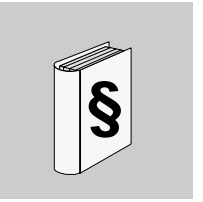
# Premium and Atrium using Unity Pro
# Weighing module
# User manual

June 2006 eng

35006238 03

# Table of Contents

# Safety Information




## Important Information

**NOTICE**

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

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

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

**⚠ DANGER**

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

**⚠ WARNING**

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

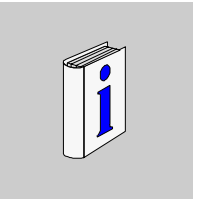
**⚠ CAUTION**

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result** in injury or equipment damage.

**PLEASE NOTE**

Electrical equipment should be serviced only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material. This document is not intended as an instruction manual for untrained persons.

# About the Book



## At a Glance

**Document Scope**

This manual describes the hardware and software implementation of the weighing module for Premium and Atrium PLCs.

**Validity Note**

The data and illustrations found in this documentation are not binding. We reserve the right to modify our products in line with our policy of continuous product development.

The information in this document is subject to change without notice and should not be construed as a commitment by Schneider Electric.

**Product Related Warnings**

Schneider Electric assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product.

For reasons of safety and to ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When controllers are used for applications with technical safety requirements, please follow the relevant instructions.

Failure to observe this product related warning can result in injury or equipment damage.

**User Comments**

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

# Hardware installation of the weighing module

I

## At a glance

**In This Chapter**

This part introduces the hardware installation of the weighing module of the Premium PLC range and its dedicated display accessory.

**What's in this Part?**

This part contains the following chapters:

# General introduction to the weighing module

# 1

## At a Glance

**Aim of this Chapter**

This chapter is a general introduction to the weighing module.

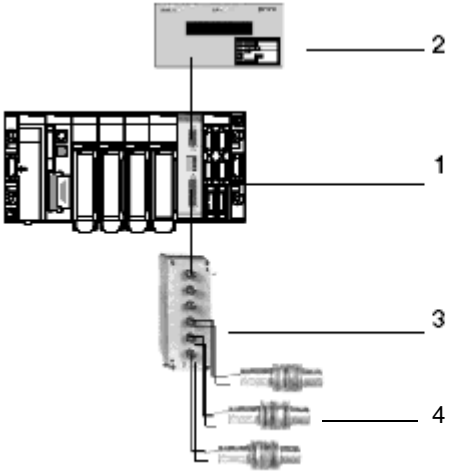
**What's in this Chapter?**

This chapter contains the following topics:

## Introduction to the weighing system

**General**

The following illustration presents the Premium weighing system range.



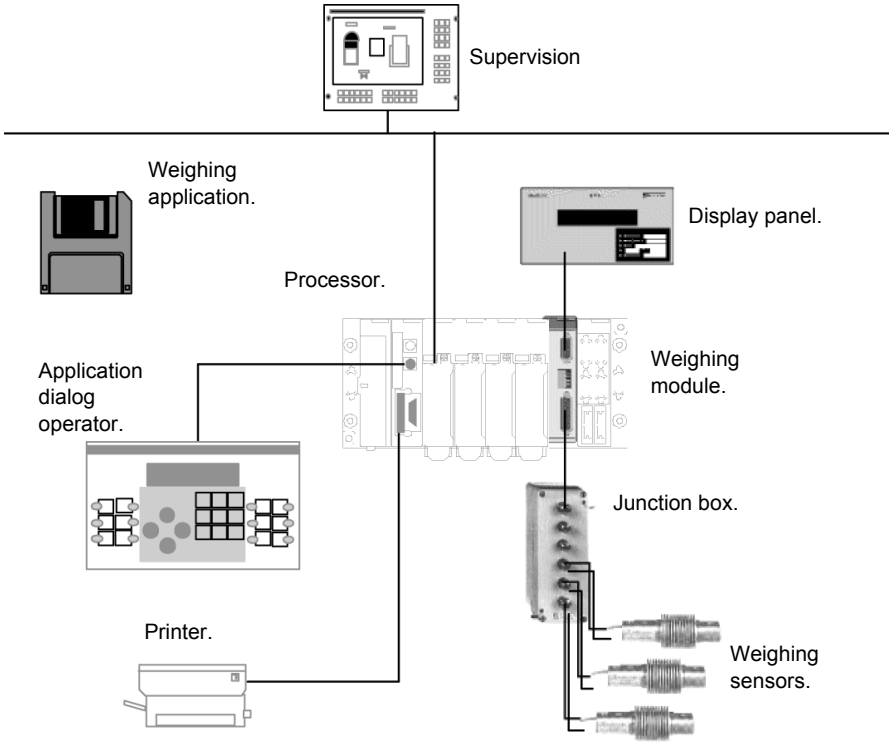

The various elements of the weighing system are described in the following table.

| Number | Description |
|---|---|
| 1 | The **TSX ISP Y101** weighing module. |
| 2 | The **TSX XBT N410** display panel with weighing protocol. |
| 3 | Junction box. |
| 4 | Weighing sensors. |

**Note:** the complete Premium weighing range also integrates the following process control applications:

- filling unit,
- multi-product metering,
- flow regulator,
- graded weighing unit,
- batch and continuous totalizer.

The following illustration presents the different elements that may make up a complete weighing system configuration.



## General description of the weighing module

**Introduction**

The **TSX ISP Y101** weighing module comes with:

- one **measurement input** channel capable of receiving up to 8 constraint gauge sensors,
- **2 discrete callback outputs** for graded filling applications,
- one **sealable digital link** enabling the weight or manual tare to be displayed on a **TSX XBT N410** external display panel.

In order to ensure the integrity of the measurements made, the measurement input, weighing module and display panel can all be lead sealed to meet legal metrology requirements concerning weighing instruments used in commercial transactions.

**Maximum number of TSX ISP Y101 modules per station**

The maximum number of **TSX ISP Y101** modules in a PLC station depends on:

- the type of processor installed (see table below),
- of the number of specific-application channels already used.

**Number of "application-specific" channels supported:**

- Premium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of TSX 57 Processors)
- Atrium (See Premium and Atrium using Unity Pro Manual, Processors, racks and power supply modules, Catalog of Atrium Processors)

# General rules for implementation of the weighing module

# 2

## At a Glance

**Aim of this Chapter**

This chapter presents the general rules for implementation of the weighing module.

**What's in this Chapter?**

This chapter contains the following topics:

## Recommendations on how to install a measurement system

**General**

The quality of the measurement provided by the module may be reduced considerably if the sensor set-up and installation precautions have not been observed. Thus in place of exhaustive information, these few lines should make you aware of some of the precautions which need to be taken.

**Dividing up the loads**

In a measurement system, the weighing sensors support the following weights :

- the maximum weight to be weighed,
- the weight of the loading receiver and its structures (or metrological tare).

This total weight is divided up between 1, 2, 3, 4, 6, even 8 sensors. The design of the mechanical structures, the shape of the loading receiver and the dividing of the load on or within the receiver, means that the total weight is not always equally divided between all the sensors (except of course in the case of a single sensor).

It is therefore a good idea to make sure that the dimensions of the weighing sensors are calculated in such a way as to be able to support the total weight (maximum weight + tare) to which they will be subjected

**Inhibiting interference on the load receiver**

As a weighing sensor deflection is very weak (a few tenths of a millimeter), all interference on the load receiver or any friction on the permanent framework will cause an invalid weight measurement and make correct adjustment of the module impossible.

**Mechanical installation of the weighing sensors**

The sensors in traction or compression must be used vertically respecting their action direction (traction or compression). The maximum admissible tolerance on the installation's verticality is in the region of the degree according to the installation and the required precision.

**Protecting the sensors from interference currents**

It is recommended that each sensor be provided with a mass flex which plays the role of the electric " shunt " with the aim of protecting sensors from currents capable of circulating in the metallic framework (ground currents, from the terminal to be connected, and electrostatic discharges…).
This flex will be of a sufficient length to not result in mechanical constraints and it will be placed directly next to the sensors, between the permanent framework and the load receiver.

**Contact with water and corrosive products**

Weighing sensors are manufactured as waterproof. It is recommended, however, that they be prevented from coming into contact with water, corrosive products and direct sunlight.

**Preventive maintenance of the installation and accessories**

The weighing module requires no special maintenance. The weighing sensors, however, should be cleaned periodically if used in a difficult environment.
It is advisable to periodically test and service the mechanical state of the load receiver.

- Cleaning the receiver and its structures because of a product deposit or various material deposits may result in a noticeable variation of the tare.
- Checking the verticality of the weighing sensors.
- Checking the sensor and actuator states according to their period of use.
- Etc.

**Note:** Statistics show that 90% of breakdowns occurring on a weighing/dosing installation are not attributable to the electric command device, but to the installation itself (defective limit switches, mechanical faults…).

## Installation of the weighing module

**At a Glance**

The method and precautions relating to the installation of the weighing module are detailed below:

**Installation**

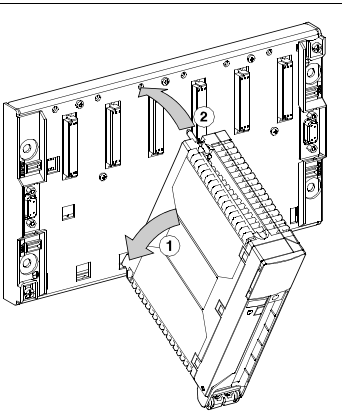
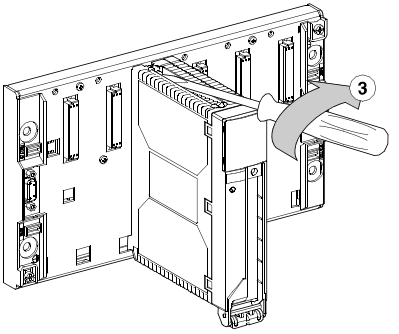
The weighing module **TSX ISP Y101** is standard format and therefore occupies a single position in the **TSX RKY•••** racks. It can be installed in all positions on the rack except for the first two (PS and 00) which are reserved for the rack power supply module (**TSX PSY•••**) and processor module (**TSX 57•••**) respectively.
They are powered by the rack back bus, and can be positioned either in the standard rack or in an extendable rack.

**Installation precaution**

The installation and removal of the weighing module can be done with the PLC **switched on** (without risk of damage to the module or disruption to the PLC).

## Installing the module on the rack

Installation of the weighing module on the rack is carried out as follows:

| Step | Action | |
|---|---|---|
| 1 | Position the two lugs at the rear of the module (the lower part of the module) in the centering holes located on the lower part of the rack. |  |
| 2 | Pivot the module upwards so as to plug it into the rack's back connector. | |
| 3 | Fix the module to the rack by tightening the fixing screw located on the upper part of the module.<br>**Note**: If this screw is not tightened, the module will not stay in the rack position. |  |

## Cabling precautions on the weighing module

**At a Glance**

To protect the signal from external noises induced in serial mode and from noises in common mode, you are advised to take the following precautions.

**Kind of conductors**

Use shielded twisted pairs of a minimum section of 0.28 $mm^2$ (AWG24 gage).

**Cable shielding**

The measurement cable shielding should only be connected to the ground on the module side. If problems arise, if the grounds on either side of the connection are of good quality, then both ends of the shield can be connected to the ground.
On the Sub-D connectors connect the cable shield to the cover of the connector, the PLC ground being connected by the tightening screws of the Sub-D connector. For this reason, the male Sub-D connector **must** be screwed onto its female connection base.

**Cable routing**

Keep the measurement wires as far as possible from the discrete input/output cables (particularly relay outputs) and the cables which transmit "power" signals.
Avoid :

- parallel routing (maintain a distance of at least 20 cm between the cables),
- and cross them at right-angles.

**Note:** The measurement input is grounded via the module.

# Debugging of the weighing module

# 3

## At a Glance

**Aim of this Chapter**

This chapter presents the general rules for debugging the weighing module.

**What's in this Chapter?**

This chapter contains the following topics:

## Weighing module fault display

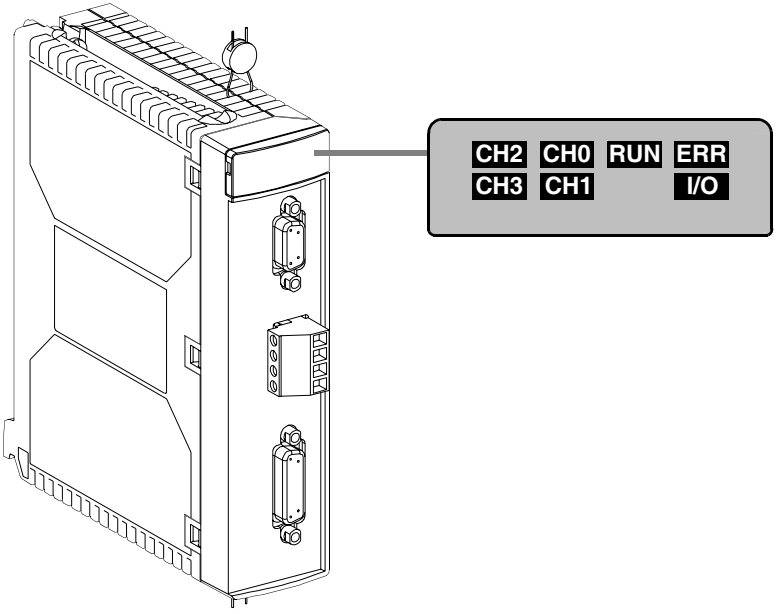
**At a Glance**

The weighing module is fitted with LEDs which display the status of the module and the status of the channels. We distinguish between :

- the module status LEDs : RUN, ERR and I/O,
- the channel status LEDs : CH•.

**Illustration**

The following diagram shows the weighing module display screen :



**Description**

Three LEDs located on the display panel of each module indicate the operational status of the module (LED on, flashing and off) :

- The green LED RUN : indicates the module's operational status
- The red LED ERR : indicates an internal fault in the module or a fault between the module and the rest of the configuration
- The red LED I/O : indicates an external fault.

**Note:** The CH• status LEDs are not used in analog modules.

The various possible faults are grouped in the following table :

| **LED** | **On** ● | **Flashing** ⊗ | **Off** ○ |
|---|---|---|---|
| **RUN (green)** | Normal operation | - | Module faulty or switched off |
| **ERR (red)** | Internal error, module faulty | Communication error, missing, invalid or faulty application. | No internal error |
| **I/O (red)** | External errors:<br>● overload or underload error during calibration,<br>● range overshoot error,<br>● measurement error,<br>● sealed module (configuration refused). | No connector to the weighing sensors. | No external error |
| **CH•** | No channel status LEDs | | |

## Weighing module diagnostics

**At a Glance**

A faulty module is makes itself evident be means of lit or flashing RUN, ERR and I/O LEDs.
There are three groups of faults: external errors, internal errors and other faults.

**Fault diagnostics**

The following table can be used to diagnose faults relating to the three LEDs: RUN, ERR and I/O :

| **Module status** | **Status of LEDs** | | |
|---|---|---|---|
| | **RUN** | **ERR** | **I/O** |
| Normal operation | ● | ○ | ○ |
| Module faulty or switched off | ○ | ⊗ | ○ |
| Internal errors (module broken down):<br>• communication with CPU possible<br>• communication with CPU impossible | <br>●<br>○ | <br>●<br>● | <br>○<br>○ |
| External errors :<br>• overload or underload error during calibration,<br>• range overshoot error,<br>• measurement error,<br>• sealed module (configuration refused) | ● | ○ | ● |
| Other faults :<br>• communication error (absent, invalid or faulty application) | ● | ⊗ | ○ |

| **Key :** |
|---|
| ○ **LED unlit** |
| ⊗ **LED flashing** |
| ● **LED lit** |

# Description of the weighing module connections

# 4

## At a Glance

**Aim of this Chapter**

This chapter introduces the weighing module connections.

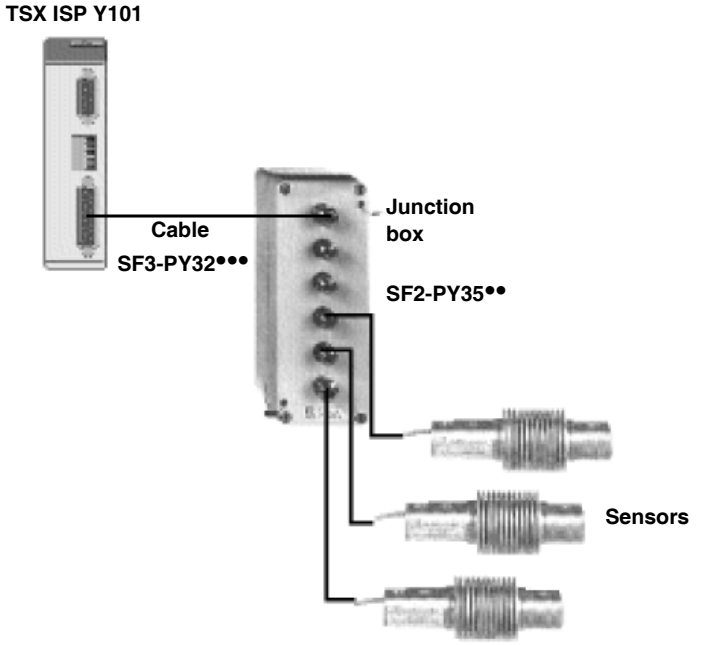
**What's in this Chapter?**

This chapter contains the following topics:

## Measurement connection

**General**

The measurement sensors are connected using a 15 pin female Sub-D connector on the module side.



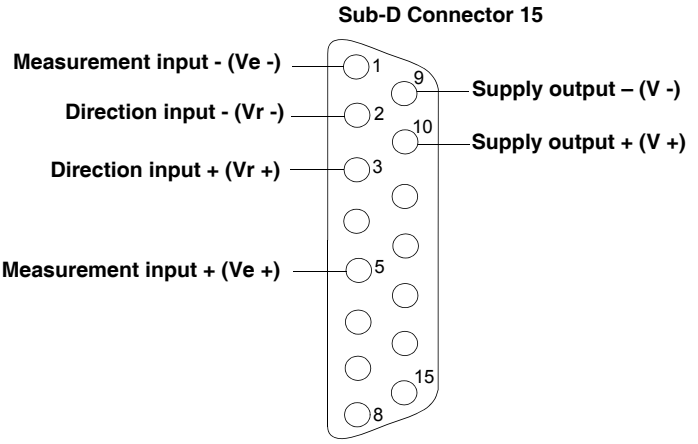

The module and the Sub-Ds can be plugged-in and unplugged with the power on.

The type of cable to be used is a 6 conductor cable with a 15 pin Sub-D connector.

Sensor supply is exclusively provided by the module.

**Sub-D Connector 15**

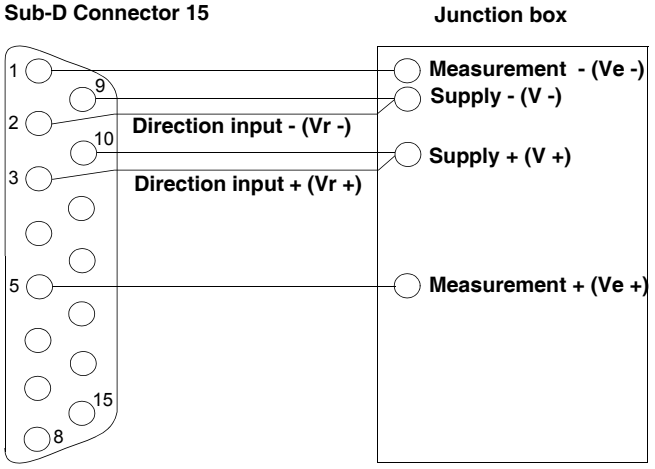
Connection of the 15 pin Sub-D connector is shown in the following illustration:



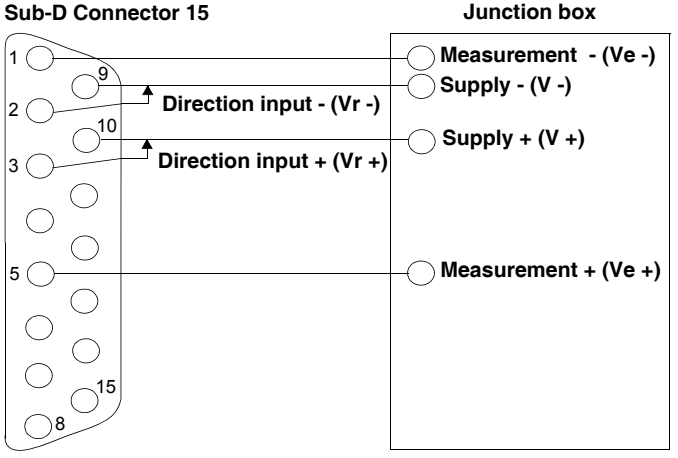
**Cabling of sensor inputs**

Depending on the accuracy desired, there are two ways to cable the sensor inputs. In each case **it is essential to cable them in order to obtain a measurement**.

