ENGLISH  ENGLISH  ENGLISH  ENGLISH

User Manual
version 1.08

TLB4

COMMUNICATION PROTOCOLS

SYMBOLS

Here are the symbols used in the manual to draw the reader's attention:

Caution! Risk of electric shock.

Caution! This operation must be performed by skilled personnel.

Pay particular attention to the following instructions.

Further information.

TABLE OF CONTENTS

INTRODUCTION TO THE USER MANUAL

This manual explains the operation of the instrument through the use of the communication interfaces, to which reference is made as per the table; moreover, the abbreviation of the registers will be used instead of the extended name (see section **FIELDBUSES**). For more information on the parameters and functions illustrated, consult the user manual of the instrument.

INTERFACE	FIELDBUS
MODBUS	MODBUS-RTU – MODBUS/TCP
PROFIBUS-DP	PROFIBUS-DP
GENERIC	CANOPEN – DEVICENET – ETHERCAT – ETHERNET/IP POWERLINK – PROFINET IO – SERCOS III
CC-LINK	CC-LINK

FIELDBUSES

PARAMETERS OF THE COMMUNICATION INTERFACES

Not all parameters listed in this section are used by all interfaces; for more information refer to the section of the specific fieldbus.

GROSS WEIGHT (GW) – NET WEIGHT (NW) – PEAK WEIGHT (PW)

The weight values are expressed as positive integer numbers, including decimal figures, but without decimal point. Refer to the section of the specific fieldbus to obtain information about sign and possible errors on the weight.

EXCHANGE REGISTERS (R1, W1)

These registers are used for instrument management operations. There is a read (R1) and a write (W1) exchange register.

STATUS REGISTER (SR1)

Bit 0	Load cell error	**Bit 8**	Net weight negative sign
Bit 1	AD converter malfunction	**Bit 9**	Peak weight negative sign
Bit 2	Maximum weight exceeded by 9 divisions	**Bit 10**	Net display mode
Bit 3	Gross weight over 110% of full scale	**Bit 11**	Weight stability
Bit 4	Gross weight over 999999 or less than -999999	**Bit 12**	Weight within ±¼ of a division around ZERO
Bit 5	Net weight over 999999 or less than -999999	**Bit 13**	
Bit 6		**Bit 14**	
Bit 7	Gross weight negative sign	**Bit 15**	Load cells references not connected

Refer to the section of the specific fieldbus to identify the position of the Status Register among the data exchanged at the instrument output.

CC-LINK INTERFACE

The CC-Link protocol provides for the Status Register only when the instrument is in a four-station configuration and the values of the divisions between the data exchanged at the instrument output are enabled (see section **READING DIVISIONS WITH SIGN OF EACH WEIGHTING CHANNEL**).

INSTRUMENT STATUS (IS)

This register is used for the LOAD program only.

0	Instrument in idle condition (weight displaying)	**13**	SLAVE alarm
1	Batching not possible/programming mode	**14**	EMPtY alarm
2	Batching phase	**15**	MASFOr alarm
3	Waiting phase	**16**	------ alarm
4	Cycle end phase	**17**	tArEP alarm
5	Batching paused	**18**	LOAd alarm
6	UnLOAd alarm	**19**	PArSEr alarm
7	FALL alarm	**20**	
8	tOL alarm	**21**	
9		**22**	
10	ErUEI G alarm	**23**	
11		**24**	Temporary message tOtAL
12			

INPUTS (INS) AND OUTPUTS (OUTS) STATUS

DIGITAL INPUTS STATUS (INS)

Bit 0	INPUT 1 status
Bit 1	INPUT 2 status
Bit 2	
Bit 3	
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Bit=1: high input; Bit=0: low input

DIGITAL OUTPUTS STATUS (OUTS)

Bit 0	OUTPUT 1 status
Bit 1	OUTPUT 2 status
Bit 2	OUTPUT 3 status
Bit 3	
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Bit=1: output is closed; Bit=0: output is open

DIGITAL OUTPUTS COMMAND (CMDOUT)

This register is used for the BASE program only; it allows to control the outputs set to *PLC* mode (see section **OUTPUTS AND INPUTS CONFIGURATION (BASE program)**).

Bit 0	OUTPUT 1 status
Bit 1	OUTPUT 2 status
Bit 2	OUTPUT 3 status
Bit 3	
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Bit=1: output is closed; Bit=0: output is open

Bit 8	
Bit 9	
Bit 10	
Bit 11	
Bit 12	
Bit 13	
Bit 14	
Bit 15	Force outputs

Setting bit 15 to 1 on the PLC, the *master* takes control of all the outputs, whatever their setting.

ERROR CODE (ERC) – AUXILIARY ERROR CODE (AERC)

These registers allow you to read the error codes (see section **ALARMS MANAGEMENT**).

EXECUTION CODE COMMAND (EXC) – AUXILIARY EXECUTION CODE COMMAND (AEXC)

These registers allow you to read the execution status of the commands sent to the CMDR (see section **COMMAND EXECUTION STATUS**).

COMMAND REGISTER (CMDR)

The commands are sent to the instrument through this register.

COMMAND NUMBER (CMDN)

The last command sent to the instrument is read through this register.

SETPOINT (SPn) –HYSTERESIS (HYSn)
These registers allow you to manage the setpoint and hysteresis values (see section **SETPOINT PROGRAMMING (BASE program)**).

DIVISIONS AND UNITS OF MEASURE (DU)
This register contains the current setting of the divisions and of the units of measure (see sections **DIVISION** and **SETTING UNITS OF MEASURE**).

H byte	L byte
Unit of measure	Division

DISPLAY COEFFICIENT (COF)
This register allows you to set the display coefficient (see section **DISPLAY COEFFICIENT (BASE program)**).

SAMPLE WEIGHT FOR INSTRUMENT CALIBRATION (CALW)
This register allows you to set the value of the sample weight (see section **REAL CALIBRATION (WITH SAMPLE WEIGHTS)**).

ANALOG ZERO (ANA0) – ANALOG FULL SCALE (ANAFS)
These registers allow you to set the zero and full scale of the analog output (see section **ANALOG OUTPUT (TLB4 ONLY)**).

PRESET TARE (PT)
This register allows you to set the preset tare value (see section **PRESET TARE**).

SET (SET) – PRESET (PSET) – FALL (FALL) – TOLERANCE (TOLL) – MAXIMUM (MAX) – MINIMUM (MIN)
These registers are used only for the LOAD program and allow you to set the values of the main batching parameters.

COMMAND EXECUTION STATUS

EXC Execution code command	AEXC Auxiliary execution code command	Description
Command code sent to CMDR	0	Execution completed successfully
1	0	Running
-2	0	The limits allowed for the parameter have been exceeded
-3 (execution error)	1	Sample weight equal to zero
	2	Maximum number of calibration points reached
	3	Sample weight already used in the current calibration
	4	Number of active channels other than one: This filtering level cannot be selected
	5	No active channels
	6	Invalid function selected
	7	Equalization disabled
	8	Slow set, function not available
	9	The current load cell has already been equalized
	10	Preset tare equal to zero
	11	Semiautomatic tare active: preset tare not allowed
	12	Gross weight equal to zero
	14	Set value less than or equal to the Preset value
	15	Preset value greater than the Set value
	16	Incorrect requested configuration
	17	Invalid output index
	18	Invalid channel index
	21	Semiautomatic tare active: semiautomatic zero not allowed
	22	Weight greater than the value of parameter 0 SEt
-4	0	Qualified access required for parameter modification
-5	0	Command not available in the current configuration

MODBUS INTERFACE

- Read the execution code command in EXC
- Read the auxiliary execution code command in AEXC

PROFIBUS-DP, GENERIC, CC-LINK INTERFACES

- Read EXC in the two H byte of R1
- Read AEXC in the two L byte of R1

Command execution codes are expressed through 4-byte numbers in which the two H byte represent EXC and the two L byte represent AEXC.

Example: content of R1 when prompted to enter a calibration point when all available points are already used

	H (2 byte)	L (2 byte)	Total
Hexadecimal	0xFFFD	0x0002	0xFFFD0002
Decimal	-3	2	-196606

MODBUS-RTU

The MODBUS-RTU protocol allows the management of the reading and writing of the following registries according to the specifications found on the reference document for this **Modicon PI-MBUS-300** standard.

To select the MODBUS-RTU communication see section **SERIAL COMMUNICATION SETTING** in instrument manual.

Check if the *master* MODBUS-RTU in use (or the development tool) requires the disclosure of registers based on 40001 or 0. In the first case the registers numbering corresponds to the one in the table; in the second case the register must be determined as the value in the table minus 40001. E.g.: the register 40028 shall be reported as 27 (= 40028-40001).

Certain data, when specifically indicated, will be written directly in the EEPROM type memory. This memory has a limited number of writing operations (100000), therefore it is necessary to pay particular attention to not execute useless operations on said locations. The instrument in any case makes sure that no writing occurs if the value to be memorised is equal to the value in memory.

The numerical data found below are expressed in decimal notation; if the prefix 0x is entered the notation will be hexadecimal.

MODBUS-RTU DATA FORMAT

The data received and transmitted by way of the MODBUS-RTU protocol have the following characteristics:
- 1 start bit
- 8 bit of data, *least significant bit* sent first
- Settable parity bit
- Settable stop bit

FUNCTIONS SUPPORTED IN MODBUS

Among the commands available in the MODBUS-RTU protocol, only the following are utilised for management of communication with the instruments; other commands could be incorrectly interpreted and generate errors or blocks of the system:

FUNCTIONS	DESCRIPTION
03 (0x03)	READ HOLDING REGISTER (READ PROGRAMMABLE REGISTERS)
16 (0x10)	PRESET MULTIPLE REGISTERS (WRITE MULTIPLE REGISTERS)

Interrogation frequency is linked to the communication speed set (the instrument stands by for at least 3 bytes before starting calculations an eventual response to the interrogation query). The dELAY parameter present in section **SERIAL COMMUNICATION SETTING** in instrument manual, allows the instrument to respond with a further delay and this directly influences the number of interrogations possible in the unit of time.
For additional information on this protocol refer to the general technical specifications PI_MBUS_300.

In general queries and answers toward and from one slave instrument are composed as follows:

FUNCTION 3: Read holding registers (READ PROGRAMMABLE REGISTERS)

QUERY

Address	Function	1st register address	No. registers	2 byte
A	0x03	0x0000	0x0002	CRC

Tot. byte=8

RESPONSE

Address	Function	No. bytes	1st register	2nd register	2 byte
A	0x03	0x04	0x0064	0x00C8	CRC

Tot. byte=3+2*No. registers+2

where: No. registers ..... number of Modbus registers to write beginning from the address no. 1
No. byte ..... number of bytes of the following data

FUNCTION 16: Preset multiple registers (WRITE MULTIPLE REGISTERS)

QUERY

Address	Function	1st reg. add.	No. reg.	No. bytes	Val.reg.1	Val.reg.2	2 byte
A	0x10	0x0000	0x0002	0x04	0x0000	0x0000	CRC

Tot. byte=7+2*No. registers+2

RESPONSE

Address	Function	1st reg. address	No. reg.	2 byte
A	0x10	0x0000	0x0002	CRC

Tot. byte=8

where: No. registers ..... number of Modbus registers to read beginning from the address no. 1
No. byte ..... number of bytes of the following data
Val.reg.1 ..... contents of the register beginning from the first

The response contains the number of registers modified beginning from the address no. 1.

COMMUNICATION ERROR MANAGEMENT

The communication strings are controlled by way of the CRC (Cyclical Redundancy Check).
In case of communication error the slave will not respond with any string. The master must consider a time-out for reception of the answer. If it does not receive an answer it deduces that there has been a communication error.

In the case of the string received correctly but not executable, the slave responds with an EXCEPTIONAL RESPONSE. The "Function" field is transmitted with the msb at 1.

EXCEPTIONAL RESPONSE

Address	Function	Code	2 byte
A	Funct + 0x80		CRC

CODE	DESCRIPTION
1	ILLEGAL FUNCTION (the function is not valid or is not supported)
2	ILLEGAL DATA ADDRESS (the specified data address is not available)
3	ILLEGAL DATA VALUE (the data received has an invalid value)

LIST OF AVAILABLE REGISTERS

The MODBUS-RTU protocol implemented on this instrument can manage a maximum of 32 registers read and written in a single query or response.

R.....the register may only be read
W.....the register may only be written
R/W .....the register may be both read and written
H.....high half of the DOUBLE WORD containing the number
L .....low half of the DOUBLE WORD containing the number

Register	Description	ABBR	Saving in EPROM	Access
40001	Firmware version	-	-	R
40002	Instrument type	-	-	R
40003	Year of manufacture	-	-	R
40004	Serial number	-	-	R
40005	Program type	-	-	R
40006	COMMAND REGISTER	CMDR	NO	R/W
40007	STATUS REGISTER	SR1	-	R
40008	GROSS WEIGHT H	GW	-	R
40009	GROSS WEIGHT L		-	R
40010	NET WEIGHT H	NW	-	R
40011	NET WEIGHT L		-	R
40012	PEAK WEIGHT H	PW	-	R
40013	PEAK WEIGHT L		-	R
40014	Divisions and Units of measure	DU	-	R
40015	Coefficient H	COF	-	R
40016	Coefficient L		-	R
40017	INPUTS	INS	-	R
40018	OUTPUTS	OUTS	NO	R/W
40019	SETPOINT 1 H	SP1	Only after command 99 of the Command Register	R/W
40020	SETPOINT 1 L			R/W
40021	SETPOINT 2 H	SP2		R/W
40022	SETPOINT 2 L			R/W
40023	SETPOINT 3 H	SP3		R/W
40024	SETPOINT 3 L			R/W
40039	HYSTERESIS 1 H	HYS1		R/W
40040	HYSTERESIS 1 L			R/W
40041	HYSTERESIS 2 H	HYS2		R/W
40042	HYSTERESIS 2 L			R/W
40043	HYSTERESIS 3 H	HYS3		R/W
40044	HYSTERESIS 3 L			R/W
40050	INSTRUMENT STATUS	IS	NO	R

40051	REGISTER 1	R1/W1*	NO	R/W
40052	REGISTER 2			R/W
40053	REGISTER 3			R/W
40054	REGISTER 4			R/W
40055	REGISTER 5			R/W
40056	REGISTER 6			R/W
40057	REGISTER 7			R/W
40058	REGISTER 8			R/W
40059	REGISTER 9			R/W
40060	REGISTER 10			R/W
40061	REGISTER 11			R/W
40062	REGISTER 12	AEXC		R/W
40063	REGISTER 13			R/W
40064	REGISTER 14	EXC		R/W
40065	Sample weight for instrument calibration H	CALW	Use with command 101 of the Command Register	R/W
40066	Sample weight for instrument calibration L			R/W
40067	Weight value corresponding to ZERO of the analog output H	ANA0	YES	R/W
40068	Weight value corresponding to ZERO of the analog output L			R/W
40069	Weight value corresponding to the full scale of the analog output H	ANAFS		R/W
40070	Weight value corresponding to the full scale of the analog output L			R/W
40073	Preset tare H	PT	Use with command 130 of the Command Register	R/W
40074	Preset tare L			R/W

*) in this document, reference is made to R1 for access to the register in reading and W1 for access to the register in writing.

COMMUNICATION EXAMPLES

The numerical data below are expressed in hexadecimal notation with prefix h.

EXAMPLE 1
Command for multiple writing of registers (command 16, h10 hexadecimal):
Assuming that we wish to write the value 0 to the register 40017 and the value 2000 to the register 40018, the string to generate must be:

h01 h10 h00 h10 h00 h02 h04 h00 h00 h07 hD0 hF1 h0F

The instrument will respond with the string:

h01 h10 h00 h10 h00 h02 h40 h0D

Query field name	hex	Response field name	hex
Instrument address	**h01**	Instrument address	**h01**
Function	**h10**	Function	**h10**
Address of the first register H	**h00**	Address of the first register H	**h00**
Address of the first register L	**h10**	Address of the first register L	**h10**
Number of registers H	**h00**	Number of registers H	**h00**
Number of registers L	**h02**	Number of registers L	**h02**
Byte count	**h04**	CRC16 L	**h40**
Datum 1 H	**h00**	CRC16 H	**h0D**
Datum 1 L	**h00**		
Datum 2 H	**h07**		
Datum 2 L	**hD0**		
CRC16 L	**hF1**		
CRC16 H	**h0F**		

EXAMPLE 2
Command for multiple writing of registers (command 16, h10 hexadecimal):
Assuming that we wish to write two setpoint values on the instrument, at 2000 (setpoint 1: 40019-40020) and 3000 (setpoint 2: 40021-40022) respectively, the string must be sent:

h01 h10 h00 h12 h00 h04 h08 h00 h00 h07 hD0 h00 h00 h0B hB8 h49 h65

The instrument will respond with the string:

h01 h10 h00 h12 h00 h04 h61 hCF

Query field name	hex	Response field name	hex
Instrument address	**h01**	Instrument address	**h01**
Function	**h10**	Function	**h10**
Address of the first register H	**h00**	Address of the first register H	**h00**
Address of the first register L	**h12**	Address of the first register L	**h12**
Number of registers H	**h00**	Number of registers H	**h00**
Number of registers L	**h04**	Number of registers L	**h04**
Byte count	**h08**	CRC16 L	**h61**
Datum 1 H	**h00**	CRC16 H	**hCF**
Datum 1 L	**h00**		
Datum 2 H	**h07**		
Datum 2 L	**hD0**		
Datum 3 H	**h00**		
Datum 3 L	**h00**		
Datum 4 H	**h0B**		
Datum 4 L	**hB8**		
CRC16 L	**h49**		
CRC16 H	**h65**		

EXAMPLE 3
Multiple commands reading for registers (command 3, h03 hexadecimal):
Assuming that we wish to read the gross weight value (in the example 4000) and net weight value (in the example 3000), reading from address 40008 to address 40011 must be performed by sending the following string:

h01 h03 h00 h07 h00 h04 hF5 hC8

The instrument will respond with the string:

h01 h03 h08 h00 h00 h0F hA0 h00 h00 h0B hB8 h12 h73

Query field name	hex	Response field name	hex
Instrument address	**h01**	Instrument address	**h01**
Function	**h03**	Function	**h03**
Address of the first register H	**h00**	Byte count	**h08**
Address of the first register L	**h07**	Datum 1 H	**h00**
Number of registers H	**h00**	Datum 1 L	**h00**
Number of registers L	**h04**	Datum 2 H	**h0F**
CRC16 L	**hF5**	Datum 2 L	**hA0**
CRC16 H	**hC8**	Datum 3 H	**h00**
		Datum 3 L	**h00**
		Datum 4 H	**h0B**
		Datum 4 L	**hB8**
		CRC16 L	**h12**
		CRC16 H	**h73**

For additional examples regarding the generation of correct control characters (CRC16) refer to the manual **Modicon PI-MBUS-300.**

CANOPEN

TECHNICAL SPECIFICATIONS AND CONNECTIONS

Baud rate [kb/s]	10, 20, 50, 125, 250, 500, 800, 1000
Node ID	1÷127

Terminals legend	47 .....CAN GND 46 .....CAN L 45 .....CAN SHLD 44 .....CAN H 43 .....NC

The instrument features a CANopen port that allows to exchange the weight and the main parameters with a CANopen *master*.

