SPD

USER MANUAL

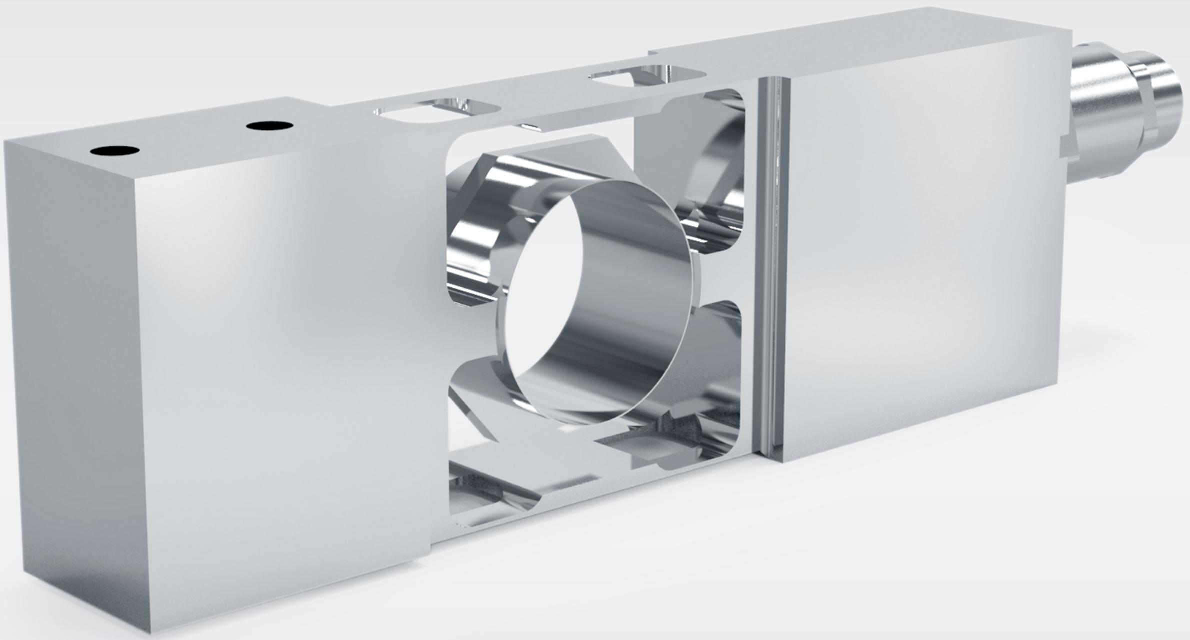


TABLE OF CONTENTS

1. APPLICATIONS

The type SPD is a digital version of single point load cell with complete hermetic sealing. It is a perfect fit for use in harsh industrial environments and wash-down applications.

- Bench scales
- Conveyor scales
- Filling machines
- Packaging machines
- Check weighers
- Industrial process control

2. CHARACTERISTIC FEATURES

- Capacities from 10 to 50 kg.
- Stainless steel construction.
- Environmental protection IP69k with complete hermetic sealing.
- AD conversion rate up to 1200upd/sec.
- Free professional software for setting up the digital load cell.
- Maximum platform size up to 450 x 450 mm.

3. INTERFACES

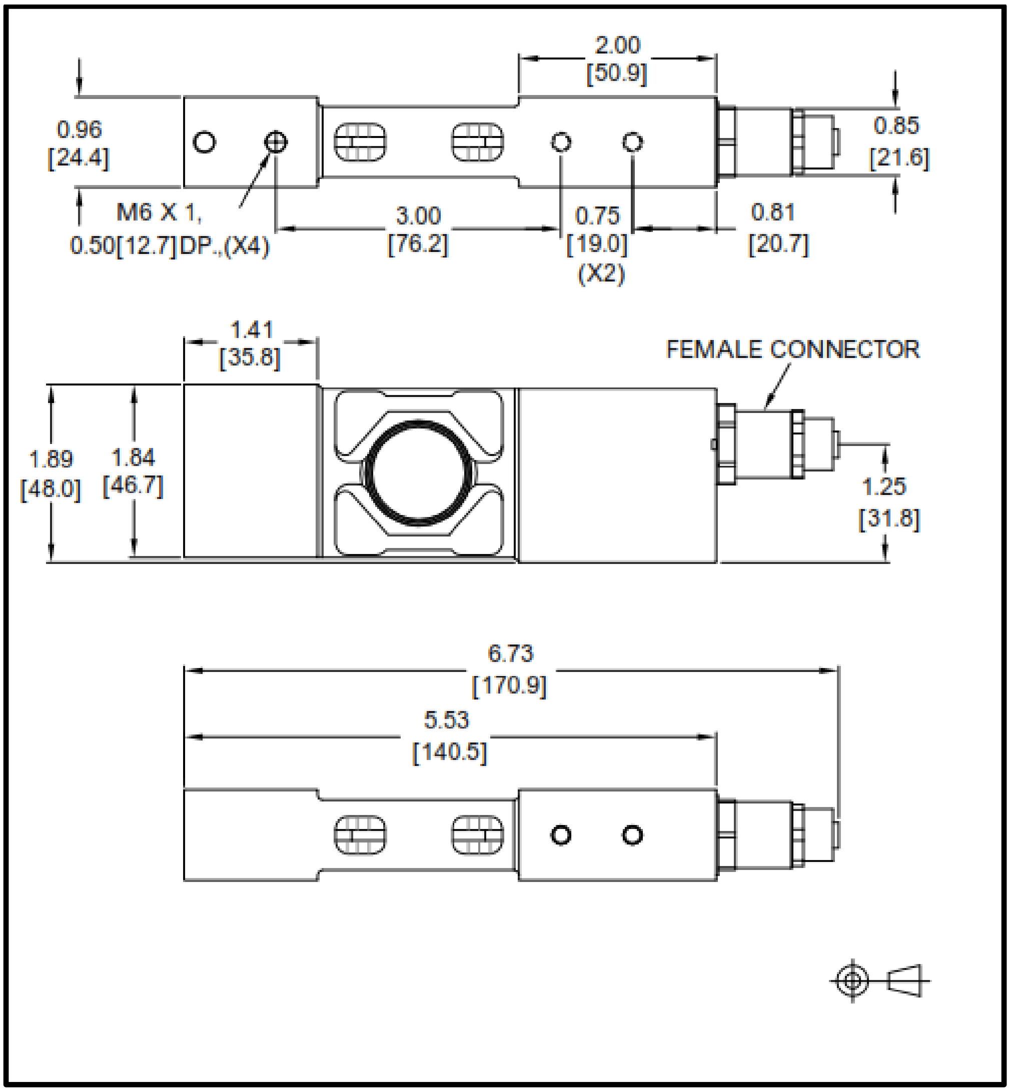
SPD load cells are available with the following interfaces.

1. RS232 and CANopen
2. RS485 and RS422 (Both 4 wires)

Additional details are in section 6.3.

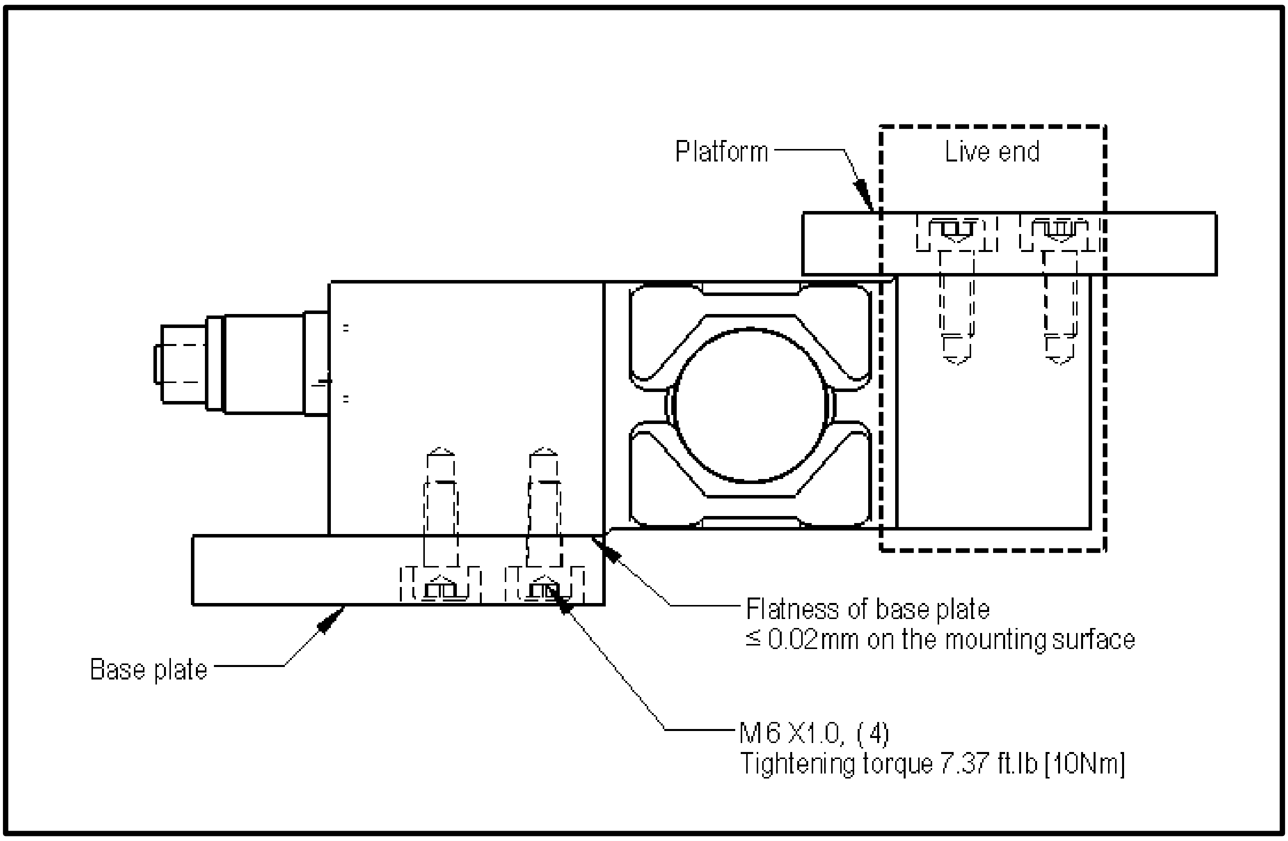
4. MECHANICAL CONSTRUCTIONS

4.1 Outline dimensions



*All Dimensions are in INCH [mm]

5. INSTALLATION INSTRUCTIONS



Note: During installation consider following measures in preventing damages to 20kg and below load cells.

Step 1:- Clamp the live end of the load cell to a vice, mount the platform and tighten to the specified torque.

