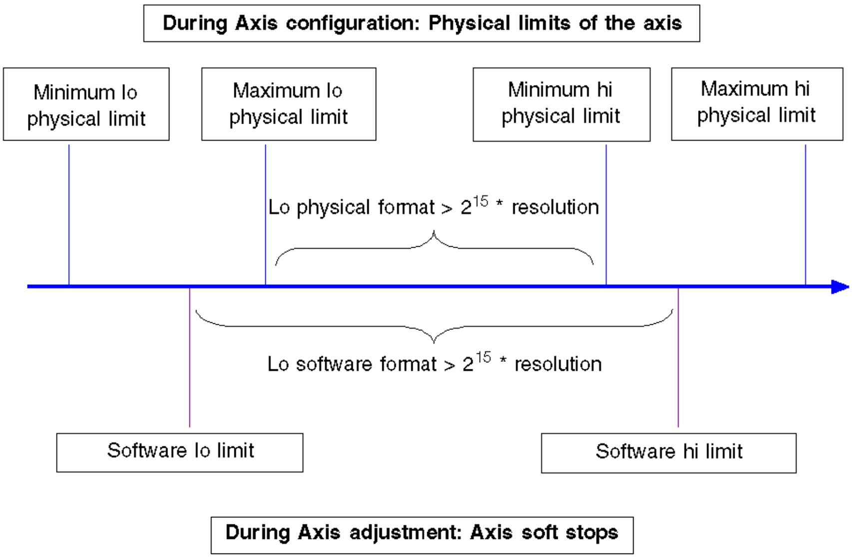
Hi and Lo Limits

At a Glance

The hi and lo limits apply in the case of a limited machine and correspond to the physical extremities of the axis. These limits are themselves limited according to the value of the chosen resolution.

TSX CAY •1 Modules

Graphic presentation



Restrictions

Determining the physical limits depending on the association of an encoder with the CAY •1 module. Take the smallest absolute value of the encoder or the CAY •1 module.

Encoder type	Incremental encoder	Absolute encoder	TSX CAY •1
Minimum lo physical limit	$-16 * 10^6$ * resolution	$-16 * 10^6$ * resolution * 2^{n-25}	-10^8
Maximum lo physical limit	0	0	0
Minimum hi physical limit	0	0	0
Maximum hi physical limit	$16 * 10^6$ * resolution	$16 * 10^6$ * resolution * 2^{n-25}	10^8

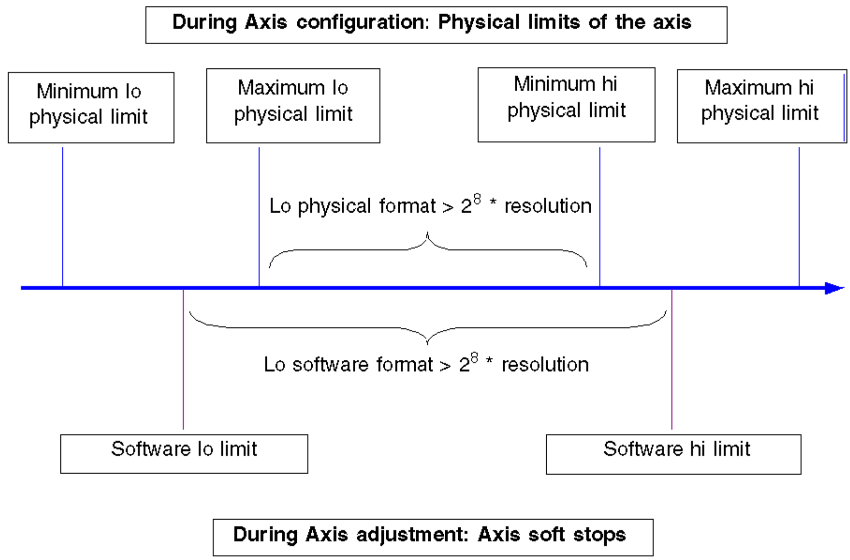
Where n = number of encoder bits

The values to be entered should respect the following equations:

- Minimum lo physical limit $\leq$ Software lo limit $\leq$ Maximum lo physical limit $\leq$ 0 $\leq$ Minimum hi physical limit $\leq$ Software hi limit $\leq$ Maximum hi physical limit.
- Minimum lo physical limit $\leq$ **Entry value** of the lo physical limit $\leq$ **Entry value** of the software lo limit $\leq$ Maximum lo physical limit.
- Minimum hi physical limit $\leq$ **Entry value** of the software hi limit $\leq$ **Entry value** of the hi physical limit $\leq$ Maximum hi physical limit.
- Minimum physical format = Minimum hi physical limit - Maximum lo physical limit > 2^{15} * resolution.
- Minimum software format = Software hi limit - Software lo limit > 2^{15} * resolution.

TSX CAY •2/33 modules

Graphic presentation



Restrictions

Determining the physical limits according to the association of an encoder with the CAY •2/33 module. Take the smallest absolute value of the encoder or the CAY •2/33 module.

Encoder type	Incremental encoder	Absolute encoder	TSX CAY •2/33
Minimum lo physical limit	$-16 * 10^6$ * resolution	$-16 * 10^6$ * resolution * 2^{n-25}	$-6 * 10^8$
Maximum lo physical limit	0	0	0
Minimum hi physical limit	0	0	0
Maximum hi physical limit	$16 * 10^6$ * resolution	$16 * 10^6$ * resolution * 2^{n-25}	$6 * 10^8$

Where n = number of encoder bits

The values to be entered should respect the following equations:

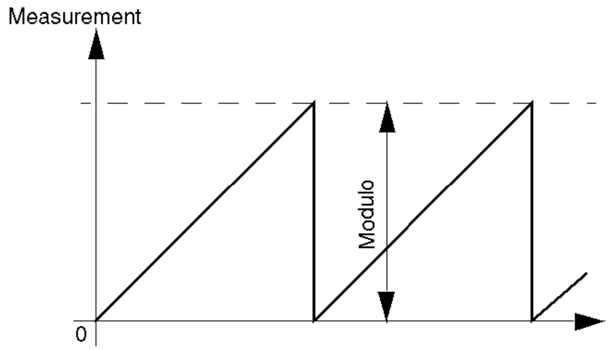
- Minimum lo physical limit $\leq$ Software lo limit $\leq$ Maximum lo physical limit $\leq$ 0 $\leq$ Minimum hi physical limits $\leq$ Software hi limit $\leq$ Maximum hi physical limit.
- Minimum lo physical limit $\leq$ **Entry value** of the lo physical limit $\leq$ **Entry value** of the software lo limit $\leq$ Maximum lo physical limit.
- Minimum hi physical limit $\leq$ **Entry value** of the software hi limit $\leq$ **Entry value** of the hi physical limit $\leq$ Maximum hi physical limit.
- Minimum physical format = Minimum hi physical limit - Maximum lo physical limit > 2^8 * resolution.
- Minimum software format = Software hi limit - Software lo limit > 2^8 * resolution.

Modulo

Introduction

The modulo is applicable with an infinite machine when position measurement moves between 0 and modulo.

Measurement progresses as follows:



Max. Modulo

For the **Max Modulo** parameter you must enter the modulo value in encoder points. The equivalent in user units is displayed automatically in the following field.

Max. Modulo thus defines the hi limit authorized for the adjustable **Modulo** parameter. The values which can be entered for Modulo and the associated parameters depend on the encoder type used.

For an Absolute Encoder

The modulo is always a power of 2 because it is defined by the number of data bits from the SSI Frame encoder. Thus, rather than entering the number of modulo bits, you must enter the corresponding power of 2.

For example, for a max modulo of 4096, you must enter the value 12 (because 4096 = 2^{12}).

Limits According to Resolution

The limit values of Max. Modulo are also limited according to the value of the chosen resolution:

Encoder type	Limits
Incremental encoder	1000 -> 6×10^8 / RESOL limited to 16×10^6
Absolute encoder	n: 12 -> 23 (under condition $2^n \times$ RESOL $< 6 \times 10^8$)

Maximum Speed

Introduction

The maximum speed VMAX must be such that the resultant frequency satisfies the following condition:

1.8 kHz < FMAX < 900 kHz

with FMAX = VMAX x m / RESOL

m = 2 with a x1 incremental encoder or an absolute encoder,

m = 4 with a x4 incremental encoder.

VMAX Value

The condition on the maximum frequency is translated onto the VMAX value by the following relation:

108 x RESOL / m < VMAX < 54000 x RESOL / m

restricted to the following limits:

270 < VMAX < 270000

NOTE: VMAX and RESOL are expressed in the units defined in the configuration screen, either RESOL in microm and VMAX in mm/min, or RESOL in mm and VMAX in m/min, etc.

Authorized Speed

Whatever the speed programmed in the instructions, the module is authorized to operate at a speed equal to VMAX + 10%, on the transitories, in order to reduce following error.

Maximum Setpoint

At a Glance

The maximum setpoint UMAX is the voltage which has to be applied to the variable speed controller, in order to reach a speed equal to the maximum speed.

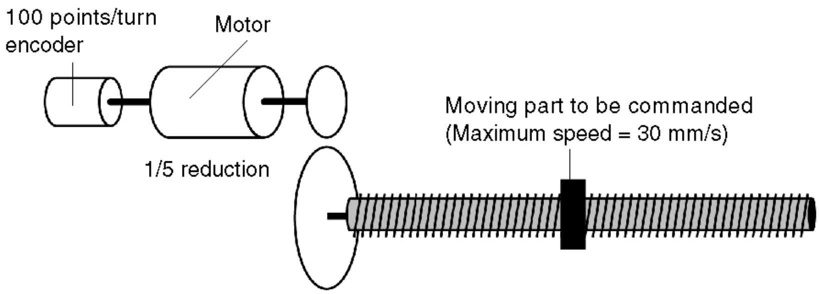
Adjusting the Variable Speed Controller

In the absence of any particular speed constraint, the variable speed controller can be adjusted to obtain the maximum speed for a voltage as close as possible to, but less than 9 V.

Limiting voltage to 9 V enables a reserve to be made available during transitory periods allowing a surge speed to be attained. If there are no constraints imposed by the mechanics, or by the maximum acceptable frequency, choose the following value: Maximum setpoint = 9000 mV

Example

We would like to control an axis with the following characteristics:



Desired maximum linear speed of 30 mm/s, or 1800 mm/m. Thread of 5 mm.

The axis is controlled by a motor capable of 3000 rev/min, which drives a ball screw via a speed reducer with a ratio of 1/5. The encoder is on the motor shaft. Let us assume that it is an incremental encoder without multiplication by 4.

The RESOL parameter (distance covered by the moving part between 2 encoder increments) is equal to: Ne x Thread / N = 1/5 x 5 / 1000 = 1 microm

- The maximum operating speed is 1800 mm/m,
- Maximum setpoint is the voltage value at which maximum speed can be reached. Considering the reduction ratio (1/5) and the thread size (5 m), the maximum linear speed (1800 mm/min) corresponds to a motor speed of 1800 rev/min.
  If the variable speed controller is adjusted to reach a speed of 3000 rev/min with an input voltage of 10 V; in order to reach 1800 rev/min, the voltage must be 6 V (maximum setpoint = 6000 mV).

Consistency of Parameters

The consistency of the RESOL, Maximum speed and Maximum setpoint parameters must be ensured, otherwise inconsistent control loop behavior will result.

Event

Introduction

The **Event** parameter is used to associate an event processing task with a channel. To do this, an event task number from 0 – 63 must be entered.

Inversion

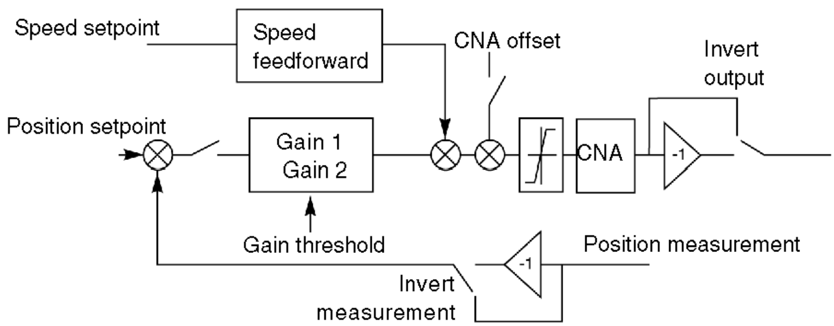
Introduction

The inversion parameters means that it is not necessary to reconnect the analog output when the axis is moving in the opposite direction to that required. These parameters define the setpoint inversion between the digital/analog converter (D/A converter) output and the variable speed controller and / or the measurement inversion (for an incremental encoder).

The measurement inversion allows you to define the direction of measurement for an encoder rotation direction.

Inversion Possibilities

The inversion possibilities are as follows:



- No Inversion,
- Inversion of measurement direction,
- Variable speed controller setpoint inversion,
- Setpoint and measurement inversion.

Using the Inversion Parameter

Defining this parameter requires a number of operations to be performed. It is advisable to save the values by default in the first instance and to modify this parameter in the adjustment phase if necessary.

Sequence Control

Introduction

The **Sequence control** parameter is used to define the action to be taken when movement without stop (G01, G11 ou G30) is not followed by a movement command.

Enabled Sequence Control

If the sequence control is enabled (default value):

- G01, G11 and G30 movements not followed by a movement command are stopped (equivalent to a STOP command) and a command refusal is generated.
  In this case, the movements without stop cannot be sequenced by synchronization on the DONE bit.
- G01, G11 and G30 movements followed by G05, G07 or G62 movements are stopped if this second command is not itself followed by a movement command.

Disabled Sequence Control

If sequence control is disabled, G01, G11 and G30 movements not followed by a movement command continue at their target speed.

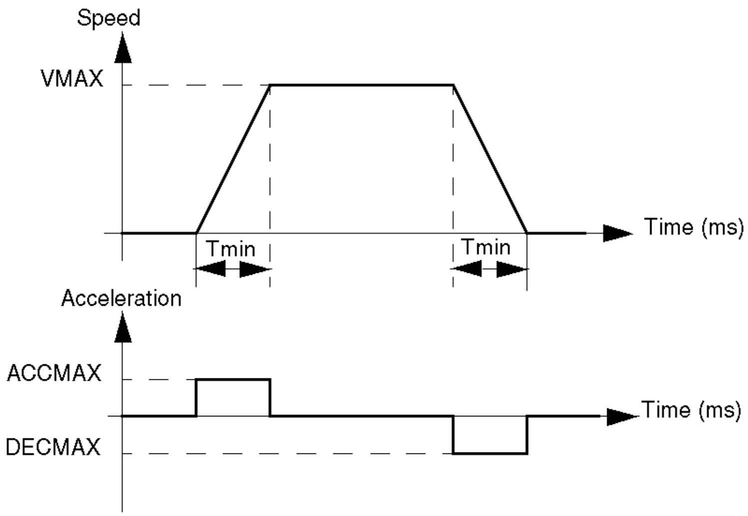
Maximum Acceleration or Deceleration

Introduction

Maximum acceleration (or deceleration) is defined by the minimum time (in ms) necessary to get from zero speed to VMAX speed.

Graphic Presentation

The following diagram shows the maximum acceleration (ACCMAX) and the maximum deceleration (DECMAX):



Limits

The minimum time Tmin is of between 16 and 10 000 ms.

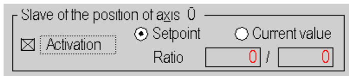
Slave of the Position of Axis 0

At a Glance

This zone is used to activate the function of slaving another axis to the position of axis 0. To do this the **Activation** box must be selected to enable the function.

Module TSX CAY •1

The axis 0 position tracking screen of the TSX CAY •1 module is as follows:

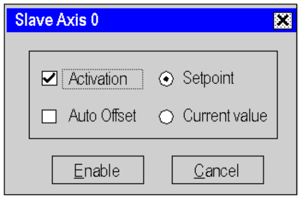


The following parameters must be defined:

Parameter	Description
Setpoint or Measurement	These 2 radio buttons allow you to define the slave axis setpoint: ● Master axis setpoint (axis 0), or ● Master axis measurement (axis 0).
Ratio	These 2 entry fields are used to define the ratio setting the slave axis setpoint value: `Slave axis setpoint = Ratio x Setpoint or Master axis measurement` (The Ratio has to be between 0.1 and 10, as each of the Ratio parameter entry fields are between 1 and 1000).

Module TSX CAY •2

The axis 0 position tracking screen of the TSX CAY •2 module is as follows:



The following parameters must be defined:

Parameter	Description
Setpoint or Measurement	These 2 radio buttons allow you to define the slave axis setpoint: ● Master axis setpoint (axis 0), or ● Master axis measurement (axis 0).
Auto Offset	This check box allows you to choose the shift register between the master and the slave: ● By slave axis learning (if the box is selected), ● By an adjustable parameter: `Slave axis setpoint = Ratio x Setpoint or Measurement of the master axis + Offset`. The Ratio and Offset parameters are theadjustment parameters *(see page 275)*.

Axis 0

The Position tracking function (slave movement) is not provided for axis 0, which can only be the master.

Event Input

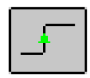
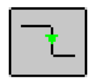
Introduction

The **Event input** selection zone is used to:

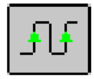
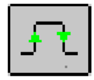
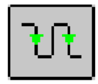
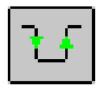
- define the type of event to be detected on the event input of the channel , for G05, G07, G10 and G11 instructions.
- define the type of event used for the storage of position function, which can store one or two positions (PREF1 and PREF2).

Storage of Position

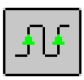
If the application does not require a length measurement, the storage possibilities are as follows:

Storage possibilities	Storage	Icon
Standard rising edge and PREF1 processing	PREF1	(1) 
Falling Edge and PREF1	PREF1	(1) 

If the application requires a length measurement, the storage possibilities are as follows:

Storage possibilities	Storage	Icon
Rising edge and PREF1, then rising edge and PREF2	PREF1, PREF2	(1) 
Rising edge and PREF1, then falling edge and PREF2	PREF1, PREF2	(1) 
Falling edge and PREF1, then falling edge and PREF2	PREF1, PREF2	(1) 
Falling edge and PREF1, then rising edge and PREF2	PREF1, PREF2	(1) 

(1) The icon shows when storage occurs. For example,



The position PREF1 is detected on the first rising edge of the event input and the position PREF2 is detected on the second rising edge of the event input.

Reference Point

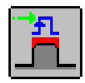
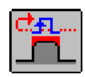
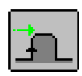
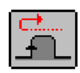
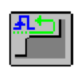
Introduction

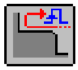
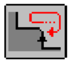
An incremental encoder does not measure the position but supplies a number of pulses proportional to a movement.In order for this movement to be transformed into a position, a known value must assigned to a particular point on the axis (usually chosen as 0).This operation is called setting a reference point.An axis which has been given a reference point, is classed as referenced.

Reference Point Field

The Reference point field defines the type and direction of the reference point (only when an incremental encoder has made the position measurement).

The type is defined by using 2 source detection inputs :zero marker input and cam input.

Possibilities	Approach speed (1)	RP Speed	Icon
short cam (2) and zero latch, + direction	F	F	(3) 
short cam (2) and zero latch, - direction	F	F/8	(3) 
Short cam (2), + direction	F	F	(3) 
Short cam (2), - direction	F	F/8	(3) 
Long cam (2) in stop and zero latch, + direction	F	F/8	(3) 

Possibilities	Approach speed (1)	RP Speed	Icon
Long cam (2) in stop and zero latch, - direction	F	F/8	(3) 
Long cam (2) in stop, + direction	F	F/8	(3) 
Long cam (2) in stop, - direction	F	F/8	(3) 

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

(2) Only short cam reference points can be used when the machine is of infinite type.

(3) The icon illustrates the reference point.

Reference Point Command

The reference point command is given:

- in automatic mode, by instruction code 14:reference point,
- in manual mode, by the SETRP command:manual reference point.

Forced Reference Point

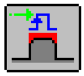
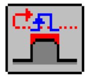
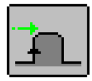
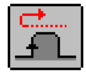
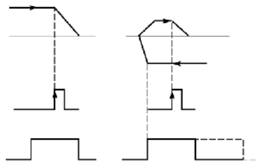
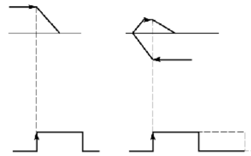
There is also a forced reference point mechanism:

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

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

Reference Point Short Cam

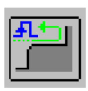
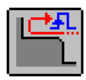
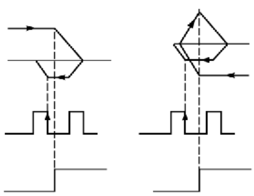
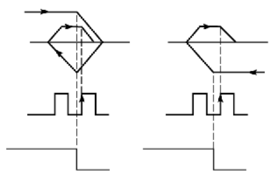
The following table provides a detailed description of short cam reference points :

Type	Short cam / zero latch		Short cam only	
Direction	+ Direction	- Direction (1)	+ Direction	- Direction (1)
Icon				
Movement Zero latch Cam				

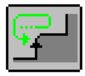
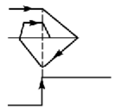
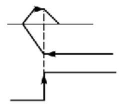
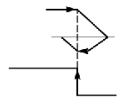
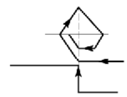
(1) or start on cam

Reference Point Long Cam

The following table provides a detailed description of the long cam and zero latch reference points:

Type	Long cam in stop / zero latch			
Direction	+ Direction (2)		- Direction (2)	
Start	Start off cam	Start on cam	Start on cam	Start off cam
Icon				
Movement Zero latch Cam				

The following table provides a detailed description of long cam only reference points:

Type	Long cam in stop			
Direction	+ Direction (2)		- Direction (2)	
Start	Start off cam	Start on cam	Start on cam	Start off cam
Icon				
Cam Movement				

(2) defines the place on the machine where the cam is situated.

TSX CAY •2 and TSX CAY 33: Zero Latch Presence Monitoring

Reference point short cam monitors the presence of the zero latch along the length of the cam.

During a short cam type of reference point setting, in + direction or – direction, with zero latch, if all the cam is covered without detecting any zero latches, the axis stops at the output of the cam and an error is signaled.The axis goes into a non-referenced state.

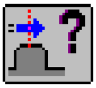
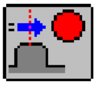
Recalibration

Introduction

This function is used to compensate for a possible slip in the measurement if the encoder used is an incremental encoder. Each time the moving part passes in front of the detector, the measurement is recalibrated to the specified value.

Recalibration Function

The recalibration possibilities are as follows:

Possibilities	Recalibration function	Icon
Recalibration function missing	Inactive	
Recalibration function and error on threshold overshoot	Active	

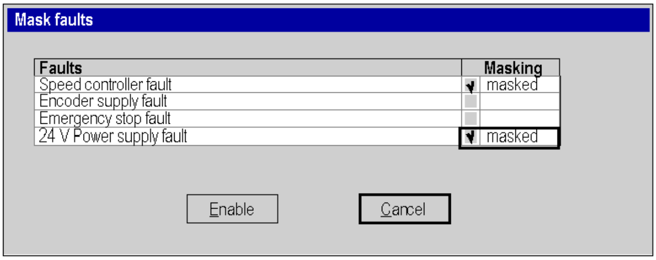
Masking of Faults

At a Glance

With a TSX CAY •2 or TSX CAY 33 module it is possible to individually mask 4 of the 8 external faults (hardware) during channel configuration (MSK_HDERR parameter).

Mask Faults Screen

The Mask faults screen is used to define the faults that you would like to mask:



The 4 faults that you can mask are as follows:

Fault	Associated parameter
Speed controller fault	DRV_FLT
Encoder supply fault	ENC_SUP
Emergency stop fault	EMG_STP
24 V Power supply fault	AUX_SUP

Associated Variables

During a power outage, the masked faults are not signaled and the associated variables are not refreshed:

- channel error bit `CH_ERROR` (%I.r.m.c.ERR),
- status word`CH_FLT`(%MWr.m.c.2).

The AX_FLT, AX_OK and HD_ERR information bits are updated without taking account of masking. All axis movement is **interrupted** by a stop and disabling of the variable speed controller.

Confirming Configuration Parameters

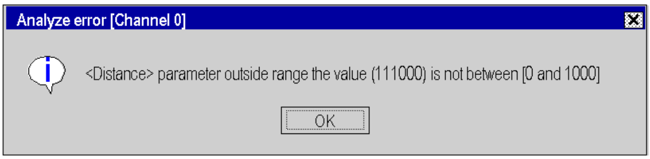
Introduction

When all the configuration parameters have been defined, the configuration must be confirmed using the **Edit → Confirm** command or by closing the module configuration window or 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 **Distance** value of the **Initial resolution** field is invalid:



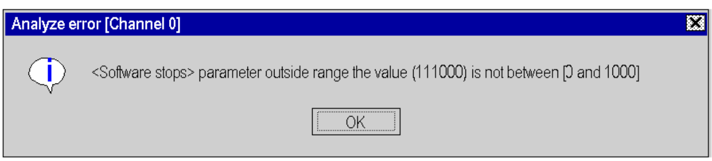
The invalid parameters must be corrected before your configuration can be confirmed.

NOTE: In the configuration screens, invalid parameters are shown in red. The grayed out parameters can not be modified because they are linked with the invalid parameters (for example, an invalid resolution prevents the lower and upper limits from being entered).

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 soft stops are incompatible:



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 10
Adjusting Independent Axes

Subject of this Section

This section describes the principle of parameter adjustment: accessing screens, description of parameters and adjustment procedure.

What Is in This Chapter?

This chapter contains the following topics:

Preliminary Operations Prior to Adjustment

Preliminary Conditions

- TSX CAY module(s) installed in PLC.
- Axis control application(s) connected to the TSX CAY module(s).
- Terminal connected to the PLC by the terminal port or the network.
- Configuration and axis control program completed and transferred to the PLC processor.
- PLC in RUN. It is advisable to inhibit 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 connections.
- Check that the movements can occur safely.
- Check that the mechanical stops are connected in accordance with safety regulations (they generally act directly on the variable speed controller supply sequence).
- Check the direction of the tachymetric dynamo connection.

Adjusting the Variable Speed Controller

The variable speed controller can be adjusted by following the manufacturer's instructions. To do this, connect up a control box instead of the axis control module.

Adjusting the current loop

Step	Action
1	Modify the maximum current value provided by the variable speed controller to a value accepted by the motor (dissipation switching) and by the mechanics (accelerator torque)
2	Adjust the current loop stability

Adjusting the speed loop

Step	Action
1	Adjust the maximum operating speed Allocate a setpoint to the variable speed controller equal to the Maximum operating voltage (UMAX)
2	Adjust the speed loop gain
3	Adjust the offset

Adjusting current limitation according to speed

Step	Action
1	Reconnect the axis control module when adjustment is complete.
2	Readjust the current loop

Adjusting the Inversion Parameter

Adjustment Procedure

Determine the inversion parameter as follows:

- Select **Debug** mode.
- Select **DIRDRIVE loop control disabled** mode.
- Acknowledge faults: **Acq.** button in the **Faults** zone.
- Enter details in succession into the **PARAM** field in accordance with the following table:
  - +100 mV (positive analog output),
  - -100 mV (negative analog output).

Action to be Taken

The following table shows the action to be taken, according to the change in position and value. This depends on the voltage (positive or negative) which has been defined for the analog output:

Analog output	Position	Measurement	Action to be taken
Positive	Increase	Increase	None (connection OK)
Positive	Increase	Decrease	Invert the measurement
Positive	Decrease	Decrease	Invert the setpoint
Positive	Decrease	Increase	Invert the setpoint and the measurement
Negative	Decrease	Decrease	None (connection OK)
Negative	Decrease	Increase	Invert the measurement
Negative	Increase	Increase	Invert the setpoint
Negative	Increase	Decrease	Invert the setpoint and the measurement

NOTE: If the offset is greater than 100 mV, you must first adjust it before proceeding.

Description of the Axis Control Module Adjustment Screen

General Points

The adjustment screen is a graphic tool intended for the adjustment *(see EcoStruxure™ Control Expert, Operating Modes)* of a module selected in a rack. It displays the current and initial parameters associated with the channels of this module, and allows these to be modified in offline and online modes.

Initial Parameters

The initial parameters are:

- Parameters entered (or defined by default) in the configuration screen in offline mode. These parameters have been confirmed in the configuration, and transferred to the PLC.
- Parameters taken into account during the last reconfiguration in online mode. These parameters cannot be modified from this screen. On the other hand, they can be updated using the current parameters.

Current Parameters

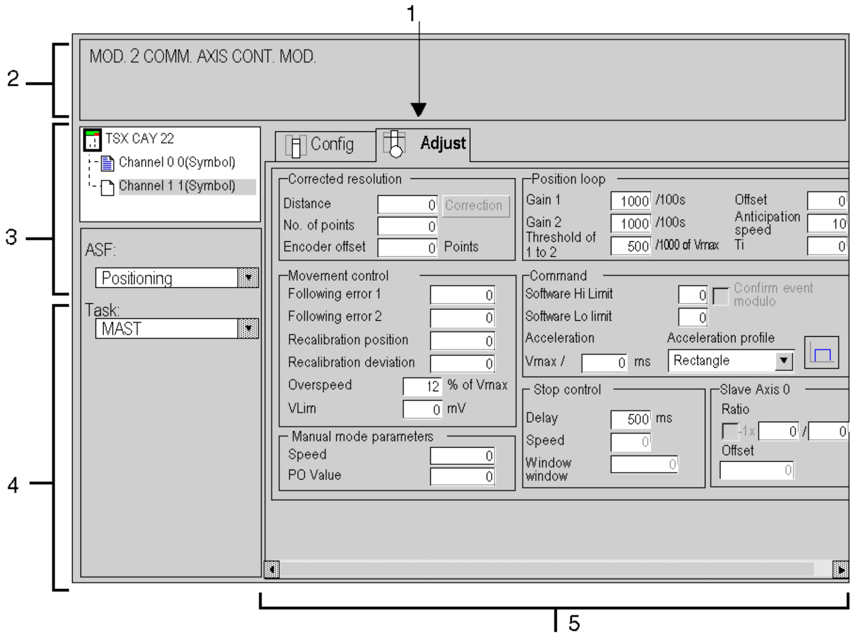
Current parameters are those which have been modified and confirmed from the adjustment screen in online mode (or by program via an explicit exchange). These parameters are replaced by the initial parameters after a cold restart.

Saving Parameters

NOTE: It is mandatory to save adjustment parameters after they have been determined.

Illustration

The diagram below shows an adjustment screen.



Description

The following table presents the various elements of the adjustment screen and their functions.