INSTRUMENT SETUP

[↵] + [✕] → CAnOPn

- Addr (default: 1): set the instrument address in the CANopen network
- bAUd (default: 10 kb/s): set the instrument baud rate in the CANopen network
- SWAP (default: nO): it allows to select the reading/writing of the byte in LITTLE-ENDIAN or BIG-ENDIAN mode
  - YES: BIG ENDIAN
  - nO: LITTLE ENDIAN

In order to apply the changes, press [✕] until the display shows CAnOPn.

PC/PLC SETUP

The instrument works as *slave* in a synchronous CANopen network (activate the SYNC object on the network master).
Load the eds file attached to the instrument to the CANopen *master* development system.
When configuring CANopen Guard Time and Lifetime Factor, set values 100 ms and 4.
The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Index	Sub-Index	Data type	Addresses
Gross Weight [4 byte]	GW	4100	01	UNSIGNED32	0x0000-0x0003
Net Weight [4 byte]	NW	4100	02	UNSIGNED32	0x0004-0x0007
Exchange Register [4 byte]	R1	4101	01	UNSIGNED32	0x0008-0x000B
Status Register [2 byte]	SR1	4101	02	UNSIGNED16	0x000C-0x000D
Digital Inputs status [1 byte]	INS	4101	03	UNSIGNED8	0x000E
Digital Outputs status [1 byte]	OUTS	4101	04	UNSIGNED8	0x000F

Input Data to instrument (Writing)	ABBR	Index	Sub-Index	Data type	Addresses
Command Register [2 byte]	CMDR	4000	01	UNSIGNED16	0x0000-0x0001
Digital Outputs Command [2 byte]	CMDOUT	4000	02	UNSIGNED16	0x0002-0x0003
Exchange Register [4 byte]	W1	4000	03	UNSIGNED32	0x0004-0x0007

TECHNICAL SPECIFICATIONS AND CONNECTIONS

Baud rate [kb/s]	125, 250, 500
Addresses	1÷63

Terminals legend	47 .....CAN V - 46 .....CAN L 45 .....CAN SHLD 44 .....CAN H 43 .....CAN V +

It is necessary to activate the termination resistance on the two devices located at the ends of the network closing the jumper.

The instrument features a DeviceNet port that allows to exchange the weight and the main parameters with a DeviceNet *master*.

INSTRUMENT SETUP

[↵] + [✕] → dEUnEt

- Addr (default: 1): set the instrument address in the DeviceNet network
- bAUd (default: 125 kb/s): set the instrument baud rate in the DeviceNet network
- SWAP (default: nO): it allows to select the reading/writing of the byte in LITTLE-ENDIAN or BIG-ENDIAN mode
  - YES: BIG ENDIAN
  - nO: LITTLE ENDIAN

In order to apply the changes, press [✕] until the display shows dEUnEt.

PC/PLC SETUP

The instrument works as *slave* in a DeviceNet network.
Load the eds file attached to the instrument to the DeviceNet *master* development system.
The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Addresses
Gross Weight [4 byte]	GW	0x0000-0x0003
Net Weight [4 byte]	NW	0x0004-0x0007
Exchange Register [4 byte]	R1	0x0008-0x000B
Status Register [2 byte]	SR1	0x000C-0x000D
Digital Inputs status [1 byte]	INS	0x000E
Digital Outputs status [1 byte]	OUTS	0x000F

Input Data to instrument (Writing)	ABBR	Addresses
Command Register [2 byte]	CMDR	0x0000-0x0001
Digital Outputs Command [2 byte]	CMDOUT	0x0002-0x0003
Exchange Register [4 byte]	W1	0x0004-0x0007

TECHNICAL SPECIFICATIONS AND CONNECTIONS

Baud rate	156 k, 625 k, 2500 k, 5 M, 10 M
Addresses	1÷64
Stations	1, 2, 4

Status LED indications (red)	off .....timeout/reset on .....CC-LINK OK

Terminals legend	10 .....CCL DA 11 .....CCL DB 12 .....CCL DG 13 .....CCL SLD 14 .....CCL FG

To activate the termination resistance of CC-LINK network close the related jumper.

The instrument features a CC-LINK port that allows to exchange the weight and the main parameters with a CC-LINK *master*.

INSTRUMENT SETUP

↵ + ☒ → CCLInH

- Addr (default: 1): set the instrument address in the CC-LINK network
- bAUd (default: 156 kb/s): set the instrument baud rate in the CC-LINK network
- nUMSTA (default: 4): set the number of stations of the instrument on the CC-LINK network

In order to apply the changes, press ☒ until the display shows CCLInH.

PC/PLC SETUP

The instrument works as *Remote Device Station* in a CC-LINK network and occupies 1, 2 or 4 stations.
Load the csp file attached to the instrument to the CC-LINK *master* development system.
The data exchanged by the instrument varies according to the number of stations set and is divided into:

- registers managed by bit: RX/RY
- registers managed by word: RWr/RWw

RX/RY

ONE STATION

RX

Output Data from instrument (Reading)	Device No. bit
Response to Storage request for batching data[18]	RXn0
Error[31]	RXn1
Response to Generic Command Execution Request[1]	RXn2
Writing/Reading Response[2]	RXn3
Net display mode[9]	RXn4
-	RXn5
Correct operation of the instrument[3]	RXn6
SET contact[19]	RXn7
PRESET contact[20]	RXn8
Tapping function[21]	RXn9
Tolerance[22]	RXnA
Gross weight negative sign[5]	RXnB
Net weight negative sign[5]	RXnC
Cycle end[23]	RXnD
Weight stability[7]	RXnE
Weight within ±¼ of a division around ZERO[6]	RXnF
-	RX(n+1)0
-	RX(n+1)1
-	RX(n+1)2
-	RX(n+1)3
-	RX(n+1)4
-	RX(n+1)5

RY

Input Data to instrument (Writing)	Device No. bit
Storage request for batching data[18]	RYn0
	RYn1
Generic Command Execution Request[1]	RYn2
Writing/Reading Selection[2]	RYn3
SEMI-AUTOMATIC ZERO[15]	RYn4
SEMI-AUTOMATIC TARE enabling (net weight displaying)[16]	RYn5
SEMI-AUTOMATIC TARE disabling (gross weight displaying)[17]	RYn6
Batching start[24]	RYn7
Batching pause[25]	RYn8
Batching resume[26]	RYn9
Batching stop[27]	RYnA
Accept batching alarm[28]	RYnB
Ignore tArEP[29]	RYnC
Ignore tOL[30]	RYnD
-	RYnE
-	RYnF
-	RY(n+1)0
-	RY(n+1)1
-	RY(n+1)2
-	RY(n+1)3
-	RY(n+1)4
-	RY(n+1)5

-	RX(n+1)6
-	RX(n+1)7
-	RX(n+1)8
-	RX(n+1)9
-	RX(n+1)A
System ready[14]	RX(n+1)B
-	RX(n+1)C
-	RX(n+1)D
-	RX(n+1)E
-	RX(n+1)F

-	RY(n+1)6
-	RY(n+1)7
-	RY(n+1)8
-	RY(n+1)9
-	RY(n+1)A
-	RY(n+1)B
-	RY(n+1)C
-	RY(n+1)D
-	RY(n+1)E
-	RY(n+1)F

TWO STATIONS

RX

Output Data from instrument (Reading)	Device No. bit
Response to Storage request for batching data[18]	RXn0
-	RXn1
Response to Generic Command Execution Request[1]	RXn2
Writing/Reading Response[2]	RXn3
	RXn4
-	RXn5
Correct operation of the instrument[3]	RXn6
-	RXn7
Decimal point 1[4]	RXn8
Decimal point 2[4]	RXn9
Decimal point 4[4]	RXnA
Gross weight negative sign[5]	RXnB
Net weight negative sign[5]	RXnC
	RXnD - RXnF
Weight within ±¼ of a division around ZERO[6]	RX(n+1)0
SET contact[19]	RX(n+1)1
PRESET contact[20]	RX(n+1)2
Tapping function[21]	RX(n+1)3
Tolerance[22]	RX(n+1)4
-	RX(n+1)5

RY

Input Data to instrument (Writing)	Device No. bit
Storage request for batching data[18]	RYn0
	RYn1
Generic Command Execution Request[1]	RYn2
Writing/Reading Selection[2]	RYn3
	RYn4
-	RYn5
-	RYn6
-	RYn7
-	RYn8
-	RYn9
-	RYnA
-	RYnB
	RYnC
	RYnD - RYnF
SEMI-AUTOMATIC ZERO[15]	RY(n+1)0
	RY(n+1)1
SEMI-AUTOMATIC TARE enabling (net weight displaying)[16]	RY(n+1)2
SEMI-AUTOMATIC TARE disabling (gross weight displaying)[17]	RY(n+1)3
	RY(n+1)4
	RY(n+1)5

-	RX(n+1)6		RY(n+1)6
Weight stability[7]	RX(n+1)7	Batching start[24]	RY(n+1)7
Cycle end[23]	RX(n+1)8	Batching pause[25]	RY(n+1)8
Maximum gross weight exceeded by 9 divisions[8]	RX(n+1)9	Batching resume[26]	RY(n+1)9
	RX(n+1)A	Batching stop[27]	RY(n+1)A
	RX(n+1)B	Accept batching alarm[28]	RY(n+1)B
	RX(n+1)C	Ignore tArEP [29]	RY(n+1)C
	RX(n+1)D	Ignore tOL [30]	RY(n+1)D
	RX(n+1)E		RY(n+1)E
	RX(n+1)F		RY(n+1)F
Net display mode[9]	RX(n+2)0	-	RY(n+2)0
Load cells reference not connected[10]	RX(n+2)1	-	RY(n+2)1
AD convertor malfunction[11]	RX(n+2)2	-	RY(n+2)2
Load cell error[12]	RX(n+2)3	-	RY(n+2)3
Net weight over the maximum displayable value[13]	RX(n+2)4	-	RY(n+2)4
Gross weight over the maximum displayable value[13]	RX(n+2)5	-	RY(n+2)5
-	RX(n+2)6 - RX(n+3)A	-	RY(n+2)6 - RY(n+3)A
System ready[14]	RX(n+3)B	-	RY(n+3)B
-	RX(n+3)C - RX(n+3)F	-	RY(n+3)C - RY(n+3)F

FOUR STATIONS

RX

Output Data from instrument (Reading)	Device No. bit
Response to Storage request for batching data[18]	RXn0
-	RXn1
Response to Generic Command Execution Request [1]	RXn2
Writing/Reading Response[2]	RXn3
	RXn4
-	RXn5
Correct operation of the instrument[3]	RXn6
-	RXn7
Decimal point 1[4]	RXn8
Decimal point 2[4]	RXn9
Decimal point 4[4]	RXnA

RY

Input Data to instrument (Writing)	Device No. bit
Storage request for batching data[18]	RYn0
	RYn1
Generic Command Execution Request[1]	RYn2
Writing/Reading Selection[2]	RYn3
	RYn4
-	RYn5
-	RYn6
-	RYn7
-	RYn8
-	RYn9
-	RYnA

Gross weight negative sign[5]	RXnB
Net weight negative sign[5]	RXnC
	RXnD - RXnF
Weight within ±¼ of a division around ZERO[6]	RX(n+1)0
SET contact[19]	RX(n+1)1
PRESET contact[20]	RX(n+1)2
Tapping function[21]	RX(n+1)3
Tolerance[22]	RX(n+1)4
-	RX(n+1)5
-	RX(n+1)6
Weight stability[7]	RX(n+1)7
Cycle end[23]	RX(n+1)8
Maximum gross weight exceeded by 9 divisions[8]	RX(n+1)9
	RX(n+1)A
	RX(n+1)B
	RX(n+1)C
	RX(n+1)D
	RX(n+1)E
	RX(n+1)F
Net display mode[9]	RX(n+2)0
Load cells reference not connected[10]	RX(n+2)1
AD convertor malfunction[11]	RX(n+2)2
Load cell error[12]	RX(n+2)3
Net weight over the maximum displayable value[13]	RX(n+2)4
Gross weight over the maximum displayable value[13]	RX(n+2)5
	RX(n+2)6 - RX(n+6)F
Reserved	RX(n+7)0 - RX(n+7)7
	RX(n+7)8
	RX(n+7)9
	RX(n+7)A
System ready[14]	RX(n+7)B
Reserved	RX(n+7)C - RX(n+7)F

-	RYnB
	RYnC
	RYnD - RYnF
SEMI-AUTOMATIC ZERO[15]	RY(n+1)0
	RY(n+1)1
SEMI-AUTOMATIC TARE enabling (net weight displaying)[16]	RY(n+1)2
SEMI-AUTOMATIC TARE disabling (gross weight displaying)[17]	RY(n+1)3
	RY(n+1)4
	RY(n+1)5
	RY(n+1)6
Batching start[24]	RY(n+1)7
Batching pause[25]	RY(n+1)8
Batching resume[26]	RY(n+1)9
Batching stop[27]	RY(n+1)A
Accept batching alarm[28]	RY(n+1)B
Ignore tArEP [29]	RY(n+1)C
Ignore tOL [30]	RY(n+1)D
	RY(n+1)E
	RY(n+1)F
	RY(n+2)0
	RY(n+2)1
	RY(n+2)2
	RY(n+2)3
	RY(n+2)4
	RY(n+2)6
	RY(n+2)5 - RY(n+6)F
Reserved	RY(n+7)0 - RY(n+7)7
	RY(n+7)8
	RY(n+7)9
	RY(n+7)A
Reserved	RY(n+7)B
Reserved	RY(n+7)C - RY(n+7)F

[1] Generic Command Execution Request
The request to execute a command must be sent via the RYn2 bit and checked via the RXn2 bit (see section **CER PROCEDURE**).

[2] Writing/reading selection
Write in RYn3 how the command sent to CMDR should be executed:

0=writing
1=reading

RXn3 automatically assumes the value of RYn3 (echo of RYn3).

[3] Correct operation of the instrument
Operating condition of the CC-Link instrument: if the bit changes state by alternating 1 and 0 once per second, the operation is correct.

[4] Decimal point
Read RXn8, RXn9, RXnA to know the number of decimals of the weight value.

bit			Number of decimals	Weight value representation
RXnA	RXn9	RXn8		
0	0	0	0	000000
0	0	1	1	00000.0
0	1	0	2	0000.00
0	1	1	3	000.000
1	0	0	4	00.0000

Example

Weight value	Number of decimals	bit		
		RXnA	RXn9	RXn8
0100.52	2	0	1	0
001937	0	0	0	0
34.0612	4	1	0	0

[5] Weight negative sign

0=the weight has a positive sign
1=the weight has a negative sign

[6] Weight within ±¼ of a division around ZERO

0=il peso non è entro ± ¼ di divisione attorno allo ZERO
1=il peso è entro ± ¼ di divisione attorno allo ZERO

[7] Weight stability

0= the weight is not stable
1= the weight is stable

[8] Maximum gross weight exceeded by 9 divisions

0=the weight has not exceeded the maximum weight value of 9 divisions
1=the weight has exceeded the maximum weight value of 9 divisions

[9] Net display mode

0=gross weight displaying
1=net weight displaying

[10] Load cells reference not connected

0=load cell reference connected correctly
1=load cell reference not connected correctly

[11] AD convertor malfunction

0=AD converter working properly
1=AD converter faulty

[12] Load cell error

0=load cells functioning correctly
1=error relating to the load cells

[13] Gross/net weight over the maximum displayable value

0=weight between 999999 and -999999
1= weight over 999999 or less than -999999

[14] System ready

0 = system initialization in progress
1 = system initialization completed

[15] SEMI-AUTOMATIC ZERO

Write 1 to enable the SEMI-AUTOMATIC ZERO.

[16] SEMI-AUTOMATIC TARE enabling (net weight displaying)

Write 1 to enable the SEMI-AUTOMATIC TARE.

[17] SEMI-AUTOMATIC TARE disabling (gross weight displaying)

Write 1 to disable the SEMI-AUTOMATIC TARE.

[18] Storage request for batching data

The batching data storage request must be sent via the RYn0 bit and checked via the RXn0 bit (see section **BPSR PROCEDURE**).

[19] SET contact

0=quantity of batched product greater than or equal to the SET FORMULA value
1=quantity of batched product lower than the SET FORMULA value

[20] PRESET contact

0=quantity of batched product greater than or equal to the PRESET FORMULA value
1=quantity of batched product lower than the PRESET FORMULA value

[21] Tapping function

0=tapping phase not active
1=tapping phase active

[22] Tolerance

0=the weight is within the set tolerance value
1=the weight is outside the set tolerance value

[23] Cycle end

0=cycle end phase not active
1=cycle end phase active

[24] Batching start

Write 1 to start the batching cycle.

[25] Batching pause

Write 1 to pause the batching cycle.

[26] Batching resume

Write 1 to resume the batching.

[27] Batching stop

Write 1 to stop the batching cycle.

[28] Accept batching alarm

EMPtY, ErUEI G, FALL, PArSEr alarms (see section **ALARMS MANAGEMENT DURING THE BATCHING**)
Write 1 to accept the batching alarm.

[29] Ignore tArEP

Write 1 to ignore the tare alarm.

[30] Ignore tOL

Write 1 to ignore the tolerance alarm.

