Modicon TSX CTY 4A

Counting Module

Limited Availability
Used and in Excellent Condition

Open Web Page

https://www.artisantg.com/79829-1

ARTISAN TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

- Critical and expedited services
- In stock / Ready-to-ship
- We buy your excess, underutilized, and idle equipment
- Full-service, independent repair center

PL7 Junior/Pro
Premium PLC Application-Specific Functions
Counting

eng March 2005

35009577_00

Clipsal
Merlin Gerin
Square D
TAC
Telemecanique

Schneider Electric
Building a New Electric World

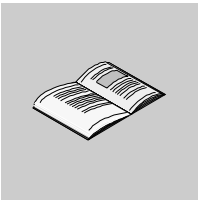
Document Set

At a glance

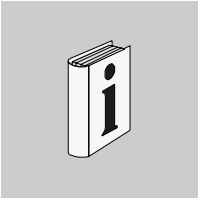
This manual consists of 8 volumes:

- Volume 1
  - Shared task functions
  - Discrete task
  - AS-i implementation
  - Dialog operator task
- Volume 2
  - Upcounting Task
- Volume 3
  - Axes command task
- Volume 4
  - Step by step axes Command task
- Volume 5
  - Electronic Cam task
- Volume 6
  - SERCOS Task
- Volume 7
  - Analog Task
  - PID Control Task
  - Weighing Task
- Volume 8
  - Regulation Task

Table of Contents



About the Book



At a Glance

Document Scope

This manual deals with implementation of tasks (except for communication tasks) on the Premium/Atrium via the PL7 software.

Validity Note

The update of this publication takes into account the standard functions of the PL7 V4.5; nevertheless, it supports implementation of previous PL7 versions.

Related Documents

Title of Documentation	Reference Number
Hardware implementation manual	TSX DM 57 xxE

User Comments

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

Introduction to the application-specific Counting function

1

General introduction to the counting function

At a Glance

The counting function is used to carry out fast counting using data modules, PL7 screens and specialized language objects. General operation of the data modules is described in the *Introduction to operation of data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C, p. 13* section and their functions are described in detail in the *Description of the standard functions of the TSX CTY •• counting modules, p. 31* section.

The physical context in which the counting will be carried out must be defined (rack, power supply, processor, modules or devices,...) during installation, then the software must be installed.
The latter is carried out by various PL7 editors:

- either offline,
- or on-line.

Set up principle

The table below shows the different installation phases for the application specific Counting function.

Mode	Phase	Description
Offline	Module declaration	Choice: ● of geographical position: number and slot where rack-based modules are concerned, ● module type.
		Entering the configuration parameters.
	Confirming configuration parameters (See: PL7 Junior, Pro; Application-specific functions volume 1)	Confirmation at module level.
	Global application confirmation (See: PL7 Junior, Pro; Application-specific functions volume 1)	Confirmation at application level.
Offline or online	Symbolization (See: PL7 Junior, Pro; Application-specific functions volume 1)	Symbolizing the variables associated with the application specific function.
		Programming the functions that the specific function must carry out using: word and bit objects associated with the module and event processing.
Online	**Transfer**	Transferring the application to the PLC.
	and	Debugging the application using: ● debugging help screens to control inputs and outputs, to modify the thresholds and set point values, to trigger events and carry out initializations, ● diagnostic screens are used to identify faults.
Offline or online	**Documentation**	Printing different information relating to the application.

Note: the order defined above is given as an indication. The PL7 software can use editors interactively in any order you wish (however you cannot use the data or program editors without configuring the counting module first).

Introduction to operation of data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C

2

At a Glance

Subject of this chapter

This chapter describes the operating principles of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C data modules.

What's in this Chapter?

This chapter contains the following sections:

2.1 General on the TSX CTY 2A/4A and TSX CTY 2C modules

Introduction to the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules

At a glance

The TSX CTY 2A, 4A and 2C modules are counting modules made for the whole range of Premium modular PLCs. As such, they support all the operations of the PL7 software.
These modules have available:

- functions for offloading tasks directly linked to the counting (comparisons, captures, presetting or resetting to zero, error detection, etc.) from the processor,
- functions for generating events for the application program,
- configurable discrete outputs in reflex outputs, adapted to rapid actions.

These modules constitute a range to the varied characteristics, adapted to the different situations met in the industrial monitoring.

Main characteristics

The main characteristics are as follows:

Type	Functions	Number of channels per module	Number of physical outputs per channel	Maximum frequency (kHz)
TSX CTY 2A	Upcounting, downcounting, upcounting/downcounting.	2	1 or 2 (according to the function)	40
TSX CTY 4A	Upcounting, downcounting, upcounting/downcounting.	4	1 or 2 (according to the function)	40
TSX CTY 2C	Upcounting/ downcounting, speed measurement/monitoring.	2	4	1000

The TSX CTY 2A and 4A modules are functionally identical. Only their number of channels is different.

2.2 Principle functions of the TSX CTY 2A and TSX CTY 4A modules

At a glance

The subject of this section

This section describes the principle operations of the TSX CTY 2A and TSX CTY 4A modules.

What's in this Section?

This section contains the following topics:

Introduction to the TSX CTY 2A and 4A modules

Description

The TSX CTY 2A and 4A modules support upcounting, downcounting, or upcounting/downcounting pulses. They have the following functions available:

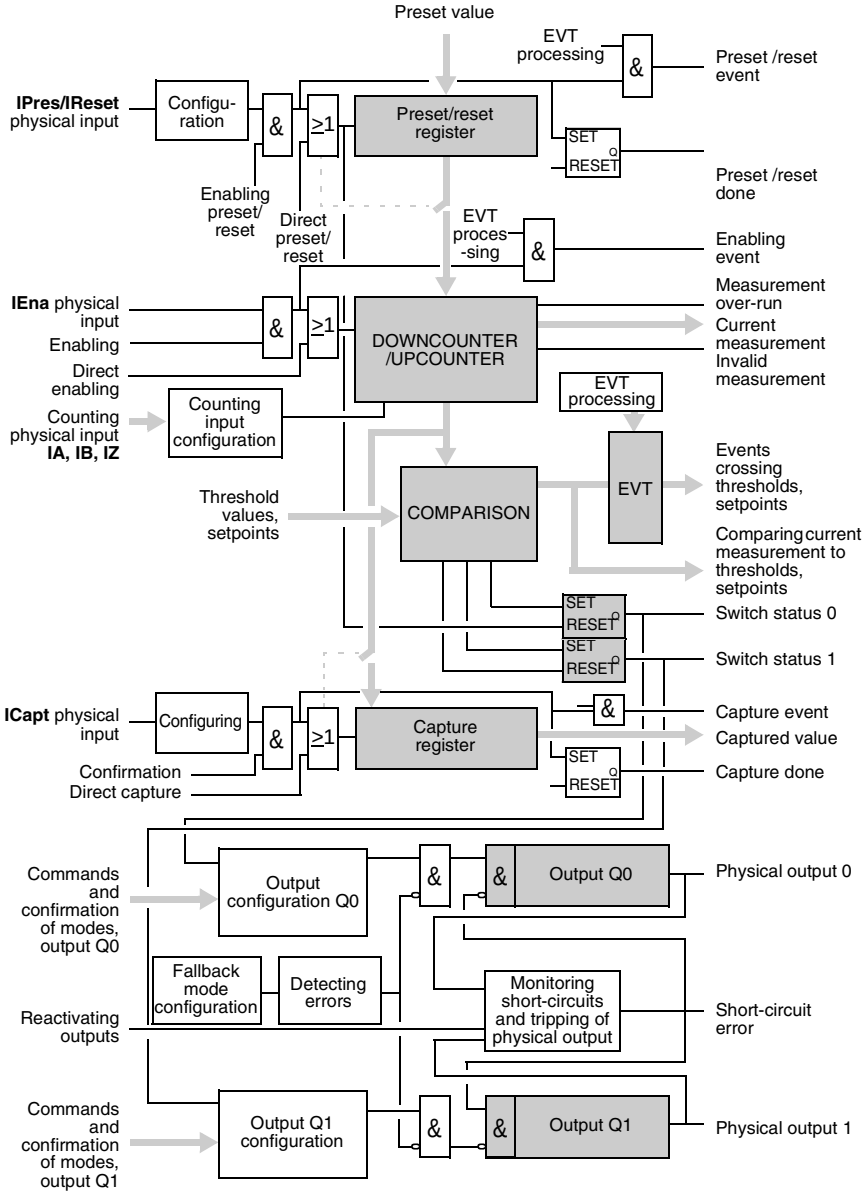
- confirmation (See *Introduction to an up counting or down counting channel, p. 18*),
- capture (See *Description of the capture function in counting modules, p. 41*) (in upcounting/downcounting),
- preset or reset (See *Description of the preset and reset function in counting modules, p. 45*),
- comparisons (See *Description of the comparison function for counting modules, p. 59*),
- storing transient events using two counter outputs (See *Description of switches associated with counting modules, p. 65*),
- two physical outputs (See *Description of physical outputs associated with counting modules, p. 75*),
- event processing (See *Event processing installation, p. 152*).

These modules are **the same**, except for the number of channels:

- TSX CTY 2A: 2 channels,
- TSX CTY 4A: 4 channels.

Illustration

The diagram below introduces the global structure of a channel. According to the selected function (upcounting, downcounting or upcounting/downcounting), some operations cannot be active.



Introduction to an up counting or down counting channel

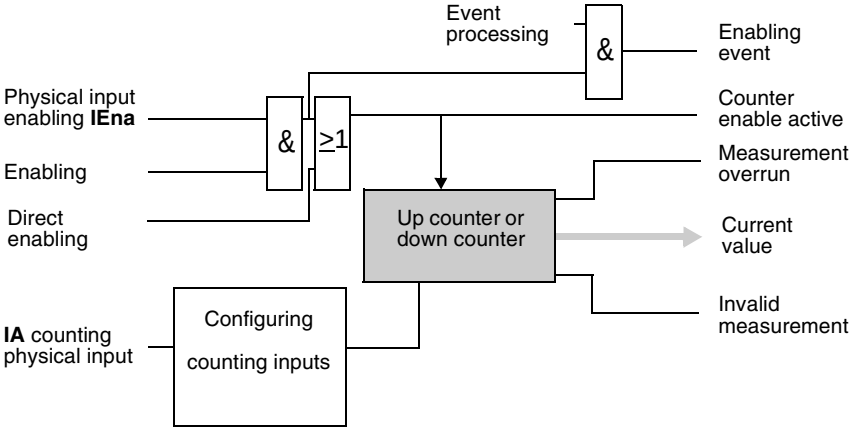
At a glance

This section introduces the functional kernel of a channel, namely the following blocks:

- up counting or down counting,
- confirmation.

Illustration

The up counting or down counting functional kernel of the TSX CTY 2A or 4A modules is shown below.



Note

Confirmation (hardware or software) is a special auxiliary function, linked intrinsically to the up counting or down counting. In fact, without confirmation, there can be no up counting or down counting. Because of this, contrary to other auxiliary functions, confirmation is introduced in this section. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.

Physical input

The up counting or down counting functions alone admit only one physical counting input, as shown above (the signal is on the **IA** input).

Input characteristics

This table summarizes the input characteristics of the up counting or down counting functions, for each channel.

Modules involved	**TSX CTY 2A/4A**
Main physical input	**IA**
Confirmation of up counting or down counting	• hardware: **IVal** input, conditioned by the **Confirm** software command, • direct, via the software (**Direct confirmation** command).

Operating in up counting or down counting

At a glance

The standard operation of the TSX CTY 2A and 4A modules in up counting or down counting only is summarized below.

Standard principles

Up counting and down counting are developments that happen in the same register of the module, only the direction of the development is different. This is set by the software configuration (up counting or down counting functions alone).
The counting register's developments are only possible when the function is **confirmed**:

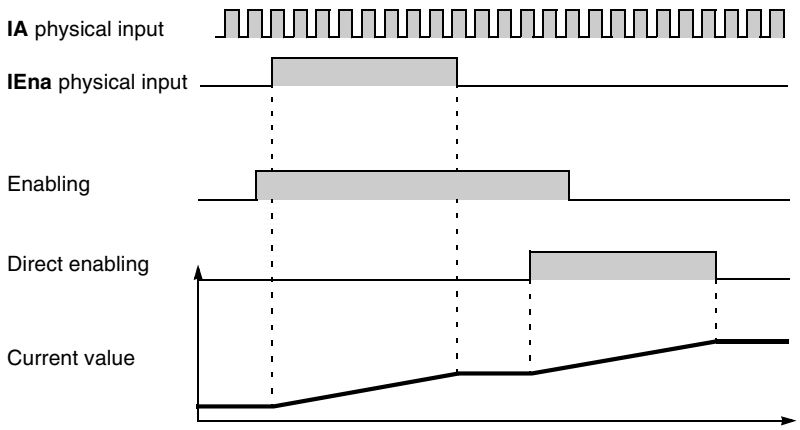
- either via a physical input (**IVal**) with a software prepositioning (**Confirm** command),
- or directly, via the software: **Direct confirmation** command.

The counting values develop between:

- 0 and +16 777 215 in up counting (24 unsigned bits),
- -16 777 216 and +16 777 215 in down counting (24 + signed bits).

Operation

This timing diagram illustrates the up counting process. Down counting is similar, only the measurement's current development direction is the opposite. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.



Introduction to an up counting/down counting channel (TSX CTY 2A/4A)

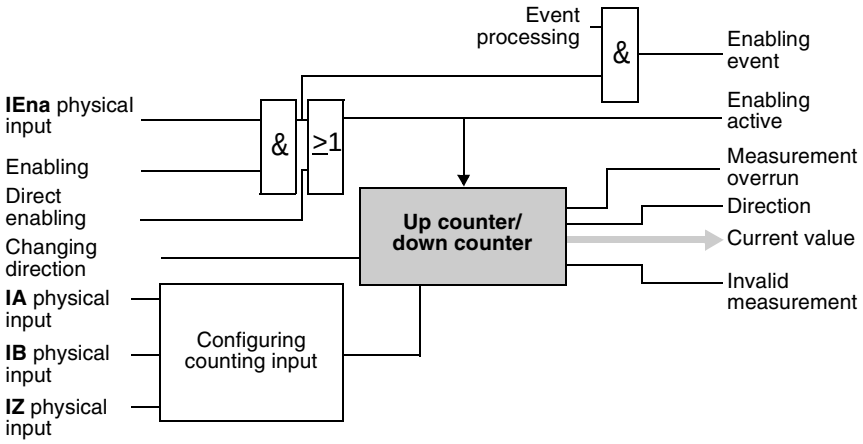
At a glance

This section introduces the functional kernel of a channel, namely the following blocks:

- up counting/down counting,
- confirmation.

Illustration

The up counting/down counting functional kernel of the TSX CTY 2A or 4A modules is shown below.



Note: this illustration is very similar to the one showing the up counting or down counting function individually. The main differences are to do with the physical inputs. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.

Important note

Confirmation (hardware or software) is a special function, linked intrinsically to the main up counting or down counting function. In fact, without confirmation, there can be no up counting or down counting. Because of this, contrary to other functions, confirmation is introduced in this section.

Physical inputs

The up counting/down counting function takes four configurations of counting physical inputs, as described in the table below.

Input characteristics

This table summarizes the input characteristics of the up counting or down counting functions (for each channel).

Modules involved	**TSX CTY 2A/4A**
Input configuration	• An **IA** physical input and a software input (counting direction). • Two physical inputs: **IA** (main) and **IB** (counting direction). • Two physical inputs: **IA** (up counting) and **IB** (down counting). • Three physical inputs: **IA** and **IB** to the incremental encoder, and **IZ** when there is a pulse to the encoder tower.
Confirmation	• hardware: **IVal** physical input, • software.

Operating in up counting/down counting

At a glance

Operating the TSX CTY 2A and 4A in combined up counting/down counting is introduced below.

Standard principles

Up counting or down counting are developments that happen in the same register of the module, only the direction of the development is different. This is set via the software configuration, a physical input or applying the counting signal to a particular input.
The counting register's developments are only possible when the function is **confirmed**:

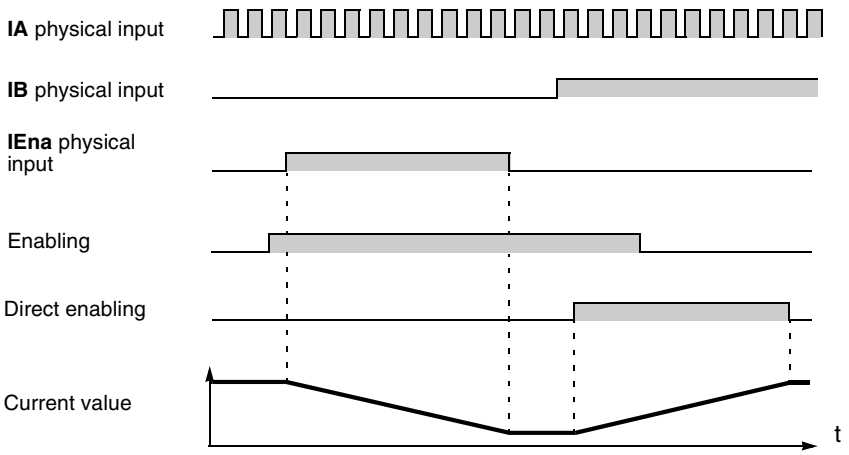
- either via a physical input **IVal** with software prepositioning (**confirm** command),
- or directly, via the software: **direct confirmation** command.

The counting values develop between:

- -16 777 216 and +16 777 215 (24 + signed bits).

Operation

The timing diagram below shows the up counting/down counting process when the counting signal is applied to the **IA** input, with the **IB** input defining the counting direction. As shown earlier, there are several other ways of defining the counting direction, but the principle of up counting/down counting remains the same. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.



2.3 Main functions of the TSX CTY 2C module

At a Glance

Contents of this section

This section describes the main functions of the TSX CTY 2C fast counting module.

What's in this Section?

This section contains the following topics:

Introduction to the TSX CTY 2C module

Description

The TSX CTY 2C is a counting module for the TSX/PMX/PCX 57 PLCs.
This module has two identical channels with the following main functions:

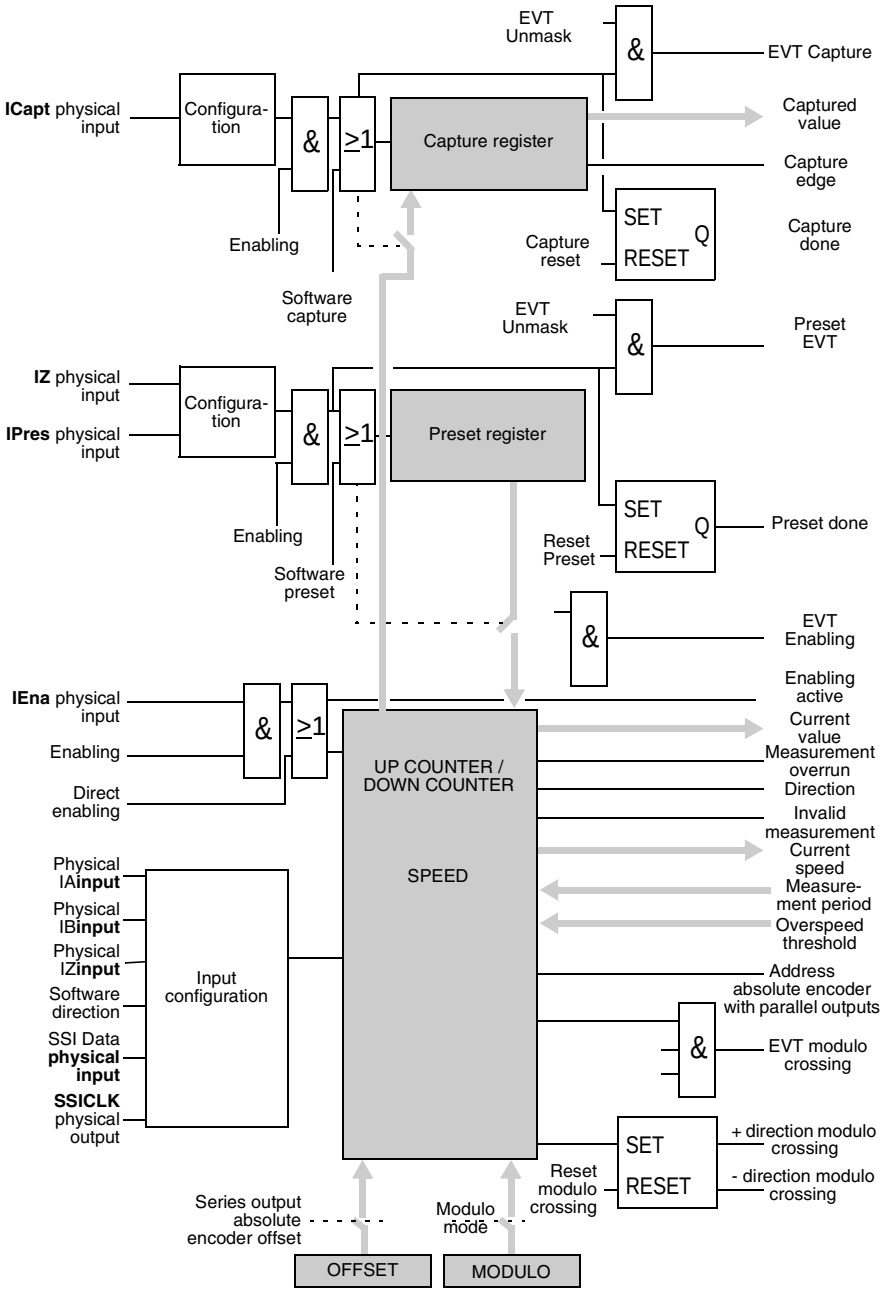
- up counting/down counting pulses (See *Introduction to an up counting/down counting and measurement channel , p. 26*),
- placing an absolute encoder (See *Specialized interface for an absolute encoder (TSX CTY 2C), p. 40*),
- measuring and monitoring speed (pulse frequency) (See *Speed Monitoring Function (TSX CTY 2C), p. 89*).

It also has the following functions:

- confirmation (See *Introduction to an up counting/down counting and measurement channel , p. 26*),
- capture (See *Description of the capture function in counting modules, p. 41*),
- preset (See *Preset in up/down counting (TSX CTY 2A/4A/2C), p. 53*),
- comparisons (See *Comparison in up/down counting and measurement mode (TSX CTY 2C), p. 64*),
- storing (See *Switches in up/down counting mode (TSX CTY 2A, 4A, 2C), p. 71*) transient events using two switching circuits,
- four physical outputs (See *Introduction to TSX CTY 2C module outputs, p. 81*),
- event processing (See *Event processing installation, p. 152*).

Illustration

The diagram below introduces a channel's functional kernel structure (main up counting/down counting functions and speed, confirmation, preset and capture measurement). To find out the associated language objects, see the default objects (See *Details of implicit exchange objects, p. 161*).



Introduction to an up counting/down counting and measurement channel

Introduction

This section introduces the functional kernel of a channel in the TSX CTY 2C module, namely the following blocks:

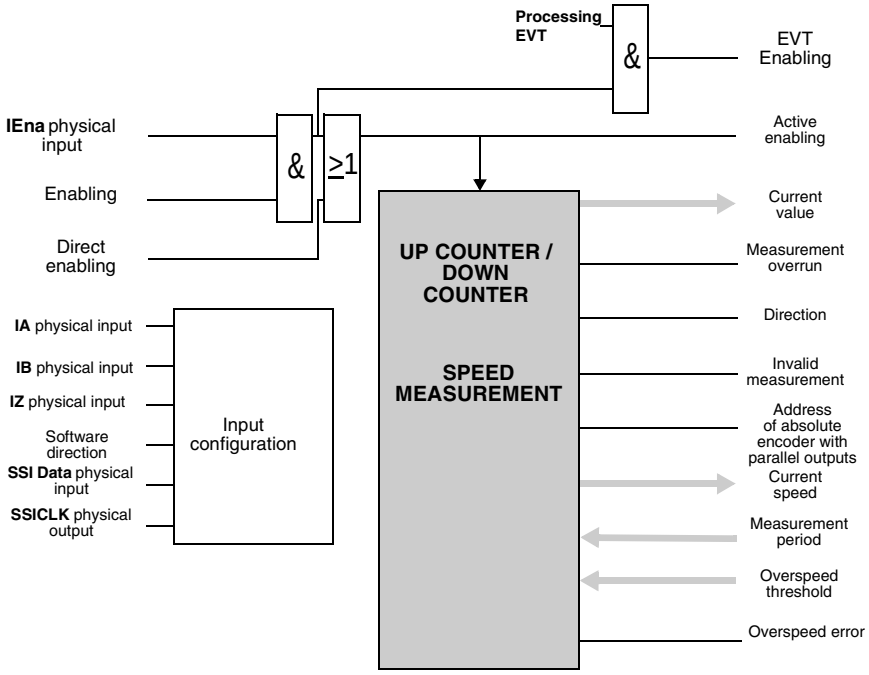
- up counting/down counting and speed measurement,
- confirmation.

Note

Confirmation (hardware or software) is a special function, linked intrinsically to the up counting/down counting. It is introduced here because of this, as opposed to other functions.

Illustration

The diagram below introduces the main functions of a channel in the TSX CTY 2C module. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.



Input characteristics

The TSX CTY 2C takes five physical input configurations, as described in the table below (for each channel).

Module involved	TSX CTY 2C

Input configuration	• An **IA** physical input and a software input (counting direction).
	• Two physical inputs: **IA** (main) and **IB** (counting direction).
	• Two physical inputs: **IA** (up counting) and **IB** (down counting).
	• Three physical inputs: **IA** and **IB** to the incremental encoder, and **IZ** when there is a pulse to the encoder tower.
	• An absolute SSI series encoder input/output interface, with:
	  ◦ an **SSI Data** physical input,
	  ◦ an **SSICLK** transmission timer output.

Confirmation of up counting/ down counting	• hardware: **IVal** input (combined with the **Q2** output, to be configured), conditioned by the **Confirm** software command, • direct, via the software (**Direct confirmation** command).

Operating in up counting/down counting

At a glance

Operating the TSX CTY 2C module in up counting/down counting mode is introduced below.

Standard principles

Up counting or down counting are developments that happen in the same register of the module, only the direction of the development is different. This is set via the software configuration, a physical input or applying the counting signal to a particular input.
The counting register's developments are only possible when the function is **confirmed**:

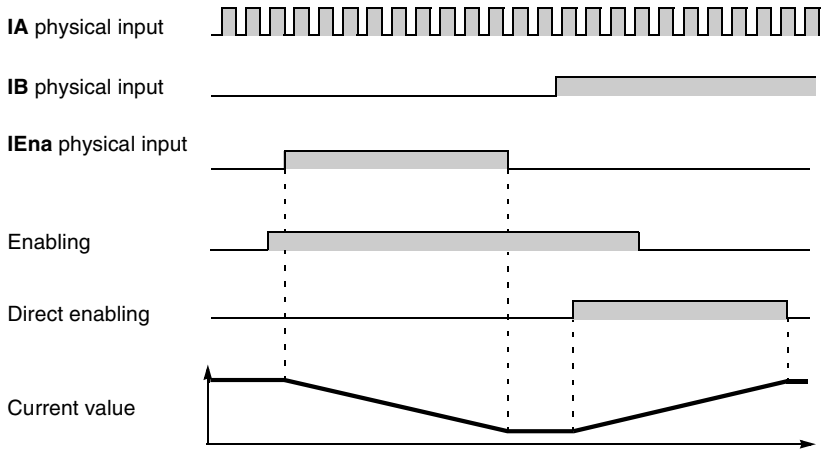
- either via a physical input (**IVal**) with a software prepositioning (**Confirm** command),
- or directly, via the software: **Direct confirmation** command.

The counting values develop between:

- -16 777 216 and +16 777 215 in normal mode (24 + signed bits),
- 0 and +33 554 431 in modulo mode (25 unsigned bits).

Operation

The timing diagram below shows the up counting/down counting process when the counting signal is applied to the **IA** input, with the **IB** input defining the counting direction. As shown earlier, there are several other ways of defining the counting direction, but the principle of up counting/down counting remains the same. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) part.



Operating in speed measurement mode

At a glance

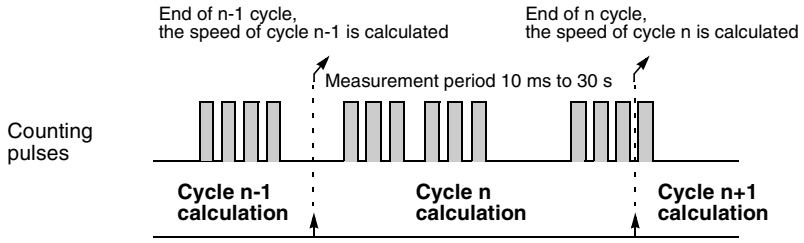
Apart from the up counting/down counting function, the TSX CTY 2C module also has the speed measurement function.
This function can be used with counting sensors (generating pulses) or with absolute encoders (generating a word describing a position).

Note

Getting a position is not an up counting or down counting of pulses as such, but it has the same result, that is to say successive writes of the counting register.

Principle of the measurement

The principle of calculating the speed is the same as that of the frequency meter: with a measurement period (sampling) that can be adjusted by the user, the speed is calculated and updated, in number of pulses per second. By default, the measurement period's value is 1 second.
The following diagram shows the principle of the measurement:



The sampling period must be chosen according to the required precision and the low limit of the speed (frequency) of the signal to be measured. This is done using the following formula:

$$\text{Sampling period} \geq \frac{1}{\text{precision x speed}}$$

where the precision is shown in decimal value (for example: 0.1% = 0.001) and the speed (pulse frequency) in thousands of pulses per second (kHz). The sampling period is acquired in milliseconds.
Example: for a pulse frequency of 40 000 to 250 000 pulses per second and a precision established at 0.1%, the minimum sampling period is 25 ms.

Note: the minimum sampling period is 10 ms.

Principle of monitoring the speed

Monitoring the speed supports setting the reflex outputs to 0 when there is an overshoot of a configurable speed threshold, and directly commanding a security device (for example).
For more details on the speed measurement function, see *Description of the speed measuring function for the TSX CTY 2C module, p. 89*.

Description of the standard functions of the TSX CTY •• counting modules

3

At a glance

Subject of this chapter

This chapter comprehensively describes all the standard functions of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C counting modules.

What's in this Chapter?

This chapter contains the following sections:

3.1 Introduction to functions associated with counting modules

Introduction to input configurations and functions associated with counting

At a Glance

TSX CTY 2A, 4A and 2C modules offer several possibilities for processing unprocessed results in offline mode, according to the type of module, the type of sensor and the mode (counting function) selected. This chapter describes input configurations and the associated functions which are available.