Number	Element	Function
1	Tabs	The tab in the foreground indicates the current mode (**Adjustment** in this example). Each mode can be selected using the respective tab. The available modes are: ● **Adjust** ● **Configuration** ● **Debugging** (or Diagnostics), accessible only in online mode.
2	**Module** area	Contains the abbreviated title of the module.
3	**Channel** area	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 a channel, ● To display the **Symbol**, name of the channel defined by the user (using the variable editor).
4	**General parameters** area	Allows you to choose the axis control function and the task associated with the channel: ● **Function:** axis control function among those available for the modules involved. Depending on this choice, the headings of the configuration zone may differ. By default **No** function is configured. ● **Task:** defines the (**MAST**, **FAST** or **AUX0/1**) task in which the explicit exchange objects of the channel will be exchanged.
5	**Adjustment** field	This field comprises the various values of the adjustment parameters.

Adjusting the Encoder Offset

Introduction

This parameter is concerns absolute encoders only. It is used to make the real position of the moving part coincide with the position provided by the encoder (shift from zero).

- In **direct offset** the offset value must be entered, expressed in encoder points, into the ABS_OFF parameter.
- In **assisted offset**, use the RP_HERE and SET_RP commands. *(see page 215)*

Encoder Offset

This parameter can only be modified if **direct offset** has been chosen in the configuration.

The encoder offset is defined as follows:

Encoder offset = value to be added (expressed in number of encoder points) to the measurement provided by the absolute encoder, in order to obtain the real measurement.

Limits are as follows: $-2^{n-1} + 1$ to $2^{n-1} - 1$, where n= number of absolute encoder data bits.

NOTE: The encoder offset parameter is adjusted in measurement mode (DRV_OFF).
If the absolute encoder has been declared in assisted offset, this parameter is not taken into account. The assisted offset procedure is used to avoid having to make any calculations. However, after reading READ_PARAM, SAVE_PARAM or saving parameters, the encoder offset value reflects the offset used by the channel.

Example

If the absolute encoder shows a measurement of 100 mm for position 0 and the resolution is 2 microms, the offset value is:

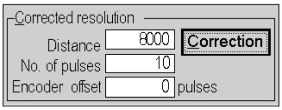
-100000 / 2 = -50000 encoder points.

Adjusting Resolution

At a Glance

This adjustment is used to compensate for error resulting firstly from imprecise entry of configuration parameter values, and secondly from imperfections in the kinematics string.

Resolution adjustment screen

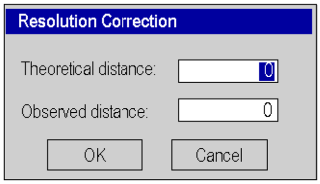


Adjustment Procedure

Perform the following operations on the TSX CAY Debug screen:

Step	Action
1	Select **Manual** mode
2	Create a manual reference point if the encoder is of incremental type
3	Choose **Theoretical distance** as the value to run which corresponds to the greatest possible range of movement: position 1 and enter this value into the **Param** field (e.g. 300,000 microns)
4	Give the **Inc-** or **Inc+** command depending on the direction of movement
5	Measure, with a sufficiently precise external device, the distance actually covered by the moving part (**Observed distance**)
6	Change to measurement mode DRV_OFF

Carry out the following operations from the Adjustment screen:

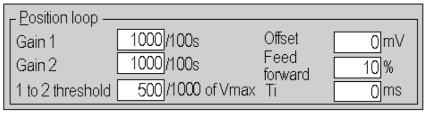
Step	Action
7	Press the **Correction** button, which displays the following dialog box:  ● Enter the distance to be covered in the **Theoretical distance** field. For example: 300,000 microns ● Enter the distance done in the **Observed distance** field. For example: 293,000 microns
8	Press **OK** to activate the automatic resolution calculations. The new **Distance** and **No. of pulses** values are then re-calculated.
9	Repeat steps 2, 3, 4 and 5. If the distance measured reveals a deviation below requirements, then adjustment is finished. If this is not the case, carry out a new correction (steps 7 and 8).

NOTE: If you modify the **Initial resolution** and **VMAX** parameter values after adjusting the resolution, it is mandatory to resume the adjustment.
In general, modifying any configuration in offline mode means that the resolution must be re-adjusted in online mode.

Description of Loop Control Parameters

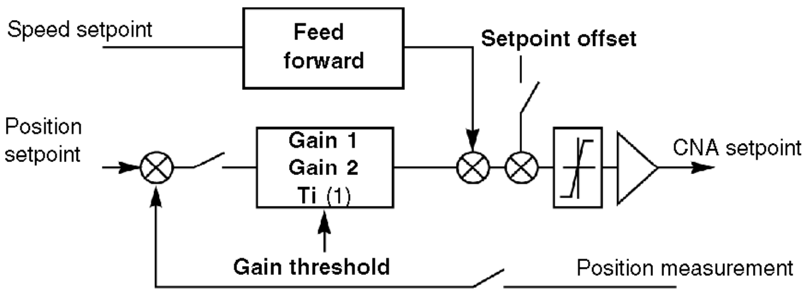
At a Glance

The following parameters are used to adjust the position loop:



Position Control Loop

The following synoptic shows the position control loop:



(1) **Integral action** active on stop only

Creating References

Position and speed references are created according to the movement which is required (e.g. speed, target position), and the parameters which have been defined in the Adjustment screen.

Description of Gain Parameters

This table describes the gain parameters:

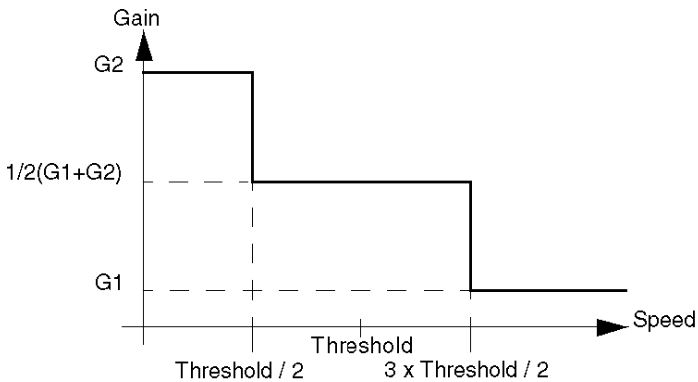
Parameter	Meaning
Gain 1 and Gain 2	Position loop gains (from 50 to 12,000 1/100 s). By default: Gain 1 = Gain 2 = 1000 1/100 s The axis control module uses the 2 gain values: ● Gain 1: gain value for high operating speeds. This value is used to avoid overshoot and instability. ● Gain 2: gain value for low operating speeds. This value is used to obtain very slight position deviations.
Threshold of 1 to 2	Gain switching threshold (from 20 to 500% of VMAX) By default: Gain threshold = 500% of VMAX

Position Gain

The applied position gain is as follows:

- If the current speed > = 3 x Threshold / 2, Gain = Gain 1
- If 3 x Threshold / 2 > current speed> = Threshold / 2, Gain = (Gain 1 + Gain 2) / 2
- If current speed < Threshold / 2, Gain = Gain 2

This drawing shows gain value according to current speed:



Proportional Gain Coefficient

The axis control module calculates the **KP proportional gain coefficient** on the basis of the **Gain** adjustment: KP = C x UMAX x Gain

C = constant, and UMAX = variable speed controller setpoint value in order to obtain the speed VMAX (UMAX < 9 V).

NOTE: In general, Gain 1 = Gain 2.

Description of Feed Forward Parameter

This table shows the feed forward parameter:

Parameter	Meaning
Feed forward	Feed forward adjustment coefficient (from 0 to 100%). By default: Feed forward = 10%

The feed forward coefficient is expressed as a percentage. 100% corresponds to the value, which can completely absorb the position error at constant speed for a variable speed controller without continuous error.

When the feed forward coefficient rises, the position error is reduced. However this results in a risk of overrun, including when approaching the breakpoint. A compromise must therefore be found.

NOTE: In certain cases, the position error crosses a minimum threshold with a change of sign if the feed forward rises.

Description of Offset Parameter

This table describes the offset parameter:

Parameter	Meaning
Offset	Offset adds to the analog output value calculated by the loop (from –250 mV to +250 mV). By default: Offset = 0 mV

Adjusting Loop Control Parameters

At a Glance

Adjusting the controls requires you to enter particular values for certain operating parameters. The values of the other parameters are defined by the application.

To do this, enter these parameters in the adjustment screens, then confirm in order to send the parameters to the axis control module.

Initial Operation

This operation involves entering a forced reference point in Manual mode.

The forced reference point allows there to be a referenced axis from start-up, and thus enables the following checks and functions:

- software limits,
- clearing outside software limits.

NOTE: Operation will only be correct if the direction of the moving part is the same as that of the measurement.

Forced Reference Point Procedure

To create a forced reference point, carry out the following operations:

Step	Action
1	Select the TSX CAY Debug screen
2	Select **Manual** mode
3	Acknowledge faults using the **Ack.** command
4	Using an external device, measure the position of the moving part in relation to the reference point cam (approximate measurement)
5	Create a forced reference point: ● enter the measured value with its sign as the source position value in the **Param** field, ● select the **Forced reference point** command.

Adjusting High-Speed Gain

To determine the value of the **Gain 2** parameter, carry out the following operations: It is assumed that the moving part has an inertia equal to the maximum value encountered in the application.

Step	Action
1	Make movements from position 1 to position 2 and vice versa. To do this: ● select an average speed, using the **speed** parameter in manual mode, ● enter the movement value in the **Param** field, ● activate the **Inc+** (position 1), then **Inc-** (position 2) commands in succession,
2	Check following error when the moving part is at a stabilized speed.
3	Adjust **Gain 2** for an acceptable deviation, while maintaining an appropriate stability (otherwise, check the machine definition). For each new **Gain 2** value entered, use the same value for **Gain 1** and transfer by confirming at the adjustment screen

Adjusting Low-Speed Gain

To determine the value of the **Gain 1** parameter, carry out the following operations: This adjustment must be made for machines that involve friction. Otherwise, retain the value of **Gain 2** for parameter **Gain 1**. To obtain greater gain at low speed, set **Gain 1** to a value greater than **Gain 2** then transfer the values by validating the adjustment screen:

Step	Action
1	Make movements from position 1 to position 2 and vice versa. To do this: ● select a very low movement speed: by choosing a low **CMV** coefficient value, ● enter a low movement value in the **Param** field, ● activate the **Inc+** (position 1), then **Inc-** (position 2) commands in succession,
2	Check following error when the moving part is stationary
3	Adjust **Gain 1** for an acceptable deviation, while maintaining an appropriate stability. For each new **Gain 1** value entered, transfer by confirming at the adjustment screen

Adjusting the Gain Threshold

The gain threshold must be set at a speed which overcomes friction.

Adjusting Feed Forward Gain

Perform the following to determine speed feed forward gain:

Step	Action
1	Make movements at speed **VMAX** from position 1 to position 2 and vice versa. To do this: ● select a high movement speed: while choosing a high **CMV** coefficient value, ● enter a movement value in the **Param** field, ● activate the **Inc+** (position 1), then **Inc-** (position 2) commands in succession,
2	Adjust the **feed forward** for the value and sign of the error required

NOTE: If the overshoot is too large, you may wish to reduce **feed forward**slightly.

Offset Adjustment

When the moving part is stationary, select loop control disabled mode **DIRDRIVE**.

Adjust the offset in the window from –250 mV to +250 mV to cancel out any slipping of the moving part.

Adjusting Integral Action

For a TSX CAY 22/42 or TSX CAY 33 module, integral action is used to compensate for the different offsets in the string (e.g. module, drive, motor, mechanical equipment) as well as drift. The **Ti** position loop parameter is used to achieve this compensation.

The gain is only active when the axis is theoretically stationary (theoretical speed is zero, following error absorption phase). Is it active in automatic and manual modes in the absence of a blocking fault on the axis (**AX_OK=1**). It is inactive in the automatic modes **EXT_CMD** and **SLAVE**.

The principle is to add an additional continuous action, which is updated during stop phases.

The integral action is expressed in ms, in the interval [100, 5000] ms. By default the value 0 indicates that there is no integral action.

Description of Movement Control Parameters

Description of Following Error Parameters

This table describes the following error parameters:

Parameter	Indication
Following error 1	Moving part stopped by critical deviation between calculated position (setpoint) and measured position of moving part. By default Following error 1 = (LMAX – LMIN)/100 Critical error: 0 to (SL_MAX - SL_MIN)/4 Following error 1 = 0: no control
Following error 2	Deviation between calculated position (setpoint) and measured position of moving part, which only causes a fault to be signalled. By default Following error 2 = (LMAX – LMIN)/100 Preventive error: 0 to (SL_MAX - SL_MIN)/4 Following error 2 = 0: no control

Description of Recalibration Parameters

This table describes recalibration parameters:

Parameter	Indication
Recalibration position	Value that measured position must have during a recalibration event. By default Recalibration position = (LMAX - LMIN)/4 + LMIN (if it is a configured recalibration function) Limits: SL_MIN + TW to SL_MAX - TW, where TW = tolerance for target window control Recalibration position = 0 : no control
Recalibration deviation	Maximum gap between recalibration position and measured position of moving part during a recalibration event. A larger deviation will trigger a recalibration. By default Recalibration deviation = (LMAX - LMIN)/100 + LMIN (if it is a configured recalibration function) Static error: 0 to DMAX1/2 Recalibration deviation = 0: no control

Description of Overspeed Parameter

This table describes the overspeed parameter:

Parameter	Indication
Overspeed	Fault threshold for measured overspeed, expressed as a % of VMAX. By default Overspeed = 10% Overspeed: 0 to 20% Overspeed = 0: no control

Description of VLIM Parameter

This table describes the VLIM parameter:

Parameter	Indication
VLIM	Detection threshold for movement control. Unit: mV Limits: 0 to 9000 VLIM = 0: no control

Description of Command Parameters

Description of Soft Stop Parameters

This table describes soft stop parameters. These parameters can only be accessed if the axis is limited.

Parameter	Significance
Software hi limit Software lo limit	Hi and lo limits of the position measurement that the moving part must not exceed. In the case of overflows, the moving part stops and signals a soft stop fault. By default for the TSX CAY *2 and 33: - SL_MIN = LMIN and SL_MAX = LMAX - LMIN <= SL_MIN < SL_MAX <= LMAX and - SL_MAX - SL_MIN > RESOL x 256 By default for TSX CAY *1: - SL_MAX - SL_MIN > RESOL x 2^{15}

Description of Modulo Parameter

This table describes the modulo parameter. In the case of an infinite axis, this parameter can be accessed with a TSX CAY 22/42 or TSX CAY 33 module.

Parameter	Significance
Modulo	Measurement area, in the case of an infinite run-time machine. For infinite axes, adjustment must be less than or equal to the modulo defined in the **Max modulo** field during configuration. By default Modulo = Max modulo Limits: Modulo <= Max modulo

Description of the Acceleration Parameter

This table describes the acceleration parameter:

Parameter	Significance
Acceleration	Value of acceleration and deceleration. This is defined by the Taccrec time (in ms) which is used to change from zero speed to VMAX speed, in the case of a Rectangle profile. In the case of a different profile, it is defined by: Tacc = Taccrec x (2t1 + t2) / (t1 + t2) where, Taccrec: acceleration for a rectangle profile (this value must be entered), t1 and t2 are defined by the acceleration profile (see below). By default Taccrec = TACCMIN Limits: TACCMIN <= Tacc < 10000 ms (where TACCMIN = maximum acceleration)

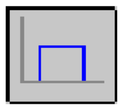
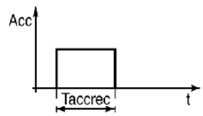
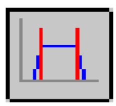
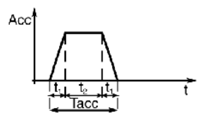
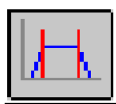
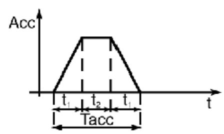
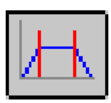
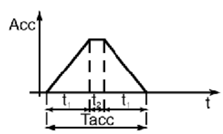
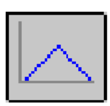
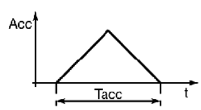
Description of the Acceleration Profile Parameter

This table describes the acceleration profile parameter:

Parameter	Significance
Acceleration profile	Acceleration law applied to moving part. By default: Rectangle

Acceleration Profiles

The following acceleration profiles may be used:

Acceleration profile	Icon	Description	
Rectangle			t1 = 0 Tacc = Taccrec
Trapezoid 1			t2 = 3 t1 Tacc = 1,25 Taccrec
Trapezoid 2			t1 = t2 Tacc = 1.5 Taccrec
Trapezoid 3			t1 = 3 t2 Tacc = 1.75 Taccrec
Triangle			Tacc = 2 Taccrec

Description of Stop Control Parameters

Description of Delay Parameter

This parameter describes the Delay parameter:

Parameter	Indication
Delay	As soon as the speed setpoint value calculated by the module is equal to 0, the module activates a time-out (equal to the Delay parameter). When this time-out has elapsed, the axis control module compares the moving part’s measured speed with the stop speed. By default Delay = 500 ms Stop delay: 0 to 10000 ms Delay = 0 no control of stop fault

Description of Stop Speed Parameter

This parameter describes the stop speed parameter:

Parameter	Indication
Stop speed	Speed at which moving part is considered to be at a stop. Stop speed: 0 to VMAX/10, limited to 30000

Description of Target Window Parameter

This parameter describes the Target window parameter

Parameter	Indication
Target window	Tolerance for position reached by module, after time-out defined by the Delay parameter. ● For a limited axis: ❍ By default Target window = (LMAX – LMIN)/100 ❍ Limits: 0 to (SL_MAX - SL_MIN)/20 ❍ Target window = 0: no control ● For an unlimited axis: ❍ By default Target window = Max modulo/100 Limits: 0 to Modulo /20

Adjustment of Monitoring Parameters

Parameter Adjustment Procedure

To adjust monitoring parameters, the following procedure must be followed :

Step	Action
1	Enter the monitoring parameter values required, then confirm.
2	From the Debug screen: ● Select **Manual** mode, ● select a high movement speed: ● Move from position 1 to position 2 and vice versa. To do this : ❍ enter a movement value in the **Param** field, ❍ select **Inc+** (position 1), then **Inc-** (position 2) commands in succession. The module must not become faulty. Check that there are no **Axis** faults (the **DIAG** button shows more detail).
3	If a fault is detected: ● increase the parameter values (i.e. greater tolerances) ● or re-adjust then adapt the loop control parameters.
4	Return to the adjustment screen and adjust the following parameters: **Following error 1** and **Following error 2**.
5	Adjust the **Stop speed** and **Stop delay** parameters. The speed must be less than **Stop speed** at the end of the **Stop delay** . The **Stop delay** is counted in relation to the moment where position reference reaches the value of the requested position.
6	Adjust the **Target window** parameter.
7	Adjust the **Overspeed** parameter. For this adjustment, select a movement speed equal to VMAX.

Description of Manual Mode Parameters

Description of Speed parameter

This table describes the speed parameter:

Parameter	Meaning
Speed	Speed of moving part in manual mode. Just as in automatic mode, the real movement speed is modulated by the CMV modulation coefficient. Actual speed setpoint = Speed x CMV/100 By default Speed = VMAX/2 Limits: 10 to VMAX

Description of RP Value parameter

This table describes the Reference point value parameter:

Parameter	Meaning
RP Value	Value set in current position during manual reference point setting. ● For a limited axis: ❍ By default RP Value = (SL_MAX - SL_MIN)/4 + SLMIN ❍ Limits: SL_MIN + 1 to SL_MAX - 1 ● For an unlimited axis: ❍ By default RP Value = Modulo / 4 ❍ Limits: 1 to Modulo - 1

Parameters Associated with Master / Slave Axes

TSX CAY •2: Link between Master and Slave Axes

A ratio and an offset define the link between the master axis and the slave axis.

`ConsignePositionEsclave = PositionMaître x (Ratio1 / Ratio2) + SlaveOffset`

If the slave is an infinite type machine, the Modulo operator is applied in the calculations of the slave position setpoint. Modulo value: ModuloValue is defined by %Mdxy.i.33.

TSX CAY •2: Ratio

The ratio defined by: `Ratio = Ratio1 / Ratio2` is adjustable.

Dynamic ratio values are set between 0.01 and 100. The ratio can be negative.

TSX CAY •2: Offset

The `SlaveOffset` parameter corresponds to the position offset value between master and slave.

If the SlaveOffset parameter is zero, the slave axis setpoint = Ratio x master axis Setpoint or Measurement.

The offset value must be between -2^{30-1} and 2^{30-1} and the resulting `ConsignePositio-nEsclave` parameter must stay within the software limits of the slave.

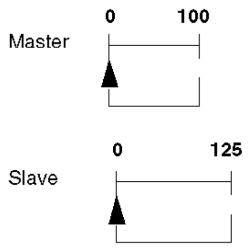
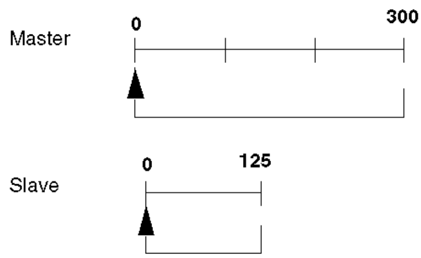
TSX CAY •1

TSX CAY •1 modules are used to carry out master / slave applications, with non-modifiable ratio, and without offset shift.

NOTE: In the case of 2 infinite axes (master and slave):
Modulo $_{\text{Master}}$ x Ratio = Modulo $_{\text{Slave}}$ x k where k = whole number
k represents the number of revolutions made by the slave in the time taken for the master to make one revolution.

Examples

Examples

Ratio=1.25 k=1 (for 1 master revolution, the slave makes 1 revolution)	
Ratio=1.25 k=3 (for 1 master revolution, the slave makes 3 revolutions)	

Confirming Adjustment Parameters

Introduction

When the adjustment parameters have been entered, they must be confirmed by using the **Edit → Confirm** command, or by activating the icon

Parameters Outside Limits

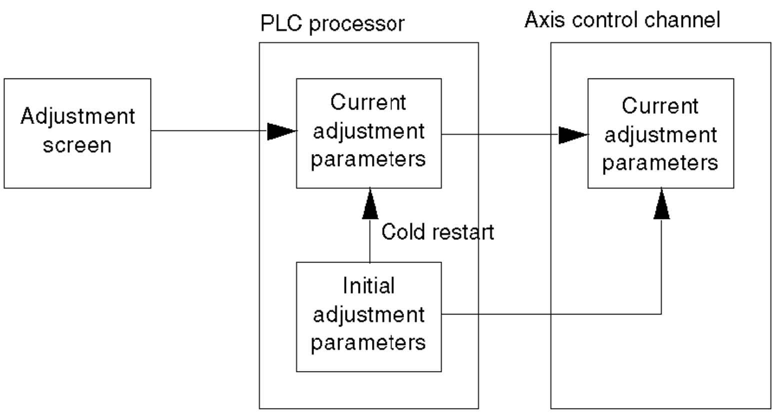
If one or several parameter values are outside the permitted limits, an error message appears indicating the parameter concerned.

The faulty parameter(s) must be corrected and then confirmed.

No Modification of Configuration Parameters

If configuration parameters have not been changed, modifying the adjustment parameters will not affect axis operation, but will modify its behaviour.

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

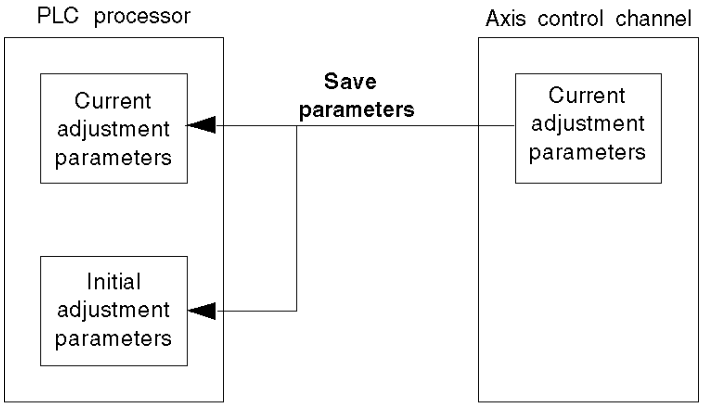


NOTE: On cold restart, the current parameters are replaced by the initial parameters. The initial parameters can be updated by using the save command, or by reconfiguration.

Saving / Restoring Adjustment Parameters

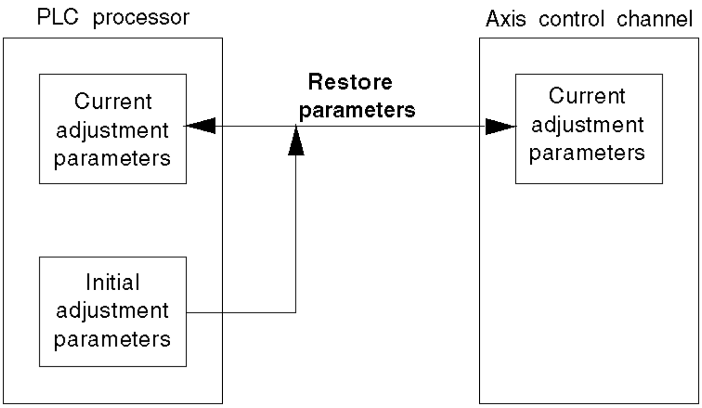
Saving Parameters

To save the current parameters (update initial parameters), activate the **Utilities** → **Save parameters** command.



Restoring Parameters

To replace current parameters with the initial values, activate the **Utilities** → **Restore parameters** command.

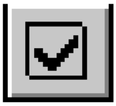


NOTE: The RESTORE_PARAM instruction enables the application program to perform this restore operation. The restore function can also be performed automatically on a cold restart.

Online Reconfiguration

At a Glance

When the configuration parameters have been modified, they must be confirmed with the **Edit → Confirm** command, or by closing the configuration window or activating the icon

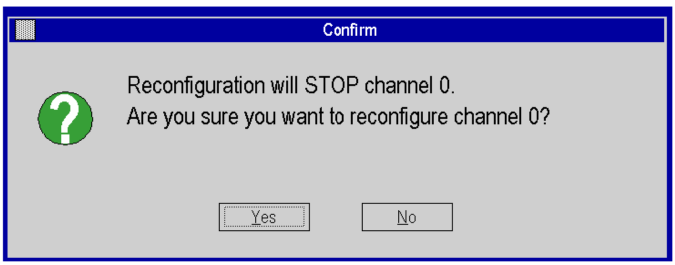


Parameters Which Can Be Modified in Online Mode

Only those parameters which are not grayed-out can be modified in online mode. Other parameters (e.g. resolution, encoder type, activating an event task) must be modified in offline mode. However upon reconfiguration, the corrected resolution becomes the initial resolution.

Stopping a Movement in Progress

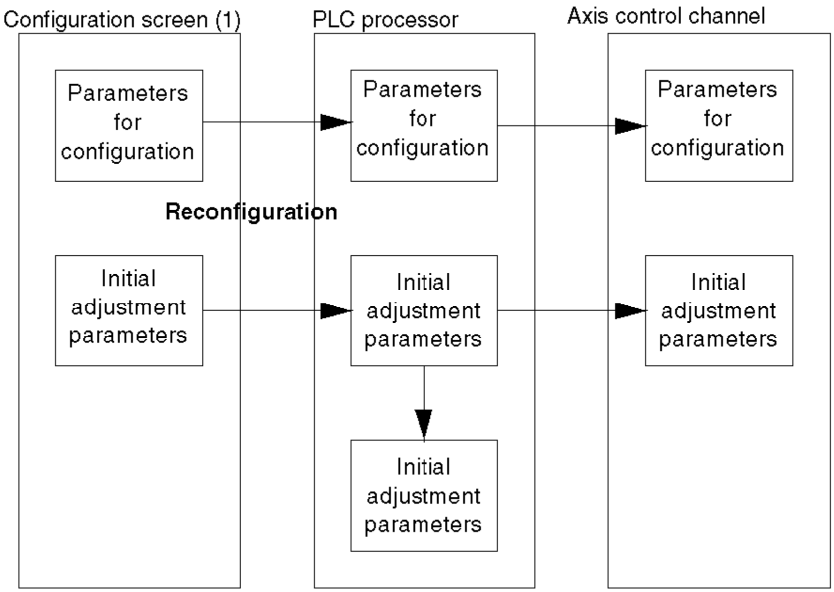
Any reconfiguration in online mode stops the operation of the channel concerned. This therefore also stops the movement in progress. This is indicated by a dialog box:



NOTE: Reconfiguring in online mode is not available in V1.0.

Exchanging Parameters Upon Reconfiguration

The following diagram shows how to exchange parameters during reconfiguration in online mode:



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

Chapter 11
Debugging an Independent Axis Control Program

Subject of this Section

This section describes the axis control channel debugging functions, in the different modes: Measurement, Manual, Loop control disabled, Automatic. It also describes the diagnostics screen which gives access to possible faults.

What Is in This Chapter?

This chapter contains the following topics:

Debugging Principles

At a Glance

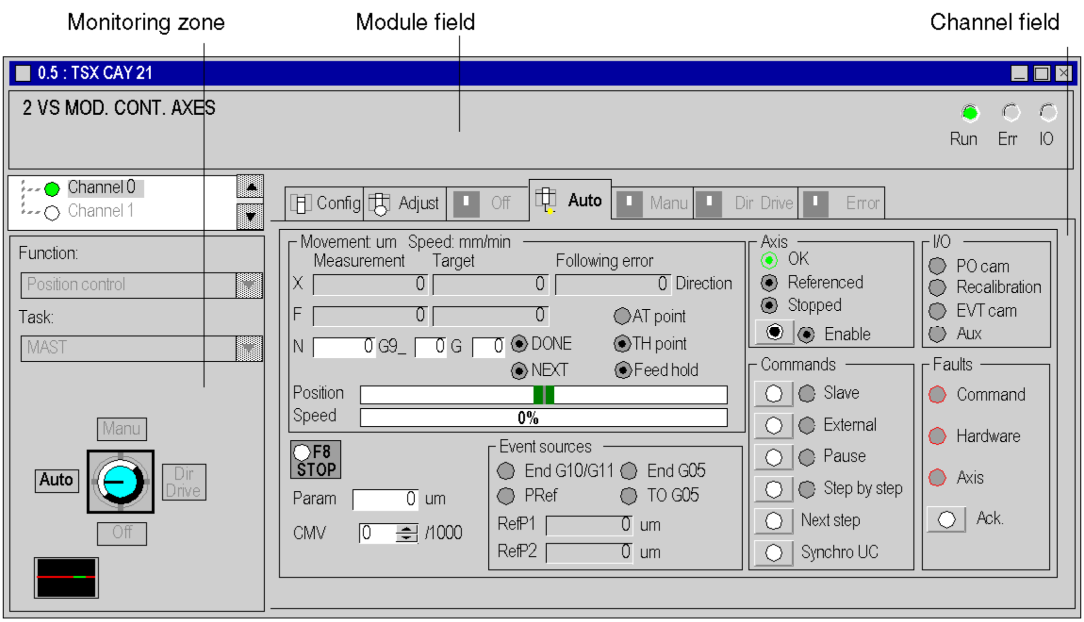
The axis control capabilities, integrated into Control Expert programming, use Control Expert debugging functions.