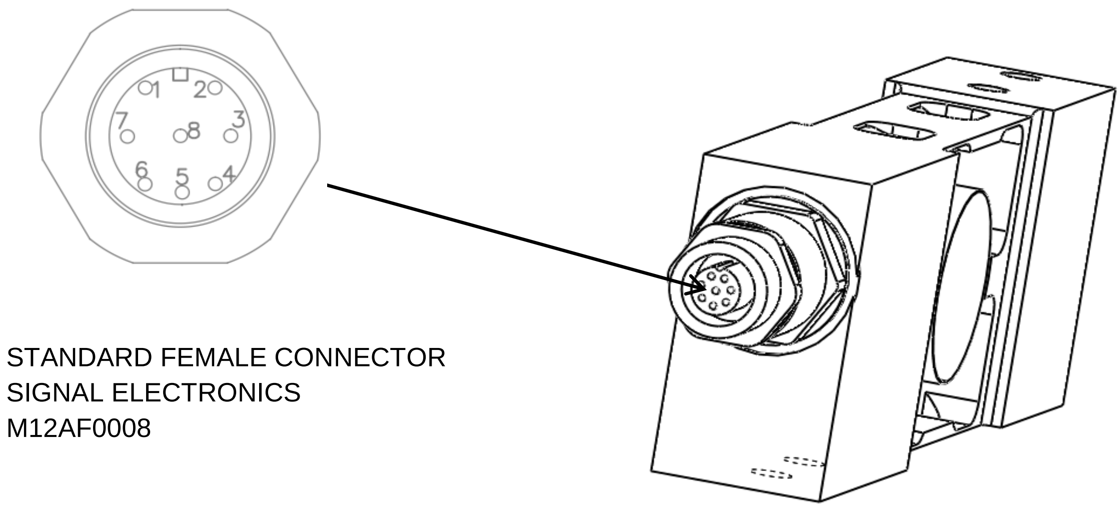
Step 2:- Then mount the load cell to the base plate.

Reversing this sequence can cause damages to the load cell.

6. ELECTRICAL CONNECTIONS

Caution!
Operating voltage for this load cell is 10-30 VDC. Incorrect connections or exceeding the operating voltage can cause irreversible damages.

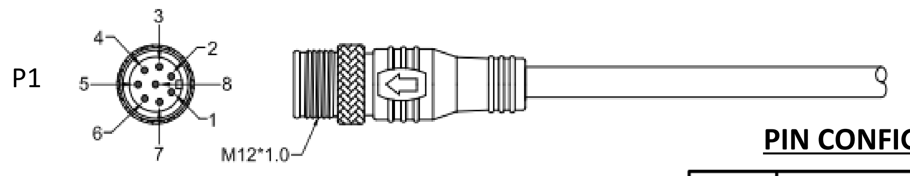
6.1 Wiring diagram



PIN CONFIGURATION

STANDARD FEMALE CONNECTOR SIGNAL ELECTRONICS M12AF0008		
Pin No	Rs-232+CAN	Rs-485+Rs-422
1	GND1	GND1
2	Program	Program
3	CANH	Rx+
4	Trigger input	Trigger Input
5	CANL	Rx-
6	RxD	Tx-
7	TxD	Tx+
8	PWR+	PWR+

6.2 Mating connector pin configuration



M12 A CODING MALE TO OPEN CABLE
SIGNAL ELECTRONICS P/N
SGNTZ1409053-1

PIN CONFIGURATION

P1	COLOR	P2
1	WHITE	OPEN
2	BROWN	
3	GREEN	
4	YELLOW	
5	GRAY	
6	PINK	
7	BLUE	
8	RED	

6.3 Interfaces

SPD load cell offers the following interfaces

RS 232

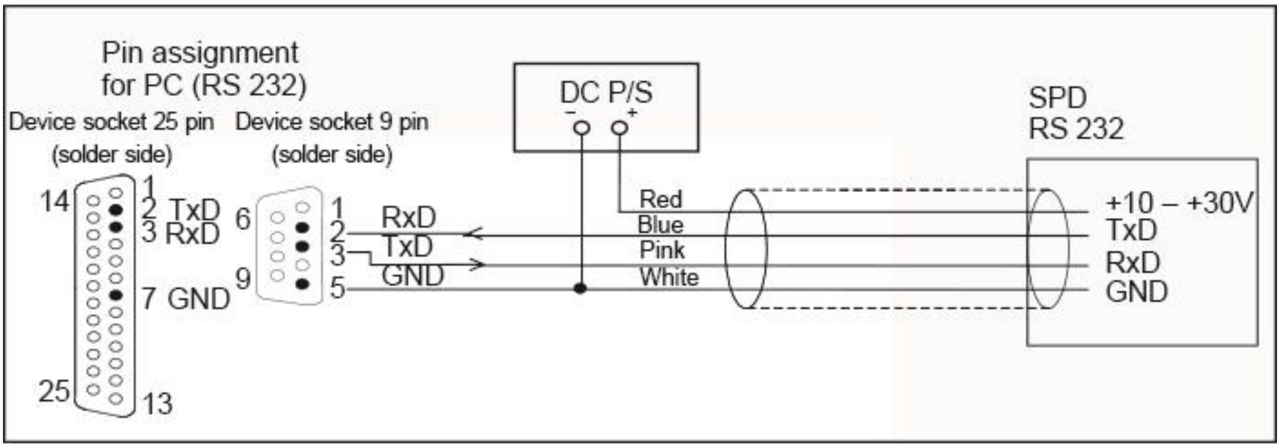
CANopen

RS 485

RS422

The power supply ground and the communication ground are common.

6.3.1 Connecting a SPD load cell to a computer via RS 232 interface



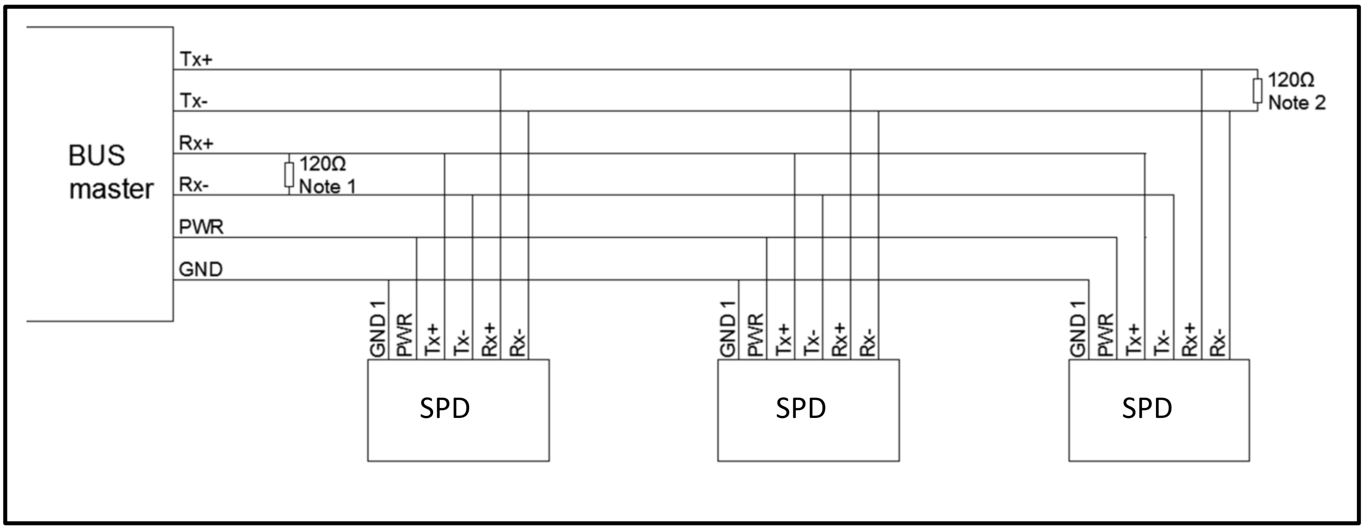
The RS 232 interface is suitable for a point to point connection (one SPD load cell to one interface). Only the signals RxD (Receive Data), TxD (Transmit Data) and GND1 are required. For communication with an external device, the TxD line must be connected to the RxD of the SPD and vice versa.

6.3.2 Connection of SPD load cells to a bus master via RS485

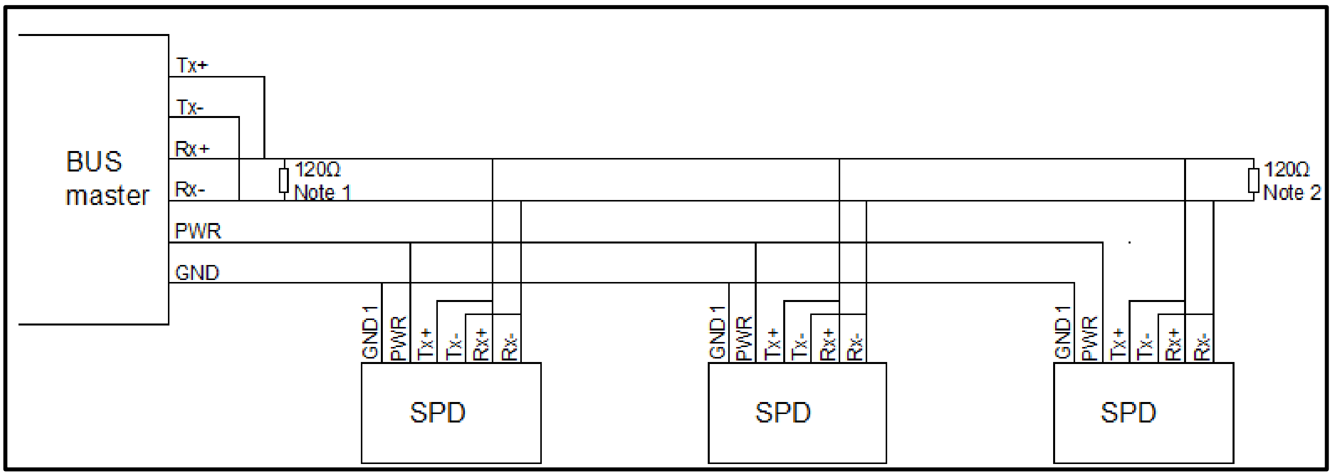
Note 1: Place termination resistor close to the master receiver. In some cases this resistor is already installed inside the master. Please check your documentation for the bus master.

Note 2: Place this resistor close to the last SPD on the bus

6.3.2.1 Four wire RS 485



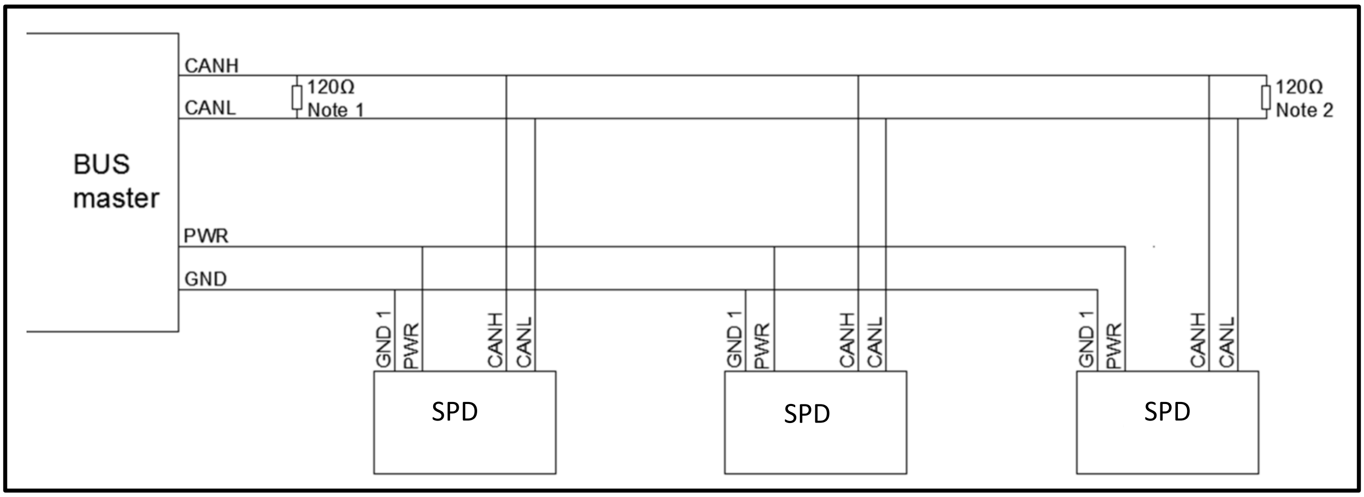
6.3.2.2 Two wire RS485



6.3.3 Connection of SPD load cells to a bus master via CAN bus.

Note 1: Place termination resistor close to the master transceiver. In some cases this resistor is already installed inside the master. Please check your documentation for the bus master.