[31] Error

0= no error detected
1= one or more errors detected (see sections **ALARMS MANAGEMENT** and **COMMAND EXECUTION STATUS**)

RWw/RWr

ONE STATION

RWr

Output Data from instrument (Reading)	ABBR	Dimension (byte)	Addresses
Net Weight	NW	4	Wr0000 – Wr0001
Exchange Register	R1	4	Wr0002 – Wr0003

RWw

Input Data to instrument (Writing)	ABBR	Dimension (byte)	Addresses
Exchange Register	W1	4	Ww0000 – Ww0001
Command Register	CMDR	2	Ww0002
-		2	Ww0003

TWO STATIONS

RWr

Output Data from instrument (Reading)	ABBR	Dimension (byte)	Addresses
Net Weight	NW	4	Wr0000 – Wr0001
Exchange Register	R1	4	Wr0002 – Wr0003
-		2	Wr0004
Command Number	CMDN	2	Wr0005
Error Code	ERC	2	Wr0006
Auxiliary Error Code	AERC	2	Wr0007

RWw

Input Data to instrument (Writing)	ABBR	Dimension (byte)	Addresses
-		10	Ww0000 – Ww0004
Exchange Register	W1	4	Ww0005 – Ww0006
Command Register	CMDR	2	Ww0007

FOUR STATIONS

RWr

Output Data from instrument (Reading)	ABBR	Dimension (byte)	Addresses
Net Weight	NW	4	Wr0000 – Wr0001
Gross Weight	GW	4	Wr0002 – Wr0003
-		4	Wr0004 – Wr0005
Error Code	ERC	2	Wr0006
Auxiliary Error Code	AERC	2	Wr0007
-		8	Wr0008- Wr000B
Exchange Register	R1	4	Wr000C – Wr000D
Command Number	CMDN	2	Wr000E
-		2	Wr000F

RWw

Input Data to instrument (Writing)	ABBR	Dimension (byte)	Addresses
SET*	SET	4	Ww0000 – Ww0001
PRESET*	PSET	4	Ww0002 – Ww0003
Fall*	FALL	4	Ww0004 – Ww0005
Tolerance*	TOLL	4	Ww0006 – Ww0007
Maximum*	MAX	4	Ww0008 – Ww0009
Minimum*	MIN	4	Ww000A – Ww000B
Exchange Register	W1	4	Ww000C – Ww000D
Command Register	CMDR	2	Ww000E
-		2	Ww000F

* These registers allow you to set the values of the main batching parameters: to store them, see the **BPSR PROCEDURE** section.

CER PROCEDURE

After writing a valid command code in CMDR, proceed as follows to execute the command:
- write 1 in RYn2 to send the command execution request
- read RXn2 to verify the execution of the command (1=executed, 0=not executed)
- if the command has been executed write 0 in RYn2
- RXn2 is automatically reset to 0 (echo of RYn2)

BPSR PROCEDURE

After having filled in the set (SET), preset (PSET), fall (FALL), tolerance (TOLL), maximum weight (MAX), minimum weight (MIN) registers with the values to be stored, proceed as follows to simultaneously store the batching parameters:
- write 1 in RYn0 to send the request for storing the batching parameters
- read RXn0 to verify the execution of the command (1 = executed, 0 = not executed)
- if the command has been executed write 0 in RYn0
- RXn0 is automatically reset to 0 (echo of RYn0)

The BPSR procedure is performed (RXn0=1) only if all registers contain a valid value (see section **BATCHING** in instrument manual).

ETHERNET TCP/IP

TECHNICAL SPECIFICATIONS

Port	RJ45 10Base-T or 100Base-TX (auto-detect)

Link LED indications	off .....Ethernet link not established amber .....10 Mb/s green .....100 Mb/s
Activity LED indications	off .....Ethernet activity not detected amber .....Half Duplex green .....Full Duplex

The instrument features an ethernet TCP/IP port that allows to exchange the weight and the main parameters in an ethernet network, for example with a PC.

INSTRUMENT SETUP

↵ + ✕ → EtHnEt

- IPAddr (default: 10.2.0.170): set instrument IP address
- SUbnEt (default: 255.255.255.0): set instrument Subnet Mask
- GAtUAY (default: 0.0.0.0): set Gateway address of Ethernet network
- nOdE: select communication protocol.
  - nOnE: it disables any type of communication (default).
  - ModbUS: MODBUS-RTU protocol; possible addresses: from 1 to 99.
  - ASCII : ASCII bidirectional protocol; possible addresses: from 1 to 99.
    - nOdU60
    - nOd td
  - CONtIn: continuous weight transmission protocol, at the frequency set in HErt2 item (from 10 to 200).
    - nOd t
    - nOd td
  - rIP: continuous weight transmission protocol to RIP5/20/60, RIP50SHA, RIPLED series remote displays; the remote display shows the net weight or gross weight according to its settings.
  - HdrIP: continuous weight transmission protocol to RIP6100, RIP675, RIP6125C series remote displays; the remote display shows the net weight or gross weight according to its settings.
  - HdrIPn: continuous weight transmission protocol to RIP6100, RIP675, RIP6125C series remote displays, when the remote display is set to gross weight:
    - if the instrument displays the gross weight, the remote display shows the gross weight.
    - if the instrument shows the net weight, the remote display shows the net weight alternated with the message nEt.

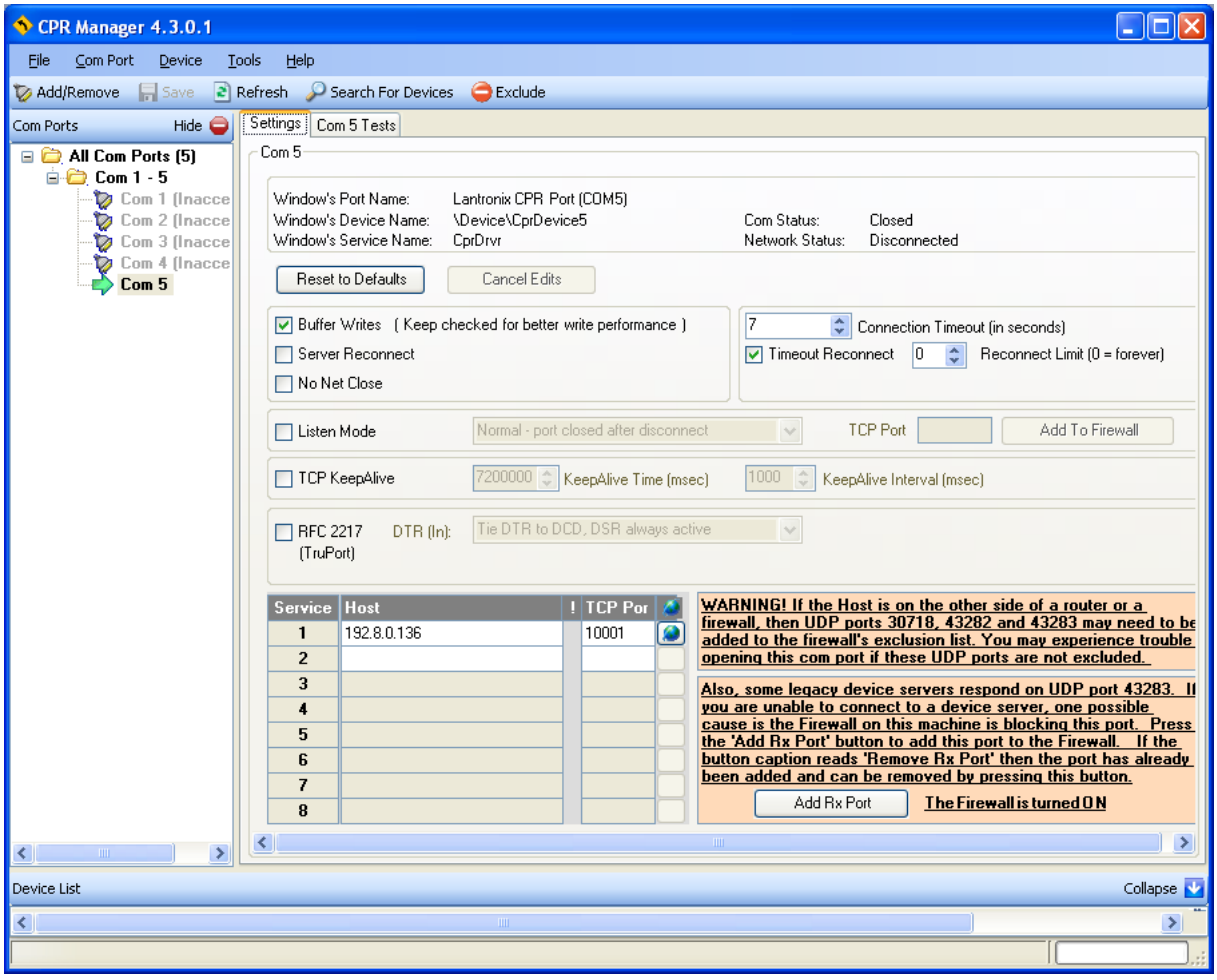
- UEbSru: see section **WEBSITE**.

  - Addr: instrument address (from 1 to 99; default: 1).
  - HErtZ: maximum transmission frequency (10 – 20 – 30 – 40 – 50 – 60 – 70 – 80 – 100 – 200; default: 10); to be set when the CONtI n transmission protocol is selected.
  - dELAY: delay in milliseconds which elapses before the instrument replies (from 0 to 200 ms; default: 0).

In order to apply the changes, turn the instrument off, wait for 10 seconds and turn it back on.

PC SETUP

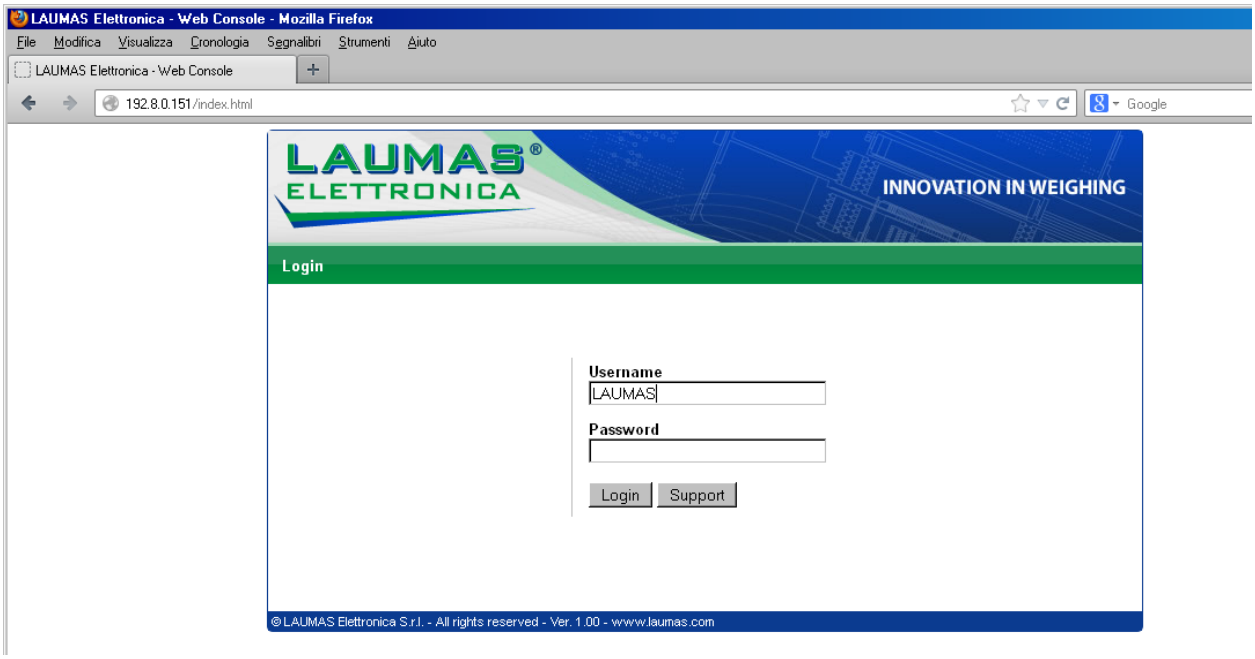
A PC can be connected, by a virtual serial port, to the instrument via ethernet TCP/IP.
To install the virtual COM port, use the CPR Manager included in the supply: run file *CPR.exe* on CD, add a serial port, set an IP address (host) and a TCP port (10001), then save.



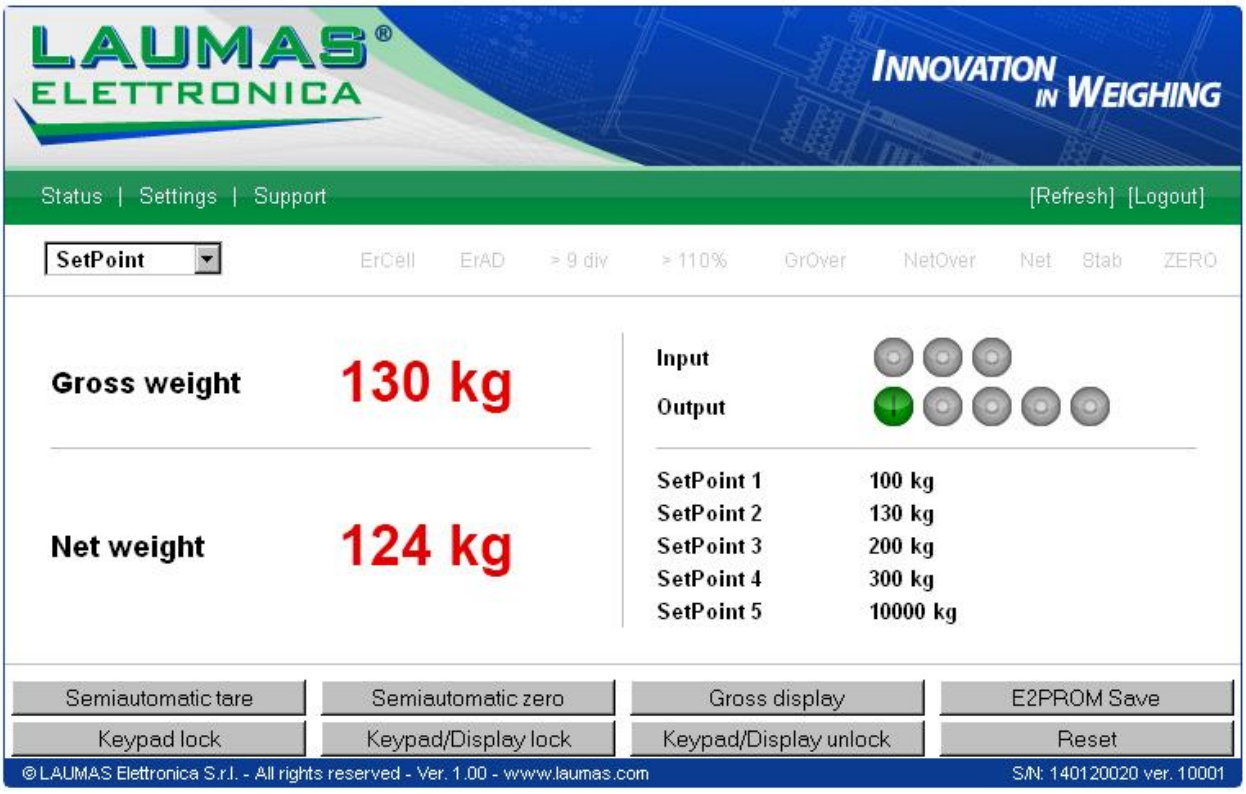
Use the just created virtual COM port to communicate with the instrument, using the protocol selected on it.
Alternatively connect to the instrument using a socket (e.g.: Winsock) on port 10001.

WEBSITE

Set UEb5ru operation mode (into EtHnEt menu on the instrument) and restart the instrument to apply changes. Open your web browser and point to the instrument address to be monitored; it will open the following page:



Enter the “LAUMAS” user name and the password supplied with the instrument in respective fields, then press Login to enter the status page:



In case of incorrect parameter setting, the “INSTRUMENT DATA READING ERROR” message is displayed.

The instrument status page shows the gross and net weight read, the setpoint values set and allows you to send the main commands (Tare, Zero setting, E2PROM saving, etc.); it also shows instrument status, including possible anomalies:

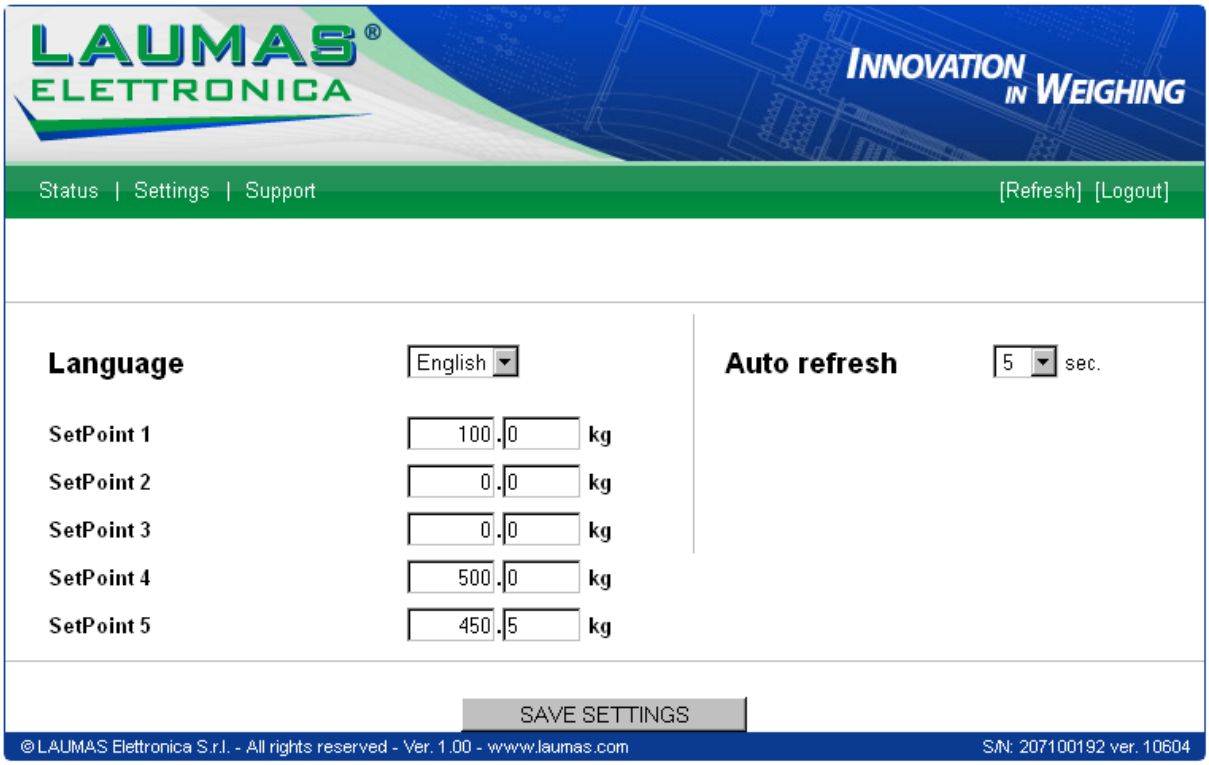
ErCell:..... load cell error
ErAD:..... instrument converter error
>9div:..... weight exceeds maximum weight by 9 divisions
>110%..... weight exceeds 110% of full scale
GrOver..... gross weight over 999999
NetOver..... net weight over 999999
Net..... instrument shows the net weight
Stab..... weight is stable
ZERO ..... weight is zero

Number of decimals and unit of measure are read by the instrument; if outputs are set in PLC mode, click on related icons to do a remote status check.