Input configurations

Modules TSX CTY 2A, 4A and 2C have three counting inputs (See *Description of counting input interfaces (TSX CTY 2A / 4A / 2C), p. 36*) for each channel: **IA**, **IB** and **IZ**. These inputs cannot be changed round.
The TSX CTY 2C module also has an SSI series interface (See *Specialized interface for an absolute encoder (TSX CTY 2C), p. 40*) for each channel.

Advantages of associated functions

Associated functions are used to adapt modules to the application. As long as the main counting function for one module is similar enough to another, selecting the latter may depend on these other functions.

Common functions

The table below summarizes the main characteristics of the functions common to the three module types, and their availability according to the operating mode.

Function	Description	Availability
Invalid measurement	Detecting a loss of pulse caused by defective operating conditions, or when capacity has been exceeded.	All modes.
Preset or reset	Pre-setting a counting register to a defined value (zero for a reset).	Resetting to zero in counting, presetting in down counting or up/down counting combined.
Capture	Storing an immediate up/down counting register value.	Only in combined up/down counting.
Comparison	Comparing a **current** value with a preset value (zero in down counting only). Comparing the **captured** value with a preset value.	All modes. Only in up/down counting.
Switches	Storing a preset temporary event.	All modes. Number of switches (one or two) varies according to the module and the mode.
Outputs	According to configuration: physical outputs linked to switches (reflexive outputs), or positioned by software (discrete "manual" outputs).	Number of outputs (one to four) variable according to the module and the mode.
Event processing	Several events can trigger a processing operation and a reflex action: • enabling up or down counting, • presetting or resetting, • crossing the threshold or setpoint, • capture (when up/down counting).	All modes

Specific function

Speed monitoring (See *Description of the speed measuring function for the TSX CTY 2C module, p. 89*) is only available with the TSX CTY 2C module. One or two physical outputs can be linked to the comparators, and can then translate the result of the comparison between the immediate speed and a preset value.

Special functions (TSX CTY 2C)

The TSX CTY 2C module can also configure three special functions (See *Description of the special functions of the TSX CTY 2C counting module , p. 91*). These functions are derived from the previous standard functions, and they respond to the specific requirements of certain counting applications. These are:

- Special Function Number 1: time elapsed since the last pulse and a capture.
- Special Function Number 2: triggering a capture and a direct (software) preset synchronized with the programmable frequency output.

- Special Function Number 3: monitoring the **correct speed** and **stationary moving part**. Tolerance values for the **correct** speed and **stop** speed can be configured.

3.2 Description of counting module input interfaces

At a Glance

Subject of this section

This section deals with the input interfaces for counting modules.

What's in this Section?

This section contains the following topics:

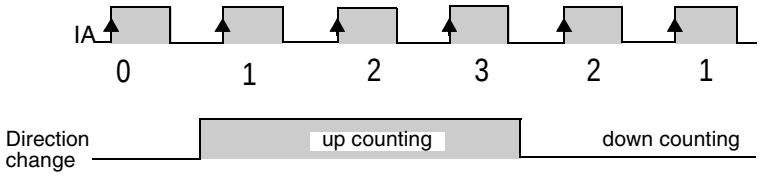
Description of counting input interfaces (TSX CTY 2A / 4A / 2C)

At a Glance

This section describes the operation of counting input interfaces which are common to modules TSX CTY 2A, 4A and 2C.

Configuration: IA up/down counts, direction according to application

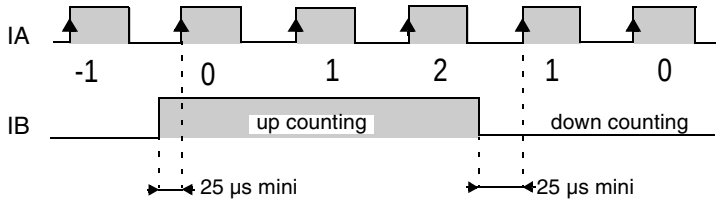
The figure below represents the counting signal applied to **IA** input.



In this configuration, the **change of direction** bit managed by the application determines direction of the up or down counting on the rising edges of the signal.

Configuration: IA up/down counts, direction according to IB

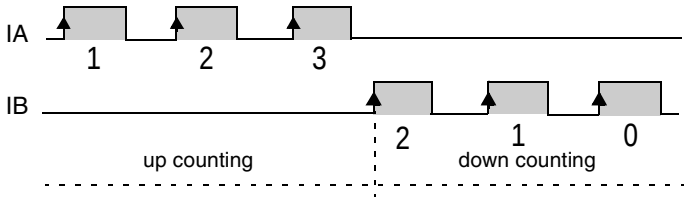
The following figure represents the counting signal applied to **IA** and **IB** inputs which determine the direction of the counting.



Up/down counting is done on the rising edges of pulses received on **IA** input.

Configuration: IA up counts, IB down counts

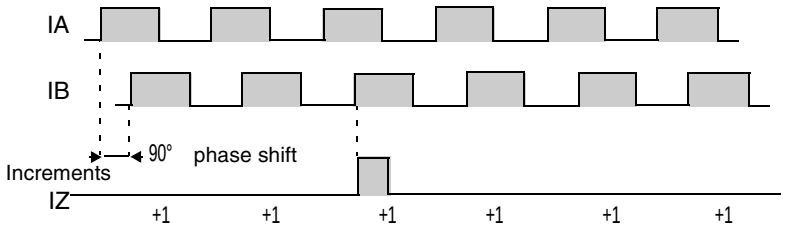
The figure below represents the counting signal applied to **IA** input (in up counting) or to **IB** input (in down counting).



Pulses are taken into account by the up/down counter on the rising edges of the physical input in progress. **IA** input increments the up/down counter (upcounting) and **IB** input decrements it (downcounting). If pulses on the two inputs are simultaneous, the up/down counter does not change.

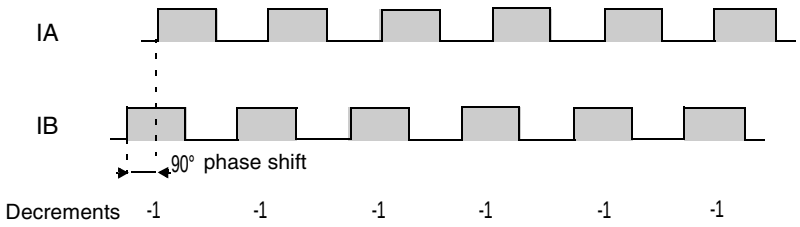
Configuration: incremental encoder

In this operating mode, physical inputs **IA** and **IB** are connected to an incremental encoder which supplies two out-of-phase signals of 90°. The extent to which inputs **IA** and **IB** are out of phase determines the direction of up/down counting.
The figure below represents **counting** (**IA** ahead of **IB**).



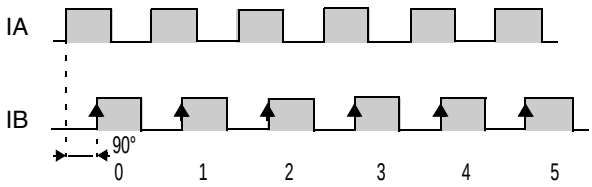
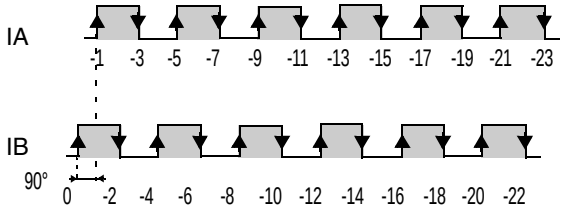
Note: the incremental encoder also provides marker information on the **IZ** input. This pulse on **IZ** input enables an up/down counter to be preset.

The figure below represents **down counting** (**IA** behind **IB**).



Incremental encoder interface options

Several options are available on the configuration screen when an incremental encoder is connected:

If…	**Then…**
the **line check** is configured (encoder with RS 422 / 485 output),	the PLC indicates a fault when it detects a break in the encoder cable on one of the **IA, IB** or **IZ** physical inputs. It is then possible to start the application procedure which corresponds to the fault.
multiplication by 1 is configured,	up/down counting is done on rising edges of **IB** physical input (case shown: counting). 
multiplication by 4 is configured,	up/down counting is done on all the rising and falling edges of **IA** and **IB** physical inputs (case shown: down counting). 

Specialized interface for an absolute encoder (TSX CTY 2C)

At a Glance

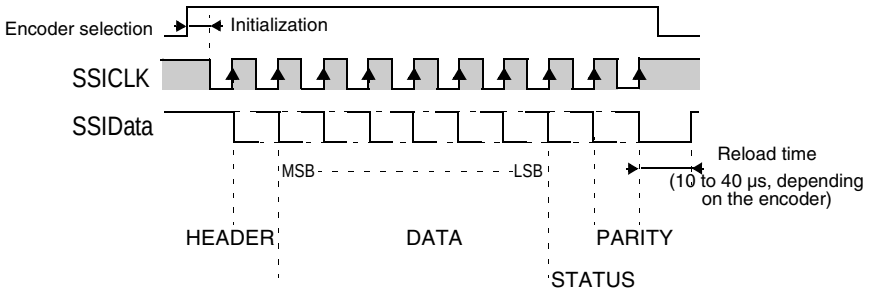
Module TSX CTY 2C also has a specialized interface for a series absolute encoder, for applications that measure and monitor speed.
This section describes the operation of this specific input interface.

Inputs

In this configuration, **SSI Data** physical input and **SSICLK** physical output are connected to the series output absolute encoder.
It is also possible to connect up one to four parallel output absolute encoders by using the adaptation bases (see the Installation Manual).

Description of the SSI interface

The figure below represents an SSI frame.



The main other frame and interface features are as follows:

Parameters	Values or observations
Code	Binary or Gray
SSICLK transmission speed	150 kHz, 200 kHz, 375 kHz, 500 kHz, 750 kHz or 1 MHz
Header bits	Ignored
Data bits	• 8 active data bits minimum. • 17 masked most significant bits maximum (rollover counting). • 17 masked least significant bits maximum (resolution reduction).
Status bits	An error bit specific to the encoder. Its frame position and significance can be configured.
Parity	Even, odd (not monitored by the module) or without parity.

Note

With an absolute encoder, the up/down counting is carried out implicitly in **rollover** mode. The number of unmasked bits directly gives the rollover value. The counting register changes in the [0, rollover] interval. The minimum rollover value is 1 and its maximum value is +33 554 432 (25 data bits without masked bit).

3.3 Description of the capture function in counting modules

At a Glance

Subject of this section

This section describes the **capture** function for counting modules.

What's in this Section?

This section contains the following topics:

Capture function for data module counters

Description

Capture is used to copy the current value of the up/down counting register to a capture register. It therefore fixes the immediate value at the precise moment the operation started.
Capture is only possible in combined up/down counting mode (all modules).
Module TSX CTY 2C also has a combined captured mode in the hardware preset (see: *Capture, TSX CTY 2C module specific features, p. 44*.
Capture done information is an event which can undergo an event processing operation.

Triggering a capture

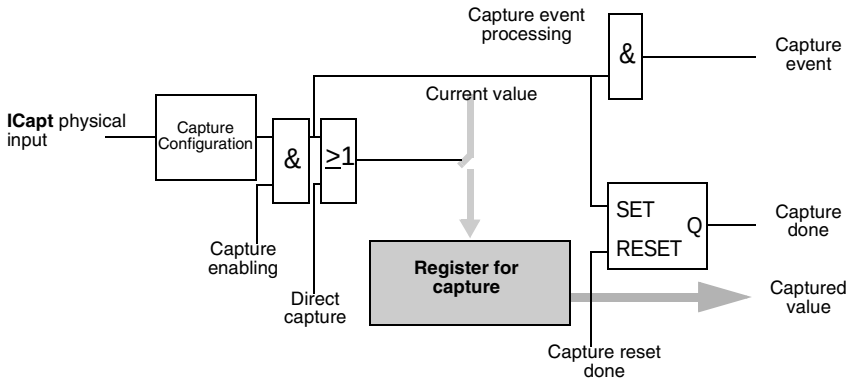
The operation is triggered:

- either directly, via the software: **Direct capture** command,
- or by the hardware: when the **ICapt** physical input status has changed, with pre-enabled software (**Capture enable** command). This signal can be:
  - the rising edge of **ICapt** input,
  - the falling edge of **ICapt** input,
  - the rising edges **and** the falling edges of **ICapt** input (TSX CTY 2C only).

Note: time performances are at a maximum when **ICapt** input is configured on the rising or falling edge.

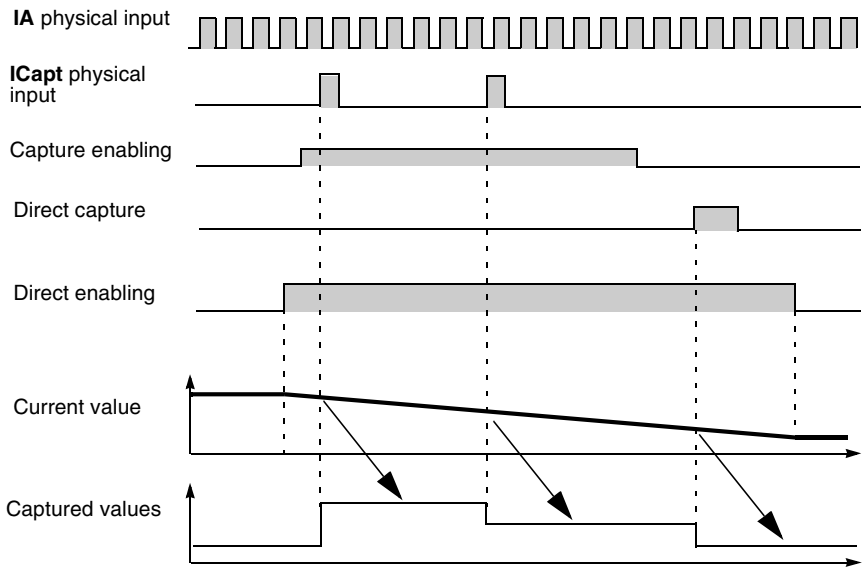
Function hardware structure

The figure below shows the hardware structure of the capture function. The associated language objects are described in the implicit objects (See *Details of implicit exchange objects, p. 161*) section.



Operation

The time diagram below shows the capture mode on the rising edge of **ICapt**. The other modes (capture on falling edge, on rising **and** falling edges) are similar.



Capture, TSX CTY 2C module specific features

Specific modes

Except for the **single** capture modes, described in the previous paragraph, module TSX CTY 2C has two specific modes:

- capture of rising edges **and** falling edges of the input **ICapt**. This is an extension to the **single** capture principle, described in the previous Section,
- capture combined at hardware preset (**capture before preset** mode).

Capture mode on rising and falling edges

The capture mode on rising edges **and** falling edges of the **ICapt** physical input can be used, for example, to measure the lengths of parts. This depends on an exterior pulse for two successive captures to be carried out.
The allowable interval between the two capture fronts must be **a minimum of 0.5 ms**. This is therefore the minimum size of the capture pulse.
Respecting this condition guarantees the correct running of the function, i.e. taking into account of all the edges present on the physical capture input.

Examples

The table below gives as an example this interval according to counting input frequency.

Counting input frequency	Minimum interval between capture edges (in number of counting pulses)
125 kHz	63
250 kHz	125
500 kHz	250
1 MHz	500

Capture before preset mode

This capture mode, specific to module TSX CTY 2C, applies to the up/down counting of pulses (proximity detector, incremental encoder), but not to the acquisition of data from the absolute encoder
The preset **IPres** physical input triggers successively:

- a capture,
- followed by the preset.

3.4 Description of the preset and reset function in counting modules

At a Glance

Subject of this section

This section describes the **preset** and **reset** functions for counting modules.

What's in this Section?

This section contains the following topics:

Introduction to the preset/reset function

General

The preset initializes the up/down counting register to a predefined value (by software). Reset initializes to zero the same register.
The preset concerns the down counting mode and the combined up/down counting mode. Resetting only involves the counting mode (TSX CTY2A/4A).
The operation can be **triggered** or **automatic**.
The following sections detail the conditions for preset/resetting to zero, according to the counting method and module used.

Notes

- Preset (or reset) acts on the object (see *How to deal with an invalid measurement, p. 97*).
- Preset (or reset) done is an event which can undergo an event processing operation.

Preset in downcounting mode (TSX CTY 2A/4A)

At a Glance

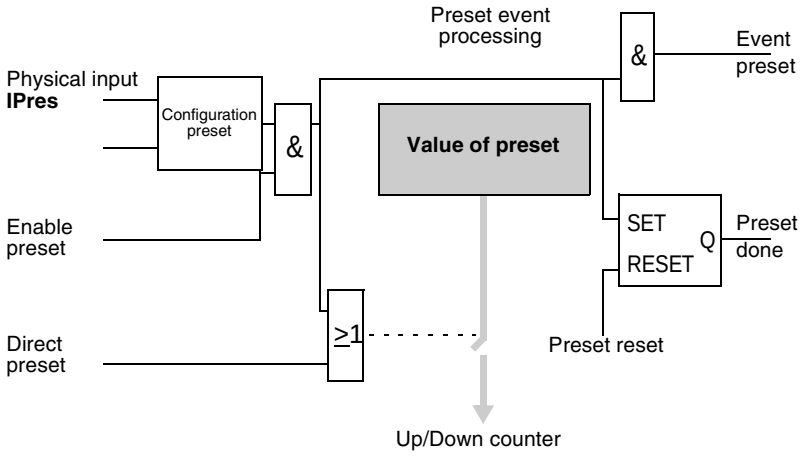
Presetting initializes the down counting register to a value preset by the software. Preset is:

- **triggered**
  - by the hardware: on the rising or falling edge of the **IPres** physical input with pre-enabled software, **Preset enable** command,
  - directly by the software: **Direct preset** command.
- **automatic**, it will be done when the zero value is crossed.

Note: the two types of preset (triggered and automatic) are independent and can co-exist.

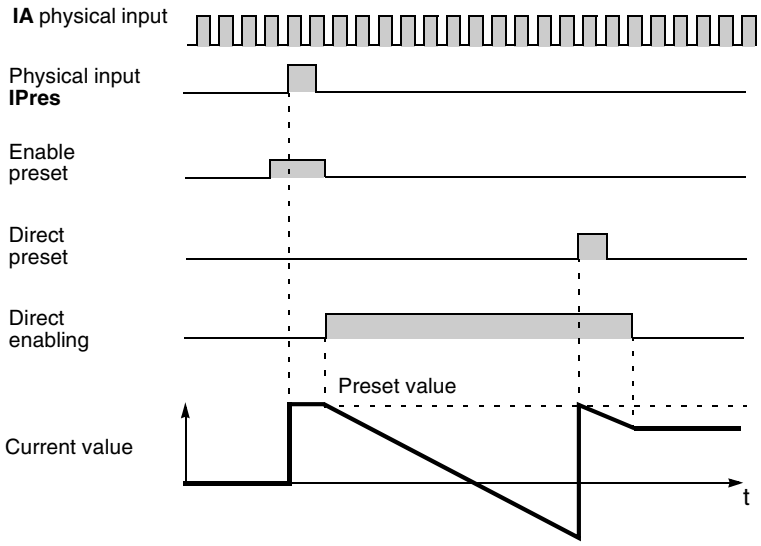
Illustration

The figure below represents the preset function triggered by the hardware. The associated language objects are described in the section on implicit objects (See *Details of implicit exchange objects, p. 161*).

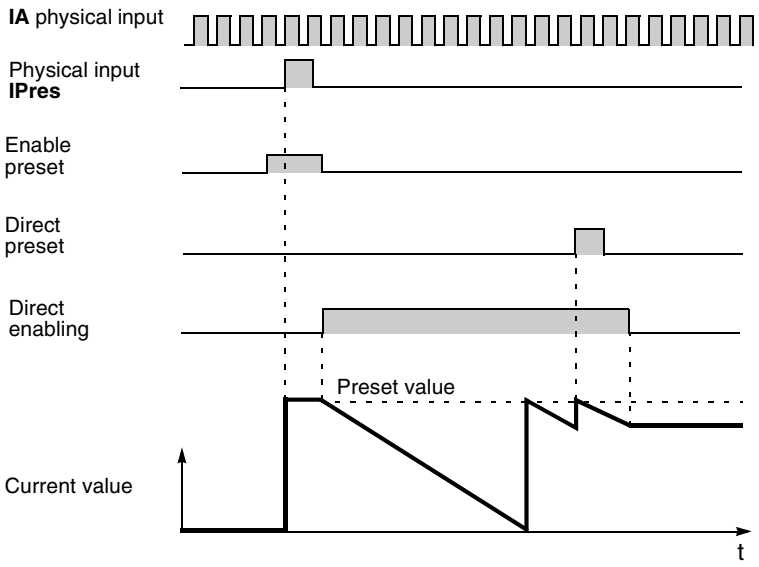


Time diagrams

The following time diagrams show the different cases for preset with or without automatic reloading when crossing the zero value threshold.
The time diagram below shows a preset triggered by **IPres**, then a direct (software) preset. Zero value is crossed without automatic preset.



The time diagram below shows a preset triggered by **IPres**, a direct preset, and an automatic preset when crossing the zero value.



Resetting when counting (TSX CTY 2A/4A)

At a Glance

Resetting initializes the counting register at zero value. Resetting is:

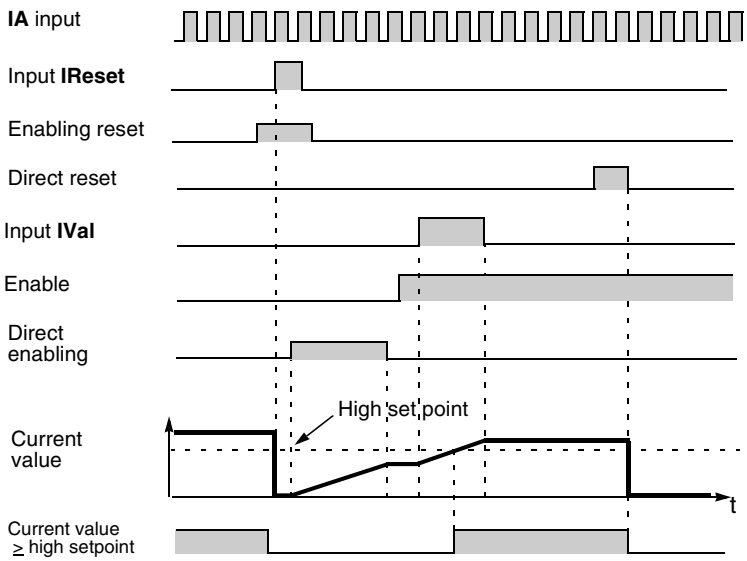
- **triggered**
  - by the hardware: on the rising or falling edge of the **IReset** physical input with the software being pre-enabled, **Reset enable** command,
  - directly by the software: **Direct reset** command.
- **automatic**, will be done when the zero value is crossed.

Note: IReset input is physically the same as the **IPres** input used for presetting in down counting mode.
The diagram block of the **Reset** function is the same as that of the function for **Preset** (See *Preset in downcounting mode (TSX CTY 2A/4A), p. 47*).

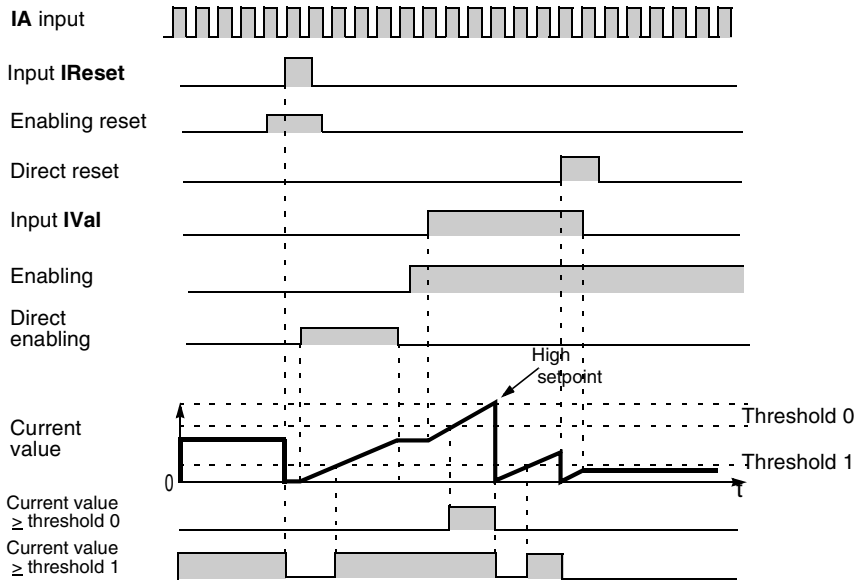
Note: the two types of Reset (triggered and automatic) are independent and can co-exist.

Time diagrams

The following figures show examples of triggered and automatic resetting to zero: The time diagram below shows both a triggered reset on the rising edge of the **IReset** input, and a direct (software) reset. The high setpoint is crossed without automatic reset. The associated language objects are described in the section on implicit objects (See *Details of implicit exchange objects, p. 161*).



The time diagram below shows a triggered reset on the rising edge of **IReset** input, a direct (software) reset and an automatic reset when crossing the high setpoint.



Preset in up/down counting (TSX CTY 2A/4A/2C)

Presetting modes

In the combined up/down counting modes there are 7 **hardware preset** modes which are combinations relating to states and/or edges of **Ipres** and **IZ** physical inputs:

- **IPres** rising edge,
- **IPres** falling edge,
- rising edge of **IPres** + direction / falling edge of **IPres** - direction,
- rising edge of **IPres** - direction / falling edge of **IPres** + direction,
- status of **IPres**,
- reference point short cam (with incremental encoder),
- reference point long cam (with incremental encoder).

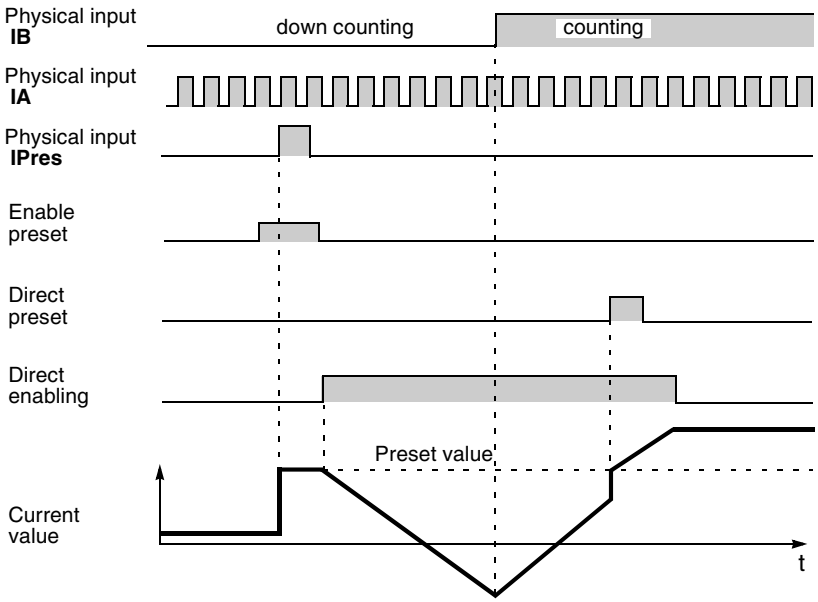
Direct preset (by software) exists alongside the hardware presetting modes mentioned above.

Notes

- There is no automatic preset in combined up/down counting (on crossing a value as there is for up or down counting alone).
- There is no preset in conjunction with absolute encoders (TSX CTY 2C).

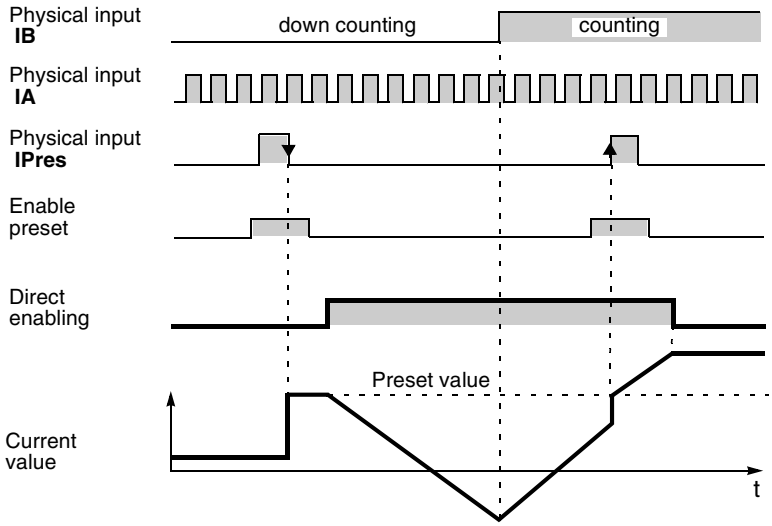
Presetting on IPres edge

The first two presetting modes (on **IPres** rising or falling edge) are the same as those described for down counting only.
The figure below shows an example of preset on the **IPres** rising edge, as well as direct (software) preset. Preset on the **IPres** falling edge is similar. The associated language objects are described in the section on implicit objects (See *Details of implicit exchange objects, p. 161*).



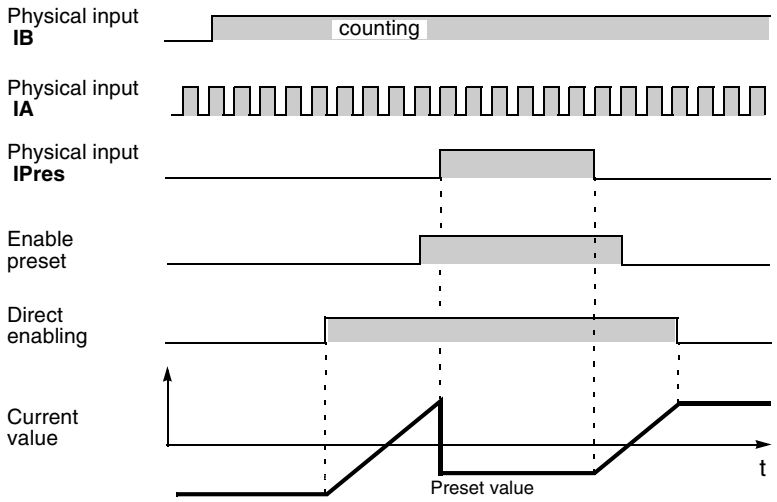
Preset on IPres edge combined with counting direction

The figure below shows an example of preset on the rising edge of **IPres** in up counting (+ direction), and on the falling edge of **IPres** in down counting (- direction). The reverse case is similar.