High accuracy cabling or where the length of the cable between the module and the sensor connection unit is great:




Average accuracy cabling or where the length of the cable between the module and the sensors connection is not great:

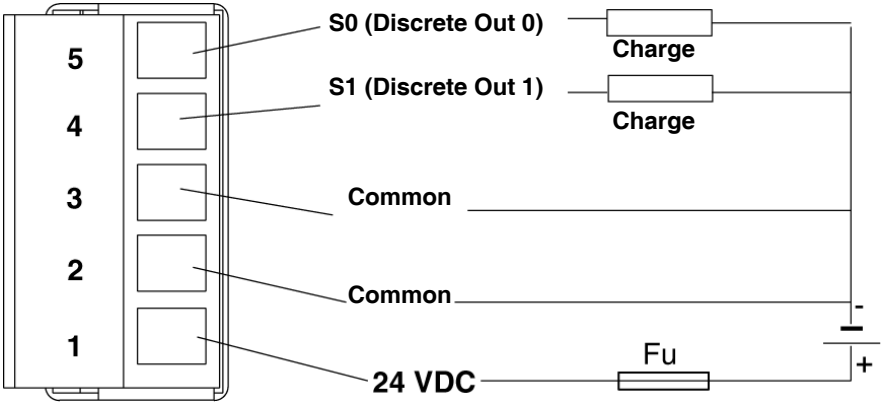


## Connection of discrete outputs of the weighing module

**General**

Weighing module discrete outputs are used to trigger actions on threshold crossings. This functionality is used in the "filling unit" application.

Discrete outputs are connected using a screw terminal block:




The common 2 and 3 are linked by the card.

**Characteristics of the discrete outputs**

The following table shows the characteristics of the discrete outputs of the module **TSX ISP Y101**:

| Discrete output | Characteristics |
|---|---|
| **Number of channels** | 2 |
| **Type** | A transistors |
| **Response time** | 1 ms discrimination. The threshold crossover point between two measurements is calculated by interpolation to the millisecond. |
| **Nominal supply voltage** | 24 V |
| **Insulation voltage** | 1500 Vrms |
| **Maximum current** | 500 mA |
| **Protection** | Reverse polarity and short circuit.<br>Install a fuse on the +24 V pre-actuator supply. |

**Protection**

The outputs are galvanically protected by the ground.

Each of the two output channels is protected against:

- short-circuits and overloads,
- reverse polarities.

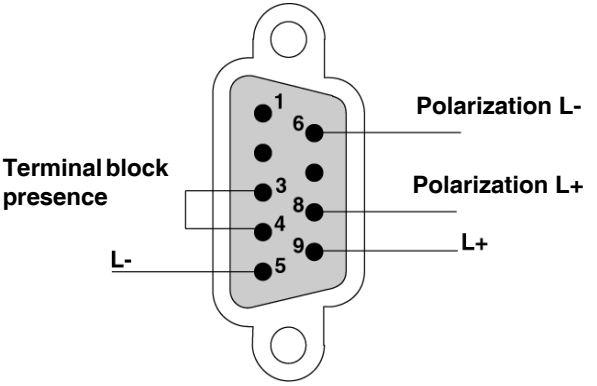
**Note:** In order to best protect against reverse polarities , it is essential **to place** a fast-acting **fuse** on the supply, upstream of the load (shown as Fu in the diagram above).

## Pins of the serial link for the display panel

**General**

The serial link is used to carry over the weight to an external display panel.

The connection to the terminal is via a female 9 pin Sub-D connector on the module side, the link is RS485, the connections are described below :




The line on the module side is polarized using straps 6-5 and 8-9.

**Characteristics**

The following table shows the characteristics of output display panel :

| Output display panel | Characteristics |
|---|---|
| **Physical interface** | non-isolated RS 485 |
| **Bit rate** | 9,6 Kbits/s |
| **Format** | 1 start bit, 8 data bits and 1 stop bit |
| **Remote distance** | Maximum 30 m |

## The TSX XBT N410 display panel

**General**

The **TSX XBT N410** is designed to be connected to the weighing module, in order to act as the main display panel and display weight information.

The display panel should be connected to the module by a shielded cable with two conductors. It is equipped with a 15-pin female SUB-D connector.

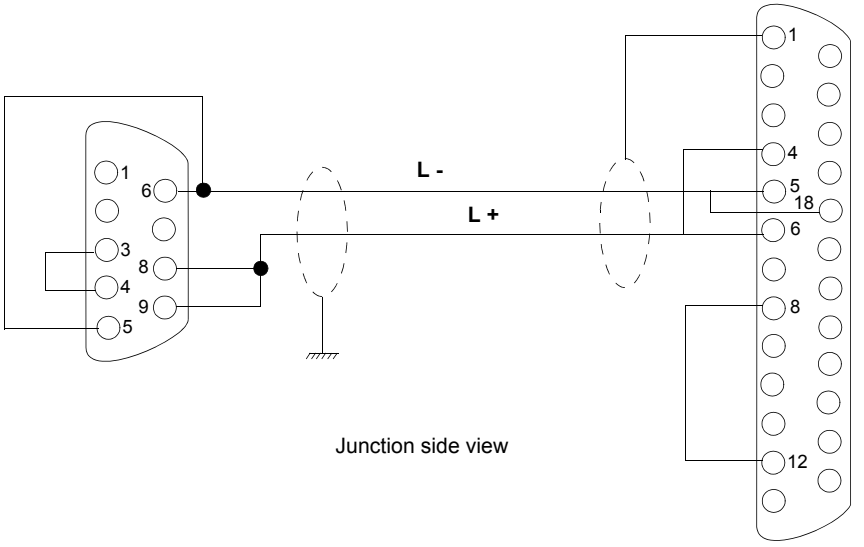
**Characteristics**

The following table shows the electrical characteristics of the display panel:

| | | |
|---|---|---|
| **Display** | | Green back-lit LCD (122x32 pixels) |
| **Display capacity** | | 2 lines of 20 characters |
| **Refresh period** | | 100 ms |
| **Serial link** | | RS 485 |
| **Transmission speed** | | 9.6 Kbits/s |
| **Connection** | | With 25-way SUB-D connector |
| **Power supply** | | Plug-in terminal block, 3 screw terminal (pitched at 5.08 mm [0.2 in]). Maximum clamping capacity: 1.5 mm (0.06 in). |
| **Voltage limits** | | 18 to 30 V, DC |
| **Ripple** | | 5% maximum |
| **Consumption** | | 5 W maximum |
| **Ambient temperatures** | for operation | 0...55°C (32...130°F) |
| | for storage | -20...60°C (-4...140°F) |
| **Degree of protection** | front | IP65 and in compliance with IEC 60529 and Nema 4X ("outdoor use") |
| | rear | IP20, in compliance with IEC 60529 |
| **Compliant with standards** | | IEC 61131-2, IEC 60068-2-6, IEC 60068-2-27, UL 508 and CSA C22-2 No.14 |

**Display panel connections**

The following diagram shows the link between the **TSX ISP Y101** weighing module and the **TSX XBT N410** display panel:




Junction side view

The cable, **a shielded twisted pair**, from the weighing module to the **TSX XBT N410** cannot exceed 100 meters (328 feet).

The shield on the module side should be connected to the metal part of the SUB-D connector.

# Module TSX ISP Y101

# 5

## Overview of this Chapter

**Aim of this Chapter**

This chapter introduces the general characteristics of the weighing module **TSX ISP Y101**.

**What's in this Chapter?**

This chapter contains the following topics:

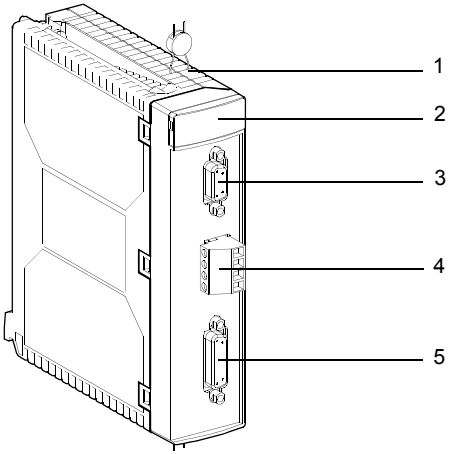
## Physical description of the weighing module

**General**

The following is a physical description of the weighing module **TSX ISP Y101**.

**Illustration**

The following illustration presents the module **TSX ISP Y101**.




**Description**

The following table describes the different elements of the weighing module:

| Number | Description |
|---|---|
| **1** | A plastic case equipped with shielding plates protecting the electronic circuits and providing protection against radiant interference. |
| **2** | Display block |
| **3** | A 9 pin Sub-D female connector for connecting the remote display (**TSX XBT N410**). |
| **4** | A 5 pin screw terminal block for connecting discrete outputs. |
| **5** | A 15 pin Sub-D female connector for connecting weighing sensors. |

# General characteristics of the module TSX ISP Y101

**At a Glance**

This part introduces the general characteristics of the module **TSX ISP Y101**.

**The measurement device**

This following table provides the general characteristics of the module **TSX ISP Y101**.

| | |
|---|---|
| **Electrical range** | 0 to 25 mV |
| **Minimum dynamic** | 4.5 mV |
| **Maximum dynamic** | 25 mV |
| **Converter resolution** | 20 bits (1 048 576 pulses |
| **Limitations of use** | 50 000 pts |
| **Conversion speed** | 50 measurements/second |
| **Zero drift** | < 200 nV/°C |
| **Gain drift** | < 10 ppm/°C |
| **Non linearity** | < 20 ppm (FS) |
| **50 Hz series mode rejection** | > 120 dB |
| **Maximum length of the measurement cable** | 100 m for cable at 0.4 $mm^2$<br>200 m for cable at 0.6 $mm^2$<br>For 1 to 8 sensors |

**Consumption**

The following table shows the consumption values of the module **TSX ISP Y101**:

| | | **Typical** | **Maximum** |
|---|---|---|---|
| **Consumption** | on 5 VDC | 150 mA | 330 mA |
| | on 24 VR (1) | 7 mA + 17 mA x N | 14 mA + 17 mA x N |
| **Dissipated power** | | 0.75 W | 1.65 W |
| | | | |
| **Key:** | | | |
| (1) | Consumption depends on the number of sensors (N) present on the measurement input. | | |

# Software installation of the weighing module

II

## At a glance

### In This Chapter

This part introduces the Weighing application-specific function on Premium PLC and describes its installation with Unity Pro software.

### What's in this Part?

This part contains the following chapters:

# General introduction to the weighing application-specific function

# 6

## At a glance

**Aim of this section**

This section introduces the weighing application-specific function on Premium PLCs.

**What's in this Chapter?**

This chapter contains the following topics:

## Installation Phase Overview

**Introduction**

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

- in offline mode
- in online mode

If you do not have a processor to connect to, Unity Pro allows you to carry out an initial test using the simulator. In this case the installation (See *Implementation Phases with Simulator, p. 47*) is different.

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

**Installation Phases with Processor**

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

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

## Implementation Phases with Simulator

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

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

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

## Operation of the weighing module

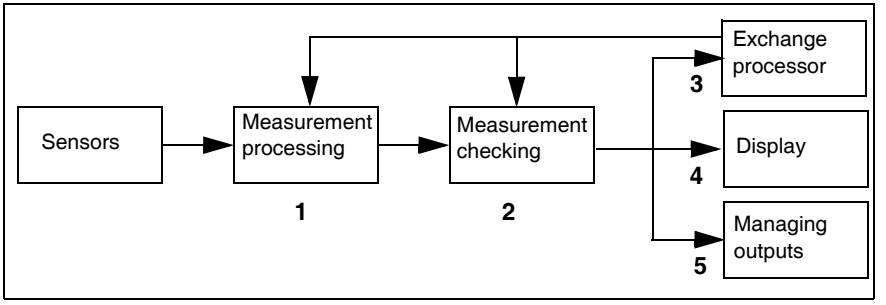
**General**

In the PLC environment, the module uses, in the same way as the other modules, a set of data which are specific to it.

This information is used for the exchanges (report and commands) with the processor.

**Structural diagram**

The following operation diagram shows the processes executed by the module and gives all the elements to be configured.



## Description of operation

The following table describes the various operating phases of the module.

| Phase | Operation | Description |
|---|---|---|
| 1 | Measurement processing. | The signal from the weighing sensors is:<br>● converted,<br>● filtered according to the choice made in the configuration screen,<br>● scaled, the scaling characteristics are determined at the end of a calibration. |
| 2 | Measurement checking. | The measurement taken from the processing is subject to the following checks:<br>● a underload or overload check,<br>● a stability check defined by a stability range and a stability time,<br>● a check for presence in the zero zone. |
| 3 | Data exchanges with the processor. | The module receives and processes the commands from the processor (Set to zero, tare mode semi automatic, etc.).<br>It also prepares the data in legal format for display on the **TSX XBT N410**.<br>It returns various information to the processor such as the gross weight, the net weight, the flow, the tare and the statuses. |
| 4 | Data display. | The **TSX XBT N410** displays the weight or the manual tare, in the unit chosen in the configuration and 4 items of additional information: the net weight, the stability, the presence in the zero zone and the unit of weight. |
| 5 | Output management. | The card can directly manage 2 discrete outputs and control them according to thresholds transmitted to the module by the application program.<br>The elements used for this management are:<br>● the switchover thresholds,<br>● the direction of movement of the weight (Weighing or Downweighing),<br>● the switchover logic of the outputs. |

# Configuration of the Weighing application

# 7

## At a glance

**Subject of this chapter**

This chapter describes how to select and modify the parameters of the weighing module's configuration.

**What's in this Chapter?**

This chapter contains the following sections:

# 7.1 Configuration of the weighing module : general

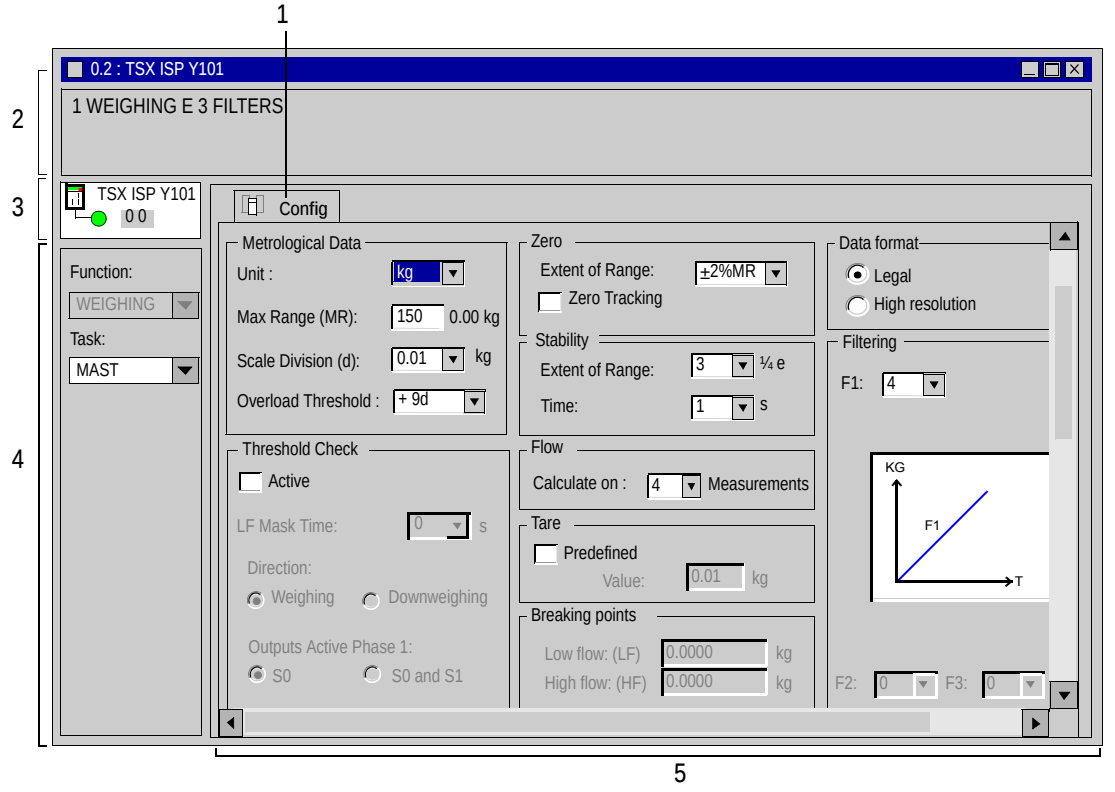
## Description of the weighing application function configuration screen

**General**

Configuration (See Unity Pro, Operating Modes Manual, Access to the module configuration editor) information is used to define the measuring characteristics and to adapt the operation of the module to the application for which it is intended.

**Illustration**

This screen allows you to display and modify parameters in offline mode, as well as to debug in online mode.



**Description**

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

| Address | Element | Function |
|---|---|---|
| 1 | Tabs | The front tab indicates the current mode (**Configuration** in this example). Each mode may be selected by the corresponding tab. The **Debug** and **Calibration** modes are only accessible in online mode. |
| 2 | **Module** zone | Shows the abbreviated name of the module.<br>In online mode, this zone also contains the three LEDs **Run**, **Err** and **I/O**. |
| 3 | **Channel** field | Is used:<br>● By clicking on the reference number, to display the tabs:<br>  ● **Description** which gives the characteristics of the device.<br>  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.<br>  ● **Fault** which shows the device faults (in online mode).<br>● To select the channel,<br>● To display the **Symbol**, name of the channel defined by the user (using the variable editor). |
| 4 | **General parameters** zone | Comprises:<br>● the dropdown **Function** menu,<br>● the dropdown **Task** menu defining the task (**MAST** or **FAST**) in which the implicit exchange objects of the channels will be exchanged. |
| 5 | **Configuration** zone | Is used to configure the various channels' configuration parameters. |

# 7.2 Parameters of the weighing module channels

## Weighing module configuration parameters

**List of parameters**

The following table shows the parameters available for the weighing module.

| Parameters | | Possible choices | Unit of measurement |
|---|---|---|---|
| **Task** | | **Mast**/Fast | - |
| **Metrology/Unit** | | **kg** | kilogram |
| | | g | gram |
| | | t | tonne (metric) |
| | | lb | pound (= 453 g) |
| | | oz | ounce (= 28.35 g) |
| | | <none> | - |
| **Metrology/Max range** | | from 0 to 65535 (**150**) | in the weight unit chosen |
| **Metrology / Scale division** | | 1 x $10^n$ (1 x $10^{-2}$ = **0.01**) | in the weight unit chosen |
| | | 2 x $10^n$ | |
| | | 5 x $10^n$ | |
| **Metrology / Overload threshold** | | **+9 e** | scale divisions |
| | | +2 % PM | % of Max range |
| | | +5 % PM | % of Max range |
| **Zero / Zero tracking** | | **Inactive**/active | - |
| **Zero / Extent of range** | | **+/-2 % PM** or +/-5 % PM | - |
| **Data format** | | **Legal**/High resolution | - |
| **Stability / Extent of range** | | 2, **3**, 4, 6 or 8 | 1/4 of the scale division |
| **Stability / Time** | | 0.4, 0.5, 0.7 or **1** | seconds |
| **Filtering / Coefficients** | F1 | from 0 to 19 (**4**) | - |
| | F2, F3 | from 0 to 19 (**0**) | - |
| **Flow / Calculation** | | 2, **4**, 8, 16, 32 or 64 | measurements |
| **Tare** | | **Not predefined**/predefined | in the weight unit chosen |
| **Threshold check** | | **Inactive**/Active | - |
| **LF mask time** | | **0** to 1.5 s per 0.1 s step | seconds |
| **Output logic** | | **Weighing**/Downweighing | - |
| **Phase 1 active outputs** | | **S0** or (S0 and S1) | - |
| **Breaking points PD and GD** | | from **0** to Max Range | in 1/100ths of the unit |

**Note:** the parameters in bold correspond to the parameters configured by default.

# 7.3 Configuration of the weighing module parameters

## At a glance

**Aim of this sub-section**

This sub-section shows the implementation of the various weighing module configuration parameters.