Note 2: Place this resistor close to the last SPD on the bus or use the integrated termination resistor in the last SPD.



7. SPECIFICATIONS

Model	SPD			
Capacity(E_{max})	kg	10	20	50
Recommended min. External division	g	1	2	5
PERFORMANCES				
Accuracy class according to the OIML R60		C3		
Minimum load cell verification interval(V_{min})	g	E_{max}/10000		
Combined error	%FS	≤±0.017		
Creep error(30min)	%FS	≤±0.017		
Zero balance, raw counts	increments	±2000		
Output resolution at full load, raw counts	increments	256000	512000	512000
Internal AD conversion rate	upd./sec	1200		
Fix, digital Low pass IIR filter, default	Hz	18(Supress 50Hz and 60Hz influence)		
Adjustable , digital Low pass IIR filter Adjustable , digital Low pass FIR filter	Hz	18-0.25; Selectable in 8 steps 40-5; Selectable in 8 steps		
Adjustable, external output update rate	upd./sec	1200-9; Selectable in 8 steps		
GENERAL I/O's				
Hardware interface, CAN version Hardware interface, RS version		CAN and RS232 RS485 and RS422 (Both four wire)		
Data transmission rates CAN Data transm. rates RS485/RS422/RS232	kb	125;250;500;1000 9.6;19.2;38.4;57.6;115.2;230.4;460.8		
Protocol CAN Protocol RS485/RS422/RS232		CANopen ASCII or Modbus RTU		
Logical input, programmable		Trigger level 2-30Vdc,<3mA,Ref to Gnd.		
Power supply	VDC	+10 - +30 ≤ 0.4 Watt		
Connections	Standard 8 pin, female M12AF 0008			
INFLUENCES				
Safe load limit	%* E_{max}	300	150	150
Ultimate load	%* E_{max}	600	300	300
Eccentric loading error acc. to OIML R76	%FS	±0.0233		
Max platform size	mm	450 x 450		
Temperature effect on zero	%FS/ °C	0.001		
Temperature effect on Span	%FS/ °C	0.001		
Temperature range	°C	Operating: -10/+40 Storage: -40/+70		
EMC performance		MID Class E2 (Industrial locations)		
I/O protection, all pins		Reversed polarity; Excess voltage and Surge		
Isolation Body/Electronics at 500VDC	GΩ	≥1		
Vibration		2.5G operational; 5G non-operational		
Environmental protection per IEC 529		Body IP69k; Connectors IP68		
Corrosion resistance		All stainless steel		

8. COMMANDS

For ease of interpretation, the commands have been grouped together and will be described in the following sequence:

8.1 System diagnostics commands
8.2 Calibration commands
8.3 Motion detection commands
8.4 Filter setting commands
8.5 Weigher control commands
8.6 Output commands
8.7 Auto-transmit commands
8.8 Commands for External Input Control
8.9 Communication set-up commands
8.10 Save set-up parameters command
8.11 Trigger Commands
8.12 Re-Trigger Commands
8.13 User defined information

8.1 System diagnostics commands

The following three commands provide a means of interrogating the device to confirm the type of device present, the software version of that device and the status. The commands require no parameters and are used as follows:

ID Determine the device ID code - this is a code, which identifies the type of device, which is currently open for communications. Issuing the ID command, which has no parameters, will return the code **D:1510** This code is useful when mixed devices may be present on the bus.

IV Determines the device software version - this identifies the release of software that is installed in the device. This is useful when determining the availability of special commands or features that may have been requested for special applications. Issuing the IV command, which has no parameters, will return the software identification code in the format **V:0104**

IS Determine the device status - Issuing the ID command, which has no parameters, will return a result in the format **S:000000**. This result comprises two 3-digit decimal values, which can be decoded according to the table below:

Leftmost 3-digit value:		Rightmost 3-digit value:	
1	Signal stable	1	not used
2	Zero action performed	2	not used
4	Tare active	4	not used
8	not used	8	not used
16	not used	16	not used
32	not used	32	not used
64	not used	64	not used
128	not used	128	not used

Therefore, the example result **S:001000** decodes as signal stable (no-motion) no zero action and no tare.

SR Software Reset – this command will respond with ‘OK’ and after maximum 400ms perform a complete reset of the LDB.

8.2 Calibration commands

CE Set the calibration functions to the enabled state. This command must be issued PRIOR to any attempt to set the calibration parameters CM, CI, MR, DS, DP, CZ, CG, ZT, ZR, FD or CS. Issuing the command without any parameters results in the response **E+XXXXX** where **XXXXX** is the Traceable Access Code (TAC). This is an internal code that is used to record any changes in the calibration settings of the device. This is a critical feature that is required for “approved” applications, as it provides for the control of access to any command that has the potential of changing the weigher calibration value.

CM n Set the maximum allowable output value in interval or range n (1 <= n <= 3). Lower limit 1, upper limit 999999. Issuing the command without any parameters returns the current CM value. This value will determine the point at which the output will change to **ooooooo**, signifying over-range or change to the next range or interval. To set a new value for CM, the command must be preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. The new CM value required is then input as a parameter of CM, in the format **CM 1 4010**. Factory default setting: Max 1 = 999999, Max 2 = 0, Max 3 = 0. For further information, please refer to section 9 CALIBRATION PROCEDURE.

CI Set the minimum allowable output value. Lower limit -999999, upper limit 0. Issuing the command without any parameters returns the current CI value. This value will determine the point at which the output will change to **uuuuuuu**, signifying under-range. To set a new value for CI, the command must be preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. The new CI value required is then input as a parameter of CI, in the format **CM -200**. Factory default setting: -9. For further information, please refer to section 9 CALIBRATION PROCEDURE.

MR Select multi range / multi interval.0 = Multi interval, 1 = Multi range. Issuing the command without any parameters returns the current MR value. To set a new value for MR, the command must be preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. The new MR value required is then input as a parameter of MR, in the format **MR 1**. Factory default setting: 0. For further information, please refer to section 9 CALIBRATION PROCEDURE.

DS Set the display step size - this allows the output to step up or down by a unit other than 1. Permitted values are 1, 2, 5, 10, 20, 50, 100, 200 and 500. To set a new value for DS, the command must be preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. The new DS value required is then input as a parameter of DS, in the format **DS 100**. In multi range / multi interval applications DS will define the step size in the lowest range / interval. The higher ranges / intervals will use the next step sizes from the list of allowable step sizes. For further information, please refer to section 9 CALIBRATION PROCEDURE. Factory default setting: 1.

DP Set the decimal point position - this allows the decimal point to be positioned anywhere between leftmost and rightmost digits of the 6-digit output result. Permitted values are 0 for the rightmost position, and 6 for the leftmost position. To set a new value for DP, the command must be preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. The new DP value required is then input as a parameter of DP, in the format **DP 2**. For further information, please refer to section 9 CALIBRATION PROCEDURE. Factory default setting: 3.

CZ Set the calibration zero point - this is the reference point for all weight calculations, and is subject to TAC control. The command returns **ERR** and has no action unless it is preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. Confirmation of action is provided by the return **OK**. For further information, please refer to section 9, CALIBRATION PROCEDURE.

CG Set the calibration gain (span) value - this is the reference point for the calibration under load, and is subject to TAC control. The lower limit for CG is 1, the upper limit is 999999. The weight signal used for calibration should be as close as possible to the maximum allowable display value (CM) so as to ensure optimum calibration accuracy. A feature provided is the ability to recall the value of the calibration weight used for the current calibration by the issue of a CG command without any parameters. This is useful information for future calibration purposes or for diagnostics. When calibrating the span, the actual value of the calibration weight must be entered as a parameter of the CG command, for example if the output 25000 is required for the weight placed on the load cell, then the calibration command becomes **CG 25000**. The command return "**ERR**" and has no updating action unless it is preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. **If the load applied to calibrate the span is less than 1% of full scale (2mV/V), the gain calibration will fail and 'ERR' will be returned.** For further information, please refer to section 9 CALIBRATION PROCEDURE.

ZT Zero tracking - this command set the zero track band in divisions (d). Issuing the command without any parameters returns the current ZT value. The command returns **ERR** and has no updating action unless it is preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC. Zerotracking will be performed only on results less than +/-(0.5 * ZT) at a rate of 0.4 d/sec where d = display step size (see DS command). The zero can only be tracked to +/- ZR (see ZR command). A value of zero turns off the zero tracking. Factory default setting: 0.

FD Factory default settings – this command put the LDB back to a known state. The data will be written to the EEPROM and the TAC will be incremented by 1.
NOTE: All calibration and setup information will be lost, by issuing this command. The command returns **ERR** and has no updating action unless it is preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC.

IZ Correction of System Zero

This command can correct the system zero after a successful calibration, e.g. to correct the unknown weight of a mounting accessory which was used to hold the calibration weight during the calibration procedure. By a simple parallel shift of the gain curve the sensitivity of the scale will stay unaffected.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
IZ↵	**OK**	System zero corrected

ZR Zero range

Sets the zero range manually – this is the range in increments within which the weighing scale can be zeroed. Issuing the ZR command without any parameter will return the current value. Permitted values are between the lower limit of 0 (= factory default setting) and the upper limit of 999999. A value of zero enables the standard zero range of +/-2% of max.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
ZR100↵	**OK**	Setup: Zero range = 100 d

ZI Initial Zero Range

Define the initial zero range (0...999999 d). If ZI is non-zero the device will perform an automatic Set-Zero when the weight stabilizes with the No-motion settings and the weight is within the ZI range. Factory default: 0.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
ZI100↵	**OK**	Setup: Initial Zero range = 100 d

TM Tare mode

This command sets the tare mode. The tare modes are defined in the table below.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
TM1↵	**OK**	Setup: Tare mode = 1

Tare modes:

TM	Allow tare of negative values	Clear preset tare at return to range 1
0 (Default)	**Yes**	**Yes**
1	**No**	**Yes**
2	**Yes**	**No**
3	**No**	**No**

Note: For OIML R76 compatible applications a tare mode of 1 must be used.