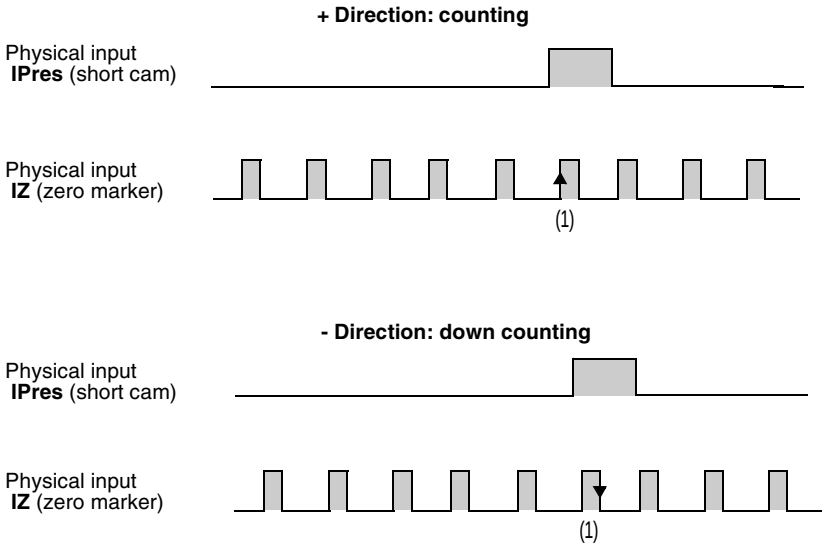
Preset on IPres state

The figure below shows an example of preset on **IPres** state (high level). The counting value is fixed at the preset value for the duration of the active state of **IPres**.



Preset on reference point short cam

This mode and the one that follows (reference point long cam) are to be used with an incremental encoder.
The time diagrams below show preset mode on a **reference point short cam** .

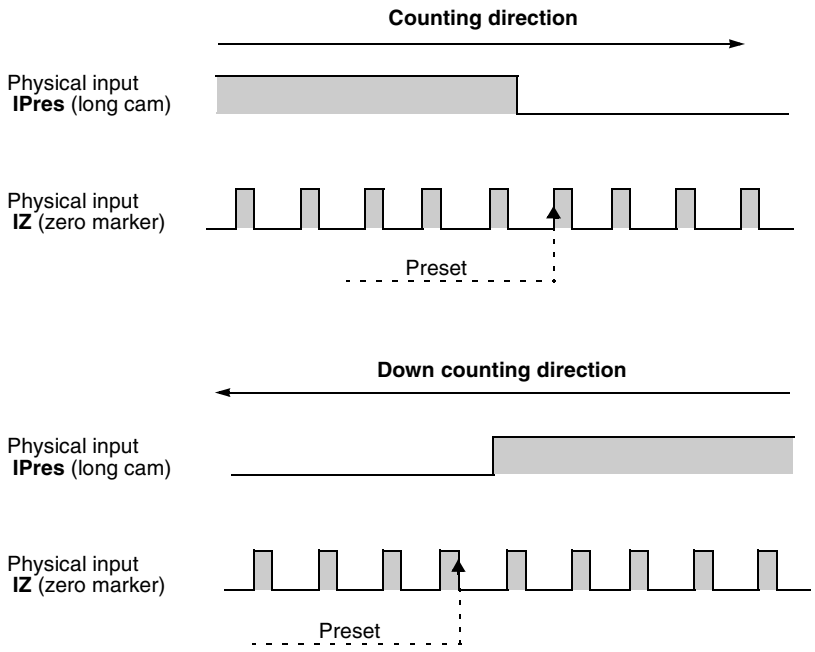


Note: (1) preset is taken into account:
- in + direction (up counting): **IPres** input at state 1, rising edge of marker input at **IZ** revolution and software enabling,
- in direction - (down): **IPres** input in state 1, falling edge of marker input at **IZ** revolution and software enabling.

Note: in principle, as the short cam is less than an incremental encoder revolution, the revolution marker is only produced once in the cam. If, however, there are several incremental revolutions in the cam, the last active edge of the revolution marker signal triggers the preset.

Preset on long cam reference point

The time diagrams below show preset mode on a **reference point long cam**.



Note: preset is taken into account on the first rising edge of the **IZ** revolution marker input, which follows the change of **Ipres** input to status 0, both in the counting and the down counting directions and software enabling.

3.5 Description of the comparison function for counting modules

At a Glance

Subject of this section

This section describes the **comparison** function for counting modules.

What's in this Section?

This section contains the following topics:

Introduction to the comparison function

General

Modules TSX CTY 2A, 4A and 2C have a **comparison** function which compares the current value and the captured value with:

- zero value,
- threshold 0,
- threshold 1,
- the high setpoint,
- the low setpoint,
- the rollover crossing.

The table below summarizes the various possibilities:

Element for comparison	**The comparison in relation to the current value is possible for:**	**The comparison in relation to the captured value is possible for:**
Zero value	CTY 2A/4A (down counting only)	No module
Threshold 0	CTY 2A/4A (up counting only) CTY 2A/4A (up/down counting) CTY 2C (up/down counting)	CTY 2A/4A (up/down counting) CTY 2C (up/down counting)
Threshold 1	CTY 2A/4A (up counting only) CTY 2A/4A (up/down counting) CTY 2C (up/down counting)	CTY 2A/4A (up/down counting) CTY 2C (up/down counting)
High setpoint	CTY 2A/4A (up counting only) CTY 2A/4A (up/down counting)	CTY 2A/4A (up/down counting)
Low setpoint	CTY 2A/4A (up/down counting)	CTY 2A/4A (up/down counting)
Rollover crossing	CTY 2C (up/down counting)	CTY 2C (up/down counting)

Note

Crossing thresholds, setpoints and rollover can be subjects of an event process operation.

Comparison in counting or down counting (TSX CTY2 A/4A)

Comparisons in down counting

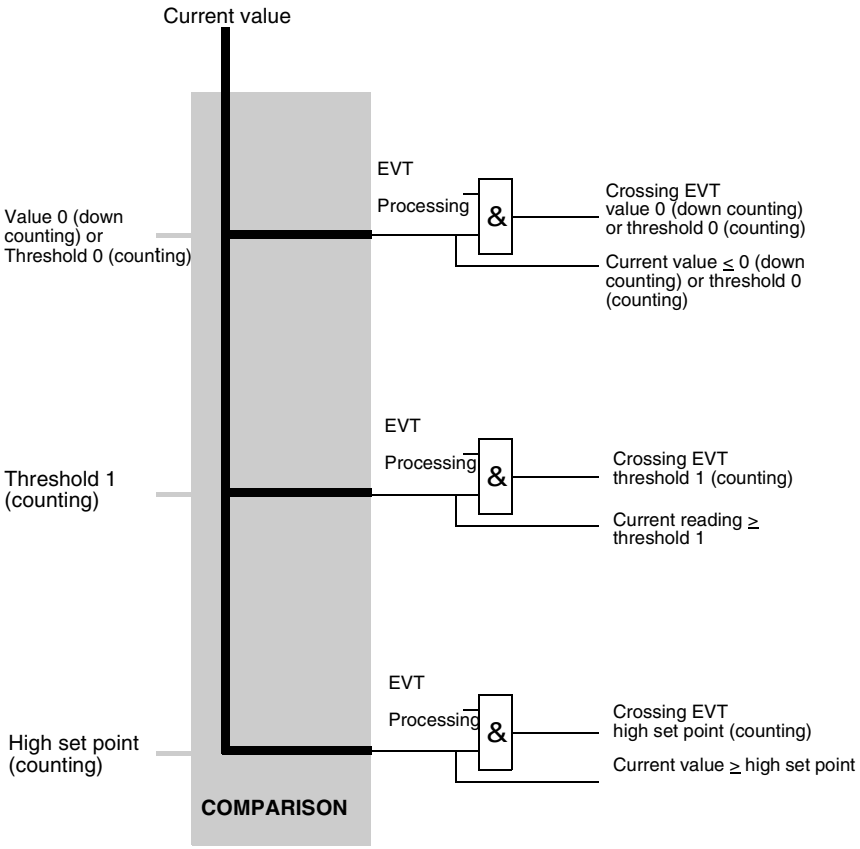
In down counting only mode, a single possibility is autorized:
- comparison of the current value to the zero value.

Comparisons in up counting

In up counting only mode, three possibilities are offered:
- comparison of the current value to the 0 threshold,
- comparison of the current value to the 1 threshold,
- comparison of the current value to the high setpoint.

Operation

The diagram below illustrates the operation of the comparisons available in up counting only and down counting only modes in the TSX CTY 2A/4A modules. The associated language objects are described in the section default objects (See *Details of implicit exchange objects, p. 161*).



Comparison in up counting/down counting mode (TSX CTY 2A/4A)

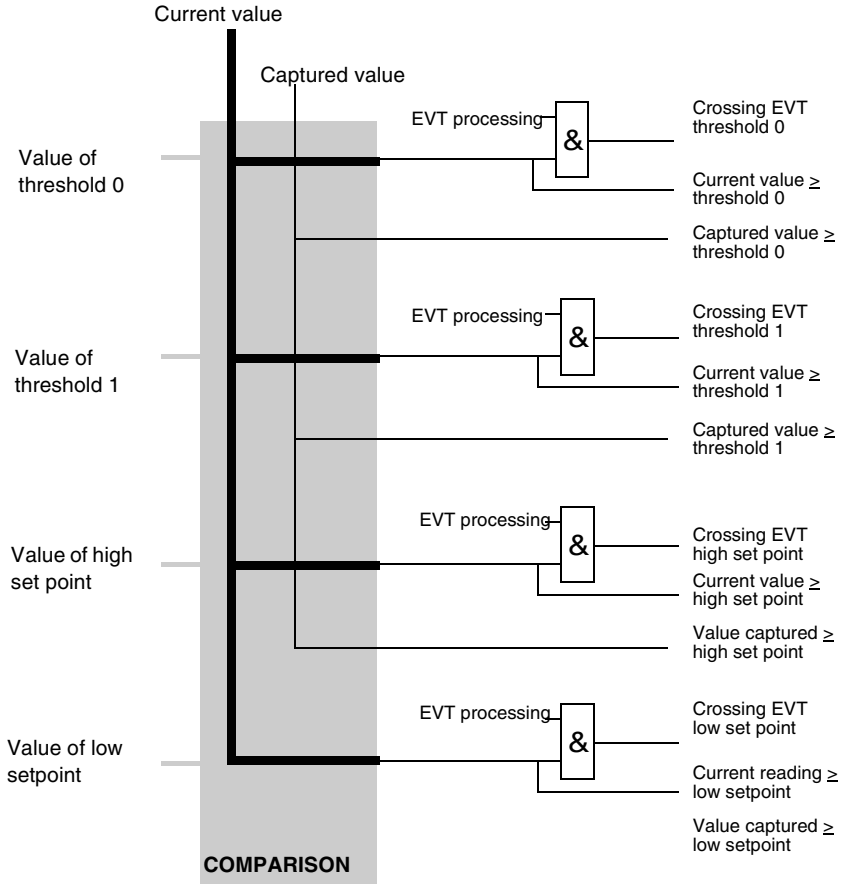
Possible comparisons

In combined up counting/down counting mode, comparisons are possible with:
- two thresholds (0 and 1 threshold),
- and two setpoint values (high and low).

There are therefore 8 possible comparisons.

Operation

The figure below illustrates the operation of comparisons available in up counting / down counting mode in TSX CTY 2A /4A modules. The associated language objects are described in the section default objects (See *Details of implicit exchange objects, p. 161*).



Comparison in up/down counting and measurement mode (TSX CTY 2C)

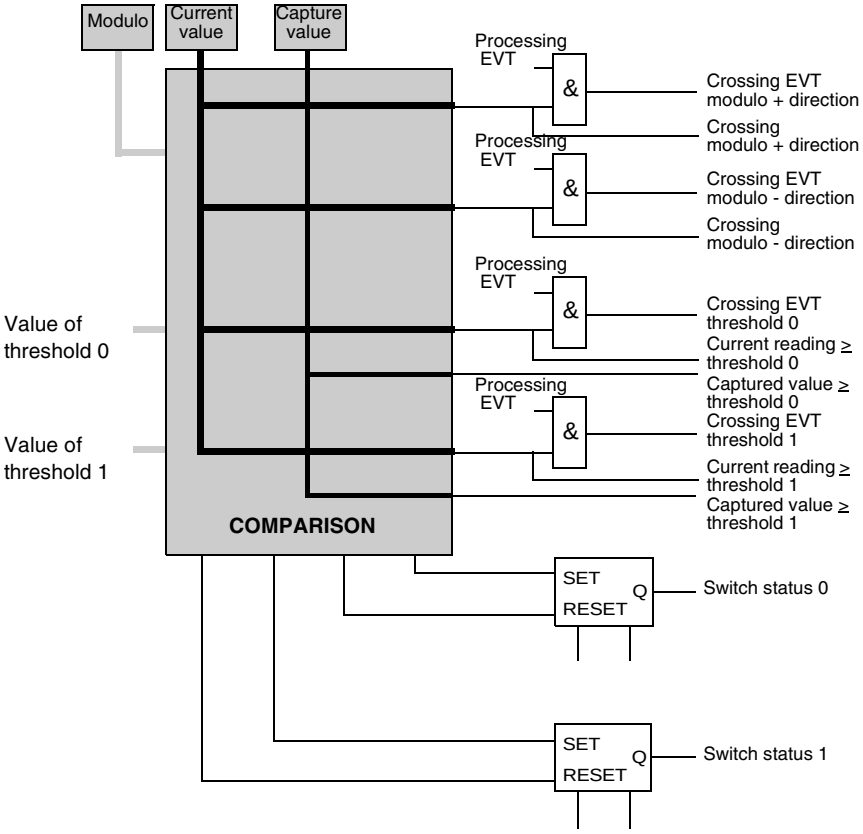
At a Glance

In up/down counting and measurement mode, comparison can be used with:
- threshold 0,
- threshold 1.

There are four ways of comparing: current value and captured value in relation to the thresholds.

Operation

The figure below illustrates the operation of comparisons available in up/down counting mode in TSX CTY 2C modules. The associated language objects are described in the implicit objects (See *Details of implicit exchange objects, p. 161*) section.



3.6 Description of switches associated with counting modules

At a Glance

Subject of this section

This section describes how switches associated with counting modules operate.

What's in this Section?

This section contains the following topics:

Introduction to storing switches

General

Some temporary circumstances that occur during up or down counting are stored in switches. The number of these switches (one or two) depends on the operating mode.
Switch outputs can be tested by the software and in some cases sent to physical outputs (reflex outputs).
Switches have automatic adjustable setting to 1 conditions (SET) and resetting to zero conditions (RESET) and direct setting to 1 or 0 conditions by the software. The latter always have the highest priority. Priority rules are set out in the sections *Switches in counting mode (TSX CTY 2A/4A), p. 68* and *Switches in up/down counting mode (TSX CTY 2A, 4A, 2C), p. 71*.
This section describes SET and RESET conditions according to the modules and their operating modes.
Adjusting switch conditions is described in the section *Adjustment of the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C, p. 121*.

Switches in down counting mode (TSX CTY 2A/4A)

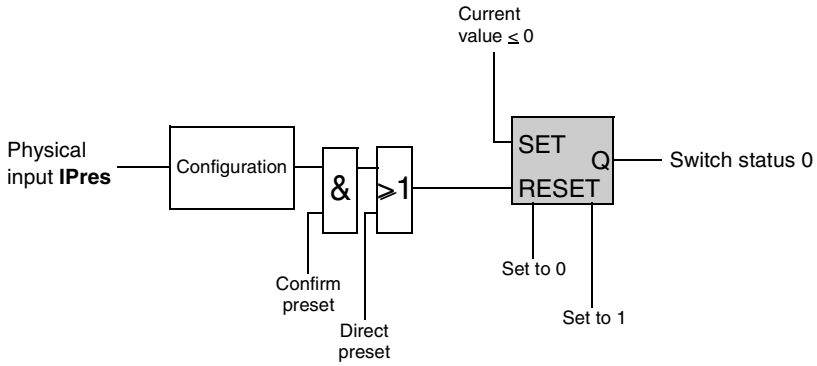
General

In counting down mode alone, TSX CTY 2A/4A modules only have one switch setting (0) with pre-defined automatic SET and RESET conditions:

- SET: current value less than or equal to 0,
- RESET: direct (software) preset or hardware preset done.

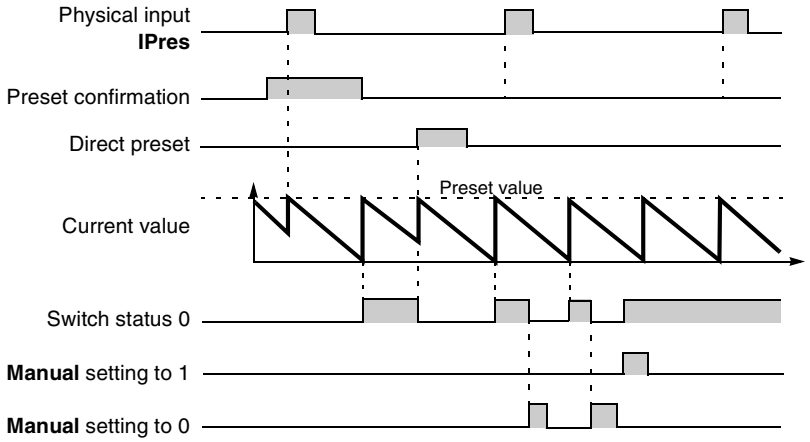
Inputs for switch 0

The following figure shows inputs for switch 0 in down counting mode. You will also see **manual** inputs for setting to 0 or 1.



Operating example

The following time diagram shows how switch 0 operates.



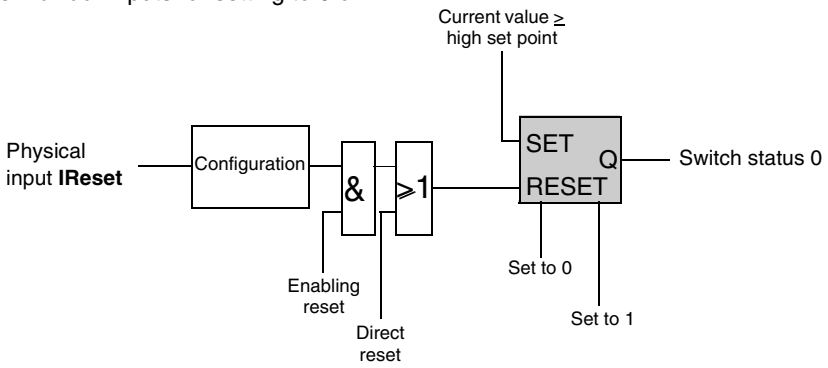
Switches in counting mode (TSX CTY 2A/4A)

General

In counting mode alone, TSX CTY 2A/4A modules have two switch settings:
- switch 0 where automatic SET and RESET conditions are predefined,
- switch 1 where automatic SET and RESET conditions can be adjusted using 5 combinations.

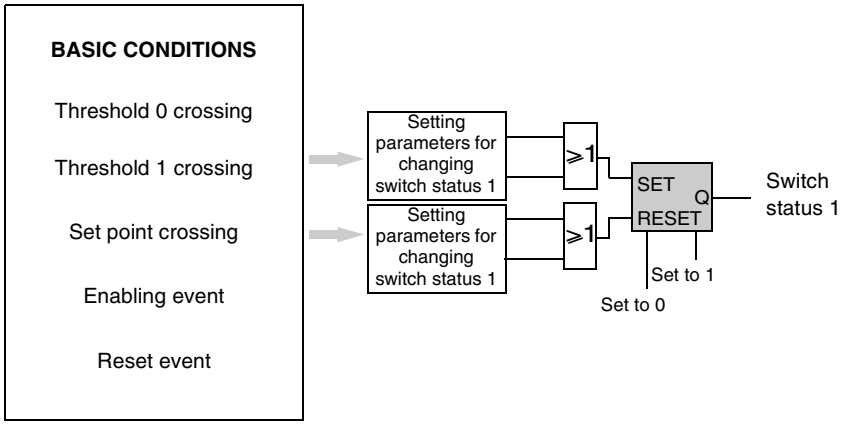
Inputs for switch 0

The following figure shows inputs for switch 0 in counting mode alone. You will also see **manual** inputs for setting to 0 or 1.



Inputs for switch 1

The following figure shows inputs for switch 1 in counting mode alone, as well as direct inputs for setting to 0 or to 1.



Note: switches 0 and 1 have the same basic SET and RESET conditions. If these happen at the same time the global RESET combination takes priority over the SET combination.

Switch setting conditions and priorities

The following tables shows the conditions for switches 0 and 1 with their related priorities.

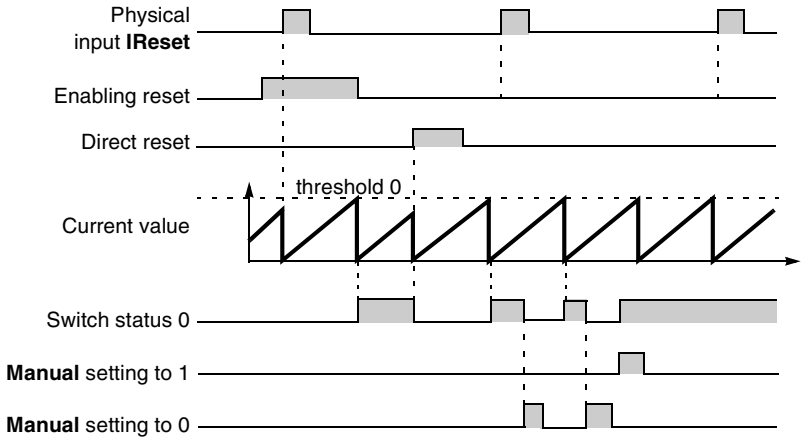
Priority	Switch 0	Switch 1
Greater priority ↓ Lesser priority	**Manual** setting to 0. **Manual** setting to 1. RESET: Reset to zero (direct or done). SET: current value greater than or equal to the setpoint value	**Manual** setting to 0. **Manual** setting to 1. RESET SET Enabling event Resetting to zero event Setpoint crossing Crossing threshold 1 Crossing threshold 0

Notes

- Adjustable conditions (events, crossing thresholds or set points) are the same for switch 1 SET and RESET conditions.
- The RESET input takes priority over the SET input.

Operating example (switch 0)

The following time diagram shows how switch 0 operates.

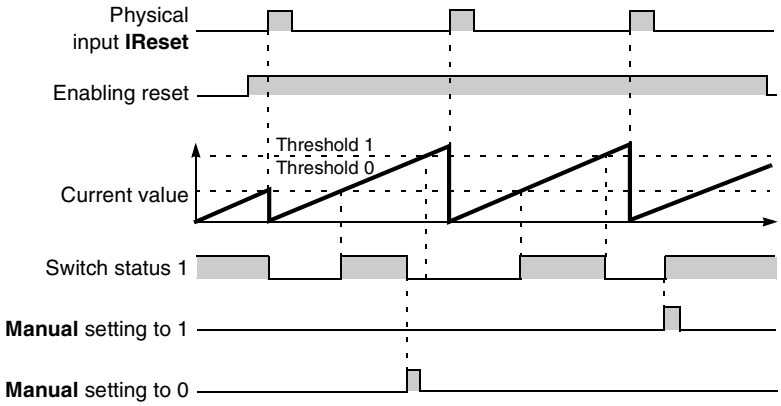


Operating example (switch 1)

The following time diagram shows how switch 1 operates, with the adjustments shown below:

If…	**Then: final state…**
Enabling event	-
Resetting event	R
Setpoint crossing	-
Crossing threshold 1	R
Crossing threshold 0	S

Time diagram showing how switch works:



Switches in up/down counting mode (TSX CTY 2A, 4A, 2C)

General

These modules offer two storing switches in up/down counting mode (and also measurement mode where applicable for the TSX CTY 2C module).
Switch SET (setting to 1) and RESET (setting to 0) conditions can be adjusted using combinations of:

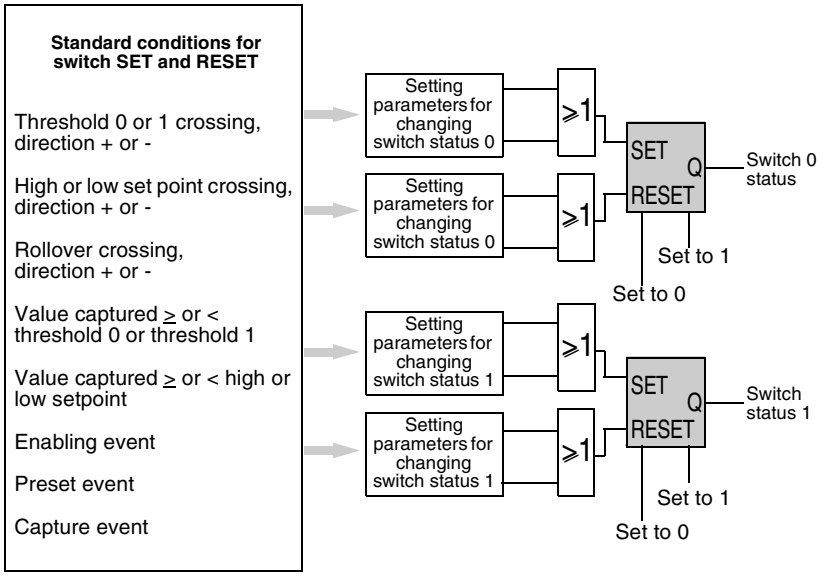
- 17 basic conditions for TSX CTY 2A/4A modules,
- 13 basic conditions for TSX CTY 2C modules.

These basic conditions are relative to:

Module	Overshooting by the current value of:	Captured value settings in relation to:	Events:
TSX CTY 2A/4A	thresholds and setpoints	thresholds and setpoints	enabling, preset and capture
TSX CTY 2C	thresholds and rollover	thresholds	enabling, preset and capture

Switch inputs

The following figure shows switch inputs in up/down counting mode. Also note **manual** inputs for setting to 0 and 1.



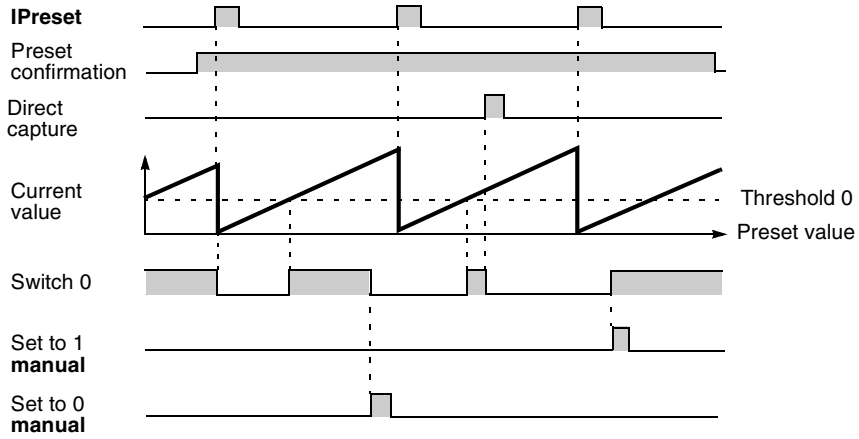
Note: switches 0 and 1 have the same basic SET and RESET conditions. If these happen at the same time the global RESET combination takes priority over the SET combination.
According to the type of module, some conditions cannot be set. Conditions and their priority status are listed later on.

Operating example

The time diagram below shows how switch 0 operates with the following adjustments:

If…	**Then: final state…**
Crossing threshold 0 in + direction	S
Captured value > threshold 0	R
Other conditions	-

Time diagram showing how switch 1 operates.



Basic conditions and priorities

The following table shows the basic conditions for setting switches 0 and 1 with their related priorities.

Priority	TSX CTY 2A/4A	TSX CTY 2C
Greater priority  Lesser priority	Manual setting to 0 Manual setting to 1 RESET SET Enabling event Captured value position in relation to low setpoint Captured value position in relation to high setpoint Captured value position in relation to threshold 1 Captured value position in relation to threshold 0 Capture event Low setpoint crossing High setpoint crossing Crossing threshold 1 Crossing threshold 0 Preset event	Manual setting to 0 Manual setting to 1 RESET SET Enabling event Crossing threshold 1 Crossing threshold 0 Rollover crossing Captured value position in relation to threshold 1 Captured value position in relation to threshold 0 Capture event Preset event

Note: in reality, crossing setpoints, thresholds and rollovers group two conditions each time, according to the direction (+ or -) of the crossing.

3.7 Description of physical outputs associated with counting modules

At a Glance

Subject of this section

This section describes how physical outputs associated with counting modules operate.

What's in this Section?

This section contains the following topics:

Counting modules physical outputs

General

Each counting module channel offers one to four physical outputs, **Q0** to **Q3**, according to the module.
Q0 and **Q1** outputs, available in all modules, can be configured using two modes:

- **manual** mode: the state of the output is monitored by the software as for a discrete output,
- **automatic** mode: the output copies the associated switch state (switch 0 or 1), and therefore the latches defined for these switches.
  This mode is used to set up reflex actions in the module.

Q2 and **Q3** outputs (only available in TSX CTY 2C modules), have more limited configuration options (see: *Introduction to TSX CTY 2C module outputs, p. 81*.
The behavior of outputs in the event of a fault (ie. in fallback mode) varies according to whether the module is a TSX CTY 2A / 4A (See *Outputs fallback mode for TSX CTY 2A/4A modules in the event of a fault, p. 80*) or a TSX CTY 2C (See *Outputs fallback mode for TSX CTY 2C modules in the event of a fault, p. 84*).
In the event of a trip (ie. through a voltage surge or short-circuit), outputs must be reset **manually** using the application, or **automatically** when the voltage surge disappears.