**What's in this Section?**

This section contains the following topics:

## How to modify the task parameter

**At a glance**

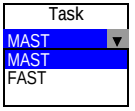
This parameter defines the processor task in which the input acquisition and the output update take place.
Possible choices are:

- The **MAST** task,
- The **FAST** task.

| |
|---|
| **Note:** It is only possible to modify this parameter in offline mode. |

**Procedure**

The following table shows the procedure to define the type of task assigned to the module channels.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Click on the button of the drop-down menu **Task** in the **General parameters** zone.<br>**Result**: a drop-down list appears.<br>  |
| 3 | Select the required task. |
| 4 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify metrological information

**At a glance**

The configuration screen offers the following metrological information.

| Designation | Description |
|---|---|
| **Unit** | gives the choice for the weight measurement unit:<br>● **g**: gram,<br>● **kg**: kilogram,<br>● **t**: tonne (metric),<br>● **lb**: pound (lb = 453 g),<br>● **oz**: ounce (oz = 28.35 g),<br>● **without**: any unit. |
| **Max Range (MR)** | This is the maximum weight that it is possible to weigh with the instrument and without including the weight of the empty load receiver (in the legal format (See *How to modify the data format, p. 60*)). |
| **Scale Division** | The value of the scale division is in the form of 1, 2 or 5 multiplied by $10^n$ (n being a positive or negative integer or zero with its absolute value $\leq 3$).<br>**Example**: for a scale division of 0.002 (if the unit chosen is the kg), the measurement increases by 2 g at a time. |
| **Overload Threshold** | The threshold is the value of the weight above which the display panel no longer indicates the weight (the overload is then indicated with a **>** line on the display panel).<br>It may take the values:<br>● +9 scale divisions,<br>● +2 % of the maximum range,<br>● +5 % of the maximum range.<br>**Example**: the maximum range has been established at 150 kg, the scale division at 10 g, according to the choice of the user, the operating limit will be for:<br>● 9 e: Max range + 9 scale divisions i.e. 150.09 kg,<br>● +2 %MR: 102 % of Max range i.e. 153 kg,<br>● +5 %MR: 105 % Max range i.e. 157.5 kg,<br>**Note**: The under-load threshold cannot be parameterized: it defines the permissible limit of the indication below zero. It is -2 % of the maximum range (the under-load is then indicated with a **<** line on the display panel). |

**Note:** In an industrial context, because of the environment of the weighing installation, choosing a resolution of more than 3000 points assumes that there are rigorous installation precautions.
At the level of the programming screen, it will not be possible to input a resolution greater than 50 000 points. In other words the following disparity should be observed: Maximum Range (MR) ≤ 50000 x Scale division.

**Procedure**

The following table shows the procedure for defining the metrological information.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Select the parameter values **Unit**, **Scale division** or **Threshold overload** using the drop-down lists offered and enter the **Maximum Range** value. |
| 3 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the zero

**At a glance**

The configuration screen offers the following information for zero setting.

| Designation | Description |
|---|---|
| Extent of the recalibration range | Any deviation from zero can be corrected as long as it does not exceed this range.<br>It is defined as a % of the maximum range. It may take the values:<br>● +/-2 % PM (+/- 2 % of the maximum range),<br>● +/-5 % PM (+/-5 % of the maximum range), |
| Zero Tracking | This optional function is used to compensate for slow deviations from zero in the extent of range (+/-2 % of the maximum range).<br>You are not advised to choose this option in the automatic installations. |

**Note:** The discrimination between a slow deviation and a true weight is based on the following rule: any variation in weight lower than the semi-scale division whose repetition frequency is sufficiently weak to preserve the stability of the measurement is considered to be a deviation. The correction engendered by the function is limited to +/-2 % of the maximum range of the flip-flop. When this limit is exceeded, there is no automatic correction.

**Procedure**

The following table shows the procedure for defining zero.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Select the extent of range by using the drop-down menu. |
| 3 | If necessary, check the **Zero tracking** box to confirm this function. |
| 4 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the data format

**At a glance**

The configuration screen allows you to choose the measurement's display format. You may enter the weight value:

- either as a fixed point physical unit: **Legal** format,
- or as a hundredth of a fixed point physical unit: **High resolution**.

> **Note:** A fixed point physical unit is called a whole number expressed as a unit of weight where a point can be put. The position of this is given by the power of ten of the scale division.

**Example**

**Legal format**: the value 3014 means 301.4 kg if the scale is $2*10^{-1}$ kg.

**High resolution format**: the value 301403 means 301,403 kg if the scale is $2*10^{-1}$ kg. This unit offers greater accuracy but it is not accepted by the Legal Metrology Department.

**Procedure**

The following table shows the procedure for modifying the data format.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Check the data format required. |
| 3 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the stability

**At a glance**

The configuration screen offers the following parameters to define the stability.

| Designation | Description |
|---|---|
| **Extent of range** | A weight cannot be measured immediately after receiving a load because the inevitable oscillations affect the mechanical part.<br>The stability range shows the size below which the measurement is considered stable.<br>It is parameterized on 2, 3, 4, 6 or 8 quarters of a scale. |
| **Time** | The stability time shows how long the measurement must stay in the stability range before it is considered stable.<br>It is parameterized on 0.4, 0.5, 0.7 or 1 second. |

**Procedure**

The following table shows the procedure for defining the stability.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Select the extent of range using the drop-down menu **Extent of range** in the **Stability** field. |
| 3 | Select the stability time using the drop-down menu **Time** in the **Stability** field. |
| 4 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify measurement input Filter(s)

**At a glance**

Filters concern the measurement input of the weighing sensors.
By default, a unique filter is offered which is defined for the total duration of the weighing action.
To increase the speed/precision performance of weighing, 3 different filters can be used for the same weighing action, as follows:

- filter **F1** associated with phase 1 (default phase),
- filter **F2** associated with phase 2,
- filter **F3** associated with phase 3.

Each filter can either have:

- a sliding average (filtering coefficients from 1 to 11), where the measurement is the average of the last n values,
- or be of second order (filtering coefficients from 12 to 19), which are referenced by their cut-off frequencies.

**Measurement Phases**

The various phases of a continuous weighing action can be broken down into:

- a phase 1, where speed is the prime feature of control precision (High flow),
- a measurement refining phase 2 (Low flow),
- a final phase 3, where the measurement value differs very little and requires a high level of precision (Residual flow).

## Filter Coefficients

The following list gives the meanings of the filter coefficients:

| Value | Filter type | Characteristics |
|---|---|---|
| 0 | none | not filtered |
| 1 | sliding average | average of the last 2 measurements |
| 2 | sliding average | average of the last 3 measurements |
| 3 | sliding average | average of the last 4 measurements |
| 4 | sliding average | average of the last 5 measurements |
| 5 | sliding average | average of the last 8 measurements |
| 6 | sliding average | average of the last 16 measurements |
| 7 | sliding average | average of the last 25 measurements |
| 8 | sliding average | average of the last 32 measurements |
| 9 | sliding average | average of the last 40 measurements |
| 10 | sliding average | average of the last 50 measurements |
| 11 | sliding average | average of the last 64 measurements |
| 12 | second order filter | cut-off frequency at 15 Hz |
| 13 | second order filter | cut-off frequency at 10 Hz |
| 14 | second order filter | cut-off frequency at 8 Hz |
| 15 | second order filter | cut-off frequency at 6 Hz |
| 16 | second order filter | cut-off frequency at 4 Hz |
| 17 | second order filter | cut-off frequency at 2 Hz |
| 18 | second order filter | cut-off frequency at 1 Hz |
| 19 | second order filter | cut-off frequency at 0.8 Hz |

## Procedure

The following table shows the procedure for defining the filter:

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | If filters F2 et F3 are being used check the **Active** box in the **Threshold check** field. |
| 3 | Select for each phase, the filtering coefficient using the drop-down menus **F1**, **F2** and **F3** in the **Filtering** field. |
| 4 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the flow calculation

**At a glance**

You can choose the number of measurements for calculating the flow. (One measurement is taken every 20 milli-seconds).
The flow is a difference in the values of filtered weights for a number of configured measurements. The flow is calculated using the following formula:
Flow n = Valn - (Valn-b)
where:

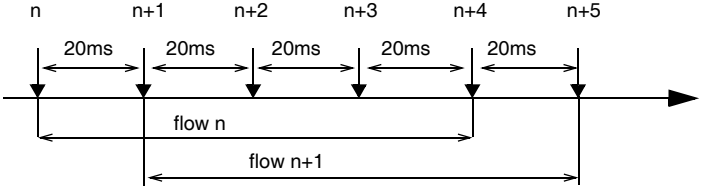
- b = the number of measurements for calculating the flow,
- Valn = the filtered weight value at the moment n,
- Valn-b = the filtered weight value at the moment n-b.

**Operation**

At each instant, the frequency is calculated and implicitly sent back to the processor as the weight measurement, in order to allow threshold corrections.
The flow is always calculated in high resolution format.
This calculation can be done on 2, 4,8, 16, 32 or 64 measurements.
By default, the number of measurements is 4.

**Example**

The following figure illustrates a calculation on 4 measurements.




flow n+4 = Val n+4 - Val n
flow n+5 = Val n+5 - Val n+1

**Procedure**

The following table shows the procedure for defining flow calculation.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Select the number of measurements using the drop-down menu from the **Flow** field. |
| 3 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the tare

**At a glance**

The tare is the weight measurement memorized during the last semi-automatic tare unit command.
However, if necessary, you can introduce a tare value manually. Therefore, this tare value is known as predefined or manual and can be transmitted to the module. It is expressed in legal format (physical unit with fixed decimal point).
The tare must be positive or zero and be lower than the Max. Range.
Once a device of this sort is used, the predefined tare (PT) indicator is positioned. It is disabled once a **Taring** order is executed.

**Note:** The entry range extends from 0 to 65535. If you want a greater tare, you must modify the scale and enter the tare accordingly.

**Procedure**

The following table shows the procedure for defining a predetermined tare and the tare value.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | If necessary check the **Predetermined** box in the **Tare** field to confirm this function.<br><br>**Note**: If this box is already checked, you must first:<br>• uncheck this box,<br>• enable the configuration screen,<br>• check the **Predefined** box again. |
| 3 | Enter the tare value in the **Value** entry field. |
| 4 | Confirm the modification with the command **Edit** → **Validate**. |

## How to modify the threshold check

**At a glance**

The threshold check manages the module's discrete outputs:
- the **High Flow cut-off point** is associated with output S0,
- the **Low Flow cut-off point** is associated with output S1.

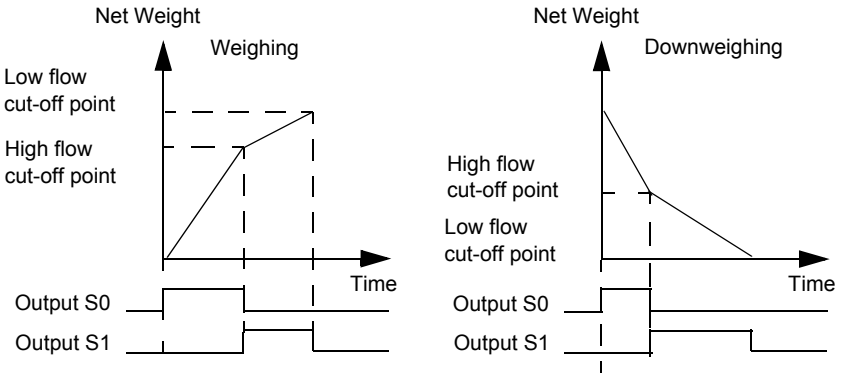
The configuration screen gives the following threshold check information.

| Designation | Description |
|---|---|
| Active | Discrete output management is operational if this box is checked.<br>It is not checked by default. |
| Direction | The detection direction corresponds to the direction in which the thresholds are recognized i.e.:<br>● **Weighing** (filling),<br>● **Downweighing** (emptying).<br>This is the concept of exceeding by a greater value, in the case of weighing, or by a lower value, in the case of downweighing.<br>By default, **Weighing** is selected. |
| Phase 1 active outputs | The choice concerns the control of the **S0** output on its own, or the **S0** and **S1** outputs at the same time.<br>See the explanation that follows.<br>By default, the module only activates **S0** in the first phase. |
| Cut-off points | The measurement can be associated with 2 thresholds for the following dosages: A **High Flow cut-off point** and a **Low Flow cut-off point**.<br>Depending on the logic defined, the **S0** and **S1** outputs go to zero when these thresholds are met.<br>The threshold values allowed lie between 0 and the maximum range.<br>They are expressed in high resolution (one hundredth of a physical unit with fixed decimal point). |
| LF (Low Flow) mask time | It defines the time after the high flow cut-off, during which the module no longer checks the Weight/Threshold;<br>This is to mask the overshoot caused when the product has a drop in voltage. The values allowed lie between 0 and 1.5 seconds per 1/10th second step.<br>See the explanation that follows.<br>By default, this time is zero. |

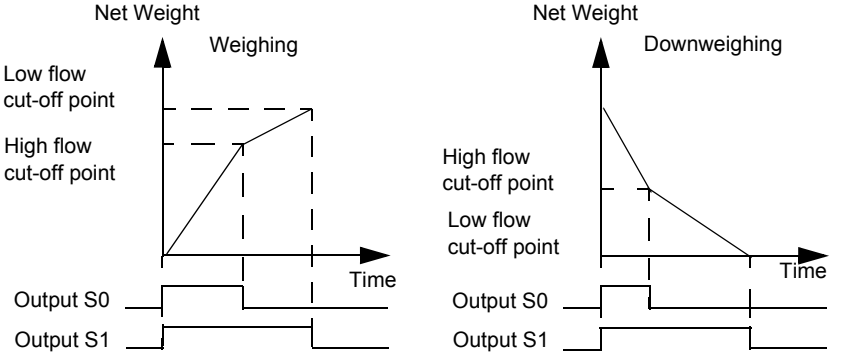
**Activating Outputs**

The following illustration describes the output operating differences between the choice of active phase 1 Outputs: S0 or S0 and S1.

**Active output phase 1 (S0)**

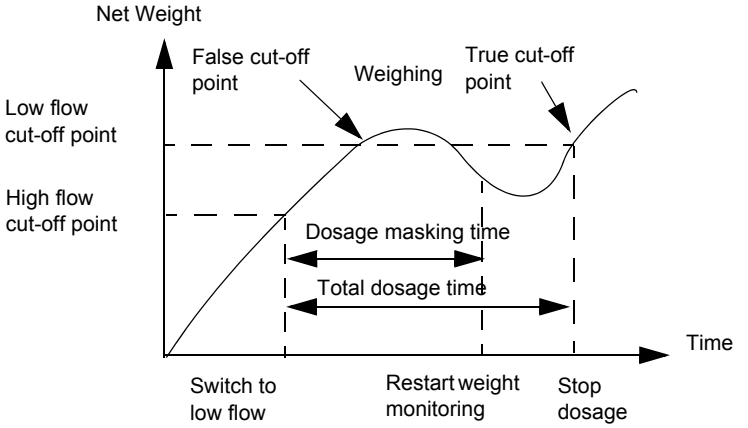



**Active output phase 1 (S0 and S1)**



**Masking Time**

The following illustration shows the role of the masking time, the aim of which is to mask the overshoot caused when the product has a drop in voltage.




**Procedure**

The following table shows the procedure for the threshold check.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | If necessary check the **Active** box in the **Threshold check** field to activate this function. |
| 3 | Check the selection boxes corresponding to the detection direction (**Weighing** or **Downweighing**) and to the phase 1 active outputs (**S0** or **S0 and S1**). |
| 4 | Enter the **Low flow** and **High flow** in the **Breaking points** field. |
| 5 | Using the drop-down menu in the **Threshold check** field, select the **PD masking time**. |
| 6 | Confirm the modification with the command **Edit** → **Validate**. |

# Presentation of the weighing module language objects

# 8

## At a glance

**Aim of this section**

This section describes the language objects associated with the weighing module from the various IODDTs.

**What's in this Chapter?**

This chapter contains the following sections:

# 8.1 General on the weighing programming

## At a glance

**Subject of this section**

This section describes the general principles of programming a weighing application.

**What's in this Section?**

This section contains the following topics:

## Weighing application programming principle

**General**

Once it is configured, the weighing module is fitted with sensors and linked to a TSX XBT display. The TSX ISPY101 can operate autonomously (without a program). These outputs may be controlled without the intervention of the PLC processor program.

Programming at the level of the PLC processor makes it possible to:

- make weighing information available in order to carry out other processes or to drive other control devices,
- to dynamically modify the weighing function parameters by means of explicit commands.

**Access to measurements**

The numerical values of the (GROSS or NET) weights and flow rate are stored in 2 double word input registers (%ID). They are completed by 1 measurement Status word (%W), 1 tare value double word (%ID) and 1 calibration memory double word (%ID) (offset from zero).

i.e. a variable `IODDT_VAR1` IODDT `T_WEIGHING_ISPY101`.

The table below lists the weighing numerical values transmitted by the weighing function.

| Symbol | Register address | Meaning of the register |
|---|---|---|
| IODDT_VAR1.WEIGHT | %IDr.m.0.0 | Weight value (GROSS or NET) |
| IODDT_VAR1.FLOW_RATE | %IDr.m.0.2 | Flow |
| IODDT_VAR1.MEASURE_DATA | %IWr.m.0.4 | Measurement state: stability, zero ... |
| IODDT_VAR1.TARE | %IDr.m.0.5 | Tare value |
| IODDT_VAR1.OFFSET_MEM_VAL | %IDr.m.0.7 | Recalibration memory (zero offset) |

These data are returned automatically to the processing unit at the start of the task associated with the channel, whether the task is in Run or in Stop mode. The data are directly accessible:

- by the application via an operator dialog (access to PLC memory image objects),
- by the terminal using the animation tables.

**Dynamic modification of the parameters**

The preset adjustment parameters may be modified automatically during the operation of the program with the explicit exchange instruction `WRITE_PARAM`.

**Example**: modification of the SO high flow and SI low flow cut-off-points.

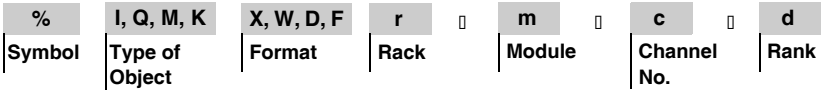
## Addressing language objects associated with the weighing Module

**At a glance**

This page introduces addressing specifics linked to weighing modules.