TN Set / Clear Non-Volatile Tare

This command sets the tare mode to volatile or non-volatile. Value range is 0 or 1; Factory default is 0 (volatile). If set to 1 (non-volatile), every set/clear tare will write the value directly to the EEPROM.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
TN↵	**T:000**	Actual setting: TN = volatile
TN1	**OK**	Setup: TN = non-volatile

ZN Set / Clear Non-Volatile Zero

This command sets the zero mode to volatile or non-volatile. Value range is 0 or 1; Factory default is 0 (volatile). If set to 1 (non-volatile), every set/clear zero will write the value directly to the EEPROM.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
ZN↵	**Z:000**	Actual setting: ZN = volatile
ZN1	**OK**	Setup: ZN = non-volatile

AZ Absolute zero point calibration (eCal)

The command AZ is used as reference point for all weight calculations and will setup in mV/V.
Permitted values are ± 33000 (= ± 3.3000 mV/V).

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
AZ↵	**Z+000796**	Request: Zero point @ 0.0796 mV/V
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
AZ_00500↵	**OK**	New: Zero point @ 0.0500 mV/V

Factory default: 00 000d @ 0.0000mV/V input signal.

AG Absolute gain calibration (eCal)

The command AG is used as absolute gain (or measuring range) for all weight calculations and will setup in mV/V.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
AG↵	**G+001868,+010000**	Request: gain 10 000d @ 0.1868 mV/V
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
AG_+011200_+005000↵	**OK**	New: gain 5 000d @ 1.12 mV/V

Factory default: 20 000d @ 2.0000mV/V input signal.

CS Save the calibration values

This command results in the calibration values being saved to EEPROM, and causes the TAC to be incremented by 1. The CS command save all of the calibration group values, as set by CM, CI, MR, DS, DP, CZ, CG, ZT and ZR. The command returns **ERR** and has no updating action unless it is preceded by the **CE XXXXX** command, where **XXXXX** is the current TAC.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
CS↵	**OK**	Calibration values saved

8.3 Motion detection commands

The Motion Detection facility provides a means of disabling certain functions whenever a condition of instability, or “motion”, is detected. The “no-motion”, or “stable” condition is achieved whenever the signal is steady for the period of time set by NT, during which it cannot fluctuate by more than NR increments. The stable condition activates the relevant bit of responses to “Info Status” (IS - see section 8.1 System diagnostics commands for further information)
The functions, which are disabled whenever motion is detected, are “Calibrate Zero” (CZ) “Calibrate Gain” (CG) “Set zero” (SZ) and “Set tare” (ST).

NR Set the “no-motion” range - this is the range within which the weighing signal is allowed to fluctuate and still be considered as “stable”. Issuing the NR command without any parameter will return the current value. Set a new value by issuing the NR command followed by the desired value, in the format **NR 5**. Permitted values are between the lower limit of 0 and the upper limit of 65535. Factory default setting: 1.

NT Set the stabilization time for the “in motion” band. This is the time parameter that defines the period during which the output must not fluctuate more than NR increments in order to be considered “stable”. Set a new value by issuing the NT command followed by the desired value in milliseconds, in the format **NT 1000**. Permitted values are between the lower limit of 0 and the upper limit of 65535. Factory default setting: 1000.

8.4 Filter setting commands

The facility exists for the setting of a digital filter via the command parameter FL, and this filter can be adjusted to eliminate most unwanted disturbances. Note that this filter is positioned immediately after the A/D Converter, and will therefore have an effect on all aspects of weigher operation.

FM Filter mode- Permitted values are 0 and 1, see below table.
Issuing the FM command without any parameters will return the current mode value. Set a new value by issuing the FM command by the desired value, in the format **FM 1**.
Factory default setting: 0.

FM value	Filter mode
0	IIR
1	FIR

FL Set the filter - permitted values are between 0 and 8, see table below.
Issuing the FL command without any parameters will return the current filter value. Set a new value by issuing the FL command followed by the desired value, in the format **FL 4**.

FL Value	Cutoff frequency (Hz)	
	IIR	FIR
0	None	None
1	18	19.8
2	8	9.7
3	4	6.5
4	3	4.9
5	2	3.9
6	1	3.2
7	0.5	2.8
8	0.25	2.5

UR Set the update rate - this command defines the number of available updates per second, see table below. Issuing the UR command without any parameters will return the current update rate. Set a new value by issuing the UR command followed by the desired value, in the format **UR 2**.

0	1	2	3	4	5	6	7
1200 u/sec	600 u/sec	300 u/sec	150 u/sec	75 u/sec	37.5 u/sec	18.8 u/sec	9.4 u/sec

8.5 Weigher control commands

The following commands provide the means to control the setting and resetting of the zero and tare points. The availability of net weighing depends on these functions. The zero point which is set at calibration time, remains the "true" zero, but the "current" zero will be the basis for the output result. Remember that the "current" zero can be influenced by the "zero tracking" function, and this should be taken into account when designing the application. A basic system control is the disabling of the "set zero"and "set tare"functions whenever the weighing signal is not stable, as defined by the "no-motion" function. Furthermore, the zero point cannot be reset if it has moved more than ZERORANGE away from the original calibration zero point.

SZ Set the system zero - this command will create a "current" zero point which will become the basis for all weigher operation, until further updated by the zero tracking function, or another SZ command or the "reset zero" command (RZ). As previously stated, any attempt to zero a drift of more than +/- ZERORANGE will result in the SZ command being rejected (**ERR**). The SZ command is also rejected if the weighing signal is fluctuating, as defined by the "no-motion" function parameters (NR and NT). The "signal stable" bit in the responses to the "info status" (IS) command must therefore be active before a SZ command can be accepted. Issuing the SZ command, which has no parameters, will return the **OK** or **ERR** response. If **OK** is returned, then the "zero action performed" bit in the response to the "info status" (IS) command will be activated.

ZA Set Averaged System Zero

This command will set the system zero as SZ, but using an average over the TI time period.

RZ Reset the zero point to the "calibration" zero - this command will return the zero point to that which was stored during the calibration procedure. Issuing the RZ command, which has no parameters, will return the **OK** or **ERR** response. If **OK** is returned, then the "zero action performed" bit in the response to the "info status" (IS) command will be de-activated.

ST Set the tare point - this command will activate the net weighing function, by storing the current weighing signal output value as a tare value. The ST command is rejected if the weighing signal is fluctuating, as defined by the "no-motion" function parameters NR and NT. The "signal stable" bit in the "info status" return must therefore be active before a ST command can be accepted. Issuing the ST command, which has no parameters, will return the **OK** or **ERR** response. If **OK** is returned, then the "tare active" bit in the response to the "info status" (IS) command will be activated.

RT Reset the tare - this command cancels the net weighing mode, and restores the current zero. The weighing signal output returns to the gross mode. Issuing the RT command, which has no parameters, will return the **OK** or **ERR** response. If **OK** is returned, then the "tare active" bit in the response to the "info status" (IS) command will be activated.

SP Set Preset Tare

This command sets a tare value.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
SP↵	**T+000000**	Tare value 0 (factory default
SP1000	**OK**	Setup tare value 1000d

8.6 Output commands

The following command provides the means of obtaining an output results from the device.

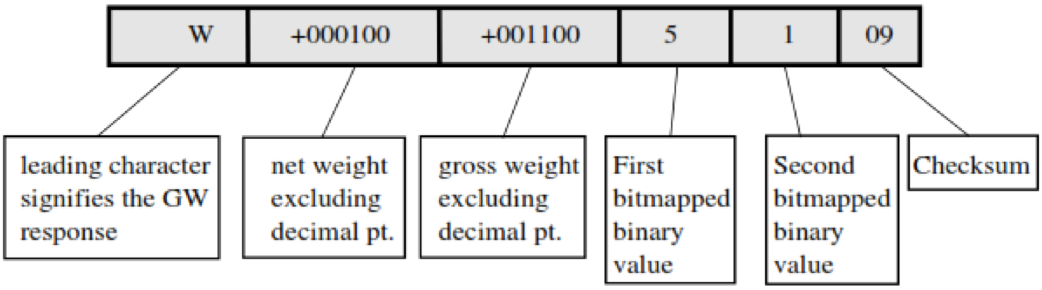
GG Get the gross value - returns the current gross weight value. Issuing the GG command, which has no parameters, will return the gross weight value in the format **G+001.100**.

GN Get the net value - returns the current net weight value. Issuing the GN command, which has no parameters, will return the net weight value in the format **N+001.100**.

GT Get the tare value - returns the current tare weight value. Issuing the GT command, which has no parameters, will return the tare weight value in the format **T+001.100**.

GS Get the A/D sample value – returns the current output result of the A/D converter (ADC). This facility is useful when developing the application, or when calibrating the system, as it allows a check to be made of the operating range of the ADC. Issuing the GS command, which has no parameters, will return the ADC output value in the format **S+100000.**

GW Get the "long" weight values - returns the current net, gross and status values. Issuing the GW command, which has no parameters, will return the net weight, the gross weight, the status and the checksum values, all combined into one single string in the format **W+000100+0011005109**. The first two sections of the return string comprise the net weight and gross weight results, followed by two hexadecimal characters, which represent two bitmapped status indicators. The last two hexadecimal characters represent the checksum, which is the inverse of the sum of all the ASCII values of the string, not including the checksum characters.



The bitmapped characters are:

First Bit value	description	Second Bit value	description
1	not used	1	Signal stable
2	not used	2	Zero action performed
4	not used	4	Tare active
8	not used	8	not used

The checksum is derived as follows:

a. Add the ASCII values of all the 17 characters in the string
b. Convert the decimal result to hexadecimal
c. Remove the most significant digit from the hexadecimal result
d. Invert the remaining hexadecimal value
e. Convert the hexadecimal value to characters

GA Get Triggered Average Value

This command reads the measurement result of a measurement cycle. The measurement value has been averaged according the defined measuring time. The trigger commands can be found in chapter 8.11 and 8.12.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
GA↵	**A+001.100**	Request: GA = 1100 g

Note: For preventing errors during the read out of the data, the register GA has stored the value 999999 at the beginning of the measurement cycle. The measurement result can only be read after the defined measuring time MT has been elapsed and before a new measurement cycle has been started.

GL Get Data String “Average, Gross and Status”

Master (PC / SPS) sends	Slave (SPD) resp.	Meaning
GL↵	**L+000100+0011005109 (example)**	Average value: +000100 d (no decimal point) Gross value: +001100 d (no decimal point) Status bit 1: 5 (not used) Status bit 2: 1 (Hex) Check sum: 09 (Hex)

For check sum, status bit 1 and status bit 2, see command GW.

OF Output Format for Data String GW and GL

This command puts the range information and/or the decimal point into the “long” data strings of the GW and GL output response.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
OF1↵	**OK**	Setup: OF = 1

Output Format		
Parameter setting	Range Information	Decimal Point in GW/GL response
0 (= factory default)	No	No
1	Yes	No
2	No	Yes
3	Yes	Yes

E.g. when the range information is selected, the data strings will change from G+000000 to Gn+000000,
where $1 \leq n \leq 3$.