Introduction to modules TSX CTY 2A/4A physical outputs

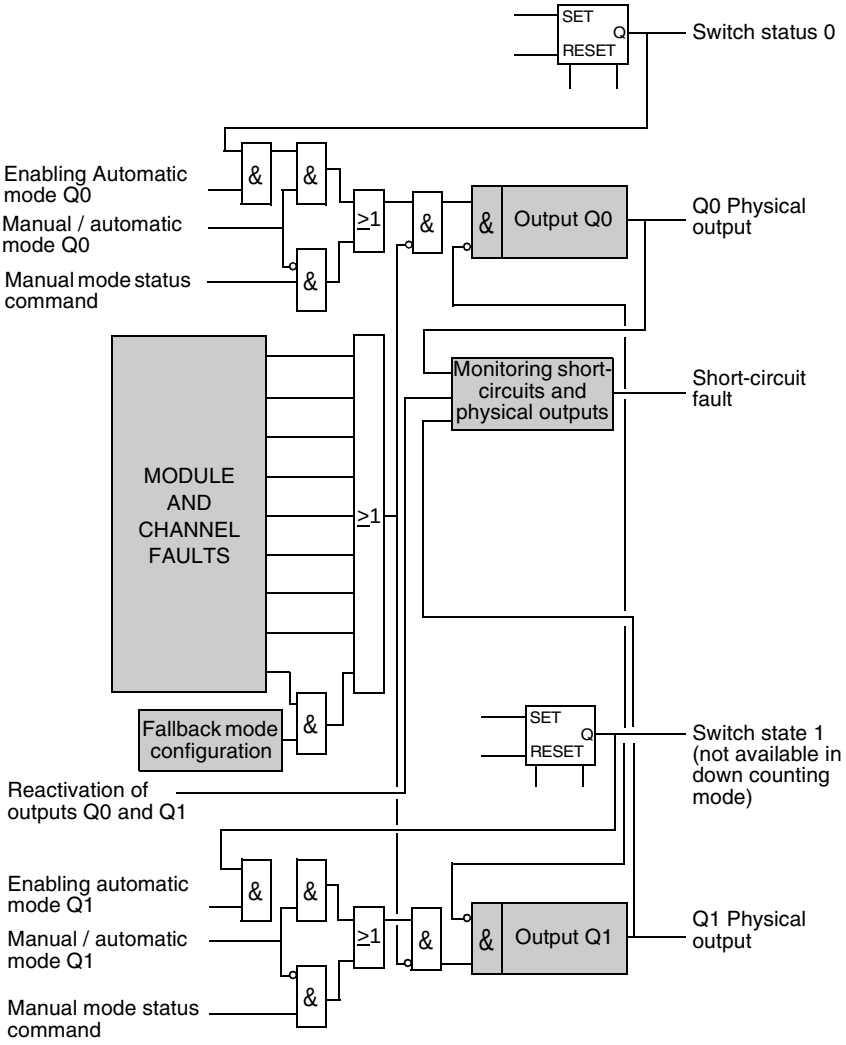
Description

Each channel in TSX CTY 2A and 4A modules has two physical outputs: **Q0** and **Q1**. These outputs can be used in manual mode (discrete). The state of these outputs is then set by commands sent by the software.
Q0 and **Q1 physical** outputs can also be configured automatically. Automatic mode is used to install reflex actions in the module by copying the state of the storing switches respectively.

- Counting: two switches are available. The two outputs can be configured automatically.
- Down counting: only switch **0** is available. Only one output (**Q0**) can be configured automatically.

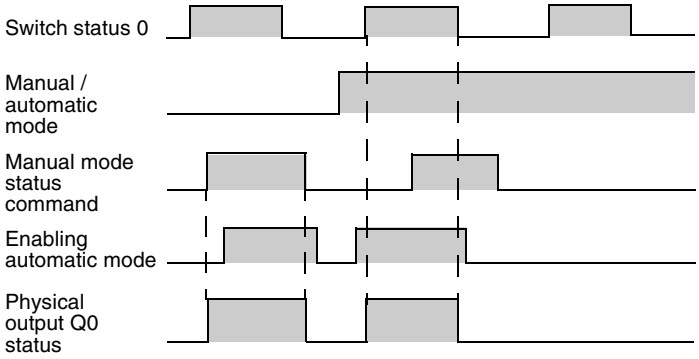
Structure of outputs

The figure below shows the structure of physical outputs from a TSX CTY 2A/4A channel. The associated language objects are described in the implicit objects (See *Details of implicit exchange objects, p. 161*) section.



How outputs operate

The time diagram below shows how outputs operate (using **Q0** output as an example).



Outputs fallback mode for TSX CTY 2A/4A modules in the event of a fault

Why have a fallback mode?

The physical outputs from TSX CTY 2A/4A modules can directly control certain parts of the process. It is therefore necessary for the safety of the operation to have a fallback mode in the event of a fault in the outputs themselves, the module or the application.

Fallback mode

The state of **Q0** and **Q1** physical outputs is **forced to 0** when one of the following faults occurs:

- tripped output (short-circuit or voltage surge),
- module failure,
- self-testing in progress,
- auxiliary input/output fault,
- counting input fault,
- fault in the counting application,
- application fault,
- stop in communication with the PLC: special fault for which the fallback mode can also be configured following the table below.

All these faults are indicated by the software. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) section.

Communication stop – special case

If communication with the PLC stops:

If the fallback mode is configured to…	Then outputs are…
Reset,	forced to 0,
maintain	maintained in the state they were in before the fault appeared.

Indicating a trip

In the event of a voltage surge or short-circuit of an output, a trip is indicated:

- by software via language objects,
- and by LEDs:
  - the module **I/O** LED is always lit,
  - the **CH** LED associated with the channel with voltage overload flashes.

In addition, a current limit of 625 mA is put in place.

Introduction to TSX CTY2C module outputs

Description

Each TSX CTY 2C module channel has four physical outputs: **Q0, Q1, Q2** and **Q3**. All outputs can be used in manual (discrete) mode. The state of these outputs is then set with commands sent by software. The associated language objects are described in the implicit objects (See *Details of implicit exchange objects, p. 161*) section.
Q0 or **Q1** physical outputs can also be configured automatically. The state of **Q0** and **Q1** outputs is thus that of counter output **0** and **1** outputs respectively. Automatic mode is used to implement reflex actions in the module, according to the state of the up/down counter.

Q2 output – special case

Q2 is actually an input/output which can be configured:
- either as an enabling physical input **IEna**,
- or as a physical output **Q2** (in manual mode only).

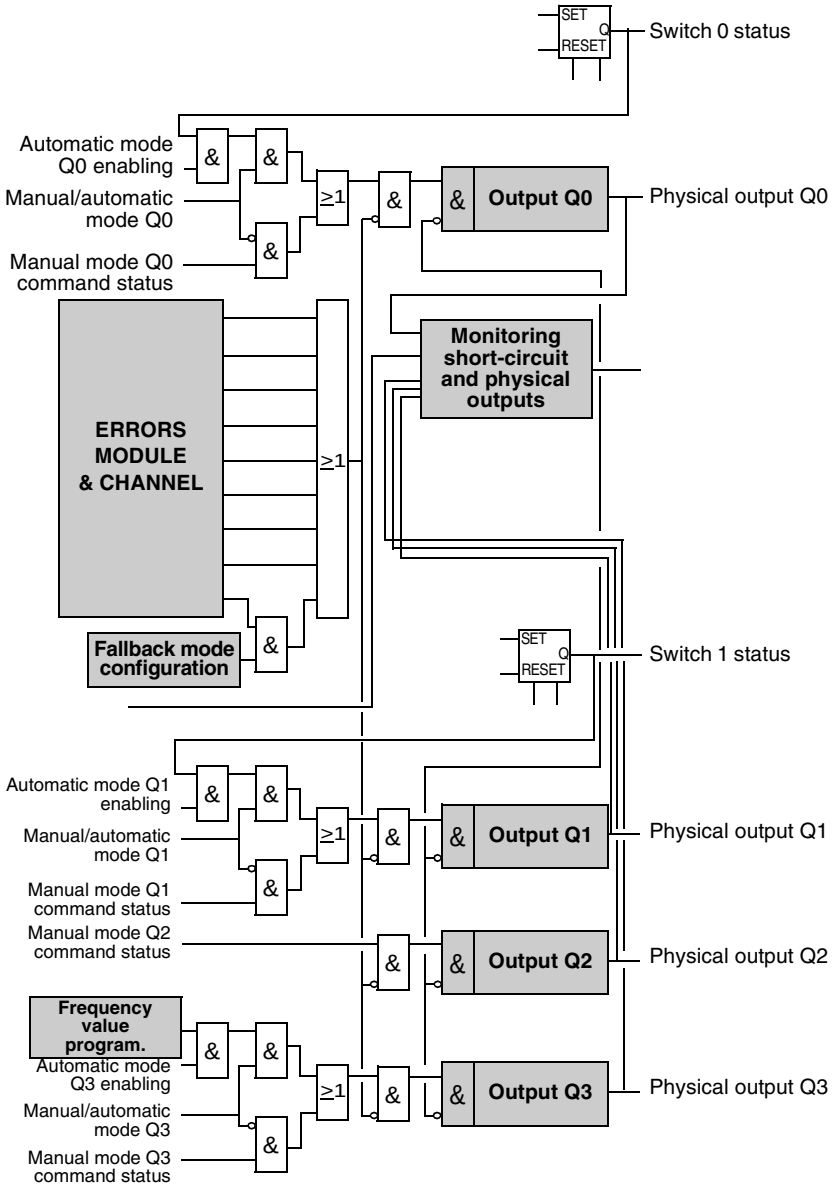
Q3 output – special case

Q3 output can be used:
- either in manual mode,
- or in frequency mode which can be programmed from 1 ms to 4000 s by increments of 1 ms.

Programmable frequency output enables an external synchronization marker to be used on several channels of modules.

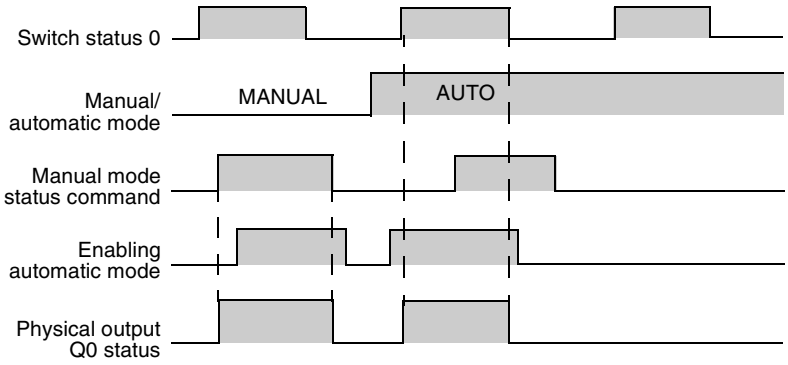
Structure of outputs

The figure below shows the structure of module TSX CTY 2C physical outputs. The associated language objects are described in the default objects (See *Details of implicit exchange objects, p. 161*) section.



How outputs operate

The time diagram below shows how outputs operate (using Q0 output as an example).



Outputs fallback mode for TSX CTY 2C modules in the event of a fault

Why have a fallback mode?

The physical outputs from TSX CTY 2C modules can directly control certain parts of the process. It is therefore necessary for the safety of the operation to have a fallback mode in the event of a fault in the outputs themselves, the module or the application.
For a TSX CTY 2C modules, the fallback modes vary according to:

- the nature of the fault,
- the operating mode (manual or automatic) of the faulty output.

Fallback conditions in manual mode

In the event of one of the following faults…:	Q0 to Q3 outputs in manual mode…:
• module failure, • self-testing in progress, • counting application fault (invalid software configuration or channel reconfiguration), • auxiliary inputs/outputs fault, • auxiliary input power supply fault, • tripping of any output (short circuit or voltage surge), • measurement overrun or overspeed.	are reset to zero.
• counting input fault, • counting application fault (adjustment fault), • fault in the encoder power supply or sensor, • encoder line break or short-circuit, • SSI series frame fault, • specific absolute encoder fault.	are **not** reset to zero.

All these faults are indicated by the software via language objects. To find out the associated language objects, see CH_FLT word for explicit language objects (See *Channel standard fault, %MWxy.i.2, p. 168*).

Fallback conditions in automatic mode

When outputs are in automatic mode, they are reset to zero whatever the fault. In this way the module ensures the safety of the operation, even if the fault is masked. These faults are also indicated by the software.

Communication fault – special case

In the event of PLC communication stopping, independently of manual or automatic output mode:

If the fallback mode is configured to…	Then outputs are…
Reset,	forced to 0,
maintain.	maintained in the state they were before the fault appeared.

Indicating a trip

In the event of a voltage surge or short-circuit of an output, a trip is indicated:
- by software via language objects, as mentioned before,
- and by LEDs:
  - the module **I/O** LED is always lit,
  - the **CH** LED associated with the voltage overload channel flashes.

In addition, a current limit of 625 mA is put in place.

Reactivating outputs after a trip (TSX CTY 2A/4A/2C)

At a Glance

When a fault has tripped one or several physical outputs, it is essential to reactivate them. This operation varies according to whether the configuration is set up to be manual or automatic.
When a trip causes an incorrect operation of a PLC-controlled process, it is recommended to link output reactivation to a manual operation (for example: press a button to acknowledge, etc.). The operator can then take the necessary action with regard to the automatic process and personnel safety (e.g. request to change to manual mode).

Note: it is possible to program an automatic reactivation, if under the responsibility of the user and PLC-controlled process allowing.

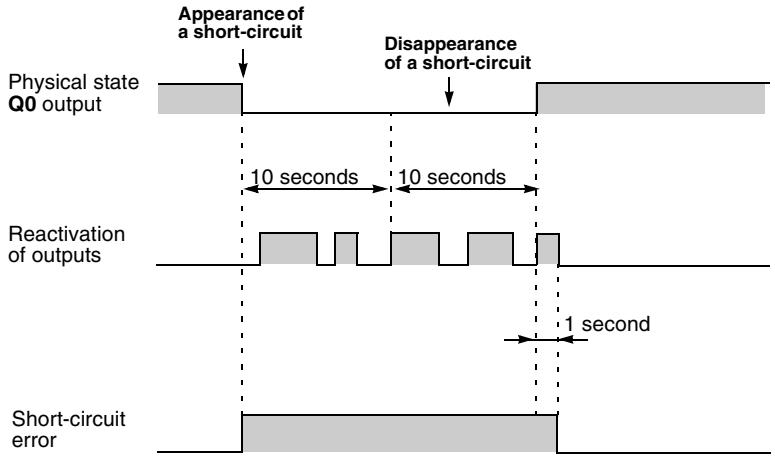
Basic principle

When one of the physical outputs short-circuits, all outputs are set to 0 by the down counting module. In spite of this, as long as the short circuit is present, physical outputs on 0 must be blocked for safety reasons:

- whatever the mode, (**manual** or **automatic**), you must **disable** output: set physical output enabling bits to 0,
- in **manual** mode: program a **set to 0** of physical output manual command objects.

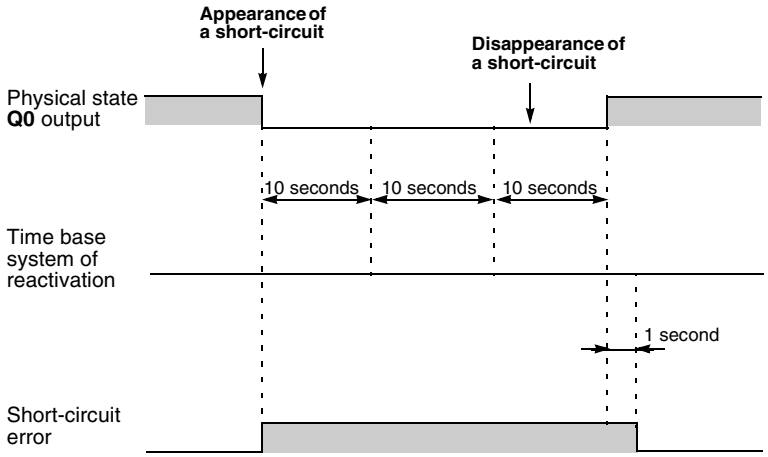
Principle of manual reactivation

The **short-circuit fault** bit is set on 1 as soon as a short-circuit appears. The **output reactivation** bit must be enabled to reactivate physical output, providing the **manual reactivation** mode has been configured.
The program can, for example, wait until an acknowledgement button has been pressed before activating this bit.
Reactivation will take effect at least **10** seconds after the short-circuit has been detected, providing that the short-circuit is no longer present. To find out the associated language objects, see implicit objects (See *Details of implicit exchange objects, p. 161*) and explicit objects (See *Breakdown of objects for user defined exchange, p. 168*).
The time diagram below shows how a short-circuit is manually acknowledged.



Principle of automatic reactivation

Reactivation is automatically requested by the module every **10** seconds. The basic time of 10 seconds is synchronous in relation to the appearance of the fault.
The time diagram below shows how a short-circuit is automatically acknowledged.



3.8 Description of the speed measuring function for the TSX CTY 2C module

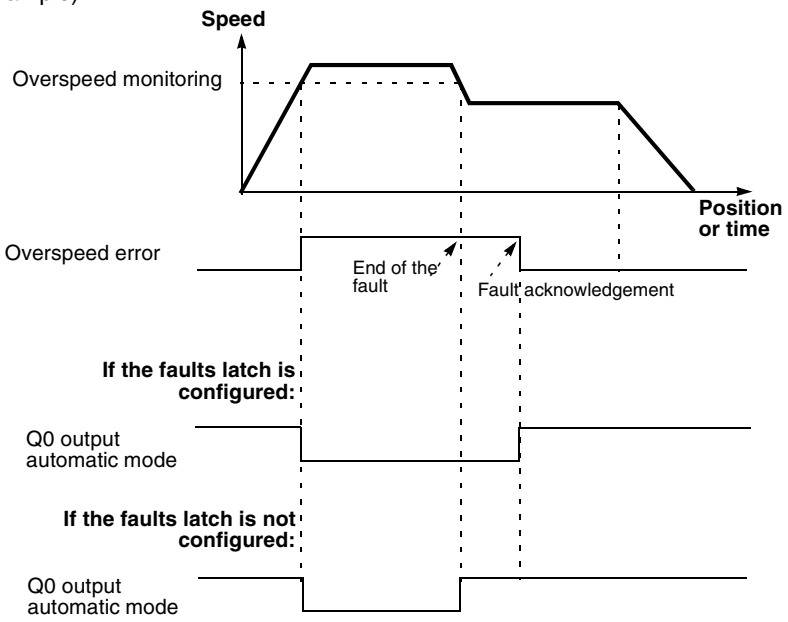
Speed Monitoring Function (TSX CTY 2C)

At a Glance

Speed monitoring, via user adjustable overspeed monitoring, provides a safety measure for the output(s) during an overspeed monitoring overshoot (the outputs are then set to 0). These outputs must be configured in automatic mode.

Operation

The figure below shows reflex output changes according to configuration (Q0 output example).



Faults Latch

When the output is in automatic mode:

- if the faults latch is configured: when the fault has disappeared, it must be acknowledged in order to set the output to its initial state before the fault. A premature acknowledgement has no effect.
- if the faults latch is not configured: when the overspeed error disappears, the output is reset to its initial state.

Note: outputs in manual mode

The outputs in **manual** mode are reset to zero by the overspeed error.

Choice of the Measurement Period

The default measurement period is 1 second, which can cause an excessive delay in the detection of overspeed, or an insufficient precision. It is possible to program the measurement period according to the speed to be monitored and the required precision:

$$\text{Measurement period} \geq \frac{1}{\text{precision x speed}}$$

where the precision is expressed as a decimal value (for example: 0.1% = 0.001) and the speed in pulses/second. The sampling period is expressed in seconds.
Admissible values range from 10^{-2} s to 30 s.
The following table shows the minimum measurement period as a function of the speed to be measured, in order to ensure for example a precision of 0.1%.

Speed to be measured (pulses/second)	Minimum measurement period (s)	Precision (%)
250000 ... 1000000	Greater than or equal to 10^{-2}	0.1
40000 ... 250000	Greater than or equal to 25 10^{-3}	0.1
10000 ... 40000	Greater than or equal to 0.1	0.1
1000 ... 10000	Greater than or equal to 1	0.1
100 ... 1000	Greater than or equal to 10	0.1

3.9 Description of the special functions of the TSX CTY 2C counting module

At a Glance

Subject of this section

This section describes the three special functions for the TSX CTY 2C counting module.

What's in this Section?

This section contains the following topics:

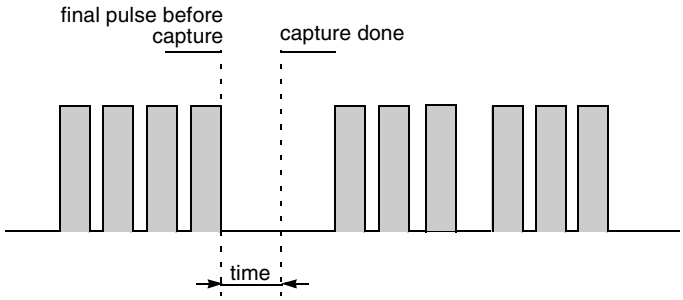
Special function number 1 (TSX CTY 2C)

At a Glance

Special function number 1 calculates the time elapsed between the last up/down count pulse and a capture event. The time between the last pulse and the capture is expressed in milliseconds with an accuracy within + or – 1ms.

Operation

The time diagram below shows how special function number 1 operates.



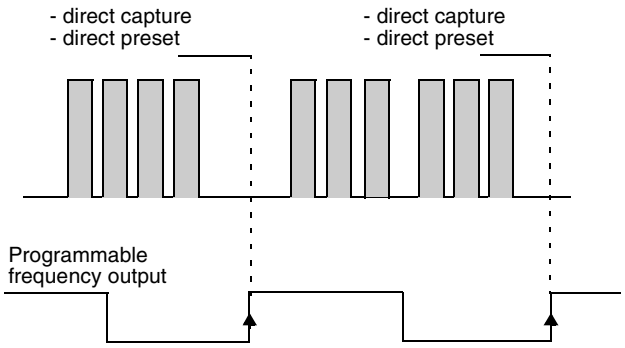
Special function number 2 (TSX CTY 2C)

At a Glance

Special function number 2 triggers a direct (software) capture and a direct (software) preset of the up/down counter on the counting channel. These two events are synchronized with the programmable output frequency (this output must not be used externally in this case).
Each rising edge of the programmable frequency output of the counting channel enables a direct (software) capture of the counter value followed by a direct (software) preset.

Operation

The principle of special function number 2 is shown below:



Notes

The following language objects behave differently in relation to the standard direct capture and preset functions:

- the **%Ixy.i.2 capture done** bit is then positioned on 1 (standard direct capture does not set this bit to 1),
- the **%Ixy.i.1 preset done** bit is then positioned on 1 (standard direct preset does not set this bit to 1),

For more information on associated language objects, see default objects (See *Details of implicit exchange objects, p. 161*).

Special function number 3 (TSX CTY 2C)

At a Glance

Special function number 3 is an extension of the speed monitor and measurement function. It enables the counting and measurement channel to:

- monitor the **correct speed**, corresponding to the boolean function:

(Target speed - X% ≤ Speed measure) AND (Speed measure ≤Target speed + X%)

- and detect a **stationary moving part**, which corresponds to the following boolean function:

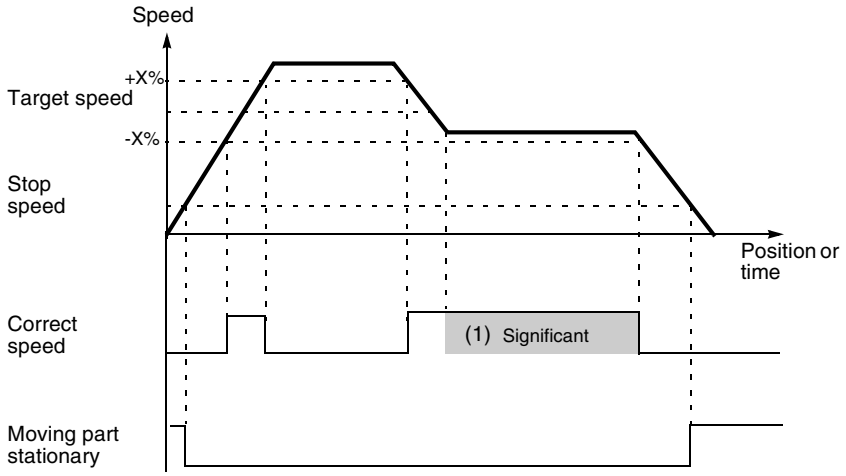
(Speed measure ≤ Stop speed)

Function parametering

Tolerance on the **X%** speed is a configuration parameter completed by the user. The **Target speed** and the **Stop speed** are adjustment parameters set by the application program using a WRITE_PARAM %CHxy.i (*List of explicit exchange objects, p. 167*) or a variables animation table in online mode.

Operation

The time diagram below shows the principle of special function number 3.



(1) Significant zone, no other zone for correct speed or stationary moving part is significant.

Notes

Correct speed and **Moving part stationary** information is only significant for speed plateaus. Managing the relevance of this information is the responsibility of the application program.

3.10 Description of how counting modules deal with faults

At a Glance

Subject of this section

This section describes how faults are processed which can occur in a counting application. It also describes the tools provided by the counting modules to detect and deal with them.

What's in this Section?

This section contains the following topics:

Introduction to dealing with channel and module faults

At a Glance

Counting modules can indicate and identify faults occurring in their configuration and during operation. These indication/identification functions vary according to the type of module.
In addition, module outputs have fallback modes, which are either preset or can be configured, in order to ensure safe operation. Output behavior is described in detail in the *Description of physical outputs associated with counting modules, p. 75* section.

Dealing with TSX CTY 2A/4A module faults

These modules indicate that a fault has occurred, shown by the following default exchange language objects:
- channel error (**%Ixy.i.ERR**),
- module error (**%Ixy.MOD.ERR**),
- invalid measurement (**%IxWxy.i.2:X7**).

The application can identify the cause of the fault by examining the status words. This is only possible if the fault is still present when the processing takes place.

Dealing with TSX CTY 2C module faults

This module uses the same language objects as before to indicate the occurrence of faults.
However, in addition, it has functions for:
- storage, which is used to acknowledge and identify temporary faults,
- and masking, which is used selectively to avoid certain faults being brought up again.

How to deal with an invalid measurement

At a Glance

Invalid measurement information is available to the user, in addition to diagnostics. It is used to detect a loss of counting or measurement data.
This fault management, which is similar to latching, is provided in all counting modules.

Fault context

Faults can be indicated by:
- a cold start or warm restart of the application,
- a fault on counting input:
  - a power supply fault or a break in the sensor line (proximity detector or encoder),
  - a series frame transmission error (TSX CTY 2C),
  - a specific absolute encoder fault (TSX CTY 2C);
- exceeding the capacity of the counting register,
- exceeding the speed threshold (TSX CTY 2C).

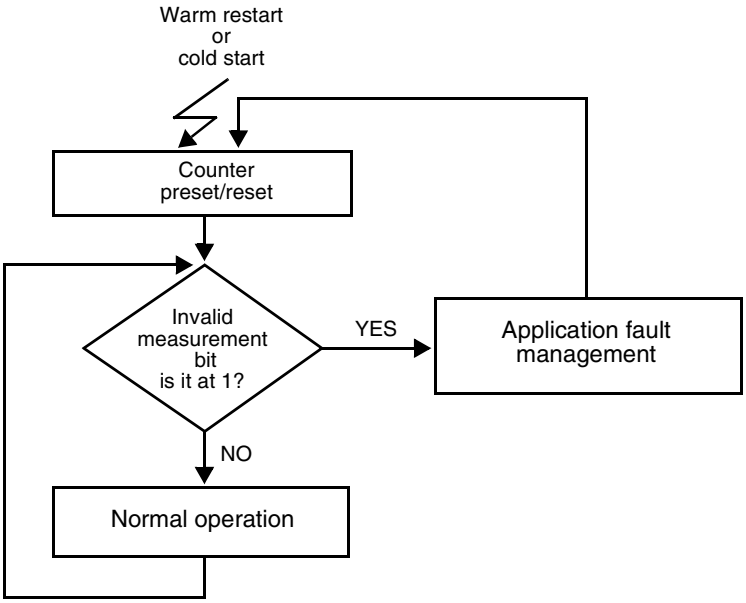
In this case the contents of the counting register cannot be used and counter outputs are set to 0. If this happens, the **Invalid measurement** bit (%IWxy.i.2:X7) is set to 1. When the register is initialized or re-initialized by presetting (or reset) and as long as none of the faults above are still present, the **Invalid measurement** bit switches to 0.

Notes

- The **Invalid measurement** bit set to 1 does not give any information about the fault which caused it. To find this out, the application must examine status words **%MWxy.i.2** and **3** (See *Breakdown of objects for user defined exchange, p. 168*), as long as the fault is still there (TSX CTY 2C).
- Masked faults do not trigger the **Invalid measurement** indicator (TSX CTY 2C).

How to manage the incident

The procedure for managing and removing the invalid measurement by the application is as follows:



Dealing with faults (TSX CTY 2C)

At a Glance

The TSX CTY 2C module has two independent and additional functions, which can be configured by the user, for dealing with channel faults:

- **Storing** is used to indicate occurrence of a fault, however temporary.
- **Masking** certain faults enables the application to continue operating in downgraded mode.

These two functions are selected using the configuration screen (See *Description of the configuration screen for a counting module, p. 102*).

Principle of fault storing

Storing is used to:

- indicate to the application the occurrence of a fault, temporary or otherwise, shown by the channel error bit **%Ixy.i.ERR** and the module error bit **%Ixy.MOD.ERR** (implicit exchanges),
- and identify it via status words (explicit exchanges).

In the absence of a latch, there is a risk that temporary faults will not be detected in the regular scans by the processor, for these language objects are reset to zero as soon as the fault has disappeared.

Principle of masking faults

Masking involves preventing, according to the situation, channel error bits, module error bits and I/O and ERR LEDS being set. The faults concerned are selected individually (fault masking by default).

Note: in the event of an error whether masking was enabled or not, outputs will change to fallback mode to ensure safety, and the CH LED will flash.

Masked faults therefore risk being ignored by the application. Nevertheless, the application can access the fault warning if the COUNT_FLT bit test (**%Ixy.i.3**) has been programmed. Status words continue to be positioned normally, whether faults are masked or not.

Important notes

- Even if a fault has been unmasked, it can pass unnoticed by the application if the store has not been configured, and if this fault is temporary.
- There is also another parallel function which indirectly indicates faults linked to counting, using an Invalid measurement (See *How to deal with an invalid measurement, p. 97*) which has been detected.

Principle of reading and acknowledging faults

When the fault has been indicated to the processor, the latter must read the module status words using a READ_STS instruction.
Module status words are reset to zero when:

- the fault(s) have disappeared,
- and if there is a latch, after the acknowledgement command (bit **%Qxy.i.3**).

Configuration of the TSX CTY2A, TSX CTY4A and TSX CTY 2C modules

4

At a glance

Subject of this chapter

This chapter describes the different configuration options of the TSX CTY2A, TSX CTY4A and TSX CTY2C modules.

What's in this Chapter?