The screen to be displayed is selected through the drop down menu:

SetPoint:..... setpoint values
Load Distr.:.. percentage load distribution
mV: ..... current response signal of each load cell expressed in mV
mV zero: ..... response signal of each load cell, stored during zero setting, expressed in mV
Points:..... current response signal of each load cell expressed in converter points

Click on Settings to enter the instrument configuration page:



In the configuration page you can:

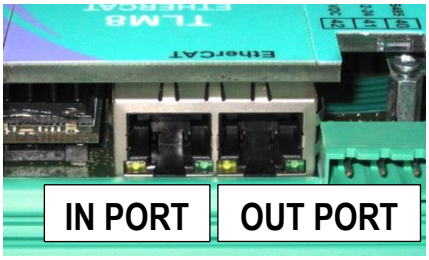
- set language and page refresh time: by pressing SAVE SETTINGS data are saved on the instrument and will be used for subsequent accesses;
- set setpoint: by pressing SAVE SETTINGS the new values are sent to the instrument and activated, but will be lost at instrument restart or power off; to permanently save setpoint values, press E2PROM Save in status page.

TECHNICAL SPECIFICATIONS

Port	2x RJ45 10Base-T or 100Base-TX (auto-detect)

Link/activity LED indications (green)	off .....Ethernet link not established on .....Ethernet link established blinking .....Ethernet activity detected

The instrument features an ETHERCAT dual port that allows to exchange the weight and the main parameters with an ETHERCAT *master*.



PC/PLC SETUP

The instrument works as *slave* in an ETHERCAT network.
Load the xml file attached to the instrument to the ETHERCAT *master* development system.

The Ethernet over EtherCAT (EoE) protocol is not supported.

The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Addresses
Gross Weight [4 byte]	GW	0x0000-0x0003
Net Weight [4byte]	NW	0x0004-0x0007
Exchange Register [4 byte]	R1	0x0008-0x000B
Status Register [2 byte]	SR1	0x000C-0x000D
Digital Inputs status [2 byte]	INS	0x000E-0x000F
Digital Outputs status [2 byte]	OUTS	0x0010-0x0011

Input Data to instrument (Writing)	ABBR	Addresses
Command Register [2 byte]	CMDR	0x0000-0x0001
Digital Outputs Command [2 byte]	CMDOUT	0x0002-0x0003
Exchange Register [4 byte]	W1	0x0004-0x0007

TECHNICAL SPECIFICATIONS

Port	2x RJ45 10Base-T or 100Base-TX (auto-detect)

Link LED indications (green)	off.....Ethernet link not established on.....Ethernet link established
Activity LED indications (amber)	off.....Ethernet activity not detected blinking .....Ethernet activity detected

The instrument features an Ethernet/IP dual port that allows to exchange the weight and the main parameters with an Ethernet/IP *scanner*.

INSTRUMENT SETUP

↵ + ☒ → EtHnEt

- SWAP (default: nO): it allows to select the reading/writing of the byte in LITTLE-ENDIAN or BIG-ENDIAN mode
  - YES: BIG ENDIAN
  - nO: LITTLE ENDIAN
- I PAddr (default: 10.2.0.170): set instrument IP address
- SUbnEt (default: 255.255.255.0): set instrument Subnet Mask
- GAtWAY (default: 0.0.0.0): set Gateway address of Ethernet network

In order to apply the changes, press ☒ until the display shows EtHnEt.

PC/PLC SETUP

The instrument works as *adapter* in an Ethernet/IP network.
Use one of the following communication types.

CLASS 1 CONNECTION (implicit messages)
Refer to one of the following procedures to configure the communication with the instrument:

- load the eds file attached to the instrument to the Ethernet/IP *scanner* development system (see table “32-BIT RUN/IDLE HEADER” for the output data interface);
- use a generic Ethernet/IP module: configure it with the parameters of the table “Parameters for class 1 communication” and choose the real-time transfer format from instrument to scanner (Target to Originator – T2O) between “32-BIT RUN/IDLE HEADER” and “PURE DATA” (see the respective tables for the output data interface).

Parameters for class 1 communication			
Assembly	**Assembly Instance**	**Size [Byte] 32-bit run/idle header**	**Size [Byte] Pure data**
Input	101	18	22
Output	100	8	8
Configuration	128	0	0

CLASS 3 CONNECTION (explicit messages)
Manually generate the request to be sent to the PLC using the parameters shown in the table “Manual settings for communication” (see table “PURE DATA” for the output data interface).

Manual settings for communication		
Field	**Read**	**Write**
Service	0x0E	0x10
Class	0x04	0x04
Instance	0x65	0x64
Attribute	0x03	0x03
Data	NO	Byte array to be written

The data exchanged by the instrument are:

32-BIT RUN/IDLE HEADER

Output Data from instrument (reading)	ABBR	Addresses input assembly
Gross Weight [4 byte]	GW	0x0000-0x0003
Net Weight [4 byte]	NW	0x0004-0x0007
Exchange Register [4 byte]	R1	0x0008-0x000B
Status Register [2 byte]	SR1	0x000C-0x000D
Digital Inputs status [2 byte]	INS	0x000E-0x000F
Digital Outputs status [2 byte]	OUTS	0x0010-0x0011

PURE DATA

Output Data from instrument (reading)	ABBR	Addresses input assembly
Ethernet/IP Header* [4 byte]		0x0000-0x0003
Gross Weight [4 byte]	GW	0x0004-0x0007
Net Weight [4 byte]	NW	0x0008-0x000B
Exchange Register [4 byte]	R1	0x000C-0x000F
Status Register [2 byte]	SR1	0x0010-0x0011
Digital Inputs status [2 byte]	INS	0x0012-0x0013
Digital Outputs status [2 byte]	OUTS	0x0014-0x0015

* registers used by the ETHERNET/IP *scanner* to manage the communication.

Input Data to instrument (Writing)	ABBR	Addresses
Command Register [2 byte]	CMDR	0x0000-0x0001
Digital Outputs Command [2 byte]	CMDOUT	0x0002-0x0003
Exchange Register [4 byte]	W1	0x0004-0x0007

TECHNICAL SPECIFICATIONS

Port	RJ45 10Base-T or 100Base-TX (auto-detect)

Link LED indications	off .....Ethernet link not established amber .....10 Mb/s green .....100 Mb/s
Activity LED indications	off .....Ethernet activity not detected amber .....Half Duplex green .....Full Duplex

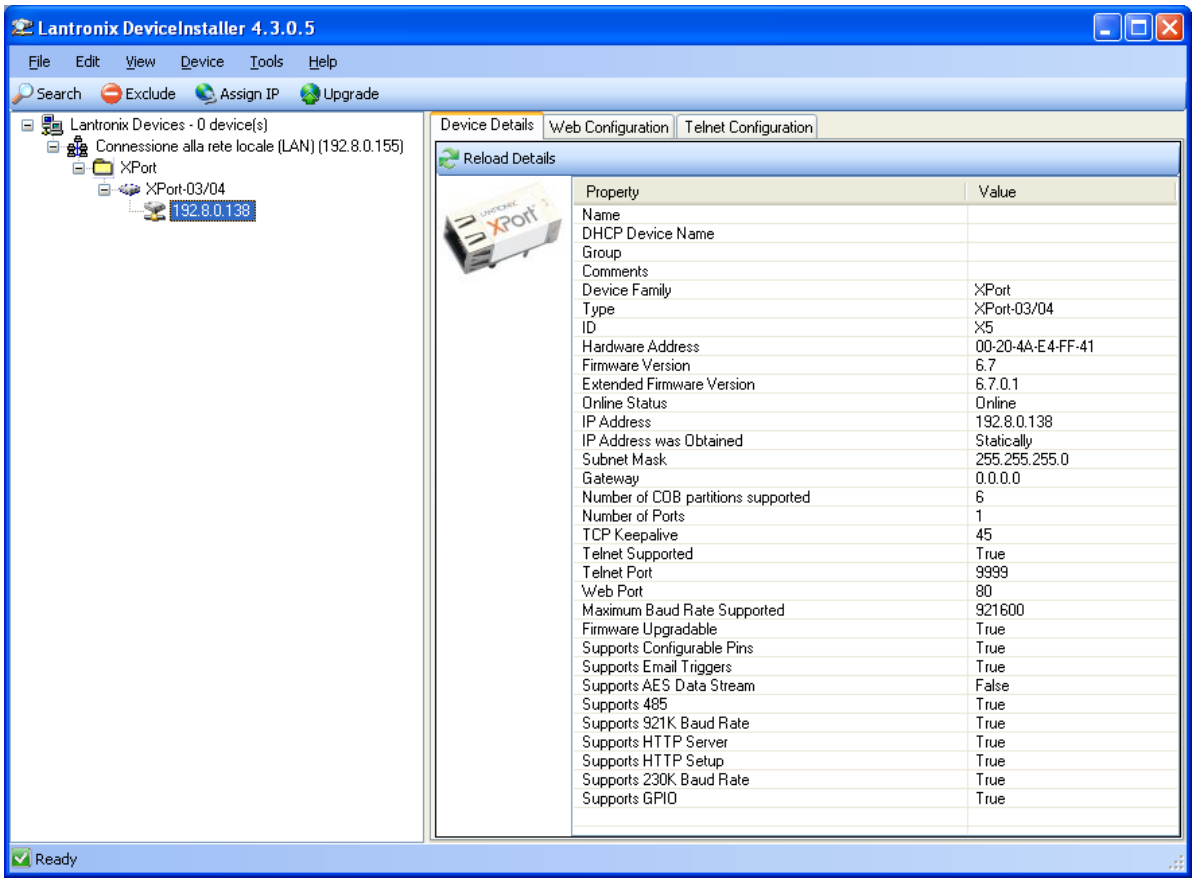
The instrument features a Modbus/TCP port that allows to exchange the weight and the main parameters with a Modbus/TCP *master*.

PC/PLC SETUP

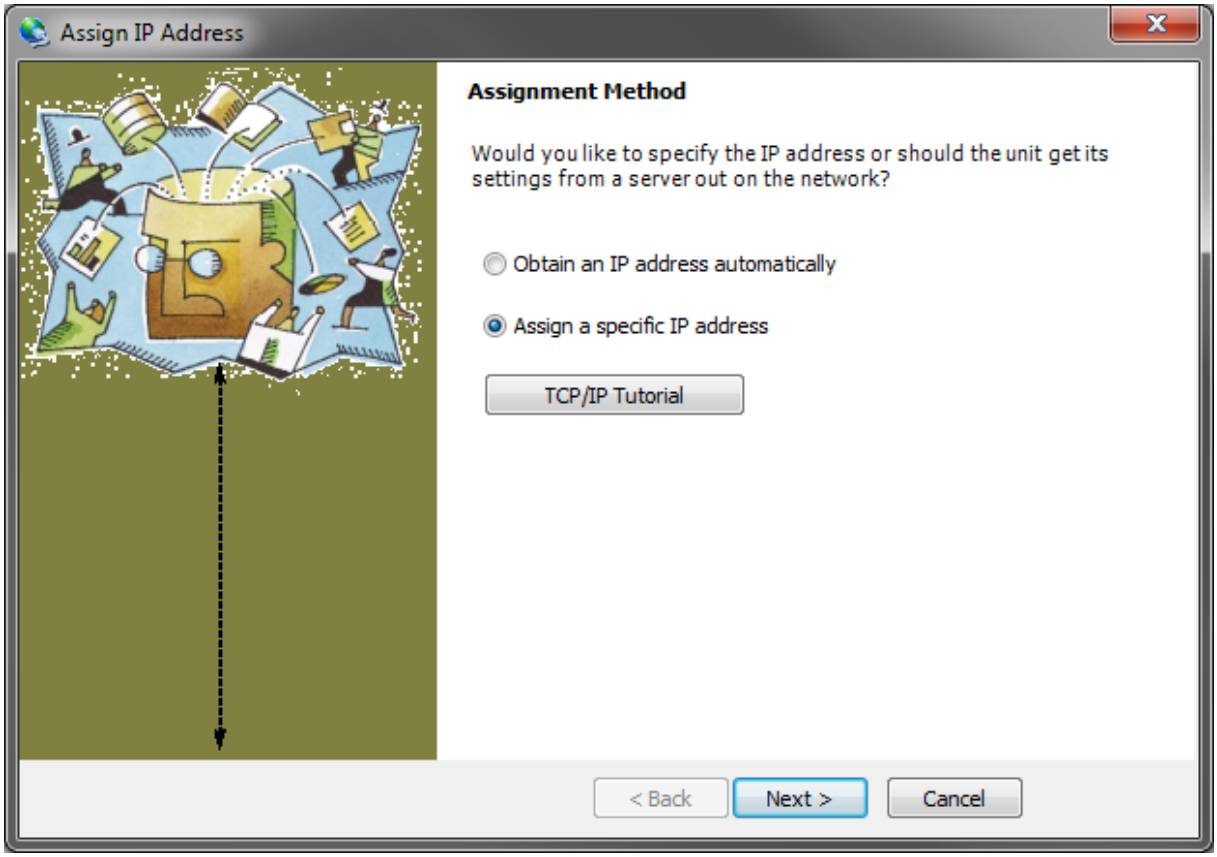
The instrument works as *slave* in a Modbus/TCP network. Use port 502 for the communication.

IP ADDRESS SETTING

Install the Lantronix DeviceInstaller application on a PC with Microsoft Windows operating system (run the *DEVINST.exe* file on the CD). Connect the PC to the instrument via LAN (point-to point or by hub/switch), run the application and click on Search:



Select the device found and click on Assign IP.



Select Assign a specific IP address, enter the desired values and click on Assign; wait for the procedure to complete (no need to restart the instrument).

Modbus/TCP commands and registers are the same as ModbusRTU protocol: for details see section **MODBUS-RTU**.

TECHNICAL SPECIFICATIONS

Port	2x RJ45 10Base-T or 100Base-TX (auto-detect)
Addresses	1÷239

Link/activity LED indications (green)	off .....Ethernet link not established on .....Ethernet link established blinking .....Ethernet activity detected

The instrument features a POWERLINK dual port that allows to exchange the weight and the main parameters with a POWERLINK *controller*.

INSTRUMENT SETUP

[↵] + [⊠] → EtHnEt

- nOdEI d (default: 1): set the instrument address

In order to apply the changes, press [⊠] until the display shows EtHnEt.

PC/PLC SETUP

The instrument works as *slave* in a POWERLINK network.
Load the xdd file attached to the instrument to the POWERLINK *master* development system.
The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Addresses
Gross Weight [4 byte]	GW	0x0000-0x0003
Net Weight [4byte]	NW	0x0004-0x0007
Exchange Register [4 byte]	R1	0x0008-0x000B
Status Register [2 byte]	SR1	0x000C-0x000D
Digital Inputs status [2 byte]	INS	0x000E-0x000F
Digital Outputs status [2 byte]	OUTS	0x0010-0x0011

Input Data to instrument (Writing)	ABBR	Addresses
Command Register [2 byte]	CMDR	0x0000-0x0001
Digital Outputs Command [2 byte]	CMDOUT	0x0002-0x0003
Exchange Register [4 byte]	W1	0x0004-0x0007

TECHNICAL SPECIFICATIONS

Baud rate	Up to 12 Mb/s
Addresses	1÷125

Status LED indications (red)	blinking (fast) .....Bus OK blinking (slow).....Bus error

It is necessary to activate the termination resistance on the two devices located at the ends of the network.

The instrument features a Profibus-DP port that allows to exchange the weight and the main parameters with a Profibus-DP *master*.

INSTRUMENT SETUP

[↵] + [✕] → PrOFI

- Addr (default: 1): set the instrument address in the Profibus network

In order to apply the changes, turn the instrument off, wait for 10 seconds and turn it back on.

PC/PLC SETUP

The instrument works as *slave* in a Profibus-DP network.
Load the gsd file attached to the instrument to the Profibus-DP development system.
Usable software modules are:

NAME	DESCRIPTION	ABBR	R/W	SIZE
TLB4 Gross Weight	Gross Weight	GW	R	4 byte
TLB4 Net Weight	Net Weight	NW	R	4 byte
TLB4 Peak Weight	Peak Weight	PW	R	4 byte
TLB4 Set-Point 1	Setpoint 1	SP1	R/W*	4 byte / 4 byte
TLB4 Set-Point 2	Setpoint 2	SP2	R/W*	4 byte / 4 byte
TLB4 Set-Point 3	Setpoint 3	SP3	R/W*	4 byte / 4 byte
TLB4 Hysteresis 1	Setpoint 1 Hysteresis	HYS1	R/W*	4 byte / 4 byte
TLB4 Hysteresis 2	Setpoint 2 Hysteresis	HYS2	R/W*	4 byte / 4 byte
TLB4 Hysteresis 3	Setpoint 3 Hysteresis	HYS3	R/W*	4 byte / 4 byte
TLB4 Division/Unit	Divisions and Units of Measure	DU	R	2 byte
TLB4 VisualCoeff	Display coefficient	COF	R	4 byte
TLB4 Inputs	Inputs status	INS	R	2 byte
TLB4 Outputs	Outputs status	OUTS	R/W	2 byte / 2 byte
TLB4 Status Reg	Status register	SR1	R	2 byte
TLB4 Command Reg	Command register	CMDR	W	2 byte
TLB4 Exchange Reg**	Exchange register	R1/W1	R/W*	4 byte / 4 byte
TLB4 ZeroAn Weight	Zero Weight-Analog Output	ANA0	R/W*	4 byte / 4 byte
TLB4 FSAn Weight	Full Scale Weight-Analog Output	ANAFS	R/W*	4 byte / 4 byte
TLB4 Divisions 1	Channel 1 divisions		R	4 byte
TLB4 Divisions 2	Channel 2 divisions		R	4 byte
TLB4 Divisions 3	Channel 3 divisions		R	4 byte
TLB4 Divisions 4	Channel 4 divisions		R	4 byte
TLB4 Preset Tare	Preset tare (use with command 130 of the Command Register)	PT	R/W	4 byte / 4 byte

***)** 0x00000000 value in writing is ignored. To reset the value, write out 0x80000000.
****)** It also performs the sample weight register function (CALW), in accordance with previous versions.

TECHNICAL SPECIFICATIONS

Port	2x RJ45 100Base-TX

Link LED indications (green)	off.....Ethernet link not established on.....Ethernet link established
Activity LED indications (amber)	off.....Ethernet activity not detected blinking .....Ethernet activity detected

The instrument features a Profinet-IO dual port that allows to exchange the weight and the main parameters with a Profinet-IO *controller*.