8.7 Auto-transmit commands

The following command provide the means to output the weight results in a continuous stream, which starts upon the issue of the relevant command, and ends upon the issue of any other command.

SG Auto-transmit the gross weight value - continually returns the current gross weight value. Issuing the SG command, which has no parameters, will continually return the gross weight value in the format **G+001.100**, until interrupted by any other command.

SN Auto-transmit the net weight value - continually returns the current net weight value. Issuing the SN command, which has no parameters, will continually return the net weight value in the format **N+001.100**, until interrupted by any other command.

SX Send ADC Sample Value continuously

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
SX↵	**S+125785**	ADC sample value = 125785 d

SA Send Triggered Average Value automatically

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
SA↵	**OK**	Auto-Transmit: triggered average value

This command will start to auto-transmit the measurement value of the current trigger cycle. The trigger setup commands are described in the chapters 8.11 and 8.12.

SL Send Data String “Average, Gross and Status“ automatically

Master (PC / SPS) sends	Slave (SPD) resp.	Meaning
SL↵	**L+000100+0011005109 (example)**	Average value: +000100 d (no decimal point) Gross value: +001100 d (no decimal point) Status bit 1: 5 (not used) Status bit 2: 1 (Hex) Check sum: 09 (Hex)

For check sum, status bit 1 and status bit 2 see command SW.

SW Auto-transmit the long weight value - continually returns the current net weight, the gross and status values. Issuing the SW command, which has no parameters, will continually return the net weight, the gross weight, the status and the checksum values, all combined into one single string in the format **W+000100+0011005109**, until interrupted by any other command. The decode of the string is exactly as per the "long" weight command GW, listed in section 8.6 Output commands. Note that decimal point information is not transmitted.

8.8 Commands for External Input Control – IN

IN Read status of the logic inputs

This command reads the status of the logic inputs.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
IN↵	**IN:0000**	Input 0 and 1 inactive
IN↵	**IN:0001**	Input 0 active
IN↵	**IN:0010**	Input 1 active

The status response is in the form of a four digit code where 0 = false and 1 = true (inputs are active 'high'). The least significant bit corresponds to Input 0.

8.9 Communication set-up commands

AD Set the address of the LDB for networking (0-255). Setting the device address to 0 will set the continuously active mode, where the device becomes permanently active, and will listen and respond to any command on the bus, without the need for an OP xxx command. Issuing the AD command without any parameters will return the current address. Factory default setting: 0.

NOTE: this setting will take effect after power on reset (remember to store the setting using the **WP** command before turning off the power)

NA Network Address - CAN

This command displays or sets a network address for the CAN interface. The permitted range is from1 to 127.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
NA↵	**A:001**	Show CAN interface address
NA_15↵	**OK**	Set CAN interface address to15

Factory default: 1

NS Network Settings – For Serial Channel and CAN Interface

The command **NS** <Interface> <Param> [New Value] can display or set various communication parameters in the device.

The parameter "Interface" addresses the physical interface on the device and the parameter "Param" addresses the available parameters for this interface. All LDx device have a serial channel (UART) and some of the LDx devices also have a CAN interface.
Serial channel ("Interface" = 0)

The following parameters are defined for the serial channel:

"Param"	Parameter name	Allowed values
0, *Note 1*	Device ID (Read only)	N/A
1, *Note 2*	Baud rate	9600, 19200, 38400, 57600, 115200, 230400 and 460800 bit/sec.
2, *Note 3*	Loop address	0 to 255
3	Serial mode	See table below
4, *Note 4*	Tx Delay	0 to 255

The Serial mode is a bit mapped 16 bit value.
- bit 0 controls parity: 1 = enabled. *Note 5.*
- bit 1 controls parity type: 0 = odd, 1 = even. *Note 5.*
- bit 7 controls the duplex mode: 0 = Full duplex, 1 = Half duplex.
- bit 15..8 specify the protocol in use: 0 = ASCII, 1 = Modbus RTU.

Allowed bit combinations:

Bit 15..8	Bit 7	Bit 1	Bit 0	= Hexadecimal	= Decimal
0	0	0	0	0x0000	0
0	1	0	0	0x0080	128
1	0	0	0	0x0100	256
1	0	0	1	0x0101	257
1	0	1	0	0x0102	258
1	0	1	1	0x0103	259
1	1	0	0	0x0180	384
1	1	0	1	0x0181	385
1	1	1	0	0x0182	386
1	1	1	1	0x0183	387

CAN Interface ("Interface" = 1)

The following parameters are defined for the CAN interface:

"Param"	Parameter name	Allowed values
0, *Note 1*	Device ID (Read only)	N/A
1	CANopen address	1 to 127
2	Bit rate	10, 20, 50, 125, 250, 500, 800 and 1000 kbit/sec.

Notes for the interfaces 0 (serial) and 1 (CAN)
Note 1: Identical to the ID command.
Note 2: Identical to the BR command.
Note 3: Identical to the AD command.
Note 4: Identical to the TD command.
Note 5: Parity check/generation is only available for Modbus RTU.

Examples:

Master (PC / PLC) sends	Slave (LDx) responds	Meaning
NS_0_0↵	**D:1510**	The device type is LDB 151
NS_0_1↵	**B 115200**	The serial channel baud rate is 115200
NS_0_2↵	**A:000**	The serial channel address is 0
NS_0_3↵	**S 00000**	The serial channel mode is ASCII protocol and full duplex.
NS_0_1_230400↵	**OK**	Set the serial channel baud rate to 230400

BR Set the LDB baud rate. Issuing the **BR** command without any parameters will return the current baud rate. Set a new value by issuing the **BR** command followed by the desired value, in the format **BR 115200**. Factory default setting: 115200.

NOTE: this setting will take effect after power on reset (remember to store the setting using the **WP** command before turning off the power)

DX Half or full duplex – this command select half or full duplex communication. Parameter = 0 select half duplex communication and parameter = 1 select full duplex communication - use half duplex setting when using two wire RS485.

8.10 Save set-up parameters command

The calibration and setup parameters can be divided in 2 groups:

- **Calibration**: CM, DS, DP, CZ, CG, ZT, IZ and FD, etc. saved by command **CS**
- **Setup**: FL, FM, NR, NT, BR, AD, DX and others, saved by command **WP**

Note: Calibration data can only be saved if the TAC code is known and precedes the CS command. The setup data and the setpoint data will be stored non-volatile in the EEPROM using the **WP** respective **SS** command.

CS Save the Calibration Data

This command results in the calibration data being saved to the EEPROM and causes the TAC to be incremented by 1.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
CE↵	**E+00017 (example)**	Request: TAC counter CE17
CE17↵	**OK**	Calibration sequence active
CS↵	**OK**	Calibration values saved

The CS command saves all of the calibration group values, as set by CZ, CG, CM'n', DS, DP and ZT. The command returns ERR and has no updating action unless it is preceded by the CE_XXXXX.

WP Save the Setup Parameters

With this command the settings of the "Filter" (FL, FM), the "No-motion" (NR, NT) and the communication (AD, BR, DX) will saved in the EEPROM.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
WP↵	**OK**	Setup data saved
WP↵	**ERR**	Error

GI Get an Image File from the EEPROM

Retrieves a HEX-INTEL formatted EEPROM image file from the EEPROM of the source SPD. The image file contains all stored information except the calibration data. This image file can be downloaded to any SPD with the same firmware type and revision No. as the source SPD.

PI Download an Image File to the EEPROM

Downloads a HEX-INTEL formatted EEPROM image file to the target SPD EEPROM. The image file contains all stored information except the calibration data.
Attention: The target SPD must have same firmware type and revision no. as the source SPD.

8.11 Trigger Commands – SD, MT, GA, TE, TR, TL, SA

Note: All changes to the trigger commands have to be stored in the EEPROM using the WP command.

SD Start Delay Time

This command defines a time delay between the trigger and the start of the measurement.
Setting range: 0 ms to 65535 ms.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
SD↵	**S+00100**	Request: SD = 100 ms
SD200↵	**OK**	Setup: SD = 200 ms

Default setting: SD = 0 ms; time plot of a typical checkweigher cycle see below

MT Measuring Time

This command defines the measuring time for the averaged measurement result.
Setting range: 0 ms to 3000 ms.

Master (PC / SPS) sends	Slave (179.1) responds	Meaning
MT↵	**M+00100**	Request: MT = 100 ms
MT500↵	**OK**	Setup: MT = 500 ms

Note: The setting MT = 0 disables the trigger function and the averaging.
Default setting: MT = 0 [= trigger function disabled]; time plot of a typical checkweigher cycle see below

GA Get Triggered Average Value

This command reads the measurement result of a measurement cycle. The measurement value has been averaged according the defined measuring time.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
GA↵	**A+001.100**	Request: GA = 1100 g

Note: For preventing errors during the read out of the data the register GA has stored the value 999999 at the beginning of the measurement cycle. The measurement result can only be read after the defined measuring time MT has been elapsed and before a new measurement cycle has been started.

TE Trigger Edge

This command defines the trigger edge. Allowed settings are “0” for falling edge and “1” for rising edge. This command can only be used in conjunction with a hardware trigger on the digital input channel 0.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TE↵	**E:001**	Request: TE = 1 (rising edge)
TE0↵	**OK**	Setup: TE = 0 (falling edge)

Default setting: TE = 0 [= falling edge]; time plot of a typical checkweigher cycle see next page.

TR Software Trigger

This command starts a measurement cycle. Its execution can be compared to a hardware trigger on the digital input channel 0.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TR↵	**OK**	Trigger event

TL Trigger Level

This command defines a level for a rising edge trigger on the measurement signal. Setting range: 0 to 999999.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TL↵	**T+999999**	Request: TL = 999999
TL1000↵	**OK**	Setup: TL = 1000

In the example a new measurement cycle would automatically start, if the signal exceeds 1000 d (e.g. 100,0 g; trigger commands SD and TL).
Default setting: TL = 999999 [= trigger level disabled]

Note: All trigger possibilities are always available in parallel. If a software trigger (command TR) or a hardware trigger (Digital input 0) will be used the trigger level should be set to its maximum value (TL = 999999). This setting disables the trigger level.