This chapter contains the following topics:

Description of the configuration screen for a counting module

General

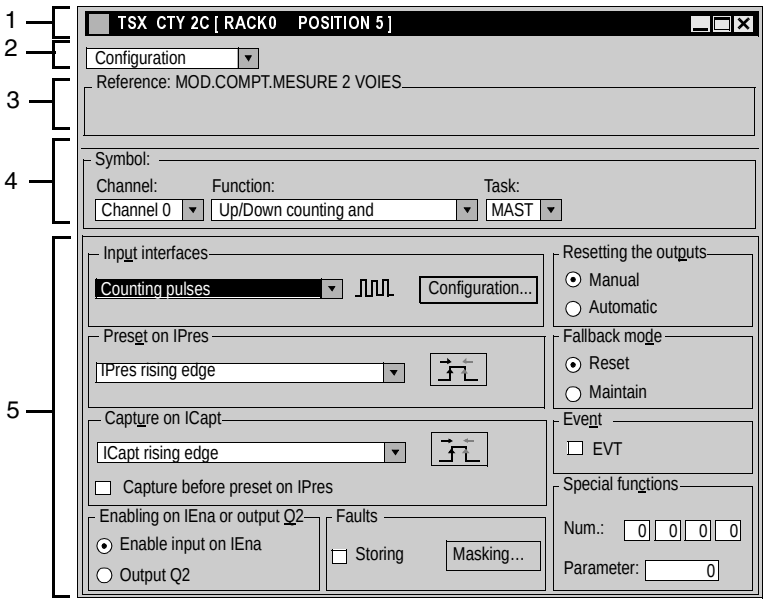
The configuration screen is a graphic tool used to configure a module selected from a rack. It displays the parameters associated with the module channels and is used to modify them in both offline and on-line modes.
It also enables access to adjustment and debugging screens (only in on-line mode for the latter).

Note: refer to the Common application functions (See: PL7 Junior, Pro ; Application-specific functions volume 1) section to see how to configure, adjust and debug an application.

Note: it is not possible to configure a module with the program by using language objects %KW directly. These words are accessed in read-only format.

Illustration

The diagram below shows a configuration screen.



Description

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

Address	Element	Function
1	Title bar	Calls up the reference for the selected module and its geographical address (rack number and physical position in the rack).
2	Command zone	Shows the mode in progress (**Configuration** in this example) and is used to select the other modes: • **Adjusting** • **Debugging** (or diagnostics), can only be accessed in on-line mode.
3	**Module** zone	Calls up the abbreviated name of the module.
4	**Channel** field	Is used to select the counting channel to be configured and the associated counting function: • **Symbol:** channel name set by the user (using the variables editor). • **Counter** or **Channel:** counter number, i.e. of the module channel. • **Function:** counting function from those available for the module concerned. Depending on this selection the headings of the configuration field can differ. By default, **No** function is configured. • **Task:** defines the (**MAST** or **FAST** task in which the channel default exchange objects will be exchanged).
5	Configuration zone	Is used to configure the channel parameters. This field contained various headings, displayed according to the counting function selected. Some selections can be fixed and appear grayed out. In this example, the headings **Input interface** and **Masking** have a button which accesses a sub-menu which must be completed.

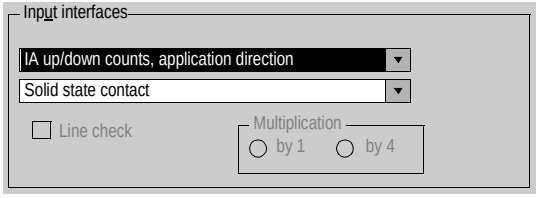
How to configure the counting inputs (TSX CTY 2A/4A)

At a Glance

The counting modules TSX CTY 2A and 4A have several possible input interface types, according to the selected counting function. The interface is configured using the configuration editor.

Procedure

The table below summarizes the procedure for configuring the input interface for a TSX CTY 2A or 4A.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** field: select the counter (i.e. the channel) concerned and its function.
3	In the **Input interfaces** field, select from the drop-down list in the first field:  ● the type of input interface. The details of the different types of interface are described in the section named *Description of counting module input interfaces, p. 35*. **Note:** in up or down counting only, the **IA Input** selection is frozen.
4	Select input filtering from the drop-down list of the second field: ● **Solid state contact** (reduced filtering), or ● **Mechanical contact** (anti-rebound filtering, pulse frequency limited to 100 Hz).
5	If an **incremental encoder** interface was selected at step 3, fill in the following fields, otherwise, go to step 6 to complete the configuration: ● **Line check** (is used to indicate a channel fault in the event of a break in the physical connection of the encoder). ● **Multiplying by 1 or by 4** (is used to increase counting accuracy at the expense of maximum frequency).
6	Configuration of the input interface is complete. Confirm the new configuration. To do this: ● pull down the **Edit** menu and select the **Enable** command.

How to configure the counting and measurement inputs (TSX CTY 2C)

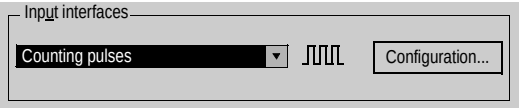
At a Glance

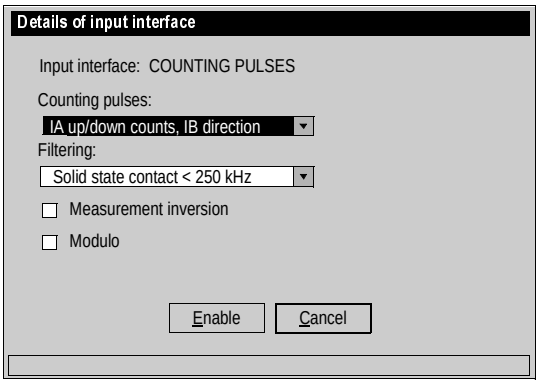
The TSX CTY 2C speed counting and measurement module has several possible input interface types. The interface is configured using the configuration editor.

Procedure

The table below summarizes the procedure for configuring a TSX CTY 2C module input interface.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** field: select the counter (i.e. the channel) concerned and its function.

3	In the **Input interface** field, select from the drop-down list.
	 
	• the type of input interface:
	  • Counting pulses,
	  • Incremental encoder, or
	  • Absolute encoder.

4	Click on the **Configuration** button to access these details.
5	The following screen entitled **Input interface details** depends on the type of interface chosen in step 3: • **Counting pulses** interface: choose the configuration for the physical inputs **IA, IB, IZ**, then the filtering (signal frequency limit).  • **Incremental encoder** interface: choose the filtering, while taking into account the multiplication by 1 or by 4. • **Absolute encoder** interface: refer to the part *How to configure an absolute encoder interface (TSX CTY 2C), p. 107*.

Step	Action
6	Then select, according to the application: ● Measurement inversion (invert the measurement change direction given by the input definition). ● The modulo mode and its value.
7	Configuration of the input interface is complete. Confirm the new configuration. To do this: ● enable the **Input interface details** screen, ● pull down the **Edit** menu and select the **Enable** command.

How to configure an absolute encoder interface (TSX CTY 2C)

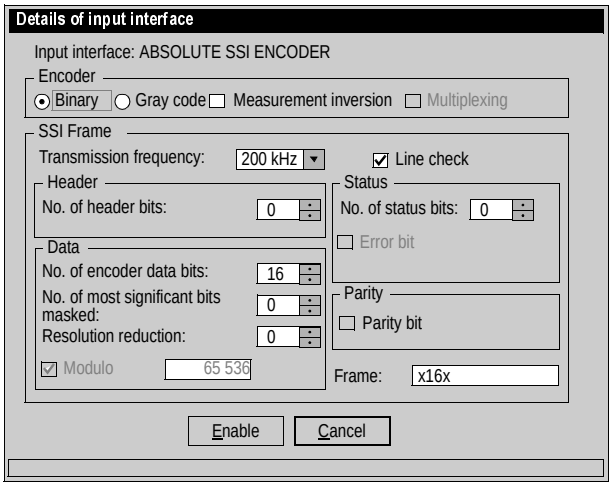
At a Glance

Module TSX CTY 2C as well as having counting pulse inputs, has a specific interface for data acquisition:

- of data from a series output absolute encoder (SSI),
- or of data from one of four parallel output absolute encoders, with adaptation base(s) ABE-7CPA11.

Procedure

The table below shows the interface configuration steps.

Step	Action
1	Follow the general procedure for the TSX CTY 2C module *How to configure the counting and measurement inputs (TSX CTY 2C), p. 105* to start configuring, as far as the next **Input interface details** entry screen.  Some headings are frozen, or not displayed, according to the type of encoder output (series or parallel).
2	Fill in the accessible headings according to the following table.
3	Confirm the new configuration. To do this: ● enable the entry screen above, ● pull down the **Edit** menu and select the **Enable** command.

Definitions and heading options

The table below shows the possible definitions and heading options on the **Input interface details** screen according to the type of absolute encoder output.

Zone	Heading	SSI Series output encoder	Parallel output encoder(s)
Encoder	**Binary** or **Gray**	Coding type.	idem series output
	Measurement inversion	Invert the measurement change direction to a rotation direction given by the encoder.	idem series output
	Multiplexing	Does not exist.	Authorizes encoder multiplexing by the application (See *How to program multiplexing of absolute encoders with parallel outputs, p. 113*).
SSI Frame	**Transmission frequency**	150, 200 (by default), 375, 500, 750 kHz or 1 MHz.	idem series output
	Line check	Line check selection (rupture or short circuit).	Frozen: line check control present.
Header	**Number of header bits**	0 to 4 (0 by default)	Frozen at 0.
Data	**No. of encoder data bits**	8 to 25 (16 by default)	8 to 24 (24 by default)
	No. of masked MSBs	0 to 17 (0 by default)	0 to 16 (0 by default)
	Reducing the resolution	0 to 17 (least significant masked bits, 0 by default).	0 to 16 (0 by default).
	Limit: No. of encoder data bits – No. of most significant masked bits – No. of least significant masked bits > 8 active data bits.		
	Modulo	Frozen (depending on the number of active data bits).	idem series output
Status	**No. of status bits**	0 to 4 (0 by default)	Frozen at 3.
	Error bit	**none** by default. To gain access to this choice, the number of status bits must be > 1.	**none** by default.
	Position	1 to 4 (1 by default). The heading only appears if the error bit is selected.	Frozen at 3. The heading only appears if the error bit is selected.
	Active at 0/1	Error bit active level (1 by default). The heading only appears if the error bit is selected.	idem series output
Parity	**Parity bit**	**none** by default. The choice of parity only appears if the bit is selected. If the parity is odd, the number of status bits is limited to 3.	Frozen: **with**, even parity.
Frame		Information: summary of the series frame characteristics.	idem series output

How to configure the capture of a counting register

At a Glance

This parameter defines how to capture the current counting register value at a precise moment defined by the signal on the physical input **ICapt**. This function only exists in up/down counting (TSX CTY 2A / 4A) and up/down counting and measurement (TSX CTY 2C).
Hardware capture configuration is defined in the **Capture on ICapt** zone in the configuration screen. There are two available options:

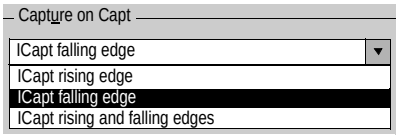
- capture on rising edge,
- capture on falling edge.

Notes

So that capture takes place correctly, it must be enabled beforehand by the software using the corresponding language objects (See *Software commands, %Qxy.i.r bits, p. 163*), run by the application.
Direct capture by the software does not require the following procedure to be set up.

Procedure

The table below summarizes the procedure for configuring hardware capture with TSX CTY 2A, 4A and 2C modules.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Capture on ICapt** zone click on the drop-down menu button. **Result:** the list of options below appears. 
4	Select the hardware capture edge required.
5	Hardware capture configuration is complete. If there are no more parameters to configure, enable the new configuration. To do this: • scroll down the **Edit** menu and select the **Enable** command.

How to configure preset or reset in a counting function

At a Glance

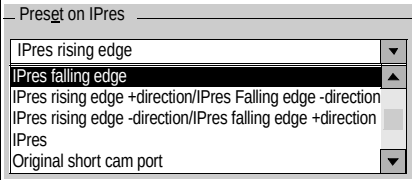
This parameter defines the initialization mode of the counting register at a moment defined by the signal on the **IPres** physical input or **IReset** (depending on the counting function).
Configuring hardware preset or reset is defined in the **Preset on IPreset** zone or **Reset on IReset** zone in the configuration screen.

Notes

- So that the hardware preset **IPreset** or Reset **IReset** does take place, it must first be enabled by the software using the corresponding language objects (See *Software commands, %Qxy.i.r bits, p. 163*), run by the application.
- Direct presetting or resetting by the software does not require implementation of the following procedure.
- The preset value is defined in the adjustment screen (See *How to adjust the preset value, p. 125*).

Procedure

The table below summarizes the hardware preset configuration procedure for the TSX CTY 2A, 4A and 2C modules.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Preset on IPres** or **Reset on IReset** zone click on the drop-down menu button. **Result:** a list of options similar to this one appears. The options that are actually available depend on the type of module and the counting function selected. 
4	Select the hardware preset (or reset) required.
5	Hardware preset (or reset) configuration is complete. If there are no more parameters to configure, enable the new configuration. To do this: • scroll down the **Edit** menu and select the **Enable** command.

How to configure event processing

At a Glance

This parameter is used to associate event processing to the counting channel. This configuration is defined in the **Event** zone of the configuration screen.

Procedure

The table below summarizes the procedure for configuring event processing for all TSX CTY 2A, 4A and 2C modules.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Event** zone, check the **EVT** box. 
4	Select the number of the event task which is associated to the counting channel (this task must then be programmed).
5	Configuration of the event processing is complete. If there are no more parameters to configure, enable the new configuration. To do this: ● scroll down the **Edit** menu and select the **Enable** command.

How to configure the combined input IEna/Q2 (TSX CTY 2C)

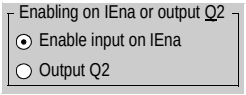
At a Glance

This parameter defines the use of this combined input/output:
- either as counter enable input **IEna**,
- or as physical output **Q2** (in manual mode).

These parameters are set in the **Enable on IEna or Q2 output** zone in the configuration screen.

Procedure

The table below summarizes the procedure for configuring the combined input/output **IEna / Q2** of the TSX CTY 2C module.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Enable on IEna or Q2 output** zone, click on the button corresponding to the required option. 
4	Configuration of the combined input/output **IEna/Q2** is complete. If there are no more parameters to configure, enable the new configuration. To do this: • scroll down the **Edit** menu and select the **Enable** command.

How to program multiplexing of absolute encoders with parallel outputs

At a Glance

Each channel of the TSX CTY 2C module is used to acquire the signals supplied by an absolute encoder with parallel outputs, via a series frame, using adaptation TELEFAST ABE-7CPA11. Use of several TELEFASTs enables up to 4 absolute encoders with parallel outputs to be multiplexed on the same channel.
Multiplexing is managed by the counting application.

Principals of multiplexing

The encoders are addressed by two discrete outputs (belonging to the TSX CTY 2C module, preferably Q2 and Q3 outputs, or to a discrete module). These outputs are looped onto the dedicated TELEFAST inputs. This sends the acquisition value and the address of the present encoder to the TSX CTY 2C module.
The context linked to the encoder (offset value, threshold values, SET and RESET values of the switches), which must change when a new encoder is addressed, is controlled by the application program.
Furthermore, this must take into account the fact that the information regarding position/crossing of thresholds, rollover, the speed value and the overspeed fault is invalid when an encoder is being changed.

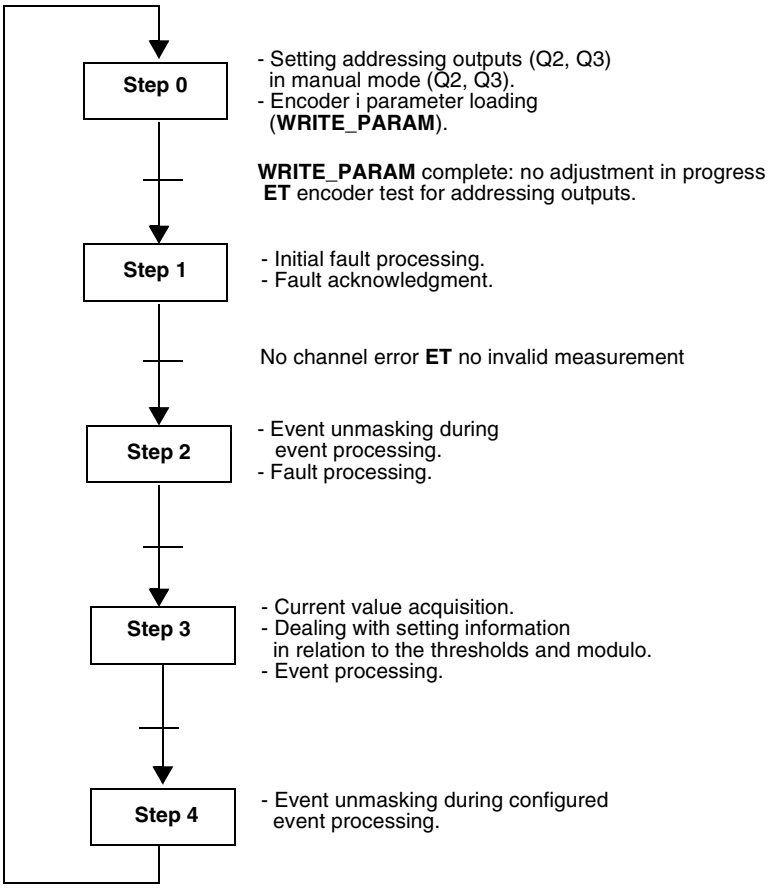
Analysis of multiplexing

This operation involves the following steps for each absolute encoder and at each acquisition:

Step	Action
1	Application loading of the context for the encoder concerned.
2	Application addressing of the encoder concerned.
3	Acquisition of data.
4	Possible wait linked to the acquisition period, then return to step 1 to process the next encoder.

Multiplexing flowchart

The diagram below shows an example of how to program multiplexing:



How to configure operation on switch to zero or setpoint crossing

At a Glance

This parameter defines the automatic reset mode of a TSX CTY 2A/4A module's counting register:

- on switch to the value zero (only during down counting),
- on high setpoint crossing (only during up counting).

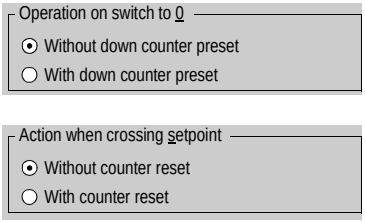
This configuration is defined in the **Operation on switch to 0** or **Operation on setpoint crossing** zone in the configuration screen.

Notes

- These operations are similar to automatic preset or reset respectively. They do not require any preset or reset software enabling.
- The setpoint values are defined in the adjustment screen.

Procedure

The table below summarizes the procedure for configuring the operation on zero value crossing or high setpoint crossing of the TSX CTY 2A and 4A modules.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Operation on switch to 0** or **Operation on setpoint crossing** zone, click on the button corresponding to the required option. 
4	Configuration of the operation on zero or setpoint crossing is complete. If there are no more parameters to configure, enable the new configuration. To do this: • scroll down the **Edit** menu and select the **Enable** command.

How to configure the behavior of faulty outputs

At a Glance

These parameters define the behavior of physical outputs of a channel on TSX CTY 2A, 4A and 2C modules when faults occur due to a voltage surge or short-circuit:

- reactivation of outputs,
- fallback mode.

These configurations are defined in the zones of the same name in the configuration screen.

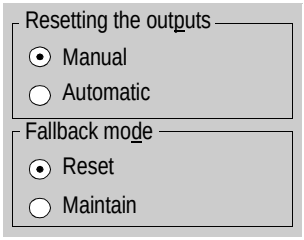
	**CAUTION**
	Ensure that the desired configuration, and in particular the reactivation mode, is compatible with the operation safety requirements **Failure to follow this precaution can result in injury or equipment damage.**

Note

Configurations defined in this way are only valid when there is an overload or short-circuit of one or more outputs. In the event of a different kind of fault, the fallback mode is predefined according to the fault and the module type (see *Description of physical outputs associated with counting modules, p. 75*).

Procedure

The table below summarizes the procedure for adjusting the behavior of faulty outputs.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Reactivation of outputs** zone, select: ● **Manual,** or ● **Automatic**. 
4	In the "Fallback mode" zone, select: ● **Reset,** or ● **Maintain**.
5	Configuration of physical output behavior is complete. If there are no more parameters to configure, enable the new configuration. To do this: ● scroll down the **Edit** menu and select the **Enable** command.

How to configure a special function (TSX CTY 2C)

At a Glance

The special functions of the TSX CTY 2C module are used to respond to certain particular needs that are not covered by the standard functions.
The table below shows the objects of these functions and the numbers of the version required for the module:

Function No.	Description	Version module
1	Time lapsed between the last pulse and a hardware capture occurrence.	1.0
2	Triggering a direct and synchronized capture and preset with each rising edge of the programmable frequency output.	1.0
3	Speed checks **correct** and **moving part stationary**.	1.1

Functions that are specific to a case can also be developed.

Compatibility of special functions

Any attempt to configure a counting module with an unsupported special function will result in an application error.
The counting module version is indicated:

- on the product reference label on the right-hand side of the module,
- in the **Module** zone of the debugging screen (on-line mode).

Configuring several special functions

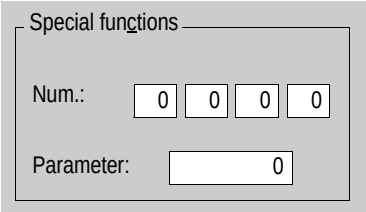
It is possible to configure two special functions simultaneously if they are not exclusive, that is to say if there is no overlap between their language objects. In practice, these are functions nos. 1 and 2.
The table below summarizes the properties of reciprocal exclusions.

	Funct. No. 1	Funct. No. 2	Funct. No. 3
Funct. No. 1	-	Non exclusive (1)	Exclusive
Funct. No. 2	Non exclusive (1)	-	Exclusive
Funct. No. 3	Exclusive	Exclusive	-

(1) When special functions 1 and 2 are used simultaneously, the time calculation of special function number 1, %IDxy.i.11, can only be accessed in a Fast or Mast task, on capture done %Ixy.i.2=1.

Procedure

The table below summarizes the procedure for configuring the special functions of the TSX CTY2C module.

Step	Action
1	Access the module configuration screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Special functions** zone, enter the number of the special function in one of the **Num** fields.  Enter the number of the second special function, if necessary. A zero means that no function has been selected.
4	Complete the **Parameter** field if necessary, for special function number 3 (See *Special function number 3 (TSX CTY 2C), p. 94*) for example.
5	Special function(s) configuration is complete. If there are no more parameters to configure, enable the new configuration. To do this: ● scroll down the **Edit** menu and select the **Enable** command.

Adjustment of the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C

5

At a Glance

Subject of this chapter

This chapter describes the different adjustment options of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules.

What's in this Chapter?

This chapter contains the following topics:

Description of the adjustment screen of a counting module

General

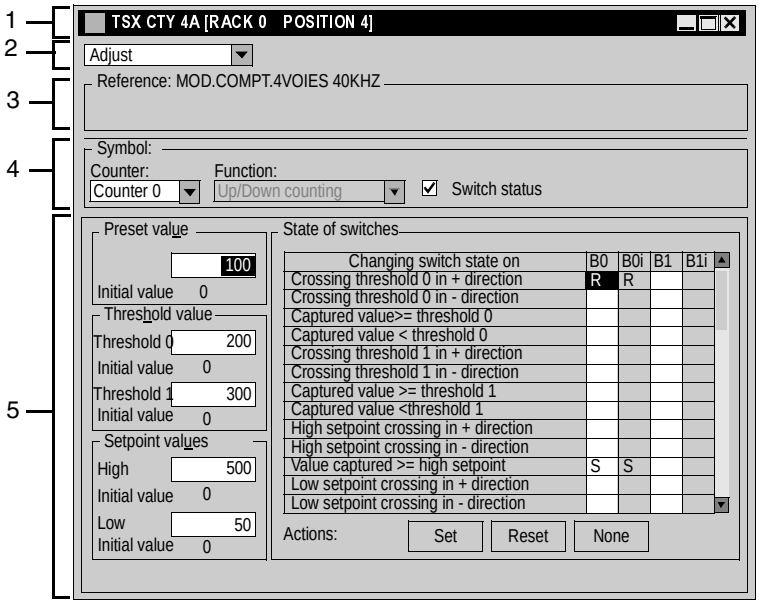
The adjustment screen displays adjustment parameters for the module and is used to modify them in both offline and on-line mode.
It also gives access to configuration and debugging screens.
The structure of the adjustment screen is very similar to that of the configuration screen.

Note: consult the Common task functions (See: PL7 Junior, Pro; Application-specific functions volume 1) section to see how to configure, adjust and debug an application.

Note: the adjustment screen is a graphic tool for facilitating an application development. Unlike in configuration, it is possible to program adjustments by using language objects directly.

Illustration

The diagram below shows an example of an adjustment screen.



Description

The table below shows the different elements of the adjustment screen and their functions.

Address	Element	Function
1	Title bar	Shows the selected module's reference and geographical address (rack number and physical position in the rack).
2	Command zone	Shows the screen in progress (**Adjustment**) and is used to switch to or switch back to other screens: • **Configuration,** • **Debugging** (or diagnostics), can only be accessed in on-line mode.
3	**Module** zone	Shows the abbreviated name of the module.
4	**Channel** zone	Is used to select the channel to be adjusted and the heading for setting switch parameters: • **Symbol:** channel name set by the user (using the variables editor). • **Counter:** counter number, i.e. of the module channel. • **Function:** calls up the configured counting function. This heading is fixed. • **Switch status:** box to be checked if you wish to set latch change of status parameters. Otherwise, the heading does not appear in the adjustment field.
5	Adjustment field	This field contains various headings to be completed (parameter values), displayed according to the counting function selected.

Current value and initial value

For each heading **two** values appear:
- the entered value which can be modified, in one window,
- and the **initial** value which cannot be directly modified.

The behavior of these values depends on the PLC connection mode.
- **In offline mode:** after enabling (**Edit** → **Enable** command), the entered value becomes the initial value and appears in the corresponding field. On transfer the initial value becomes the current value.
- **In on-line mode:** after enabling, the entered value becomes the current value. The **Edit** → **Save adjustment parameters** command is used to copy the current value in the initial value.

Note: the initial value is the one that the parameter concerned takes when the PLC starts from cold.

How to adjust the offset value of an absolute encoder

At a Glance

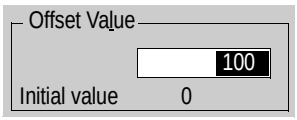
This parametering is an option when the input interface of the TSX CTY 2C module is configured for a series output absolute encoder.
This parameter is used to perform a shift from zero by adding the offset value to the current value supplied by the encoder.
Parameters are set in the **Offset value** zone in the adjustment screen.

For encoders with parallel outputs

For multiplexed encoders with parallel outputs, the offsets must be managed by the application (See *How to program multiplexing of absolute encoders with parallel outputs, p. 113*).

Procedure

The table below summarizes the procedure for adjusting the offset value.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Offset value** zone enter the value desired.  This value must be between: **0** and the **rollover** value (this mode is implicit for an absolute encoder interface).
4	Offset value adjustment is complete. ● If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Note: in offline mode, the value entered is then copied again in the **Initial value** field. In on-line mode, the value entered then becomes the current value.

How to adjust the preset value

At a Glance

This parameter defines the preset value (in down counting only or up/down counting) i.e. the value reloaded in the counting register after:

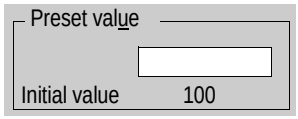
- a hardware or software preset command, especially after an invalid measurement,
- or automatic preset when switching to zero.

This parameter exists when the input is configured for counting pulses or an incremental encoder.

Parameters are set in the **Preset value** zone in the adjustment screen.

Procedure

The table below summarizes the procedure for adjusting the preset value.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Preset value** zone enter the required value.  This value must be between: ● -16 777 216 and +16 777 215 in normal mode, ● 0 and the rollover value if this mode is configured (TSX CTY 2C).
4	Reset value adjustment is complete. ● If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Note: in offline mode, the value entered is then copied again in the **Initial value** field. In on-line mode, the value entered then becomes the current value.

How to adjust channel fault processing (TSX CTY 2C)

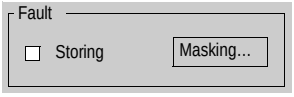
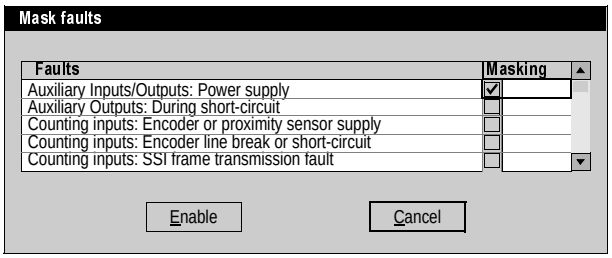
At a Glance

The TSX CTY 2C module has two independent channel fault processing mechanisms which can be configured by the user:

- **Storing** enables the application to detect the occurrence of a fault temporary or otherwise.
- **Masking** certain faults enables the application to continue operating in downgraded mode (safeguards activated).

Procedure

The table below summarizes the procedure for configuring fault processing for a TSX CTY 2C module:

Step	Action
1	Access the adjustment screen of the TSX CTY2C module required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	Under the **Faults** heading, check the **Recording** box if required or leave blank. 
4	Click on the **Mask** button. The following dialog box appears: 
5	Check the boxes corresponding to the faults which are to be masked. The **masked** information is displayed next to each checked box.
6	Enable the dialog box.
7	Fault processing adjustment is complete. • If there are no more parameters to be defined, enable the new configuration with the command **Edit** → **Enable**.

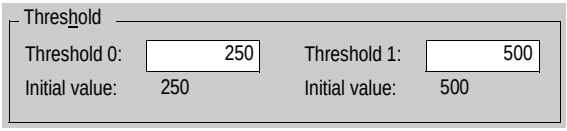
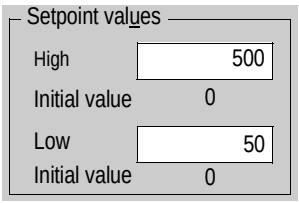
How to adjust threshold and set point values

At a Glance

These parameters define threshold values 0 and 1 and high and low setpoints. These objects have very similar roles in comparisons.
Parameters are set in the **Threshold value** and **Setpoint value** zones in the adjustment screen.

Procedure

The table below summarizes the procedure for adjusting threshold values.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Threshold values** zone enter the required values.  These values must be between: ● -16 777 216 and +16 777 215 in normal mode, ● 0 and the rollover value if this mode is configured or implicit (TSX CTY 2C, absolute encoder interface). **Note:** the respective values of **threshold 0** and **threshold 1** do not have to be in this order.
4	In the **Set point value** zone enter the value(s) required; when they are defined (depending on the module and the counting function configured).  These values must be between: -16 777 216 and +16 777 215.
5	Threshold and set point value adjustment is complete. ● If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Note: in offline mode the value entered is then copied in the **Initial value** fields. In on-line mode the values entered then become the current values.