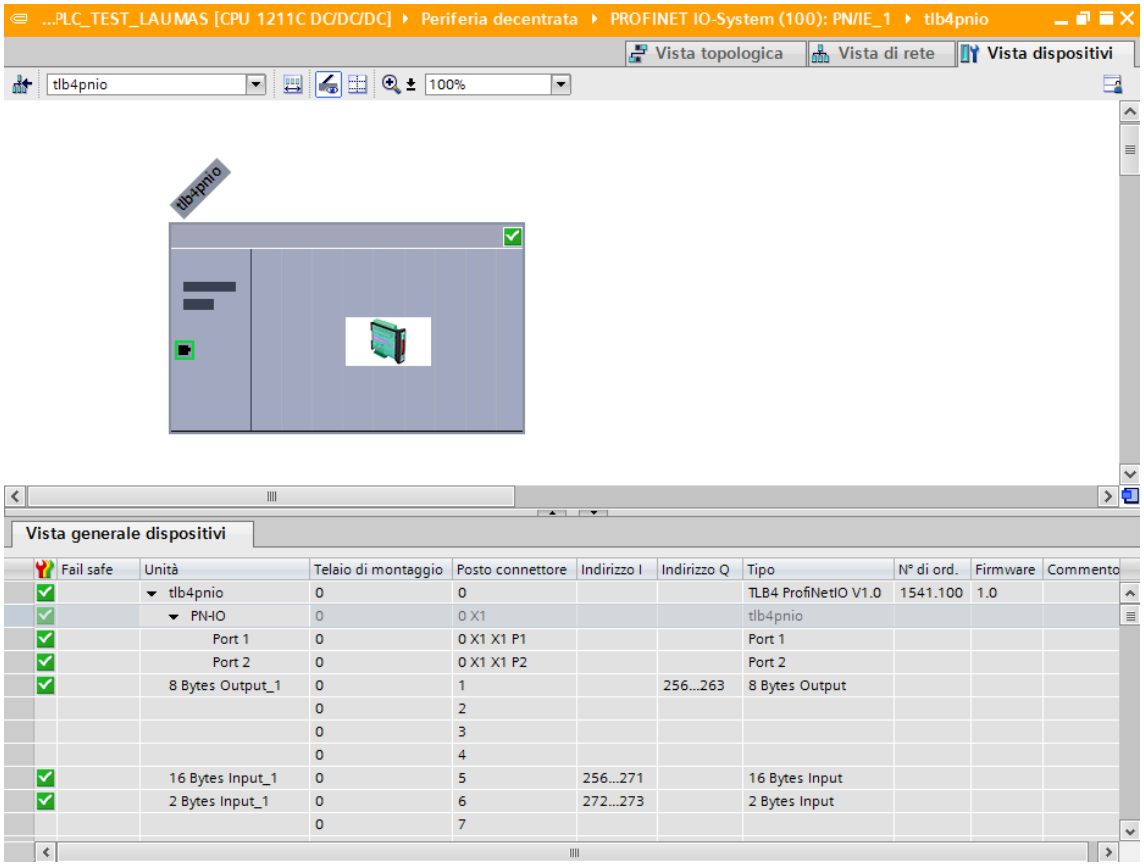
INSTRUMENT SETUP

↵ + ✕ → EtHnEt

- SWAP (default: no): it allows to select the reading/writing of the byte in LITTLE-ENDIAN or BIG-ENDIAN mode
  - YES: LITTLE ENDIAN
  - no: BIG ENDIAN

PC/PLC SETUP

The instrument works as *device* in a Profinet-IO network and supports the MRP Client functionality.
Load the gsdml file attached to the instrument to the Profinet-IO *controller* development system.
Assign a name to the device (function *Assign Device Name*) using the following characters: lower case letters (a-z), numbers (0-9), minus character (-).
Set at least 1 ms as Profinet's I/O refresh time.



The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Addresses	Type
Gross Weight [4 byte]	GW	0x0000-0x0003	16 byte input
Net Weight [4 byte]	NW	0x0004-0x0007	
Exchange Register [4 byte]	R1	0x0008-0x000B	
Status Register [2 byte]	SR1	0x000C-0x000D	
Digital Inputs status [2 byte]	INS	0x000E-0x000F	
Digital Outputs status [2 byte]	OUTS	0x0010-0x0011	2 byte input

Input Data to instrument (Writing)	ABBR	Addresses	Type
Command Register [2 byte]	CMDR	0x0000-0x0001	8 byte output
Digital Outputs Command [2 byte]	CMDOUT	0x0002-0x0003	
Exchange Register [4 byte]	W1	0x0004-0x0007	

TECHNICAL SPECIFICATIONS

Port	2x RJ45 10Base-T or 100Base-TX (auto-detect)
Addresses	1÷511

Link/activity LED indications (green)	off .....Ethernet link not established on .....Ethernet link established blinking .....Ethernet activity detected

The instrument features a SERCOSIII dual port that allows to exchange the weight and the main parameters with a SERCOSIII *master*.

INSTRUMENT SETUP

[↵] + [✕] → EtHnEt

- Addr (default: 1): set the instrument address

In order to apply the changes, press [✕] until the display shows EtHnEt.

PC/PLC SETUP

The instrument works as *slave* in a SERCOSIII network.
Load the sddml file attached to the instrument to the SERCOSIII *master* development system.
The data exchanged by the instrument are:

Output Data from instrument (Reading)	ABBR	Addresses
AT Connection Control [2 byte]		0x0000-0x0001
AT IO Status [2 byte]		0x0002-0x0003
Gross Weight [4 byte]	GW	0x0004-0x0007
Net Weight [4byte]	NW	0x0008-0x000B
Exchange Register [4 byte]	R1	0x000C-0x000F
Status Register [2 byte]	SR1	0x0010-0x0011
Digital Inputs status [2 byte]	INS	0x0012-0x0013
Digital Outputs status [2 byte]	OUTS	0x0014-0x0015

Input Data to instrument (Writing)	ABBR	Addresses
MDT Connection Control [2 byte]		0x0000-0x0001
MDT IO Control [2 byte]		0x0002-0x0003
Command Register [2 byte]	CMDR	0x0004-0x0005
Digital Outputs Command [2 byte]	CMDOUT	0x0006-0x0007
Exchange Register [4 byte]	W1	0x0008-0x000B

AT Connection Control, **AT IO Status**, **MDT Connection Control** and **MDT IO Control** are registers used by the SERCOSIII *master* to manage the communication.

PROGRAMMING OF SYSTEM PARAMETERS

This section contains the commands and procedures for using the instrument through the communication interfaces; the abbreviation of the registers will be used instead of the extended name (see section **FIELDBUSES**).

WARNING: FOR THE DESCRIPTION, THE ALLOWED VALUES AND THE EXAMPLES CONCERNING ALL THE FUNCTIONS MENTIONED IN THIS SECTION, REFER TO THE USER MANUAL OF THE INSTRUMENT.

WEIGHT VALUES

The weight values are expressed as positive integer numbers, including decimal figures, but without decimal point. Read the Status Register (SR1) to get more information on the weight.

THEORETICAL CALIBRATION

THEORETICAL FULL SCALE

When the default theoretical full scale is active, its fieldbus reading returns 0.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 6501 to CMDR

READING

- Send command 6502 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6501 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6501 to CMDR
- Run the CER procedure
- Read the value in R1

SENSITIVITY

The sensitivity values used by the instrument are expressed as six-digit integer numbers. To write the sensitivity in the register, you must first multiply the value by 100000; to read the sensitivity, divide the value in the register by 100000.
Example: to set the sensitivity to 2.00175, write 200175.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Multiply the value by 100000
- Write the value in W1
- Send command 6503 to CMDR

READING

- Send command 6504 to CMDR
- Read the value in R1
- Divide the value by 100000

CC-LINK INTERFACE

WRITING

- Multiply the value by 100000
- Write the value in W1
- Write 0 in RYn3
- Send command 6503 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6503 to CMDR
- Run the CER procedure
- Read the value in R1
- Divide the value by 100000

DIVISION

DIVISIONS						
Index	**Division**	**Active decimals**		**Index**	**Division**	**Active decimals**
0	100	0		10	0.05	2
1	50	0		11	0.02	2
2	20	0		12	0.01	2
3	10	0		13	0.005	3
4	5	0		14	0.002	3
5	2	0		15	0.001	3
6	1	0		16	0.0005	4
7	0.5	1		17	0.0002	4
8	0.2	1		18	0.0001	4
9	0.1	1				

When a parameter is expressed in weight value, it is necessary to consider the number of divisions and active decimals set on the instrument: the value must be multiplied or divided by 10^n (n=active decimals, see table DIVISIONS) and rounded to the set divisions.

Examples: writing a value

Weight value to set	Division	Active decimals	Value to write in the register	Value rounded to the division
100	0.1	1	1000 given by $100x10^1$	100.0
12.00	0.05	2	1200 given by $12.00x10^2$	12.00
33	5	0	33 given by $33x10^0$	35
20.123	0.002	3	20123 given by $20.123x10^3$	20.122

Examples: reading a value

Weight value read by the instrument	Division	Active decimals	Corresponding weight value
1000	0.1	1	100.0 given by $1000/10^1$
1200	0.05	2	12.00 given by $1200/10^2$
35	5	0	35 given by $35/10^0$
20122	0.002	3	20.122 given by $20122/10^3$

MODBUS, PROFIBUS-DP INTERFACES

WRITING

- Write the index* in W1
- Send command 6505 to CMDR

READING

- Read the index* in the least significant byte (L byte) of DU

*see table DIVISIONS

GENERIC INTERFACE

WRITING

- Write the index* in W1
- Send command 6505 to CMDR

READING

- Send command 6506 to CMDR
- Read the index* in R1

*see table DIVISIONS

CC-LINK INTERFACE

WRITING

- Write the index* in W1
- Write 0 in RYn3
- Send command 6505 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6505 to CMDR
- Run the CER procedure
- Read the index* in R1

*see table DIVISIONS

MAXIMUM CAPACITY (BASE program)

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 6511 to CMDR

READING

- Send command 6512 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6511 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6511 to CMDR
- Run the CER procedure
- Read the value in R1

TARE WEIGHT ZERO SETTING

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 100 to CMDR

CC-LINK INTERFACE

- Send command 100 to CMDR
- Run the CER procedure

ZERO VALUE MANUAL ENTRY

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 6527 to CMDR

READING

- Send command 6528 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6527 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6527 to CMDR
- Run the CER procedure
- Read the value in R1

REAL CALIBRATION (WITH SAMPLE WEIGHTS)

In order to correctly set a sample weight of negative value, it is necessary to consider the contents of the register you write in as a 32-bit signed number. If the development system does not handle signed numbers, enter the values in two's complement.
Example: to set the sample weight to -56 kg, enter the value indicated in the table into the register.

DECIMAL VALUE	*HEXADECIMAL VALUE*
-56	0xFFFFFFC8

When acquiring a single calibration point, the instrument replaces the existing calibration with the new one.

MODBUS INTERFACE

ACQUISITION OF A SINGLE CALIBRATION POINT

- Load onto the weighing system a sample weight equal to at least 50% of the maximum quantity to be weighed
- Write the loaded weight value in CALW
- Send command 101 to CMDR
- Read the outcome* of the operation in CALW

* 0=OK

ACQUISITION OF MULTIPLE CALIBRATION POINTS

- Load onto the weighing system a sample weight
- Write the loaded weight value in CALW
- Send command 106 to CMDR
- Read the outcome* of the operation in CALW
- Repeat the procedure up to a maximum of 8 sample weights

CANCELLATION OF THE REAL CALIBRATION

- Send command 104 to CMDR

PROFIBUS-DP INTERFACE

ACQUISITION OF A SINGLE CALIBRATION POINT

- Load onto the weighing system a sample weight equal to at least 50% of the maximum quantity to be weighed
- Write the loaded weight value in W1
- Write 0 in W1
- Send command 101 to CMDR
- Read the outcome* of the operation in R1

* 0=OK

ACQUISITION OF MULTIPLE CALIBRATION POINTS

- Load onto the weighing system a sample weight
- Write the loaded weight value in W1
- Write 0 in W1
- Send command 106 to CMDR
- Read the outcome* of the operation in R1
- Repeat the procedure up to a maximum of 8 sample weights

CANCELLATION OF THE REAL CALIBRATION

- Send command 104 to CMDR

GENERIC INTERFACE

ACQUISITION OF A SINGLE CALIBRATION POINT

- Load onto the weighing system a sample weight equal to at least 50% of the maximum quantity to be weighed
- Write the loaded weight value in W1
- Send command 103 to CMDR
- Send command 101 to CMDR
- Send command 102 to CMDR
- Read the outcome* of the operation in R1

ACQUISITION OF MULTIPLE CALIBRATION POINTS

- Load onto the weighing system a sample weight
- Write the loaded weight value in W1
- Send command 103 to CMDR
- Send command 106 to CMDR
- Send command 102 to CMDR
- Read the outcome* of the operation in R1
- Repeat the procedure up to a maximum of 8 sample weights

* 0=OK

CANCELLATION OF THE REAL CALIBRATION

- Send command 104 to CMDR

CC-LINK INTERFACE

ACQUISITION OF A SINGLE CALIBRATION POINT

- Load onto the weighing system a sample weight equal to at least 50% of the maximum quantity to be weighed
- Write the loaded weight value in W1
- Write 0 in RYn3
- Send command 6555 to CMDR
- Run the CER procedure
- Send command 101 to CMDR
- Run the CER procedure
- Write 1 in RYn3
- Send command 6555 to CMDR
- Run the CER procedure
- Read the outcome* of the operation in R1

ACQUISITION OF MULTIPLE CALIBRATION POINTS

- Load onto the weighing system a sample weight
- Write the loaded weight value in W1
- Write 0 in RYn3
- Send command 6555 to CMDR
- Run the CER procedure
- Send command 106 to CMDR
- Run the CER procedure
- Write 1 in RYn3
- Send command 6555 to CMDR
- Run the CER procedure
- Read the outcome* of the operation in R1
- Repeat the procedure up to a maximum of 8 sample weights

* 0=OK

CANCELLATION OF THE REAL CALIBRATION

- Send command 104 to CMDR
- Run the CER procedure

tYPE

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING	READING
• Write the value* in W1 • Send command 6579 to CMDR	• Send command 6580 to CMDR • Read the value* in R1

*0=MODE0; 1=MODE 1

CC-LINK INTERFACE

WRITING	READING
• Write the value* in W1 • Write 0 in RYn3 • Send command 6579 to CMDR • Run the CER procedure	• Write 1 in RYn3 • Send command 6579 to CMDR • Run the CER procedure • Read the value* in R1

*0=MODE0; 1=MODE 1

tIME

The time required to consider the weight stable is expressed in tenths of a second.
Example: to set up tIME to 2.2 seconds, write 22 in W1.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING	READING
• Write the value in W1 • Send command 6581 to CMDR	• Send command 6582 to CMDR • Read the value in R1

CC-LINK INTERFACE

WRITING	READING
• Write the value in W1 • Write 0 in RYn3 • Send command 6581 to CMDR • Run the CER procedure	• Write 1 in RYn3 • Send command 6581 to CMDR • Run the CER procedure • Read the value in R1

ACTIVE CHANNELS MANAGEMENT

ACR (ACTIVE CHANNELS REGISTER)			
bit 0	channel 1	bit 4	0
bit 1	channel 2	bit 5	0
bit 2	channel 3	bit 6	0
bit 3	channel 4	bit 7	0

bit=1: active channel; bit=0: not active channel;

Example: channels configurations

	ACR CONTENT	CHANNELS CONFIGURATION
Hexadecimal	0x0B	channel 1, channel 2, channel 4: active channel 3: not active
Binary	0b00001011	

After editing, you must repeat equalization, zero setting and calibration using a sample weight.

MANUAL SETTING OF ACTIVE CHANNELS

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the ACR value in W1
- Send command 6575 to CMDR

READING

- Send command 6576 to CMDR
- Read the ACR value in R1

CC-LINK INTERFACE

WRITING

- Write the ACR value in W1
- Write 0 in RYn3
- Send command 6575 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6575 to CMDR
- Run the CER procedure
- Read the ACR value in R1

AUTOMATIC SETTING OF ACTIVE CHANNELS

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 6094 to CMDR

CC-LINK INTERFACE

- Send command 6094 to CMDR
- Run the CER procedure

At the end of the equalization you must perform the tare weight zero setting and, if necessary, the real calibration.

REAL EQUALIZATION

Use a sample weight equal to at least 50% of the single load cell capacity.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

START OF REAL EQUALIZATION
- Send command 6700 to CMDR

ACQUISITION OF THE EQUALIZATION ZERO
- Unload the weighing system
- Wait for the weight to be stable
- Write 0 in W1
- Send command 6701 to CMDR

ACQUISITION OF EQUALIZATION POINTS
- Place the sample weight in correspondence with a load cell
- Wait for the weight to be stable
- Write the index of the equalization point (from 1 to total active channels) in W1
- Send command 6701 to CMDR
- Repeat the procedure moving the sample weight in correspondence with the remaining load cells

CONCLUSION OF THE REAL EQUALIZATION
- Send command 6702 to CMDR

CC-LINK INTERFACE

START OF REAL EQUALIZATION
- Send command 6700 to CMDR
- Run the CER procedure

ACQUISITION OF THE EQUALIZATION ZERO
- Unload the weighing system
- Wait for the weight to be stable
- Write 0 in W1
- Send command 6701 to CMDR
- Run the CER procedure

ACQUISITION OF EQUALIZATION POINTS

- Place the sample weight in correspondence with a load cell
- Wait for the weight to be stable
- Write the index of the equalization point (from 1 to total active channels) in W1
- Send command 6701 to CMDR
- Run the CER procedure
- Repeat the procedure moving the sample weight in correspondence with the remaining load cells

CONCLUSION OF THE REAL EQUALIZATION

- Send command 6702 to CMDR
- Run the CER procedure
- If the CER procedure does not end correctly, the equalization has failed and must be repeat

THEORETICAL EQUALIZATION

The sensitivity values used by the instrument are expressed as six-digit integer numbers. To write the sensitivity in the register, you must first multiply the value by 100000; to read the sensitivity, divide the value in the register by 100000.
Writing example: to set the sensitivity to 2.00175, write 200175.
Reading example: if the read value is 203170, the sensitivity is 2.03170.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING OF SENSITIVITY VALUES

- Write the channel index in W1
- Send command 6703 to CMDR
- Multiply the load cell sensitivity value by 100000
- Write the value in W1 (write 0 if the channel is not connected to the load cell)
- Send command 6563 to CMDR
- Repeat the procedure for all 4 channels

READING OF SENSITIVITY VALUES

- Write the channel index in W1
- Send command 6564 to CMDR
- Read the value in R1
- Divide the value by 100000

CONCLUSION OF THEORETICAL EQUALIZATION

- Send command 6704 to CMDR

CC-LINK INTERFACE

WRITING OF SENSITIVITY VALUES

- Write the channel index in W1
- Send command 6703 to CMDR
- Run the CER procedure
- Multiply the load cell sensitivity value by 100000
- Write the value in W1 (write 0 if the channel is not connected to the load cell)
- Write 0 in RYn3
- Send command 6563 to CMDR
- Run the CER procedure
- Repeat the procedure for all 4 channels

READING OF SENSITIVITY VALUES

- Write the channel index in W1
- Write 1 in RYn3
- Send command 6563 to CMDR
- Run the CER procedure
- Read the value in R1
- Divide the value by 100000

CONCLUSION OF THEORETICAL EQUALIZATION

- Send command 6704 to CMDR
- Run the CER procedure

EQUALIZATION DELETION

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 6046 to CMDR

CC-LINK INTERFACE

- Send command 6046 to CMDR
- Run the CER procedure

FILTER ON THE WEIGHT

The type and level of the filter are set through a 4-byte number in which the two H byte indicate the type of filter and the two L byte indicate the level of the filter.