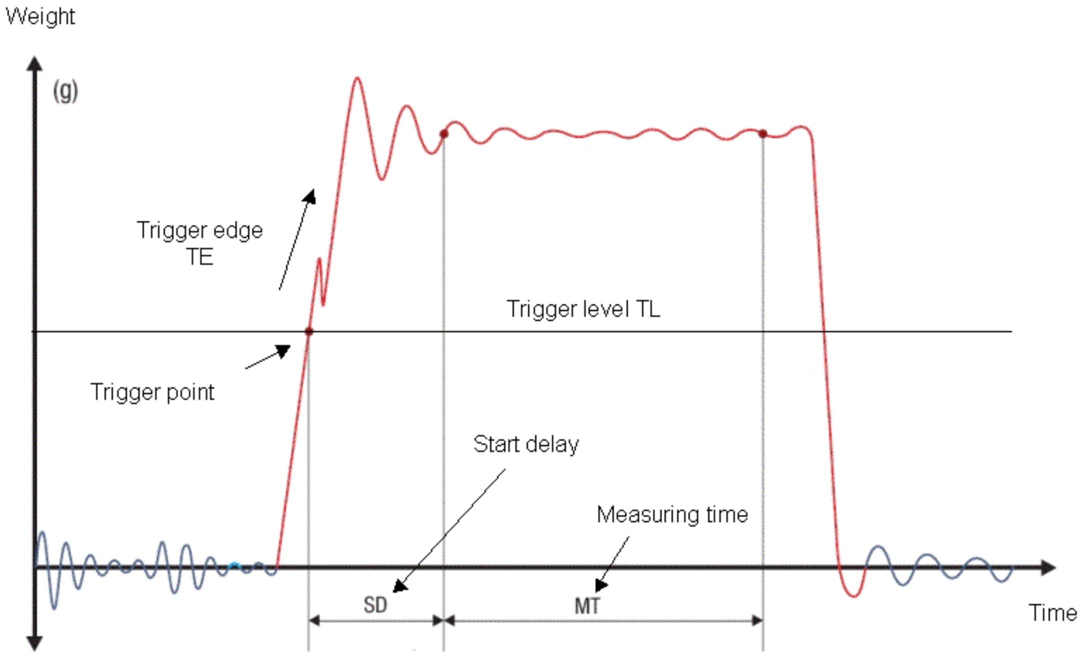


Figure: Time plot of a typical checkweigher cycle

SA Send Triggered Average Value automatically

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
SA↵	**OK**	Auto-Transmit: triggered average value

This command will start to auto-transmit the measurement value of the current trigger cycle.

8.12 Re-Trigger Commands – RW, TT, TS, DT, TW and TI

Note: All changes to the re-trigger commands have to be stored in the EEPROM using the WP command.

RW Trigger Window for Re-Trigger Function

This command defines a trigger window in unit d (digits) around the current cycle average value. If the signal leaves this window even for one sample, then the averaging over the time period TT will be started again.
For using the automatic re-trigger function, it is required to define a short-time averaging period (command DT, see below) before you can use this function.
Default value: RW = 65535 d.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
RW↵	**R+65535**	Request: RW = 65535 d
RW500↵	**OK**	Setup: RW = 500 d

TT Averaging Time for Re-trigger Function

This command defines an averaging time for calculating the cycle average value. If this time period has been elapsed, the measurement cycle will be finished at the latest.
The setting TT = 0 disables the re-trigger function. Default setting: TT = 65535 ms.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TT↵	**T+65535**	Request: TT = 65535 ms
TT300↵	**OK**	Setup: TT = 300 ms

TS Stop Value for Re-trigger Function

This command defines a stop criteria in unit d (digits) for the re-trigger function. If the signal falls more than this value TS below the cyclic average value, then the measurement cycle will be finished.
Default setting: TS = 0 d.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TS↵	**T+65535**	Request: TS = 65535 d
TS480↵	**OK**	Setup: TS = 480 d

DT Short-time Averaging Period

This command defines a time period to calculate short-time averages. If the short-time average falls outside the trigger window, then the measurement will be started again.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
DT↵	**T+00050**	Request: DT = 50 ms
DT25↵	**OK**	Setup: DT = 25 ms

TW Window for Automatic Taring

This command defines an amplitude window for the automatic taring. The setting TW = 100 means, that the system calculates a new tare value, if the averaged net value of the empty scale falls within 100 digits of the net zero point. The new tare value will be averaged over the time period TI (see below). If the averaged tare value falls outside this window, then the tare value will not be updated.
Default setting: TW = 0 [= automatic taring disabled]

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TW↵	**T+00000**	Request: TW = 0 d
TW100↵	**OK**	Setup: TW = 100 d

TI Averaging Time for Automatic Taring

This command defines the averaging time for the automatic taring. Within this time period the system calculates an averaged tare value. Default setting: TI = 0 ms.

Master (PC / SPS) sends	Slave (SPD) responds	Meaning
TI↵	**T+00000**	Request: TI = 0 ms
TI200↵	**OK**	Setup: TI = 200 ms

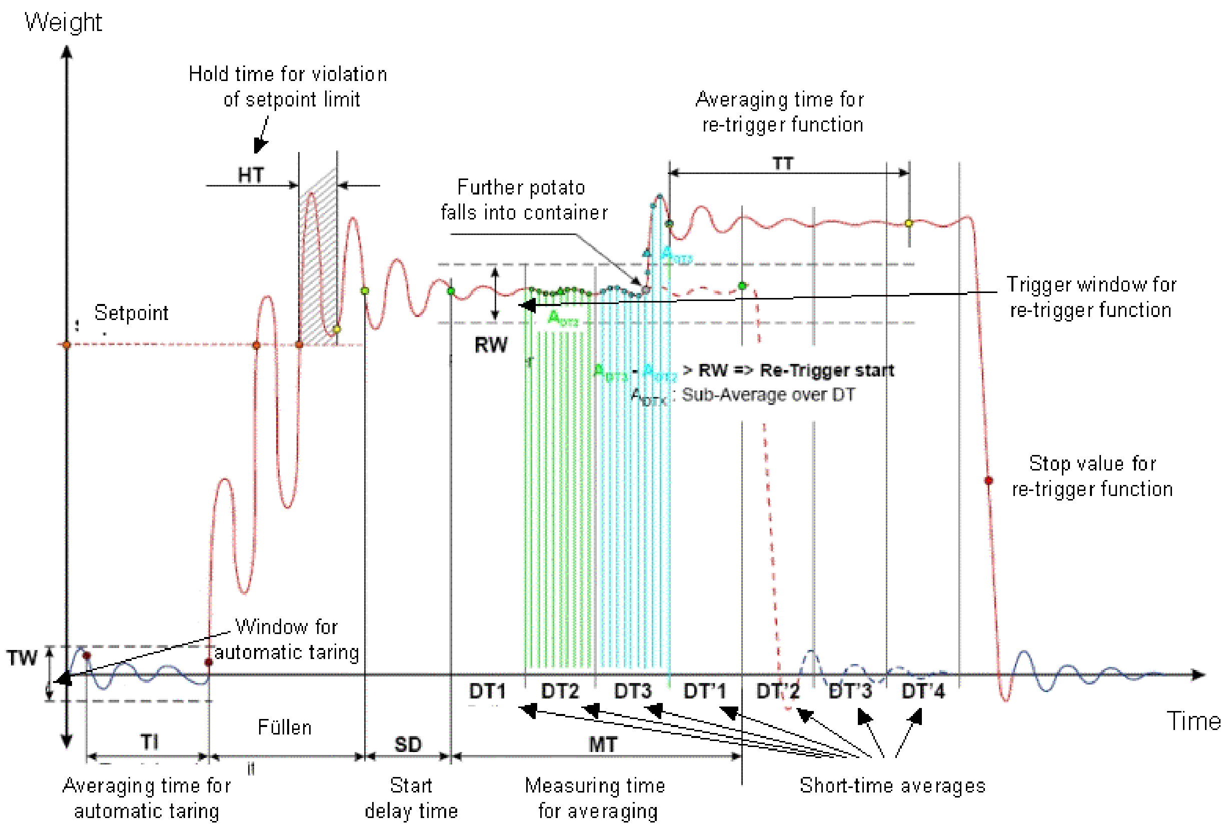


Figure: Time plot of a measurement cycle with the re-trigger function

8.13 User defined information

This command enables the user to set a serial number and a text information for identification of the system.

RS Read serial number. Issuing the **RS** command without any parameters will return the current serial number in the format **S:12345678**. Set a new value by issuing the **RS** command followed by a password and then the desired number (Up to 8 digits) in the format **RS 330130 12345678**. Factory default setting: undefined. The serial number will automatically be stored in the internal EEPROM.

9. CALIBRATION PROCEDURE

The calibration interface features a "**TRACEABLE ACCESS CODE**" (**TAC**), as is required for use in "Approved" applications. This feature also ensures that access to the calibration functions is protected from inadvertent or unauthorized change. The following parameters are considered as CALIBRATION commands:

CE: Calibration enable - returns the current TAC value.
CZ: Calibrate zero - sets the system zero point.
CG: Calibrate gain - sets the system gain.
CM: Calibrate maximum - sets the maximum allowable display value in each range.
CI: Calibrate minimum - sets the minimum allowable display.
DS: Display step size - sets the output incremental step size.
DP: Display decimal point - sets the position of the output decimal point.
ZT: Zero track band.
ZR: Zero track range.
FD: Factory default setting (return to)
CS: Calibration save.

To make an enquiry as to the setting of any of these commands requires the issue of the command with no parameters attached. The responses are explained in section 8.2 Calibration commands.

To make a change to the settings of any of these commands **REQUIRES THE ISSUE OF THE CE COMMAND FOLLOWED BY THE CURRENT TAC VALUE (CE XXXXX).**
For example, if the output step value needs to be changed from 1 to 5, the following steps would be required:

Master send: **CE**
Slave returns: **E+00016**
Master send: **CE 16**
Slave send: **OK**
Master send : **DS 5**
Slave send: **OK**

The output will now increment in steps of 5 divisions. It will then be necessary to SAVE the calibration parameters to non-volatile memory, by issuing the CS command. The CS command, which has no parameters and must be preceded by the **CE XXXXX** command, will return **OK** to signify successful update. The TAC is then incremented by 1.

An example of the recommended calibration procedure follows:

9.1 To set the system zero and the system gain

With the device selected a suitable load cell in place, with known test weights available, (the example uses 5000 as a test weight value):

Step 1	Master sends: **CE** Slave returns: **E+00017**	Query the current TAC value
Step 2	Master sends: **CE 17** Slave returns: **OK**	Enable the calibration sequence
Step 3	Master sends: **CZ** Slave returns: **OK**	Ensure that the weigher is unloaded
Step 4	Master sends: **CE 17** Slave returns: **OK**	Enable the calibration sequence
Step 5	Master sends: **CG 5000** Slave returns: **OK**	Where 5000 is the weight value added
Step 6	Master sends: **GG** Slave returns: **G+005000.**	Confirm the calibration is correct
Step 7	Master sends: **CE 17** Slave returns: **OK**	Enable the calibration sequence
Step 8	Master sends: **CS** Slave returns: **OK**	Write the calibration data to memory

The system zero and system gain value will have been updated and written to EEPROM, and the TAC will have been incremented.

10. CANopen INTERFACE

10.1 General

The CAN interface follows the CAN2.0B recommendations. It receives both - 11 bit identifiers, and tolerates 29 bit identifiers. It only transmits 11 bit identifiers.

The **CAN rate** is setup as default to **500 kbit/s**.

The LDB is always quiet on the CAN bus until the NMT Start command is received, except for the very first ‘node guard’ message.

When started by the NMT Start the LDB starts transmitting TPDO1 messages with weight and status.

The default is the net value. When filling is in progress the gateway transmits a TPDO2 every time a module changes state to ‘wait for trigger’. This TDPO2 contains the module number, the module status and the dosed weight. In checkweigher applications the TPDO2 is used to send triggered measurements.

With RDPO1 frames you can send simple commands without an acknowledgement. The functions are: select gross or net value in TPDO1, set or clear system zero, set or clear tare.

With RPDO2 frames you can send triggers or stop triggers. For the filling application the trigger can be used to start the filling cycle. On checkweigher applications the trigger can start measurements and a stop-trigger will stop further internal re-triggers.