Reminder of Capabilities Offered by Control Expert

- Program viewing and animation in real time.
  For example, if each movement is programmed in one step in Grafcet, you will be able to easily determine the movement in progress.
- Setting of breakpoints and program execution: cycle by cycle, network by network, or sequence by sequence.
- Accessing animation tables. This allows you to display the status words and bits and control the command bits for the `SMOVE` function. Is it also possible to force bit objects and block Grafcet changes.

Debugging Screen

Control Expert software also offers a debug screen specially designed for TSX CAY module which provides access to all essential information and commands:



This screen is composed of 3 zones:

- Module field.
- Channel field.
- A moving part and program monitoring zone. This zone depends on the operating mode that has been selected via the mode switch: Automatic (Auto), Manual (Manu), Loop control disabled (Dir Drive) or Measurement (Off) modes.

The Debug Screen User Interface

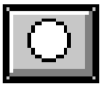
Accessing the Debug Screen

It is not possible to access the debug screen if the terminal is in **online mode**.

If this is the case, access the debug screen as follows:

- Select the Configuration Editor.
- Select and confirm (or double-click on) the rack position which contains the axis control module.
- In online mode, the debug screen is viewed by default.

Command Buttons

The  command buttons work in the following way:

- For status commands (except JOG commands):
  pressing then releasing a button activates the associated command. The button's internal LED is lit when this command is taken into account (the corresponding %Q command bit is set to 1).
  pressing then releasing the button a second time deactivates the command. The button's internal LED is off when this command is taken into account (the corresponding %Q command bit is set to 0).
- For commands on edge:
  the command is activated as soon as the button is pressed and released. The button's internal LED lights up then goes out automatically.

The LED next to a button indicates when the module has taken the command into account.

Entry Field

Any value entered into an entry field must be confirmed by the key

Using the Keyboard

The keyboard can be used to browse through the screens or to activate a command:

Keys	Action
Shift F2	Used to pass from one zone to another
Tab	Used to pass from one set of commands to another within the same zone
Arrow keys	Used to pass from one command to another within a set of commands
Space bar	Used to activate or deactivate a command

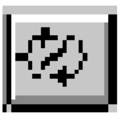
Program Conflicts

Conflicts may occur between the Control Expert program which carries out commands or writes variables, and the commands executed from the debug screen. In any case, the active command will be the one most recently taken into account.

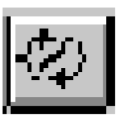
Animation

It is possible to stop animation in the display zones:

- The **Utilities** → **Stop Animation** command stops animation in the display zones, and inhibits the command buttons. This function can also be carried out by using the icon



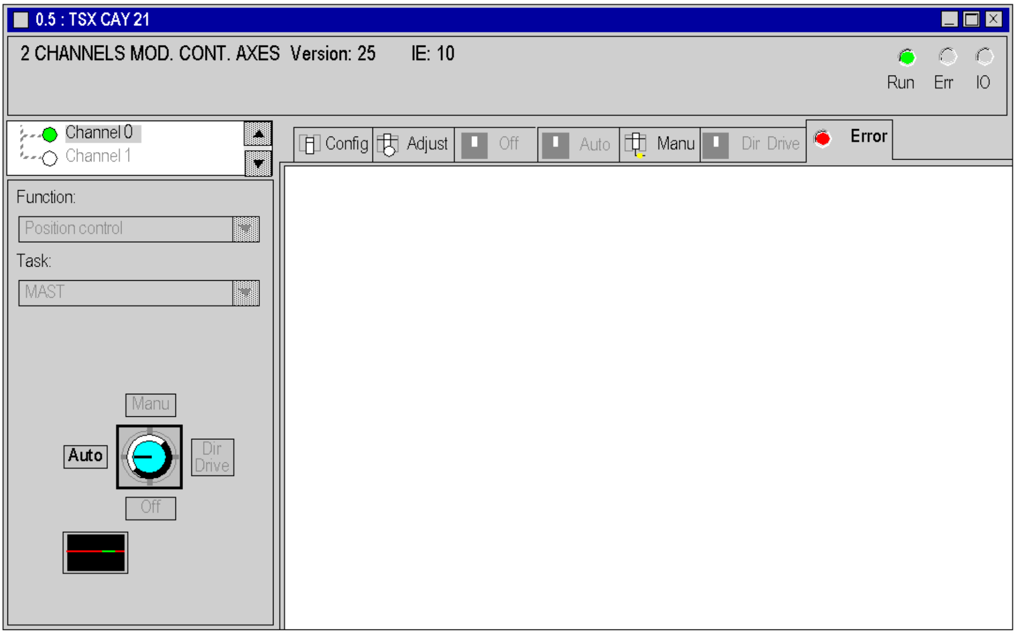
- The **Utilities** → **Animate** command reactivates animation. The following icon can also be used:



Description of Debug Screens

At a Glance

The debug screens have a common part as seen below:



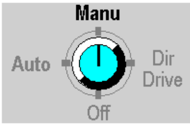
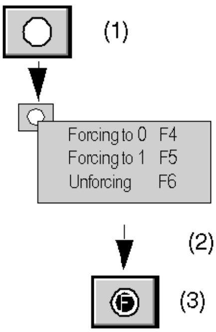
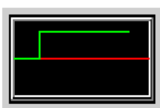
Module Field

This table describes the module zone:

LED	State	Indication
RUN	Lit	Module in operation
ERR	Lit Blinking	Module inoperative Communication fault
I/O	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs)
DIAG	Lit	Faulty module. On pressing the associated tab, a module diagnostics window will appear to indicate the source of the fault.

Channel Field

In addition to the **Axis choice** and **Function** fields (common to all screens), this zone includes the following commands and LEDs:

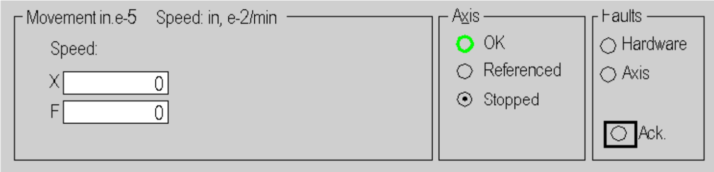
Command	Function
	Operating mode selector button. If you wish to change operating mode, click on the name of the new mode to be selected (or click as many times as necessary on the button). Using the keyboard, select the button with the **Tab** key, then press as many times as is necessary on the **Space Bar**. It is also possible to access operating modes using the **View** menu. When the selected mode is taken into account by the module, the movement monitoring zone for that mode is displayed. **Caution**: Even though selected, the module channel may not take the mode into account (e.g. if the PLC is in STOP mode).
	Forcing command menu. If an object can be forced, a right click on the corresponding button (1) displays a menu (2) which allows access to the forcing commands: **Forcing to 0**, **Forcing to 1** or **Unforcing**. After clicking on a command to select it, forcing is applied and the forcing status is indicated by the button (3): ● F for forcing to 0 ● F in inverse video for forcing to 1. The **Global unforcing** button in the module zone is used to unforce a set of forced objects.
	This zone displays a summary (with background on certain points similar to an oscilloscope) of the value of the analog output. This value is between +10 V and –10 V.
CHi	Lit: Configured non-faulty axis (channel). Blinking: Faulty axis. Off: Non-configured axis.
DIAG	Lit: Channel fault. By pressing on the button associated with this LED, a dialog box appears, specifying the source of the fault *(see page 300)*.

Measurement Mode (Off)

At a Glance

In this mode, the axis control channel only reports back information on position and current speeds. Movement of the moving part is not monitored.

The position loop is inoperative and the variable speed controller enable relay is unlocked, whatever the state of the variable speed controller enable bit ENABLE (%Qxy.i.9).



Description of Movement / Speed Field

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

Display zone	Description
X	Displays the moving part position using the measurement unit defined in the configuration.
F	Displays the moving part speed using the measurement unit defined in the configuration.

Description of Axis Field

This table shows the display zones for the Axis field:

LED	State	Indication
OK	Lit	Axis in operational state (no blocking fault)
Referenced	Lit	Referenced axis
Stopped	Lit	Moving part stationary

Description of Faults Field

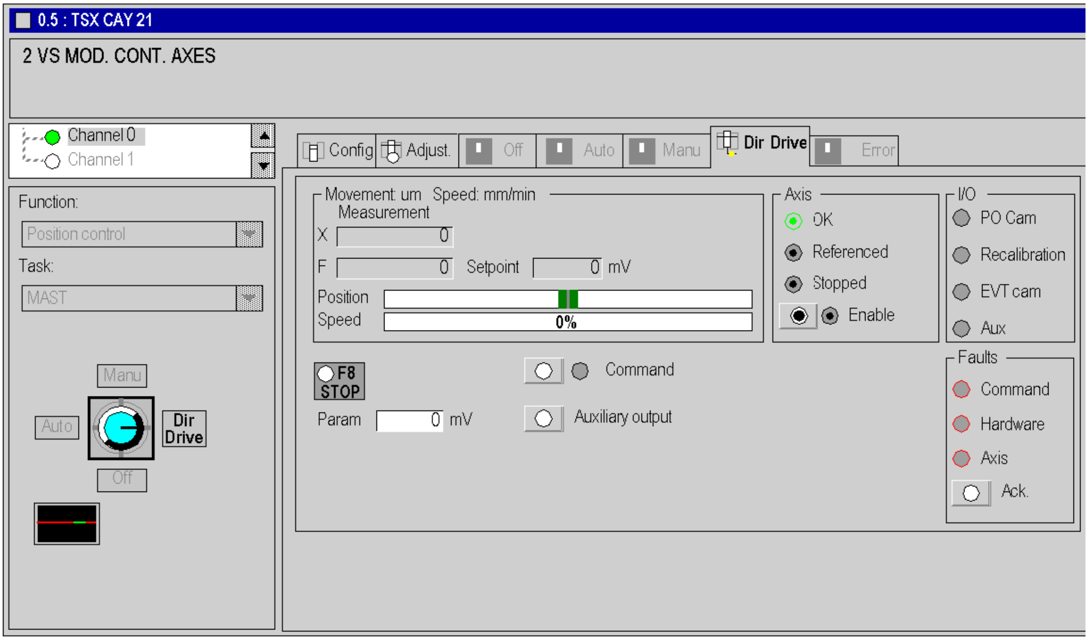
This table shows the display and command zones for the Faults field:

LED / Button	State	Indication
Hardware	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs, etc.)
Axis	Lit	Application fault (e.g. following error, software limits, etc.)
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

Loop Control Disabled Mode (Dir Drive)

At a Glance

The loop control disabled mode is used to directly control the movement of the moving part, with the control loop inoperative.



Description of Movement / Speed Field

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

Display zone	Description
X	Displays the moving part position using the measurement unit defined in the configuration.
F	Displays the moving part speed using the measurement unit defined in the configuration.
Setpoint	Displays (in mV) the setpoint value applied to the analog output
Position	This bar chart shows the progress of the moving part between the soft stops. The bar chart is colored green, and becomes red if there is a soft stop overshoot
Speed	The bar chart expresses the speed of the moving part in relation to maximum speed as a percentage. The bar chart is colored green, and becomes red if the maximum speed is exceeded

Description of Axis Field

This table shows the display and command zones for the Axis field:

LED / Button	State	Indication
OK	Lit	Axis in operational state (no blocking fault)
Referenced	Lit	Referenced axis
Stopped	Lit	Moving part stationary
Enable	/	This button is used to control the variable speed controller enable relay

Description of I/O Field

This table shows the display zones for the I/O field:

LED	Indication
PO Cam	Signal state (0 or 1) on **Reference point** input
Recalibration	Signal state (0 or 1) on **Recalibration** input
Event Cam	Signal state (0 or 1) on **Event** input
Aux	Signal state (0 or 1) on auxiliary output

1 = LED lit, 0 = LED off

Description of Commands

This table describes command buttons:

Command	Description
STOP	Sets analog output to 0 while taking deceleration into account
Param	Used to enter setpoint value from –9000 mV to +9000 mV
Command	Applies the value entered into the **Param** field to the analog output
Auxiliary output	Sets auxiliary output to 1 or 0

Description of Faults Field

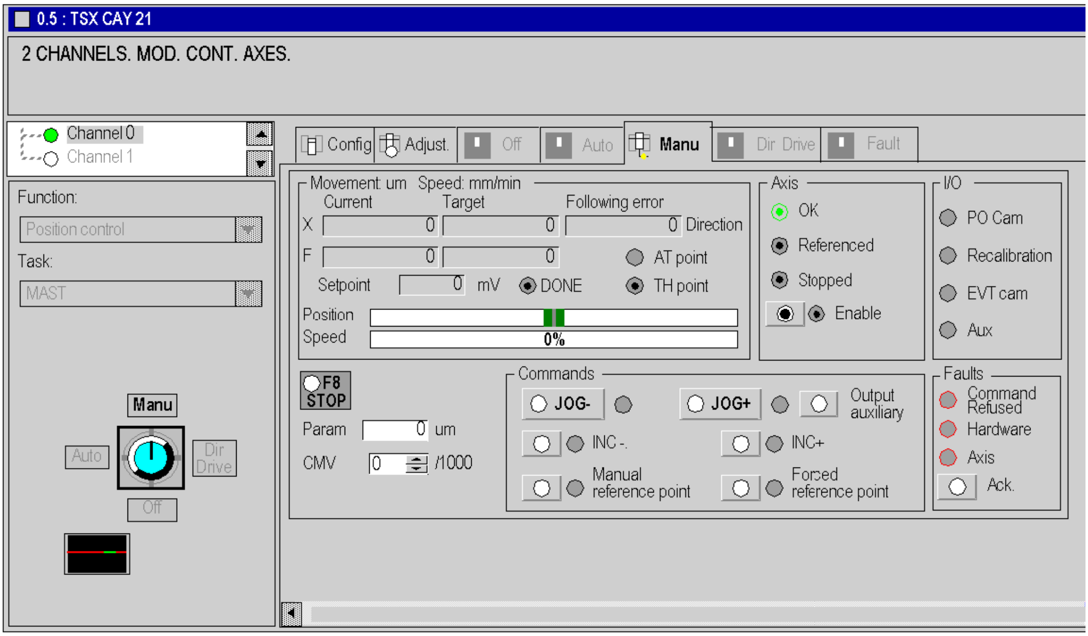
This table shows the display and command zones for the Faults field:

LED / Button	State	Indication
Command Refused	Lit	Last command refused
Hardware	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs, etc.)
Axis	Lit	Application fault (e.g. following error, software limits, etc.)
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

Manual Mode (Manu)

At a Glance

Manual mode is used to directly control the movement of a moving part from the debug screen. To do this, the JOG+, JOG-, INC+ commands must be used.



Description of Movement / Speed Field

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

Display zone	Description
X Current	Displays the moving part position using the measurement unit defined in the configuration.
X Target	Displays the moving part position setpoint (target position)
X Following error	Displays the deviation between the setpoint position calculated and the actual position of the moving part (following error).
F Current	Displays the moving part speed using the measurement unit defined in the configuration.
F Target	Displays the setpoint speed of the moving part: target speed (manual speed modified by CMV coefficient)

Display zone	Description
Setpoint	Displays (in mV) the setpoint value applied to the analog output
Position	The bar chart shows the progress of the moving part within the limits defined in the configuration screen. The bar chart is colored green, and becomes red if the limits are exceeded
Speed	The bar chart expresses the speed of the moving part in relation to maximum speed as a percentage. The bar chart is colored green, and becomes red if the maximum speed is exceeded

This table shows indicators for the Movement/ Speed field:

Indicator	State	Indication
+ Direction - Direction	/	Indicates that the part is moving in a positive direction Indicates that the part is moving in a negative direction
AT Point	Lit	Indicates that the movement in progress has finished, and that the moving part is in the target window (with INC+ or INC- commands)
DONE	Lit	Indicates that the movement in progress has finished
TH Point	Lit	Indicates that the theoretical setpoint has been reached

Description of Axis Field

This table shows the display and command zones for the Axis field:

LED / Button	State	Indication
OK	Lit	Axis in operational state (no blocking fault)
Referenced	Lit	Referenced axis
Stopped	Lit	Moving part stationary
Enable	/	This button is used to control the variable speed controller enable relay

Description of I/O Field

This table shows the display zones for the I/O field:

LED	Indication
PO Cam	Signal state (0 or 1) on **Reference point** input
Recalibration	Signal state (0 or 1) on **Recalibration** input
Event Cam	Signal state (0 or 1) on **Event** input
Aux	Signal state (0 or 1) on auxiliary output

1 = LED lit, 0 = LED off

Description of Commands

This table describes the command zone:

Command	Description
STOP	Stops the moving part according to deceleration defined in the configuration
Param	Used to enter the value of either an incremental movement (INC+ or INC- command) or a forced reference point
CMV	Used to enter a value from 0 to 2000 which determines the speed multiplier coefficient (0.000 to 2000 in intervals of 1/1000)

Description of Commands Field

This table describes the buttons in the Commands field:

Command	Description
JOG-	Command to move view in a negative direction (1)
JOG+	Command to move view in a positive direction (1)
INC-	Incremental movement command in negative direction for a distance defined in the **Param** field
INC+	Incremental movement command in positive direction for a distance defined in the **Param** field
Manual reference point	With an incremental encoder, manual reference point and search order. The current position takes on the value of the **PO Value** defined in the adjustment screen, with the reference point found in accordance with the type defined in the configuration.
Forced reference point	Forced reference point order with an incremental encoder. The current position is forced to the value defined in the **Param** field. This type of reference point does not shift the moving part
Reference cancellation	In order to be able to shift the moving part without soft stop fault: non-referenced axis crossing order with an absolute encoder
Referencing	With an absolute encoder in direct offset, referenced axis crossing order
Calculation offset	With an offset-assisted absolute encoder, trigger the encoder offset calculation to make the current position coincide with the value in length units entered in the **Param** field. The axis is referenced at the end of this calculation.
Auxiliary output	Sets auxiliary output to 1 or 0

(1) These commands remain active for as long as the button is pressed. They are used to disengage the moving part outside of soft stops (after acknowledging a fault).

Description of Faults Field

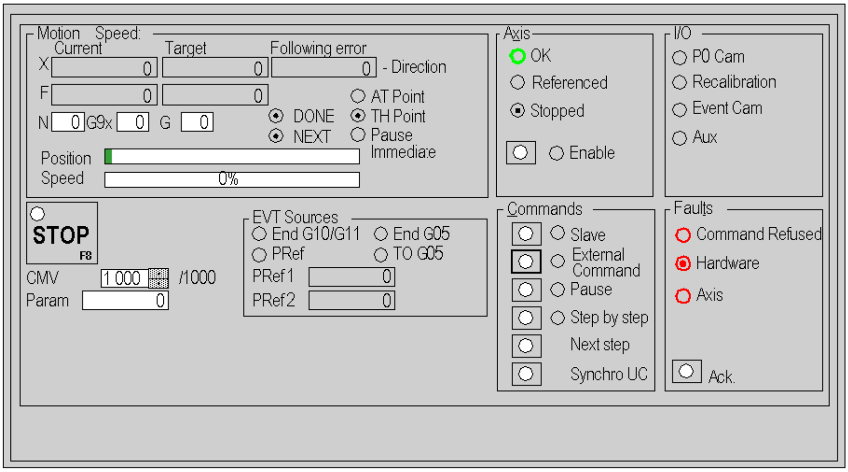
This table shows the display and command zones for the Faults field:

LED / Button	State	Indication
Command Refused	Lit	Last command refused
Hardware	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs, etc.)
Axis	Lit	Application fault (e.g. following error, soft stops etc.)
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

Automatic Mode (Auto)

At a Glance

Automatic mode is used for executing SMOVE functions.



Description of Movement / Speed Field

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

Display zone	Description
X Current	Displays the moving part position using the measurement unit defined in the configuration.
X Target	Displays the moving part setpoint position: target position (defined in instruction) (1)
X Following error	Displays the deviation between the setpoint position calculated and the actual position of the moving part (following error).
F Current	Displays the moving part speed using the measurement unit defined in the configuration.
F Target	Displays the setpoint speed of the moving part: target speed (speed defined by CMV coefficient in modulated instruction) (2)
N / G9x / G	These fields display the instruction which is being executed. N = step number, G9x = movement type, G = instruction code
Position	The bar chart shows the progress of the moving part within the limits defined in the configuration. The bar chart is colored green, and becomes red if the limits are exceeded
Speed	The bar chart expresses the speed of the moving part in relation to maximum speed as a percentage. The bar chart is colored green, and becomes red if the maximum speed is exceeded

NOTE: The number of display digits is limited to 10. A line of dots () is displayed for any higher value.

(1) Displays the number of latches (1 or 2) for instruction G07.

(2) Displays the Time Out for instruction G05.

This table shows indicators for the Movement/ Speed field:

Indicator	State	Indication
+ Direction - Direction	/	Indicates that the part is moving in a positive direction Indicates that the part is moving in a negative direction
DONE	Lit	Indicates that the movement(s) in progress are now finished
NEXT	Lit	Indicates that the module is ready to receive a movement command
AT Point	Lit	Indicates that the movement in progress has finished, and that the moving part is in the target window (for instructions with stop)
TH Point	Lit	Indicates that the theoretical setpoint has been reached
Feed hold	Lit	Indicates that the Feed hold function has been activated (the CMV coefficient is set at 0)

Description of Axis Field

This table shows the display and command zones for the Axis field:

LED / Button	State	Indication
OK	Lit	Axis in operational state (no blocking fault)
Referenced	Lit	Referenced axis
Stopped	Lit	Moving part stationary
Enable	/	This button is used to control the variable speed controller enable relay

Description of I/O Field

This table shows the display zones for the I/O field:

LED	Indication
PO Cam	Signal state (0 or 1) on **Reference point** input
Recalibration	Signal state (0 or 1) on **Recalibration** input
Event Cam	Signal state (0 or 1) on **Event** input
Aux	Signal state (0 or 1) on auxiliary output

1 = LED lit, 0 = LED off

Description of Commands

This table describes the command zone:

Command	Description
STOP	Stops the moving part according to deceleration defined in the configuration
Param	Used to enter external values (position tracking function)
CMV	Used to enter a value from 0 to 2000 which determines the speed multiplier coefficient (0.000 to 2000 in intervals of 1/1000)

Description of Sources EVT Field

This table shows the display zones for the Sources EVT field:

Indicator	State	Indication
PRef	Lit	Indicates PREF position latching (1)
PRef1	/	This field displays the memorized PREF1 position (1)
PRef2	/	This field displays the memorized PREF2 position (1)
End G10/G11	Lit	Indicates event arrival while G10 or G11 instruction is being executed
End G05	Lit	Indicates that execution of instruction G05 is complete
TO G05	Lit	Indicates that the Time Out, defined in instruction G05, has elapsed

(1) Providing that an event processing task has been associated with command G07.

There is no indicator associated with the **Modulo crossing** event.

Description of Commands Field

This table describes the buttons in the Commands field:

Command	Description
Slave	Changes the axis to **slave** mode (slave of another axis). 0 axis cannot be set to slave axis mode
External Command	Changes the axis to a slave of periodic setpoint
Pause	Commands the moving part to stop at the end of a movement with a stop in progress
Step by step	Changes the axis to **Step by step** mode
Next step	In **Step by step** mode, activates the waiting movement
Synchro UC	Triggers a PLC event

Description of Faults Field

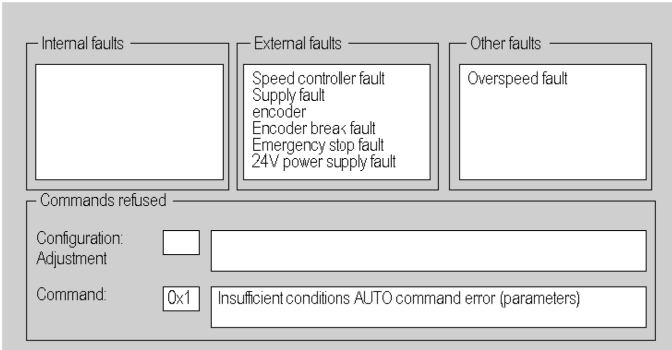
This table shows the display and command zones for the Faults field:

LED / Button	State	Indication
Command Refused	Lit	Last command refused
Hardware	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs, etc.)
Axis	Lit	Application fault (e.g. following error, software limits, etc.)
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

Channel Diagnostics

At a Glance

The various debugging, adjustment and configuring screens offer a **DIAG** tab in online mode, which gives access to details of faults detected by the module.



Description of the Different Fields

The **Channel diagnostics** screen offers the following fields:

Field	Description
Internal faults	Internal faults within the module which generally require it to be replaced
External faults	Faults originating from the operating part
Other faults	Application faults
Command refused	Indicates the cause and the message number of a command refused *(see page 338)*

Archiving and Documentation

Archiving

When the program has been debugged in online mode, the following saves must be carried out:

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

Documentation

Application documentation for axis control is included in the complete Control Expert application documentation. The documentation contains the following in one file:

- the program,
- the **Configuration** parameters and saved **Adjust** parameters.

Chapter 12
Operation

Human-Machine Interface Design

Button Box

To design a simple or complex button box, commands and elementary information are available in the form of bits, command and status words and bits.

Chapter 13
Diagnostics and Maintenance

Subject of this Section

This section describes the procedure to follow when encountering various possible situations, (symptoms, diagnostics and action to be taken)

What Is in This Chapter?

This chapter contains the following topics:

Fault and Command Executability Monitoring

Fault Monitoring

There are several ways of detecting a possible fault:

- LEDs on the module front panel.
- Diagnostic screens which can be accessed by the **DIAG** key in online mode from all the application-specific axis control module screens.
- Debug screens.
- Fault bits and status words

Movement Commands

The following conditions must be fulfilled in order to execute movement commands (in automatic or manual mode):

- The axis is configured and is without a blocking fault.
- The speed controller validation command ENABLE (%Qr.m.c.9) is active and the command STOP (%Qr.m.c.15) is inactive.
- Automatic or manual mode is selected.
- For commands in absolute position, the position is:
  - between SL_MIN (%MDr.m.c.33) and SL_MAX (%MDr.m.c.31) ranges for a limited axis,
  - between values of 0 and modulo –1 for an unlimited axis.
- For commands on a 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 the reference point commands.
- The SMOVE function parameter F is equal to VMAX.

Modifying the CMV Parameter

If modifying a CMV speed modulation parameter implies a resulting speed greater than VMAX, this speed is limited to VMAX.

Sequence Control

If the **Sequence control** option has not been selected in the configuration, a non-stop movement followed by no sequencing command will continue as far as a soft stop.

Diagnostics Help

At a Glance

Situations may arise which you have to resolve. The following procedure is designed to help diagnose these situations, and indicates the path to follow.

What to Do in Situations

New parameters not taken into account

Problem	The TSX CAY module does not seem to have registered the new parameters written by WRITE_PARAM.
Diagnostics	Program a READ_PARAM instruction into the application to find out the values actually being used by the module. A WRITE_PARAM is triggered and ignored while exchanging other adjustment parameters
What to do	Test the ADJ_IN_PROGR (%MWr.m.c.0.2) bit before exchanging any adjustment parameters.

Event processing

Problem	The event processing associated with the axis control channel has not been executed.
Diagnostics	Check that the whole event feedback string has been confirmed: ● Event number declared in the configuration is identical to the one in event processing. ● Origin of event unmasked (M code from the SMOVE command). ● Events authorized at system level (AUX0MINTIME(%S38) = 1). ● Unmasked events at system level (UNMASKEVT()).
What to do	Refer to event usage.

Lost adjustments

Problem	Adjustments have been lost.
Diagnostics	A cold restart can cause current adjustments that have been carried out via a screen or by a WRITE_PARAM instruction to be lost.
What to do	Save current adjustments by using the **Utilities** → **Save parameters** command, or by using the SAVE_PARAM instruction.

Inconsistent status words

Problem	EXCH_RPT (%MWr.m.c.1) and CH_FLT (%MWr.m.c.2) status words are inconsistent with the status of the axis control channel.
Diagnostics	These words are only updated through an explicit READ_STS request.
What to do	Program a READ_STS instruction into the application.

Encoder supply fault

Problem	The encoder supply fault persists while the encoder is properly supplied and the current value is changing.
Diagnostics	The encoder supply return signal has not been wired correctly.
What to do	Check the encoder connections.

Ineffective commands

Problem	The debugging screen commands are ineffective.
Diagnostics	The application or the task is in STOP mode.
What to do	Change the application or the task to RUN mode.

Non-modifiable commands

Problem	Certain debugging screen commands cannot be modified.
Diagnostics	These bits are written by the application.
What to do	Use bit forcing (for %Qr.m.c.d type objects) or re-do the application to avoid systematic writing of these bits (modification on transition and not on status).

Character entry not possible

Problem	It is not possible to enter more than 3 characters into the digital fields in the adjustment and configuration screens.
Diagnostics	The thousands separator has not been selected in the Windows configuration panel.
What to do	In the Windows configuration panel, select the **International** icon in the **Number format** field. Activate the **Modify** command and choose a thousands separator.

Chapter 14
Additional Functions

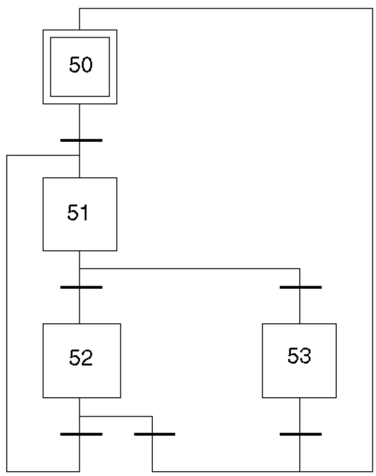
Dimension Learning

At a Glance

The example from the following Control Expert program shows the learning and usage of 16 dimensions.

Dimension learning

This graph is used to program learning for 16 dimensions.