FILTER ON THE WEIGHT				
LEVEL	Response time* [ms]		Display and serial port refresh frequency* [Hz]	
	tYPE 0	tYPE 1	tYPE 0	tYPE 1
0	12	100	300	100
1	150	330	100	100
2	260	500	50	100
3	425	700	25	100
4	850	1100	12.5	100
5	1700	1600	12.5	100
6	2500	2700	12.5	100
7	4000	3500	10	100
8	6000	5000	10	100
9	7000	6800	5	100
A	6	—	600	—

*indicative values

Example: set the filter on the weight as type 1 and level 5

	H (2 byte)	L (2 byte)	Total
Hexadecimal	0x0001	0x0005	0x00010005
Decimal	1	5	65541

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the level* in the two L byte of W1
- Write the type* in the two H byte of W1
- Send command 6519 to CMDR

READING

- Send command 6520 to CMDR
- Read the level* in the two L byte of R1
- Read the type* in the two H byte of R1

* see table FILTER ON THE WEIGHT

CC-LINK INTERFACE

WRITING

- Write the level* in the two L byte of W1
- Write the type* in the two H byte of W1
- Write 0 in RYn3
- Send command 6519 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6519 to CMDR
- Run the CER procedure
- Read the level* in the two L byte of R1
- Read the type* in the two H byte of R1

* see table FILTER ON THE WEIGHT

ANTI-PEAK

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6541 to CMDR

ENABLING READING

- Send command 6542 to CMDR
- Read the enabling status* in R1

*0=disabled; 1=enabled

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 6541 to CMDR
- Run the CER procedure

ENABLING READING

- Write 1 in RYn3
- Send command 6541 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

*0=disabled; 1=enabled

ZERO PARAMETERS

RESETTABLE WEIGHT SETTING FOR SMALL WEIGHT CHANGES

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 6529 to CMDR

READING

- Send command 6530 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6529 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6529 to CMDR
- Run the CER procedure
- Read the value in R1

AUTOMATIC ZERO SETTING AT POWER-ON

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 6521 to CMDR

READING

- Send command 6522 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 6521 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6521 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

ZERO TRACKING

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 6531 to CMDR

READING

- Send command 6532 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 6531 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6531 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

SETTING UNITS OF MEASURE

UNITS OF MEASURE			
Unit of measure index	**Visualisation**	**Description**	**Display coefficient effect on the weight**
0	HI LOG	Kilograms	No effect
1	G	Grams	No effect
2	t	Tons	No effect
3	Lb	Pounds*	Multiplies
4	nEUtOn	Newton*	Multiplies
5	LI trE	Litres*	Divides
6	bAr	Bar*	Multiplies
7	AtM	Atmospheres*	Multiplies
8	PI ECE	Pieces*	Divides
9	nEU-M	Newton metres*	Multiplies
10	HI LO-M	Kilogram metres*	Multiplies
11	OtHEr	Other*	Multiplies

MODBUS, PROFIBUS-DP INTERFACES

WRITING

- Write the index* in W1
- Send command 6523 to CMDR

READING

- Read the index* in the most significant byte (H byte) of DU

*see table UNITS OF MEASURE

GENERIC INTERFACE

WRITING

- Write the index* in W1
- Send command 6523 to CMDR

READING

- Send command 6524 to CMDR
- Read the index* in R1

*see table UNITS OF MEASURE

CC-LINK INTERFACE

WRITING

- Write the index* in W1
- Write 0 in RYn3
- Send command 6523 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6523 to CMDR
- Run the CER procedure
- Read the index* in R1

*see table UNITS OF MEASURE

DISPLAY COEFFICIENT (BASE program)

For the units marked with * (see table UNITS OF MEASURE) the display coefficient can be set: the value must be multiplied by 10000.
If you intend to use the display coefficient you must enable it. The GW register contains the modified value according to the set coefficient.

Example: to set the display coefficient to 8.5711, write 85711 in W1 as follows:

	H (2 byte)	L (2 byte)	Total
Hexadecimal	0x0001	0x4ECF	0x00014ECF
Decimal	1	20175	85711

MODBUS, PROFIBUS-DP INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6595 to CMDR

*0=disabled; 1=enabled

ENABLING READING

- Send command 6596 to CMDR
- Read the enabling status* in R1

COEFFICIENT WRITING

- Multiply the value by 10000
- Write the value in W1
- Send command 6525 to CMDR

COEFFICIENT READING

- Read the value in COF
- Divide the value by 10000

GENERIC INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6595 to CMDR

*0=disabled; 1=enabled

ENABLING READING

- Send command 6596 to CMDR
- Read the enabling status* in R1

COEFFICIENT WRITING

- Multiply the value by 10000
- Write the value in W1
- Send command 6525 to CMDR

COEFFICIENT READING

- Send command 6526 to CMDR
- Read the value in R1
- Divide the value by 10000

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 6595 to CMDR
- Run the CER procedure

ENABLING READING

- Write 1 in RYn3
- Send command 6595 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

*0=disabled; 1=enabled

COEFFICIENT WRITING

- Multiply the value by 10000
- Write the value in W1
- Write 0 in RYn3
- Send command 6525 to CMDR
- Run the CER procedure

COEFFICIENT READING

- Write 1 in RYn3
- Send command 6525 to CMDR
- Run the CER procedure
- Read the value in R1
- Divide the value by 10000

SEMI-AUTOMATIC TARE (NET/GROSS)

the semi-automatic tare operation is lost upon instrument power-off.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ACTIVATION

- Send command 7 to CMDR

DEACTIVATION

- Send command 9 to CMDR

CC-LINK INTERFACE

ACTIVATION

- Write 1 in RYn5

DEACTIVATION

- Write 1 in RYn6

PRESET TARE

Example: set the preset tare to 69312

	H (2 byte)	L (2 byte)	Total
Hexadecimal	0x0001	0x0EC0	0x00010EC0
Decimal	1	3776	69312

MODBUS, PROFIBUS-DP INTERFACES

WRITING

- Write the value in PT

READING

- Read the value in PT

ACTIVATION

- Send command 130 to CMDR

DEACTIVATION

- Send command 9 to CMDR

GENERIC INTERFACE

WRITING

- Write the value in W1
- Send command 6543 to CMDR

READING

- Send command 6544 to CMDR
- Read the value in R1

ACTIVATION

- Send command 130 to CMDR

DEACTIVATION

- Send command 9 to CMDR

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6543 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6543 to CMDR
- Run the CER procedure
- Read the value in R1

ACTIVATION

- Send command 130 to CMDR
- Run the CER procedure

DEACTIVATION

- Send command 9 to CMDR
- Run the CER procedure

SEMI-AUTOMATIC ZERO (WEIGHT ZERO-SETTING FOR SMALL VARIATIONS)

The zero-setting is lost upon instrument power-off.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 8 to CMDR

CC-LINK INTERFACE

- Write 1 in RYn4

PEAK (BASE program)

MODBUS, PROFIBUS-DP INTERFACES

By enabling the peak function, the value contained in PW is updated with the peak weight value recorded by the instrument; disabling the function, the update is interrupted.

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6597 to CMDR

ENABLING READING

- Send command 6598 to CMDR
- Read the enabling status* in R1

*0=disabled; 1=enabled

GENERIC INTERFACE

By enabling the peak function, the value contained in GW is updated with the peak weight value recorded by the instrument; disabling the function, the update is interrupted.

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6597 to CMDR

ENABLING READING

- Send command 6598 to CMDR
- Read the enabling status* in R1

*0=disabled; 1=enabled

CC-LINK INTERFACE

By enabling the peak function, the value contained in GW is updated with the peak weight value recorded by the instrument; disabling the function, the update is interrupted.

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 6597 to CMDR
- Run the CER procedure

ENABLING READING

- Write 1 in RYn3
- Send command 6597 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

*0=disabled; 1=enabled

ANALOG OUTPUT (TLB4 ONLY)

MODBUS, PROFIBUS-DP INTERFACES

WRITING THE ANALOG OUTPUT ZERO

- Write the value in ANA0

READING THE ANALOG OUTPUT ZERO

- Read the value in ANA0

WRITING THE ANALOG OUTPUT FULL SCALE

- Write the value in ANAFS

READING THE ANALOG OUTPUT FULL SCALE

- Read the value in ANAFS

The threshold values used by the instrument are expressed as integer numbers with a decimal, therefore they must be multiplied by 10.
Example: to set the threshold percentage to 15% it is necessary to write 150.

LOAD DIAGNOSTICS

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6571 to CMDR

*0=disabled; 1=enabled

ENABLING READING

- Send command 6572 to CMDR
- Read the enabling status* in R1

THRESHOLD WRITING

- Multiply the threshold value by 10
- Write the value in W1
- Send command 6567 to CMDR

THRESHOLD READING

- Send command 6568 to CMDR
- Read the value in R1
- Divide the value by 10

CANCELLATION OF THE STORED LOAD DISTRIBUTIONS

- Send command 6072 to CMDR

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 6571 to CMDR
- Run the CER procedure

*0=disabled; 1=enabled

ENABLING READING

- Write 1 in RYn3
- Send command 6571 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

THRESHOLD WRITING

- Multiply the threshold value by 10
- Write the value in W1
- Write 0 in RYn3
- Send command 6567 to CMDR
- Run the CER procedure

THRESHOLD READING

- Write 1 in RYn3
- Send command 6567 to CMDR
- Run the CER procedure
- Read the value in R1
- Divide the value by 10

CANCELLATION OF THE STORED LOAD DISTRIBUTIONS

- Send command 6072 to CMDR
- Run the CER procedure

DIAGNOSTICS ON ZERO

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

TARE WEIGHT ZERO SETTING AND STORAGE OF THE LOAD DISTRIBUTION ON ZERO

- Send command 6122 to CMDR

ENABLING WRITING

- Write the enabling status* in W1
- Send command 6573 to CMDR

*0=disabled; 1=enabled

ENABLING READING

- Send command 6574 to CMDR
- Read the enabling status* in R1

THRESHOLD WRITING

- Multiply the threshold value by 10
- Write the value in W1
- Send command 6569 to CMDR

THRESHOLD READING

- Send command 6570 to CMDR
- Read the value in R1
- Divide the value by 10

CC-LINK INTERFACE

TARE WEIGHT ZERO SETTING AND STORAGE OF THE LOAD DISTRIBUTION ON ZERO

- Send command 6122 to CMDR
- Run the CER procedure

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 6573 to CMDR
- Run the CER procedure

*0=disabled; 1=enabled

ENABLING READING

- Write 1 in RYn3
- Send command 6573 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

THRESHOLD WRITING

- Multiply the threshold value by 10
- Write the value in W1
- Write 0 in RYn3
- Send command 6569 to CMDR
- Run the CER procedure

THRESHOLD READING

- Write 1 in RYn3
- Send command 6569 to CMDR
- Run the CER procedure
- Read the value in R1
- Divide the value by 10

CONFIRMATION OF THE DIAGNOSTICS ERROR

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 6084 to CMDR

CC-LINK INTERFACE

- Send command 6084 to CMDR
- Run the CER procedure

READING OF THE LOAD PERCENTAGES ON EACH CHANNEL

The load percentage values used by the instrument are expressed as integer numbers with a decimal, so they must be divided by 10.
Example: a value of 152 corresponds to a load percentage of 15.2%.

MODBUS INTERFACE

When the function is enabled, the exchange registers contain the value of each channel:

Output Data from instrument (Reading)	Register
Load percentage on channel 1 [2 byte]	40053
Load percentage on channel 2 [2 byte]	40054
Load percentage on channel 3 [2 byte]	40055
Load percentage on channel 4 [2 byte]	40056

ENABLING

- Write the percentage type* in W1
- Send command 6808 to CMDR

DISABLING

- Send command 6809 to CMDR

* 0=total load %; 1=load % without the zeroing component

If it is necessary to execute the same command twice consecutively, send command 0 between the first command and the following one.

READING OF THE LOAD PERCENTAGES

- Write the percentage type in W1:
  0=total load %; 1=load % without the zeroing component
- Send the command relating to the channel concerned (see table) to CMDR
- Read the value in R1

COMMAND	R1 CONTENT	
	H (2 byte)	L (2 byte)
6804	Load percentage on channel 1	Load percentage on channel 2
6805	Load percentage on channel 3	Load percentage on channel 4

CC-LINK INTERFACE

READING OF THE LOAD PERCENTAGES

- Write the percentage type in W1:
  0=total load %; 1=load % without the zeroing component
- Write 0 in RYn3
- Send the command relating to the channel concerned (see table) to CMDR
- Run the CER procedure
- Read the value in R1

COMMAND	R1 CONTENT	
	H (2 byte)	L (2 byte)
6804	Load percentage on channel 1	Load percentage on channel 2
6805	Load percentage on channel 3	Load percentage on channel 4

READING OF THE RESPONSE SIGNALS OF THE CELLS IN mV

The response signals of the load cells in mV are expressed as integers with two decimals, so they must be divided by 100.
Example: a value of 520 corresponds to a mV reading of 5.20 mV.

MODBUS INTERFACE

When the function is enabled, the exchange registers contain the value of each channel:

Output Data from instrument (Reading)	Register
mV on channel 1 [2 byte]	40053
mV on channel 2 [2 byte]	40054
mV on channel 3 [2 byte]	40055
mV on channel 4 [2 byte]	40056

ENABLING

- Send command 6902 to CMDR

DISABLING

- Send command 6903 to CMDR

PROFIBUS-DP, GENERIC INTERFACES

If it is necessary to execute the same command twice consecutively, send command 0 between the first command and the following one.

READING OF THE RESPONSE SIGNALS OF THE CELLS IN mV

- Send the command relating to the channel concerned (see table) to CMDR
- Read the value in R1

COMMAND	R1 CONTENT	
	H (2 byte)	L (2 byte)
6904	mV on channel 1	mV on channel 2
6905	mV on channel 3	mV on channel 4

CC-LINK INTERFACE

READING OF THE RESPONSE SIGNALS OF THE CELLS IN mV

- Write 0 in RYn3
- Send the command relating to the channel concerned (see table) to CMDR
- Run the CER procedure
- Read the value in R1

COMMAND	R1 CONTENT	
	H (2 byte)	L (2 byte)
6904	mV on channel 1	mV on channel 2
6905	mV on channel 3	mV on channel 4

OUTPUTS AND INPUTS CONFIGURATION (BASE program)

OUTPUTS

The configuration of the outputs is set through 4-byte numbers in which the two H byte indicate the number of the output and the two L byte indicate the operating mode of the output.

OCR (OUTPUTS CONFIGURATION REGISTER)				
Bit 7	Bit 6÷5	Bit 4	Bit 3÷1	Bit 0
0 - OFF 1 - On	00 - POSnEG 01 - POS 10 - nEG 11 - not used	0 - GrOSS 1 - nEt	000 - SEt 001 - PLC 010 - StAbLE 011 - ALArM 100 - CHOL	0 - OPEn 1 - CLOSE

Example: configuration of an output

	H (2 byte)	L (2 byte)	Total	OUTPUT CONFIGURATION
Hexadecimal	0x0003	0x00B1	0x000300B1	Output 3/CLOSE/SEt/nEt/POS/On
Decimal	3	177	196705	

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the OCR value in the two L byte of W1
- Write the output number (1÷3) in the two H byte of W1
- Send command 6559 to CMDR

READING

- Write the output number (1÷3) in W1
- Send command 6560 to CMDR
- Read the OCR value in R1

CC-LINK INTERFACE

WRITING

- Write the OCR value in the two L byte of W1
- Write the output number (1÷3) in the two H byte of W1
- Write 0 in RYn3
- Send command 6559 to CMDR
- Run the CER procedure

READING

- Write the output number (1÷3) in W1
- Write 1 in RYn3
- Send command 6559 to CMDR
- Run the CER procedure
- Read the OCR value in R1

INPUTS

The configuration of the inputs is set through 4-byte numbers in which the two H byte indicate the number of the input and the two L byte indicate the operating mode of the input.

INPUTS CONFIGURATION						
Index	1	2	3	4	5	6
Function	nE-LO	2Er0	PEAH	PLC	CONtIn	COEFF

Example: configuration of an input

	H (2 byte)	L (2 byte)	Total	INPUT CONFIGURATION
Hexadecimal	0x0002	0x0002	0x00020002	Input 2/2Er0
Decimal	2	2	131074	

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the index* in the two L byte of W1
- Write the input number (1÷2) in the two H byte of W1
- Send command 6561 to CMDR

*see table INPUTS CONFIGURATION

READING

- Write the input number (1÷2) in W1
- Send command 6562 to CMDR
- Read the index* in R1

CC-LINK INTERFACE

WRITING

- Write the index* in the two L byte of W1
- Write the input number (1÷2) in the two H byte of W1
- Write 0 in RYn3
- Send command 6561 to CMDR
- Run the CER procedure

*see table INPUTS CONFIGURATION

READING

- Write the input number (1÷2) in W1
- Write 1 in RYn3
- Send command 6561 to CMDR
- Run the CER procedure
- Read the index* in R1

READING OF THE DIGITAL INPUTS AND OUTPUTS STATUS

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Read the status of digital inputs in INS and of digital outputs in OUTS (see sections **PARAMETERS OF THE COMMUNICATION INTERFACES** and that of the specific fieldbus for the identification of the registers).

CC-LINK INTERFACE

DIGITAL INPUTS AND OUTPUTS STATUS			
Bit 0	INPUT 1 status	**Bit 16**	OUTPUT 1 status
Bit 1	INPUT 2 status	**Bit 17**	OUTPUT 2 status
Bit 2		**Bit 18**	OUTPUT 3 status
Bit 3		**Bit 19**	
Bit 4		**Bit 20**	
Bit 5÷Bit 15		**Bit 21÷Bit 31**	

Bit=1: high input; Bit=0: low input Bit=1: output is closed; Bit=0: output is open

Example:

INPUT 1	high	OUTPUTS 1 and 2	open
INPUT 2	low	OUTPUT 3	closed

R1 CONTENT						
Bit 0	Bit 1	Bit 2÷15	Bit 16	Bit 17	Bit 18	Bit 19÷31
1	0		0	0	1	

READING OF THE DIGITAL INPUTS AND OUTPUTS STATUS

- Send command 6801 to CMDR
- Run the CER procedure
- Read the value in R1 (see table DIGITAL INPUTS AND OUTPUTS STATUS)

DIGITAL OUTPUTS COMMAND

MODBUS INTERFACE

- Write the digital outputs status in OUTS (see sections **PARAMETERS OF THE COMMUNICATION INTERFACES** and that of the specific fieldbus for the identification of the registers).