**Illustration**

Reminder of addressing principle:



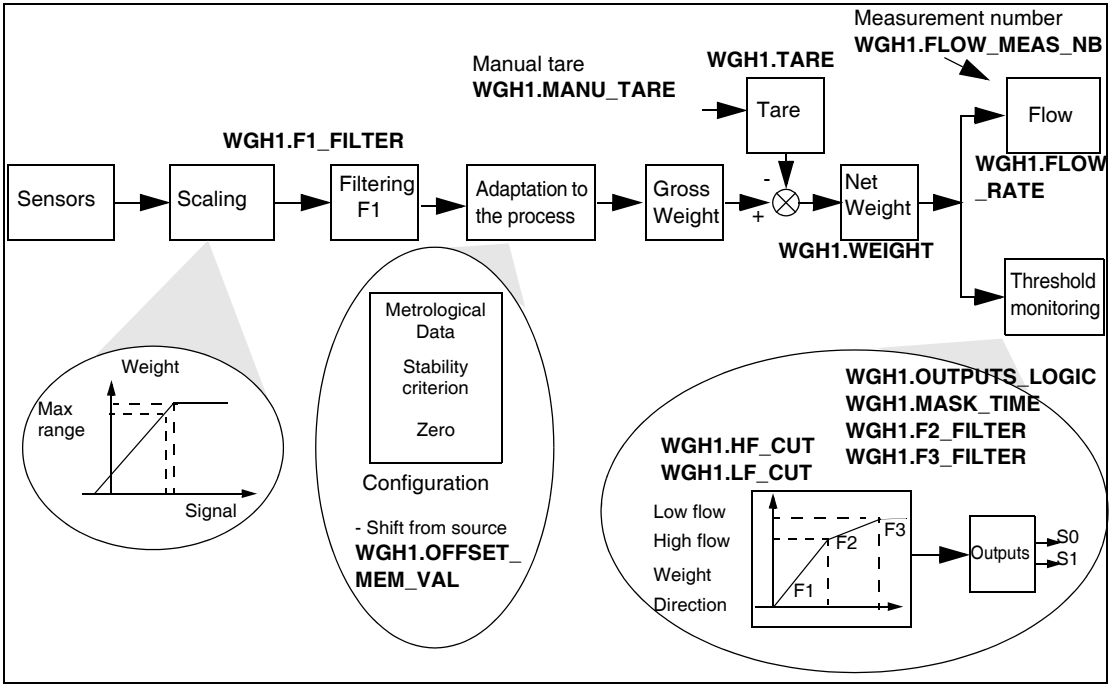

**Specific values**

The table below gives the values that are specific to weighing module objects.

| Element | Values | Comment |
|---|---|---|
| r | 0 to 7 | Rack address. |
| m | 0 to 14 | Position of the module in the rack. |
| c | 0 or MOD | 0 channel number.<br>MOD: channel reserved for module management and parameters common to all channels. |
| d | 0 to 16 or ERR | ERR: indicates a module or channel fault. |

## Description of the Main Objects Linked to the Weighing Function

**Illustration** The illustration presents the formation of the different functions executed by the module, and the associated language objects.



**Description** The following table describes the main language objects. i.e. a `WGH1` IODDT `T_WEIGHING_ISPY101`variable.

| Symbol | Address | Type of object to exchange | Role |
|---|---|---|---|
| WGH1.WEIGHT | %IDr.m.0.0 | Implicit | Weight value (gross or net). |
| WGH1.FLOW_RATE | %IDr.m.0.2 | Implicit | Flow. |
| WGH1.TARE | %IDr.m.0.5 | Implicit | Tare value. |
| WGH1.OFFSET_MEM_VAL | %IDr.m.0.7 | Implicit | Recalibration memory (zero offset). |
| WGH1.F1_FILTER | %MWr.m.0.6 | Explicit | F1 filter coefficient. |
| WGH1.MANU_TARE | %MWr.m.c.0 | Explicit | Manual tare. |
| WGH1.HF_CUT | %MDr.m.0.8 | Explicit | S0 high flow cut-off point (dosage). |
| WGH1.LF_CUT | %MDr.m.0.10 | Explicit | S1 high flow cut-off point (dosage). |
| WGH1.OUTPUTS_LOGIC | %MWr.m.0.12 | Explicit | Logic of outputs S0 and S1 (dosage). |
| WGH1.MASK_TIME | %MWr.m.0.13 | Explicit | PD mask time. |
| WGH1.FLOW_MEAS_NB | %MWr.m.0.14 | Explicit | Number of measurements used to calculate flow rate. |
| WGH1.F2_FILTER | %MWr.m.0.15 | Explicit | F2 filter coefficient. |
| WGH1.F3_FILTER | %MWr.m.0.16 | Explicit | F3 filter coefficient. |

# 8.2 Language objects and IODDT

## At a glance

**Aim of this sub-section**

This sub-section shows the general features of the language objects and IODDT of the weighing module.

**What's in this Section?**

This section contains the following topics:

## Presentation of language objects associated with the Weighing function

**General**

The weighing module has various associated IODDTs.

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

The Weighing function IODDT is of the type `T_WEIGHING_ISPY101`.

**Note:** IODDT variables can be created in two different ways:
- Using the **I/O objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) tab,
- Data Editor (See Unity Pro, Operating Modes Manual, Creation of an IODDT type data instance).

**Language object types**

In each of the IODDTs there is a set of language objects enabling them to be controlled and check their operation.

There are two types of language objects:

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

Implicit exchanges concern the inputs/ouputs of the module: measuring results, information and commands.

Explicit exchanges are used to parameterize the module and to diagnose it.

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

**At a Glance**

An integrated application-specific interface or the addition of a module automatically enhances the language objects application used to program this interface or module.

These objects correspond to the input/output images and software data of the module or integrated application-specific interface.

**Reminders**

The module inputs (%I et %IW) are updated in the PLC memory at the start of the task, the PLC being in RUN or STOP mode.

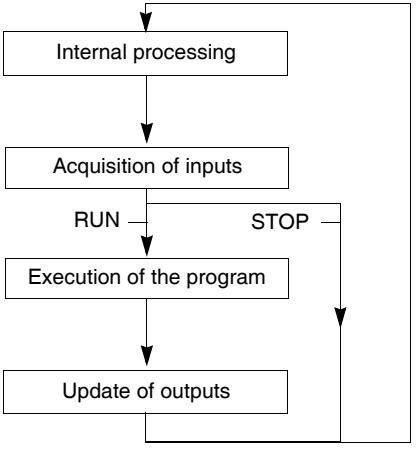
The outputs (%Q et %QW) are updated at the end of the task, only when the PLC is in RUN mode.

**Note:** When the task occurs in STOP mode, either of the following are possible, depending on the configuration selected:

- Outputs are set to fallback position (fallback mode),
- Outputs are maintained at their last value (maintain mode).

**Figure**

The following diagram shows the operating cycle of a PLC task (cyclical execution).



# Explicit Exchange Language Objects Associated with the Application-Specific Function

**Introduction**

Explicit exchanges are exchanges performed at the user program's request, and using instructions:

- READ_STS (See Unity Pro, I/O Management Manual, RESTORE_PARAM) (read status words),
- WRITE_CMD (See Unity Pro, I/O Management Manual, WRITE_CMD) (write command words),
- WRITE_PARAM (See Unity Pro, I/O Management Manual, WRITE_PARAM) (write adjustment parameters),
- READ_PARAM (See Unity Pro, I/O Management Manual, READ_PARAM) (read adjustment parameters),
- SAVE_PARAM (See Unity Pro, I/O Management Manual, SAVE_PARAM) (save adjustment parameters),
- RESTORE_PARAM (See Unity Pro, I/O Management Manual, READ_STS) (restore adjustment parameters).

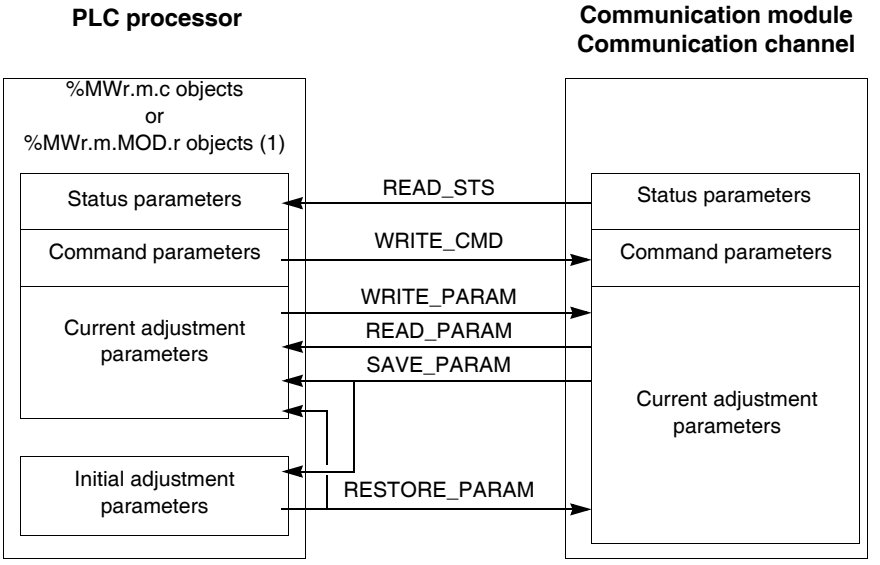
These exchanges apply to a set of %MW objets of the same type (status, commands or parameters) that belong to a channel.

**Note:**
These objects can:

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

### General Principle for Using Explicit Instructions

The diagram below shows the different types of explicit exchanges that can be made between the processor and module.




(1) Only with READ_STS and WRITE_CMD instructions.

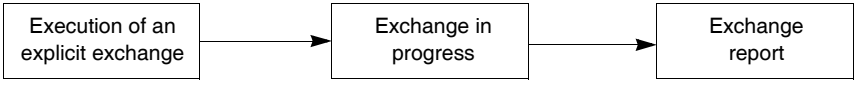
### Managing Exchanges

During an explicit exchange, it is necessary to check its performance in order that data is only taken into account when the exchange has been correctly executed.

To do this, two types of information is available:

- information concerning the exchange in progress (See *Execution Indicators for an Explicit Exchange: EXCH_STS, p. 84*),
- the exchange report (See *Explicit Exchange Report: EXCH_RPT, p. 84*).

The following diagram describes the management principle for an exchange



## Management of Exchanges and Reports with Explicit Objects

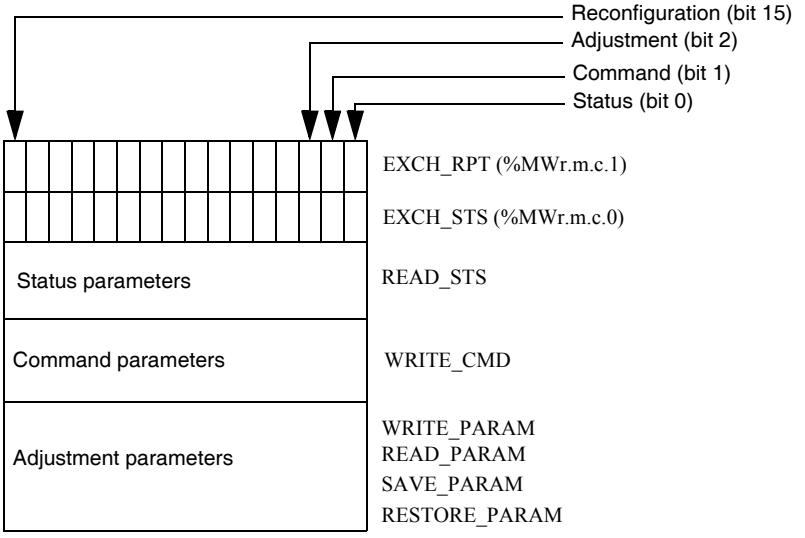
**At a Glance**

When data is exchanged between the PCL memory and the module, the module may require several task cycles to acknowledge this information. All IODDTs use two words to manage exchanges:

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

**Illustration**

The illustration below shows the different significant bits for managing exchanges:



**Description of Significant Bits**

Each bit of the words EXCH_STS (%MWr.m.c.0) and EXCH_RPT (%MWr.m.c.1) is associated with a type of parameter:

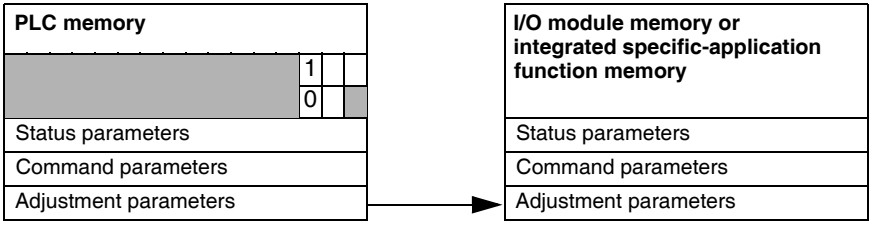
- Rank 0 bits are associated with the status parameters:
  - the STS_IN_PROGR bit (%MWr.m.c.0.0) indicates whether a read request for the status words is in progress,
  - the STS_ERR bit (%MWr.m.c.1.0) specifies whether a read request for the status words is accepted by the module channel.
- Rank 1 bits are associated with the command parameters:
  - the CMD_IN_PROGR bit (%MWr.m.c.0.1) indicates whether command parameters are being sent to the module channel,
  - the CMD_ERR bit (%MWr.m.c.1.1) specifies whether the command parameters are accepted by the module channel.
- Rank 2 bits are associated with the adjustment parameters:
  - the ADJ_IN_PROGR bit (%MWr.m.c.0.2) indicates whether the adjustment parameters are being exchanged with the module channel (via WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM),
  - the ADJ_ERR bit (%MWr.m.c.1.2) specifies whether the adjustment parameters are accepted by the module. If the exchange is correctly executed, the bit is set to 0.
- rank 15 bits indicate a reconfiguration on channel **c** of the module from the console (modification of the configuration parameters + cold start-up of the channel).

**Note: r** represents the rack number and**m** the position of the module in the rack, while **c** represents the channel number in the module.

**Note:** Exchange and report words also exist at module level EXCH_STS (%MWr.m.MOD) and EXCH_RPT (%MWr.m.MOD.1) as per IODDT type T_GEN_MOD.

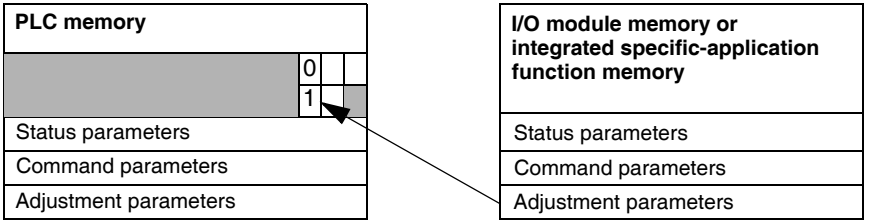
**Example**

Phase 1: Sending data by using the WRITE_PARAM instruction.




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

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




When the data is exchanged between the PLC memory and the module, acknowledgement by the module is managed by the ADJ_ERR bit (%MWr.m.c.1.2): Report (0 = correct exchange, 1 = faulty exchange).

**Note:** There is no adjustment parameter at module level.

### Execution Indicators for an Explicit Exchange: EXCH_STS

The table below shows the control bits of the explicit exchanges : EXCH_STS (%MWr.m.c.0).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| STS_IN_PROGR | BOOL | R | Reading of channel status words in progress | %MWr.m.c.0.0 |
| CMD_IN_PROGR | BOOL | R | Command parameters exchange in progress | %MWr.m.c.0.1 |
| ADJ_IN_PROGR | BOOL | R | Adjust parameters exchange in progress | %MWr.m.c.0.2 |
| RECONF_IN_PROGR | BOOL | R | Reconfiguration of the module in progress | %MWr.m.c.0.15 |

**Note:** If the module is not present or is disconnected, explicit exchange objects (READ_STS for example) are not sent to the module (STS_IN_PROG (%MWr.m.c.0.0) = 0), but the words are refreshed.

### Explicit Exchange Report: EXCH_RPT

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

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| STS_ERR | BOOL | R | Error reading channel status words<br>(1 = failure) | %MWr.m.c.1.0 |
| CMD_ERR | BOOL | R | Error during a command parameter exchange<br>(1 = failure) | %MWr.m.c.1.1 |
| ADJ_ERR | BOOL | R | Error during an adjust parameter exchange<br>(1 = failure) | %MWr.m.c.1.2 |
| RECONF_ERR | BOOL | R | Error during reconfiguration of the channel<br>(1 = failure) | %MWr.m.c.1.15 |

# 8.3 The IODDTs of the weighing module

## At a glance

**Aim of this sub-section**

This sub-section shows the various IODDTs and language objects associated with the weighing module.

**What's in this Section?**

This section contains the following topics:

## Detail of the implicit exchange objects of the IODDT of type T_WEIGHING_ISPY101

**At a glance**

The table below shows the implicit exchange objects of the IODDT of type `T_WEIGHING_ISPY101` which applies to the module TSX ISPY 101.

**Error bit %Ir.m.c.ERR**

The table below shows the meaning of the error bit `CH_ERROR` (%Ir.m.c.ERR):

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| CH_ERROR | BOOL | R | Error bit of the weighing channel. | %Ir.m.c.ERR |

**Word objects** The table below shows the various implicit exchange word objects:

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| WEIGHT | DINT | R | **Weight value** (gross or net).<br>By default, if no taring command has been executed, the weight value is expressed as a GROSS weight. It changes to NET weight as soon as the taring command is executed or a tare has been manually introduced.<br>The measurement is expressed in legal format or high resolution according to the choice created at configuration. | %IDr.m.0.0 |
| FLOW_RATE | DINT | R | **Flow**.<br>Example: %IDxy.0.2 = 450 000 means that, if the scale division is equal to $1.10^{-2}$ kg, a weight difference of 45 kg has been measured between n measurements (sampling every 20 ms).<br>The n number of measurements is defined at configuration. | %IDr.m.0.2 |
| TARE | DINT | R | **Tare value**.<br>This word allows the current tare value to be visualized in the same format as the weight. It is memorized by the module.<br>It is reset to 0 at each calibration. | %IDr.m.0.5 |
| OFFSET_MEM_VAL | DINT | R | **Recalibration memory** (zero offset)<br>This word allows the offset to be visualized currently in high resolution format. This value is memorized by the module. It is reset to 0 at each calibration. | %IDr.m.0.7 |

**Measurement state word objects**

The table below shows the various status word bits `MEASURE_DATA` (%IWr.m.0.4):

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| Q0_OUT | BOOL | R | Image of output S0. | %IWr.m.0.4.0 |
| Q1_OUT | BOOL | R | Image of output S1. | %IWr.m.0.4.1 |
| UNDERLOAD | BOOL | R | Indicator that voltage is too low. The measurement is deviating. There is a strong possibility of an error on a sensor or in the wiring. | %IWr.m.0.4.2 |
| OVERLOAD | BOOL | R | Voltage too high on module input. | %IWr.m.0.4.3 |
| SEALED_ON | BOOL | R | Sealed module. | %IWr.m.0.4.4 |
| IN_PROGR | BOOL | R | Processing in progress (Taring, Reset, etc.). | %IWr.m.0.4.5 |
| CAL_IN_PROGR | BOOL | R | Calibration during processing. | %IWr.m.0.4.6 |
| CMD_FLT | BOOL | R | Fault during command. | %IWr.m.0.4.7 |
| NET | BOOL | R | NET weight measurement. | %IWr.m.0.4.8 |
| STABILITY | BOOL | R | Measurement instability.<br>This is set when the measurement is outside the stability range during the defined time. The extent of the stability range and the time are defined during configuration. | %IWr.m.0.4.9 |
| ZERO | BOOL | R | Zero indicator. This is set when the deviation from zero is no greater than +/- 1/4 of the scale division. | %IWr.m.0.4.10 |
| ZERO_TRACK | BOOL | R | Zero tracking indicator active. | %IWr.m.0.4.11 |
| PREDEF_TARE | BOOL | R | Predefined or manual tare indicator (language element specific to module, accessible in read only). This is set when the tare is not the result of a taring command but rather an entry by the user. | %IWr.m.0.4.12 |

## Explicit exchange objects of the IODDT of type T_WEIGHING_ISPY101

**At a glance**

This part shows the explicit exchange objects of the IODDT of type `T_WEIGHING_ISPY101` which apply to the TSX ISPY101 module. It groups together the word objects, the bits of which have a particular meaning. These objects are shown in detail below.

Example of declaration of a variable: IODDT_VAR1 `T_WEIGHING_ISPY101`.