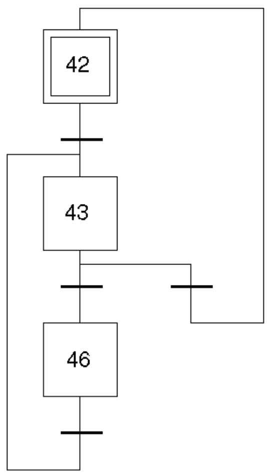



```
STEP 50 ACTION ON ACTIVATION
<memorizes %MW99 with a view to using it as a limit
! %MW98 := %MW99;
< Initializes the index during the learning phase
!  %MW99 := -1;
TRANSITION: X50 -> X51
! RE %I2.0
STEP 51 ACTION ON ACTIVATION
```

```
< brings the index up to date
! %MW99 := %MW99+1;
< positions learning
! %MD200[%MW99]: = X_POS;
TRANSITION: X51 -> X52
! %MW99 <= 16
TRANSITION: X51 -> X53
! %MW99 > 16
TRANSITION: X53 -> X50
! RE %I2.1
TRANSITION: X52 -> X51
! RE %I2.0
TRANSITION: X52 -> X50
! RE %I2.1
```

Using Dimensions

This graph is used to program the utilization of 16 dimensions.




```
STEP 42 ACTION ON ACTIVATION
<initializes %MW97 as the execution index
! %MW97 := -1;
TRANSITION: X42 -> X43
```

```
! RE %I2.2
STEP 43 ACTION ON ACTIVATION
< increments the execution index
! %MW97 := %MW97+1;
< executes the following segment
! SMOVE (AXIS_CH0,%MW97,%KW8,%KW1,%MD200[%MW97],150000,0);
```

%KW8 : 90 movement in the absolute value

%KW1: 09 go to point with break

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

Chapter 15
Language Objects of the Independent Axis Specific Application

Aim of this Chapter

This chapter describes the language objects associated with the axis specific application as well as the different ways of using them.

What Is in This Chapter?

This chapter contains the following topics:

Presentation of the language objects of the axis specific function

General

The IODDT's are preset by the constructor and contains input/output language objects belonging to the channel of an application-specific module.

There are three IODDT types for the axis specific application:

- `T_AXIS_AUTO` that applies to the 5 TSX CAY21/41/22/42/33 modules
- `T_AXIS_STD` that applies to the 5 TSX CAY21/41/22/42/33 modules
- `T_INTERPO_STD` is specific to the TSX CAY33 modules

NOTE: IODDT variables can be created in two different ways:

- Using the **I/O objects** *(see EcoStruxure™ Control Expert, Operating Modes)* tab,
- Data Editor.

Language object types

In each of the IODDT's there is a set of language objects allowing command of these and control of their operation.

There are two types of language objects:

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

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

Explicit exchanges allow module parameter setting and diagnostics.

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

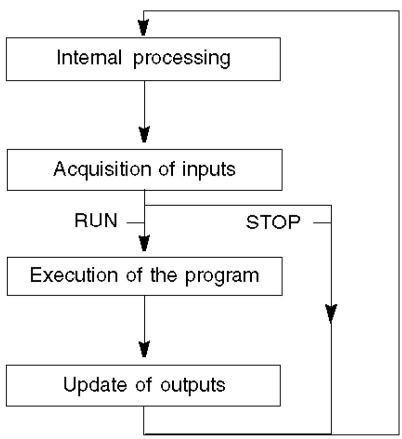
The module inputs (`%I` and `%IW`) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

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:

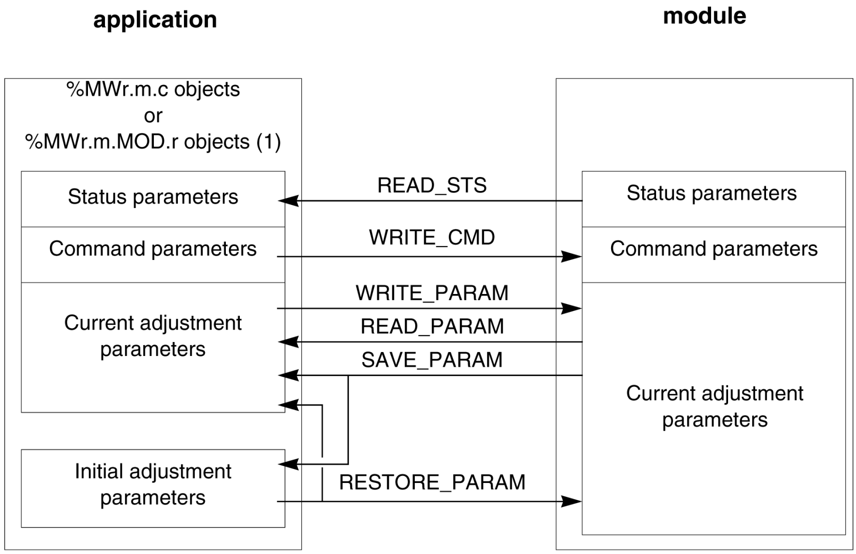
- provide information about the module (for example, type of error detected in a channel)
- have command control of the module (for example, switch command)
- define the module's operating modes (save and restore adjustment parameters in the process of application)

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.



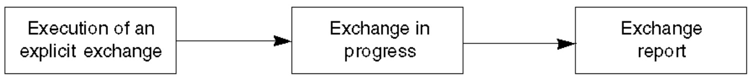
(1) Only with READ_STS and WRITE_CMD instructions.

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 321)*
- the exchange report *(see page 321)*

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:

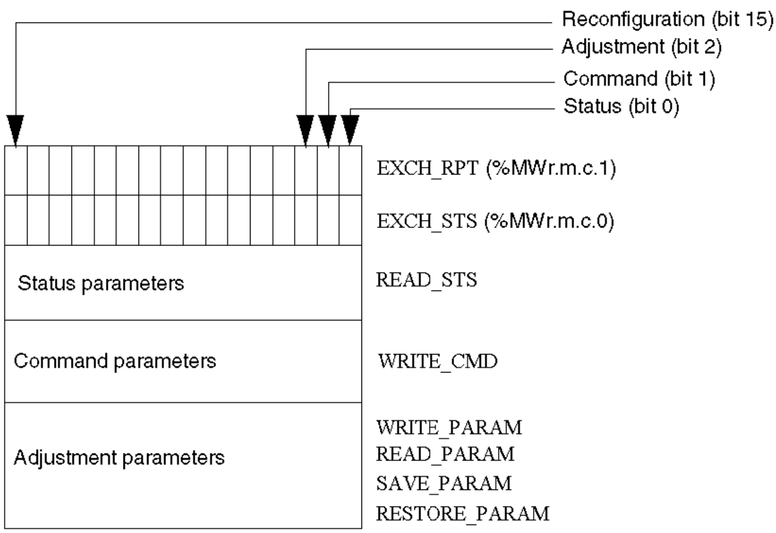
- `EXCH_STS` (`%MWr.m.c.0`): exchange in progress
- `EXCH_RPT` (`%MWr.m.c.1`): report

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:

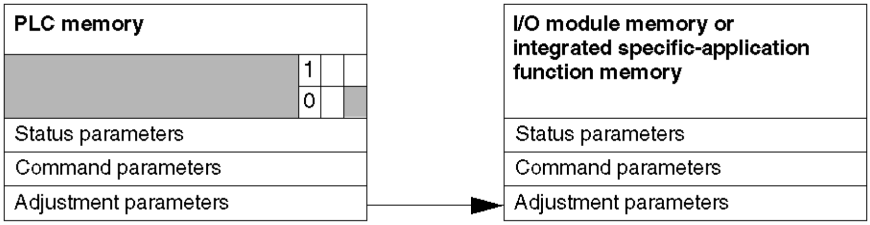
- Rank 0 bits are associated with the status parameters:
  - The STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress.
  - The STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits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

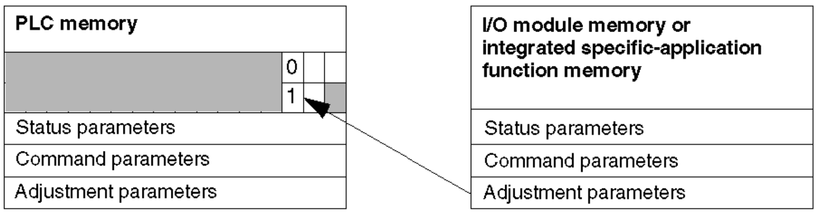
Example

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.

Implicit Exchange Internal Command Objects of the T_AXIS_AUTO Type IODDT

List of Implicit Exchange Objects

The following table presents the implicit exchange internal command objects of the T_AXIS_AUTO type IODDT

Standard symbol	Type	Access	Active on	Description	Address
ACK_FLT	EBOOL	R/W	Edge	Fault acknowledgement	%Qr.m.c.8
ENABLE	EBOOL	R/W	State	Enabling axis speed drive safety relay	%Qr.m.c.9
EXT_EVT	EBOOL	R/W	Edge	Trigger event order from processor	%Qr.m.c.10
PAUSE	EBOOL	R/W	State	Suspend movements command at the end of a movement in progress	%Qr.m.c.16
MOD_STEP	EBOOL	R/W	State	Change to step by step mode command	%Qr.m.c.19
NEXT_STEP	EBOOL	R/W	Edge	Activate next step command	%Qr.m.c.22
MOD_SELECT	INT	R/W		mode selector	%QWr.m.c.0
CMV	INT	R/W		speed modulation Value = speed modulation setpoint value This setpoint is in the range 0 to 2, in intervals of 1/1000.	%QWr.m.c.1

Mode Selector

MOD_SELECT: mode selector

Value	Mode	Description
0	DRV_OFF	Measurement mode: inhibition of CNA output
1	DIRDRIVE	Loop control disabled mode: direct voltage command
2	MANU	Manual mode
3	AUTO	Automatic mode

Internal Status Objects (Implicit Exchanges) of the IODDT of Type T_AXIS_AUTO

List of Objects with Implicit Exchanges

The tables below presents the internal objects (implicit exchanges) of the IODDT of the `T_AXIS_AUTO` type.

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)	%Ir.m.c.0
DONE	EBOOL	R	All instructions are executed: no instructions in the stack	%Ir.m.c.1
AX_FLT	EBOOL	R	Fault present on axis	%Ir.m.c.2
AX_OK	EBOOL	R	No fault causing moving part to stop	%Ir.m.c.3
EX_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.8
AT_PNT	EBOOL	R	Moving part position on target (in the point window, on instruction with stop)	%Ir.m.c.9
TH_PNT	EBOOL	R	Theoretical setpoint reached	%Ir.m.c.10
REF_OK	EBOOL	R	Reference point taken (axis referenced)	%Ir.m.c.14
DIRECT	EBOOL	R	Indicates direction of movement	%Ir.m.c.17
IN_AUTO	EBOOL	R	Automatic mode active	%Ir.m.c.23
IN_INTERPO	EBOOL	R	Interpolated movement in progress	%Ir.m.c.32
ON_PAUSE	EBOOL	R	Movements sequence suspended	%Ir.m.c.33
IM_PAUSE	EBOOL	R	Movement suspended (PAUSE immédiate)	%Ir.m.c.34
ST_IN_STEP	EBOOL	R	Step by step mode in progress	%Ir.m.c.39
DRV_ENA	EBOOL	R	Speed drive enable output image	%Ir.m.c.40
OVR_EVT	EBOOL	R	Event overrun	%Ir.m.c.46
EVENT_G07	EBOOL	R	Event source: latch position	%Ir.m.c.47
EVENT_G05	EBOOL	R	Event source: G05 end on event	%Ir.m.c.48
TO_G05	EBOOL	R	Event source: G05 time delay expired	%Ir.m.c.49
EVT_G1	EBOOL	R	Event source: G10 or G11 end on event	%Ir.m.c.50
EVT_MOD	EBOOL	R	Modulo crossing	%Ir.m.c.51
X_POS	DINT	R	measured position	%IDr.m.c.0

Standard symbol	Type	Access	Description	Address
SPEED	DINT	R	measured speed	%IDr.m.c.2
FOL_ERR	DINT	R	current position deviation	%IDr.m.c.4
SYNC_N_RUN	INT	R	step number in progress	%IWr.m.c.7

NOTE: If 0, 1 and 2 channels are interpolated`IN_INTERPO` bits are in position 1 (%Ir.m.0.32, %Ir.m.c.1.32 and %Ir.m.2.32).

Internal Status Objects (Explicit Exchanges) of IODDT Type T_AXIS_AUTO

At a Glance

This part presents the internal status object (explicit exchanges) of the IODDT of type `T_AXIS_AUTO` which applies to modules TSX CAY21/41/22/42/33. It groups togethr word type objects whose bits have a special significance. These objects are presented in details below.

Notes

- Generally speaking the meaning of the bits is given for the 1 state of this bit. In each specific case the state of the bit is explained.
- Not all the bits are used.

Managing Exchanges: EXCH_STS

The table below presents the significances of the channel exchange control bits `EXCH_STS` (%MWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
CMD_IN_PROGR	BOOL	R	Command parameters exchange in progress	%MWr.m.c.0.1

Exchanges Report : EXCH_RPT

The table below presents the meanings of the report bits `EXCH_RPT` (%MWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
CMD_ERR	BOOL	R	Command parameters exchange report	%MWr.m.c.1.1

Other Status Data

The table below presents the meanings of other state data.

Standard symbol	Type	Access	Meaning	Address
AX_STS	INT	R	Axis operating status	%MWr.m.c.3
N_RUN	INT	R	step number in progress	%MWr.m.c.4
G9_COD	INT	R	Movement type in progress	%MWr.m.c.5
G_COD	INT	R	Instruction code in progress	%MWr.m.c.6
CMD_FLT	INT	R	refusal report	%MWr.m.c.7
T_XPOS	DINT	R	reach position target	%MDr.m.c.9
T_SPEED	DINT	R	speed to be reached	%MDr.m.c.13

Internal Command Objects (Implicit Exchanges) of the IODDT of Type T_AXIS_STD

List of Objects with Implicit Exchanges

The tables below presents the internal objects (implicit exchanges) of the IODDT of the `T_AXIS_STD` type.

Standard symbol	Type	Access	Active on	Description	Address
DIRDRV	EBOOL	R/W	Edge	Movement command in loop control disabled mode	%Qr.m.c.0
JOG_P	EBOOL	R/W	Edge	Manual movement unlimited in + direction	%Qr.m.c.1
JOG_M	EBOOL	R/W	Edge	Manual movement unlimited in - direction	%Qr.m.c.2
NC_P	EBOOL	R/W	Edge	Incremental movement order (PARAM) in + direction	%Qr.m.c.3
INC_M	EBOOL	R/W	Edge	Incremental movement order (PARAM) in - direction	%Qr.m.c.4
SET_RP	EBOOL	R/W	Edge	Manual reference point (RP_POS = original value) or change to non-referenced state	%Qr.m.c.5
RP_HERE	EBOOL	R/W	Edge	Reference point forced to a value defined in PARAM or change to a referenced state / offset calculation	%Qr.m.c.6
ACK_FLT	EBOOL	R/W	Edge	Fault acknowledgement	%Qr.m.c.8
ENABLE	EBOOL	R/W	State	Enabling axis speed drive safety relay	%Qr.m.c.9
EXT_EVT	EBOOL	R/W	Edge	Trigger event order from processor	%Qr.m.c.10
AUX_OUT	EBOOL	R/W	State	Auxiliary output command	%Qr.m.c.11
STOP	EBOOL	R/W	State	Immediate stop command (halting of moving part)	%Qr.m.c.15
PAUSE	EBOOL	R/W	State	Suspend movements command at the end of a movement in progress	%Qr.m.c.16
SLAVE	EBOOL	R/W	State	Setpoint in progress = 0 axis position	%Qr.m.c.17
MOD_STEP	EBOOL	R/W	State	Change to step by step mode command	%Qr.m.c.19
NEXT_STEP	EBOOL	R/W	Edge	Activate next step command	%Qr.m.c.22
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 in the range 0 to 2, in intervals of 1/1000.	%QWr.m.c.1
PARAM	INT	R/W		value of movement increment	%QWr.m.c.2

Mode Selector

MOD_SELECT: mode selector

Value	Mode	Description
0	DRV_OFF	Measurement mode: inhibition of CNA output
1	DIRDRIVE	Loop control disabled mode: direct voltage command
2	MANU	Manual mode
3	AUTO	Automatic mode

Internal Status Objects (Implicit Exchanges) of the IODDT of Type T_AXIS_STD

List of Objects with Implicit Exchanges

The tables below presents the internal objects (implicit exchanges) of the IODDT of the T_AXIS_STD£££ type.

Standard symbol	Type	Access	Description	Number
CH_ERROR	EBOOL	R	Channel fault	%Ir.m.c.ERR
NEXT	EBOOL	R	Ready to receive a new movement command (in AUTO)	%Ir.m.c.0
DONE	EBOOL	R	All instructions are executed: no instructions in the stack	%Ir.m.c.1
AX_FLT	EBOOL	R	Fault present on axis	%Ir.m.c.2
AX_OK	EBOOL	R	No fault causing moving part to stop	%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.8
AT_PNT	EBOOL	R	Moving part position on target (in the point window, on instruction with stop)	%Ir.m.c.9
TH_PNT	EBOOL	R	Theoretical setpoint reached	%Ir.m.c.10
CONF_OK	EBOOL	R	Configured axis	%Ir.m.c.12
REF_OK	EBOOL	R	Reference point taken (axis referenced)	%Ir.m.c.14
AX_EVT	EBOOL	R	Recopy physical event inputs	%Ir.m.c.15
HOME	EBOOL	R	Recopy CAME physical input of module reference point	%Ir.m.c.16
DIRECT	EBOOL	R	Indicates direction of movement	%Ir.m.c.17
IN_REC	EBOOL	R	Recopies recalibration input on-the-fly	%Ir.m.c.18
IN_DROFF	EBOOL	R	Measurement mode active	%Ir.m.c.20
IN_DIRDR	EBOOL	R	Loop control disabled mode active	%Ir.m.c.21
IN_MANU	EBOOL	R	Manual mode active	%Ir.m.c.22
IN_AUTO	EBOOL	R	Automatic mode active	%Ir.m.c.23
ST_JOG_P	EBOOL	R	Unlimited movement in + direction in progress	%Ir.m.c.26
ST_JOG_M	EBOOL	R	Unlimited movement in - direction in progress	%Ir.m.c.27
ST_INC_P	EBOOL	R	Incremental movement in + direction in progress	%Ir.m.c.28
ST_INC_M	EBOOL	R	Incremental movement in - direction in progress	%Ir.m.c.29
ST_SETRP	EBOOL	R	Manual reference point in progress	%Ir.m.c.30
ST_DIRDR	EBOOL	R	Loop control disabled movement in progress	%Ir.m.c.31
IN_INTERPO	EBOOL	R	Interpolated movement in progress	%Ir.m.c.32
ON_PAUSE	EBOOL	R	Movements sequence suspended	%Ir.m.c.33

Standard symbol	Type	Access	Description	Number
IM_PAUSE	EBOOL	R	Movement suspended (PAUSE immédiate)	%Ir.m.c.34
IN_SLAVE	EBOOL	R	Setpoint in progress = 0 axis position	%Ir.m.c.36
IN_EXT_CMD	EBOOL	R	Setpoint in progress = processor setpoint	%Ir.m.c.37
ST_IN_STEP	EBOOL	R	Step by step mode in progress	%Ir.m.c.39
DRV_ENA	EBOOL	R	Speed drive enable output image	%Ir.m.c.40
IN_AUX0	EBOOL	R	AUX0 output image	%Ir.m.c.41
OVR_EVT	EBOOL	R	Event overrun	%Ir.m.c.46
EVENT_G07	EBOOL	R	Event source: latch position	%Ir.m.c.47
EVENT_G05	EBOOL	R	Event source: G05 end on event	%Ir.m.c.48
TO_G05	EBOOL	R	Event source: G05 time delay expired	%Ir.m.c.49
EVT_G1	EBOOL	R	Event source: G10 or G11 end on event	%Ir.m.c.50
EVT_MOD	EBOOL	R	Modulo crossing	%Ir.m.c.51
X_POS	DINT	R	measured position	%IDr.m.c.0
SPEED	DINT	R	measured speed	%IDr.m.c.2
FOL_ERR	DINT	R	current position deviation	%IDr.m.c.4
ANA_OUT	INT	R	current analog output	%IWr.m.c.6
SYNC_N_RUN	INT	R	step number in progress	%IWr.m.c.7
PREF1	DINT	R	Capture of position of axis PREF1	%IDr.m.c.11
PREF2	DINT	R	Capture of position of axis PREF2	%IDr.m.c.13

NOTE: If 0, 1 and 2 channels are interpolated`IN_INTERPO` bits are in position 1 (%Ir.m.0.32, %Ir.m.c.1.32 and %Ir.m.2.32).

Internal Status Objects (Explicit Exchanges) of IODDT Type T_AXIS_STD

At a Glance

This part presents the internal status object (explicit exchanges) of the IODDT of type `T_AXIS_STD` which applies to modules TSX CAY21/41/22/42/33. It groups together word type objects whose bits have a special significance. These objects are presented in details below.

Notes

- Generally speaking the meaning of the bits is given for the 1 state of this bit. In each specific case the state of the bit is explained.
- Not all the bits are used.

Managing Exchanges: EXCH_STS

The table below presents the significances of the channel exchange control bits `EXCH_STS` (%MWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Status parameters (STATUS) exchange 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	Exchanging adjustment parameters	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Module reconfiguration in progress	%MWr.m.c.0.15

Exchanges Report : EXCH_RPT

The table below presents the meanings of the 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 Operating Status: CH_FLT

The table below presents the meanings of the report bits `CH_FLT` (%MWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
EXT_FLT	BOOL	R	External fault (same as HD_ERR bit)	%MWr.m.c.2.0
MOD_FLT	BOOL	R	Internal error Module absent, inoperative, or in self-test mode	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Hardware or software configuration fault	%MWr.m.c.2.5
COM_FLT	BOOL	R	Communication fault with processor	%MWr.m.c.2.6
APP_FLT	BOOL	R	Application fault (errored configuration) or command fault	%MWr.m.c.2.7
CH_LED_LOW	BOOL	R	Channel LED status	%MWr.m.c.2.8
CH_LED_HIGH	BOOL	R	Channel LED status	%MWr.m.c.2.9

Axis Operating Status : AX_STS

The table below presents the meanings of the report bits `AX_STS` (%MWr.m.c.3).

Standard symbol	Type	Access	Meaning	Address
Hardware faults : HD_ERR (%Ixy.i.4) (groups the faults below)				
ANA_FLT	BOOL	R	Analog output short-circuit fault	%MWr.m.c.3.0
AUX_FLT	BOOL	R	Auxiliary output short-circuit fault	%MWr.m.c.3.1
DRV_FLT	BOOL	R	Speed drive fault	%MWr.m.c.3.2
ENC_SUP	BOOL	R	Encoder supply fault	%MWr.m.c.3.3
ENC_BRK	BOOL	R	Encoder break fault	%MWr.m.c.3.4
EMG_STP	BOOL	R	Emergency stop fault	%MWr.m.c.3.5
AUX_SUP	BOOL	R	24 V supply fault	%MWr.m.c.3.0
ENC_FLT	BOOL	R	Absolute encoder parity series or E bit fault	%MWr.m.c.3.7
Application faults : AX_ERR (%Ixy.i.5)(groups the faults below)				
SLMAX	BOOL	R	Maximum soft stop overshoot	%MWr.m.c.3.8
SLMIN	BOOL	R	Minimum soft stop overshoot	%MWr.m.c.3.9
SPD_FLT	BOOL	R	Overspeed fault	%MWr.m.c.3.4
FE1_FLT	BOOL	R	MAX_F1 position deviation fault	%MWr.m.c.3.11
REC_FLT	BOOL	R	Recalibration fault	%MWr.m.c.3.12
TW_FLT	BOOL	R	Debugging window fault	%MWr.m.c.3.13
STP_FLT	BOOL	R	Stopping fault	%MWr.m.c.3.14
FE2_FLT	BOOL	R	MAX_F2 deviation fault	%MWr.m.c.3.15

Other Status Data

The table below presents the meanings of other state data.

Standard symbol	Type	Access	Meaning	Address
N_RUN	INT	R	step number in progress	%MWr.m.c.4
G9_COD	INT	R	Movement type in progress	%MWr.m.c.5
G_COD	INT	R	Instruction code in progress	%MWr.m.c.6
CMD_FLT	INT	R	refusal report	%MWr.m.c.7
T_XPOS	DINT	R	reach position target	%MDr.m.c.9
MAX_FER	DINT	R	maximum following error	%MDr.m.c.11
T_SPEED	DINT	R	speed to be reached	%MDr.m.c.13

Adjustment Parameters Objects (Explicit Exchanges) of the IODDT of Type T_AXIS_STD

Adjustment Parameters

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

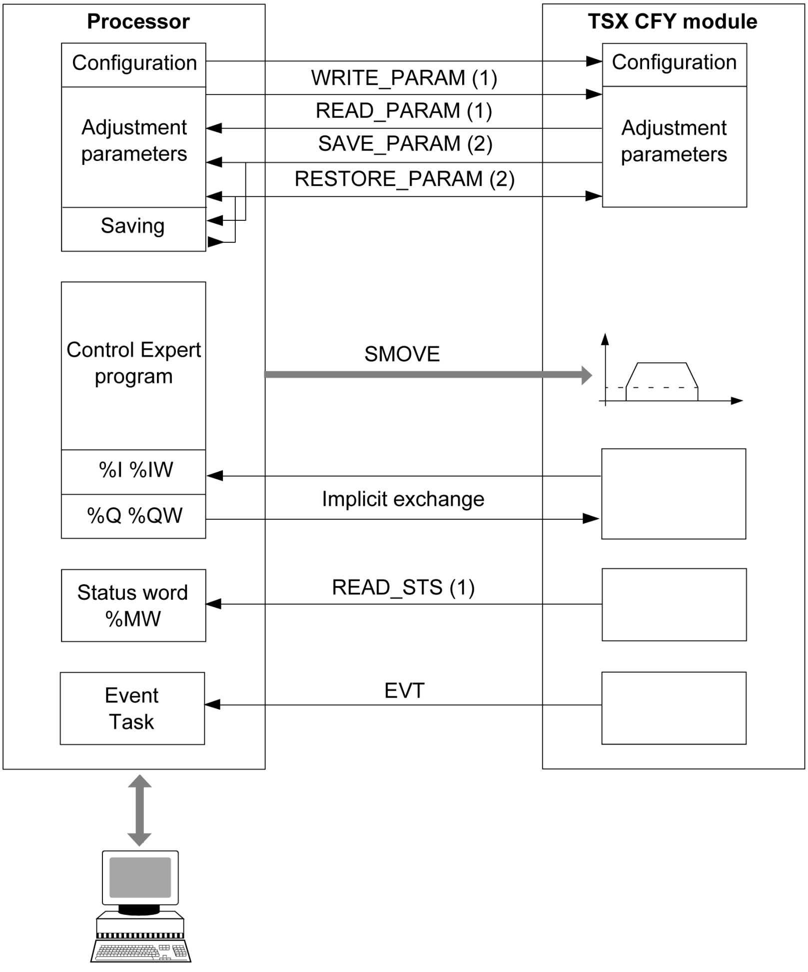
Standard symbol	Type	Access	Description	Address
SLOPE	INT	R/W	Acceleration rule 0 = rectangle, 1 to 3 = trapezoid, 4 = triangle	%MWr.m.c.15
KPOS1	INT	R/W	Position loop Gain 1: 0 to 120.00 (in 1/s)	%MWr.m.c.16
KPOS2	INT	R/W	Position loop Gain 2: 0 to 120.00 (in 1/s)	%MWr.m.c.17
SP_THR	INT	R/W	Switching threshold for gain: 20 to 500 VMAX/1000	%MWr.m.c.18
IPOS	INT	R/W	Ti integral action: integral time = 0 to 5000 ms 0 = no integral action (TSX CAY 22/42 and TSX CAY 33)	%MWr.m.c.19
			Reserved	%MWr.m.c.20
KV	INT	R/W	Loop feedforward speed gain: 0 to 100%	%MWr.m.c.21
OFFSET	INT	R/W	Loop CAN offset: -250 to 250 mV	%MWr.m.c.2.4
OVR_SPD	INT	R/W	Overspeed threshold: 0 to 20%	%MWr.m.c.23
S_STOP	INT	R/W	Stopping speed: 0 to VMAX/10 or 30000	%MWr.m.c.24
T_STOP	INT	R/W	Maximum delay for detecting a stop: 0 to 10000 ms	%MWr.m.c.25
TACC	INT	R/W	Acceleration / deceleration time: TACCMIN to 10000 ms	%MWr.m.c.26
VLIM	INT	R/W	Movement control activation threshold	%MWr.m.c.27
RATIO1	INT	R/W	Slave axis ratio (TSX CAY 22/42)	%MWr.m.c.29
RATIO2	INT	R/W	Slave axis ratio (TSX CAY 22/42)	%MWr.m.c.30
SL_MAX	DINT	R/W	High soft stop: SLMIN to LMAX for limited axis Modulo in points for an infinite axis	%MDr.m.c.31
SL_MIN	DINT		Lower soft stop: LMIN to SLMAX for a limited axis Modulo value in user units for an infinite axis	%MDr.m.c.33
MAN_SPD	DINT	R/W	Speed in manual mode: 10 to VMAX	%MDr.m.c.35
K_RES1	DINT	R/W	Resolution multiplier: 1 to 1000000	%MDr.m.c.37
K_RES2	DINT	R/W	Resolution divisor: 1 to 1000000	%MDr.m.c.39
RP_POS	DINT	R/W	Reference point value in manual mode: SLMIN to SLMAX	%MDr.m.c.41
RE_POS	DINT	R/W	Recalibration reference value: SLMIN to SLMAX	%MDr.m.c.43
MAX_F1	DINT	R/W	Position 1 deviation threshold: 0 to (SLMIN – SLMAX)/4	%MDr.m.c.45

Standard symbol	Type	Access	Description	Address
MAX_F2	DINT	R/W	Position 2 deviation threshold: 0 to (SLMIN – SLMAX)/4	%MDr.m.c.47
TW	DINT	R/W	Debugging window : 0 to (SLMIN – SLMAX)/20	%MDr.m.c.49
RE_WDW	DINT	R/W	Recalibration deviation threshold : 0 to (SLMIN – SLMAX)/20	%MDr.m.c.51
ABS_OFF	DINT	R/W	Absolute encoder offset	%MDr.m.c.53
SLAVE_OFF	DINT	R/W	Axis follower offset (TSX CAY 22/42)	%MDr.m.c.55
VALID_EVT_MOD	BOOL	R/W	Modulo crossing EVT	%MDr.m.c.63.0

Exchanges between Processor and Axis Control Module

Diagram of Exchanges

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



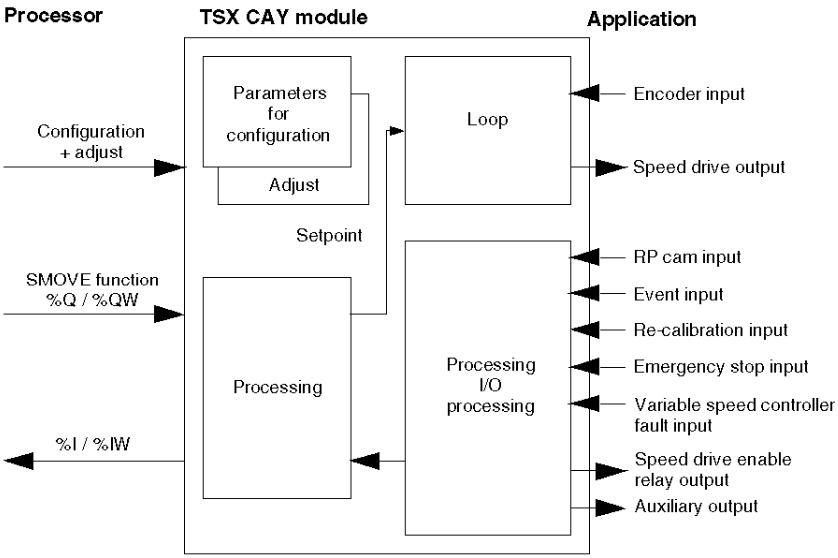
(1) Read or write from the adjustment or application screen, by using explicit exchange instructions.