This register is used for the BASE program only; it allows to control the outputs set to *PLC* mode (see section **OUTPUTS AND INPUTS CONFIGURATION (BASE program)**).

PROFIBUS-DP, GENERIC INTERFACES

- Write the digital outputs status in CMDOUT (see sections **PARAMETERS OF THE COMMUNICATION INTERFACES** and that of the specific fieldbus for the identification of the registers).

CC-LINK INTERFACE

DIGITAL OUTPUTS COMMAND			
Bit 0	OUTPUT 1 status	**Bit 8**	
Bit 1	OUTPUT 2 status	**Bit 9**	
Bit 2	OUTPUT 3 status	**Bit 10**	
Bit 3		**Bit 11**	
Bit 4		**Bit 12**	
Bit 5		**Bit 13**	
Bit 6		**Bit 14**	
Bit 7		**Bit 15**	Force outputs

Bit=1: output is closed; Bit=0: output is open

Setting bit 15 to 1 on the PLC, the *master* takes control of all the outputs, whatever their setting.

WRITING OF THE DIGITAL OUTPUTS STATUS

- Write the digital outputs status in W1 (see table DIGITAL OUTPUTS COMMAND)
- Write 0 in RYn3
- Send command 6802 to CMDR
- Run the CER procedure

OUTPUTS AND INPUTS CONFIGURATION (LOAD program)

In the LOAD program the inputs and outputs are not configurable but operate as follows:

- INPUT 1: START
- INPUT 2: STOP

- OUTPUT 1: PRESET
- OUTPUT 2: SET
- OUTPUT 3: CYCLE END

SETPOINT

These values are set to zero if the calibration is changed significantly (see sections **THEORETICAL CALIBRATION** and **REAL CALIBRATION (WITH SAMPLE WEIGHTS)**).

Setpoint are stored to RAM and lost upon instrument power off; to save them in EEPROM, so that they are maintained upon instrument power on, a specific command must be sent to CMDR.

MODBUS, PROFIBUS-DP INTERFACES

WRITING

- Write the value in the register SPn*

READING

- Read the value in the register SPn*

*n=setpoint number (see sections **PARAMETERS OF THE COMMUNICATION INTERFACES** and that of the specific fieldbus)

GENERIC INTERFACE

WRITING AND READING COMMANDS

SETPOINT	WRITING	READING
Setpoint 1	93	90
Setpoint 2	94	91
Setpoint 3	95	92

WRITING

- Write the value in W1
- Send command* to CMDR

READING

- Send command* to CMDR
- Read the value in R1

*see table WRITING AND READING COMMANDS

CC-LINK INTERFACE

WRITING AND READING COMMANDS

SETPOINT	WRITING	READING
Setpoint 1	6545	6545
Setpoint 2	6547	6547
Setpoint 3	6549	6549

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command* to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command* to CMDR
- Run the CER procedure
- Read the value in R1

*see table WRITING AND READING COMMANDS

SETPOINT STORAGE IN EEPROM

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 99 to CMDR

CC-LINK INTERFACE

- Send command 99 to CMDR
- Run the CER procedure

HYSTERESIS

These values are set to zero if the calibration is changed significantly (see sections **THEORETICAL CALIBRATION** and **REAL CALIBRATION (WITH SAMPLE WEIGHTS)**).

MODBUS, PROFIBUS-DP INTERFACES

WRITING

- Write the value in the register HYSn*

READING

- Read the value in the register HYSn*

*n=hysteresis number (see sections **PARAMETERS OF THE COMMUNICATION INTERFACES** and that of the specific fieldbus)

GENERIC INTERFACE

WRITING AND READING COMMANDS

HYSTERESIS	WRITING	READING
Hysteresis 1	6583	6584
Hysteresis 2	6585	6586
Hysteresis 3	6587	6588

WRITING

- Write the value in W1
- Send command* to CMDR

READING

- Send command* to CMDR
- Read the value in R1

*see table WRITING AND READING COMMANDS

CC-LINK INTERFACE

WRITING AND READING COMMANDS

HYSTERESIS	WRITING	READING
Hysteresis 1	6583	6583
Hysteresis 2	6585	6585
Hysteresis 3	6587	6587

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command* to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command* to CMDR
- Run the CER procedure
- Read the value in R1

*see table WRITING AND READING COMMANDS

OVERLOAD THRESHOLD FOR SINGLE CHANNEL

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 6577 to CMDR

READING

- Send command 6578 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 6577 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 6577 to CMDR
- Run the CER procedure
- Read the value in R1

BATCHING SEQUENCE

This section only shows the commands of the phases that can be managed via the fieldbus.

BATCHING SEQUENCE PROGRAMMING

Select the number of cycles to run (from 1 to 9999).

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1063 to CMDR

READING

- Send command 1064 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1063 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1063 to CMDR
- Run the CER procedure
- Read the value in R1

BATCHING START

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 201 to CMDR

CC-LINK INTERFACE

- Send command 201 to CMDR
- Run the CER procedure

BATCHING STOP AND PAUSE

The START contact must be open.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

STOP	PAUSE	RESUME
• Send command 204 to CMDR	• Send command 202 to CMDR	• Send command 203 to CMDR

CC-LINK INTERFACE

STOP	PAUSE	RESUME
• Send command 204 to CMDR • Run the CER procedure	• Send command 202 to CMDR • Run the CER procedure	• Send command 203 to CMDR • Run the CER procedure

WAITING PHASE

This operation is required after the SET is opened, only if *CONAnd*=1.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 209 to CMDR

CC-LINK INTERFACE

- Send command 209 to CMDR
- Run the CER procedure

BATCHING DATA READING

Data is only available at the end of the batching phase.

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 1094 to CMDR
- Read the data processing status in R1 (1= data ready; 0= data not ready)
- If data are ready, send command 2100 to CMDR to make them available
- Read the data (see table DATA READING)
- Only if SLAVE=1: send command 250 to CMDR to confirm the data reading and proceed to a new batching

DATA READING

ACTUAL BATCHED WEIGHT	• Send command 2101 to CMDR • Read the value in R1 expressed as 32 bit signed number
STARTING TARE	• Send command 2103 to CMDR • Read the value in R1 expressed as 32 bit signed number

CC-LINK INTERFACE

- Write 1 in RYn3
- Send command 1094 to CMDR
- Run the CER procedure
- Read the data processing status in R1 (1= data ready; 0= data not ready)
- If data are ready, send command 2100 to CMDR to make them available
- Run the CER procedure
- Read the data (see table DATA READING)
- Only if SLAVE=1: send command 250 to CMDR and run the CER procedure to confirm the data reading and proceed to a new batching

DATA READING

ACTUAL BATCHED WEIGHT	• Send command 2101 to CMDR • Run the CER procedure • Read the value in R1 expressed as 32 bit signed number
STARTING TARE	• Send command 2103 to CMDR • Run the CER procedure • Read the value in R1 expressed as 32 bit signed number

BCR (BATCHING CONFIGURATION REGISTER)						
Bit 7	Bit 6	Bit 5÷4	Bit 3	Bit 2	Bit 1	Bit 0
EndnEt	not used	PSS	not used	nEHt P		
				StAbLE	CONAnd	tI ME
0 – nO 1 - YES	0	00 - not used 01 - 1 10 - 2 11 - 3	0	0 – nO 1 - YES	0 – nO 1 - YES	0 – nO 1 - YES

Examples:

BCR CONTENT			PARAMETERS CONFIGURATION				
			EndnEt	PSS	nEHt P		
Binary	Hexadecimal	Decimal			StAbLE	CONAnd	tI ME
00010010	0x12	018	nO	1	nO	YES	nO
10110001	0xB1	177	YES	3	nO	nO	YES
10100110	0xA6	166	YES	2	YES	YES	nO

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the BCR value in W1
- Send command 1081 to CMDR

READING

- Send command 1082 to CMDR
- Read the BCR value in R1

CC-LINK INTERFACE

WRITING

- Write the BCR value in W1
- Write 0 in RYn3
- Send command 1081 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1081 to CMDR
- Run the CER procedure
- Read the BCR value in R1

PROGRAMMING OF BATCHING CONSTANTS

WARNING: FOR THE DESCRIPTION, THE ALLOWED VALUES AND THE EXAMPLES CONCERNING ALL THE FUNCTIONS MENTIONED IN THIS SECTION, REFER TO THE USER MANUAL OF THE INSTRUMENT.

The time values of the batching constants are expressed in tenths of a second.
Example: to set WAITING TIME to 10.2 seconds, write 102 in W1.

MINIMUM WEIGHT

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1003 to CMDR

READING

- Send command 1004 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1003 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1003 to CMDR
- Run the CER procedure
- Read the value in R1

MAXIMUM WEIGHT

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1001 to CMDR

READING

- Send command 1002 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1001 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1001 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

SAFE EMPTYING TIME

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1005 to CMDR

READING

- Send command 1006 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1005 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1005 to CMDR
- Run the CER procedure
- Read the value in R1

WAITING TIME

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1007 to CMDR

READING

- Send command 1008 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1007 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1007 to CMDR
- Run the CER procedure
- Read the value in R1

NO COMPARISON TIME

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1057 to CMDR

READING

- Send command 1058 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1057 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1057 to CMDR
- Run the CER procedure
- Read the value in R1

NO PRODUCT LOAD TIME

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1027 to CMDR

READING

- Send command 1028 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1027 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1027 to CMDR
- Run the CER procedure
- Read the value in R1

NO PRODUCT UNLOAD TIME

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1029 to CMDR

READING

- Send command 1030 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1029 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1029 to CMDR
- Run the CER procedure
- Read the value in R1

FALL

nOdFAL

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING	READING
• Write the value in W1 • Send command 1031 to CMDR	• Send command 1032 to CMDR • Read the value in R1

CC-LINK INTERFACE

WRITING	READING
• Write the value in W1 • Write 0 in RYn3 • Send command 1031 to CMDR • Run the CER procedure	• Write 1 in RYn3 • Send command 1031 to CMDR • Run the CER procedure • Read the value in R1

FALdI u

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING	READING
• Write the value in W1 • Send command 1033 to CMDR	• Send command 1034 to CMDR • Read the value in R1

CC-LINK INTERFACE

WRITING	READING
• Write the value in W1 • Write 0 in RYn3 • Send command 1033 to CMDR • Run the CER procedure	• Write 1 in RYn3 • Send command 1033 to CMDR • Run the CER procedure • Read the value in R1

FALL

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING	READING
• Write the value in W1 • Send command 1009 to CMDR	• Send command 1010 to CMDR • Read the value in R1

CC-LINK INTERFACE

WRITING	READING
• Write the value in W1 • Write 0 in RYn3 • Send command 1009 to CMDR • Run the CER procedure	• Write 1 in RYn3 • Send command 1009 to CMDR • Run the CER procedure • Read the value in R1

TOLERANCE

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1011 to CMDR

READING

- Send command 1012 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1011 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1011 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

SLOW

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1015 to CMDR

READING

- Send command 1016 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1015 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1015 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

TAPPING FUNCTION

SLOW ON

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1017 to CMDR

READING

- Send command 1018 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1017 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1017 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

SLOW OFF

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1019 to CMDR

READING

- Send command 1020 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1019 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1019 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

AUTOTARE

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value* in W1
- Send command 1021 to CMDR

READING

- Send command 1022 to CMDR
- Read the value* in R1

*0=function disabled

CC-LINK INTERFACE

WRITING

- Write the value* in W1
- Write 0 in RYn3
- Send command 1021 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1021 to CMDR
- Run the CER procedure
- Read the value* in R1

*0=function disabled

AUTOTARE DELAY

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

WRITING

- Write the value in W1
- Send command 1023 to CMDR

READING

- Send command 1024 to CMDR
- Read the value in R1

CC-LINK INTERFACE

WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1023 to CMDR
- Run the CER procedure

READING

- Write 1 in RYn3
- Send command 1023 to CMDR
- Run the CER procedure
- Read the value in R1

STABLE TARE

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 1025 to CMDR

ENABLING READING

- Send command 1026 to CMDR
- Read the enabling status* in R1

*0=disabled; 1=enabled

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 1025 to CMDR
- Run the CER procedure

ENABLING READING

- Write 1 in RYn3
- Send command 1025 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

*0=disabled; 1=enabled

CONSUMPTION

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 1039 to CMDR

ENABLING READING

- Send command 1040 to CMDR
- Read the enabling status* in R1

*0=disabled; 1=enabled

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 1039 to CMDR
- Run the CER procedure

ENABLING READING

- Write 1 in RYn3
- Send command 1039 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

*0=disabled; 1=enabled

WAITING CONFIRMATION FROM PC (SLAVE)

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

ENABLING WRITING

- Write the enabling status* in W1
- Send command 1047 to CMDR

*0=disabled; 1=enabled

ENABLING READING

- Send command 1048 to CMDR
- Read the enabling status* in R1

CC-LINK INTERFACE

ENABLING WRITING

- Write the enabling status* in W1
- Write 0 in RYn3
- Send command 1047 to CMDR
- Run the CER procedure

*0=disabled; 1=enabled

ENABLING READING

- Write 1 in RYn3
- Send command 1047 to CMDR
- Run the CER procedure
- Read the enabling status* in R1

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

SET WRITING

- Write the value in W1
- Send command 1087 to CMDR

SET READING

- Send command 1088 to CMDR
- Read the value in R1

PRESET WRITING

- Write the value in W1
- Send command 1089 to CMDR

PRESET READING

- Send command 1090 to CMDR
- Read the value in R1

CANCELLATION OF THE FORMULA

- Send command 1091 to CMDR

CC-LINK INTERFACE

SET WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1087 to CMDR
- Run the CER procedure

SET READING

- Write 1 in RYn3
- Send command 1087 to CMDR
- Run the CER procedure
- Read the value in R1

PRESET WRITING

- Write the value in W1
- Write 0 in RYn3
- Send command 1089 to CMDR
- Run the CER procedure

PRESET READING

- Write 1 in RYn3
- Send command 1089 to CMDR
- Run the CER procedure
- Read the value in R1

CANCELLATION OF THE FORMULA

- Write 0 in RYn3
- Send command 1091 to CMDR
- Run the CER procedure

BATCHING INSTRUMENT STATUS (BIS)

This register contains information relating to the batching phases of the instrument.

BIS (BATCHING INSTRUMENT STATUS)		
Decimal	Binary	Instrument condition
0	0b000000000	Instrument in idle condition (weight displaying)
1	0b000000001	Formulas displaying
2	0b000000010	Batching constants displaying
3	0b000000011	Consumption displaying
4	0b000000100	System parameters displaying
5	0b000000101	Setting of formula number and cycles to batch
6	0b000000110	Instrument in batching condition
7	0b000000111	EMPtY alarm
8	0b000001000	------ alarm
9	0b000001001	CONSP alarm
10	0b000001010	tArEP alarm
11	0b000001011	LOAd alarm
12	0b000001100	Batching in waiting phase
13	0b000001101	Batching paused
14	0b000001110	Batching in cycle end phase
15	0b000001111	UnLOAd alarm
18	0b000010010	FALL alarm
20	0b000010100	Weight not stable
25	0b000011001	tOL alarm
33	0b000100001	ErUEI G alarm
40	0b000101000	Waiting for the PC to read the batching data

PROFIBUS-DP, GENERIC INTERFACES

- Send command 6803 to CMDR
- Read BIS in R1

MODBUS INTERFACE

- Read BIS in IS

CC-LINK INTERFACE

- Send command 6803 to CMDR
- Run the CER procedure
- Read BIS in R1

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

CONSUMPTION READING

- Send command 1083 to CMDR
- Read the value in R1

CONSUMPTION DELETION

- Send command 1085 to CMDR
- Read the value in R1

CC-LINK INTERFACE

CONSUMPTION READING

- Write 1 in RYn3
- Send command 1083 to CMDR
- Run the CER procedure
- Read the value in R1

CONSUMPTION DELETION

- Write 0 in RYn3
- Send command 1085 to CMDR
- Run the CER procedure
- Read the value in R1

ALARM MANAGEMENT

The status of the instrument alarms is expressed through 4-byte numbers in which the two H byte represent ERC and the two L byte represent AERC.

ERC error code		**AERC auxiliary error code**		**Cause of error**
Decimal	Binary	Decimal	Binary	
Load cell error				
1	0b000000001	1	0b000000001	ErCELr
		2	0b000000010	ErCEL 1
		4	0b000000100	ErCEL2
		8	0b000001000	ErCEL3
		16	0b000010000	ErCEL4
Instrument error				
4	0b000000100	1	0b000000001	------
		2	0b000000010	Er OL
		4	0b000000100	Er Ad
		16	0b000010000	Gross weight over the maximum displayable value
		32	0b000100000	Net weight over the maximum displayable value
Diagnostics error, if the load diagnostics is enabled (AUt=YES)				
8	0b000001000	1	0b000000001	Load percentage on channel 1 > ErSEt
		2	0b000000010	Load percentage on channel 2 > ErSEt
		4	0b000000100	Load percentage on channel 3 > ErSEt
		8	0b000001000	Load percentage on channel 4 > ErSEt

Diagnostics error, if the diagnostics on zero is enabled (AUt 0=YES)				
8	0b000001000	1	0b000000001	Load percentage on channel 1 > ErSEt0
		2	0b000000010	Load percentage on channel 2 > ErSEt0
		4	0b000000100	Load percentage on channel 3 > ErSEt0
		8	0b000001000	Load percentage on channel 4 > ErSEt0

Writing error				
16	0b000010000	1	0b000000001	The data to be saved is incorrect

Batching error				
32	0b000100000	2	0b000000010	CONSP
		3	0b000000011	tArEP
		4	0b000000100	LOAd
		5	0b000000101	UnLOAd
		7	0b000000111	FALL
		10	0b000001010	tOL
		11	0b000001011	ErUEI G
		12	0b000001100	SLAuE
		13	0b000001101	EMPtY

Overload error				
64	0b001000000	1	0b000000001	CHOL 1
		2	0b000000010	CHOL 2
		4	0b000000100	CHOL 3
		8	0b000001000	CHOL 4

Command error				
256	0b100000000	0	0b000000000	The received command does not exist

READING OF THE ALARMS STATUS

MODBUS, PROFIBUS-DP, GENERIC INTERFACES

- Send command 6800 to CMDR
- Read ERC in the two H byte of R1
- Read AERC in the two L byte of R1

CC-LINK INTERFACE

- Send command 6800 to CMDR
- Run the CER procedure
- Read ERC in the two H byte of R1
- Read AERC in the two L byte of R1

ALARMS MANAGEMENT DURING THE BATCHING

ACTIVE ALARM	MODBUS, PROFIBUS-DP, GENERIC INTERFACES	CC-LINK INTERFACE
tAr EP	• Send command 206 to CMDR to cancel the alarm and continue with the batching	• Send command 206 to CMDR to cancel the alarm and continue with the batching • Run the CER procedure
tOL	• Send command 207 to CMDR to cancel the alarm and continue with the batching	• Send command 207 to CMDR to cancel the alarm and continue with the batching • Run the CER procedure
EMPtY	• Send command 205 to CMDR to accept the alarm and stop the batching	• Send command 205 to CMDR to accept the alarm and stop the batching • Run the CER procedure
ErUEI G		
FALL		
PArSEr		

READING DIVISIONS WITH SIGN OF EACH WEIGHTING CHANNEL

When this operating mode is enabled, the output data from the instrument transmits the points of each connected load cell, in low (16 bit) or high (24 bit) resolution.