**Notes**

- Generally speaking the meaning of the bits is given for the state 1 of this bit. In specific cases each bit state is explained.
- Not all the bits are used.

**Managing exchanges: EXCH_STS**

The table below shows the meaning of the exchange control bits of the channel `EXCH_STS` (%MWr.m.0.0).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| STS_IN_PROGR | BOOL | R | Exchange of the status parameters (STATUS) in progress. | %MWr.m.0.0.0 |
| CMD_IN_PROGR | BOOL | R | Command parameters exchange in progress | %MWr.m.0.0.1 |
| ADJ_IN_PROGR | BOOL | R | Adjustment parameters exchange in progress | %MWr.m.0.0.2 |
| RECONF_IN_PROGR | BOOL | R | Reconfiguring the module. | %MWr.m.0.0.15 |

**Exchange report: EXCH_RPT**

The table below shows the meaning of the report bits `EXCH_RPT` (%MWr.m.0.1).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| STS_ERR | BOOL | R | exchange report of the status parameters (STATUS). | %MWr.m.0.1.0 |
| CMD_ERR | BOOL | R | Exchange report of the command parameters. | %MWr.m.0.1.1 |
| ADJ_ERR | BOOL | R | Exchange report of the adjustment parameters. | %MWr.m.0.1.2 |
| RECONF_ERR | BOOL | R | Configuration fault. | %MWr.m.0.1.15 |

**Operating state of the channel: CH_FLT**

The table below shows the meaning of the bits of the status word `CH_FLT` (%MWr.m.0.2), read is carried out by a `READ_STS (IODDT_VAR1)`.

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| CAL_OVERLOAD_FLT | BOOL | R | External error: Overload or underload during calibration | %MWr.m.0.2.0 |
| OVER_RANGE_FLT | BOOL | R | Range (2) overshoot fault or dynamics lower than 4.5mV at calibration. | %MWr.m.0.2.1 |
| SAT_FLT | BOOL | R | External error: saturation of the measurement circuit. | %MWr.m.0.2.2 |
| SEALED_FLT | BOOL | R | External error: sealed module, configuration refused. | %MWr.m.0.2.3 |
| INTERNAL_FLT | BOOL | R | Internal error: module failure. | %MWr.m.0.2.4 |
| CONF_FLT | BOOL | R | Configuration fault: the current module is not the one declared on configuration. | %MWr.m.0.2.5 |
| COM_FLT | BOOL | R | Communication fault with the processor. | %MWr.m.0.2.6 |
| APPLI_FLT | BOOL | R | Application fault. | %MWr.m.0.2.7 |
| PROTECT_FLT | BOOL | R | Protected module error, parameter refused: the module refuses the parameter if it influences the current value. | %MWr.m.c.2.8 |
| NON_CAL_FLT | BOOL | R | Module not calibrated. | %MWr.m.0.2.9 |
| OVERLOAD_FLT | BOOL | R | Overload error. | %MWr.m.0.2.10 |
| UNDERLOAD_FLT | BOOL | R | Underload error. | %MWr.m.0.2.11 |
| TARE_MODE | BOOL | R | Taring mode. | %MWr.m.0.2.12 |
| ZERO_MODE | BOOL | R | Zero mode. | %MWr.m.0.2.13 |
| CAL_MODE | BOOL | R | Calibration mode. | %MWr.m.0.2.14 |
| FORCED_CAL_MODE | BOOL | R | Forced calibration mode. | %MWr.m.0.2.15 |

## Explicit exchange objects of the IODDT of type T_WEIGHING_ISPY101

**At a glance**

This word type object allows commands to be sent to the weighing module by explicit exchange (`WRITE_CMD` instruction).

**Managing exchanges: CMD_TYPE**

The table below shows the meaning of the word bits `CMD_TYPE` (%MWr.m.0.3).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| MOD_CAL_SAVE_CMD | BOOL | R/W | Save calibration coefficients in module. | %MWr.m.0.3.0 |
| ZERO_LOAD_CAL_CMD | BOOL | R/W | Zero Load calibration. | %MWr.m.0.3.1 |
| STD_LOAD_CAL_CMD | BOOL | R/W | Calibration Weight Calibration (Normal condition). | %MWr.m.0.3.2 |
| CANCEL_CMD | BOOL | R/W | Cancellation of command (calibration, setting to zero, taring). | %MWr.m.0.3.3 |
| TARE_CMD | BOOL | R/W | Taring order. | %MWr.m.0.3.4 |
| ZERO_CMD | BOOL | R/W | Reset order. | %MWr.m.0.3.5 |
| GROSS_CMD | BOOL | R/W | Return to GROSS weight order. | %MWr.m.0.3.6 |
| MANU_TARE_DISP_CMD | BOOL | R/W | 3 second display of manual tare. | %MWr.m.0.3.7 |
| TH_EN_CMD | BOOL | R/W | Enable thresholds. | %MWr.m.0.3.8 |
| TH_DIS_CMD | BOOL | R/W | Disable thresholds. | %MWr.m.0.3.9 |
| FORCED_CAL_CMD | BOOL | R/W | Forced calibration (CPU -> Module). | %MWr.m.0.3.10 |
| CPU_CAL_SAVE_CMD | BOOL | R/W | Save calibration coefficients in processor. | %MWr.m.0.3.11 |
| DG_CAL_CMD | BOOL | R/W | Standard load calibration in degraded condition (Standard load < 70 % of maximum weight). | %MWr.m.0.3.12 |

## Explicit exchange objects of the IODDT of type T_WEIGHING_ISPY101

**At a glance**

This part shows the explicit exchange objects of the IODDT of type T_WEIGHING_ISPY101 which apply to the TSX ISPY101 module. It groups together word objects. These objects are shown in detail below.

Example of declaration of a variable: IODDT_VAR1 of type T_WEIGHING_ISPY101.

**Word objects**

The table below shows the meaning of the words. The requests used are those associated with the parameters (READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| STD_LOAD | DINT | R/W | Standard load weight for the calibration command. | %MDr.m.0.4 |
| F1_FILTER | INT | R/W | F1 filter coefficient. | %MWr.m.0.6 |
| MANU_TARE | INT | R/W | Manual tare value. | %MWr.m.0.7 |
| HF_CUT | DINT | R/W | S0 high flow cut-off point (dosage). | %MDr.m.0.8 |
| LF_CUT | DINT | R/W | S1 high flow cut-off point (dosage). | %MDr.m.0.10 |

**Output logic**

The table below shows the meaning of the word bits OUTPUTS_LOGIC (%MWr.m.0.12). The requests used are those associated with the parameters (READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| WEIGH_UNWEIGH | BOOL | R/W | Low Flow mask time. | %MWr.m.0.12.0 |
| Q0_OR_QOQ1 | BOOL | R/W | Number of measurements used to calculate flow rate. | %MWr.m.0.12.1 |

**Word objects**

The table below shows the meaning of the words. The requests used are those associated with the parameters (READ_PARAM, WRITE_PARAM, SAVE_PARAM, RESTORE_PARAM).

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| MASK_TIME | INT | R/W | Low Flow mask time. | %MWr.m.0.13 |
| FLOW_MEAS_NB | INT | R/W | Number of measurements used to calculate flow rate. | %MWr.m.0.14 |
| F2_FILTER | INT | R/W | F2 filter coefficient. | %MWr.m.0.15 |
| F3_FILTER | INT | R/W | F3 filter coefficient. | %MWr.m.0.16 |

## Details of the Language Objects of the IODDT of Type T_GEN_MOD

**At a Glance**

All the modules of Premium PLCs have an associated IODDT of type T_GEN_MOD.

**Observations**

- In general, the meaning of the bits is given for bit status 1. In specific cases an explanation is given for each status of the bit.
- Not all bits are used.

**List of Objects**

The table below presents the objects of the IODDT:

| Standard symbol | Type | Access | Meaning | Address |
|---|---|---|---|---|
| MOD_ERROR | BOOL | R | Module error bit | %Ir.m.MOD.ERR |
| EXCH_STS | INT | R | Module exchange control word. | %MWr.m.MOD.0 |
| STS_IN_PROGR | BOOL | R | Reading of status words of the module in progress. | %MWr.m.MOD.0.0 |
| EXCH_RPT | INT | R | Exchange report word. | %MWr.m.MOD.1 |
| STS_ERR | BOOL | R | Fault when reading module status words. | %MWr.m.MOD.1.0 |
| MOD_FLT | INT | R | Internal error word of the module. | %MWr.m.MOD.2 |
| MOD_FAIL | BOOL | R | Internal error, module failure. | %MWr.m.MOD.2.0 |
| CH_FLT | BOOL | R | Faulty channel(s). | %MWr.m.MOD.2.1 |
| BLK | BOOL | R | Terminal block fault. | %MWr.m.MOD.2.2 |
| CONF_FLT | BOOL | R | Hardware or software configuration fault. | %MWr.m.MOD.2.5 |
| NO_MOD | BOOL | R | Module missing or inoperative. | %MWr.m.MOD.2.6 |
| EXT_MOD_FLT | BOOL | R | Internal error word of the module (Fipio extension only). | %MWr.m.MOD.2.7 |
| MOD_FAIL_EXT | BOOL | R | Internal fault, module unserviceable (Fipio extension only). | %MWr.m.MOD.2.8 |
| CH_FLT_EXT | BOOL | R | Faulty channel(s) (Fipio extension only). | %MWr.m.MOD.2.9 |
| BLK_EXT | BOOL | R | Terminal block fault (Fipio extension only). | %MWr.m.MOD.2.10 |
| CONF_FLT_EXT | BOOL | R | Hardware or software configuration fault (Fipio extension only). | %MWr.m.MOD.2.13 |
| NO_MOD_EXT | BOOL | R | Module missing or inoperative (Fipio extension only). | %MWr.m.MOD.2.14 |

# 8.4 Description of the commands conveyed by program

## At a glance

**Subject of this section**

This section describes the different commands that can be executed by program.

**What's in this Section?**

This section contains the following topics:

## Sending commands to the weighing module by program

**General**

Commands are sent to the module using the `WRITE_CMD` instruction with the following syntax: `WRITE_CMD (%CHr.m.0)`

This instruction sends the order to the module and awaits its acknowledgement. This wait may require several task cycles.

**Note:** The module only interprets one command at a time. If a command is requested while the previous command is being processed, the new command is refused. There should never be more than one bit at 1 in the command word.

**Monitoring parameter recognition**

As the module may require several task cycles to recognize commands, two memory words are standardized to control the %MWr.m.0.0 and %MWr.m.0.1 exchanges

The first word %MWr.m.0.0 indicates a current exchange.

The second word %MWr.m.0.1 gives the exchange report.

The following table describes the objects used for controlling the sending of commands to the module.

| Address | Meaning (for the bit at state 1) |
|---|---|
| %MWr.m.0.0.1 | Indicates that the command has been sent to the module. |
| %MWr.m.0.1.1 | Shows if the command is accepted by the module. |
| %MWr.m.0.2.7 | Signals that a command or parameter has been refused (application fault). |

## How to perform a tare mode by program

**At a glance**

This function involves bringing the value of the measured net weight to zero when a load, or tare, is placed on the load holder.

It therefore supports movement of the measurement with an offset value, in order to make it conform to the user's expected value.

When no tare mode operation has been carried out, the net weight is the same as the gross weight.

**Note:**
- When changing configuration, all tares are deleted. Any Tare mode command cancels all tares entered in manual mode and resets the "manual" tare indicator to zero,
- Similarly, an return to gross weight order allows you to delete all tare modes. It does not need any acceptance condition.

**Execution conditions for the tare mode**

The acceptance conditions for performing the Tare mode command are the following:

- the measurement is stable,
- the measurement is below the maximum range,
- the measurement is strictly positive.

## Procedure

The following table describes the procedure for executing a tare mode operation.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Input the `WRITE_CMD` instruction while positioning the tare mode order (%MWr.m.0.3.4 = 1). | - |
| 2 | Confirm the execution, with the application in RUN mode. | The module switches to tare mode and sends the %Iwr.m.0.4.5 = 1 **Processing _ in_ progress** report.<br>Acquires the tare.<br>**Note:** The value of the weight is measured and stored in the associated %IDr.m.0.5. It will be deducted from all further gross weight measurements to determine the net weight.<br>End of acquisition: **Processing_in_progress** = 0 |
| 3 | Monitor the command's smooth execution: State of **Processing_in_progress**: %IWr.m.0.4.5 | The module stays in the **Processing_in_progress** state for as long as the acceptance conditions are not met or until it receives an order to cancel the command. |

## Summary of the data used

The table below provides a table of the data used for a tare mode.

| Type | Role | Associated data |
|---|---|---|
| Command | Tare mode order | %MWr.m.0.3.4 |
| Display | Tare value | %IDr.m.0.5 |
| | Tare mode in progress | %IWr.m.0.4.5 |

**Example**

The following example in instruction list language describes a tare mode order being sent to the weighing module in slot 2, rack 0.

```
LD TRUE
S  %MW 2.0.3.4
[WRITE_CMD   (%CH2.0)]
```

Executing the program involves:

| Phase | Description |
|---|---|
| 1 | Sending the command. |
| 2 | Setting the %MW2.0.0.1 bit to 1, showing that the command is being sent. |
| 3 | This bit remains at 1 until the module sends a report. The bit then goes back to 0. The exchange report bit is then relevant. |
| 4 | The exchange report bit %MW2.0.1.1 rises to 1 if there is a problem during the exchange. The 0 value shows that the command has been accepted by the module. |

**Note:**
%IW2.0.4.5 stays at 1 (Processing in progress) for as long as the acceptance conditions are not met (waiting for measurement stability, for example).
The status channel's application fault bit is at 1 (module executing command). As for all commands, the order can be cancelled by sending the "cancel command in progress" command.

## How to reset the value of the weight to zero by program

**At a glance**

This function consists in moving the measured weight value to zero. The zero indicator is then put into position.

It is controlled by the command bit **Set to Zero**. The correction carried out on the measurement is stored in the word %IDr.m.0.7, in a high resolution format. It can be saved by the application. This parameter is reset to zero on each calibration.

**Note:** On changing configuration, any setting to zero is deleted.

**Conditions for carrying out zero setting**

The acceptance conditions for performing the zero setting command are the following:

- the measurement is in gross weight,
- the measurement is stable,
- the measurement is included in the preset zero range extent.

**Procedure**

The following table describes the procedure for executing a tare mode operation.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Input the `WRITE_CMD` while setting the order **Set to zero** (%MWr.m.0.3.5 = 1). | - |
| 2 | Confirm the execution, with the application in RUN mode. | The module switches to **set to zero** and sends the **Processing _in_progress** report %IWr.m.0.4.5 = 1.<br>The module proceeds to the acquisition of the measurement and memorizes the new value in the recalibration memory %IDr.m.0.7.<br>**Processing_in_progrees** = 0 signifies the end of the procedure. |
| 3 | Monitor the command's smooth execution: State of **Processing_in_progress**: %IWr.m.0.4.5 | The module stays in the **Processing_in_progress** state for as long as the acceptance conditions are not met or until it receives an order to cancel the command. |

**Summary of the data used**

The table below provides the data used for resetting to zero.

| Type | Role | Associated data |
|---|---|---|
| Command | Reset order. | %MWr.m.0.3.5 |
| Display | Recalibration memory | %IDr.m.0.7 |
| Exchange | Processing in progress | %IWr.m.0.4.5 |

**Example**

The following example in instruction list language describes a **Reset order** being sent to the weighing module in slot 2, rack 0.

```
LD TRUE
S  %MW 2.0.30.5
[WRITE_CMD   (%CH2.0)]
```

Executing this order involves:

| Phase | Description |
|---|---|
| 1 | Sending the command. |
| 2 | Setting the %MW2.0.0.1 bit to 1, showing that the command is being sent. |
| 3 | This bit remains at 1 until the module sends a report. The bit then goes back to 0. The exchange report bit is then relevant. |
| 4 | The exchange report bit %MW2.0.1.1 rises to 1 if there is a problem during the exchange. The 0 value shows that the command has been accepted by the module. |

**Note:**
%IW2.0.4.5 stays at 1 (Processing in progress) for as long as the acceptance conditions are not met (waiting for measurement stability, for example).
The status channel's application fault bit is at 1 (module executing command).
As for all commands, the order can be cancelled by sending the **cancel command in progress** command.

## How to return to gross weight measurement via the program

**At a glance**

This function cancels the tare value so that the current weight value is in gross weight.

The current weight is stored in the word %Idr.m..0.0, in the format set in configuration.

**Execution conditions for the tare mode**

This command does not require any particular execution conditions.

**Procedure**

The following table describes the procedure for executing a return to measurement in gross weight.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Input the `WRITE_CMD` while programming the tare mode order (%MWr.m.0.3.6 = 1). | - |
| 2 | Confirm the execution, with the application in RUN. | The module switches to "return to gross weight" mode.<br><br>The module then sets the tare to zero.<br><br>The NET = 0 (%Iwr.m..0.4.8=0) flag shows the end of the procedure. |
| 3 | Check that the command has been performed correctly: State of the NET flag: %IWr.m.0.4.8 | - |

**Summary of the data used**

The table below provides the data used for resetting to zero.

| Type | Role | Associated data |
|---|---|---|
| Command | Return to gross weight order | %MWr.m.0.3.6 |
| Display | Measured weight | %IDr.m.0.0 |
| | Value of the Tare in progress | %IDr.m.0.5 |
| Exchange | Processing in progress | %IWr.m.0.4.5 |
| | Gross Weight | %IWr.m.0.4.8 = 0 |

## How to display the manual tare via the program

**At a glance**

This function supports display of the manual tare on the display panel for 3 seconds.

**Execution conditions for the tare mode**

For this command, a manual tare must have already been configured.

**Procedure**

The following table describes the procedure for displaying the manual tare.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Input `WRITE_CMD` while positioning the display order (%MWr.m.0.3.7 = 1). | - |
| 2 | Confirm the execution, with the application in RUN mode. | The module manages its data normally. The values displayed on the **TSX XBT N410** display panel show the manual tare value. |
| 3 | After the 3 second delay, the display panel reverts to its current values. | - |

**Summary of the data used**

The table below provides the data used for resetting to zero.

| Type | Role | Associated data |
|---|---|---|
| Command | Tare display order | %MWr.m.0.3.7 |
| Display | Manual tare | The data on the display panel show the manual tare. |

## How to enable or disable Thresholds by program

**At a Glance**

These functions are primarily used to coordinate the output command in relation to the processor-managed mechanism.

The threshold check option must first be enabled in the configuration screen.

**Operating Principle**

Action on outputs is performed from the command **Enabling of thresholds**. Once this command has been executed, the threshold check cycle starts.

A disable command is used to stop the threshold check cycle in progress and authorize a new threshold enabling command.

If need be, this command will also reset outputs S0 and S1 to 0.

**Enabling Procedure**

The following table describes the procedure for validating thresholds.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Make the necessary changes to the threshold values, output logic and mask time. | - |
| 2 | Set the threshold enabling order (%MWr.m.0.3.8 = 1). | - |
| 3 | Launch enable thresholds using the `WRITE_CMD` instruction. | The module interprets the request, sets the S0 and S1 outputs and ensures the conformity of the image bits:<br>• %IWr.m.0.4.0 current position of S0.<br>• %IWr.m.0.4.1 current position of SI. |

**Disabling Procedure**

The following table describes the procedure for disabling thresholds.

| Step | Action | Behavior of the module |
|---|---|---|
| 1 | Set the disable thresholds (%MWr.m.0.3.9 = 1). | - |
| 2 | Launch threshold disabling using the `WRITE_CMD` instruction. | The module sets outputs to idle and image bits to 0. |

**Summary of the data used**

The table below provides the data used for enabling and disabling thresholds.

| Type | Role | Associated data |
|---|---|---|
| Command | Threshold enabling order | %MWr.m.0.3.8 |
| | Threshold disabling order | %MWr.m.0.3.9 |
| Display | Current flow | %IDr.m.0.2 |
| | High flow threshold | %MDr.m.0.8 |
| | Low flow threshold | %MDr.m.0.10 |
| | Output logic | %MWr.m.0.12 |
| | LF mask time | %MWr.m.0.13 |
| | S0 current position | %IWr.m.0.4.0 |
| | S1 current position | %IWr.m.0.4.1 |

# 8.5 Modifying the parameters by program

## At a glance

**Subject of this section**

This section describes how to dynamically modify the application's parameters by program.