(2) Save or restore from the **Save parameters** or **Restore parameters** commands from the Control Expert **Utilities** menu, or using SAVE_PARAM or RESTORE_PARAM instructions.

TSX CAY Module Channel

Channel Diagram

The following simplified diagram shows the channel functions for a TSX CAY module:



CMD_FLT Code Error List

At a Glance

Reading the CMD_FLT (%MWr.m.c.7) command refusal word is performed by explicit exchange. Non-encrypted messages are also available in the diagnostics dialog box, which can be accessed by the **DIAG** command.

Each CMD_FLT word byte is associated with an error type:

- The most significant byte indicates an error in the configuration and adjustment parameters (XX00).
- The least significant byte indicates that a movement command has been refused (00XX).

For example: CMD_FLT = 0004 (the least significant byte indicates a JOG+ command error)

Word %MWr.m.c.7

Configuration and adjustment parameters	Movement command
Most significant byte	Least significant byte

Configuration Parameters

These errors are indicated by the most significant %MWr.m.c.7 word byte. Numbers between brackets indicate hexadecimal code value.

Value	Meaning
3 (3)	Event priority configuration error
4 (4)	Machine configuration error (e.g. infinite, limited)
5 (5)	Encoder type configuration error
6 (6)	Reference point configuration error
7 (7)	Maximum setpoint configuration error
8 (8)	Maximum acceleration configuration error
9 (9)	Event configuration error
10 (A)	Follower ratio multiplier configuration error
11 (B)	Follower ratio divisor configuration error
12 (C)	Recalibration configuration error
18 (12)	Speed configuration error
19 (13)	Upper limit configuration error
20 (14)	Lower limit configuration error
21 (15)	Initial resolution distance configuration error
22 (16)	Initial resolution counts number configuration error

Value	Meaning
25 (19)	Length unit configuration error
26 (1A)	Speed unit configuration error
27 (1B)	Resolution and speed ratio configuration error
28 (1C)	Incompatible limits configuration error
29 (1D)	Follower ratio configuration error

Adjustment Parameter

These errors are indicated by the most significant %MWr.m.c.7 word byte. Numbers between brackets indicate hexadecimal code value.

Value	Meaning
82 (52)	Acceleration profile parameter error
83 (53)	Gain 1 parameter error
84 (54)	Gain 2 parameter error
85 (55)	Threshold 1 and 2 parameter error
88 (58)	Feedforward speed parameter error
89 (59)	Offset parameter error
90 (5A)	Overspeed parameter error
91 (5B)	Stop speed parameter error
92 (5C)	Stop delay parameter error
93 (5D)	Acceleration parameter error
94 (5E)	VLIM parameter error
98 (62)	Software Hi limit parameter error
99 (63)	Software Lo limit parameter error
100 (64)	Manual mode speed parameter error
101 (65)	Corrected resolution distance parameter error
102 (66)	Corrected resolution counts number parameter error
103 (67)	Reference point value parameter error
104 (68)	Recalibration position value parameter error
105 (69)	Following error 1 parameter error
106 (6A)	Following error 2 parameter error
107 (6B)	Debugging window parameter error
108 (6C)	Recalibration deviation parameter error
109 (6D)	Offset encoder parameter error
113 (71)	Resolution ratio parameter error

Value	Meaning
114 (72)	Incompatible soft stop parameter error
115 (73)	Maximal resolution and speed ratio parameter error
116 (74)	Resolution ratio parameter, VMAX and encoder multiplier error
117 (75)	Upper limit resolution ratio parameter error
118 (76)	Low limit resolution ratio parameter error
119 (77)	Resolution ratio parameter error on limit distances
120 (78)	Resolution correction parameter error (<> OFF mode)
121 (79)	Encoder offset modification parameter error (<> OFF mode)
122 (7A)	Recalibration position modification parameter error (<> OFF mode)

Movement Command Refused

These errors are indicated by the least significant %MWr.m.c.7 word byte. Numbers between brackets indicate hexadecimal code value.

Value	Message
1 (1)	Insufficient conditions MANU command error (e.g. Mode, Value etc.)
2 (2)	MANU movement in progress command error
3 (3)	Simultaneous MANU commands error
4 (4)	JOGP MANU command error
5 (5)	JOGM MANU command error
6 (6)	INCP MANU command error
7 (7)	INCM MANU command error
8 (8)	Manual PO MANU command error
9 (9)	Forced PO MANU command error
10 (A)	Encoder offset calculation error
16 (10)	Insufficient conditions AUTO command error (parameters)
17 (11)	Auto movement in progress AUTO command error (Slave or external Command activated at the same time for one movement)
18 (12)	Insufficient conditions SMOVE command error (Mode)
19 (13)	SMOVE G01 command error (1)
20 (14)	SMOVE G09 command error (1)
21 (15)	SMOVE G10 command error (1)
22 (16)	SMOVE G11 command error (1)
24 (18)	SMOVE G21 command error (1)
25 (19)	SMOVE G14 command error (1)

Value	Message
26 (1A)	SMOVE G05 command error (1)
27 (1B)	SMOVE G07 command error (1)
28 (1C)	SMOVE G62 command error (1)
29 (1D)	SMOVE execution command error
30 (1E)	Auto Slave command error
31 (1F)	External Auto Cde command error
32 (20)	Slave mode in progress AUTO command error
33 (21)	External Auto Cde in progress command error
34 (22)	SMOVE in progress external command slave axis error
xx (xx)	No zero latch on cam error during a reference point short cam with zero latch
35 (23)	Full stack error
36 (24)	Sequence control error
37 (25)	SMOVE G30 command error (1)
38 (26)	Change to next step error
48 (30)	Insufficient command - DIRDRIVE command error
64 (40)	SMOVE G01, G11 command error : already in position
80 (50)	SMOVE G30 command error: already in position
81 (51	SMOVE G30 command error: change in direction

(1) Indicates that one of the SMOVE function parameters is not compliant. Examples : Faulty movement type code, position outside soft stop, speed above VMAX, 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

Part IV
Interpolated Axes

Subject of this Part

This part introduces linear interpolation and describes how to set up an interpolated axes control with a TSX CAY 33 module.

What Is in This Part?

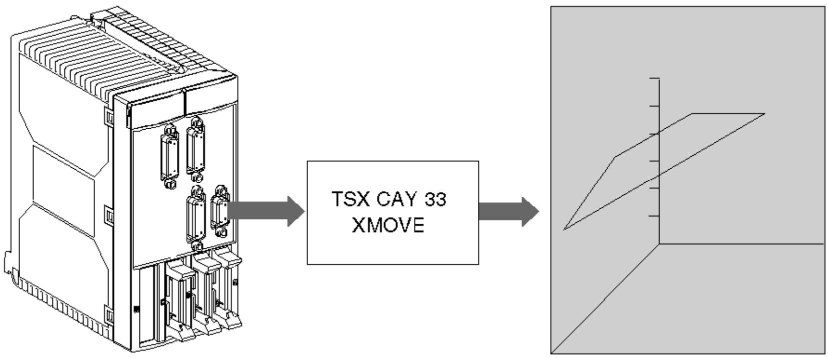
This part contains the following chapters:

Chapter 16
Introduction to Interpolation

General Information on the Interpolating Function

At a Glance

The interpolating function is available with a **TSX CAY 33** module, which is used to carry out **linear interpolation** between 2 or 3 axes.



This module consists of 3 physical channels associated with X, Y and Z axes, and a logic channel (channel 3) dedicated to interpolation.

Before implementing an application for interpolated axes, each of the axes must be set independently. Interpolation can then be carried out between 2 axes (0 and 1) on the (X, Y) plane or between 3 axes (0, 1 and 2) in the (X, Y, Z) space.

For interpolating 2 axes, the third axis (axis 2) can be used and an independent axis.

TSX CAY 33 module does not offer circular interpolation. However, to get from point A to point B following a circular trajectory, it is possible to approximate this type of trajectory by using a succession of straight segments.

Configuring Interpolation

The number of interpolated axes may be set in the interpolator configuration (channel 3), after channels 0 to 2 have been configured as independent axes.

If 2-D is specified, interpolation is implicitly declared on the XY plane (with X: 0 axis and Y: 1 axis). Channel 2 may be used as an independent axis.

If 3-D is specified, interpolation is implicitly declared in the XYZ space and/or on the planes which make up XY, YZ or XZ space (where X: 0 axis, Y : 1 axis and Z: 2 axis).

Configuring the 2 or 3 independent axes that you would like to interpolate is necessary in order to access interpolator configuration.

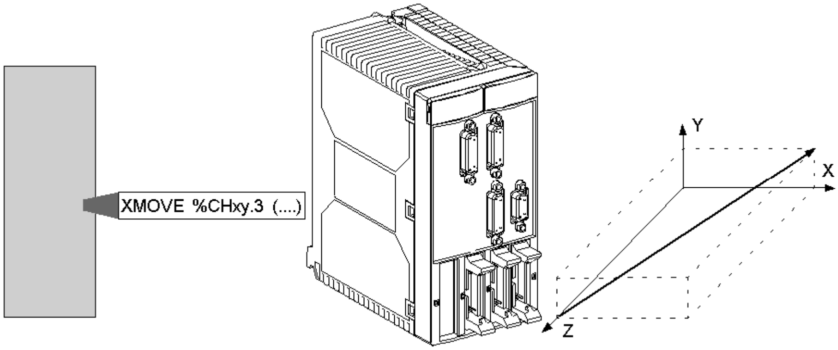
Interpolator Channel Programming

The XMOVE command is used to program interpolated movements. This command is an addition to SMOVE axes, not a substitute.

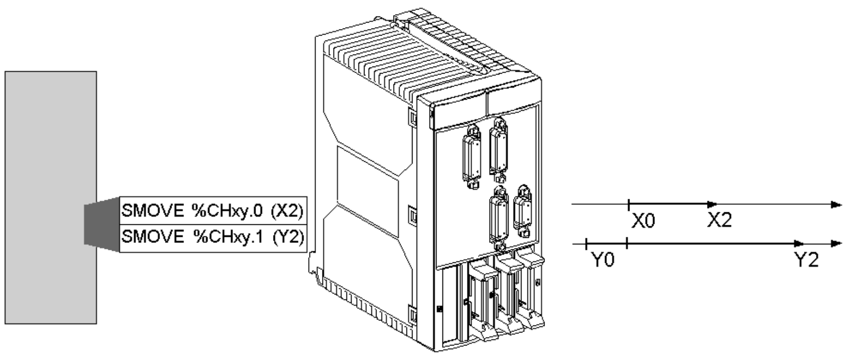
Axes are only interpolated during the execution of an XMOVE command. Outside of XMOVE commands, they can be commanded independently by an SMOVE command.

Movements Command

Interpolated axes movement command



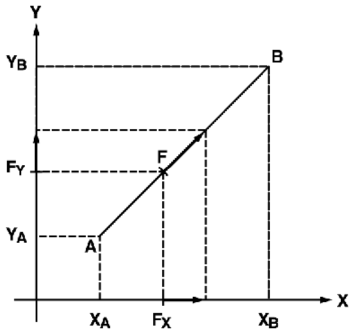
Independent axes movement command



Speed

The speed which you specify in the XMOVE command is the desired speed in the direction of movement. The movement speed of each axis is calculated by means of a projection.

Example for a 2 axis system



The moving part must move from point A (X_A, Y_A) to point B (X_B, Y_B) at speed F, which projects onto X and Y respectively in F_X and F_Y.

Using value F, which is provided in the XMOVE instruction, the interpolator calculates projections according to the formulae:

$$F_X = F \times (|X_B - X_A|) / (\Delta X)$$

$$F_Y = F \times (|Y_B - Y_A|) / (\Delta X)$$

with

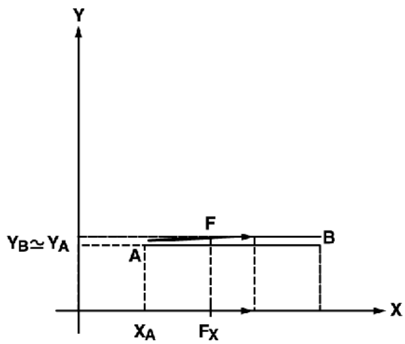
$$\Delta X = [(X_B - X_A)^2 + (Y_B - Y_A)^2]^{1/2}$$

Maximum Speed

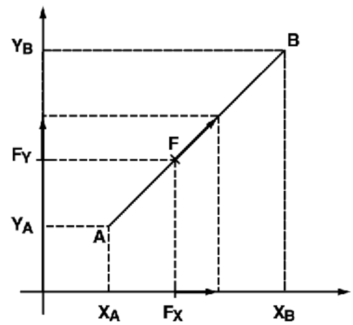
Speed F is limited to a maximum speed which depends on:

- the maximum speeds of each axis affected by the movement,
- the contribution of each axis within the movement.

Example:



FX # 0
F ≤ VMAX_axeX



$|X_B - X_A| = |Y_B - Y_A|$
$F_Y = F_X$
$F \leq (F_X^2 + F_Y^2)^{1/2}$

Acceleration

For each XMOVE movement, the duration of the acceleration phase depends on:

- the speed variations to be performed,
- the Tacc parameters for the axes affected by the movement,
- the contribution of X, Y and Z axes.

The resulting calculated acceleration is the fastest that the movement can happen, while respecting the constraints of the different axes (the most constraining axis determines the duration of the acceleration).

The acceleration rule is defined by the **SLOPE** parameter from channel 3. This imposes a common rule over all axes during an XMOVE independently of **SLOPE** parameter value for X, Y and Z axes.

Chapter 17
Programming Interpolation

Subject of this Section

This section describes the programming principle for an interpolated movement: description of the main instructions and operating modes.

What Is in This Chapter?

This chapter contains the following topics:

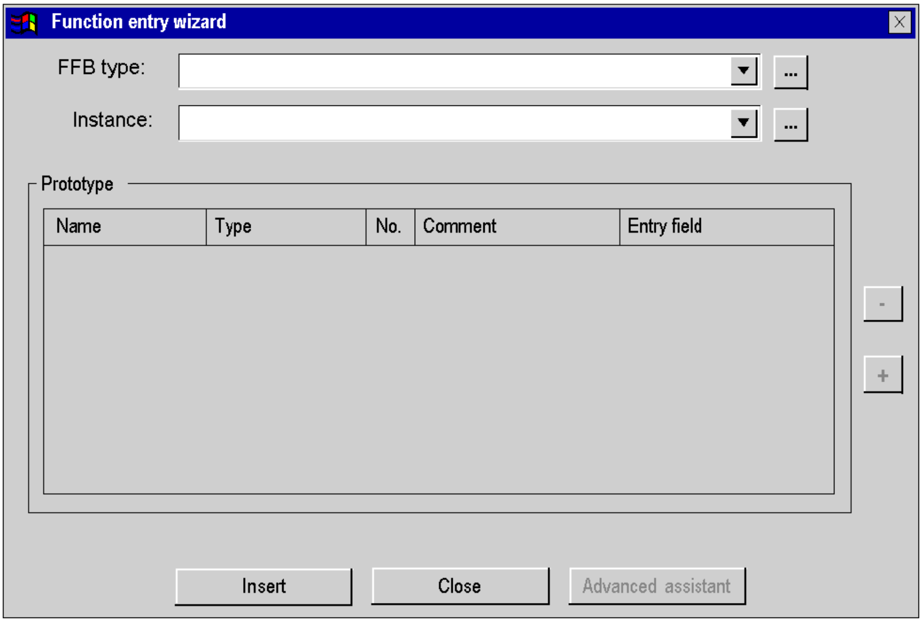
Programming Interpolated Movements

At a Glance

The XMOVE instruction must be used to program an interpolated movement. This instruction is automatically sent to channel 3 of the TSX CAY 33 module.

Assisted Entry Screen

You can enter the XMOVE function directly or using the entry help screen:



NOTE: When the XMOVE EF is displayed in red on the functions assisted entry screen, it cannot be used in the application.

- An earlier version of the EF is already used in the application.
- The name of the EF is used as a symbol associated with a variable.

Assisted Entry

In the ST editor proceed as follows:

Step	Action
1	Right click in the editor and select **FFB entry assistant...**. The **Function entry help** window will be displayed.
2	Type **XMOVE**.
3	Press the **Advanced assistant** button and fill in the different fields that are offered. Function variables can also be entered directly into the parameters entry zone.
4	Confirm with **OK** or **Enter**. The function will be displayed.

Entering XMOVE Function Parameters

At a Glance

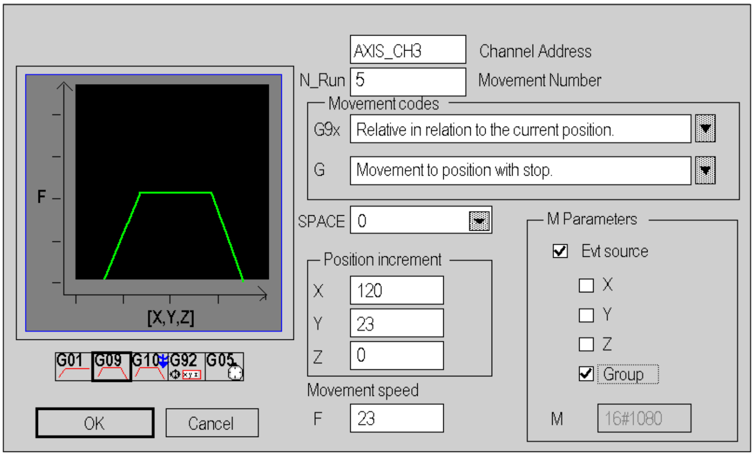
An interpolated movement command is programmed by an XMOVE function using the following syntax:

```
XMOVE (AXIS_CH3,N_Run,G9x,G,SPACE,X,Y,Z,F,M)
```

The **Details** screen will assist you in entering each parameter.

XMOVE Function Details Screen

The XMOVE function details screen is as follows:



Entry fields (for XMOVE function parameters) as are follows:

Parameter	Description
`AXIS_CH3`	IODDT type variable corresponding to channel 3 on which the function must operate. Example: `AXIS_CH3` of type `T_INTERPO_STD`
N_Run	Movement number
G9x	Movement type
G	Instruction code
SPACE	Plane or Space Number
X, Y and Z	Coordinates for Position to Reach
F	Moving Part Movement Speed
M	Event processing

Description of XMOVE Function Parameters

At a Glance

The following parameters must be entered for programming an interpolated movement function:

```
XMOVE (AXIS_CH3,N_Run,G9_,G,SPACE,X,Y,Z,F,M)
```

IODDT

`AXIS_CH3` is an IODDT type variable corresponding to channel 3 of the axis control module on which the function must apply. `AXIS_CH3` is of type `T_INTERPO_STD`

Movement Number

N_Run defines the movement number (between 0 and 32767). This number identifies the movement effected by the XMOVE function.

In debugging mode, this number is used to recognize the current movement.

Movement Type

G9_ defines the movement type:

Code	Movement type
90	Movement to an **absolute** position.
91	Movement to a relative position **with respect to the current position**.
98	Movement to a relative position **with respect to the stored position PREF1**

To choose the movement type, use the scroll button on the right of the **G9_** field, or enter the code directly with a "direct entry" (without going via the **Details** screen).

Instruction Code

G defines the XMOVE function instruction code *(see page 358)*.

Plane or Space Number

SPACE defines the plane or space number where the movement must take place. This parameter specifies the list of axes affected by the movement :

Code	Meaning
0	Movement in the XY plane.
1	Movement in the XZ plane.
2	Movement in the YZ plane.
3	Movement in the XYZ space.

NOTE: When the group of interpolated axes is in 2-D, the **SPACE** field must be at 0. The **Z** field for the XMOVE function becomes not significant and is disregarded.

Co-ordinates for Position to Reach

X, **Y**, **Z** defines the coordinates for reach position for channels 0, 1 and 2, or for the position which the moving part is heading for (in the case of a non-stop movement). This position can be:

- immediate,
- coded in pairs of %MDi internal words or %KDi internal constants (these words can be indexed).

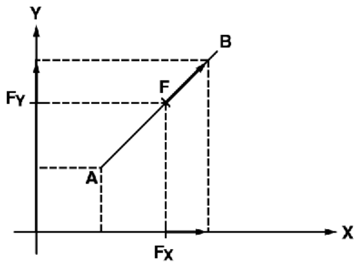
This value is expressed as a unit defined by the **Length units** configuration parameter for each axis.

Moving Part Movement Speed

F defines the speed of the moving object (or the movement speed required for the direction of movement to be carried out). This speed can be:

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

Example for a 2 axis system:



Using speed F, the interpolator calculates FX and FY projections which are used to guide the 2 axes on their trajectory.

The actual movement speed is equal to required speed F multiplied by the SMC speed modulation coefficient, whose value can be adjusted in the interval [0.001, 2.000].

M Parameter

M defines a word which codes nibbles (in hexadecimal):

- Activation or non-activation of the event processing application trigger for G05 and G10 instructions:
  - Nibble 3 to the value of 1: activating
  - Nibble 3 to the value of 0: non-activating
- List of events which can end instruction G05 or G10:
  - Nibble 1 :
    bit 0 for event entry, or EXT_EVT bit for X axis,
    bit 1 for event entry, or EXT_EVT bit for Y axis,
    bit 2 for event entry, or EXT_EVT bit for Z axis,
    bit 3 for group EXT_EVT bit.
    If several bits are set to 1, the event which ends the instruction is the first event on the resulting list. (the module carries out the logic OR of the events).

M Parameter :

Byte	3	2	1	0
16#				

NOTE: Coding is automatically completed in the **M** field on the **Details** screen, when the choices have been made using the check-boxes and buttons offered by the screen.

Instruction Codes for XMOVE Function

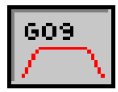
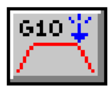
Introduction

The **G** parameter defines the instruction code.

To choose the instruction code, either use the scroll button on the right of the **G** field, or press on the icon which corresponds to the movement. You can also enter the code directly with a "direct entry" (without going via the **Details** screen).

Instruction Codes List

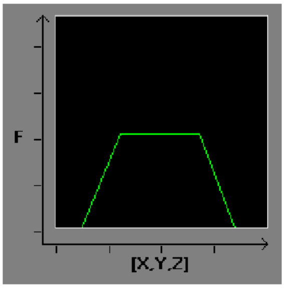
The instruction codes which can be chosen from the **Details** screen are as follows:

Instruction code	Indication	Icon
09	Movement on position with stop *(see page 362)*	
01	Movement on position without stop *(see page 361)*	
10	Movement until event with stop *(see page 363)*	
05	Await event *(see page 365)*	
92	Initializing the PREF1 registers for X, Y and Z axes *(see page 366)*	

Imaging on Details Screen

The **Details** screen also displays an image which represents the selected movement.

For example, code G09:



Description of Elementary Movements

At a Glance

2 types of movement category can be programmed:

- movements on a position (instruction codes 01 and 09),
- movements until event detection (instruction code 10).

During the program and movements, the reach positions, speed and interpolating plane / space can all be defined. The acceleration parameters are defined by adjusting them.

Types of Movement

Types of movement are as follows:

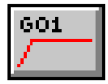
- Absolute in relation to the machine homing point (code 90).
  Example: non-stop movement on the XY plane,
  up to position (50000, 10000),
  up to a speed of 1000.
  XMOVE (AXIS_CH3,1,90,01,0,50000,10000,0,1000,0)
- Relative in relation to current position (code 91).
  Example: non-stop movement on the XY plane,
  an increment (+2000, -1000) in relation to the current position,
  up to a speed of 500.
  XMOVE (AXIS_CH3,1,91,01,0,1,2000,-1000,0,500,0)
- Relative in relation to PREF latched position (code 98)
  Example: non-stop movement on the XY plane,
  an increment (+2000, +5000) in relation to the previously latched position,
  up to a speed of 800.
  XMOVE (AXIS_CH3,(1,98,01,0,5000,2000,0,800,0)

With `AXIS_CH3` of type `T_INTERPO_STD`.

Programming a Move to Non-stop Position

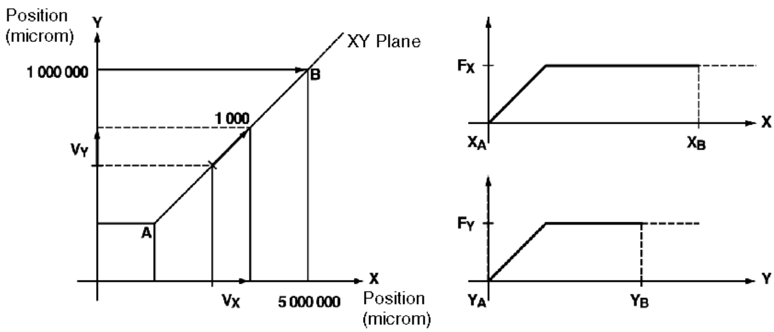
Instruction

The move to non-stop position instruction is as follows:

Instruction	Instruction code	Icon
Moving to non-stop position	01	

Example

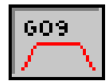
XMOVE (AXIS_CH3,1,90,01,0,5000000,1000000,01000,0)



Programming a Move to Position with Stop

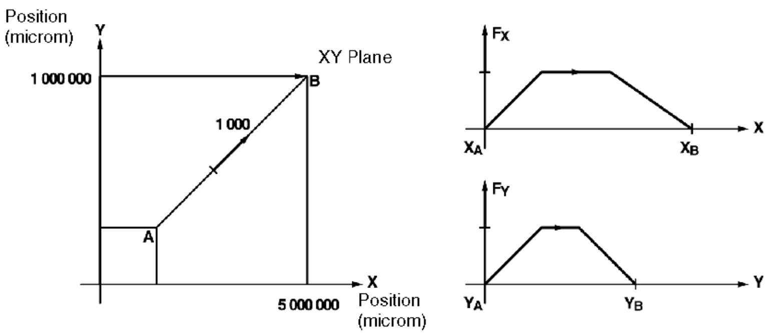
Instruction

The move to position with stop instruction is as follows:

Instruction	Instruction code	Icon
Moving to position with stop	09	

Example

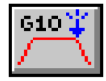
XMOVE (AXIS_CH3,1,90,09,0,5000000,1000000,01000,0)



Programming a Movement Until Event Detection

Instruction

The movement until event detection instruction is as follows:

Instruction	Instruction code	Icon
Movement until event detection with stop	10	

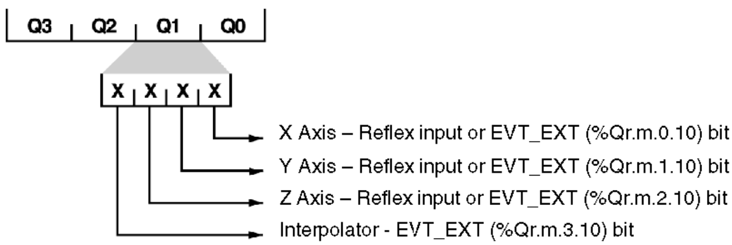
The G10 instruction moves axes until detection of an event or up to the position specified in the absence of an event.

Event

The awaited event can be:

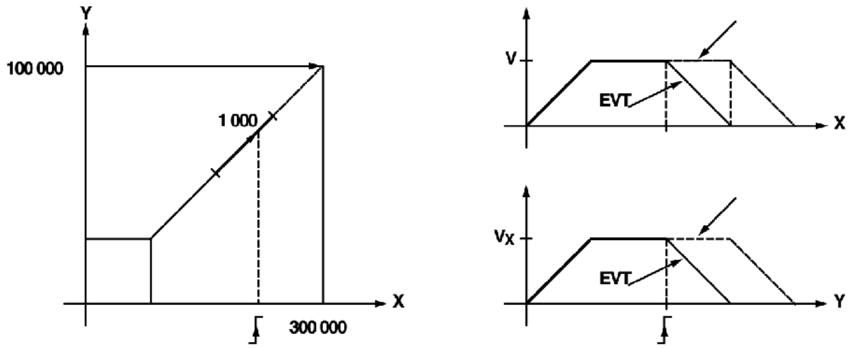
- A rising or falling edge (according to selections made while configuring) on one of the reflex inputs for one of the interpolated axes.
- A command from the application which can be:
  - a rising edge EVT_EXT bit for one of the axes,
  - a rising edge EVT_EXT interpolator bit,

Nibble 1 for parameter M is used to specify the axis (or axes) for the awaited event:



Example

Movement in the XY plane until detection of an EVT on X axis reflex input, at a speed of 1000. In the absence of an EVT, axes stop at (300000, 100000). Task event is activated on detection of the EVT



Programming an Await Event

Instruction

The await event instruction is as follows:

Instruction	Instruction code	Icon
Await event	05	

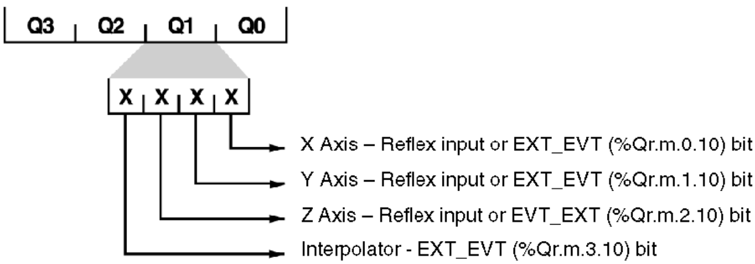
This instruction is used to wait for an event with a Time Out (in ms), defined in the F parameter. If an event does not appear within the Time Out period, the await command is then deactivated. If F parameter is defined at 0, the wait is without a time limit.