Input signal on single channel	Low resolution	High resolution
0 mV	0	0
10 mV	8000	2000000
-10 mV	-8000	-2000000

Only the points of each connected load cell are transmitted, without any filter applied; the calculation of the weight value and the zeroing and calibration operations are carried out by the customer.

Mode: 4x divisions LowRes

MODBUS INTERFACE

ENABLING

- Send command 24 to CMDR

DISABLING

- Send command 27 to CMDR

Output Data from instrument (Reading)	Register
Channel 1 reading divisions – Low Res [2 byte]	40051
Channel 2 reading divisions – Low Res [2 byte]	40052
Channel 3 reading divisions – Low Res [2 byte]	40053
Channel 4 reading divisions – Low Res [2 byte]	40054

GENERIC INTERFACE

ENABLING

- Send command 24 to CMDR

DISABLING

- Send command 27 to CMDR

Output Data from instrument (Reading)	Adresses
Channel 1 reading divisions – Low Res [2 byte]	0x0000-0x0001
Channel 2 reading divisions – Low Res [2 byte]	0x0002-0x0003
Channel 3 reading divisions – Low Res [2 byte]	0x0004-0x0005
Channel 4 reading divisions – Low Res [2 byte]	0x0006-0x0007
-	0x0008-0x000F

CC-LINK INTERFACE

ENABLING

- Send command 24 to CMDR
- Run the CER procedure

DISABLING

- Send command 27 to CMDR
- Run the CER procedure

RWr			
Output Data from instrument (Reading)	**Adresses**		
	1 station	**2 stations**	**4 stations**
Channel 1 reading divisions – Low Res [2 byte]	*	Wr0000	Wr0000
Channel 2 reading divisions – Low Res [2 byte]	*	Wr0001	Wr0001
Channel 3 reading divisions – Low Res [2 byte]	*	Wr0002	Wr0002
Channel 4 reading divisions – Low Res [2 byte]	*	Wr0003	Wr0003
-	*	Wr0004-Wr0007	Wr0004-Wr0007
Status Register [2 byte]	*	*	Wr0008

*data not available for this number of stations

Mode: 4x divisions HiRes

MODBUS INTERFACE

ENABLING

- Send command 25 to CMDR

DISABLING

- Send command 27 to CMDR

Output Data from instrument (Reading)	Registers
Channel 1 reading divisions – High Res [4 byte]	40051-40052
Channel 2 reading divisions – High Res [4 byte]	40053-40054
Channel 3 reading divisions – High Res [4 byte]	40055-40056
Channel 4 reading divisions – High Res [4 byte]	40057-40058

PROFIBUS-DP INTERFACE

The PROFIBUS-DP protocol has special software modules for the transmission of the divisions with sign in high resolution (24 bit) of each weighing channel.

NAME	DESCRIPTION	R/W	DIMENSION
TLB4 Divisions 1	Channel 1 divisions	R	4 byte
TLB4 Divisions 2	Channel 2 divisions	R	4 byte
TLB4 Divisions 3	Channel 3 divisions	R	4 byte
TLB4 Divisions 4	Channel 4 divisions	R	4 byte

GENERIC INTERFACE

ENABLING

- Send command 25 to CMDR

DISABLING

- Send command 27 to CMDR

Output Data from instrument (Reading)	Adresses
Channel 1 reading divisions – High Res [4 byte]	0x0000-0x0003
Channel 2 reading divisions – High Res [4 byte]	0x0004-0x0007
Channel 3 reading divisions – High Res [4 byte]	0x0008-0x000B
Channel 4 reading divisions – High Res [4 byte]	0x000C-0x000F

CC-LINK INTERFACE

ENABLING

- Send command 25 to CMDR
- Run the CER procedure

DISABLING

- Send command 27 to CMDR
- Run the CER procedure

RWr			
Output Data from instrument (Reading)	Adresses		
	1 station	2 stations	4 stations
Channel 1 reading divisions – High Res [4 byte]	*	Wr0000-Wr0001	Wr0000-Wr0001
Channel 2 reading divisions – High Res [4 byte]	*	Wr0002-Wr0003	Wr0002-Wr0003
Channel 3 reading divisions – High Res [4 byte]	*	Wr0004-Wr0005	Wr0004-Wr0005
Channel 4 reading divisions – High Res [4 byte]	*	Wr0006-Wr0007	Wr0006-Wr0007
Status Register [2 byte]	*	*	Wr0008

* data not available for this number of stations

EXAMPLE OF USE OF CONVERTER POINTS

When the instrument is connected to a system, it can be used to read the weight directly from the protocol or, alternatively, the weight can be calculated by the PC or PLC system to which the instrument is connected. In the latter case, the calibration can be performed as follows: if the transmitter sends e.g. 6500 divisions (converter points) when the tank is empty, and after putting a sample weight of 10000 kg it sends 49833 divisions, in order to find out the weight you will simply need to subtract from the divisions read those relative to the empty tank and then divide the result by a constant given by the following calculation:

(49833-6500)/10000=4,333

so, if the PC or PLC receives 40000 divisions, the weight will be given by:

(40000-6500)/4,333=7731 kg

FAST CONTINUOUS TRANSMISSION PROTOCOL

This protocol allows the continuous transmission of the weight at high update frequencies. Up to 300 strings per second are transmitted with a minimum transmission rate of 38400 baud.
Following communication modes availables (see section **SERIAL COMMUNICATION SETTING** in instrument manual):

- *MOd t*: communication compatible with TX RS485 instruments
- *MOd td*: communication compatible with TD RS485 instruments

If *MOd t* is set, the following string is transmitted to PC/PLC:
xxxxxxCRLF

where: **xxxxxx**.....6 characters of gross weight (48 ÷ 57 ASCII)
CR.....1 character return to the start (13 ASCII)
LF.....1 character on new line (10 ASCII)

In case of negative weight, the first character from the left of the weight characters takes on the value "-" (minus sign - ASCII 45).

If *St bIt=YES* (see section **SERIAL COMMUNICATION SETTING** in instrument manual), the following string is transmitted to PC/PLC:

yxxxxxxCRLF

where: **y** .....1 character of weight stability identification (S=weight stable, N=weight not stable)

In case of error or alarm, the 6 characters of the weight are substituted by the messages found in the table of the ALARMS section (see the instrument manual).

If mOd td is set, the following string is transmitted to PC/PLC:
&TzzzzzzPzzzzzz\ckckCR

where: **&**.....1 initial string character (38 ASCII)
T.....1 character of gross weight identification
P.....1 character of gross weight identification
zzzzzz.....6 characters of gross weight (48 ÷ 57 ASCII)
****.....1 character of separation (92 ASCII)
ckck.....2 ASCII control characters or calculated considering the characters included between "&" and "\" excluded. The control value is obtained executing the XOR operation (exclusive OR) for the 8 bit ASCII codes of the characters considered. Therefore, a character expressed in hexadecimal is obtained with 2 numbers that may assume values from "0" to "9" and from "A" to "F". "**ckck**" is the ASCII code of the two hexadecimal digits
CR.....1 character of end string (13 ASCII)

In case of negative weight, the first character from the left of the weight characters takes on the value "-" (minus sign - ASCII 45).
In case of error or alarm, the 6 characters of the gross weight are substituted by the messages found in the table of the ALARMS section (see the instrument manual).

FAST TRANSMISSION VIA EXTERNAL CONTACT: it's possible to transmit the weight, just once, even closing an input for no more than a second (see sections **OUTPUTS AND INPUTS CONFIGURATION** and **SERIAL COMMUNICATION SETTINGS** in instrument manual).

CONTINUOUS WEIGHT TRANSMISSION TO REMOTE DISPLAYS PROTOCOL

This protocol allows the continuous weight transmission to remote displays. The communication string is transmitted 10 times per second.
Following communication modes availables (see section **SERIAL COMMUNICATION SETTING** in instrument manual):

- *rIP*: communication with RIP5/20/60, RIP50SHA, RIPLED series remote displays; the remote display shows the net weight or gross weight according to its settings
- *HdrIP*: communication with RIP6100, RIP675, RIP6125C series remote displays; the remote display shows the net weight or gross weight according to its settings
- *HdrIPn*: communication with RIP6100, RIP675, RIP6125C series remote displays

The instrument sends the following string to the remote display:
&NxxxxxxLyyyyyy\ckckCR

where: **&**.....1 initial string character (38 ASCII)
N.....1 character of net weight identification (78 ASCII)
xxxxxx.....6 characters of net weight or PEAK if present (48 ÷ 57 ASCII)
L.....1 character of gross weight identification (76 ASCII)
yyyyyy.....6 characters of gross weight (48 ÷ 57 ASCII)
\.....1 character of separation (92 ASCII)
ckck.....2 ASCII checksum characters calculated considering the characters between "&" and "\" excluded. The checksum value is obtained from the calculation of XOR (exclusive OR) of the 8-bit ASCII codes of the characters considered. This obtains a character expressed in hexadecimals with two digits that can have the values from "0" to "9" and from "A" to "F". "ckck" is the ASCII code of the two hexadecimal digits
CR.....1 character of end string (13 ASCII)

In case of negative weight, the first character from the left of the weight characters takes on the value "-" (minus sign - ASCII 45).
If *HdrIP* has been set, the decimal point at the position shown on the instrument's display can also be transmitted. In this case, if the value exceeds 5 digits, only the 5 most significant digits are transmitted, while if the value is negative, no more than the 4 most significant digits are transmitted. In both cases, however, the decimal point shifts consistently with the value to display.
If *HdrIPn* has been set, in addition to what stated in *HdrIP* protocol, the instrument transmits the prompt *nEt* every 4 seconds in the gross weight field, if on the instrument, it has been carried out a net operation (see section **SEMI-AUTOMATIC TARE (NET/GROSS)** in instrument manual).
In case of weight value is under -99999, the minus sign "-" is sent alternated with the most significant figure.
In case of error or alarm, the 6 characters of the gross weight and net weight are substituted by the messages found in the table of the ALARMS section (see the instrument manual).

ASCII BIDIRECTIONAL PROTOCOL

The instrument replies to the requests sent from a PC/PLC.
It is possible to set a waiting time for the instrument before it transmits a response (see dELAY parameter in section **SERIAL COMMUNICATION SETTING** in instrument manual).
Following communication modes availables (see section **SERIAL COMMUNICATION SETTING** in instrument manual):

- MOdU60: communication compatible with instruments series W60000, WL60 Base, WT60 Base, TLA600 Base
- MOd td: communication compatible with TD RS485 instruments

Captions:

$.....Beginning of a request string (36 ASCII)
& or **&&** .....Beginning of a response string (38 ASCII)
aa.....2 characters of instrument address (48 ÷ 57 ASCII)
!.....1 character to indicate the correct reception (33 ASCII)
?.....1 character to indicate a reception error (63 ASCII)
#.....1 character to indicate an error in the command execution (23 ASCII)
ckck:.....2 ASCII characters of Check-Sum (for further information, see section **CHECK-SUM CALCULATION**)
CR.....1 character for string end (13 ASCII)
****.....1 character of separation (92 ASCII)

- **SETPOINT PROGRAMMING**

Warning: the new values of setpoint are active immediately.

The PC transmits the ASCII string: **$aaxxxxxxyckckCR**

where: **xxxxxx**.....6 characters for the setpoint value (48 ÷ 57 ASCII)
y=A .....set the value in the setpoint 1
y=B .....set the value in the setpoint 2
y=C .....set the value in the setpoint 3

Possible instrument responses:

- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

Example: to set 500 in the setpoint no. 3, the PC must transmit the following command:
$01000500C47(Cr)

- **SETPOINT STORAGE IN EEPROM MEMORY**

The setpoint are stored in the RAM memory and lost upon instrument power off. It is necessary to send a special command to save them permanently in the EEPROM memory. Please note that the writing number allowed in the EEPROM memory is limited (about 100000).

The PC transmits the ASCII string: **$aaMEMckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **READING WEIGHT, SETPOINT AND PEAK (IF PRESENT) FROM PC**

The PC transmits the ASCII string: **$aajckckCR**

where: **j**=a.....to read setpoint 1
j=b.....to read setpoint 2
j=c.....to read setpoint 3
j=t.....to read gross weight
j=n.....to read net weight
j=p .....to read the gross weight peak if the ASCII parameter is set as MODBUS; if, instead, the ASCII parameter is set on MOd Ed the gross weight will be read.
To read the points, set the FS_tEO parameter equal to 50000

Possible instrument responses:
- correct reception: **&aaxxxxxxj\ckckCR**
- incorrect reception: **&&aa?\ckckCR**
- In case of peak not configured: **&aa#CR**

where: **xxxxxx**.....6 characters of the required weight value

Notes: in case of negative weight, the first character from the left of the weight characters takes on the value "-" (minus sign - ASCII 45). In case of weight value is under -99999, the minus sign "-" is sent alternated with the most significant figure.

Error messages:
in case of an instrument alarm for exceeding 110% of the full scale or 9 divisions above the value of the parameter MASS, the instrument sends the string:
&aassO-Lst\ckck

in case of faulty connection of the load cells or of another alarm, the instrument sends:
&aassO-Fst\ckck

where: **s**.....1 separator character (32 ASCII – space)

Generally refer to the section **ALARMS** (see the instrument manual).

- **SEMI-AUTOMATIC ZERO (WEIGHT ZERO-SETTING FOR SMALL VARIATIONS)**

The PC transmits the ASCII string: **$aaZEROckckCR**

- Possible instrument responses:
  - correct reception: **&&aa!\ckckCR**
  - incorrect reception: **&&aa?\ckckCR**
  - the current weight is over the maximum resettable value: **&aa#CR**

- **SWITCHING FROM GROSS TO NET WEIGHT**

The PC transmits the ASCII string: **$aaNETckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **SWITCHING FROM NET TO GROSS WEIGHT**

The PC transmits the ASCII string: **$aaGROSSckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **READING OF DECIMALS AND DIVISION NUMBER**

The PC transmits the ASCII string: **$aaDckckCR**

Possible instrument responses:
- correct reception: **&aaxy\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

where: **x**.....number of decimals
y=3.....for division value=1
y=4.....for division value=2
y=5.....for division value=5
y=6.....for division value=10
y=7.....for division value=20
y=8.....for division value=50
y=9.....for division value=100

- **TARE WEIGHT ZERO SETTING**

The PC transmits the ASCII string: **$aazckckCR**

where: **z**.....command of weight zero-setting (122 ASCII)

Possible instrument responses:
- correct reception: **&aaxxxxxxt\ckckCR**
- incorrect reception: **&&aa?\ckckCR**
- the gross weight is not displayed on the instrument: **&aa#CR**

where: **xxxxxx**.....6 characters to indicate the required weight value
t.....character to indicate the weight (116 ASCII)

Example: zeroing the weight of the instrument with address 2
For the calibration you have to make sure that the system is unloaded or that the instrument measures a signal equal to the mV in the same condition:
query: **$02z78(Cr)**
response: **&02000000t\76(Cr)**

If the zeroing works correctly the instrument sends the zeroed weight value ("000000").

The calibration values are stored permanently in the EEPROM memory and the number of allowed writings is limited (about 100000).

- **REAL CALIBRATION (WITH SAMPLE WEIGHT)**

After the tare zero-setting, this function allow the operator to check the calibration obtained by using sample weights and correct automatically any change between the displayed value and the actual one.

Load onto the weighing system a sample weight, which must be at least 50% of the full scale, or make so that that the instrument measures a corresponding mV signal.

The PC transmits the ASCII string: **$aasxxxxxxckckCR**

where: **s**.....calibration command (115 ASCII)
xxxxxx.....6 characters to indicate the value of sample weight (negative values are not allowed)

Possible instrument responses:
- correct reception: **&aaxxxxxxt\ckckCR**
- incorrect reception or full scale equal to zero: **&&aa?\ckckCR**

where: **t**.....character of gross weight identification (116 ASCII)
xxxxxx.....6 characters to indicate the value of current weight

In case of correct reception, the read value has to be equal to the sample weight.

Example: calibration of the instrument no. 1 with a sample weight of 20000 kg:
query: **$01s02000070(Cr)**
response: **&01020000t\77(Cr)**

In case of correct calibration, the read value has to be “020000”.

- **KEYPAD LOCK (BLOCK THE ACCESS TO THE INSTRUMENT)**

The PC transmits the ASCII string: **$aaKEYckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **KEYPAD UNLOCK**

The PC transmits the ASCII string: **$aaFREckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **DISPLAY AND KEYPAD LOCK**

The PC transmits the ASCII string: **$aaKDISckckCR**

Possible instrument responses:
- correct reception: **&&aa!\ckckCR**
- incorrect reception: **&&aa?\ckckCR**

- **CHECK-SUM CALCULATION**

The two ASCII characters (**ckck**) are the representation of a hexadecimal digit in ASCII characters. The check digit is calculated by executing the operation of XOR (exclusive OR) of 8-bit ASCII codes of only the string underlined.

The procedure to perform the calculation of check-sum is the following:
- Consider only the string characters highlighted with underlining
- Calculate the exclusive OR (XOR) of 8-bit ASCII codes of the characters

Example:

character	decimal ASCII code	hexadecimal ASCII code	binary ASCII code
0	48	30	00110000
1	49	31	00110001
t	116	74	01110100
XOR =	117	75	01110101

- The result of the XOR operation expressed in hexadecimal notation is made up of 2 hexadecimal digit (that is, numbers from 0 to 9 and/or letters from A to F). In this case the hexadecimal code is 0x75.
- The checksum is made up of the 2 characters that represent the result of the XOR operation in hexadecimal notation (in our example the character "7" and the character "5").

On our website www.laumas.com there are videos on the guidelines for correct installation of weighing systems and video tutorials on configuring our transmitters and weight indicators.

All Laumas product manuals are available online. You can download the manuals in PDF format from www.laumas.com by consulting the Products section or the Download Area.
Registration is required.

Think about the environment before you print!
CERTIFICATION OF THE ENVIRONMENTAL MANAGEMENT SYSTEM
in accordance with UNI EN ISO 14001.
Laumas contributes to environmental protection by saving on paper consumption.