How to adjust the speed measurement and monitoring function

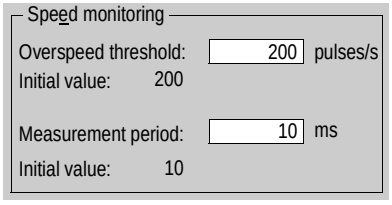
At a Glance

This parametering is offered with the TSX CTY 2C module.
The **Speed monitoring** zone is used to define:

- the overspeed threshold value,
- the speed measurement period.

Procedure

The table below summarizes the procedure for adjusting the speed measurement and monitoring function.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	In the **Speed monitoring** zone enter the required overspeed threshold value.  This value must be between: **1** and **4 000 000 pulses/s**. The value 0 inhibits the overspeed monitoring.
4	Enter the value of the measurement period according to the estimated pulse frequency and the desired precision or response time (See *Speed Monitoring Function (TSX CTY 2C), p. 89*). This value must be between: **10** and **30 000 ms**.
5	Speed reading function and monitoring adjustment is complete. ● If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Note: in offline mode, the value entered is then copied again in the **Initial value** field. In on-line mode, the value entered then becomes the current value.

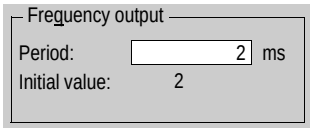
How to adjust the frequency output period (TSX CTY 2C)

At a Glance

The TSX CTY 2C module has a **Q3** output which can be programmed in automatic mode to generate an adjustable period signal, destined for various uses (for example: synchronization of several channels or modules).
Output configuration (automatic or manual) is controlled by the application, but it is possible to modify it temporarily in the Debugging (See *Debugging the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C, p. 133*) screen.
Parameters for the signal period are set under the **Frequency output** heading in the adjustment screen.

Procedure

The table below summarizes the procedure for adjusting the frequency output signal period.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	Under the **Frequency output** heading enter the required value.  This value must be between: **0** and **4 000 000 ms**, per 1 ms step.
4	The **Q3** output period adjustment is complete. • If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Note: in offline mode, the value entered is then copied again in the **Initial value** field. In on-line mode, the value entered then becomes the current value.

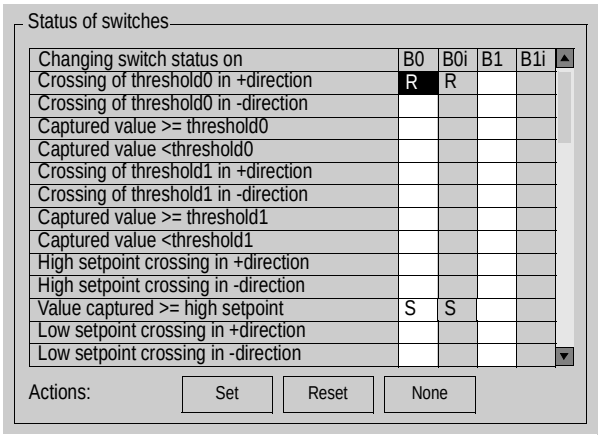
How to adjust the switch change of status conditions

At a Glance

Switch 0 and 1 change of status condition parameters can be set from the adjustment screen.
There are a great number of these conditions and they depend on the module and function configured. Priorities are defined in the *Description of switches associated with counting modules, p. 65* section.

Procedure

The table below summarizes the procedure for adjusting the change of status conditions of switches 0 and 1.

Step	Action
1	Access the module adjustment screen required.
2	In the **Channel** zone select or enable the counter selection (i.e. the channel) involved and its function.
3	Check the **Switch status** box. A heading with the same name appears in the adjustment zone. 
4	Under the heading **Switch status**: ● select the box corresponding to the condition and the switch (0 or 1) concerned, ● then click on the **SET** button (set to 1), the **RESET** button (set to 0), or the **NONE** button (delete an existing value). **Note:** the conditions are **not** shown in order of priority.
5	Adjust the other change of status conditions in the same way.
6	Adjustment of the switch change of status conditions is complete. ● If there are no more parameters to be defined, enable the new adjustment with the **Edit** → **Enable** command. Notes: in offline mode the values entered are then copied into the initial status fields **B0i** and **B1i**. Priority rules also apply to initial values. In on-line mode the values entered then become the current status values.

Debugging the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C

6

At a Glance

Subject of this chapter

This chapter describes the different debugging options of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules.

What's in this chapter?

This chapter contains the following topics:

Introducing the debugging screens

General

Debugging screens are used to debug an application. They are used to display the status of inputs and outputs on a channel, the contents of the registers, the possible faults and to control language objects (setting to 0 or 1, forcing or unforcing a bit, etc.) They can therefore only be accessed in on-line mode.
They also give access to adjustment (See *Adjustment of the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C, p. 121*) and configuration (See *Configuration of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules, p. 101*) screens.
There are two debugging screens:

- a **minimized screen**, which is the application operation monitoring tool for the counting or measurement channel. It displays the main data: contents of registers, status of inputs and outputs, fault indicators.
- an **maximized screen**, which is the debugging tool. It is used to view and control language objects.

Switching from one to the other is immediate, and there is no need to stop the application or counting in progress.
Opening a module in on-line mode displays the minimized debugging screen by default.

Note: the general principles for debugging an application are described in the Common task functions (See: PL7 Junior, Pro; Application-specific functions volume 1) section.

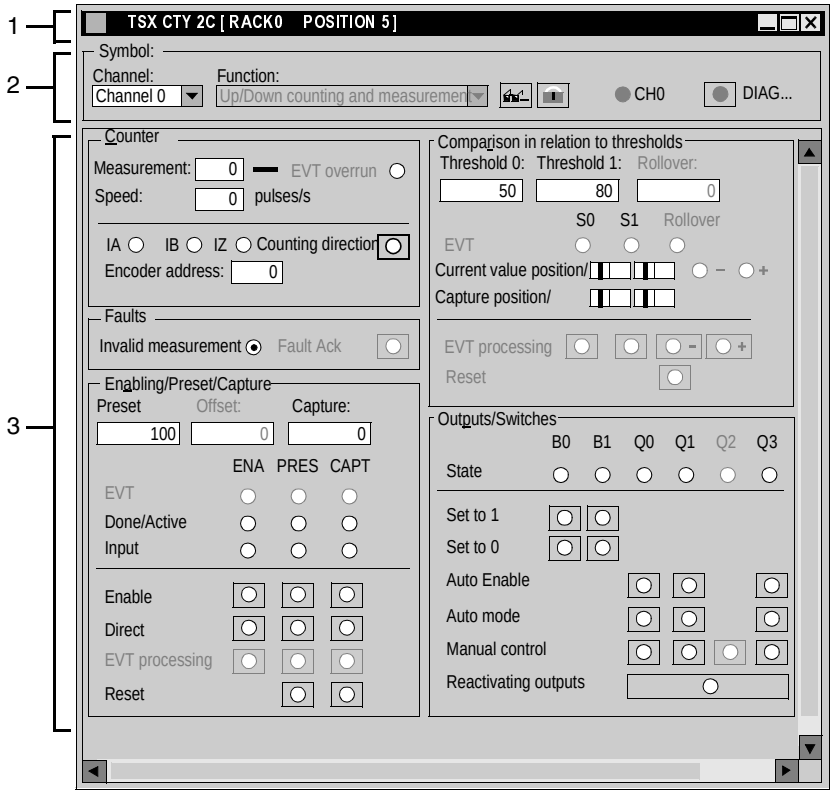
Description of the maximized debugging screen

At a Glance

The maximized debugging screen is used to provisionally modify the operation of an application at counting module level, with a view to detecting programming faults. It displays the status of inputs, outputs and main channel bits, contents of registers, and possible faults. It is used to control or force (lock) certain bits.
You return to the minimized screen simply by clicking on an icon.
The minimized screen does not include a default **module** zone. However, you can display one with the **View** menu. This **module** zone also gives access to the adjustment (See *Description of the configuration screen for a counting module, p. 102*) and configuration (See *Description of the adjustment screen of a counting module, p. 122*) screens.

Illustration

The diagram below shows an example of a maximized debugging screen.



Description

The table below shows the different elements of the maximized debugging screen and their functions.

Address	Element	Function
1	Title bar	Shows the selected module's reference and its geographical address (rack number and physical position in the rack).
2	**Channel** field	Is used to select the counting channel to be debugged in the **Channel** or **Counter** window. The symbol (name) of the channel and its function are shown. In the same zone are: ● the button to access  the minimized screen, ● the unforce button  for forced bits, ● a copy of the **CHx** channel indication LED, ● and the button to access  the channel diagnostics window, the icon ⚠ signaling the presence of masked faults (CTY2C).
3	View and command field	This field displays the status of inputs, outputs, intermediate bits and the values of the various counting in progress registers. It is also used to control and force various objects (bits). The field is divided into function groups corresponding to the main functions (See *Description of the standard functions of the TSX CTY•• counting modules, p. 31*).

Note: it is possible to display the concise definition of an object that is not selected by keeping the mouse cursor over it. The definition also provides the name of the associated language object.

Note: the LEDs and commands that are not available appear grayed out.

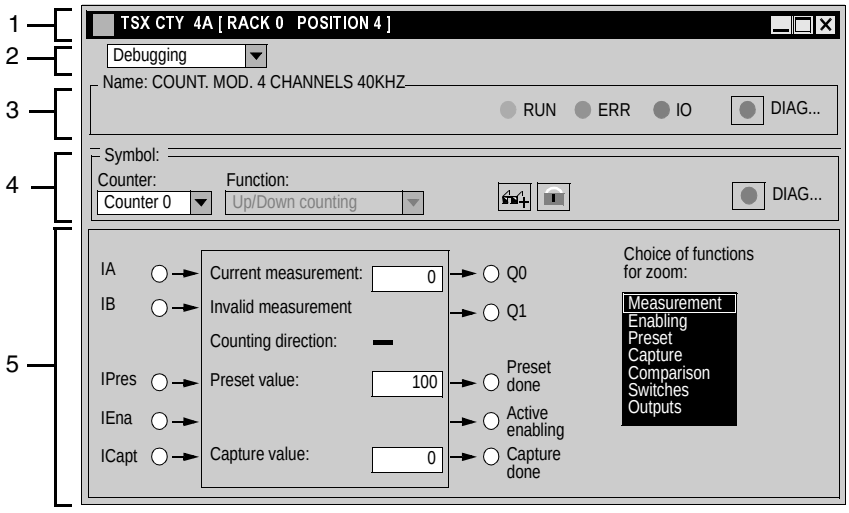
Description of the minimized debugging screen

At a Glance

The minimized debugging screen is used to monitor the operation of an application at counting module level. It displays the status of inputs, outputs and main bits of a channel, the contents of registers, possible faults.
To switch to the maximized screen, just click on an icon. It also gives access to the adjustment and configuration screens.

Illustration

The figure below shows an example of a minimized debugging screen.



Description

The table below shows the different minimized debugging screen elements and their functions.

Address	Element	Function
1	Title bar	Shows the selected module's reference and its geographical address (rack number and physical position in the rack).
2	Command zone	Shows the mode in progress (Debugging) and is used to access other modes: • **Adjusting** (See *Description of the adjustment screen of a counting module, p. 122*) • **Configuration** (See *Description of the adjustment screen of a counting module, p. 122*).

Address	Element	Function
3	**Module** zone	Shows the module's abbreviated name and its **version number** (this is the only screen where this information is displayed). In the same zone are: ● a copy of the module status LEDs: **RUN** (in operation), **ERR** (error), **I/O** (external fault), ● and the access button DIAG... to the module diagnostics (See *Displaying fault diagnostics, p. 144*) window.
4	**Channel** field	Is used to select the counting channel to be debugged in the **Channel** or **Counter** window. The following information is shown at this level: ● **Symbol:** channel name set by the user (using the variables editor). ● **Function:** counting function configured. This selection is defined in the configuration screen. In the same zone are: ● the access button for the extended screen (See *Description of the maximized debugging screen, p. 135*) (zoom), where the debugging commands are located, Note: with CTY2A/4A modules only, the functions to be displayed in the extended screen can be selected in a window in the parameter field, lower down. For the CTY2C module, all functions will be displayed systematically. ● the unforce button for inputs/outputs. ● a copy of the **CHx** channel fault LED, ● and the button to access DIAG... the channel diagnostics window, the icon ⚠ signaling the presence of masked faults (CTY2C).
5	Parameters in progress zone	This zone displays the status of the inputs and outputs and the various counting parameters in progress. If the contents of the counting register cannot be used following an input fault, the **Invalid measurement** indicator or LED appear in red.

Note: the concise definition of an object that is not selected, but which can be activated, is displayed by keeping the mouse cursor over it. The definition also gives the name of the associated language object.

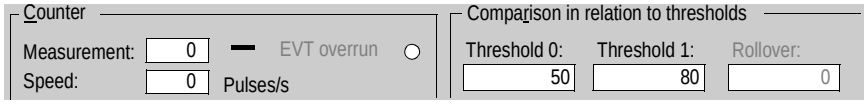
How to use value or parameter windows

At a Glance

The maximized debugging screen commands zone is divided into headings or function groups. The corresponding functions are described in detail in the *Description of the standard functions of the TSX CTY •• counting modules, p. 31* section, and their essential characteristics are shown in the module configuration (See *Configuration of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules, p. 101*) procedures.
This document module shows the general principles for using measurement or parameter windows.
A **measurement** is the result of counting, acquisition or calculation.
A **parameter** is a piece of input data by the user or application.

Principle for using measurement windows

Windows as shown in the example below are used to display the current contents of a register. The values displayed cannot be modified directly by selecting these windows.



The following table summarizes the behavior of values that cannot be modified:

Value	Behavior
Measurement	Cannot be modified. Can be preset by a preset or reset command. The + or - sign on the right of the window indicates the actual direction measurement change.
Speed (CTY 2C)	Cannot be modified or preset.
Address of multiplexed encoder (CTY 2C)	Cannot be modified. Can be preset using the **manual Q2** and **Q3** outputs if these control the multiplexing.

To modify a parameter value, such as threshold, preset, or rollover values, use the following procedure.

How to change a parameter value

The table below describes the procedure for changing a parameter.

Step	Action
1	In the **Rollover** zone select the **Adjustment** mode
2	In the adjustment screen modify the required parameter.
3	Enable (**Edit** - **Enable** menu), otherwise say **Yes** in the **Enable modifications** dialog box, which appears at the next step.
4	In the **Rollover** zone select the **Debugging** mode again. **Result:** the new parameter value is displayed.

How to use the LEDs and buttons

At a Glance

The maximized debugging screen commands zone is divided into headings or function groups. The corresponding functions are described in detail in the *Description of the standard functions of the TSX CTY •• counting modules, p. 31* section, and their essential characteristics are shown in the module configuration (See *Configuration of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C modules, p. 101*) procedures.
This section shows the general principles for using LEDs and buttons.
The LEDs are used to display the status of a bit.
The buttons are used to set a bit, triggering an action or not (depending on the case).

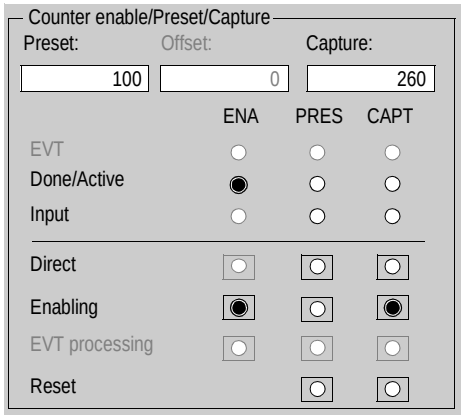
Note: it is possible to display the concise definition of an object that is not selected by keeping the mouse cursor over it. The definition also provides the name of the associated language object.

Meaning of the LEDs

When the bit is at 0, the LED is off (white) (see the example below).
When the bit is at 1, the LED is on (black, blue or red). Some LEDs display a little black dot in status 1.
There are also position LEDs which use a bold dash to display the measurement or capture position in relation to a threshold or a set point. These latter are represented by a thin central dash.

Button use principle

In the example below the **Direct enabling** and **Direct capture** buttons are set to 1.



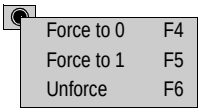
Clicking on this button sets the associated bit to 1. The button and possibly the LED above light up (switch to a color).

Clicking on this button sets the associated bit to 0. The button and possibly the LED above go out (switch to white).
These actions are provisional in the sense that the application or counting can change the bit status.

Note: the LED status can be different from the button status. Some LEDs are reserved for storing the action carried out by the physical input (capture, preset or reset). These LEDs can be recognized by the presence of a reset button (action carried out) in the same column. In this case you can switch off the LED, if necessary, by pressing this button.

Forcing a bit

To lock the status of a bit (i.e. to make it permanent), use the forcing commands, which can be accessed by right-clicking on the mouse. The button goes on or off in



the same way as before but with the letter **F** superimposed . Conversely, use the same menu to delete bit forcing.

You can also unforce all forced bits by clicking on the **padlock** icon in the **Channel** field.

Diagnostics of the data modules TSX CTY 2A, TSX CTY 4A and TSX CTY 2C

7

At a Glance

Subject of this chapter

This chapter describes the different diagnostics options of the TSX CTY 2A, TSX CTY 4A and TSX CTY 2C data modules.

What's in this Chapter?

This chapter contains the following topics:

Displaying fault diagnostics

At a Glance

Diagnostic screens at module or channel level can only be accessed in on-line mode. When a **unmasked** fault appears, this is indicated:

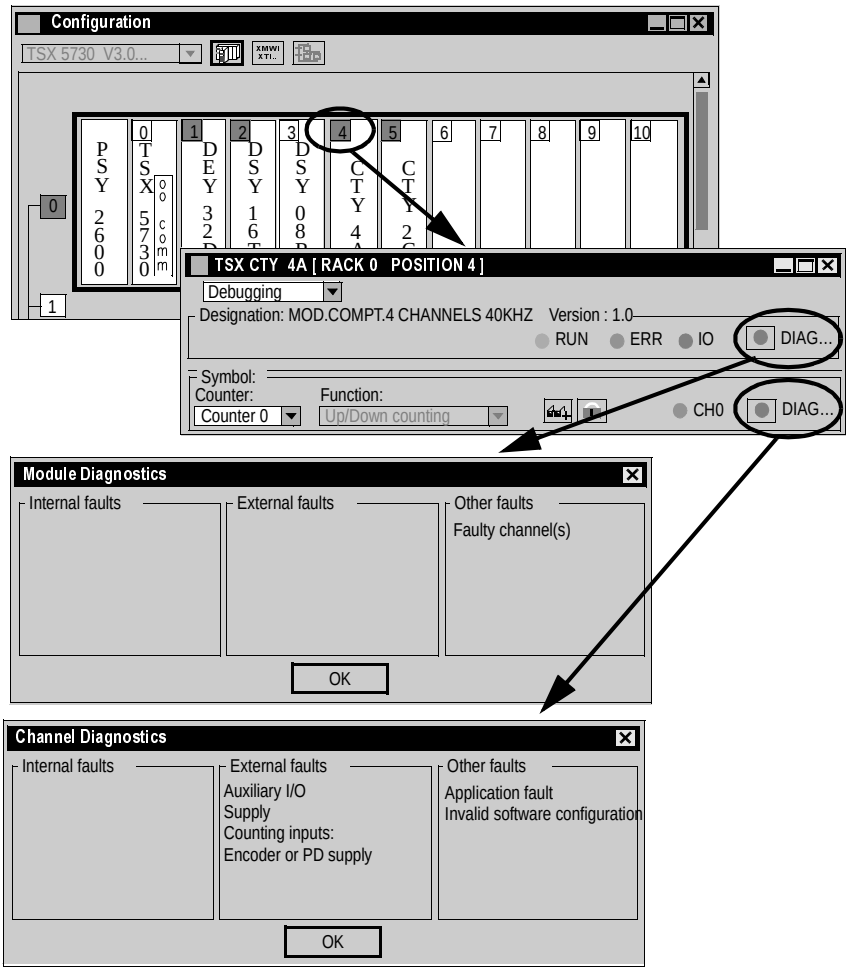
- in the rack configuration screen by the appearance of a red square in the position of the counting module that is faulty,
- in the debug screen by the DIAG buttons at module or channel level which turn red. Clicking on these buttons then gives access to the fault diagnostics.

The fault is also indicated:

- on the module via the central display,
- by the dedicated language objects: **%Ixy.i.ERR**, **%Ixy.i.MOD.ERR**, **%MWxy.i.MOD.2**, etc., and the status words (See *The counting task's language objects, p. 157*).

Note: even if the fault is masked (TSX CTY 2C), it is indicated by the flashing of the **CHx** light and by the DIAG button at channel level.

Illustration The diagram below shows how the faults are indicated.



Lists of fault diagnostics

At a Glance

The messages displayed on the diagnostics screens are used to assist in application debugging. These messages must be concise and are sometimes ambiguous (as different faults may have the same consequences).
These diagnostics are on two levels: module and channel, the latter being the most explicit. The lists below show the message headings with suggestions for finding out what is wrong.

List of the module fault messages

The table below provides a list of the module fault messages.

Fault indicated	Possible interpretation and/or action
Module failure	The module has a fault. Check the module mounting. Change the module.
Faulty channel(s)	One or more channels have a fault. Refer to channel diagnostics.
Self-test	The module is running a self-test. Wait until the self-test is complete.
Different hardware and software configurations	There is a lack of compatibility between the module configured and the module in the rack. Make the configuration and the software configuration compatible.
Module is missing or off	Install the module. Fasten the mounting screws.

List of channel fault messages

The table below gives the list of fault messages at channel level.

Fault indicated. Other consequences.	**Possible interpretation and/or action.**
External fault or counting input fault: ● encoder or proximity sensor supply fault, ● line break or short circuit of at least one encoder differential signal (1A, 1B or 1Z), ● SSI series frame fault, ● specific fault on absolute encoder. outputs are set to 0 in automatic mode. **Invalid measurement** message.	Check the sensor connections. Check the sensor power supply. Check the sensor operation. Delete the fault and acknowledge if the fault storing is configured (CTY 2C). Counting pulses or incremental encoder: preset or reset to acknowledge the Invalid measurement message.
Counting application fault: ● measurement overrun, ● overspeed. Outputs are set to 0 in automatic mode. **Invalid measurement** message.	Diagnose the fault more precisely (external causes). Check the application again, if necessary. Delete the fault and acknowledge if the fault storing is configured (CTY 2C). Counting pulses or incremental encoder: preset or reset to acknowledge the **Invalid measurement** message.
Auxiliary input/output fault: ● power supply, ● short circuit of at least one output. Outputs are set to 0 in automatic mode.	Check the output connections. Check the input/output power supply (24 V). Diagnose the fault more precisely (external causes). Delete the fault and acknowledge if the fault storing is configured (CTY 2C).
Internal fault or channel self-test: ● faulty module, ● module missing or off, ● module running self-test.	Module fault has gone down to channel level. Refer to module level diagnostics.
Different hardware and software configurations	Module fault has gone down to channel level. Refer to module level diagnostics.
Invalid software configuration: ● incorrect constant, ● bit combination not associated with any configuration.	Check and modify the configuration constants.
Communication fault	Check the connections between the racks.
Application fault: refusal to configure or adjust	Diagnose the fault more precisely.

Software implementation

8

At a glance

Subject of this chapter

This chapter describes the counting modules' operating modes as well as the operation of the event processing, which supports completion of the counting applications with optimized response times.

What's in this Chapter?

This chapter contains the following sections:

8.1 Counting module operating modes

How counting modules behave in the different operating modes

General

The counting modules operate in specific ways according to the different PLC operating modes. Knowing about these specific ways is important for programming and debugging the application.
General principles for dealing with operating modes are described:

- in the set up manual for Premium - TSX 57 PLCs, (TSX DM 57 xx E),
- the reference manual (See: PL7 Junior, Pro; Reference manual), PL7 Micro/ Junior/Pro (TLX DR PL7 xxE).

Summary table

The table below shows the special features of counting modules in different operating modes.

	Cold start	**Warm restart**	**STOP**	**Reconfiguring in online mode**
Invalid measurement bit	1	1	0	1
Adjustment parameters: set points, thresholds, preset, measurement period (speed), output period (frequency) %MDxy.i.r	initial values	unchanged	unchanged	initial values
Commands (direct software actions, various actions, event unmasking) %Qxy.i.r, %QWxy.i.r	0	unchanged	unchanged, new commands not sent	unchanged
Information and module data -> processor %IDxy.i.r, %IWxy.i.r, %Ixy.i.r	unchanged if no power outage	unchanged if no power outage	sent	unchanged
Forced objects	unforced	unchanged	unchanged	unchanged
Switches (as a result of **Invalid measurement**)	0	0	unchanged	0
Outputs	0	0	fallback mode	0

The paragraphs below show the essential properties of these operating modes, seen from a module.

Cold start	A **cold** start occurs when the application first starts running, when some power is restored, when initialization takes place from a PL7, or when the Reset button on the processor is pressed. Parameters are initialized at their set initial values using the configuration editor. The current counter measurement cannot be used (**Invalid measurement** bit set to 1). If the module has not had a power outage, the current value of the counter is unchanged although the **Invalid measurement** bit is set. The user must set the procedure to be carried out from a cold start (See *How to deal with an invalid measurement, p. 97*).
Warm restart	The program starts again from the program element where the power outage took place, but the outputs remain at 0 until refreshed by the task. The values of counting application-specific function objects are not changed by a warm restart, except for those involved with the **Invalid measurement** bit. If the module has not had a power outage, the current value of the counter is unchanged although the **Invalid measurement** bit is set. The user must define the processing procedure to be carried out on a warm restart.
Power outage and restart	When there is a power outage, the application context and time of the outage are stored. When the power is restored the saved context is compared with that in progress: • if the application context has changed (i.e. loss of system context or new application), the PLC initializes the application: see **cold start**, • if the application context is the same, the PLC carries out a **warm restart**.
STOP mode	In STOP mode the user program is not run. However the application-specific counting function does operate: the counter advances according to the status of the physical inputs (**IA**, **IB**, **IPres** or **IReset**, **IEna**, **ICapt**).
Reconfiguring in online mode	This mainly involves debugging an application. Changes must be confirmed.

8.2 Event processing installation

At a Glance

Subject of this section

This section describes how to install event processing associated with a counting module.

What's in this Section?

This section contains the following topics:

Introduction to event processing

At a Glance

Event processing is used to minimize the reaction time when installing counting modules by:

- programming reflex actions,
- extending (physical) reflex output time performances **Q0** and **Q1** to other outputs on the PLC output modules.

One event (task) process can be associated with each counting channel. The appearance of an event in the application-specific counting function re-routes the application program to the event task associated with the channel.

The priority of the task is linked to its number. There are two priority levels determined by the process number: EVT0 has priority over all other EVTi's (i: from 1 to 31 or 63, depending on processor type). You must therefore assign EVT0 to the most important channel of the application, which will not necessarily be a counting channel.

The **Multi-task application structure** section in the Premium - TSX 57 PLCs (TSX DM 57 xx E) installation manual provides details on how to integrate event tasks into the application.

Event processing principle

Event processing is enabled when:

- bit %S38 which enables PL7 event processing is set to 1,
- the UNMASKEVT is carried out in MAST and FAST tasks,
- the counting channel events involved are unmasked.

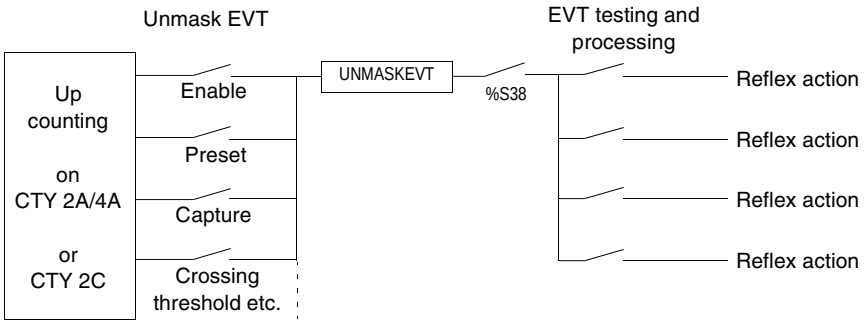
The objects indicating:

- the source of the event (event status word %IWxy.i.3),
- and the captured value,

are updated implicitly before the event process is carried out. The other counting objects are not updated.

Event processing must first of all identify the source of the event by testing the event status word bits set to 1.

Illustration The figure below illustrates the event processing principle



How to program event processing

Procedure

The table below summarizes the essential steps for programming event processing.

Step	Action
1	**Configuration phase** In offline mode, in the configuration editor, select **Event processing** and the event number for the counting channel.
2	**Unmasking phase** In particular, the **MAST** or **FAST** calling task must: ● enable processing of events at system level: set bit %S38 to 1 (default value), ● unmask events in MAST and FAST tasks with the UNMASKEVT instruction (active by default), ● unmask events concerned at channel level by setting implicit event unmasking language objects (See *Unmask event commands, word %QWxy.i.1, p. 164*). By default, the events are masked, ● check that the stack of events at system level is not saturated (bit %S39 must be at 0).
3	**Event program creation phase** On the Events tab, select **Edit** → **Create** and create the event program. In particular, this program must: ● Determine the source of the event(s) from the event status word (See *Status of events and switches, word %IWx.i.3, p. 162*) in implicit exchange. ● Carry out the reflex tasks associated with the event. This process must be as short as possible. ● Explicitly update the reflex outputs concerned. **Note**: the event status word is automatically reset to zero.

Illustration of event unmasking

This figure shows event unmasking in the MAST task.

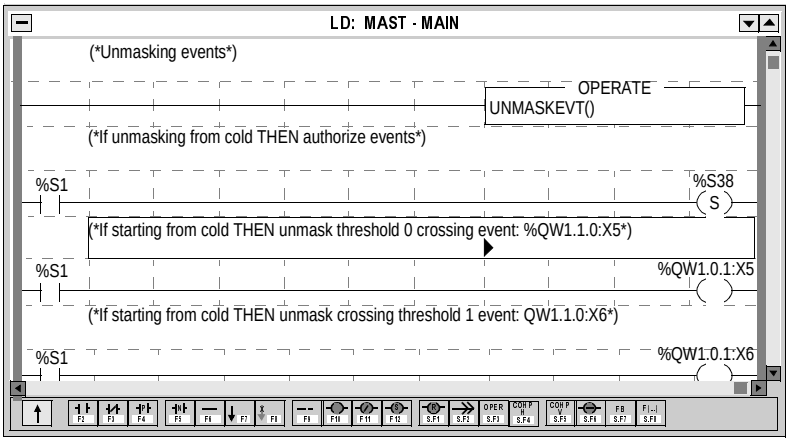
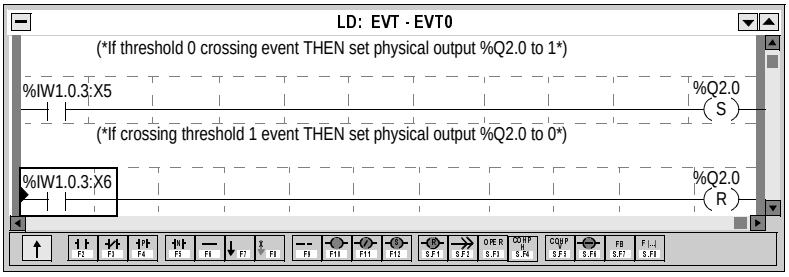


Illustration of the contents of an event task

This figure shows the possible contents of an event task (testing event and action bit).