Awaited Event

The awaited event can be:

- a change in the status of a reflex input for one of the interpolated axes,
- a command from the application.

Nibble 1 for parameter M is used to set the axis (or axes) for the awaited event:



Event Task

The G05 instruction can activate an event task on detection of an event, if nibble 3 of parameter M is set to 1.

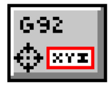
TO_G05 (Ir.m.c.49) bit is set to 1 when the Time Out period has expired without an event being detected, providing task event activation has been requested. For example, wait with a Time Out period of 1.5 s and with task event activation:

```
XMOVE (AXIS_CH3,1,90,05,0,0,0,0,1500,16#1000)
```

Programming PREF1 Register Intialization

Instruction

The initializing PREF1 registers instruction is as follows:

Instruction	Instruction code	Icon
Initializing PREF1 registers	92	

The G92 instruction is used to initialize PREF1 registers of different axes. These registers are used by relative movement instructions (code G98).

Nibble 1 of parameter M is used to select the list of axes affected by this initialization:

- bit 0 for X axis,
- bit 1 for Y axis,
- bit 2 for Z axis.

Example

Initializing PREF1 registers of X and Y axes to 2000 and 4000 respectively:

```
XMOVE (AXIS_CH3,1,90,92,0,2000,4000,0,0,16#0030)
```

General Conditions of Acceptance

Introduction

The general conditions of acceptance for the XMOVE function are as follows :

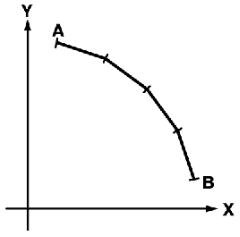
- There is no fault: bit GP_OK = 1.
- Configuration is compliant: bit CONF_OK = 1.
- Axes are referenced: bit REF_OK = 1.
- Axes affected by the movement are in automatic mode, where bits DONE = 1 and ENABLE = 1. These axes are also stopped.

Sequencing Movement Commands

Producing a Trajectory

TSX CAY 33 module does not offer circular interpolation. However it is possible to approximate any trajectory by a succession of segments.

Example of a 2 axis system trajectory:



An XMOVE command corresponds to each elementary segment of the trajectory.

NOTE: Each elementary XMOVE command must only be performed once. It is therefore necessary that the program to be executed is either

- in Grafcet: in one step, on activation or deactivation of this step,
- in structured text or contact language : on the rising edge of a bit.

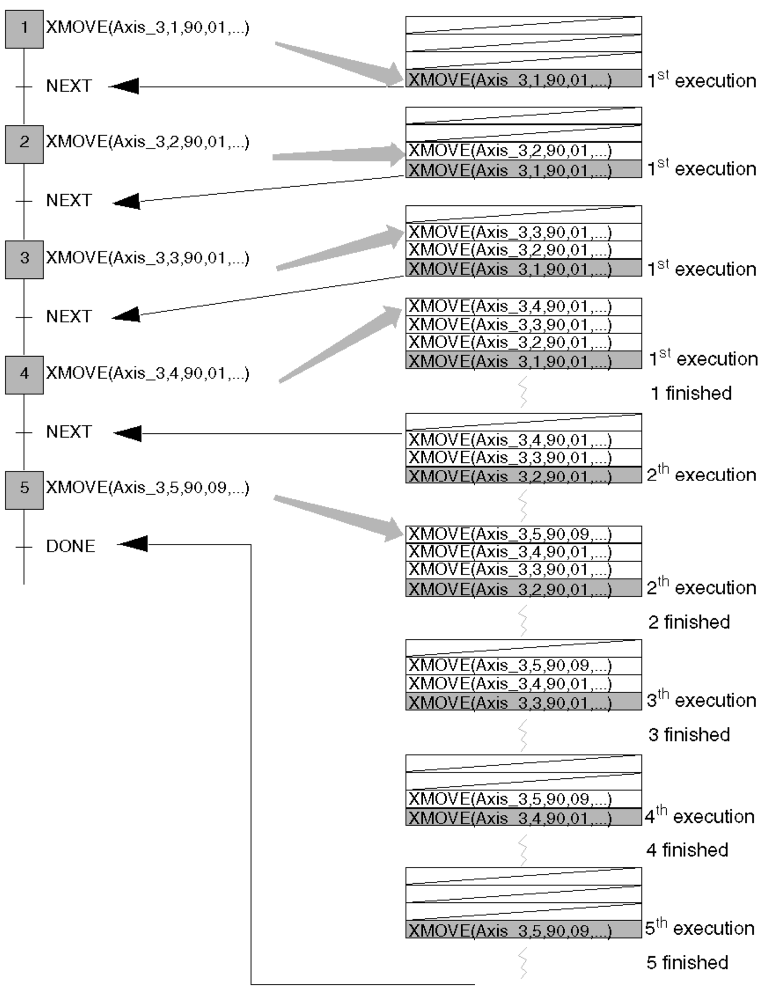
A report on function execution is provided by the module using the `NEXT` and `DONE` bits.

Buffer Memory

TSX CAY 33 possess a mechanism which is used to sequence movement commands.

The interpolator has a buffer memory (or stack) which can receive 3 movement commands in addition to what it is in the process of executing. Thus when a movement in progress has finished, it proceeds directly to the first command present in the buffer memory. In the following example we shall use the variable `Axis_3` of type `T_INTERPO_STD` associated with channel 3 of the TSX CAY 33 module

Sequencing mechanism



Empty Stack

When the stack is empty and a G1 type movement has been requested, it does not start if the module has not received the following movement.

Sequencing between 2 movement commands is as follows :

- instantaneously if the first movement is non-stop,
- as soon as the moving part is in the target window, or after time delay TSTOP has expired (which is defined in the stop control on parameter adjustment screen) if the first movement is with a stop.

Instant Sequencing

For the sequencing to be instantaneous, the instruction execution time must be more than the duration of the task in which the XMOVE commands have been programmed.

Command Refusal

Refusal of an XMOVE command is indicated by the following data:

- `CMD_NOK` bit (%Ir.m.3.6) which indicates a refusal,
- `CMD_FLT` (%MWr.m.3.7) word which indicates the cause of the refusal. Reading this word requires a `READ_STS` instruction.

Constraints on XMOVE Movements

The following XMOVE movements cause command refusal (`CMD_NOK`), stop the moving part and reset the buffer memory to zero.

- sequencing a G05 or G92 after a G01,
- absence of an instruction after a G01,
- receiving a command with a SPACE parameter affecting an axis which is not stationary, while it is not being affected by a previous XMOVE command (this is where an XMOVE is with an axis whose last movement was an SMOVE G1).

Bits Associated with a Sequencing Mechanism

The following bits are associated with the sequencing mechanism:

Bit	Description
`NEXT` (%Ir.m.3.0)	Indicates to the program application that channel 3 is ready to receive the following XMOVE command.
`DONE` (%Ir.m.3.1)	Indicates the command in progress has finished being executed and there are no new commands in the buffer memory.
`TH_PNT` (%Ir.m.3.10)	Indicates that the setpoint value has been reached on the axes affected by the XMOVE.
`AT_PNT` (%Ir.m.3.9)	At the end of a movement with a stop, this indicates that for all the axes affected by the movement the moving part is in the target window.

NOTE: Either the `NEXT` bit, or the `DONE` bit must be tested before executing an XMOVE command.

A new command must only be sent to the module if the buffer memory associated with the axis to be guided is not full.

The `SYNC_N_RUN` (%Ir.m.3.8) word periodically provides the number of the step in progress, in order to carry out movement sequencing.

Concurrent XMOVE and SMOVE Functions

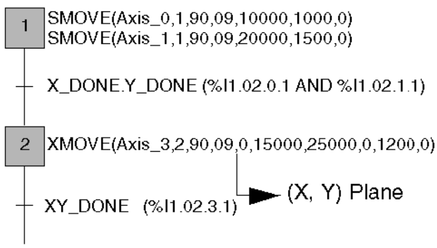
At a Glance

It is possible to combine movement instructions for a single axis (SMOVE) with movement instructions for several axes (XMOVE) within the program. This can be used to alternate interpolated movements with non-interpolated movements.

The program must refer to objects for the axis concerned before sending an SMOVE, and must also refer to objects for channel 3 before sending an XMOVE. We shall use the following three IODDT variables: `Axis_0`, `Axis_1` and `Axis_3`.

Example

Independent X and Y axis movement followed by an interpolated movement :



An interpolated axis movement prevents any SMOVE guidance relating to that axis: a running XMOVE forces `NEXT` and `DONE` bits to 0 for the axes concerned.

In addition, the `IN_INTERPO` (%Ir.m.c.32) bit for any moving axis affected by an XMOVE is set at 1. This information, which can be accessed from the application, is designed to help with programming and monitoring.

Interpolator Channel Automatic Mode

At a Glance

The **Automatic** mode is the active mode for interpolated axes. Interpolated movements can only be carried out with this mode.

Automatic mode is mainly used to send a movement command (code G) via an XMOVE function. This command is designed to perform an interpolated movement by momentarily creating a link between several axes.

Channel 3 in Automatic Mode

Channel 3 in automatic mode does not modify the current mode, nor 2 (or 3) module axis commands in progress. In this way movements (or debugging) which are carried out independently axis by axis in manual, loop disabled control and automatic modes, are always performed via the positioning function for each axis in the module.

The actual engagement of automatic mode is indicated by the **IN_AUTO** (%Ir.m.3.23) bit.

Commands in Automatic Mode

In automatic mode, the following commands are used to act on the XMOVE function:

- **CMV**: speed modulation coefficient. This coefficient affects the current tangent speed setpoint in a ratio of 1/1000 to 2000/1000 (%QWr.m.3.1)
- **CMV = 0**: **Feed hold** command which stops the moving part, while ensuring that upon the resume movement command (CMV #0), the programmed trajectory is followed. **Feed hold** status is indicated by the **IM_PAUSE** bit (%Ir.m.3.34).
- **Pause** : Pause command which is used to suspend XMOVE movement sequence. The pause is only active when the moving part is stationary. This is indicated by the **ON_PAUSE** bit (%Ir.m.3.33).
- **MOD_STEP** (%Qr.m.3.19) : is used to execute a movement sequence by stopping after each elementary instruction. Status is indicated by the **ST_IN_STEP** bit (%Ir.m.3.39). **NEXT_STEP** bit (%Qr.m.3.22) is used to execute the next step.
- **EXT_EVT** (%Qr.m.3.10): is used to terminate a G05 or a G10.

The automatic mode can also access the following 2 commands which can be active during or outside of an XMOVE:

- **STOP** (%Qr.m.3.8): is a stopping order for the different axes which make up the interpolation (according to the role of the STOP command that has been defined in the configuration).
- **ACK_FLT** (%Qr.m.3.10): a rising edge causes fault acknowledgement.

Event Processing with Interpolated Axes

At a Glance

Channel 3 on a TSX CAY 33 module can activate an event task. To do this, the functionality must have been enabled on the configuration screen by associating an event processing number to the channel.

Activating an Event Task

An event task is activated by the appearance of an event expected by G10 and G05 commands. For this to happen, nibble 3 from XMOVE function parameter M associated with the instruction must be equal to 1.

Usable Variables for the Event Task

- If several event sources are chosen, the following bits are used to determine the source trigger for event processing:
  - **EVT_G1** (%Ir.m.3.50): event during instruction G10,
  - **EVT_G05** (%Ir.m.3.48): event during instruction G05,
  - **TO_G05** (%Ir.m.3.49): G05 timer expired
- **OVR_EVT** bit (%Ir.m.3.46) is used to detect a delay in event sending or an event loss.

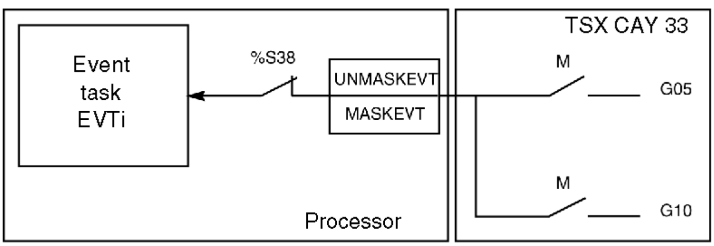
NOTE: The words and bits described above are the only ones which are refreshed during execution of a task event. They are only updated in the PLC if the task event has been activated.

Event Masking

Control Expert language offers 2 ways to mask events:

- Instruction for global event masking : MASKEVT (UNMASKEVT is used for unmasking):
- ACTIVEVT (%S38) bit = 0 (global event inhibition). ACTIVEVT (%S38) bit is normally at 1.

Summary diagram



Fault management

At a Glance

Fault monitoring is essential for axis command tasks, due to the inherent risks posed by active moving parts.

The moving part carries out checks internally and automatically.

Fault Monitoring for Interpolated Axes

TSX CAY 33 channel 3 does not have a specific fault.

Fault information indicated by this channel is the logic OR of faults on axes configured in the interpolated movement (channel 0 and 1 faults in the case of 2 axis interpolation; channel 0, 1 and 2 faults in the case of 3 axis interpolation).

Fault Information

Fault information is as follows:

Bit	Fault information
EMG_STP (%MWr.m.c.3.5)	Emergency stop
DRV_FLT (%MWr.m.c.3.2)	Speed drive fault
ENC_BRK (%MWr.m.c.3.4)	Encoder break
ANA_FLT (%MWr.m.c.3.0)	Analog output short circuit
AUX_FLT (%MWr.m.c.3.1)	Auxiliary output short-circuit
ENC_SUP (%MWr.m.c.3.3)	Encoder supply
AUX_SUP (%MWr.m.c.3.6)	24 V supply
ENC_FLT (%MWr.m.c.3.7)	Absolute encoder frame
SLMIN (%MWr.m.c.3.9)	Software lo limit
SLMAX (%MWr.m.c.3.8)	Software hi limit
SPD_FLT (%MWr.m.c.3.10)	Overspeed
FE1_FLT (%MWr.m.c.3.11)	Blocking following error
FE2_FLT (%MWr.m.c.3.15)	Non-blocking following error
REC_FLT (%MWr.m.c.3.12)	Recalibration on the fly deviation
TW_FLT (%MWr.m.c.3.13)	TW Debugging window
STP_FLT (%MWr.m.c.3.14)	TSTOP Stopping speed

NOTE: Channel fault feedback for channel 3 is the same as that for an independent axis. Fault information is only refreshed during the execution of a READ_STS instruction (AXIS_CH3).

Levels of Severity

Faults are classed in 2 levels of severity:

- **Critical or blocking** faults which cause the moving part to stop. The following processes then occur:
  - the fault is indicated,
  - the moving part slows down until analog output is zero,
  - deactivation of the speed drive enable relay,
  - clearing of all memorized commands,
  - wait for acknowledgement.

  The fault must have disappeared and been acknowledged before the application can be re-started.
- **Non-critical faults** which give rise to fault indication without stopping the moving part. The Control Expert must be programmed with what action to take in the event of this type of fault. The fault indication disappears when the fault itself has disappeared and been acknowledged.

Fault Programming

Faults can be viewed, corrected, and acknowledged from the debugging screen. However it can be useful to be able to guide the moving part and correct faults from a console during operation. For this purpose, all the information and commands necessary are available within the application.

Fault Indication

The module supports a wide range of information in the form of bits and status words, all accessible through the Control Expert program. These bits are used to process faults in hierarchical order :

- to act on the main program,
- to simply indicate the fault.

Level of Indicating

2 indicating levels are provided:

First level: general information

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

Second level: detailed information

Channel fault status word AX_STS (%MWr.m.c.3)

NOTE: With a blocking fault, it is advised to stop the changing sequential processing which is associated with the axes, and correct the fault. Correcting the fault must be followed by a fault acknowledgement.

Fault Acknowledgement

When a fault appears on one of the interpolated axes:

- The axis fault bits: AX_FLT (%Ir.m.c.2), HD_ERR (%Ir.m.c.4), AX_ERR (%Ir.m.c.5) and AX_STS (%MWr.m.c.3.j), as well as the faults bits associated with interpolating: AX_FLT (%Ir.m.c.2), HD_ERR (%Ir.m.c.4), AX_ERR (%Ir.m.c.5) and status bits (%MWr.m.c.3.j) are set to 1.
- If the fault is a blocking fault, the AX_OK (%Ir.m.c.3) bit is set to 0.

When the fault disappears, all fault bits retain their status. Faults are stored until they have been acknowledged through setting to 1 of the ACK_DEF bit (%Qr.m.c.8) where c = channel number which has the fault or the ACK_DEF interpolation channel bit. Setting to 1 of the channel 3 ACK_DEF bit generates acknowledgement for all interpolated axes. Fault acknowledgement must be carried out after the fault has disappeared (except for soft stop faults).

If several faults are detected, the acknowledgement order will only act upon faults which have effectively disappeared. Persisting faults must by acknowledged again after their disappearance.

NOTE: Channel 3 (interpolator) does not store faults.

Description of Command Refused Faults

At a Glance

A command refused fault is generated each time a command cannot be executed. This occurs when a command is not compatible with the axis state, with the mode in progress or where at least one of the parameters is not valid.

These faults are indicated by the Refus Cde LED on the debugging screens. At channel level, the DIAG key can be used to identify the source of the command refused fault. This information can also be accessed by the program with the CMD_NOK (%Ir.m.c.6) bit and CMD_FLT (%MWr.m.c.7) word.

Command Refused

Th following table shows the cause, indication and the remedy in the event of a **Command Refused** fault.

Cause	Unauthorized movement command Transfer of faulty configurations or parameters
Parameter	None
Result	Immediate stop of movement in progress Buffer memory, which receives movement commands in automatic mode, is reset to 0.
Indication	CMD_NOK bit (%Ir.m.c.6): Movement command refused CMD_FLT mot (%MWr.m.c.7): type of fault detected ● Least significant byte: movement commands, ● Most significant byte: configuration and parameter adjust.
Remedy	When a new command is received and accepted, acknowledgement is implicit. Acknowledgment is also possible via the ACK_FLT (%Qr.m.c.8) command

NOTE: For movement sequences in automatic mode, it is advisable to make the execution of each movement conditional upon the end of the previous movement, and the AX_FLT (%Ir.m.c.2) bit. This will prevent the program moving on to the following command should the present command be refused..

Managing Measurement (OFF) Mode

Introduction

Measurement (OFF) mode is the interpolator passive mode: the X, Y and Z axes are in independent axis state. It is therefore possible to command them in loop control disabled mode (DIRDRIVE) in manual or automatic mode.

No channel 3 command is taken into account in this mode, except for the fault acknowledgement command.

Stopping XMOVE

If an XMOVE is in progress, switching to OFF mode will make it stop.

Chapter 18
Interpolation Configuration

Subject of this Section

This section describes the TSX CAY interpolation configuration screens (channel 3) as well as the parameters, which must be defined for the interpolated axes.

What Is in This Chapter?

This chapter contains the following topics:

Accessing the Interpolation Parameters Configuration Screen

Preliminaries

Channel 3 is dedicated to interpolation. Before configuring channel 3, the independent channels affected by the interpolated movement must be configured.

Accessing Interpolation Parameters

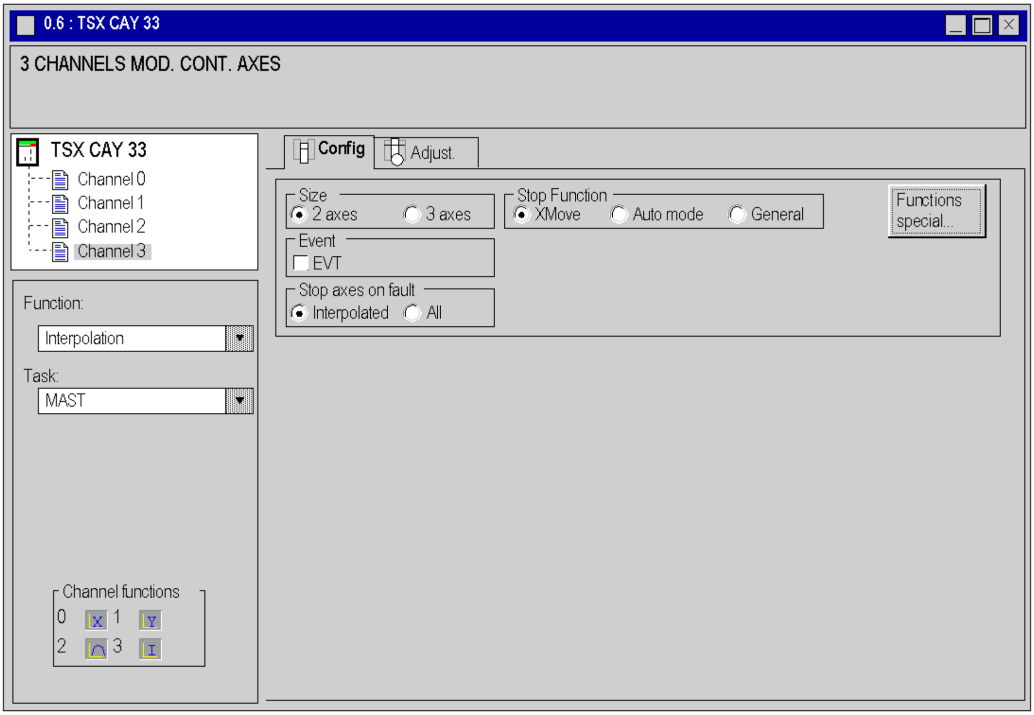
To access interpolation parameters, select the TSX CAY 33 module and confirm (double-click on the picture of the module).

Step	Description
1	Select **channel 3** in the **Channel** field
2	Select the **Interpolation** function from the **general parameters** field
3	In the **general parameters** field, select the task associated with channel 3: **MAST** or **FAST**. The task selected (MAST or FAST) must be the same for channel 3 as for the other channels affected by the interpolation. If a task is different, a dialog box appears during confirmation to indicate the channel number which does not have the same task as channel 3.

NOTE: The interpolated channel axes must be a **limited** type. It is not possible to interpolate infinite axes.
If an infinite axis has been selected, a dialog box will appear during confirmation to indicate that the channel number is not a limited axis type.

Interpolation Parameters Configuration Screen

The following screen is used to set the parameters for channel 3:

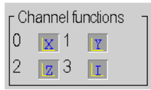


Entering Interpolation Parameters

Channel Functions

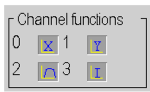
The **general parameters** zone reports on the axes affected by interpolation.

The following example represents a 3 axis interpolation:



- Channel 0 is the X axis.
- Channel 1 is the Y axis.
- Channel 2 is the Z axis.
- Channel 3 is the interpolation axis for X, Y and Z axes (letter I).

The following example represents a 2 axis interpolation:



Channel 2 may be configured as an independent channel (a curve replaces the letter Z in field 2). In this case, interpolation will only affect channels 0 and 1. Functions are displayed on the interpolation debugging screens.

Dimension

The Dimensions field is used to set the number of interpolated axes :

Command	Description
2 Axes	Channels 0 and 1 are interpolated
3 Axes	Channels 0, 1 and 2 are interpolated

Stop Function

This field is used to set the role of the STOP command in channel 3 (%Qr.m.3.15):

Command	Description
XMOVE	The STOP command is only effective on an XMOVE which is in progress.
Auto mode	The STOP command is active in auto mode and affects all axes which can be interpolated, even if they are used independently.
General	The STOP command is active in all modes (eg Auto, Manu etc) and affects all axes which can be interpolated, even if they are used independently.

NOTE: It is advised to chose the **XMOVE** command as a default.

Event

This field is used to select the event task associated with channel 3.

Stop Axes on Fault

This field is used to define the effect of a blocking fault:

Command	Description
Interpolated	A blocking fault stops axes affected by the XMOVE command which is in progress.
All	A blocking fault stops all axes which can be interpolated, even if they are being used independently when the fault appears.

NOTE: It is advised to choose the **Interpolated** command as a default.

Special Functions

This command is reserved for special usage.

Confirming Configuration Parameters

When all parameters have been configured, confirm this by using the **Edit** → **Confirm** command, or by activating the icon

Chapter 19
Adjusting Interpolated Axes

Subject of this Section

This section describes the parameter adjustment principle for the interpolation channel (channel 3 of module TSX CAY 33): accessing screens, description of parameters and adjustment procedure.

What Is in This Chapter?

This chapter contains the following topics:

Accessing the Interpolation Adjustment Parameters

At a Glance

The adjustment screen is a graphic tool intended for adjusting *(see EcoStruxure™ Control Expert, Operating Modes)* the TSX CAY 33 selected in a rack. It displays the current and initial parameters associated with the channels of this module, and allows these to be modified in offline and online modes.

Accessing the Parameters

The adjustment screen is used to select the channel to be adjusted, and enables access to current or initial parameters.

Command	Function
Choose Axis	Choose channel 3.
	This button displays either the current parameters or the initial parameters. This function can also be performed using the F7 key.

Initial Parameters

The initial parameters are:

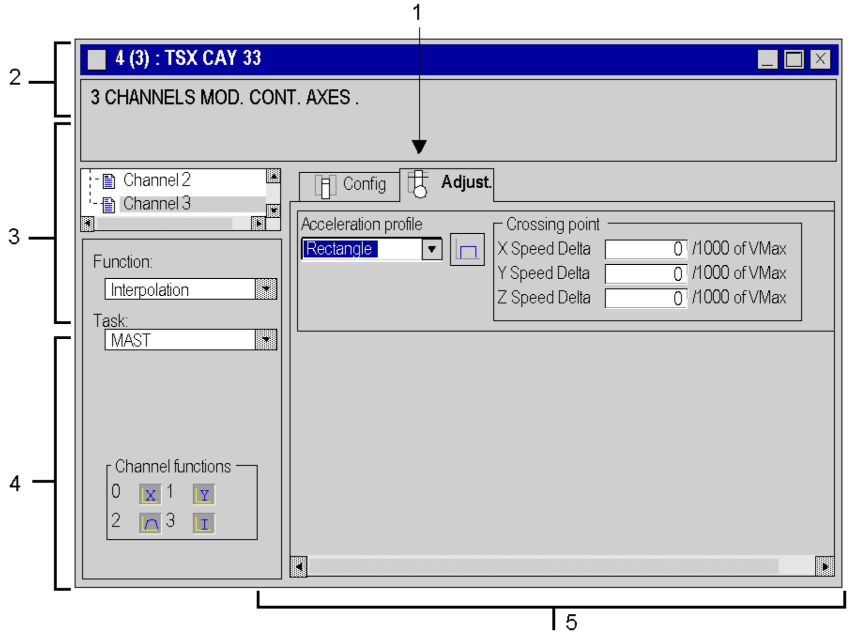
- Parameters entered (or defined by default) in the configuration screen in offline mode. These parameters have been confirmed in the configuration, and transferred to the PLC.
- Parameters taken into account during the last reconfiguration in online mode.

Current Parameters

Current parameters are those which have been modified and confirmed from the adjustment screen in online mode (or by program via an explicit exchange). These parameters have been replaced by the initial parameters after a cold restart.

Adjustment Parameters

The diagram below shows an adjustment screen.



Description

The following tables presents the various elements of the adjustment screen and their functions.

Address	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 available modes are: ● **Adjustment** ● **Configuration** ● **Debugging** (or Diagnostics), accessible only in online mode.
2	**Module** zone	Shows the abbreviated title of the module.
3	**Channel** field	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** field	Allows you to choose the axis control function and the task associated with the channel: ● **Function:** axis control function among those available for the modules involved. Depending on this choice, the headings of the configuration zone may differ. By default **No** function is configured. ● **Task:** defines the (**MAST**, **FAST** or **AUX0/1**) task in which the explicit exchange objects of the channel will be exchanged.
5	**Adjustment** field	This field allows you to define the various values of the adjustment parameters.

Acceleration Profile

Description of Acceleration Profile Parameter

The acceleration law is common to all the interpolated axes.

Parameter	Indication
Acceleration profile	Acceleration law applied to moving part. This law, common to all the axes affected by an interpolated movement, replaces the current axis parameter during the movement. By default : Rectangle

Acceleration Profiles

The following acceleration profiles *(see page 270)* can be used:

- Rectangle
- 1, 2 or 3 trapezoid
- Triangle.

Crossing Points

Description of Speed Delta Parameters

These 3 parameters (one parameter for each interpolated axis) are used to set:

Adjustment	Meaning
Speed Delta X Speed Delta Y Speed Delta Z	Variation of speed allowed on crossing points for each axis. The speed adjustment for the moving part at the crossing point allows the moving part to pass closer to the target point when the defined value is low. These parameters are expressed as a thousandth of VMAX. Limits: 0 to 500

Processing Crossing Points

In a linear interpolation, when a succession of non-stop G01 type movements are performed, the concept of crossing points becomes clear.

For example, let us assume that an ABC trajectory is required.

A specified speed maintained up to B on the AB segment causes an overshoot (figure 1). If the speed is reduced before arriving at B, then the real trajectory remains inside the ABC angle (figure 2).

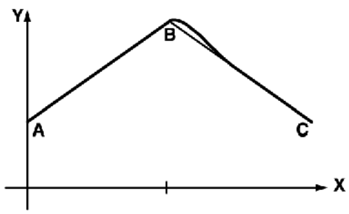


Figure 1

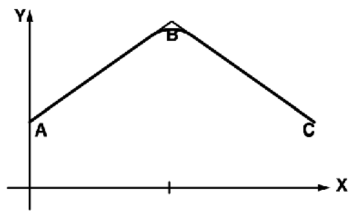


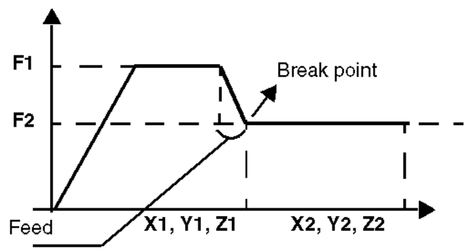
Figure 2

Overshooting

How to Avoid Overshooting:

- The natural delay (following deviation) of each axis is used. To do this, it is advised to moderate the feed forward gain adjustment KV during interpolated movements.
- In a sequence (G01, X1, Y1, Z1, F1) followed by another sequence (G01, X2, Y2, Z2, F2), if F2 is smaller than F1, the speed trajectory is modified so that the desired speed is equal to F2 at the break point.