**What's in this Section?**

This section contains the following topics:

## Modifying the parameters by program

**Principle**

You can command the modification of certain parameters by program, in order to automatically adapt the measurements to the processing applications.

**Example:** modifying the tare value by program if several types of product are to be weighed with different packagings.

**List of adjustable parameters**

The following parameters may be modified by program:

| Adjustable parameters | Corresponding data |
|---|---|
| F1 filter coefficient | %MWr.m.0.6 |
| "Manual" tare value | %MWr.m.0.7 |
| Cut-off points (thresholds) | %MDr.m.0.8 and %MDr.m.0.10 |
| Logic of S0 and S1 outputs | %MWr.m.0.12 |
| LF mask time | %MWr.m.0.13 |
| Number of measurements used to calculate flow rate | %MWr.m.0.14 |
| F2 filter coefficient | %MWr.m.0.15 |
| F3 filter coefficient | %MWr.m.0.16 |

**Possible Actions**

You can:

- modify an adjustment parameter by program,
- send adjustment parameters to the module,
- control the module's parameter recognition,
- read the adjustment parameter value in the module and thereby update the PLC memory,
- save the adjustment parameters,
- restore the value of saved parameters to the PLC memory.

**Instructions Used**

The instructions used to do these operations are as follows:

| Instruction | Function carried out |
|---|---|
| WRITE_PARAM (%CH r.m.0) | Send the parameter contents of the previous table to the weighing module. |
| READ_PARAM (%CH r.m.0) | Reads the adjustment parameters in the module and updates the above table. |
| SAVE_PARAM (%CH r.m.0) | Saves the adjustment parameter values in the memory zone of the processor. These parameter values will be the ones used when the PLC is started from cold. |
| RESTORE_PARAM (%CH r.m.0) | Allows the adjustment parameters to be reloaded with the values entered at module configuration or at the last SAVE_PARAM. |

The module can process several adjustments simultaneously.

## Instructions used for adjustments

**General**

To be able to carry out adjustment operations, you must be able to access the module's own data.

Access is gained using the following instructions.

**Sending adjustment parameters to the module**

Parameters are sent from the module channel using the `WRITE_PARAM` instruction, with the following syntax:

```
WRITE_PARAM (%CH r.m.0)
```

This instruction sends the contents of the parameters to the module and waits for its acknowledgement. This may necessitate several task cycles.

**Monitoring parameter recognition**

As the module can take several task cycles to acknowledge values, two memory words are used to monitor the exchanges: **%MWr.m.0.0** and **%MWr.m.0.1**

- The first word **%MWr.m.0.0** shows that there is an exchange in progress,
- The second word **%MWr.m.0.1** gives the exchange report,
- rank 2 bits are associated with the adjustment parameters:
  - the **%MWr.m.0.0.2** bit shows that the adjustment parameters are sent to the module,
  - the %**MWr.m.0.1.2** bit specifies whether the adjustment parameters are accepted by the module.

**Example**

Writing the module's parameters in slot 2 of rack 0,

`WRITE_PARAM (%CH2.0)` involves:

- sending the adjustment parameters,
- setting the **%MW2.0.0.2** bit to 1, showing that the adjustment parameters are being sent. This bit remains at 1 until the module sends a report. The bit then goes back to 0. The exchange report bit is then relevant,
- the **%MW2.0.1.2** exchange report bit is set at 1 if there is a problem during the exchange. The 0 value shows that the data has been accepted by the module.

**Reading the adjustment parameters**

The READ_PARAM instruction is used to read the module's adjustment parameters and to update the PLC's memory. It is particularly useful after a WRITE_PARAM that the module has not accepted. Reading the adjustment parameters can take several task cycles.

The adjustment parameters are read from the module's channel using the READ_PARAM instruction, with the following syntax:

```
READ_PARAM (%CH r.m.0)
```

**Saving the adjustment parameters**

The SAVE_PARAM Instruction allows you to copy the current values of the module's adjustment parameters in the backup zone defined in the processor memory. The backup zone is not accessible from the language.

This instruction can take several task cycles to be executed. The module's adjustment parameters are saved using the SAVE_PARAM instruction, with the following syntax:

```
SAVE_PARAM (%CH r.m.0)
```

**Restoring the saved adjustment parameters**

The RESTORE_PARAM instruction supports restoring of the saved adjustment parameter values in the processor's memory and in the module.

The adjustment parameters are restored from the module using the RESTORE_PARAM instruction, with the following syntax:

```
RESTORE_PARAM (%CH r.m.0)
```

## Description of parameters adjustable by program

**Description**

The following table describes the parameters that can be adjusted by program, using the `WRITE_PARAM` instruction.

| Word | Role | Description |
|---|---|---|
| %MWr.m.0.6<br>%MWr.m.0.15<br>%MWr.m.0.16 | Filter Coefficients, | The admissible filter coefficient values are between 0 and 19. |
| %MWr.m.0.7 | "Manual" tare value. | The admissible "manual" tare values are between 0 and 65535. They cannot exceed the maximum range. |
| %MD r.m.0.8<br>%MD r.m.0.10 | Cut-off points (thresholds). | S0 high flow cut-off point.<br>S1 low flow cut-off point.<br>The threshold values allowed lie between 0 and the maximum range in high resolution format..<br>If a threshold check has not been defined at configuration, no detection processing is done. The value of these threshold is zero by default.<br>**Note:**<br>• when weighing: GD < MW < Maximum weight,<br>• in downweighing: MW < GD < Maximum weight<br>The module carries out a consistency check of the threshold values. If this logic is not respected, the thresholds will be refused. |
| %MWr.m.0.12 | Output logic. | • %MWr.m.0.12.0:<br>  • 0: Weighing,<br>  • 1: Downweighing.<br>• %MWr.m.0.12.1:<br>  • 0: S0 then S1,<br>  • 1: S0 then S1 then S1. |
| %MWr.m.0.13 | PD mask time. | The admissible values are between 0 and 15 per 1/10 second step (0 = 0 s, 1 = 0.1 s, 2 = 0.2 s, ...). |
| %MWr.m.0.14 | Number of measurements for flow. | Admissible values are values 2, 4, 8, 16, 32 or 64. |

## Reading configuration parameters.

**General**

The set of parameters entered during the configuration of the module is accessible by program in read only.

These parameters are coded in 3 words of the %KW constant area.

**Coding of the maximum weight**

The maximum weight reading, configured for the measuring channel, may be accessed with the word %KWr.m.0.0.

**Coding the measurement unit**

Reading the unit and the scale division, configured for the measuring channel, is accessible with the word %KWr.m.0.1.

The measurement unit is coded on 3 bits of the least significant byte.

%KWr.m.0.1: Least significant byte


The table below describes the coding of the measurement unit.

| Bits 0 to 2 | Corresponding unit | Role |
|---|---|---|
| 0 | g | gram |
| 1 | kg | kilogram |
| 2 | t | tonne (metric) |
| 3 | lb | pound (=453g) |
| 4 | oz | once (=28.35g) |
| 5 | <none> | no unit |

**Scale division coding**

The reading of the unit and the scale division configured for the measuring channel, is accessible with the word %KWr.m.0.1.

**Note:** The scale division is always defined in the same unit as that of the measurement.

The measurement unit is coded on 5 bits of the most significant byte.

%KWr.m.0.1 of the most significant byte

| bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|

The following table describes the coding of the scale division.

| Bits 8 to 12 | Scale division value | Bits 8 to 12 | Scale division value |
|---|---|---|---|
| 0 | 0.001 | 11 | 5 |
| 1 | 0.002 | 12 | 10 |
| 2 | 0.005 | 13 | 20 |
| 3 | 0.01 | 14 | 50 |
| 4 | 0.02 | 15 | 100 |
| 5 | 0.05 | 16 | 200 |
| 6 | 0.1 | 17 | 500 |
| 7 | 0.2 | 18 | 1000 |
| 8 | 0.5 | 19 | 2000 |
| 9 | 1 | 20 | 5000 |
| 10 | 2 | | |

**Coding of the stability, of zero, thresholds, outputs and format**

The reading of the extent of range and the stability time and the extent of range from zero and the activity of zero tracking, of the overload threshold, the utilization of the outputs and the format of the weight values, configured for the channel measurement are accessible with the memory word %KWr.m.0.2.

Coding of the word %KWr.m.0.2.

%KWr.m.0.2 : Least significant byte

| bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 |
|---|---|---|---|---|---|---|---|
| Stability time | | | | Extent of stability | | | |

%KWr.m.0.2 of the most significant byte

| bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 |
|---|---|---|---|---|---|---|---|
| Format | Outputs | Zero Tracking | Manual tare. | Extent. Zero | Supply sensor | Overload | |

The following table describes the coding of the extent of stability (bits 0 to 2).

| Value read | Equivalence in 1/4 of the scale division |
|---|---|
| 0 | 2 |
| 1 | 3 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

The following table describes the coding of the stability time (bits 4 to 5).

| Value read | Equivalence in seconds |
|---|---|
| 0 | 0.4 |
| 1 | 0.5 |
| 2 | 0.7 |
| 3 | 1 |

The following table describes the coding of the overload (bits 8 to 9).

| Value read | Type of overload selected |
|---|---|
| 0 | Maximum weight + 9 scale divisions |
| 1 | Maximum weight + 2% of the maximum weight |
| 2 | Maximum weight +5% of the maximum weight |

The following table describes the coding of the other parameters. These parameters are each coded on one bit of the word %KWr.m.0.2.

| Bit no. | Parameter | bit at 0 | bit at 1 |
|---|---|---|---|
| 11 | Recalibration range | +/-2% of the maximum weight | +/-5% of the maximum weight |
| 12 | Tare predefined | No predetermined tare | Tare predefined |
| 13 | Activity of the zero tracker | Inactive | Active |
| 14 | Utilization of the outputs | Not used | Used |
| 15 | Format | Legal (physical unit with fixed point) | High resolution ( hundredth of a fixed point physical unit) |

# Debugging the weighing function

# 9

## At a glance

**Subject of this chapter**

This chapter introduces the debugging screen and describes the functions available for debugging the application.

**What's in this Chapter?**

This chapter contains the following topics:

## Introduction to the debug function of the weighing module

**Introduction**

The Debug function makes it possible for the weighing module of the application to display the parameters of each of its channels (channel status etc.), to access the diagnostic and the adjustment of the selected channel.

The function also gives access to the module diagnostics in the case of a fault.

**Note:** This function is only accessible in online mode.

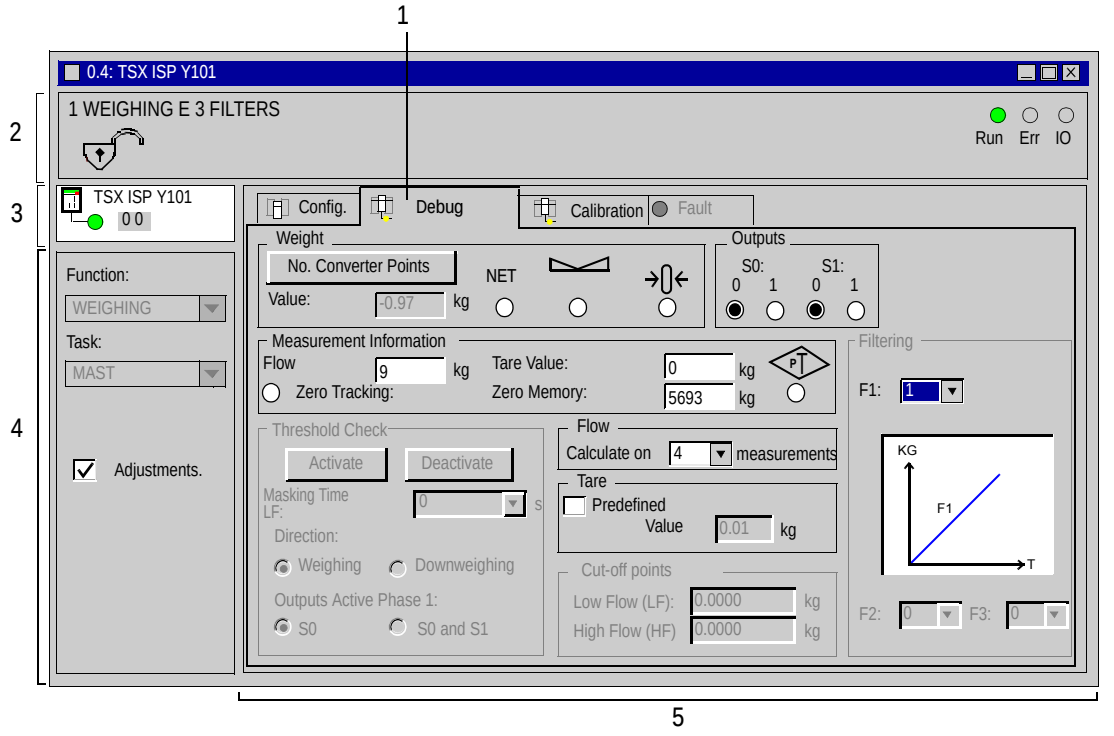
## Description of the weighing application function debug screen

**At a glance**

The debug screen (See Unity Pro, Operating Modes Manual, Access to the module configuration editor) can be used to display weighing information and adjust certain parameters.

**Illustration**

This screen can be only accessed in online mode.



**Description** The following table shows the various parts of the debug screen and their functions.

| Address | Element | Function |
|---|---|---|
| 1 | **Tab** | The tab in the foreground indicates the current mode (**Debug** for this example). Each mode may be selected by the corresponding tab. |
| 2 | **Module** zone | Shows the abbreviated name of the module.<br>In online mode, this zone also contains the three LEDs **Run**, **Err** and **IO**. |
| 3 | **Channel** field | Is used:<br>● By clicking on the reference number, to display the tabs:<br>  ● **Description** which gives the characteristics of the device.<br>  ● **I/O Objects** (See Unity Pro, Operating Modes Manual, I/O Objects Tab for a Module) which is used to presymbolize the input/output objects.<br>  ● **Fault** which shows the device faults (in online mode).<br>● To select the channel,<br>● To display the **Symbol**, name of the channel defined by the user (using the variable editor). |
| 4 | **General parameters** zone | Comprises:<br>● the dropdown **Function** menu,<br>● the dropdown **Task** menu defining the task (**MAST** or **FAST**) in which the implicit exchange objects of the channels will be exchanged.<br>● the **Adjustments** check box: allows you to access the adjustment functions. When this box is ticked, an extra zone is added to the debug screen giving access to the parameters. |
| 5 | **Display** and **adjustment** zones | Display weighing information and adjust certain module parameters. |

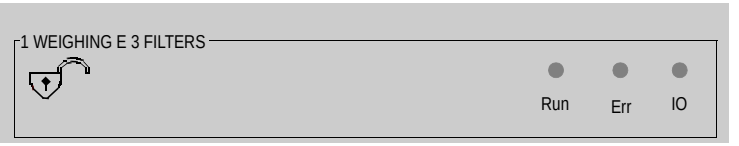
## Description of the module zone of the debug screen

**At a glance**

This zone displays general information on the module status.

**Illustration**

This zone on the screen informs you about the state of the module.




**Description**

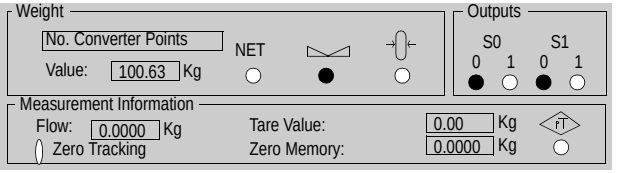
The following table describes the different elements of the module screen zone and channel state.

| Address | Description |
|---|---|
|  or  | Indicates whether the module is closed (locked padlock) or not. |
| Run | Indicator lit: normal operation<br>Indicator unlit: module error or switched off |
| Err | Indicator lit: internal error, module broken down<br>Indicator blinking: communication error, absent, invalid or faulty application<br>Indicator unlit: no error |
| I/O | Indicator lit: External error:<br>• overload or underload fault during calibration,<br>• range overshoot error,<br>• measurement error<br>• Closed module: configuration refused<br>Indicator blinking: loss of communication with the processor<br>Indicator unlit: no error |

## Description of the display zone of the debugging screen

**Illustration**

This zone is the dynamic display zone containing important information connected to weighing.




**Description**

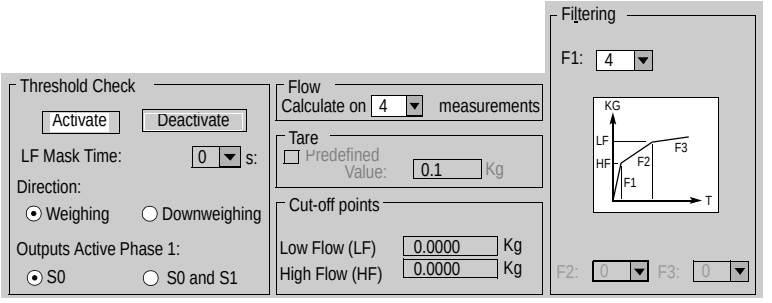
The following table describes the different elements belonging to the weighing debugging screen's display zone.

| Zone | Field | Description |
|---|---|---|
| Weight | No. Converter Points | By default, the screen displays the current weight value. Clicking the **No. Converter Points** button will allow you to switch to points mode during the next disconnection of the PLC.<br>The display weight will be reapplied when the PLC next runs. |
| | Value | The current weight value in a defined unit. If there is a fault on the measurement circuit detected by the module or when it is in calibration mode, the **ERR** note is displayed on the screen. |
| | NET | The net weight indicator is positioned if the module returns NET weight information, otherwise this relates to gross weight. |
| |  | The "stable measurement" flag specifies that the measurement is in the defined stability range. |
| |  | The zero zone indicator is activated when the measured weight is in zero format (+/- 1/4 of the scale indicator). |
| Outputs | | The supplied indications correspond to the physical state of outputs S0 and S1. |
| Measurement Information | | This zone displays:<br>• the flow value, it is indicated by the unit measurement,<br>• the current tare value,<br>• the memory zero value corresponding to zero shift from the last calibration,<br>• the PT indicator specifies the tare value which has been manually introduced and not measured,<br>• the Zero Tracking indicator shows that the function has been parameterized. |

## Description of the parameter setting zone

**Illustration**

This zone allows you to modify the adjustment parameters.



**Description**

It gives access to the modification and display of the following parameters:

| Address | Description |
|---|---|
| Filtering (See *How to modify measurement input Filter(s), p. 62*) | You have the possibility to modify for each phase the filter coefficient value of the measurement input.<br>You can choose a value from 0 to 19.<br>**Note:** The stronger the filter (value from 1 to 11), the longer the response time. |
| Flow (See *How to modify the flow calculation, p. 64*) | You have the possibility to modify the number of measurements for the flow calculation.<br>The listed choices carry the values 2, 4, 8 16, 32 and 64. |
| Tare (See *How to modify the tare, p. 65*) | You have the possibility to introduce a predetermined tare by checking the corresponding box and filling in this tare value in the defined unit. |
| Threshold Check (See *How to modify the threshold check, p. 66*) | |
| The following parameters are displayed only if the **Threshold monitoring** option has been activated during configuration. Recognition of all parameters is effected from the confirmation command in the **Edit** menu. | |
| Activate | This key activates the threshold check monitoring cycle. |
| Deactivate | This key deactivates the threshold check monitoring cycle and positions outputs S0 and S1 into fallback mode. |
| LF mask time | Lets the user modify the mask time during switch to low flow. |
| Weighing/ Downweighing direction | Lets you modify the threshold recognition. |
| Phase 1 active outputs | Lets you choose the active outputs during the first dosage phase. |
| Low flow (PD) and high flow (GD) cut off points | Lets you modify the threshold values. |

# Calibrating the measurement string

# 10

## At a glance

**Subject of this chapter**

This chapter describes how to calibrate the measurement string.