The counting task's language objects

9

At a glance

Subject of this chapter

This chapter describes the language objects associated to the counting tasks as well as the different ways of using them.

What's in this Chapter?

This chapter contains the following topics:

Introducing language objects for application-specific counting

General

Counting modules configured in a given position automatically generate a group of language objects enabling them to be programmed, and measurement results and diagnostics to be read.
There are two main types of language object:

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

The former (implicit exchanges) involve input/output module maps: measurement results, software information and commands necessary for the operation.
The latter (explicit exchanges) are used to set module parameters and bring in additional arguments (parameters, commands and information) for advanced programming. They are not necessary for normal programming.

Implicit exchange language objects associated with the application-specific counting function

At a Glance

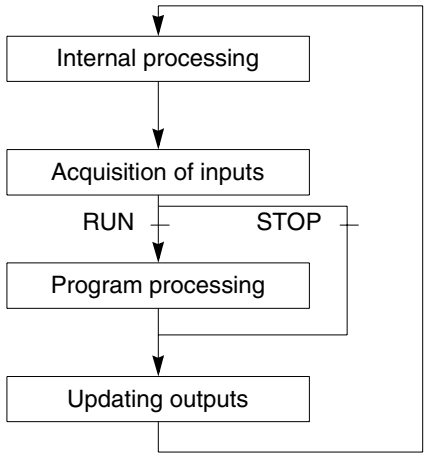
These objects are used to access inputs and software information for the application-specific counting function.
The addressing system for words and bits is given in the Common application-specific functions (See: PL7 Junior, Pro; Application-specific functions volume 1) section.

Reminders

Module input maps (%I and %IW) are updated in the processor at the beginning of the task cycle in RUN or STOP mode.
Output commands (%Q and %QW) are updated in the module at the end of the task cycle, only when in RUN mode.
xy represents the position of the module, **i** represents the channel number (counted from 0) in the module.

Illustration

The chart below illustrates the operation cycle relating to a PLC task (cyclic running).



List of implicit exchange objects

The table below summarizes the implicit exchange objects in the application-specific counting function.

Object	Content
%Ixy.i.ERR	Counting channel error bit. When set to 1 it shows that channel i of the module in position xy has a fault. The causes of the fault are listed in the explicit exchange words %MWxy.i.2 (See *Channel standard fault, %MWxy.i.2, p. 168*) and %MWxy.i.3 (See *Specific channel faults, %MWxy.i.3, p. 169*).
%Ixy.MOD.ERR	Module error bit. When set to 1 it shows that the module in position xy has a fault. The causes of the fault are listed in the explicit exchange module status word %MWxy.MOD.2 (See *Module error flags: %MWxy.MOD.2, p. 168*).
%IDxy.i.0	Current counter value (24 active bits).
%IDxy.i.4	Captured counter value (24 active bits).
%IDxy.i.6	Reserved for special functions (CTY 2C), this is the time elapsed (event task) between capture and last pulse (special function number 1).
%IDxy.i.8	Speed (number of pulses per second, 24 active bits, CTY 2C).
%IDxy.i.10	Multiplexed address of the absolute encoder with parallel outputs (CTY 2C).
%IDxy.i.11	Reserved for special functions (CTY 2C), this is the time elapsed (Mast or Fast task) between capture made and last pulse detected (special function number 1).
%Ixy.i.0 to **%Ixy.i.23**	Software information bits. (See *Software information: %Ixy.i.r bits, p. 161*)
%IWxy.i.2	Software information word showing the status of the physical counting inputs (See *Status of physical inputs/outputs, word %IWxy.i.2, p. 162*).
%IWxy.i.3	Information word showing the source of the events and the state of the software output switches (See *Status of events and switches, word %IWx.i.3, p. 162*).
%Qxy.i.0 to **%Qxy.i.23**	Software commands (See *Software commands, %Qxy.i.r bits, p. 163*).
%QWxy.i.0	Stored information reset commands, software direction commands (See *Reset and output commands, word %QWxy.i.0, p. 164*).
%QWxy.i.1	Unmask event (See *Unmask event commands, word %QWxy.i.1, p. 164*) commands.

Details of implicit exchange objects

At a Glance

This section describes the group of implicit exchange bits. The columns in the tables give the following information:

- its address,
- its symbol generated by the PL7 during Pre-symbolization,
- the description of its function.

Software information: %Ixy.i.r bits

The table below shows the meanings of the %Ixy.i.r status bits.

Address	Standard symbol	Meaning:
%Ixy.i.0	ENAB_ACTIV	Counter enable active
%Ixy.i.1	PRES_DONE	Preset done (hardware, except special function CTY 2C)
%Ixy.i.2	CAPT_DONE	Capture done (hardware, except special function CTY 2C)
%Ixy.i.4	OVERSPEED_WRN	Overspeed fault (CTY 2C)
%Ixy.i.5	CUR_MEAS_THR0	Current value more than or equal to threshold 0 (<or less than 0 when down counting, CTY 2A/4A)
%Ixy.i.6	CUR_MEAS_THR1	Current value more than or equal to threshold 1
%Ixy.i.7	CUR_MEAS_HISP	Current value more than or equal to the high setpoint (CTY 2A/4A)
%Ixy.i.8	CUR_MEAS_LOSP	Current value more than or equal to the low setpoint (CTY 2A/4A)
%Ixy.i.9	COUNT_DIR	Counting direction 0: direction - (down counting), 1: direction + (up counting)
%Ixy.i.10	CAPT_THR0	Captured value more than or equal to threshold 0
%Ixy.i.11	CAPT_THR1	Captured value more than or equal to threshold 1
%Ixy.i.12	CAPT_HISP INC_MOD_DONE	• Captured value more than or equal to the high setpoint (CTY 2A/4A) • Modulo crossing in + direction (CTY 2C)
%Ixy.i.13	CAPT_LOSP DEC_MOD_DONE	• Captured value more than or equal to the low setpoint (CTY 2A/4A) • Rollover moving in - direction (CTY 2C)
%Ixy.i.16		Reserved for special functions (CTY 2C): %Ixy.i.16: Correct speed,
%Ixy.i.17		Reserved for special functions (CTY 2C): %Ixy.i.17: moving part stationary.

Status of physical inputs/ outputs, word %IWxy.i.2

The table below shows the meanings of %IWxy.i.2 status word bits.

Address	Standard symbol	Meaning
%IWxy.i.2:X0	ST_IA	Status of physical counting input **IA**
%IWxy.i.2:X1	ST_IB	Status of physical counting input **IB**
%IWxy.i.2:X2	ST_IVAL	Status of physical enable input **IEna**
%IWxy.i.2:X3	ST_PRES	State of the **IPres** or **IReset** physical preset input
%IWxy.i.2:X4	ST_CAPT	State of physical capture input **ICapt**
%IWxy.i.2:X6	ST_IZ	Status of physical counting input **IZ**
%IWxy.i.2:X7	INVALID_MEAS	Invalid measurement
%IWxy.i.2:X8	ST1_SSI_FRAME	• Rank 1 status bit from the SSI frame, • or odd parity bit (SSI absolute encoder with odd parity, not checked by the module), • or least significant part of the address (absolute encoder with multiplexed parallel outputs and adaptation base unit).
%IWxy.i.2:X9	ST2_SSI_FRAME	• Rank 2 status bit from the SSI frame, • or most significant part of the address (absolute encoder with multiplexed parallel outputs and adaptation base unit).
%IWxy.i.2:X10	ST3_SSI_FRAME	• Rank 3 status bit from the SSI frame, • or specific fault bit on absolute encoder with parallel outputs.
%IWxy.i.2:X11	ST4_SSI_FRAME	Rank 4 status bit 4 from the SSI frame
%IWxy.i.2:X12	ST_Q2	**Q2** output status (CTY 2C)
%IWxy.i.2:X13	ST_Q3	**Q3** output status (CTY 2C)
%IWxy.i.2:X14	ST_Q0	**Q0** output status
%IWxy.i.2:X15	ST_Q1	**Q1** output status

Status of events and switches, word %IWx.i.3

The table below shows the meanings of %IWxy.i.3 status word bits.

Address	Standard symbol	Meaning
%IWxy.i.3:X0	ENAB_EVT	Enabling event
%IWxy.i.3:X1	PRES_EVT	Event preset or reset
%IWxy.i.3:X2	CAPT_EVT	Capture event
%IWxy.i.3:X3	CAPT_EDGE	Capture edge direction (TSX CTY 2C) 0: rising edge, 1: falling edge.
%IWxy.i.3:X5	THR0_EVT	Threshold 0 crossing event (or zero value when down counting with TSX CTY 2A/4A)

Address	Standard symbol	Meaning
%IWxy.i.3:X6	THR1_EVT	Threshold 1 crossing event
%IWxy.i.3:X7	HISP_EVT	• Crossing high setpoint event (CTY 2A/4A) • Reserved for special functions (CTY 2C)
%IWxy.i.3:X8	LOSP_EVT	• Crossing low setpoint event (CTY 2A/4A) • Reserved for special functions (CTY2C)
%IWxy.i.3:X9	ST_COUNT_DIR	Direction when crossing threshold or setpoint (CTY 2A/4A) 0: direction - (down counting), 1: direction + (up counting)
%IWxy.i.3:X10	ST_LATCH0	State of switch 0
%IWxy.i.3:X11	ST_LATCH1	State of switch 1
%IWxy.i.3:X12	INC_MOD_EVT	Rollover moving event in + direction (CTY 2C)
%IWxy.i.3:X13	DEC_MOD_EVT	Rollover moving event in - direction (CTY 2C)
%IWxy.i.3:X15	OVERRUN_EVT	Overrun events (channel level).

Software commands, %Qxy.i.r bits

The table below shows the meanings of the %Qxy.i.r command bits.

Address	Standard symbol	Meaning
%Qxy.i.0	DIRENAB	Direct enabling by the software
%Qxy.i.1	DIRPRES	Direct preset by the software
%Qxy.i.2	DIRCAPT	Direct capture by the software
%Qxy.i.3	FLT_ACK	Fault acknowledgement (CTY 2C)
%Qxy.i.5	ENAB_IENAB	Enabling the physical input **IEna**
%Qxy.i.6	ENAB_IPRES	Enabling the **IPres** or **IReset** physical output
%Qxy.i.7	ENAB_ICAPT	Enabling the physical input **ICapt**
%Qxy.i.9	ENAB_Q3_AUTO	Enabling the **Q3** output in automatic mode (CTY 2C)
%Qxy.i.10	SET_LATCH0	Switch 0 set to 1
%Qxy.i.11	SET_LATCH1	Switch 1 set to 1
%Qxy.i.12	RESET_LATCH0	Switch 0 set to 0
%Qxy.i.13	RESET_LATCH1	Switch 1 set to 0
%Qxy.i.14	ENAB_Q0_AUTO	Enabling the **Q0** output in automatic mode
%Qxy.i.15	ENAB_Q1_AUTO	Enabling the **Q1** output in automatic mode
%Qxy.i.16 to %Qxy.i.19		Reserved for special functions (CTY 2C)
%Qxy.i.20	MANU_CMD_Q2	Q2 output manual command (CTY 2C)

Address	Standard symbol	Meaning
%Qxy.i.21	MANU_CMD_Q3	Q3 output manual command (CTY 2C)

Reset and output commands, word %QWxy.i.0

The table below shows the meanings of the %QWxy.i.0 command word bits.

Address	Standard symbol	Meaning
%QWxy.i.0:X1	PRES_RESET	Reset of hardware preset done
%QWxy.i.0:X2	CAPT_RESET	Hardware reset capture done
%QWxy.i.0:X4	MOD_RESET	Rollover moving reset done (CTY 2C)
%QWxy.i.0:X9	COUNT_DIR_CHG	Counting direction 0: direction - (down counting), 1: direction + (up counting)
%QWxy.i.0:X10	REACTIV_Q	Reactivation of outputs **Q0**, **Q1**, and outputs **Q2**, **Q3** (CTY 2C)
%QWxy.i.0:X11	AUTO_MOD_Q3	Manual/automatic output mode **Q3** (frequency, CTY 2C) 0: manual, 1: automatic (programmable frequency)
%QWxy.i.0:X12	AUTO_MOD_Q0	Manual/automatic output mode **Q0** 0: manual, 1: automatic
%QWxy.i.0:X13	AUTO_MOD_Q1	Manual/automatic output mode **Q1** 0: manual, 1: automatic
%QWxy.i.0:X14	MANU_CMD_Q0	Output status manual command **Q0**
%QWxy.i.0:X15	MANU_CMD_Q1	Output status manual command **Q1**

Unmask event commands, word %QWxy.i.1

The table below shows the meanings of word bits %QWx.i.1.

Address	Standard symbol	Meaning
%QWxy.i.1:X0	ENAB_UNMSK	Enable event unmasking
%QWxy.i.1:X1	PRES_UNMSK	Unmask preset or reset event
%QWxy.i.1:X2	CAPT_UNMSK	Unmask capture event
%QWxy.i.1:X5	THR0_UNMSK	Unmask threshold 0 event
%QWxy.i.1:X6	THR1_UNMSK	Unmask threshold 0 event
%QWxy.i.1:X7	HISP_UNMSK	Unmask high setpoint event (CTY 2A/4A)
%QWxy.i.1:X8	LOSP_UNMSK	Unmask low setpoint event (CTY 2A/4A)
%QWxy.i.1:X12	INC_MOD_UNMSK	Unmasking rollover moving event in + direction (CTY 2C)
%QWxy.i.1:X13	DEC_MOD_UNMSK	Unmask rollover moving event in - direction (CTY 2C)

Explicit exchange language objects associated with the application-specific counting function

At a Glance

Explicit exchanges are exchanges carried out on request from the program user using the following instructions:

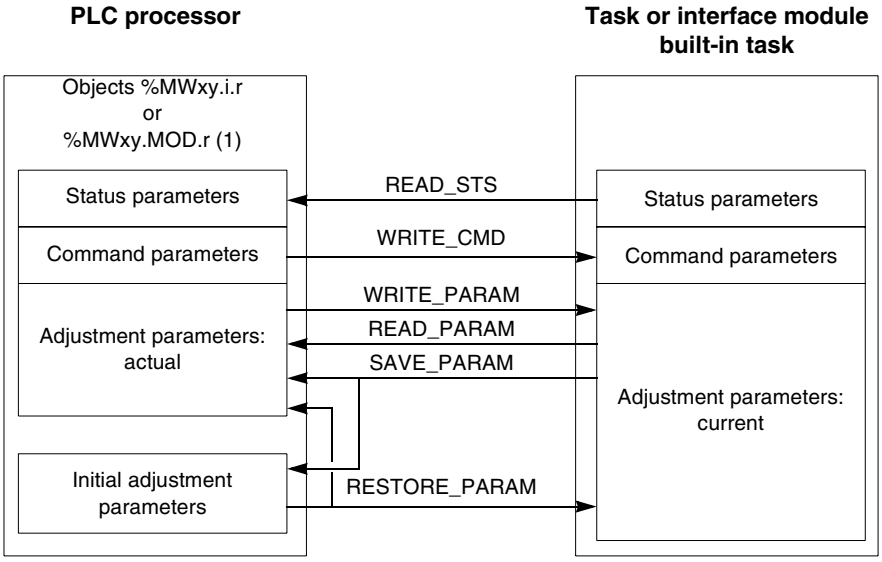
- READ_STS (See: PL7 Junior, Pro; Application-specific functions volume 1) (read status words),
- WRITE_CMD (See: PL7 Junior, Pro; Application-specific functions volume 1) (write command words),
- WRITE_PARAM (See: PL7 Junior, Pro; Application-specific functions volume 1) (write adjustment parameters),
- READ_PARAM (See: PL7 Junior, Pro; Application-specific functions volume 1) (read adjustment parameters),
- SAVE_PARAM (See: PL7 Junior, Pro; Application-specific functions volume 1) (save adjustment parameters),
- RESTORE_PARAM (See: PL7 Junior, Pro; Application-specific functions volume 1) (restore adjustment parameters).

These exchanges apply to all %MW objects that are of the same type (status, commands or parameters) on the same channel.

Note: these objects bring information (e.g.: Type of channel fault…), commands (e.g.: switch commands) and additional operating modes (saving and restoring adjustment parameters while the application is in progress) to carry out more precise operation programming of the function.

General principle for the use of explicit instructions

The diagram below shows the different types of explicit exchange that are possible between the PLC processor and the module (or the built-in interface).



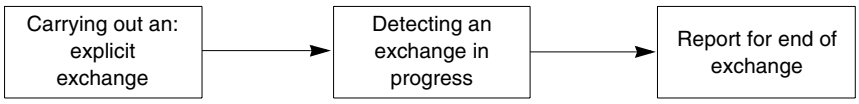
(1) Only with READ_STS and WRITE_CMD instructions.

Managing exchanges

During an explicit exchange, it may be interesting to monitor the progress that this makes, in order to, for example, only take into account the data read when the exchange has actually been carried out.
For this, two kinds of information are available:

- detecting an exchange in progress (See *Explicit exchange operation indicators: %MWxy.i.0, p. 175*),
- an end of exchange report (See *Explicit exchange report: %MWxy.i.1, p. 175*).

The diagram below describes the principle for managing an exchange.



List of explicit exchange objects

The table below summarizes the explicit exchange objects in the application-specific counting function.

Object	Content	Type
%MWxy.i.MOD.2	Module fault status word (updated by the READ_STS%CHxy.MOD instruction)	State
%MDxy.i.4	Preset value	Parameter
%MDxy.i.6	Value of threshold 0	Parameter
%MDxy.i.8	Value of threshold 1	Parameter
%MDxy.i.10	• High setpoint value (CTY 2A/4A) • Absolute encoder offset value (CTY 2C)	Parameter
%MDxy.i.12	• Low setpoint value (CTY 2A/4A) • Overspeed threshold (CTY 2C)	Parameter
%MDxy.i.22	Frequency output period **Q3** (CTY 2C)	Parameter
%MDxy.i.24	Target speed (special function number 3, CTY 2C)	Parameter
%MWxy.i.2 and **3**	Counting channel fault diagnostics **i** of the module in position **xy** (See *Channel standard fault, %MWxy.i.2, p. 168*)	State
%MWxy.i.14 and **15**	Switch 0 SET conditions (See *0 counter output SET commands, %MWxy.i.14, p. 169*)	Parameter
%MWxy.i.16 and **17**	Switch 0 RESET conditions (See *0 counter output RESET commands, %MWxy.i.16, p. 170*)	Parameter
%MWxy.i.18 and **19**	Switch 1 SET conditions (See *1 counter output SET commands, %MWxy.i.18, p. 171*)	Parameter
%MWxy.i.20 and **21**	Switch 1 RESET conditions (See *1 counter output RESET commands, %MWxy.i.20, p. 172*)	Parameter
%MWxy.i.26	Stopping speed (special function number 3, CTY2C)	Parameter
%MWxy.i.27	Speed measurement period (CTY2C)	Parameter
%MWxy.i.28	Reserved for special functions (CTY2C)	Parameter

Breakdown of objects for user defined exchange

At a glance

This part collates the word type objects for user defined exchange, whose bits have a particular meaning. This objects are introduced in detail below, bit by bit.

Observations

- A bit's meaning extends to **State 1**. The non Boolean bits (neither true nor false) are detailed to avoid any ambiguity.
- None of the bits are used.

Summary

- xy shows the module's position.
- i shows the channel number (from 0) in the module.

Module error flags: %MWxy.MOD.2

The table below introduces the meanings of the module state word's bits. These bits do not have standard symbols and are read by a READ_STS %CHxy.MOD.

Address	Meaning
%MWxy.MOD.2:X0	Broken down module
%MWxy.MOD.2:X1	Faulty channel(s)
%MWxy.MOD.2:X5	Different hardware and software configurations
%MWxy.MOD.2:X6	Module is absent or switched off

Channel standard fault, %MWxy.i.2

The table below introduces the meanings of the CH_FLT status word bits. Reading can be done via a READ_STS %CHxy.i.

Address	Standard symbol	Meaning
%MWxy.i.2:X0	COUNT_INP_FLT	Counting input error
%MWxy.i.2:X1	COUNT_APP_FLT	Counting application fault
%MWxy.i.2:X3	AUX_IO_FLT	Auxiliary I/O
%MWxy.i.2:X4	INTERNAL_FLT	Internal error or channel self-testing
%MWxy.i.2:X5	CONF_FLT	Different hardware and software configurations
%MWxy.i.2:X6	COMMUNIC_FLT	Communication error with the PLC
%MWxy.i.2:X7	APPLI_FLT	Application fault
%MWxy.i.2:X8 and %MWxy.i.2:X9	CH_LED0,1	Green indicator's command bits **CHx** (channel state) 00 = off, 01 = blinking, 11 = lit
%MWxy.i.2:X11	AUX_SUPPL_FLT	Auxiliary I/O supply fault
%MWxy.i.2:X12	SHORT_CIRC_FLT	Output short-circuit error
%MWxy.i.2:X13	ENC_SUPPLY_FLT	Supply, encoder or potential difference error
%MWxy.i.2:X14	LINE_BRK_FLT	Encoder line break or short-circuit error

Address	Standard symbol	Meaning
%MWxy.i.2:X15	SSI_TRAME_FLT	Transmission of SSI absolute encoder frame (CTY 2C) error

Specific channel faults, %MWxy.i.3

The table below introduces the meanings of the %MWxy.i.3 channel status word bits. Reading can be done via a READ_STS %CHxy.i.

Address	Standard symbol	Meaning
%MWxy.i.3:X0	CH_CONF_FLT	Channel configuration error
%MWxy.i.3:X1	XERFLOW_FLT	Measurement overrun error
%MWxy.i.3:X2	ABS_ERROR_FLT	Specific absolute encoder error
%MWxy.i.3:X3	OVERSPEED_FLT	Overspeed error

0 counter output SET commands, %MWxy.i.14

The table below introduces the meanings of the %MWxy.i.14 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 0 counter output to 1
%MWxy.i.14:X0	S0_ENAB	on confirmation
%MWxy.i.14:X1	S0_PRES	on preset
%MWxy.i.14:X2	S0_CAPT	on capture
%MWxy.i.14:X4	S0_MOD_INC	on +direction modulo overshoot
%MWxy.i.14:X5	S0_MOD_DEC	on -direction modulo overshoot

0 counter output SET commands, %MWxy.i.15

The table below introduces the meanings of the %MWxy.i.15 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 0 counter output to 1
%MWxy.i.15:X0	S0_TH0_INC	on +direction 0 threshold overshoot
%MWxy.i.15:X1	S0_TH0_DEC	on -direction 0 threshold overshoot
%MWxy.i.15:X2	S0_C_SUP_TH0	if the captured value is more than or equal to 0
%MWxy.i.15:X3	S0_C_INF_TH0	if the captured value is less than the 0 threshold
%MWxy.i.15:X4	S0_TH1_INC	on +direction 1 threshold overshoot
%MWxy.i.15:X5	S0_TH1_DEC	on -direction 1 threshold overshoot
%MWxy.i.15:X6	S0_C_SUP_TH1	if the captured value is more than or equal to 1
%MWxy.i.15:X7	S0_C_INF_TH1	if the captured value is less than the 1 threshold
%MWxy.i.15:X8	S0_HISP_INC	on high setpoint crossing

Address	Standard symbol	Meaning: Setting the 0 counter output to 1
%MWxy.i.15:X9	S0_HISP_DEC	on high setpoint crossing
%MWxy.i.15:X10	S0_C_SUP_HISP	if the captured value is more than or equal to the high setpoint
%MWxy.i.15:X12	S0_LOSP_INC	on +direction low crossing setpoint
%MWxy.i.15:X13	S0_LOSP_DEC	on -direction low crossing setpoint
%MWxy.i.15:X15	S0_C_INF_LOSP	if the captured value is less than the low setpoint

0 counter output RESET commands, %MWxy.i.16

The table below introduces the meanings of the %MWxy.i.16 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 0 counter output to 0
%MWxy.i.16:X0	R0_ENAB	on confirmation
%MWxy.i.16:X1	R0_PRES	on preset
%MWxy.i.16:X2	R0_CAPT	on capture
%MWxy.i.16:X4	R0_MOD_INC	on +direction modulo overshoot
%MWxy.i.16:X5	R0_MOD_DEC	on -direction modulo overshoot

0 counter output RESET commands, %MWxy.i.17

The table below introduces the meanings of the %MWxy.i.17 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 0 counter output to 0...
%MWxy.i.17:X0	R0_TH0_INC	on +direction 0 threshold overshoot
%MWxy.i.17:X1	R0_TH0_DEC	on -direction 0 threshold overshoot
%MWxy.i.17:X2	R0_C_SUP_TH0	if the captured value is more than or equal to the 0 threshold
%MWxy.i.17:X3	R0_C_INF_TH0	if the captured value is less than the 0 threshold
%MWxy.i.14:X4	R0_TH1_INC	on +direction 1 threshold overshoot
%MWxy.i.17:X5	R0_TH1_DEC	on -direction 1 threshold overshoot
%MWxy.i.17:X6	R0_C_SUP_TH1	if the captured value is more than or equal to the 1 threshold
%MWxy.i.17:X7	R0_C_INF_TH1	if the captured value is less than the 1 threshold
%MWxy.i.17:X8	R0_HISP_INC	on high setpoint crossing
%MWxy.i.17:X9	R0_HISP_DEC	on high setpoint crossing
%MWxy.i.17:X10	R0_C_SUP_HISP	if the captured value is more than or equal to the high setpoint

Address	Standard symbol	Meaning: Setting the 0 counter output to 0…
%MWxy.i.17:X12	R0_LOSP_INC	on +direction low crossing setpoint
%MWxy.i.17:X13	R0_LOSP_DEC	on -direction low crossing setpoint
%MWxy.i.17:X15	R0_C_INF_LOSP	if the captured value is less than the low setpoint

1 counter output SET commands, %MWxy.i.18

The table below introduces the meanings of the %MWxy.i.18 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 1 counter output to 1…
%MWxy.i.18:X0	S1_ENAB	on confirmation
%MWxy.i.18:X1	S1_PRES	on preset
%MWxy.i.18:X2	S1_CAPT	on capture
%MWxy.i.18:X4	S1_MOD_INC	on +direction modulo overshoot
%MWxy.i.18:X5	S1_MOD_DEC	on -direction modulo overshoot

1 counter output SET commands, %MWxy.i.19

The table below introduces the meanings of the %MWxy.i.19 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning of setting the 1 counter output to 1…
%MWxy.i.19:X0	S1_TH0_INC	on +direction 0 threshold overshoot
%MWxy.i.19:X1	S1_TH0_DEC	on -direction 0 threshold overshoot
%MWxy.i.19:X2	S1_C_SUP_TH0	if the captured value is more than or equal to the 0 threshold
%MWxy.i.19:X3	S1_C_INF_TH0	if the captured value is less than the 0 threshold
%MWxy.i.19:X4	S1_TH1_INC	on +direction 1 threshold overshoot
%MWxy.i.19:X5	S1_TH1_DEC	on -direction 1 threshold overshoot
%MWxy.i.19:X6	S1_C_SUP_TH1	if the captured value is more than or equal to the 1 threshold
%MWxy.i.19:X7	S1_C_INF_TH1	if the captured value is less than the 1 threshold
%MWxy.i.19:X8	S1_HISP_INC	on high setpoint crossing
%MWxy.i.19:X9	S1_HISP_DEC	on high setpoint crossing
%MWxy.i.19:X10	S1_C_SUP_HISP	if the captured value is more than or equal to the high setpoint
%MWxy.i.19:X12	S1_LOSP_INC	on +direction low crossing setpoint
%MWxy.i.19:X13	S1_LOSP_DEC	on -direction low crossing setpoint
%MWxy.i.19:X15	S1_C_INF_LOSP	if the captured value is less than the low setpoint

1 counter output RESET commands, %MWxy.i.20

The table below introduces the meanings of the %MWxy.i.20 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 1 counter output to 0…
%MWxy.i.20:X0	R1_ENAB	on confirmation
%MWxy.i.20:X1	R1_PRES	on preset
%MWxy.i.20:X2	R1_CAPT	on capture
%MWxy.i.20:X4	R1_MOD_INC	on +direction modulo overshoot
%MWxy.i.20:X5	R1_MOD_DEC	on -direction modulo overshoot

1 counter output RESET commands, %MWxy.i.21

The table below introduces the meanings of the %MWxy.i.21 command word bits. The requests used are those associated to the (READ_PARAM, WRITE_PARAM...) parameters.

Address	Standard symbol	Meaning: Setting the 1 counter output to 0…
%MWxy.i.21:X0	R1_TH0_INC	on +direction 0 threshold overshoot
%MWxy.i.21:X1	R1_TH0_DEC	on -direction 0 threshold overshoot
%MWxy.i.21:X2	R1_C_SUP_TH0	if the captured value is more than or equal to the 0 threshold
%MWxy.i.21:X3	R1_C_INF_TH0	if the captured value is less than the 0 threshold
%MWxy.i.21:X4	R1_TH1_INC	on +direction 1 threshold overshoot
%MWxy.i.21:X5	R1_TH1_DEC	on -direction 1 threshold overshoot
%MWxy.i.21:X6	R1_C_SUP_TH1	if the captured value is more than or equal to the 1 threshold
%MWxy.i.21:X7	R1_C_INF_TH1	if the captured value is less than the 1 threshold
%MWxy.i.21:X8	R1_HISP_INC	on high setpoint crossing
%MWxy.i.21:X9	R1_HISP_DEC	on high setpoint crossing
%MWxy.i.21:X10	R1_C_SUP_HISP	if the captured value is more than or equal to the high setpoint
%MWxy.i.21:X12	R1_LOSP_INC	on +direction low crossing setpoint
%MWxy.i.21:X13	R1_LOSP_DEC	on -direction low crossing setpoint
%MWxy.i.21:X15	R1_C_INF_LOSP	if the captured value is less than the low setpoint

Explicit object exchange and report management

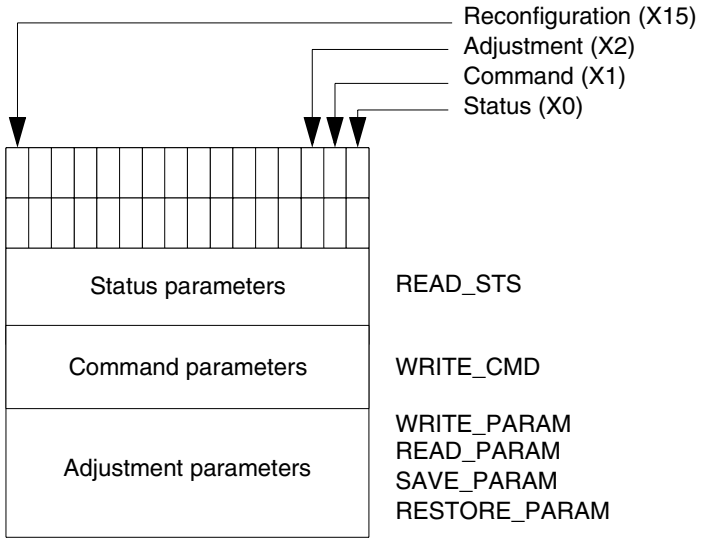
At a Glance

When the data is exchanged between the PLC memory and the module, the considerations taken by the data module may require the running of several task cycles. To manage the exchanges, two words are used:

- %MWxy.i: Exchange in progress,
- %MWxy.i.1: Report.