The following figure illustrates the modification of the speed trajectory so F2 speed is reached on the break point:



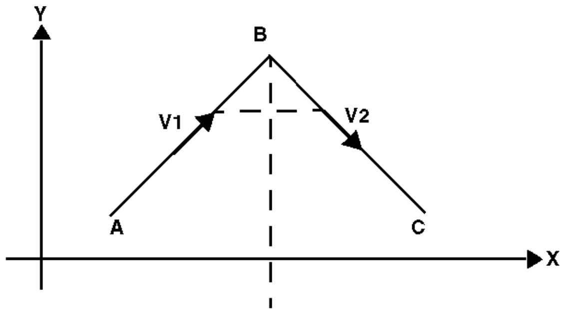
- Role of the parameter: DELTASPEEDPATH
  The module introduces a mechanism, which can reduce the speed at the point of crossing, according to the requested "Delta speed" (axial Delta speed X, Y or Z). The function is implemented from the point at which the VMAX can be adjusted.
  The function is active when:
  $\Delta V_{(\text{of an axe})}$ > VMAX * DELTASPEEDPATH/1000

NOTE: The smaller the DELTASPEEDPATH parameter, the closer the point.

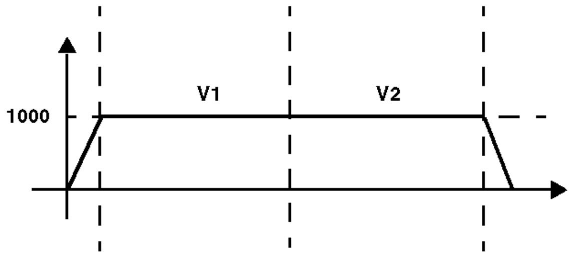
Example

V1 = V2 = 1000

$VMAX_X = VMAX_Y = 4000$

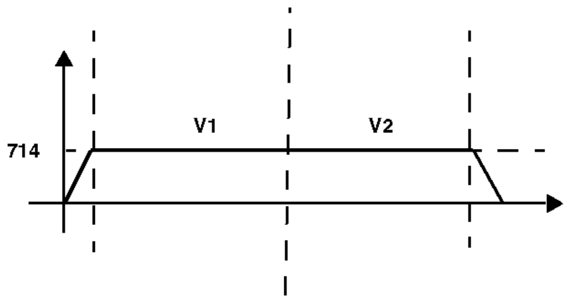


- **Crossing points mechanism inactive**
  Resulting tangential speed

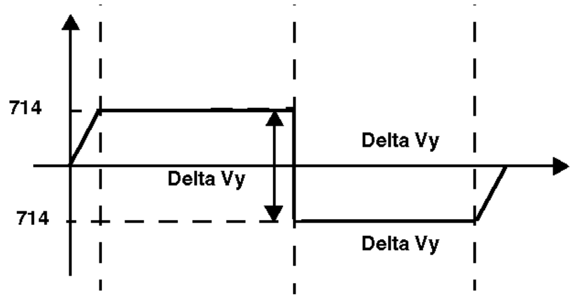


Axial speed X

$(V_X{}^2 + V_Y{}^2)^{1/2}$



Axial speed Y



Δ V_Y < VMAX * DELTASPEEDPATH/1000

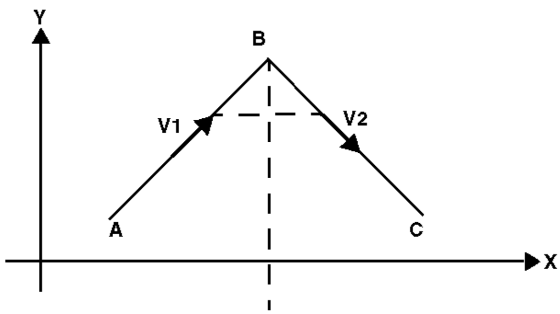
DELTASPEEDPATH = 500

Δ V_Y = 1428 < 1/2 * VMAX

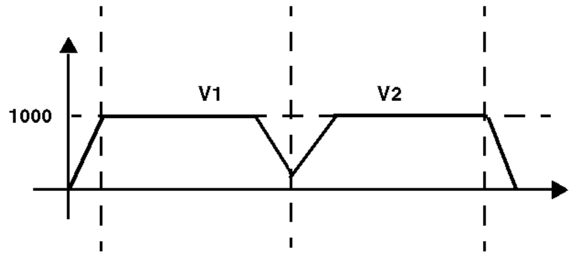
Further Example

V1 = V2 = 1000

$VMAX_X = VMAX_Y = 4000$

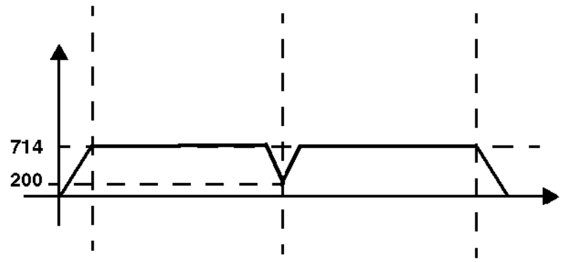


- **Crossing points mechanism active**
  Resulting tangential speed

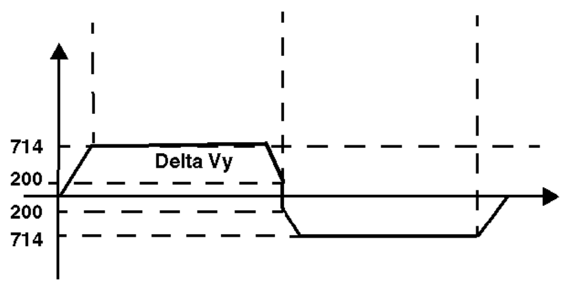


Axial speed X

$(V_X{}^2 + V_Y{}^2)^{1/2}$



Axial speed Y



DELTASPEEDPATH = 100

Δ V_Y = 1428 > 1/10 * VMAX

In this case a = 1/10 VMAX = 400

Chapter 20
Debugging an Interpolated Axis Control Program

Subject of this Section

This section describes the parameter adjustment principle for the interpolation channel (channel 3 of module TSX CAY 33): Measurement, Automatic. It also describes the diagnostics screen which gives access to possible faults.

What Is in This Chapter?

This chapter contains the following topics:

Principles of Debugging an Interpolated Axis Control Program

At a Glance

Axis control is integrated into Control Expert programming, for which debugging functions are used.

Reminder of the Possibilities Offered with Control Expert

Refer to principle for debugging an independent axis *(see page 282)*.

Debugging Screen

The task-specific debugging screen, unique to TSX CAY 33 module channel 3, allows access to all the information and commands required for debugging interpolation.

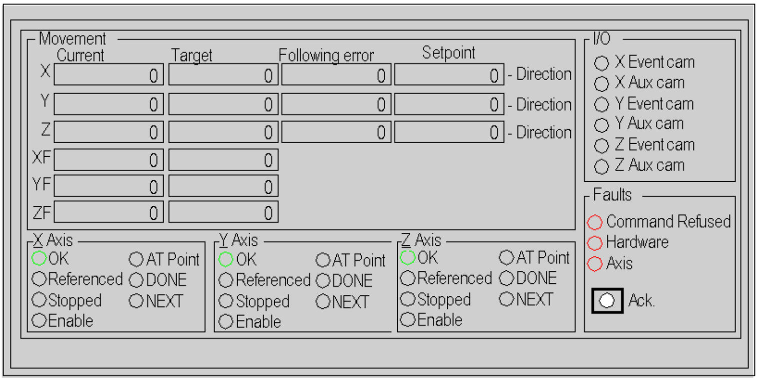
The program and moving part control zone offers 2 possible screens according to the operating mode selected via the mode switch: Measurement (OFF) or Automatic (AUTO).

Measurement Mode (Off)

At a Glance

In this mode, the interpolation channel can be used to view either 2 or 3 interpolated axes. This mode supervises the module axes.

The following screen shows the interpolation channel Off mode:



Description of Movement Field

This table shows the display zones for the Movement field:

Display zone	Description
Current X, Y, Z	Displays the current position of the moving part on the X, Y and Z axes respectively. This value is expressed in measurement units defined in the configuration
Target X, Y, Z	Displays the setpoint position of the moving part (reach position) on the X, Y and Z axes respectively
Current XF, YF, ZF	Displays the current speed of the moving part in measurement units (defined in the configuration) on the X, Y and Z axes respectively
Target XF, YF, ZF	Displays the setpoint speed of the moving part (speed to be attained) on the X, Y and Z axes respectively
Following error X, Y, Z	Displays the deviation between the calculated setpoint position and the real position of the moving part on the X, Y and Z axes respectively
Setpoint X, Y, Z	Displays the go to coordinates on the X, Y and Z axes respectively
Direction X, Y, Z	+ Direction: indicates the part is moving in a positive direction on the X, Y and Z axes respectively - Direction: indicates the part is moving in a negative direction on the X, Y and Z axes respectively

Description of X Axis, Y Axis and Z Axis Fields

This table shows the display zones for the Axis fields:

LED	State	Indication
OK	Lit	Axis in operational state (no blocking fault)
Referenced	Lit	Referenced axis
Stopped	Lit	Moving part stationary
Enable	Lit	Variable speed controller enable relay active
AT Point	Lit	The movement in progress is finished and the moving part is in the target window
DONE	Lit	The movement in progress is finished
NEXT	Lit	The next movement may be sent

NOTE: The **Confirm** command in the **Axis** field is used to control the variable speed controller enable relay.

Description of I/O Field

This table shows the display zones for the I/O field:

LED	Indication
X, Y or Z Came Evt	State of signal (0 or 1) on **Event** input for X, Y or Z axes
Aux X, Y or Z	State of signal (0 or 1) on auxiliary input for X, Y or Z axes

1 = LED lit, 0 = LED off

Description of Faults Field

This table shows the display zones for the Faults field:

LED / Button	State	Indication
Command Refused	Lit	Last movement command refused
Hardware	Lit	External hardware fault (e.g. encoder, variable speed controller, outputs, etc.)
Axis	Lit	Application fault (e.g. following error, software limits, etc.)
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

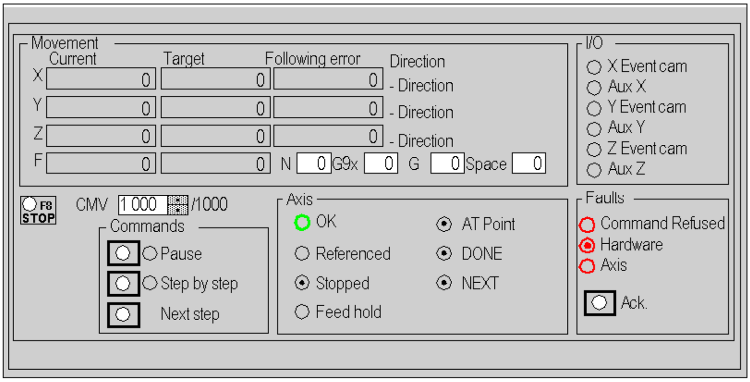
Automatic Mode (Auto)

At a Glance

Automatic mode is used for executing XMOVE functions.

The debugging screen displays information for 2 or 3 axes, according to the number of interpolated axes.

The following screen shows interpolation channel automatic mode (for 3 interpolated axes):



Description of Movement Field

This table shows the display zones for the Movement field:

Display zone	Description
Current X, Y, Z	Displays the current position of the moving part on the X, Y and Z axes respectively. This value is expressed in measurement units defined in the configuration
Target X, Y, Z	Displays the setpoint position (reach position) on the X, Y and Z axes respectively
F Current	Displays the current moving part speed using the measurement unit defined in configuration.
F Target	Displays the moving part setpoint speed (speed to be attained)
Following error X, Y, Z	Displays the deviation between the calculated setpoint position and the real position of the moving part (the following error) on the X, Y and Z axes respectively
N	Indicates the step number for the instruction which is in progress
G9x	Indicates the movement type for the instruction which is in progress
G	Indicates the instruction code which is in progress
Space	Indicates the space where the movement in progress is being carried out (0 = XY, 1 = XZ, 2 = YZ, 3 = XYZ)
Direction X, Y,	+ Direction: indicates the part is moving in a positive direction on the X, Y and Z axes respectively - Direction: indicates the part is moving in a negative direction on the X, Y and Z axes respectively

Description of Commands

This table describes the command zone:

Command	Description
STOP F8	Stops the moving part on all interpolated axes
CMV	Used to enter a value from 0 to 2000 which determines the speed multiplier coefficient (0.000 to 2000 in intervals of 1/1000)

Description of Commands Field

This table describes the buttons in the Commands field:

Command	Description
Pause	Commands the moving part to stop at the end of a movement with a stop in progress
Step by step	Changes the axis to step by step mode
Next step	Step by step mode activates the waiting movement

Description of Axis Field

This table shows the display zones for the Axis field:

LED	State	Indication
OK	Lit	All axes are in operating state
Referenced	Lit	All axes are referenced
Stopped	Lit	All axes are at a stop (moving part stationary)
Feed hold	Lit	Axis movement is suspended (Feed hold)
AT Point	Lit	The movement in progress is finished and the moving part is in the target window
DONE	Lit	The movement(s) in progress is (are) finished
NEXT	Lit	The following movement command can be sent to the module

Description of I/O Field

This table shows the display zones for the I/O field:

LED	Indication
X, Y or Z Came Evt	State of signal (0 or 1) on **Event** input for X, Y or Z axes
Aux X, Y or Z	State of signal (0 or 1) on auxiliary input for X, Y or Z axes

1 = LED lit, 0 = LED off

Description of Faults Field

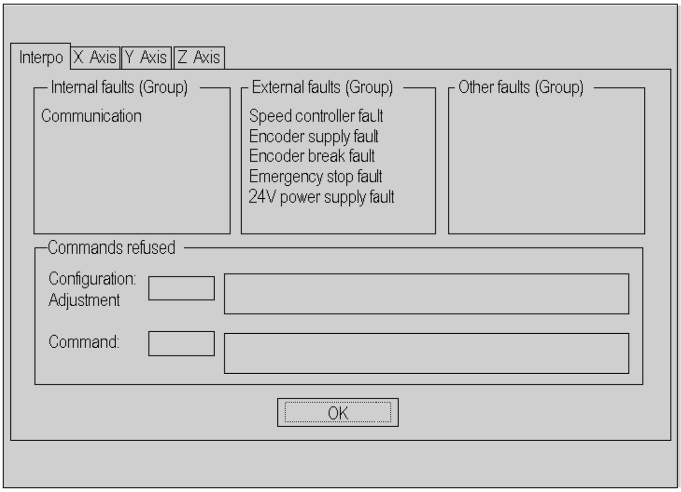
This table shows the display zones for the Faults field:

LED / Button	State	Indication
Command Refused	Lit	The last XMOVE received has been refused
Hardware	Lit	External hardware error on one of the interpolated axes
Axis	Lit	Application fault on one of the interpolated axes
Ack.	/	Fault acknowledgment button. Activating this button acknowledges all faults which have disappeared

Interpolation Diagnostics

At a Glance

The various debugging, adjustment and configuring screens feature a **DIAG** button in online mode, which accesses details on faults detected by the module.



Diagnostics Screen Tabs

The diagnostics screen for channel 3 features 4 tabs which give access to possible faults for all the interpolated axes:

Tab	Description
Interpo	Globalizes faults for all the interpolated axes
X Axis	Displays channel 0 faults
Y Axis	Displays channel 1 faults
Z Axis	Displays channel 2 faults

Description of the Different Fields

Each tab features the following fields:

Field	Description
Internal faults	Internal faults within the module which generally require it to be replaced
External faults	Faults originating from the operating part
Other faults	Application faults
Commands refused	Indicates the cause and the message number of a command refused *(see page 414)*

Chapter 21
Language Objects of the Interpolated Axis Specific Application

Aim of this Chapter

This chapter describes the language objects associated with the axis specific application as well as the different ways of using them.

Refer to the chapter on the language objects associated with independent axes *(see page 313)*.

What Is in This Chapter?

This chapter contains the following topics:

Implicit Exchange Internal Command Objects of the T_INTERPO_STD Type IODDT

List of Implicit Exchange Objects

The following table presents the implicit exchange internal command objects of the T_INTERPO_STD type IODDT

Standard symbol	Type	Access	Active on	Description	Address
ACK_FLT	EBOOL	R/W	Edge	Fault acknowledgement	%Qr.m.c.8
EXT_EVT	EBOOL	R/W	Edge	Trigger event order from processor	%Qr.m.c.10
AUX_OUT	EBOOL	R/W	State	Auxiliary output command	%Qr.m.c.11
STOP	EBOOL	R/W	State	Immediate stop command (halting of moving part)	%Qr.m.c.15
PAUSE	EBOOL	R/W	State	Suspend movements command at the end of a movement in progress	%Qr.m.c.16
MOD_STEP	EBOOL	R/W	State	Change to step by step mode command	%Qr.m.c.19
NEXT_STEP	EBOOL	R/W	Edge	Activate next step command	%Qr.m.c.22
MOD_SELECT	INT	R/W		mode selector	%QWr.m.c.0
CMV	INT	R/W		speed modulation Value = speed modulation setpoint value This setpoint is in the range 0 to 2, in intervals of 1/1000.	%QWr.m.c.1

Mode Selector

MOD_SELECT: mode selector

Value	Mode	Description
0	DRV_OFF	Measurement mode: inhibition of CNA output
1	DIRDRIVE	Loop control disabled mode: direct voltage command
2	MANU	Manual mode
3	AUTO	Automatic mode

Implicit Exchange Internal Status Objects of the T_INTERPO_STD Type IODDT

List of Implicit Exchange Objects

The following table presents the implicit exchange internal status objects of the T_INTERPO_STD type IODDT

Standard symbol	Type	Access	Description	Address
NEXT	EBOOL	R	Ready to receive a new movement command (in AUTO)	%Ir.m.c.0
DONE	EBOOL	R	All instructions are executed: no instructions in the stack	%Ir.m.c.1
AX_FLT	EBOOL	R	Fault present on axis	%Ir.m.c.2
AX_OK	EBOOL	R	No fault causing moving part to stop	%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.8
AT_PNT	EBOOL	R	Moving part position on target (in the point window, on instruction with stop)	%Ir.m.c.9
TH_PNT	EBOOL	R	Theoretical setpoint reached	%Ir.m.c.10
CONF_OK	EBOOL	R	Configured axis	%Ir.m.c.12
REF_OK	EBOOL	R	Reference point taken (axis referenced)	%Ir.m.c.14
IN_DROFF	EBOOL	R	Measurement mode active	%Ir.m.c.20
IN_AUTO	EBOOL	R	Automatic mode active	%Ir.m.c.23
ON_PAUSE	EBOOL	R	Movements sequence suspended	%Ir.m.c.33
IM_PAUSE	EBOOL	R	Movement suspended (immediate PAUSE)	%Ir.m.c.34
ST_IN_STEP	EBOOL	R	Step by step mode in progress	%Ir.m.c.39
CH_ERROR	EBOOL	R	Channel fault	%Ir.m.c.ERR
SPEED	DINT	R	measured speed	%IDr.m.c.2
FOL_ERR	DINT	R	current position deviation	%IDr.m.c.4
SYNC_N_RUN	INT	R	step number in progress	%IWr.m.c.7

Internal Status Objects (Explicit Exchanges) of IODDT Type T_INTERPO_STD

At a Glance

This part presents the internal status object (explicit exchanges) of the IODDT of type `T_INTERPO_STD` which applied to the TSX CAY33 modules. It groups together the word type objects whose bits have a special significance. These objects are presented in details below.

Notes

- Generally speaking the meaning of the bits is given for the 1 state of this bit. In each specific case the state of the bit is explained.
- Not all the bits are used.

Managing Exchanges: EXCH_STS

The table below presents the significances of the channel exchange control bits `EXCH_STS` (%MWr.m.c.0).

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Status parameters (STATUS) exchange 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	Exchanging adjustment parameters	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Module reconfiguration in progress	%MWr.m.c.0.15

Exchanges Report: EXCH_RPT

The table below presents the meanings of the 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 Operating Status: CH_FLT

The table below presents the meanings of the report bits `CH_FLT` (%MWr.m.c.1).

Standard symbol	Type	Access	Meaning	Address
EXT_FLT	BOOL	R	External fault (same as HD_ERR bit)	%MWr.m.c.2.0
MOD_FLT	BOOL	R	Internal error Module absent, inoperative, or in self-test mode	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Hardware or software configuration fault	%MWr.m.c.2.5
COM_FLT	BOOL	R	Communication fault with processor	%MWr.m.c.2.6
APP_FLT	BOOL	R	Application fault (errored configuration) or command fault	%MWr.m.c.2.7
CH_LED_LOW	BOOL	R	Channel LED status	%MWr.m.c.2.8
CH_LED_HIGH	BOOL	R	Channel LED status	%MWr.m.c.2.9

Axis Operating Status: AX_STS

The table below presents the meanings of the report bits `AX_STS` (%MWr.m.c.3).

Standard symbol	Type	Access	Meaning	Address
Hardware faults: HD_ERR (%Ir.m.c.4) (groups the faults below)				
ANA_FLT	BOOL	R	Analog output short-circuit fault	%MWr.m.c.3.0
AUX_FLT	BOOL	R	Auxiliary output short-circuit fault	%MWr.m.c.3.1
DRV_FLT	BOOL	R	Speed drive fault	%MWr.m.c.3.2
ENC_SUP	BOOL	R	Encoder supply fault	%MWr.m.c.3.3
ENC_BRK	BOOL	R	Encoder break fault	%MWr.m.c.3.4
EMG_STP	BOOL	R	Emergency stop fault	%MWr.m.c.3.5
AUX_SUP	BOOL	R	24 V supply fault	%MWr.m.c.3.0
ENC_FLT	BOOL	R	Absolute encoder parity series or E bit fault	%MWr.m.c.3.7
Application faults: AX_ERR (%Ir.m.c.5)(groups the faults below)				
SLMAX	BOOL	R	Maximum soft stop overshoot	%MWr.m.c.3.8
SLMIN	BOOL	R	Minimum soft stop overshoot	%MWr.m.c.3.9
SPD_FLT	BOOL	R	Overspeed fault	%MWr.m.c.3.10
FE1_FLT	BOOL	R	MAX_F1 position deviation fault	%MWr.m.c.3.11
REC_FLT	BOOL	R	Recalibration fault	%MWr.m.c.3.12
TW_FLT	BOOL	R	Debugging window fault	%MWr.m.c.3.13
STP_FLT	BOOL	R	Stopping fault	%MWr.m.c.3.14
FE2_FLT	BOOL	R	MAX_F2 deviation fault	%MWr.m.c.3.15

Other Status Data

The table below presents the meanings of other state data.

Standard symbol	Type	Access	Meaning	Address
N_RUN	INT	R	step number in progress	%MWr.m.c.4
G9_COD	INT	R	Movement type in progress	%MWr.m.c.5
G_COD	INT	R	Instruction code in progress	%MWr.m.c.6
CMD_FLT	INT	R	refusal report	%MWr.m.c.7
G_SPACE	DINT	R	list of axes of the XMOVE in progress (0 = X and Y, 1 = X and Z, 2 = Y and Z, 3 = X, Y and Z)	%MDr.m.c.12
T_XPOS	DINT	R	position of target on X axis (reach position)	%MDr.m.c.13
T_YPOS	DINT	R	position of target on Y axis (reach position)	%MDr.m.c.15
T_ZPOS	DINT	R	position of target on Z axis (reach position)	%MDr.m.c.13
T_SPEED	DINT	R	speed to be reached	%MDr.m.c.19

Adjustment Parameters   Objects (Explicit Exchanges) of the IODDT of Type T_INTERPO_STD

Adjustment Parameters

The table below presents the various adjustment parameters.

Standard symbol	Type	Access	Description	Address
SLOPE	INT	R/W	Acceleration rule 0 = rectangle, 1 to 3 = trapezoid, 4 = triangle	%MWr.m.c.23
TACC	INT	R/W	Acceleration / deceleration time: TACCMIN to 10000 ms	%MWr.m.c.24
SPEED_PATH_X	INT	R/W	Speed threshold allowed on X axis	%MWr.m.c.25
SPEED_PATH_Y1	INT	R/W	Speed threshold allowed on Y axis	%MWr.m.c.26
SPEED_PATH_Z	INT	R/W	Speed threshold allowed on Z axis	%MWr.m.c.27

CMD_FLT Code Error List for Interpolation

At a Glance

Reading the CMD_FLT (%MWr.m.c.7) command refusal word is performed by explicit exchange. Non-encrypted messages are also available in the diagnostics dialog box, which can be accessed by the **DIAG** command.

Each CMD_FLT word byte is associated with an error type:

- The most significant byte indicates an error in the configuration and adjustment parameters (XX00).
- The least significant byte indicates that a movement command has been refused (00XX).

For example: CMD_FLT = 0023 (least significant fault indicates that the stack is full)

Word %MWr.m.c.7

Configuration and adjustment parameters	Movement command
Most significant byte	Least significant byte

Errors Associated with Interpolation

These errors are indicated by the least significant %MWr.m.c.7 word byte. Numbers between brackets indicate hexadecimal code value.

Value	Meaning
18 (12)	The command cannot be executed for one of the following reasons: ● another command has not yet finished, ● the channel is no longer in Auto mode, ● there is a stop progressing on the channel, ● the channel relay is open (positioning only).
19 (13)	The G01 command cannot be executed
20 (14)	The G09 command cannot be executed
21 (15)	The G10 command cannot be executed
27 (1B)	The G07 command cannot be executed (positioning only)
29 (1D)	Code G_ is unknown
35 (23)	Stack is full Additional code G_ not stored
96 (60)	Code G_ is not authorized to follow a code G01
97 (61)	Code G01 is not executed without being followed by a movement code
99 (63)	Conditions concerning the execution of interpolated movements are not met on the X axis

Value	Meaning
100 (64)	Conditions concerning the execution of interpolated movements are not met on the Y axis
101 (65)	Conditions concerning the execution of interpolated movements are not met on the Z axis
102 (66)	Z axis is requested to move when this axis is not part of the group of interpolated axes
103 (67)	An axis in the process of being interpolated stops (changes mode, relay is opened, etc.)
104 (68)	Position targets requested for code G_ are outside soft stops
105 (69)	Code G01 has been refused because the next movement cannot be accepted
128 (80)	A change of direction is necessary for G09 / G10
129 (81)	G01 distance is too small
130 (82)	The movement distance which follows G01 is too small
131 (83)	The current speed is too great and/or the distance of G01 is too small to reach Vthreshold
146 (92)	Acceleration law is refused
147 (93)	Speed delta X does not conform
148 (94)	Speed delta Y does not conform
149 (95)	Speed delta Z does not conform

Part V
"Flying Shear" Utility

Aim of this Part

This part introduces the "Flying Shear" utility for the TSX CAY22 module and describes its implementation.

What Is in This Part?

This part contains the following chapters:

Chapter 22
Introduction to the "Flying Shear" Utility

Introduction to the "Flying Shear" Utility

At a Glance

The main objective of this function is to be able to respond to applications consisting of:

- a "conveyor belt" axis, which transports a product,
- a "tool carrier" cart axis, which carries out return travel between a rest point and a "working" point, and which synchronizes position and speed with the belt.

The application therefore requires:

- an axis channel, which acquires the advanced speed and position of a product to be cut
- an axis channel which controls the movement of the cutting tool support and command with a discrete output.

Principle of the "Flying Shear" Utility

The table below introduces the different stages of a cutting cycle.

Step	Action
1	The cart is in waiting position at the point of rest. **Note:**The cutting value is a reading relative to the last cut made. This position is called "source control"
2	The card calculates the source control corresponding to the next cut and checks that the cut is possible.
3	When the product feed is equal to the length to be cut less a synchronization distance (in order to be synchronized at the point of synchronization), the cart accelerates and synchronizes with the controlling axis while aiming for the source control.
4	When the cart is synchronized, it remains attached in position for the whole journey.
5	When crossing the tool descent point, an output commands the cutting action.
6	Upon indication that the cut (on a sensor input or position) is finished, the tool is rechucked.
7	Upon indication that the synchro (on a sensor input or position) is finished, the cart exits slave mode and stops as quickly as possible.
8	Upon indication of a theoretical stop, the cart returns to the idle position and slowly restarts the cycle.

Solution Adapted to the TSX CAY 22 V2.0 Module

Implementation of this utility with a TSX CAY 22 module V2.0 requires the definition of:

- channel 0 as a control axis for the acquisition of the position measurement for the product to be cut (belt, conveyor). This is an infinite axis, which continually advances in the positive direction.
- Channel 1 as an axis for the cutting tool support cart. This is a limited linear axis, whose behavior is guided by the controlling axis feed.

Channel 1 supports:

- the static command output for the cutting tool,
- the end of cutting sensor input (if necessary in the application),
- the end of synchro sensor input (if necessary in the application).

NOTE: If necessary, channel 0 can support the event cutting sensor input.

If the utility is not used, the module is totally compatible with preceding versions.

Chapter 23
Configuring the "Flying Shear" Utility

Aim of this Chapter

This chapter describes the configuration screen for the "Flying Shear" utility (channel 1 of the CAY 22 module V2.0 onwards) along with the associated parameters.

What Is in This Chapter?

This chapter contains the following topics:

How to Access the Configuration Parameters for the "Flying Shear" Utility.

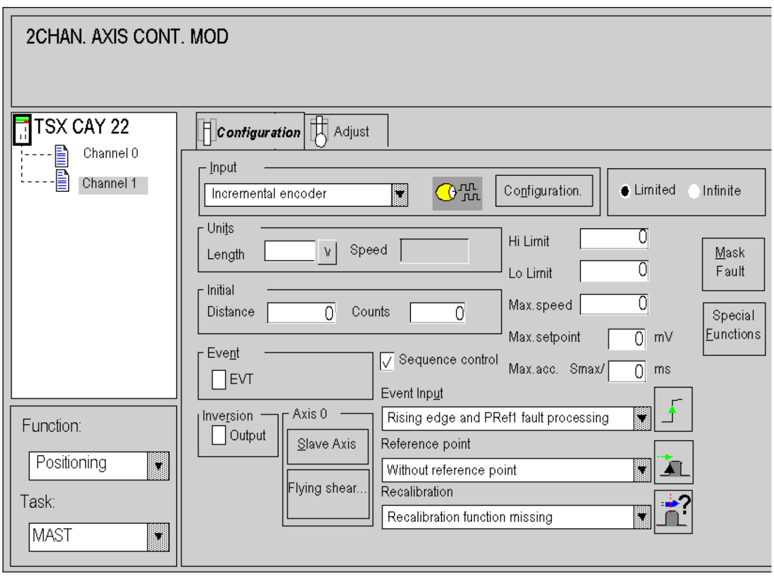
Preliminaries

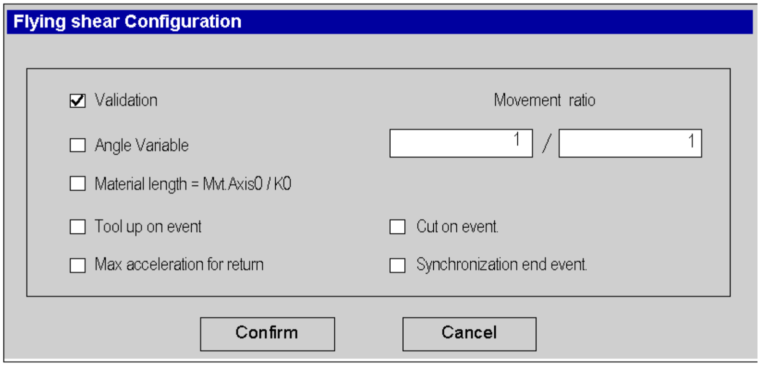
Channel 1 is dedicated to the limited tool carrier axis. Before executing a program consisting of a flying shear instruction, channel 0 should be configured as an infinite machine.