**What's in this Chapter?**

This chapter contains the following topics:

## Introduction to the Calibration Function

**General**

The calibration of the analog measurement system consists in making a weight value correspond to an electrical signal transmitted by sensors.

This adaptation is made on site when the product is set up. It is necessary to ensure the measurement's validity.

**Note:** The calibration function, forced or not, is only accessible in offline mode with the PLC in Run mode.

**Calibration rules**

Any non-calibrated module is in channel fault (can be seen in the debug screen or on the module, by the flashing of channel 0).

The first calibration must be complete:

1. Zero Load
2. Standard Load
3. Save

otherwise the information returned means nothing.

It is not possible to carry out a calibration if the PLC processor is fitted with A Flash-Eprom memory card (**TSX MFP P 128K** or **TSX MFP P 224K** or **TSX MFP P 384K**).

Calibration can be redone throughout the module's life. The electronics characteristics do not require regular recalibration. However, legal constraints or the application's mechanical characteristics may require calibration, particularly for commercial transactions.

**Note:** Calibration is independent of the configured filter, but acknowledges Metrological and Stability Information parameters.

**Calibration Type**

You can choose one of the following 4 calibration types:

- normal calibration (The calibration function must be performed with a standard load calibration that is greater than or equal to 70% of the maximum weight),
- graded calibration (if for various reasons calibration cannot performed under the conditions previously described),
- forced calibration :
  - CPU -> Module: being able to restore adjustments made to a different module in the event of a maintenance concern or duplication
  - Module -> CPU: being able to make the processor parameters conform with the parameters of a calibrated module that is connected to a new slot.
- The software calibration requires:
  - the version of the weighing module is $\geq$ 2.1,
  - not to use the standard load,
  - to modify the module configuration.

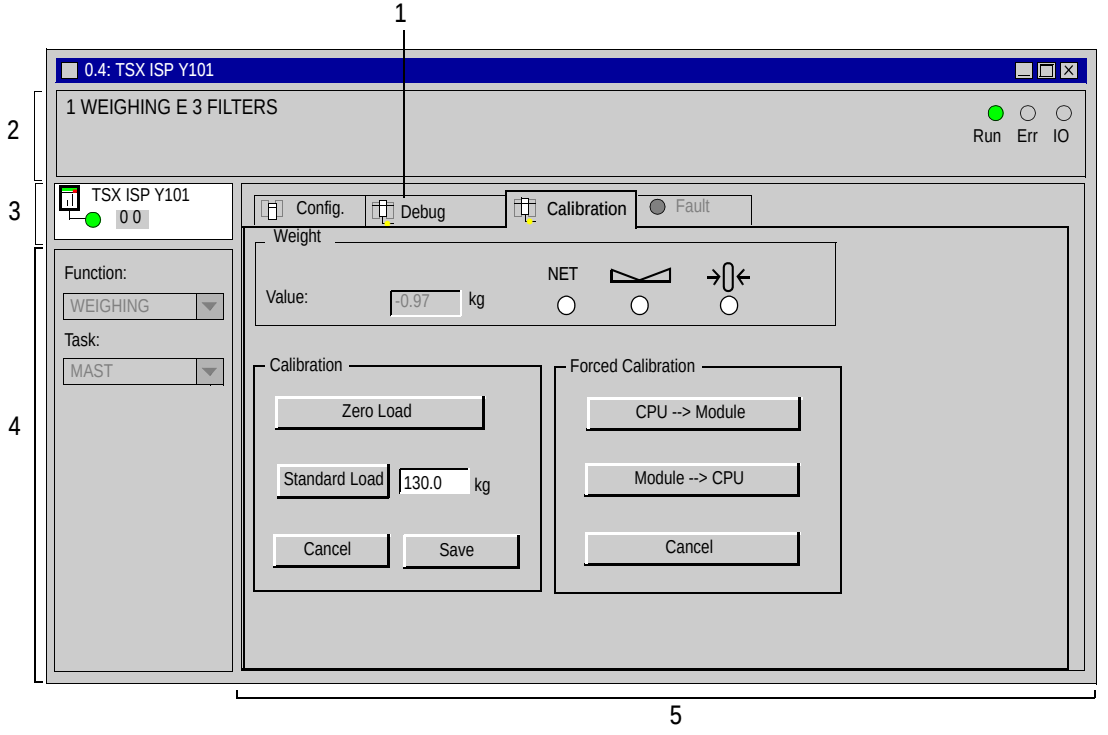
## Description of the calibration screen

**At a glance**

The adjustment screen provides access to the calibration commands.

**Illustration**

This screen can be only accessed in online mode.



**General Description**

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

| Address | Element | Function |
|---|---|---|
| 1 | **Tab** | The front tab indicates the current mode (**Calibration** in this example). Each mode may be selected by the corresponding tab. |
| 2 | **Module** zone | Shows the abbreviated name of the module.<br>In online mode, this zone also contains the three LEDs **Run**, **Err** and **IO**. |
| 3 | **Channel** field | Is used:<br>● by clicking on the module reference, to display the **Description** tab, which contains the technical specifications and the **Fault** tab, which contains the module's faults,<br>● to select the channels you wish to configure. |
| 4 | **General parameters** zone | Comprises:<br>● the dropdown **Function** menu,<br>● the dropdown **Task** menu defining the task (**MAST** or **FAST**) in which the implicit exchange objects of the channels will be exchanged. |
| 5 | **Display** and **calibration** zones | Display weighing information (See *Description of the display zone of the debugging screen, p. 122*) and the calibration commands (See *Calibrating the Analog Measurement System, p. 130*). |

## Calibrating the Analog Measurement System

**At a glance**

Calibration can be carried out on a Unity Pro station connected to the PLC using the calibration screen.

It can also be done using an operator dialog, which uses Unity Pro language instructions.

**Note:** The procedure can be stopped at any time by pressing **Cancel**. The module reverts to the previous parameters. The current calibration parameters are therefore lost.

The procedure is only enabled if the module has been calibrated correctly.

If there is a measurement saturation problem, the new parameters cannot be saved. Either the error must be corrected or the procedure be cancelled, using **Cancel**.

**Procedure** This table describes the procedure for calibrating the analog measurement system.

| Step | Action | Result |
|---|---|---|
| 1 | Switch the rack on. | The product is initialized, carries out self-tests and receives its configuration. |
| 2 | Access the calibration screen:<br>**1.** select **Configuration** in the project browser.<br>**2.** click on the module's slot<br>**3.** select the **Calibration** tab.<br>**Note:** The processor must be in RUN and the terminal in online mode. | - |
| 3 | Check that the counter output is empty. | - |
| 4 | Click the **Zero Load** button to carry out zero load calibration (recognized by the load receiver). | This phase requires around 20 seconds.<br>The **Zero Load** button switches to reverse image during this phase and an hour glass appears.<br>The module switches to channel fault and all measurements are invalid.<br>The status of the flag %Iwr.m.0.4.6 **Calibration_in_progress** changes. The module indicates the acquisition of the zero weight reference and processes the reports. |
| 5 | Place calibration weight. | - |
| 6 | Enter the calibration weight value in the "Standard Load" field (this value is equal to the maximum weight) and click on the **Standard Load** button.<br>If the command is disabled, an error message indicates the type of problem encountered. | This phase requires around 20 seconds.<br>The module checks the standard load weight against the maximum weight.<br>The status of the flag %Iwr.m.0.4.6 **Calibration_in_progress** changes.<br>The module acquires the standard load weight reference, processes and positions the report. |
| 7 | Click the **Save** button to recognize the parameters resulting from calibration. | The module and the processor recognize and save the parameters resulting from calibration.<br>During the write phase, the measurement remains in channel fault. This fault disappears as soon as writing is finished (current channels faults and calibration in progress disappear). The measurement is valid. |

## Calibrating the Analog Measurement System by Program

**General**

Several language elements are used to implement and supervise the calibration mechanism.

The calibration screen facilitates the procedure, but it can also be performed by program using reserved data.

**Procedure**

Program the following operations to perform a calibration by program.

| Step | | Action | Result |
|---|---|---|---|
| 1 | Zero weight | Enter `WRITE_CMD` while setting the calibration order of the channel using the zero weight (%MWr.m.0.3.1=1). | The status of the %IWr.m.0.4.6 **Calibration in progress** indicator changes.<br>This operation enables you to determine the **Offset** parameter. |
| 2 | Standard Load | Load the standard load weight value in the %MDr.m.0.4 word | - |
| 3 | | Enter `WRITE_CMD` while setting the calibration order of the channel using the standard load weight (%MWr.m.0.3.2=1, or %MWr.m.0.3.12=1 for a graded calibration). | The %IWr.m.0.4.6 **Calibration in progress** indicator changes status.<br>This operation is used to determine the **Gain** parameter. |
| 4 | Save the parameters in the module | Enter `WRITE_CMD` while setting the save order of the calibration in the module (%MWr.m.0.3.0=1). | - |
| 5 | Copy the module parameters in the CPU | Enter `WRITE_CMD` while setting the save order of the calibration in the processor (%MWr.m.0.3.11=1). | - |

## Summary of the data used

The table below shows the data involved in a calibration.

| Type | Role | Associated data |
|---|---|---|
| Command type | Save calibration in the module | %MWr.m.0.3.0 |
| | Zero weight | %MWr.m.0.3.1 |
| | Standard load weight (Normal) | %MWr.m.0.3.2 |
| | Forced calibration (CPU -> Module) | %MWr.m.0.3.10 |
| | Save calibration in the processor | %MWr.m.0.3.11 |
| | Standard load weight (Graded) | %MWr.m.0.3.12 |
| Command parameter | Value of standard load weight | %MDr.m.0.4 |
| Report | Calibration in progress (Normal) | %IWr.m.0.4.6 |
| | Instability | %IWr.m.0.4.9 |
| | Overload or underload during calibration | %%MWr.m.0.2.0 |
| | Non-calibrated module | %MWr.m.0.2.9 |
| | Calibration mode | %MWr.m.0.2.14 |
| | Forced calibration mode | %MWr.m.0.2.15 |

## How to achieve a forced calibration

**At a glance**

This function responds to the needs of speedy maintenance.

Forced calibration allows calibration values from a weighing module to be transferred to the central unit, and vice versa.

**Note:** This action cannot be reversed. Once the transfer has occurred, it is not possible to cancel the command.

**Note:** This function is only accessible in online mode, with the PLC in Run.

**Operating Mode**

Transferring the CPU to the weighing module is always authorized once the CPU has the requisite calibration parameters for the desired slot.

Transferring the weighing module to the central unit requires the module to be calibrated (not forced calibration).

**Procedure**

This table describes the procedure for achieving forced calibration.

| Step | Action |
|---|---|
| 1 | Switch the rack on. |
| 2 | Access the calibration screen:<br>**1.** Select **Configuration**<br>**2.** Click on the module's slot<br>**3.** Select the **Calibration** tab. |
| 3 | In the **Forced Calibration** field, click either **CPU -->Module** or **Module -->CPU**, according to the desired transfer direction. |

**Note:** The procedure is not enabled unless the transfer is performed correctly. If there is a problem, click the **Cancel** button in the **Forced Calibration** field.

## Performing a Forced Calibration by Program

**General**

Several language elements are used to implement and supervise the calibration mechanism.

The calibration screen facilitates the procedure, but it can also be performed by program using reserved data.

**Procedure**

Carry out the following operations to perform a forced calibration by program.

| Direction of copy | Action | Result |
|---|---|---|
| CPU -> Module | Enter `WRITE_CMD` while positioning the save order of the calibration in the module (%MWr.m.0.3.10=1). | This operation is used for example when replacing a module. It enables you to automatically restore the calibration parameters in the module (gain, offset and converter configuration. |
| Module -> CPU | Enter `WRITE_CMD` while positioning the save order of the calibration in the processor (%MWr.m.0.3.11=1 | This operation enables you to automatically restore the calibration parameters in the processor when, for example, you are using a module that is placed in a new slot.<br>This operation is only possible if the module is calibrated. |

## How to achieve a software calibration

**At a glance**

Calibration can be carried out on a Unity Pro station connected to the PLC using the calibration screen.

**Note:** Software calibration cannot be used for regulated weighing applications.

**Procedure** This table describes the procedure for calibrating the analog measurement system.

| Step | Action | Result |
|---|---|---|
| 1 | Switch the rack on. | The product is initialized, carries out self-tests and receives its configuration. |
| 2 | Access the calibration screen:<br>**1.** click on the module's slot,<br>**2.** select the **Configuration** tab:<br>• Set the Maximum Range (MR) to a value higher than the sum of the sensors' weight ratings,<br>• Set the Low Flow (LF) to a value equal to the average of the sensors' sensitivities,<br>• If needed, set the Scale Division (e) to less than 1.<br>**3. validate**,<br>**4.** select the **Calibration** tab.<br>**Note:** The processor must be in RUN and the terminal in online mode. | - |
| 3 | Check that the counter output is empty. | - |
| 4 | Press the **Zero Load** button. | The **Zero Load** button switches to reverse image during this phase.<br>The module switches to channel fault and all measurements are invalid.<br>The status of the flag %Iwr.m.0.4.6 **Calibration_in_progress** changes.<br>The module acquires the standard load weight reference, processes and positions the report. |
| 5 | In the **Standard Load** field, enter the value PN.xx where PN corresponds to the sensors' weight ratings and xx differs by 0.<br>If the command is disabled, an error message indicates the type of problem encountered | The status of the flag %Iwr.m.0.4.6 **Calibration_in_progress** changes.<br>The module calculates the theoretical calibration coefficients. |
| 6 | Click the **Save** button to recognize the parameters resulting from calibration. | The module and the processor recognize and save the parameters resulting from calibration.<br>During the write phase, the measurement remains in channel fault. This fault disappears as soon as writing is finished (current channels faults and calibration in progress disappear). The measurement is valid. |
| 7 | Select **Configuration** tab, reset (MR), (LF) and (e) to their initial configuration values and **validate**. | |

# Protecting the adjustments

# 11

## At a glance

**Subject of this chapter**

This chapter describes how to protect the adjustments done during the previous phases.

**What's in this Chapter?**

This chapter contains the following topics:

## Protection of the weighing parameter settings

**General**

Any weighing instrument which can be used for commercial transactions must be approved. The parameters associated with the measurement must therefore be protected. It should not be possible to introduce into an instrument, via the interface, instructions or data likely to:

- falsify the weighing results displayed,
- change an adjustment factor.

**Note:** protection by sealing aims to guarantee measurement conformity, so that the parameters accessible only apply to the exploitation aspects of the module information by the mechanism.

**Effect of Protecting the Configuration Parameters**

There are two types of information. Information, which can be protected (if a module is sealed, this type of information will be available in read only) and information with free access (Read and Write)
The table below identifies the characteristics of this information according to the protection put in place.

| Functions | Without sealing | With sealing |
|---|---|---|
| Task | Modifiable | Modifiable |
| Flow/ Calculation on n measurements | Modifiable | Modifiable |
| Tare/ Predefined | Modifiable | Modifiable |
| Threshold checking/ Active | Modifiable | Modifiable |
| Threshold checking/ Direction | Modifiable | Modifiable |
| Threshold checking/ Active outputs | Modifiable | Modifiable |
| Threshold Checking/ Cut-off points | Modifiable | Modifiable |
| Threshold checking/ LV Mask Time | Modifiable | Modifiable |
| Unit | Modifiable | Non modifiable |
| Max Range (MR) | Modifiable | Non modifiable |
| Scale Division | Modifiable | Non modifiable |
| Overload Threshold | Modifiable | Non modifiable |
| Filtering/ Coefficient | Modifiable | Modifiable |
| Data format | Modifiable | Non modifiable |
| Stability/Extent of Range | Modifiable | Non modifiable |
| Stability/Time | Modifiable | Non modifiable |
| Zero/Zero tracking | Modifiable | Non modifiable |
| Zero/Recalibration range | Modifiable | Non modifiable |

The information word %Iwr.m.0.4.4 (to 1) tells you if the measurement is protected.

**Consequences of Protection**

- A sealed module that receives a different configuration to the one memorized (before being switched off prior to the movement of the rider) is refused.
- In this case the module is seen as missing in the PLC diagnostics, but sends a weight to the display.
- A sealed module will not accept a new calibration request

**Note:** using the file allows you to keep a paper record of the configuration

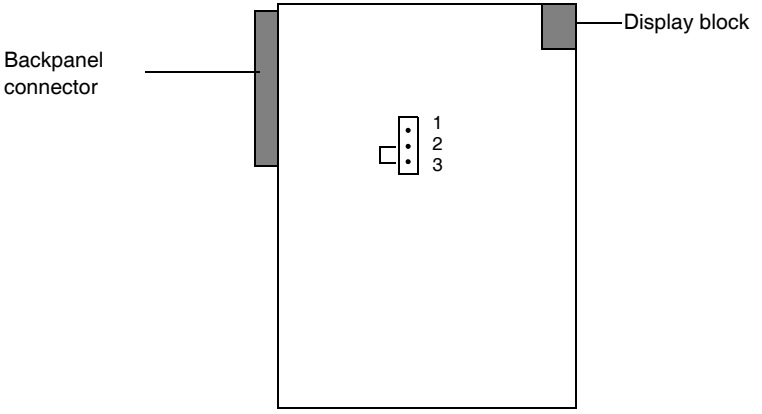
## How to protect the adjustments

**Necessary conditions**

The calibration and adjustment operations must be completed.

**Illustration**

The following illustration shows how to position the riders in order to protect the adjustments.




**Procedure**

The following table describes the operation of protecting the adjustments (leading).

| Step | Action |
|---|---|
| 1 | Take the module out of the PLC rack (the rack can remain switched on). |
| 2 | Remove the module's casing (use a TORX type screwdriver for this). |
| 3 | Place the rider in position 2-3 as shown in the illustration. |
| 4 | Put the module back into its casing. |
| 5 | Replace the module in the rack in its previous position. |

## Legal metrology and regulations

**EU approval**

The set consisting of: load holder + sensors + module can be considered as an IPFNA (non automatic weighing instrument).

As such, and to be able to use it for commercial transitions, it has been approved by the EU.

If it is only used for internal processes, the displayt must have an identification plate mentioning:

| | |
|---|---|
| Trademark | Max = |
| Type of instrument | e = |
| Serial number | |
| 'All transitions prohibited' | |

If it is used for regulated uses (e.g. commercial transitions), the display must have a identification plate, showing:

| | |
|---|---|
| Trademark | Max = |
| Type of instrument | Min = |
| Serial Number | e= |
| Number and date of EU approval of type | Number 97.00.620.016.0<br>29th September 1997 |

Moreover, it must receive a first check on leaving the factory, as well as regular on-site monitoring by a licensed body. Generally, monitoring takes place once a year, and this is the responsibility of the owner.

**Approval of the model**

**Measurement and control device for filling machine and a discontinuous counter**

This IPFNA can be supplemented by the specific software applications 'Filling Machine' or 'Discontinuous counter'. As such, it has passed national approvals, as a measurement and automatic control device for filling machines and discontinuous counters.

It is therefore up to the manufacturer of the measurer or discontinuous counter to get a complete approval of any automatic weighing instruments made up in this way, in the most straightforward conditions possible.

It is also up to the manufacturer of the machine to install the identification plate and to present the machine for its first check, when necessary.

**Approval of a continuous counter model**

Associated with a weighing table, it is authorized as a continuous counter device.

Except for when used for commercial transitions, the identification plate shows:

| | |
|---|---|
| - Mark | QMax = |
| - Type | dt = |
| - Serial number | |
| 'All transitions prohibited' | |

When used for commercial transitions, the identification plate shows:

| | |
|---|---|
| - Mark | QMax = |
| - Type | dt = |
| - Serial number | |
| Weighed products: | |
| - Max= | L = |
| - v= | d = |

It must be checked. The first phase of the first check is done in the factory on the complete instrument uncoupled from its conveyor, by means of a movement simulator; the other phases are carried out on the complete instrument.