Illustration

The illustration below shows the various data bits significant to exchange management.



Description of the significant data bits

Each of the %MWxy.i and %MWxy.i.1 word bits are associated with a parameter:

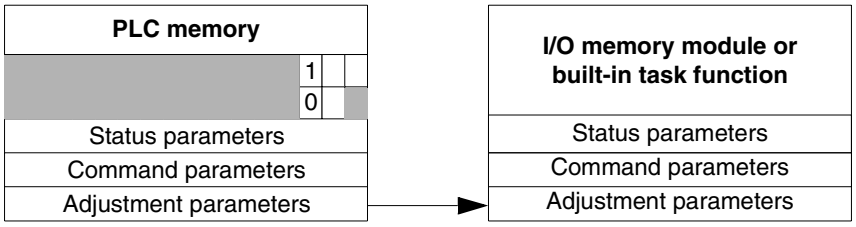
- the bits in position 0 are associated with the status parameters:
  - the %MWxy.i.X0 bit shows whether a status word read request is in progress,
- the bits in position 1 are associated with the control parameters:
  - the %MWxy.i.0:X1 bit shows whether the control parameters are sent to channel i of the module,
  - the %MWxy.i.1:X1 bit specifies whether the control parameters are accepted by channel i of the module,
- the bits in position 2 are associated with the adjustment parameters:
  - the MWxy.i.0:X2 bit indicates whether the adjustment parameters have been exchanged with channel i of the module (by WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM),

- the %MWxy.i.1:X2 bit specifies whether the adjustment parameters are accepted by the module. If the exchange has progressed correctly, the bit changes to 0,
- the bits in position 15 show a reconfiguration on channel i of the module from the console (modification of configuration parameters + channel cold start).

Note: the exchange and report words also exist at module level (%MWxy.MOD and %MWxy.MOD.1).

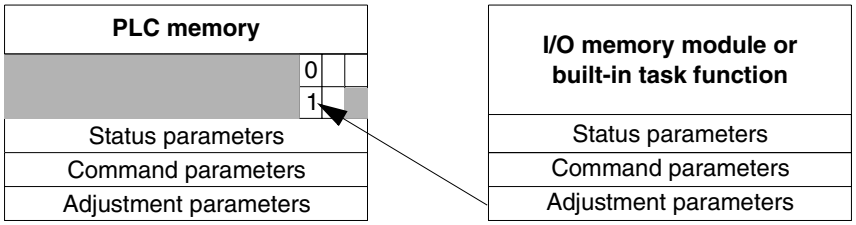
Example

Phase 1: transmitting data with the help of the WRITE_PARAM instruction.



When the instruction is scanned by the PLC processor, the **Exchange in progress** bit is set to 1 in %MWxy.

Phase 2: Analysis of data by the I/O and report module



When the data is exchanged between the PLC memory and the module, the data module acknowledgement is managed by the %MWxy.i.1:X2 bit: Report (0 = correct exchange, 1 = unsuccessful exchange).

Note: there are no adjustment parameters at the module level.

Explicit exchange operation indicators: %MWxy.i.0

The table below shows the meaning of channel exchange control bits EXCH_STS.

Bit	Standard symbol	Meaning
0	STS_IN_PROGR	Status word exchange in the channel in progress
1	COMMAND_IN_PROGR	Command word exchange in progress
2	ADJUST_IN_PROGR	Adjustment word exchange (parameters) in progress
15	RECONF_IN_PROGR	Channel reconfiguration in progress

Explicit exchange report: %MWxy.i.1

The table below shows the meaning of report bits EXCH_ERR.

Bit	Standard symbol	Meaning
0	STS_READ_ERR	Failure when reading channel status
1	COMMAND_ERR	Failure when sending a command word
2	ADJUST_ERR	Failure when sending an adjustment word
15	RECONF_ERR	Failure when reconfiguring the channel

Example of the counting application

10

At a glance

Subject of this chapter

This chapter introduces an example of the counting application, from the configuration of the modules to the application's development in the PLC.

What's in this Chapter?

This chapter contains the following sections:

10.1 Introducing and configuring the example

At a Glance

Subject of this section

This section shows an example of the counting application and describes how the hardware should be configured for the program to be used.

What's in this Section?

This section contains the following topics:

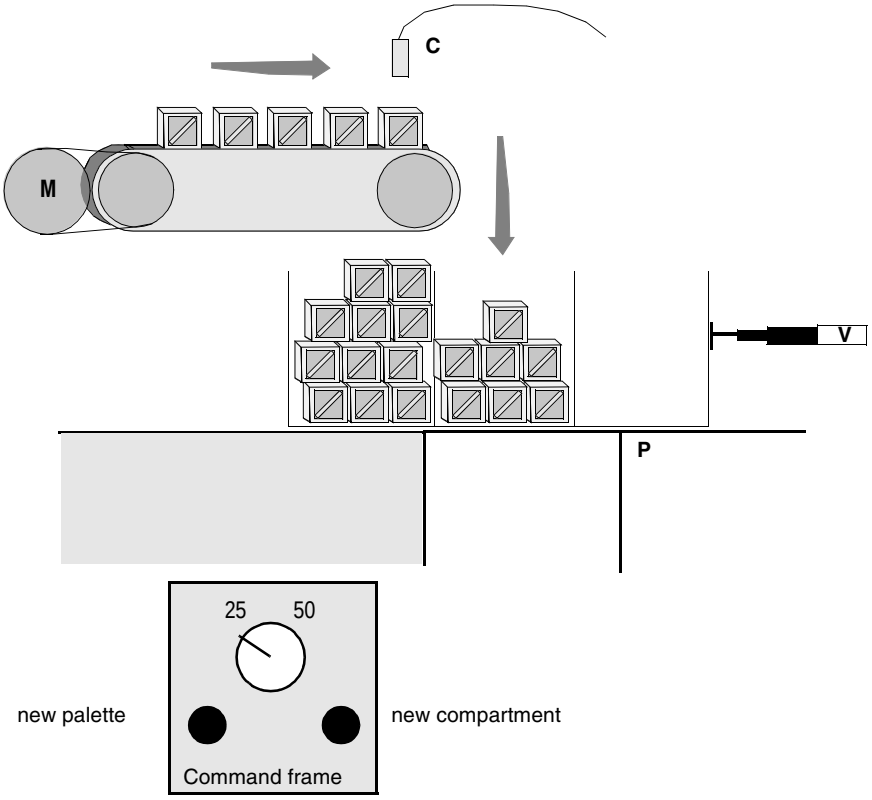
Introduction to an example of an application

At a glance

This very simple example shows a TSX CTY 2A counting module being used in tandem with a discrete module. It consists of a MAST task and an event task.

Illustration

The illustration above shows the principle of monitoring automated palette stacking.



The application's external specifications

The PLC is loaded to monitor the palette stacking. Each palette has three compartments. The compartments are put into place beneath the stacking device using a **V** jack. They are put into 4 positions, with the last one for moving out full palettes.
There are two types of palettes, one with 25 and one with 50 compartments.
The operating principle is as follows:

- the **M** motor powers a conveyor belt which carries the objects.
- a **C** sensor counts the objects before they are put back into the palette.
- a telescopic V jack is used to move on to the next compartment when the one in progress is full, and it moves out the full palette.
- a **P** plate allows you to change the palette.

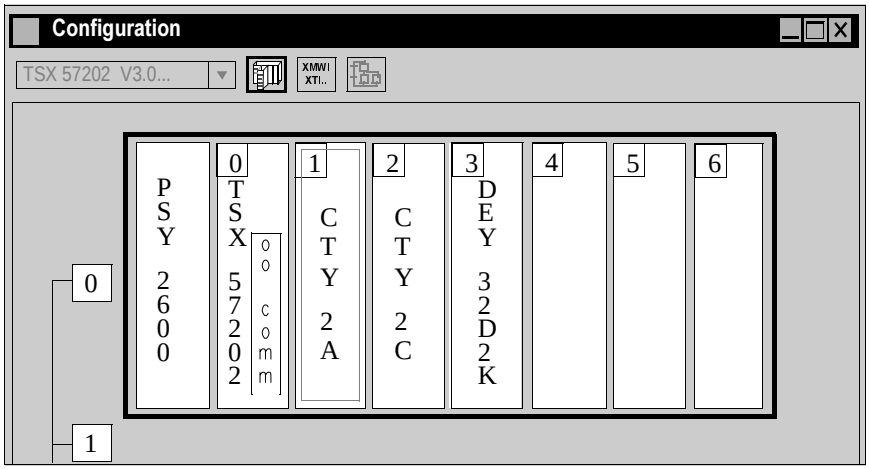
Using the motor brings about the physical confirmation of the counting.
The dialog operator consists of the following controls:

- 25/50 switch: allows you to choose the type of palette. This takes effect for the next palette (on state),
- **new palette** button: forces the palette to be changed (on the rising edge),
- **new compartment** button: forces the compartment to be changed (on the rising edge).

Configuration of the PLC

The PLC structure

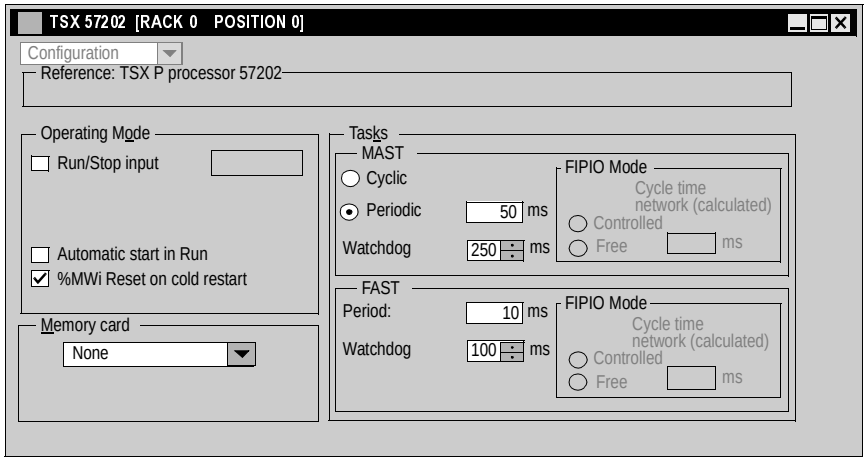
The diagram below shows the PLC's constitution.



Note: the CTY 2C module in position 2 is not used in this example.

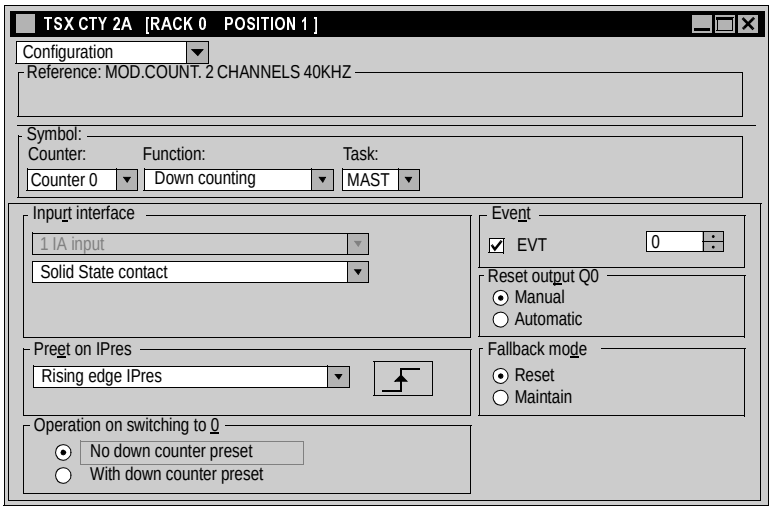
Configuration of the processor

The diagram below shows the configuration of the TSX 57202 processor used in the example.



The counting module's configuration

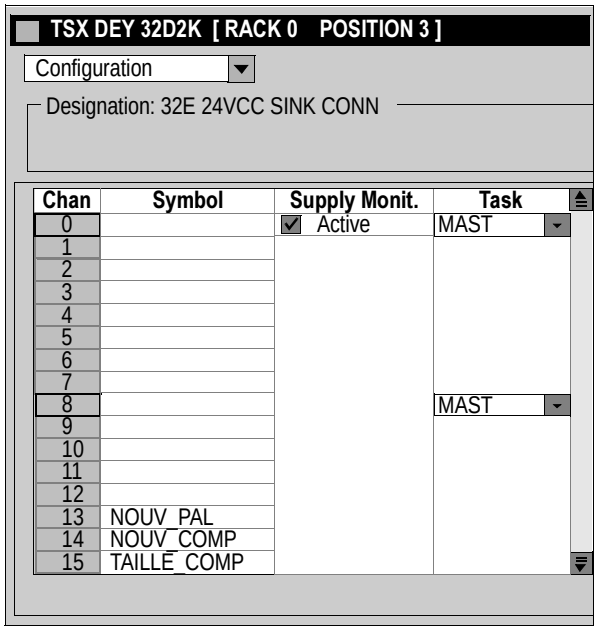
The diagram below shows the configuration of the TSX CTY 2A module used in the example.



The C sensor (for counting the objects) is mated with the 0 channel.

The discrete module's configuration

The diagram below shows the configuration of the TSX DEY 32D2K module used in the example.



The assignments of the inputs are as follows:

- input 13: **new palette** forcing button (active at 1),
- input 14: **new compartment** forcing button (active at 1),
- input 15: 25/50 objects per compartment switch (1 = 50 objects).

Note: this module is only used for input.

Assignment of the internal bits and words

The example uses the assignments of the following internal variables:

- %M0: at 1 starts the M motor, at 0 stops it.
- %M1: at 1 the palette is being positioned, at 0 it is ready to receive the objects.
- %MW0: positions of the jack: 1, 2, 3 (corresponding to the three compartments) and 4 (for moving out the palette).
- %MW1: 25 or 50 (for storing the size of a palette's compartments).

10.2 Programming details for the example

At a Glance

Subject of this section

This section describes in detail the PL7 programming needed to install the counting application.

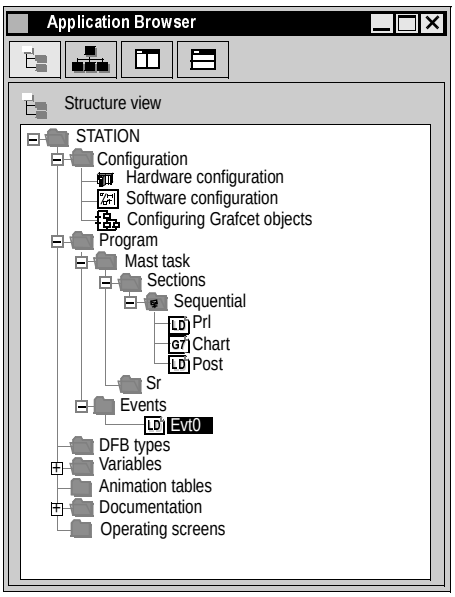
What's in this Section?

This section contains the following topics:

The application structure

Structure view of the application

The diagram below shows the application's structure.



Program: preliminary processing

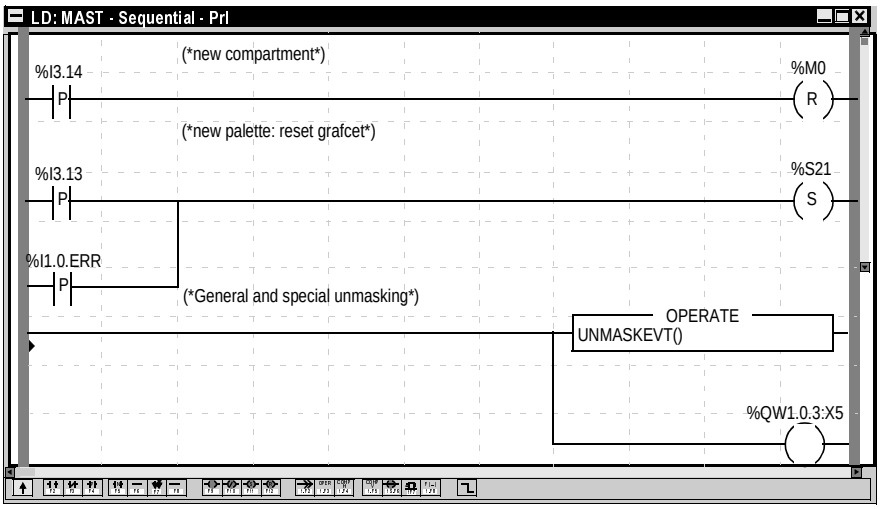
The aims of the preliminary processing

The preliminary processing manages the forced operating modes:

- changing the palette
- changing the compartment

The operation of the preliminary processing

The diagram below shows the programming of the preliminary processing.



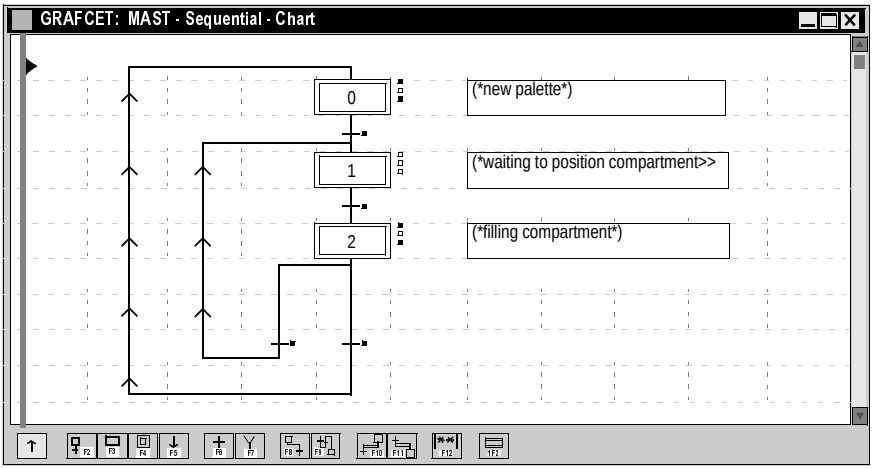
Program: sequential processing

General

Sequential processing constitutes the heart of the application. It concerns monitoring the compartments being filled and changing the palette.

Flowchart of the sequential processing

The diagram below shows the GRAFCET flowchart of the sequential processing.



Program: step 0 of the sequential processing

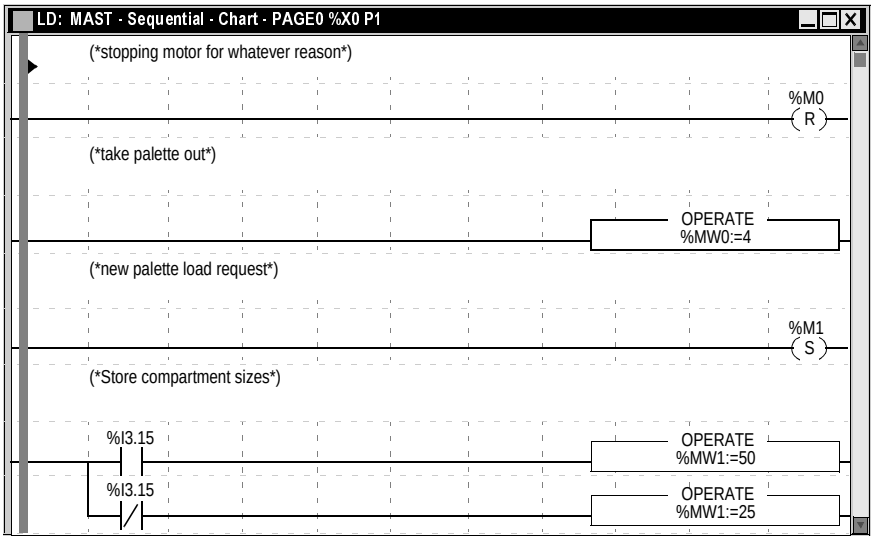
Introduction of step 0

Step 0 initializes the loop:

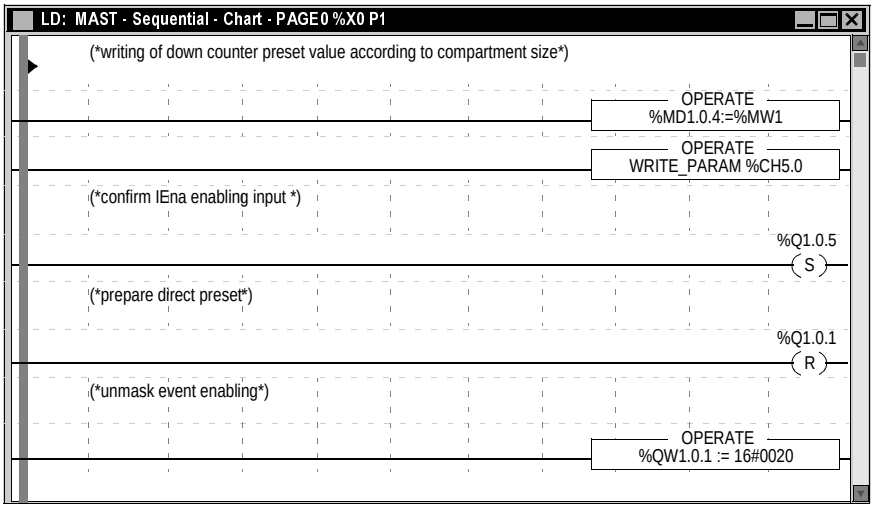
- stops the motor,
- confirms the counting module's **IVal** input,
- unmasks the **zero value overshoot** event.

Step 0: Action on P1 activation

The diagram below shows the action on activation of step 0 (P1).

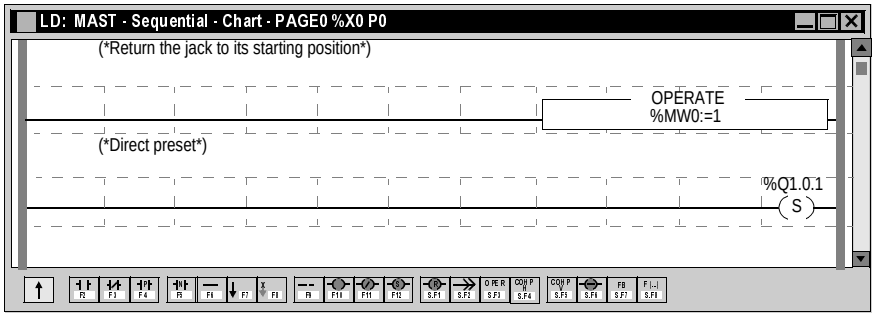


The diagram below shows the result of the action on activation of step 0 (P1).



Step 0: action on P0 deactivation

The diagram below shows the action on deactivation of step 0 (P0).



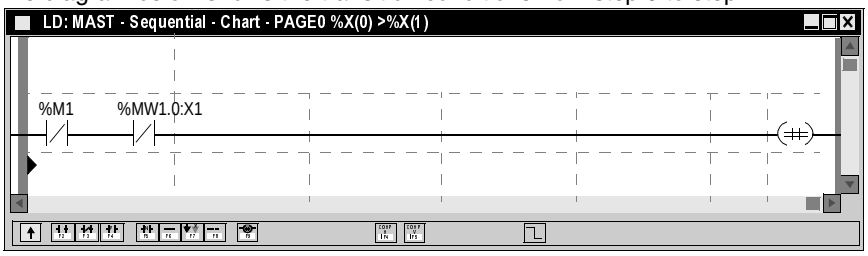
Transition of step 0 to step 1

The transition conditions from step 0 to step 1 are as follows:

- palette ready to receive objects,
- exchanges between the PLC and the counting module are terminated.

Illustration

The diagram below shows the transition conditions from step 0 to step 1.



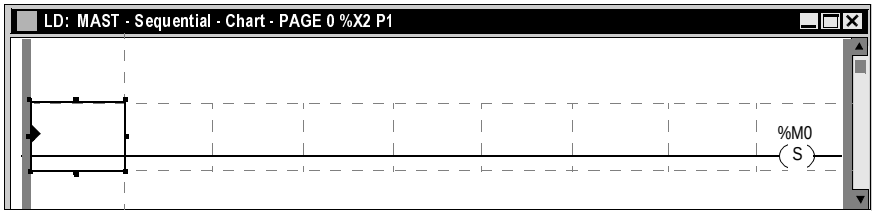
Program: step 2 of the sequential processing

Introduction of step 2

Step 2 deals with monitoring the compartments being stacked.

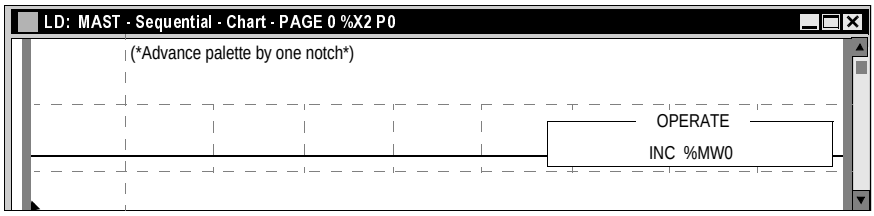
Step 2: action on P1 activation

The diagram below shows the action on activation of step 2 (P1).



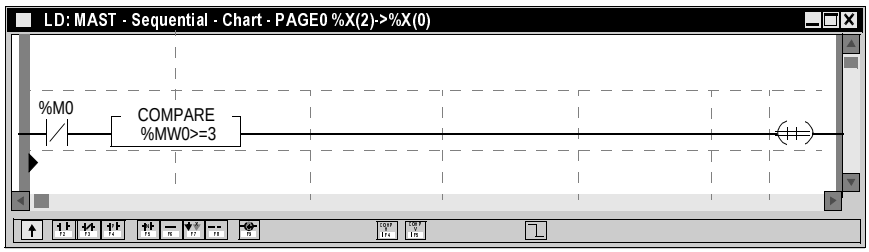
Step 2: action on P0 deactivation

The diagram below shows the action on deactivation of step 2 (P0).



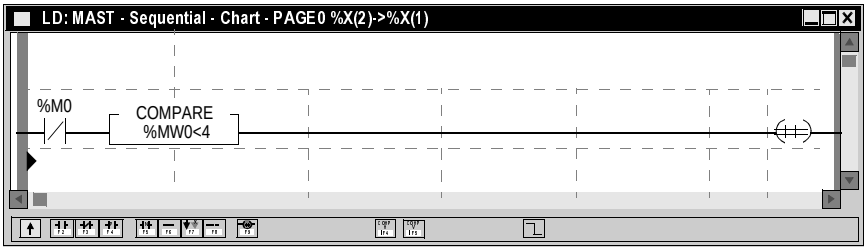
Transition from step 2 to step 0

The diagram below shows the transition conditions from step 2 to step 0.



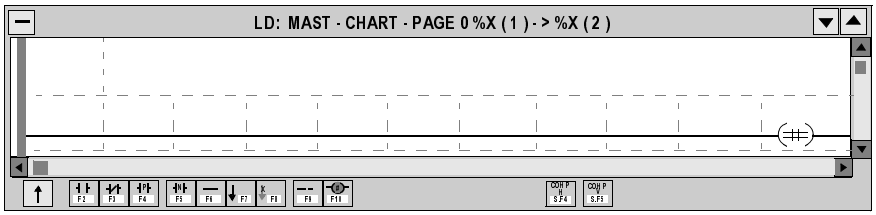
Transition from step 2 to step 1

The diagram below shows the transition conditions from step 2 to step 1.



Transition from step 1 to step 2

The diagram below shows the transition conditions from step 1 to step 2, (this always applies).



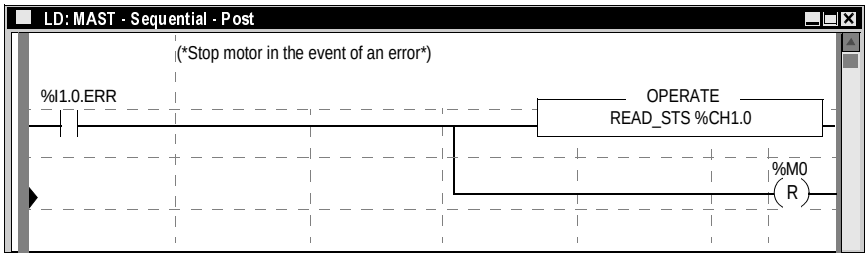
Program: subsequent processing

The aim of subsequent processing

Subsequent processing manages errors and stops the motor if there is a PLC error.

The operation of subsequent processing

The diagram below shows the programming of the subsequent processing.



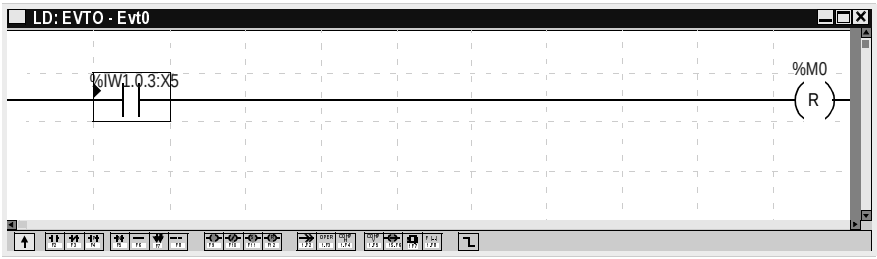
Program: event processing

At a glance

Event processing stops the motor by resetting the %M0 when the switch to zero value event is enabled.

Illustration

The diagram below shows the programming of the 0 event task.



Index

F

I

L

O

S

U