In case of an overrun, error or failure an EMERGENCY message is sent to the CAN controller indicating the nature of the error or failure.

RPDO3 and RPDO4 are ignored by the LDB.

SDOs are handled according to profile and CANopen recommendation.

The LDB supports both ‘node guarding’ and ‘heart beat’.

Group Four Transducers
22 Deer Park Drive,
E. Longmeadow, MA 01028
www.groupfourtransducers.com

Phone : (800) 419 1444
Fax : (413) 525 -6182
sales@group-4.com

10.2 PDOs

The Weight and status is sent using TPDO1. One TPDO1 is sent each time a new measurement is ready. The high measuring rate of the LDB will result in approx. 1200 TPDO1's per second. If the system can't handle so many messages the update rate can be reduced – see the UR command.

The TPDO2 is sent when an average measurement is ready. The TPDO2 has the same format as TPDO1.

The TPDO3 is sent when the tare changes. It has the same format as TPDO1.

The format of the TPDO1, TPDO2 and TPDO3 is:

32 bit	16 Bit	8 bits	8 Bit
Weight	Status	0	0

The first field is a single precision float value carrying weight information, gross or net value if it is a TDPO1, average weight if it is a TPDO2 and tare value if it is a TPDO3.

Then status follows as a 16 bit field with the following values defined:

$0001 - Under range,
$0002 - Over range,

$0008 - Center zero,
$0010 - No motion,
$0020 - Tare set,

$0080 – ADC Error,
$0100 - Set-point 0 (source>limit),
$0200 - Set-point 1,

TPDO1

- Weight values are available at all times
- The following table shows the information of TPDO1:

32 bit	16 bit	8 bit	8 bit
Weight	Module Status	0	0

- Default: Net weight.
- Refresh time: Controlled by the setup of command UR.
- Format: Floating point single precision (IEEE 754)

TPDO2
Average weight GA is available and refreshes when a new measurement is ready.

TPDO3
Tare weight GT is available and refreshes when a new tare value is set.

RPDO1
The following commands can be executed direct:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
SG	SN			ST	RT	SZ	RZ
128	64			08	04	02	01

Examples:

- Setting tare: Transmit RPDO1 [08]
- Setting gross weight in TPDO1: Transmit RPDO1 [128]

RPDO2

The following commands can be executed direct:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
TR							
128							

Example:
- Setting trigger start: Transmit RPDO2 [128]

10.3 The SDOs
The CANopen SDOs is a confirmed service, and overrun does not occur if the CAN controller communicates with the LDB in the PRE-OPERATIONAL state. When a SDO has been received by controller no further communication takes place until the service has been acknowledged (or a timeout occurs).

SDO's

Are only available on request
See tables 6.5 Object Directory
Can be used for complete setup of the LDB via CAN bus master, e.g:
- Filter setting: Index 2100, Subindex 4
- Filter Mode setting: Index 2100, Subindex 9

Can be used to get information regarding all the commands available, e.g:
- Net weight: Index 2900, Subindex 2
- AD sample: Index 2900, Subindex 7

Group Four Transducers
22 Deer Park Drive,
E. Longmeadow, MA 01028
www.groupfourtransducers.com
Phone : (800) 419 1444
Fax : (413) 525 -6182
sales@group-4.com

10.4 Communication Profile

The parameters, which are critical for communication, are determined in the communication profile.
This includes the data for manufacturer's product nomenclature, for identification, or the parameters for object
mapping.

Abbreviations used in Tables:

ro	read only
rw	read / write
wo	write only (read will not be regarded as an error, but returns undefined results)
UI8	Unsigned 8
UI16	Unsigned 16
UI32	Unsigned 32
I32	Signed 32
REAL32	32 bit IEEE754 floating point
VS	Visible String

10.5 Object Directory

The object directory of the CAN communication system is described under Appendix B
below.
These entries are in the documentation for the sake of mapping information. These functions must be used
through Process Data Objects (PDO).

11. Modbus Interface

1.1. Implemented functions for Modbus RTU.

- 0x03 Read holding registers: Used for reading 16 or 32bit values.
- 0x04 Read input registers: Same as above.
- 0x06 Write single register: Used for writing 16bit values.
- 0x10 Write multiple registers: Used for writing 32bit value

1.2. Entering Modbus RTU mode from ASCII mode.

- The desired Modbus baudrate must be set using the BR command, the NS0_1 command or left as it is. (Please note that ´_´ mean ´space´ or blank)
- The desired Modbus address (1 to 247) must be set using the AD command or the NS0_2 command.
- The serial channel mode must be changed with the NS0_3 command. The command NS0_3_259 will enable the Modbus mode, set the parity check/generation to even and select full duplex mode.
- These settings must be saved with the WP command.
- To enter Modbus mode the device must now be restarted with the SR command or a power off/on cycle. Please note that the ASCII communication will be lost after this step.

1.3. Entering ASCII mode from Modbus mode – method 1

- The desired ASCII baudrate must be set using the Modbus equivalent to the NS0_1 command or left as it is. Set index 0x2072 to 0, set index 0x2073 to 1 and set index 0x207A to the desired baud rate (9600, 19200, 38400, 57600, 115200, 230400 or 460800)
- The serial channel mode must be changed with the Modbus equivalent to the NS0_3 command. Set index 0x2072 to 0, set index 0x2073 to 3 and set index 0x207A to 0.
- These settings must be saved with the Modbus equivalent to the WP command. Write 0x0004 to index 0x2066.
- To enter ASCII mode the device must now be restarted with a power off/on cycle. Please note that the Modbus communication will be lost after this step.
- Please note that the SPD will start up in ASCII mode using the same address as it used in Modbus mode.

1.4. Entering ASCII mode from Modbus mode – method 2

ASCII mode can be entered by holding the receive data line(s) active during power up. This must be done in different ways depending on the actual device hardware.

RS232 Devices

Connect the RxD to a voltage higher than +3V (but less than +15V) with respect to Gnd, turn on the power to the device and then switch the RxD signal back to its normal position.

RS422/485 Devices

If the Rx+ and Rx- lines are connected to the Tx+ and Tx- lines of an enabled transmitter (not in tristate mode) then the lines can be exchanged (Rx+ to Tx- and Rx- to Rx+) before turning the power on to the device. After turning on the power the communication lines must be switched back to their normal position.

When powered up this way the device will enter ASCII mode with its loop address set to 0 (zero) and its baud rate set to 115200. To keep the device in this mode these settings must be saved with a WP command before resetting or removing the power from the device.

1.5. Modbus Index Tables

See the Appendix D.

12. SPD PART NUMBERS

SPD PART NUMBERS						
OPTION	GEN			A		
DESCRIPTION	LDB151 (RS232+CAN)			LDB152 (RS485+RS422)		
CAPACITY	10	20	50	10	20	50
G4 P/N	4046D-001-00	4046D-000-00	4046D-002-00	4046D-001-01	4046D-000-01	4046D-002-01

Group Four Transducers
22 Deer Park Drive,
E. Longmeadow, MA 01028
www.groupfourtransducers.com

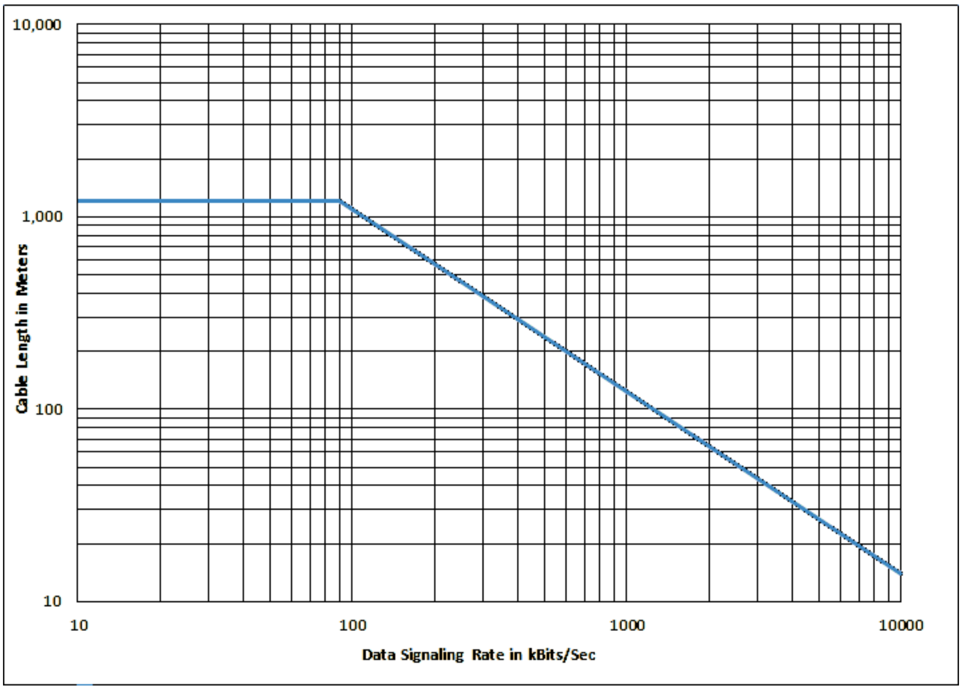
Phone : (800) 419 1444
Fax : (413) 525 -6182
sales@group-4.com

13. APPENDIX

13.1 Appendix A

Maximum cable length allow for different communication protocols.

RS422/485



CANOpen

Bus Speed	Bus Length (L)	Cable Stub Length (l)	Node Distance (d)
1 Mbit/Sec	40 meters (131 feet)	0.3 meters (1 foot)	40 meters (131.2 feet)
500 kbits/Sec	100 meters (328 feet)	0.3 meters (1 foot)	100 meters (328 feet)
100 kbits/Sec	500 meters (1640 feet)	0.3 meters (1 foot)	500 meters (1640 feet)
50 kbits/Sec	1000 meters (3280 feet)	0.3 meters (1 foot)	1000 meters (3280 feet)

RS232

Baud rate	Maximum cable length (ft)
19200	50
9600	500

13.2 Appendix B

Communication Profile (Tables)