It is advisable to enter the maximum value authorized into the modulo, so that the cutting length of the product is not limited.

Procedure

The table below describes the procedure to follow to access the configuration screen for the "Flying Shear" utility.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>On the application hardware configuration screen, double click on the TSX CAY 22 module.</td></tr>
<tr><td>2</td><td>Choose channel 1.</td></tr>
<tr><td>3</td><td>Select the Positioning function from the Function field.
Result: The following screen appears:

</td></tr>
<tr><td>4</td><td>Select the Limited option from the Machine field.</td></tr>
</table>

Step	Action
5	Click on the **Flying shear** button from the **Axis 0** field. **Result:** The following screen appears: 

The Master/Slave (Slave axis 0) function and the re-calibration function are exclusive to the "Flying Shear" utility.

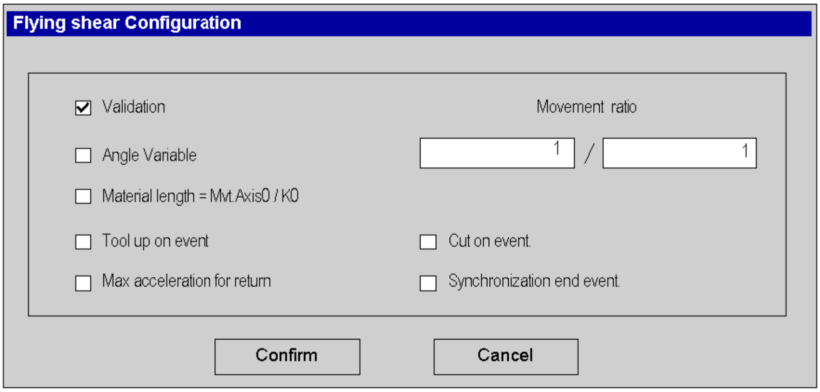
Description of Configuration Parameters

Preliminaries

This subject only deals with specific parameters for the "Flying Shear" utility. To configure an independent axis, see Configuring an Independent Axis

Illustration

The screen below groups together all configuration parameters for the "Flying Shear" utility.



Description of Parameters

The table below describes the different parameters for the "Flying Shear" utility.

Parameter	Description
Validation	If checked, the "Flying Shear" utility is activated
Angle Variable	Indicates that the cutting angle between the belt feed axis and the tool feed axis can vary dynamically. The value of the %MDr.m.0.65 and %MDr.m.0.67 words allows this variation by application. Cart speed = infinite axis speed x K0 x movement ratio using K0 = (%MDr.m.0.65 / %MDr.m.0.67) **Comment:** when this parameter is not checked, only the Movement ratio values are used The cart speed then becomes: Infinite axis speed x Movement ratio
Movement ratio	Indicates the speed movement ratio between the belt and cart. Limits [1/sin80, 1/sin10];1, with 1/sin80 = 1.015426 and 1/sin10 = 5.758770. The numerator and denominator should be whole numbers between 1 and 10.10^6
Material length...	Indicates that the K0 factor is used when calculating the material length to be cut. Material length = Mvt.Axis0 / K0 with K0 = (%MDr.m.0.65 / %MDr.m.0.67). **Comment:** when this parameter is not checked, the length of the product to be cut is equal to that of the belt.
Tool up on event	Indicates that the tool up time is ordered by the EXT_EVT output (%Qr.m.1.10) on channel 1. The tool is rechucked after this time. **Comment:** when this parameter is not checked, it indicates that the tool up time is triggered when a position crosses. The tool is rechucked after this time.
Cut on event.	Indicates that the length to be cut can be determined by an EVENT input on channel 0. The cut reading is determined by the belt position at the event rather than the cutting distance on the event.
Max acceleration for return	Indicates that the module should use the parameter Max acceleration for configuration when stopping after cutting.
Synchronization end event.	Indicates that the synchronization end time is ordered by the RECAL output on channel 1. Effective desynchronization is performed after this time. Suspension occurs between the re-calibration and "Flying Shear" utilities. **Comment:** when this parameter is not checked, it indicates that the end synchronization time is triggered by a position. Effective desynchronization will be performed after this time.

All these parameters are deactivated by default. All combinations are possible.

Chapter 24
Programming the "Flying Shear" Utility

Aim of this Chapter

This chapter describes the programming principle of the "Flying Shear" utility.

What Is in This Chapter?

This chapter contains the following topics:

Programming "Flying Shear" Utility principle

Preliminaries

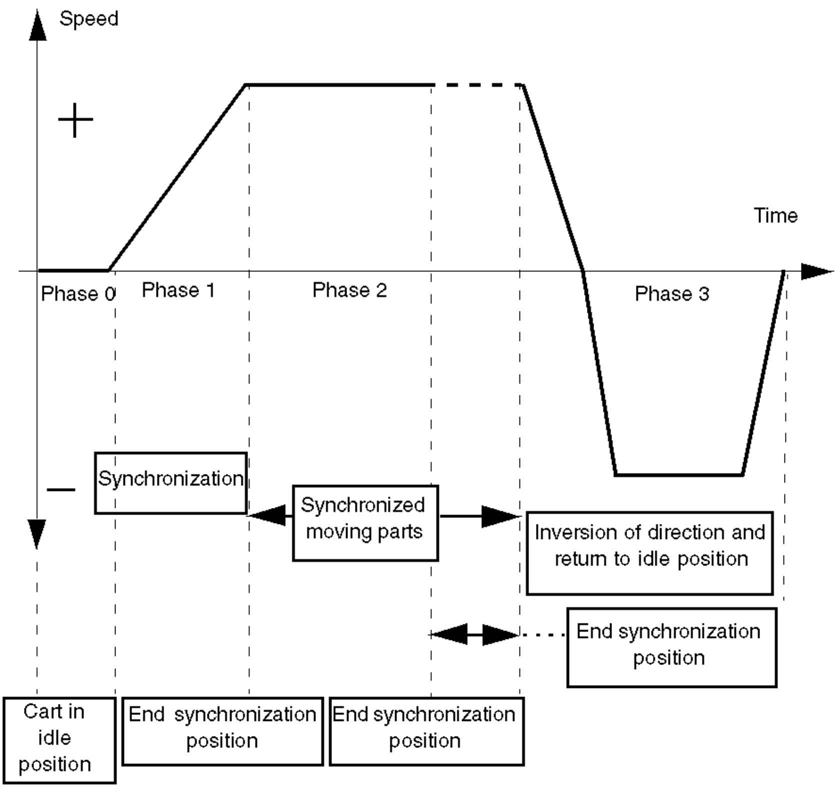
All G codes already present on the TSX CAY 22 V1.x are recognized and applied so that they are defined on channel 1 when the "Flying Shear" utility is activated.

In particular, they allow homing and initial positioning at the point of synchronization before beginning a cutting cycle.

The MANU, DRV_OFF, DIRDRIVE modes are not affected or modified when the "Flying Shear" utility is activated on channel 1.

Principle

Cutting cycle macro



The maximum acceleration of synchronization phase 1 is that of the adjustment screen.

Programming the "Flying Shear" Utility: SMOVE function

Preliminaries

This topic contains a description of the SMOVE function parameters with code 22.

Detailed Description

The SMOVE function should be used to program a "Flying Shear" movement with the following syntax: SMOVE %Chr.m.1(N,Gp,22,Dist,Vit,M)

Parameter	Description
%Chr.m.1	Channel 1 address
N	Movement Number
Gp	Preparatory code. Possible values are: ● 90 : Normal cut ● 98 : Cut on event
22	Specific instruction code for flying shear
Dist	Cutting length
Vit	Speed for return to idle point
M	Initialization of cut counter

Where:
- (Dist/K0) < modulo/2
- (Dist/K0) * K0 * Movement ratio > (PointSynchro - PointRepos)
- Vit <= Vmax
- M = 0 or 16#0100. If M = 16#0100, the counter is then initialized at 0 then incremented if the cut is performed
- The movement programmed can only be executed under the following conditions:

Axis conditions

Axis 0	Axis 1
Channel 0 should be configured as an infinite machine. It is advisable to enter the maximum value authorized into the modulo, so that the cutting length of the product is not limited.	The channel 1 "Flying Shear" utility must be configured
Channel 0 should not have a blocking fault	The operating mode should be set to automatic
It is essential that the numerator product for the Movement ratio with the K0 numerator be less than 2^{30}	Axis 1 should be referenced
It is essential that the denominator product for the Movement ratio with the K0 denominator be less than 2^{30}	The drive should be Enable
The K0 ratio should be between 0.01 and 5	The cart should have come to idle. The preceding movement should be a G09 or G22 and should have the idle position as a position setpoint. The distance between the * K0 * Movement ratio synchronization idle point and end point should be more than modulo/2 on channel 0
-	Code G98: ● (Cutting distance on evt/K0) < modulo/2 ● (Cutting distance on evt/K0) * K0 * Movement ratio > (Synchronization point position - Idle position)

Only the SMOVE with G22 instruction saves the control source point to memory. All other instructions or mode changes erase this memory.

If there is no control source point in the memory upon the next SMOVE with G22 instruction, the new control source point is equal to the conveyor belt position at the point at which the instruction was received plus the length of the Dist cut passed to parameters.

Dynamic Condition for Refusing Command

A command is refused if:

When starting the tool, the module states:

Either the belt speed is too great in relation to the Vmax cart speed,

or the synchronization distance is more than the distance between the synchronization point and the rest point.

Warning on SMOVE with G22 "Value exceeded"

A warning is generated if the module states, when receiving the SMOVE with G22 instruction:

- Either that the belt is too close to the control source point to cut, in which case, the module will calculate a new control source, which will allow cutting with the following formula:
  New control source point = n * Dist + last control source point, where n is the smallest whole number which allows the cutting cycle to be implemented
- Or that the control source point has been exceeded, in which case, the FAIL_CMD_AUTO_COTE_DEPASSE 0x0042 Warning is given to the PLC at the beginning of the execution of G22.

Long Cut on Event G98

A sensor detects the course of the product to be cut. When the event is received, the module memorizes the belt position and calculates the cutting point according to the following formula:

Control source point = Memorized position + Cutting distance on event.

The Dist parameter should have a value more than the cutting length on event so that the event triggers the tool. This forms a security system, if the event does not intervene during the final cutting distance.

Be careful not to disturb the event input on channel 0 when cutting on event.

Immediate Cut

To perform an immediate cut, the module should be in waiting phase 0 for starting a cart with the SMOVE with G22 instruction and receive a CoupeIm(%Qr.m.1.20) rising edge command. The module will calculate the minimum cutting distance for a shorter piece than the one currently being carried out. The cutting position is saved as the control source for the next cut.

If the cart begins to move, the command for an immediate cut is ignored.

Operating Mode

After carrying out each cutting instruction, the cut counter (%Iwr.m.1.7) is increased.

While executing a G22, it is possible to prepare and send the module the consecutive cutting order at the end of the current cut, which can be identical or different to the current cut.

To halt the cutting operations using the cart at the rest point, it is possible to:

- No longer send G22 and allow the current cuts (or those queuing in the module) to be completed
- Send a Pause order to allow the current cut to be completed without moving on to the orders queuing in the module
- Send a STOP order when the G22 is waiting (%Ir.m.1.44)

The Feed hold command (via the CMV command reset) is ineffective.

The CMV alteration is ignored during the SMOVE with G22 command. Only the current CMV value present from the start of the command is taken into account.

A blocking fault on channel 1 stops the cart by exiting the synchronization status axis where it is.

NOTE: Alterations to your own parameters for the flying shear are only adapted when the cart is at the rest point.

Comment: Detailed editing of the SMOVE with G22 instruction is not currently possible.

Tool Output Operating Mode

In manual mode, this is the strict copy of the AUX_OUT (%Qr.m.1.11) output bit.

In automatic mode, it is set by the M parameter for ALL instructions with a G code.

During a G22 code instruction, the status depends upon:
- The current position of the cart
- The parameters: **Tool down position, Tool up position, Tool up time.**
- The InhibTool (%Qr.m.1.21) command.

Tool descent always takes place on the **Tool down position**, without the use of a time delay.

The tool rechucking takes place on a position, after the **Tool up time**. The position is set either by the **Tool up position** on the adjustment screen, or by the tool rechucking event if the **Tool up on event.** box is checked.

If the **Tool up on event.** box is checked, and the expected tool deactivation event does not occur, the **Tool up time** will be triggered on the way to the **Tool up position**.

Upon ordering STOP (%Qr.m.1.15) the tool rechucks immediately, without any time delay. The axis begins to stop after the **Tool up time**. If this time delay is zero, a value of 4ms is used.

During a SMOVE with G22 command, when the InhibTool (%Qr.m.1.21) command is at 1, the auxiliary output is forced to 0. Therefore, the tool remains chucked, regardless of the cart's position. If the tool was low, it is immediately rechucked, without using any time delay.

If the **Tool up on event.** box is checked, the EXT_EVT (%Qr.m.1.10) command allows the tool to be rechucked before waiting for the **Tool up position**.

Chapter 25
Adjusting the "Flying Shear" Utility

Aim of this Chapter

This chapter describes the adjustment screen for the "Flying Shear" utility (channel 1 of the TSX CAY 22 module V2.0 onwards) along with the associated parameters.

What Is in This Chapter?

This chapter contains the following topics:

How to Access the Adjustment Screen for the "Flying Shear" Utility

At a Glance

This screen allows access to and modification of the adjustment parameters for the "Flying Shear" utility.

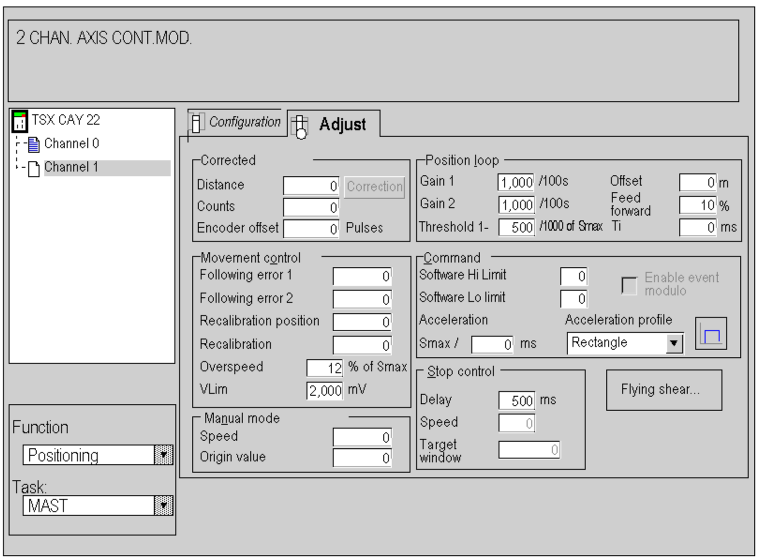
It can be accessed in local or offline mode.

To access the adjustment screen for the "Flying Shear" service, the channel must first be configured to the "Flying Shear" service. The adjustment screen is used to select the channel to be adjusted, and enables access to the parameters.

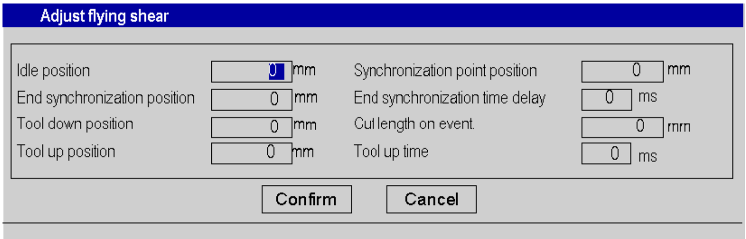
Procedure

The table below describes the procedure to follow to access the adjustment screen for the "Flying Shear" utility.

Step	Action
1	From the TSX CAY 22 module debugging or configuration screen, select the **Adjust** mode. **Result:** The following screen appears:



<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Click on the Flying Shear button.
Result: The following screen appears:



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

Description of Adjustment Parameters

Description of Screen Parameters

Description of the parameters:

Field	Description
Idle position	This parameter indicates the tool idle position in relation to the source point. The tool is set in this position following a cut. Limits [S1_min, S1_max]
Synchronization point position	This parameter indicates the synchronization start position for the tool axis with the infinite axis. Limits [Idle position, S1_max]
End synchronization position	This parameter indicates the end synchronization position. When the tool crosses this point, the end synchronization time is triggered. Limits [Synchronization position, S1_max]
End synchronization time delay	This parameter indicates that the end synchronization time delay and sets the synchronization period after crossing the end synchronization position or activating the synchronization end input (RECAL). Limits [0,10000] units ms
Tool down position	This parameter indicates the tool down position, which takes place in the direction of travel. Limits [Idle position, S1_max]
Cut length on event.	This parameter indicates the length relative to the captured value at the moment of event processing when cutting on event. Limits [0, S1_max]
Tool up position	This parameter indicates the tool up time activation position, which takes place in the direction of travel. Limits [Down position, S1_max]
Tool up time	This parameter indicates the delay time for rechucking the tool and sets the period for the lowered tool after crossing the tool up position. Comment: If STOP is ordered on the tool carrier and the tool is activated, this time is discounted before the axis stops. Limits [0,10000] units ms

Adjustment of K0 parameter

In the configuration screen, if **Angle Variable** has been validated, the K0 parameter can be modified by the application. This parameter shares channel 0 and does not appear in any entry screen. Initialization and modification are performed by using the %MDr.m.0.65, %MDr.m.0.67 variables and the WRITE_PARAM %Chr.m.0.65 and WRITE_PARAM %Chr.m.0.67 instructions. At the beginning of each cutting cycle, the module stringently checks that the numerator product for the Movement ratio with the K0 numerator and then the denominator product for the Movement ratio with the K0 denominator are both less than 2^{30}. The module also checks that the K0 ratio is between 0.01 and 5.

If this is not the case, an error code is returned.

Adjustment of DMAX2 Parameter

If the Flying Shear utility is activated and in automatic mode, the DMAX2 parameter becomes the controlling parameter for the following error between the cart and the source control point. This monitoring is only active when the synchronizing phase is ignored by the other phases.

Adjustment of the Acceleration profile Parameter

During a G22 instruction, the synchronizing acceleration phase 1 is always realized using a rectangular acceleration profile, which is independent of the acceleration profile selected on the Adjustment screen.

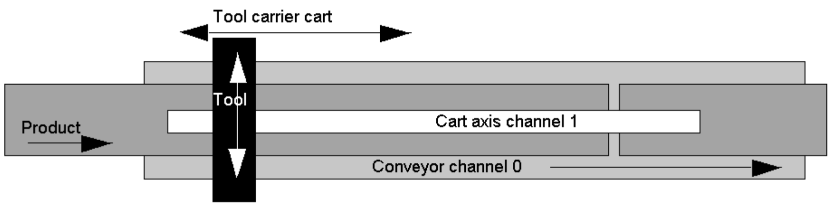
Adjusting the Ti gain

When the Flying Shear utility is activated, the Ti gain adjusted in the Adjustment screen is active at idle and also during the phase 2 of the moving part synchronization.

Applications to Various Cut Types

Parallel Axes

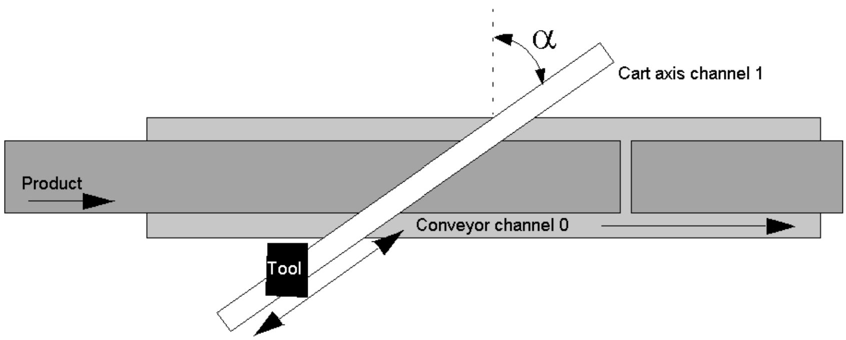
The product feed axis is parallel to the cart feed axis. The cutting tool moves perpendicularly to the product feed axis either by rotating or on a linear axis.



In this case, **Angle Variable** and **Material length = Mvt.Axis0/K0** are not checked, so **Movement ratio** = 1/1.

Axes with Angles

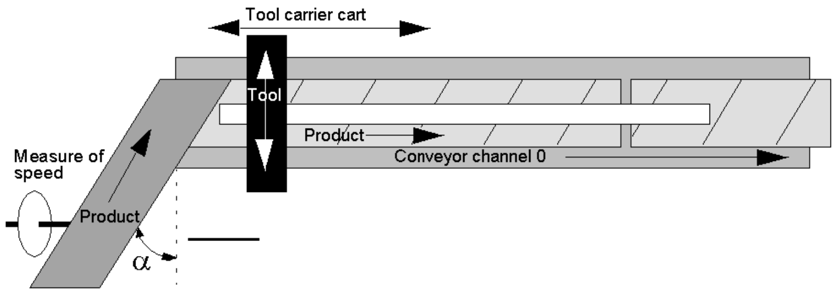
The product feed axis and the cart feed axis form an angle. The cutting tool only moves up or down. The movement of the tool carrier cart should cover the whole of the surface of the product to be cut.



In this case, **Angle Variable** and **Material length = Mvt.Axis0/K0** are not checked, so **Movement ratio** = 1/ $\sin\alpha$.

Insertion of Product with Angles

The product feed axis and the cart feed axis are parallel but the insertion of the product forms an angle to the belt. The cutting tool moves perpendicularly to the product feed axis either by rotating or on a linear axis. The feed speed is picked up when the product is inserted and varies from the belt speed.



In this case, **Angle Variable** and **Material length = Mvt.Axis0/K0** are checked, so **Movement ratio** = 1/1, K0 = $\sin\alpha$.

Variable Length Cut

The Dist parameter from the SMOVE with G22 instruction allows consecutive cuts of varying length.

Long Cut on Event

A sensor detects the course of the product to be cut. When the event is received, the module memorizes the position and calculates the cutting point according to the following formula: Control source = Memorized position + Cut length on event.

In this case, **Cut on event.** is checked.

Memorandum for the "Flying Shear" Utility

"Flying Shear" Utility Data

Adjust	Objects	Description
Kmul Axis0	%MDr.m.0.65	K0 numerator
Kdiv Axis0	%MDr.m.0.67	K0 denominator
SyncTime	%MWr.m.1.60	Synchronization support time
ToolUpTime	%MWr.m.1.61	Active tool support time
RE_POS	%MDr.m.1.43	Idle position
Slave_Off	%MDr.m.1.55	Synchronization point position
Re_WDW	%MDr.m.1.51	End synchronization position
ToolDownPos	%MDr.m.1.63	Tool down position
ToolUpPos	%MDr.m.1.65	Tool up position
DistEvt	%MDr.m.1.67	Cut length on event

Tool	Objects	Description
Inhibtool	%Qr.m.1.21	Force auxiliary output during G22
AUX0_OUT	%Qr.m.1.11	Output command in manual mode

Miscellaneous	Objects	Description
EXT_EVT	%Qr.m.1.10	Tool up command
Coupelm	%Qr.m.1.20	Immediate cut command
Waiting_Master	%Ir.m.1.44	Waiting status during control source feed to point
Synchronizing	%Ir.m.1.45	Movement status towards the synchronization point
IN_SLAVE	%Ir.m.1.36	Cart feed status synchronizing
Ret_Neutral	%Ir.m.1.43	Return status towards the idle point
ToolsOn	%Ir.m.1.42	Cut status current
SYNC_N_RUN	%IWr.m.1.7	Cut counter
AX_EVT Axis0	%Ir.m.0.15	Event input status (for cut on event)

Comment: If the pre-symbolization service is used, double use objects are viewed with their old meaning and the new objects are not pre-symbolized.

Limitations of Use of "Flying Shear" Utility

When using the "Flying Shear" utility, channel 1 draws on the position, current speed and captured position information from channel 0. However, it does not manage the channel 0 operating mode. It is therefore the responsibility of the user not to disturb the operation of a flying shear application with the following operations, which can cause disturbances:

- Channel 1 reconfiguration
- Channel 0 adjustment
- Mode change on channel 0
- Disturbance of position measurement on channel 0.

Axis 0 faults do not disturb SMOVE with G22. It is the responsibility of the application to manage the operating modes of axis 1 according to axis 0 when necessary.

The length of the maximum cut is defined by the infinite modulo channel 0 divided by 2. If the tool rechucking is to be performed when EXT_EVT is switching to 1 (tool deactivation on validated event), the module does not manage the TimeOut or associated security.

List of "Cutting On the Fly" Utility Error Codes

Error	Value	Description
FAIL_CMD_AUTO_COND_EXEC_G22_AXE0_K0	0x002B	"G22 auto command error, axis0 initial conditions" The G22 command cannot be executed following problems connected to axis 0.
FAIL_CMD_AUTO_COND_EXEC_G22_AXE1_K0	0x002C	"G22 auto command error , axis1 initial conditions" The G22 command cannot be executed following problems connected to axis 1.
FAIL_CMD_AUTO_COND_PARAM_G22_K0	0x002D	"G22 auto command error, non-conforming parameters" The G22 code parameters are inconsistent with the adjustment parameters.
FAIL_CMD_AUTO_VIT_PILOTE_TROP_GRANDE	0x0041	"G22 auto command error , axis0 control speed too fast"
FAIL_CMD_AUTO_COTE_DEPASSE	0x0042	"G22 auto command error, value exceeded"
FAIL_CMD_AUTO_SYNCHRO_TROP_COURTE	0x0043	"G22 auto command error, synchro distance too short"

Glossary

A

Absolute encoder

This type of encoder directly delivers the numerical value of the axis position. The position measurement is maintained in the event of a power outage.

Axis

The set of external elements that control the machine's movements (speed reducer, encoder, etc.).

The motor/drive/mechanics that command the moving part to move in a given direction (axis, linear movement) or around a set rotation axis (rotoid axis, circular movement).

C

Control Source Point

In a flying shear application, a sensor detects the course of the product to be cut. When the event is received, a module memorizes the belt position and calculates the cutting point according to the following formula :

Control Source Point = memorized point + cutting distance on event.

D

Direction discrimination

Micro-programmed system, which determines the movement operating direction.

E

Emergency stop

Movement stop with maximum deceleration.

Event

Modules triggered by software or hardware (application specific module).

Events take priority in Mast or Fast tasks, and are executed upon detection.

The EVT0 event has the highest priority, the others share the same level of priority.

F

Feed forward gain (KV)

Co-efficient allowing adjustment of the action and feed forward of the position control loop (compromise between following/overshoot error at the stop point).

Flying Shear

Application consisting of :

- a ‘conveyor belt’ axis, which transports a product.

- a ‘tool carrier’ cart axis, which carries out return travel beetween an idle point and a working point, and which synchronizes position and speed with the belt.

Following error

Deviation between the position setpoint and the measurement during movement.

Forced reference point movement

Write parameters procedure for measuring current position at a predefined value. This operation references the axis.

G

Gray Code

binary code known as reflected, in which the transition from term n to term n+1 is performed by modifying one single digit. Code reading is therefore made with no ambiguity.

I

Incremental encoder

pulse generator with 2 signals offset to 90º. These are produced according to the axis movement and are counted by the module.

Independent multi-axis

Movement law is applied to each axis independently. Axes depart simultaneously, movement speed is a setpoint speed, movement time depends on the distance to be covered, the "axes" do not arrive at the same time; movement through space is ordinary. The aim is to arrive at the arrival address as quickly as possible without constraining the trajectory.

Infinite machines

The moving part moves continually between the value 0 and the modulo limit (for example a conveyor belt).

Interpolation

Allows 2 or 3 axes to be linked, so that movements can be made two or three dimensionally.

ISO
International Standard Organization. The ISO code is the most used code. The transmission rules, formats and symbols are ISO standards. AFNOR is a member of ISO.

L

Limited machine
The moving part moves between two upper and lower limits in two directions.

M

Machine source
Dimensioning reference for the machine axis.

Mechanical cam
Mechanical shoulder fixed to an axis, which activates a sensor indicating the end of the moving part's journey.

Modulo
Domain of evolution for measuring an infinite axis.

Movement law
This is the variation law applying to the acceleration, speed and position setpoints. It is often illustrated by the curve: speed = F(time).

In an increasingly complex order, the following can be found: rectangular, triangular, trapezoidal, parabolic and squared sine laws.

P

Parameterable indexed position (REFP)
Index value for the indexed position calculation, absolute position = index (REFP) + indexed position.

R

Reference point
Write parameters procedure for measuring current position by moving the moving part and detecting external events (reference point input and/or cam input). This operation references the axis without moving.

Referenced axis
Module status when a reference point is taken. Position measurements are only meaningful and movements are only authorized in this status.

Resolution

this is the smallest type of input information which provides information that can be measured from the output information.

Rotary pulse

Pulse supplied by a rotary incremental encoder, detected at each complete turn of the axis.

S

Servo-system

Automation function which consists of producing a physical dimension conforming to a fixed or variable reference (position control, speed control, etc.)

Software hi limit

Upper limit for the position measurement, which must not be exceeded by the moving part (set by the SLMAX adjustment parameter).

Software lo limit

Lower limit for the position measurement, which must not be exceeded by the moving part (set by the SLMIN adjustment parameter).

Speed modulation coefficient.

(CMV) Multiplication co-efficient of a value between 0 and 2 for all speeds, in increments of 1/1000.

Speed setpoint

Theoretical speed of the moving part calculated by the module following the maximum acceleration law and the programmed speed.

T

Target window

Position control tolerance around the stop point.

Trajectory

Following elementary movements between a departure address and an arrival address, passing through intermediary addresses. Movement between two addresses is performed using a specific movement's speed or time.

V

Valid space of measurements

Total measurement points between 2 software limits.

Index

A

B

C

D

E

F

J

L

M

O

P

R

S

T

W

X