**Class of appliance**

With average precision, the appliance covers the range from the minimum (500 scale divisions) up to 6000 scale divisions. These instruments can be authorized or unauthorized to carry out commercial transitions. If it is unauthorized, 'PROHIBITED FOR ALL TRANSACTIONS' must be written on the appliance's front panel.

# Operating a weighing application

# 12

## At a glance

**Subject of this chapter**

This chapter describes the tools that allow you to operate a weighing application.

**What's in this Chapter?**

This chapter contains the following topics:

## Ways of displaying weighing information

### Description

The following table describes the different ways of displaying weighing information.

| Ways | Description |
|---|---|
| Module display panel **TSX XBT N410** (See *Description of the display report, p. 147*) | Automatically displays the weight measurement without any prior programming. |
| Debugging screen (See *Description of the display zone of the debugging screen, p. 122*) | Displays all information relevant to the weighing and allows the modification of certain parameters. |
| Animation tables | All information about the measurement can be accessed as PLC variables and can be displayed in the animation tables. |
| Operation screen | It is possible to create runtime screens using the weighing language objects in order to display the information required to run the application. |
| Supervision | The weighing language objects can be conveyed and operated by a supervision system. |

### Language Objects

The following language objects are used for operating the weighing application.

| Displayed data | Object address |
|---|---|
| Protected module | %MWr.m.0.2.8 (Explicit Exchange Object) |
| Non-calibrated module | %MWr.m.0.2.9 (Explicit Exchange Object) |
| Weight value | %IDr.m.0.0 |
| Net weight flag | %IWr.m.0.4.8 |
| Stability flag | %IWr.m.0.4.9 |
| Zero indicator | %IWr.m.0.4.10 |
| Discrete S0 output status | %IWr.m.0.4.0 |
| Discrete S1 output status | %IWr.m.0.4.1 |
| Flow | %IDr.m.0.2 |
| Tare value | %IDr.m.0.5 |
| Recalibration memory | %IDr.m.0.7 |
| Zero indicator follower | %IWr.m.0.4.11 |
| Predetermined tare flag | %IWr.m.0.4.12 |

## Description of the display report

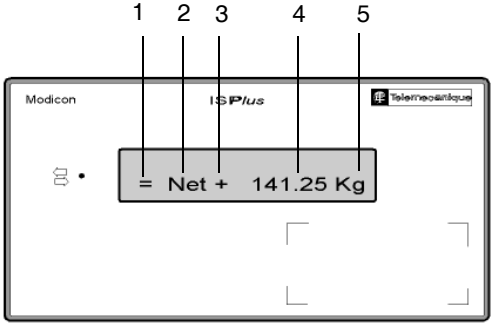
**General**

The **TSX XBT N410** display module provides measurement indications (see **TSX XBT N410** installation documentation).

This display appears automatically without prior programming.

**Illustration**

The following illustration shows **TSX XBT N410** display.




**Note:** On the **TSX XBT N410**, space is left available for the punched identification plate to meet metrological legal requirements.

**Description of the display**

All valid measurements are transmitted to the display in physical units with a fixed number of decimal places, every 100 ms.

The following table describes the indications that may appear on the display in normal operation.

| Address | Indication | Description |
|---|---|---|
| 1 | = | The measurement is stable. |
| | none | The measurement is not stable (the stability criteria are defined in the configuration). |
| 2 | Net | The measurement indicates a Net weight. |
| | none | The measurement indicates a gross weight. |
| 3 | + | The measurement is positive. |
| | 0 | The measurement is approximately 0 (between -1/4 et +1/4 scale division). |
| | - | The measurement is negative:<br>● if the associated numerical value blinks: the measurement is between -9 scale divisions and -1/4 scale division,<br>● if no associated numerical value is displayed: the measurement is below -9 scale divisions. |
| 4 | 141.25 | Numerical value of the weight. |
| 5 | kg | Symbol of measurement mass unit: g for gram, kg for kilogram, lb for pound, oz for ounces and t for metric tonne. |

**Note:** The serial link test is carried out when the weighing module is powered up. For this the **TSX XBT N410** display module must be connected to the **TSX ISP Y101** when the PLC is powered up.

## Error messages

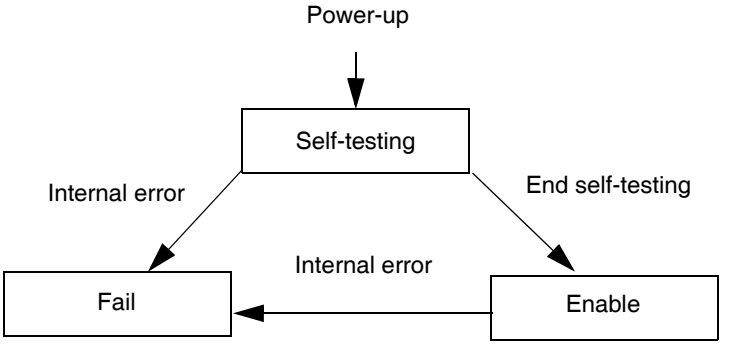
The following table describes the error indications which may appear on the display.

| Indication | Description |
|---|---|
| ---------- | The measurement is not valid, a channel fault is detected. |
| >>>>> | An overload is detected. |
| <<<<< | An underload is detected. |
| `Time out` | The display no longer receives data from the weighing module. |
| `Checksum error` | A problem was detected during the power-up. At power-up, the **TSX XBT N410** performs a test on its resources. In operation, all the information received is checked. In the event of a problem, the `Checksum` error is displayed. |

## Weighing module operating modes

**Operation**

The following illustration describes the operation of the module.




**Behavior on encountering an error**

During power-up, the module carries out its own self-tests (REPROM , RAM, Link display, etc.).

If an error is detected at the end of these tests, the module switches to fall back mode, the outputs are at 0.

Similarly, if, when operating normally, an internal malfunction (error in RAM, CDG, etc.) is detected in the module, the outputs are positioned to 0 and the display shows dashes on the screen (----).

**Behavior on power outage**

On power outage the machine parameters are saved (Tare mode, Zero offset, etc.) whereas the operating parameters are lost (Thresholds, number of measurements used to calculate flow rate, etc.).

# Diagnostics of the weighing application

# 13

## At a glance

**Subject of this chapter**

This chapter allows you to diagnose the errors detected in the weighing application.

**What's in this Chapter?**

This chapter contains the following topics:

## How to access the Diagnostics function of the weighing module

**At a glance**

The Module diagnostics function displays errors when they occur, classified according to category:

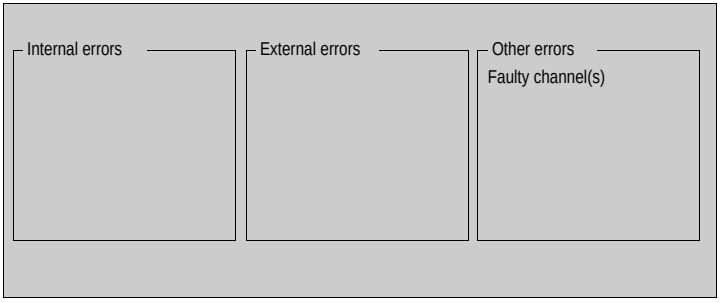
- internal errors (module breakdown, running self-test),
- external errors (terminal block fault),
- other errors (configuration error, module missing or switched off, faulty channel(s) (details in channel diagnostics)).

A module error is indicated by certain LEDs changing to red, such as:

- in the rack-level configuration editor:
  - the module position LED,
- in the module-level configuration editor:
  - the **Err** and **I/O** LEDs, depending on the type of error,
  - the **Fault** tab LED.

**Procedure**

The table below shows the procedure for accessing the Module diagnostics screen.

| Step | Action |
|---|---|
| 1 | Access the module configuration screen. |
| 2 | Click on the module reference in the channel zone and select the **Fault** tab.<br>**Result:** The list of module errors appears.<br>  <br><br>**Note:** It is not possible to access the module diagnostics screen if a configuration error, major breakdown error or module missing error occurs. The following message appears on the screen: `The module is missing or different from that configured for this position.` |

## How to access the channel Diagnostics function of the weighing module

**At a glance**

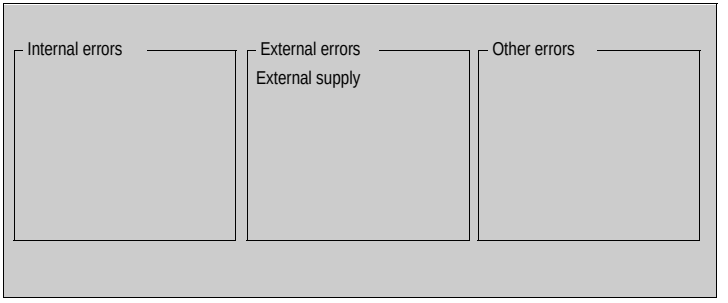
The Channel diagnostics function displays errors when they occur, classified according to category:

- internal errors (breakdown of channel),
- external errors (link or sensor link power supply error),
- other errors (terminal block error, configuration error, communication error).

A faulty channel is indicated when the **Fault** LED, situated on the **Fault** tab of the configuration editor, switches to red.

**Procedure**

The table below shows the procedure for accessing the Channel diagnostics screen.

| Step | Action |
|---|---|
| 1 | Access the weighing module configuration screen. |
| 2 | Select the **Fault** tab.<br>**Result:** The list of channel faults appears.<br>  <br>**Note:** Access to the channel diagnostics information is also possible by program (instruction READ_STS). |

# Examples of the weighing program

# 14

## At a glance

**Subject of this chapter**

This chapter provides programming examples for a weighing application.

**What's in this Chapter?**

This chapter contains the following topics:

## Example of a tare mode

**Description of the example**

This example emphasizes the running of a weighing process by focusing on the essential operations to be carried out:  it deals with carrying out a switch into NET weight (tare mode).

**Program**

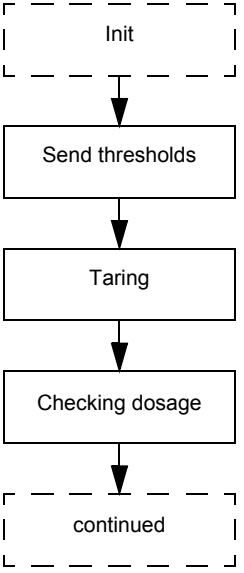
The %M101 bit is used for this action. Its positioning causes the gross weight, which is currently measured as the weighed tare, to be acknowledged, then it causes the display to be switched to NET mode.
The WEIGHT_1 variable of type WEIGHING_ISPY101 is associated with channel 0 of the weighing module situated in rack 6.

```
(* Closed weighing module, slot 6 *)
 ! (* waiting for tare mode conditions *)
      IF %M100 THEN
          IF NOT WEIGHT_1.STS_ERR AND NOT WEIGHT_1.CMD_ERR
          THEN
              SET (%M101);
              RESET (%M100);
          ELSE
              RETURN;
          END_IF;
      END_IF
 ! (* Tare mode *)
      IF %M101 THEN
          (* send tare mode order *)
          IF NOT WEIGHT_1.STS_ERR AND NOT WEIGHT_1.CMD_ERR AND NOT %M102
THEN
              WEIGHT_1.MOD_CAL_SAVE_CMD:=0;
              SET (TARE_CMD);
              WRITE_CMD (WEIGHT_1);
              SET %M102;
          END_IF;
          (* tare mode ended and OK *)
          IF NOT WEIGHT_1.STS_ERR AND NOT WEIGHT_1.CMD_ERR THEN
              WEIGHT_1.MOD_CAL_SAVE_CMD:=0;
              RESET (%M101);
              RESET (%M102);
              SET (%M103);
          ELSE
          (* tare mode refused => error *)
              IF NOT WEIGHT_1.STS_ERR AND WEIGHT_1.CMD_ERR THEN
                  SET (%M200);
                  WEIGHT_1.MOD_CAL_SAVE_CMD:=0;
                  RESET (%M101);
                  RESET (%M102);
                  RESET (WEIGHT_1.CMD_ERR);
                  SET %M100;
              END_IF;
          END_IF;
      END_IF;
```

## Example of metering flow

**Description of the example**

The following example uses a weighing module in slot 2 of the PLC.
It describes a metering flow cut up in stages as on the diagram below.



**Program**

The program is processed in structured text. The WEIGHT_1 variable of type WEIGHING_ISPY101 is associated with channel 0 of the weighing module situated in rack 6.

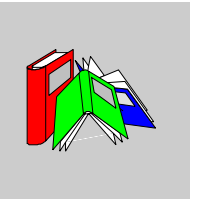
**Main Program**

```
(* ///////// Send thresholds ///////////*)
L100:
      IF NOT %M99 THEN
          JUMP L120;
      END_IF;

(*Loading and sending thresholds *)
      IF RE (%M99) THEN
          WEIGHT_1.HF_CUT:=%MD230;(* S0 High Flow cut-off point*)
          WEIGHT_1.LF_CUT:=%MD232;(* S1 Low Flow cut-off point*)
          WRITE_PARAM (WEIGHT_1);JUMP L120;
      END_IF;

(*Transmission in progress*)
      IF WEIGHT_1.ADJ_IN_PROGR THEN
          JUMP L120;
      END_IF;

(*command accepted*)
      IF NOT WEIGHT_1.ADJ_ERR THEN
          RESET (%M99);
      END_IF;

(*FIN INIT CYCLE*)
      L120:

(* //////// TARE MODE PHASE      (%MW100 =4) ////////////// *)

L260:
      IF %MW100<>4 THEN
          JUMP L300;
      END_IF;

(*Tare mode request *)
      IF %M72 THEN
          RESET (%M72);
          %MW270:2:=4;
      END_IF;

(*Management of commands *)
SR8; (* %MW270  informs you of the type of tare mode command 4 *)

(*Waiting for tare mode return*)
      IF %MW270=-1 AND %MW271=-1 THEN
          %MW100:=5;
          SET (%M72);
          JUMP L800;
END_IF;
```

## Program (continued)

```
   (* ////////////// DOSAGE PHASE   (%MW100 = 5) ////////// *)
   L300:
            IF %MW100<>5 THEN
               JUMP L340;
            END_IF;
         (*Enable thresholds *)
            IF %M72 THEN
               RESET (%M72);
               %MW270:2:=8;
            END_IF;
         (*Management of commands *)
            SR8();(* %MW270 = type of  threshold enable command 8 *)
         (*Waiting for command mode return*)
            IF %MW270>=0 OR  %MW271>=0 THEN
               JUMP L800;
            END_IF;
         (*Checking outputs to skip one after the other*)
            IF NOT WEIGHT_1.Q0_OUT AND NOT WEIGHT_1.Q1_OUT THEN
               %MW100:=6;
               SET (%M72);
               JUMP L800;
            END_IF;
         (*PHASE 6  continued  *)
   L340:
            IF %MW100<>6 THEN
               JUMP L380;
            END_IF;
   L800:
   SUBROUTINE SR8 :
         (* Send request for the module*)
            IF %MW270>=0 THEN (* %MW270  informs you of the order to carry out
 *)
               %M0:16:=0;
               SET (%M0[%MW270]);
               WEIGHT_1.MOD_CAL_SAVE_CMD:=%M0:16;
               %MW271:=%MW270;
               %MW270:=-1;
               WRITE_CMD (WEIGHT_1);
               RETURN;
            END_IF;
         (*Command in progress? *)
            IF WEIGHT_1.CMD_IN_PROGR OR WEIGHT_1.IN_PROGR THEN
               RETURN;
            END_IF;
         (*command accepted? *)
            IF NOT WEIGHT_1.CMD_ERR AND NOT WEIGHT_1.APPLI_FLT THEN
               %MW270:2:=-1;
            ELSE
               %MW270:=%MW271;
 END_IF;
```

# Glossary



## !

**(Weighing Instrument) Indicator Device** — Part of load measuring device from which direct reading of result is obtained (**TSX XBT N410**).

## C

**Calibration** — Graduates a piece of measuring apparatus.

**Configuration** — The configuration gathers together the data which characterizes the machine (invariant) and which is necessary for the module to operate. All this information is stored in the constant PLC %KW zone. The PLC application cannot modify them.

**CPU** — Central Processing Unit: generic name used for Schneider Electric processors

## D

**Debug** — Debugging is a Unity Pro service which is used to check the module directly when it is online.

**Discrete** — Discrete I/Os

## E

**Explicit exchanges** Exchanges between the CPU and the specific application modules carried out by the Unity Pro program in order to update data specific to the module.

## G

**Gross Weight** Indication of the load weight on an instrument when no tare or predefining device has been used.

## I

**I/O** Inputs/Outputs.

**IODDT** Input/Output Derived Data Type.

## L

**Lead Sealing** Sealing a piece of apparatus with lead. The positioning of a rider in the weighing module ensures this function.
The objective of this device is to guarantee measurement conformity. The accessible parameters only have influence on the aspects of the mechanism's exploitation of module information. the unit, weight, scale division, etc, are read only).

**Limit Load (Lim)** The maximum static load that can be supported by the instrument, without permanently altering its meteorological qualities.

**Load Receiver Device** Part of instrument that will receive the load.

## M

**Maximum Weight (Max)** Maximum weighing capacity, not taking account of the additive capacity of the tare.

**Metrology** The science of weights and measures.

**Minimum Weight (Min)** Load value under which weighing results can be marred by a relative error that is too large.

## N

**Net Weight (Net)** Weight indication of a load placed on an instrument after a tare device has been used.
Net weight = Gross weight - Tare weight

**Non-Automatic Functioning Weighing Instruments** Weighing instruments that require the intervention of the operator during the weighing process, in order to deposit loads on the load receiver device and retrieve loads from it, for example, as well as to obtain the result. These instruments allow the weighing result to be directly observed, either displayed or printed out. The two possibilities are covered by the word "indication".

## O

**Operating Mode** These are all the rules governing the behavior of the module during the transitional phases or on the appearance of a fault.

## P

**Predefined Tare Value (PT)** Numerical value, representing a weight, which is entered into the instrument at configuration or by adjustment or program.

**Premium** Families of Schneider Electric hardware products.

## S

**Scale Division** Value in mass units, expressing the difference between two consecutive indications for one numerical indication.

**Set to Zero Device** Device allowing the indicator to be "recalibrated" in the event of a deviation from zero (due to dirt accumulation, for example). This operation can only be carried out in the extent of zero range (+/-2 % or +/-5 % of the maximum range according to the weighing instrument).

## T

**Tare** Load placed on the load receiver along with the product to be weighed. For example: product packaging or container.

**Tare Device** Device allowing the instrument indication to be moved to zero when a load is positioned on the load receiver:
- without encroaching on the weighing range of net loads (tare additive device),
- or reducing the weighing range of net loads (tare subtracting device, such as **TSX ISP Y100**).

**Tare Predefining Device** Device allowing a predefined tare value to be subtracted from a gross weight value and indicating the result of the calculation. The weighing range is consequently reduced.

**Tare Value (T)** Weight value of a load, determined by a tare weighing device.

**Taring** Action allowing the instrument indication to be moved to zero when a load is positioned on the load receiver.

## U

**Unity Pro** Programming software of Schneider Electric PLCs.

## W

**Weighing Instruments**
Measuring instruments, which determine the mass of a body using the force of gravity.
These instruments can also be used to determine other sizes, quantities, parameters or characteristics linked to the mass. According to the way they function, weighing instruments are either classified as automatic or non-automatic functioning instruments.

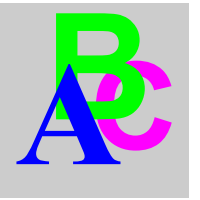
**Weighing Range**
Interval between maximum and minimum weight.

## Z

**Zero Load**
Tare weight of load receiver when equipped with its mechanical accessories (vibratory extractor, screw, trap, screw jack, etc.). It does not appear in the weight indication but must be taken into account when calculating the maximum load of the sensors.

**Zero Tracking**
Device allowing slow derivations from zero to be made up, within the limits of the extent of the zero range.

# Index



## L



## M



## O



## P



## R



## S



## T



## W

## Z