Index	Sub-index	Name	Type	Attri-bute	Default-value	Meaning
1000	0	Device Type	UI32	ro	0	Non standard device profile
1001	0	Error Register	UI8	ro	0	Bit 0: Generic error Bit 4: Communication error Bit 7: Manufacturer specific error
1002	0	Manufacturer Status Register	UI32	ro	0	Not used
1005	0	COB-ID Sync message	UI32	rw	80H	COB-ID of the SYNC object
1006	0	Communication cycle Period	UI32	rw	0	Not used
1007	0	Synchronous Window Length	UI32	rw	0	Not used
100C	0	Guard Time	UI16	rw	320	Cycle time in ms, set by the NMT Master or the configuration tool. Index 100Ch and 100Dh are used if index 1017h is zero.
100D	0	Life Time Factor	UI8	rw	3	Life time is set by the NMT Master or the configuration tool.
1014	0	COB-ID Emergency Message	UI32	ro	80H + Node ID	COB-ID of the Emergency Object
1017	0	Heartbeat Time	UI16	rw	0	Producer Heartbeat time in ms. If index 1017h is non-zero the Heartbeat protocol is used, otherwise the Node-guard protocol is used.
1018	0 1 2 3 4	Identity Object Vendor ID Product Code Revision Number Serial Number	UI8 UI32 UI32 UI32 UI32	ro ro ro ro ro	4 269H - - -	Number of entries Vendor ID Product Code Revision Number Serial Number
1400	0 1 2	Number of elements COB-ID Transmission type	UI8 UI32 UI8	ro ro ro	2 200H + Node ID FFH	Communication parameters of 1st Receive PDO Determined using the CANopen minimum system ID assignment procedure. Asynchronous communication.
1401	0 1 2	Number of elements COB-ID Transmission type	UI8 UI32 UI8	ro ro ro	2 300H + NodeID FFH	Communication parameters of 2nd Receive PDO Determined using the CANopen minimum system ID assignment procedure. Asynchronous communication.
1402	0 1 2	Number of elements COB-ID Transmission type	UI8 UI32 UI8	ro ro ro	2 80000400H + NodeID FFH	Communication parameters of 3rd Receive PDO Determined using the CANopen minimum system ID assignment procedure. Asynchronous communication.
1403	0 1 2	Number of elements COB-ID Transmission type	UI8 UI32 UI8	ro ro ro	2 80000500H + NodeID FFH	Communication parameters of 4th Receive PDO Determined using the CANopen minimum system ID assignment procedure. Asynchronous communication.
1600	0 1	Entries in Rx PDO 1 1st Object Cmd. Byte	UI8 UI32	ro ro	2 20060308H	Mapping parameters of the 1st Receive-PDO Object is a bitwise command: Cmd: Bit7: SnG, Bit6: SnN, Bit5: , Bit4: , Bit3: ST, Bit2: RT, Bit1: SZ, Bit0: RZ
1601	0 1	Entries in Rx PDO 2 1st Object Cmd. Byte	UI8 UI32	ro ro	2 20060408H	Mapping parameters of the 2nd Receive-PDO Object is a bitwise command: Cmd: Bit7: TR, Bit6: , Bit5: , Bit4: , Bit3: , Bit2: , Bit1: , Bit0: TS
1602	0	Number of mapped Entries in Rx PDO 3	UI8	ro	0	Mapping parameters of the 3rd Receive- PDO (disabled)
1603	0	Number of mapped Entries in Rx PDO 4	UI8	ro	0	Mapping parameters of the 4th Receive-PDO (disabled)
1800	0 1 2	Number of elements COB-ID Transmission type	UI8 UI32 UI8	ro ro ro	2 180H + Node ID FFH	Communication parameters of 1st Transmit PDO Determined using the CANopen minimum system ID assignment procedure. Asynchronous communication.

Index	Sub-Index	Name	Type	Attri-but	Default Value	Meaning
1801	0	Number of elements	UI8	ro	2	Communication parameters of 2nd Transmit PDO
	1	COB-ID	UI32	ro	280H + Node ID	Determined using the CANopen minimum system ID assignment procedure.
	2	Transmission type	UI8	ro	FFH	Asynchronous communication.
1802	0	Number of elements	UI8	ro	2	Communication parameters of 3rd Transmit PDO
	1	COB-ID	UI32	ro	380H + NodeID	Determined using the CANopen minimum system ID assignment procedure.
	2	Transmission type	UI8	ro	FFH	Asynchronous communication.
1803	0	Number of elements	UI8	ro	2	Communication parameters of 4th Transmit PDO
	1	COB-ID	UI32	ro	480H + NodeID	Determined using the CANopen minimum system ID assignment procedure.
	2	Transmission type	UI8	ro	FFH	Asynchronous communication. (Not used, will not be transmitted)
1A00	0	Number of mapped Entries in Tx PDO 1	UI8	ro	2	Mapping parameters of the 1st Transmit-PDO
	1	1st Object	UI32	ro	29000120H	32 bit IEEE754 floating point weight value.
	2	2nd Object	UI32	ro	29000D10H	Status
1A01	0	Number of mapped Entries in Tx PDO 2	UI8	ro	2	Mapping parameters of the 2nd Transmit-PDO
	1	1st Object	UI32	ro	29000620H	32 bit IEEE754 floating point average value.
	2	2nd Object	UI32	ro	29000D10H	Status
1A02	0	Number of mapped Entries in Tx PDO 3	UI8	ro	2	Mapping parameters of the 3rd Transmit- PDO
	1	1st Object	UI32	ro	29000320H	32 bit IEEE754 floating point Tare
	2	2nd Object	UI32	ro	29000D10H	Module Status
1A03	0	Number of mapped Entries in Tx PDO 4	UI8	ro	0	Mapping parameters of the 4th Transmit-PDO (disabled)
2004	0	Number of entries.	UI8	ro	5	Number of parameters.
	1	Dummy	UI8	wo	-	Not used
	2	Calibration	UI8	wo	-	Save calibration settings (TAC protected)
	3	General set-up	UI8	wo	-	Save general set-up parameters
	4	Dummy	UI8	wo	-	Not used
	5	Set-points	UI8	wo	-	Save set-point parameters.
2006	0	Number of entries	UI8	ro	2	Number of system entries.
	1	Dummy	UI8	wo	-	Not used
	2	Factory Default	UI8	wo	-	Set factory default values (TAC protected)
	3	Command byte 1	UI8	wo	-	See RPDO1
	4	Command byte 2	UI8	wo	-	See RPDO2
2100	0	Number of entries.	UI8	ro	23	Number of parameters.
	1	Dummy	I32	rw	0	Not used
	2	Dummy	I32	rw	0	Not used
	3	Dummy	I32	rw	0	Not used
	4	Filter setting	I32	rw	3	Filter setting
	5	Dummy	I32	rw	0	Not used
	6	Logic outputs	I32	rw	-	Digital Outputs
	7	Logic inputs	I32	ro	-	Digital Inputs
	8	Measure Time	I32	rw	0	Measuring Time
	9	Filter Mode	I32	rw	0	Filter mode
	10	No motion Range	I32	rw	1	No-motion range
	11	No motion Time	I32	rw	1000	No-motion time
	12	Output Mask	I32	rw	0	Digital outputs mask
	13	Dummy	I32	rw	0	Not used
	14	Start Delay	I32	rw	0	Start Delay
	15	Dummy	I32	rw	0	Not used
	16	Dummy	I32	rw	0	Not used
	17	Update Rate	I32	rw	0	Update rate
	18	Zero Tracking	I32	rw	0	Zero track (TAC protected)
	19	Dummy	I32	rw	0	Not used
	20	Dummy	I32	rw	0	Not used
	21	Dummy	I32	rw	0	Not used
	22	Reserved	I32	rw	1	For internal use – do not change
	23	Preset Tare	I32	rw	0	Preset Tare

Index	Sub-Index	Name	Type	Attri-but	Default Value	Meaning
2300	0	Number of entries.	UI8	ro	20	Number of calibration parameters.
	1	Absolute gain	I32	rw	20000	Absolute gain calibrate (TAC protected)
	2	Absolute zero	I32	rw	0	Absolute zero calibrate (TAC protected)
	3	Calibrate enable	I32	rw	-	Calibrate enable (enables TAC when the TAC is written)
	4	Calibrate gain	I32	rw	20000	Calibrate gain (TAC protected)
	5	Dummy	I32	rw	0	Not used
	6	Dummy	I32	rw	0	Not used
	7	Calibrate max 1	I32	rw	999999	Calibrate max 1 (TAC protected)
	8	Calibrate min	I32	rw	-999999	Calibrate min (TAC protected)
	9	Dummy	I32	rw	0	Not used
	10	Calibrate zero	I32	rw	0	Calibrate zero (TAC protected)
	11	Decimal point	I32	rw	3	Decimal point (TAC protected)
	12	Display step size	I32	rw	1	Display step size (TAC protect)
	13	Multi Range	I32	rw	0	Multi range / multi interval selection (TAC protected)
	14	Calibrate max 2	I32	rw	0	Calibrate max 2 (TAC protected)
	15	Calibrate max 3	I32	rw	0	Calibrate max 3 (TAC protected)
	16	Initial zero range	I32	rw	0	Initial zero range (TAC protected)
	17	Zero Range	I32	rw	0	Zero range (TAC protected)
	18	Tare mode	I32	rw	0	Tare mode (TAC protected)
	19	Non volatile tare	I32	rw	0	Non volatile / volatile tare select (TAC protected)
	20	Non volatile zero	I32	rw	0	Non volatile / volatile zero select (TAC protected)
2500	0	Number of entries.	UI8	ro	11	Number of Check-Weigher parameters
	1	Trigger Level	I32	rw	0	Trigger Level
	2	Trigger Egde	I32	rw	0	Trigger Egde
	3	ReTrigWindow	I32	rw	65535	ReTrigWindow
	4	ReTrigTime	I32	rw	0	ReTrigTime
	5	HoldTime	I32	rw	0	HoldTime
	6	TareWindow	I32	rw	0	TareWindow
	7	TareTime	I32	rw	0	TareTime
	8	ReTrigStop	I32	rw	65535	ReTrigStop
	9	Dummy	I32	rw	0	Not used
	10	Dummy	I32	rw	0	Not used
	11	Δ Time	I32	rw	50	Delta time
2600	0	Number of entries.	UI8	ro	2	Number of Set-point parameters.
	1	Set-point 1	I32	rw	5000	Set-point 1 value
	2	Set-point 2	I32	rw	10000	Set-point 2 value
2700	0	Number of entries.	UI8	ro	2	Number of Set-point parameters.
	1	Set-point 1	I32	rw	1	Set-point 1 hysteresis
	2	Set-point 2	I32	rw	1	Set-point 2 hysteresis
2800	0	Number of entries.	UI8	ro	2	Number of Set-point parameters.
	1	Set-point 1	UI8	rw	0	Set-point 1 source
	2	Set-point 2	UI8	rw	0	Set-point 2 source
2900	0	Number of entries	UI8	ro	12	Number of entries in info array.
	1	Gross weight	REAL32	ro	-	
	2	Net Weight	REAL32	ro	-	
	3	Tare	REAL32	ro	-	
	4	Dummy	UI32	ro	0	Not used
	5	Dummy	UI32	ro	0	Not used
	6	Average weight	REAL32	ro	-	
	7	A/D sample	I32	ro	-	
	8	H&B Device ID	UI32	ro	-	
	9	H&B FW Version	UI32	ro	-	
	10	Device Status	UI32	ro	-	
	11	Dummy	UI32	ro	0	Not used
	12	Serial Number	UI32	ro	-	
	13	Extended status	UI32	ro	-	See TPDO's

13.3 Appendix C

Multiple load cell communication example in Serial

The LDB command set is based on a simple ASCII format, as per the example below:

Master (PC or PLC) sends: OP 1 (Open the device # 1)

Slave (LDB) sends: OK (acknowledging device #1 active)

Master sends: GG (Get the gross weight result)

Slave sends: G+123.45 (the gross weight with sign and Decimal point information)

Note: Load cell should be addressed from address 1 when communicating with multiple load cells using serial communication. (Address 0 cannot be used)

13.4 Appendix D

Modbus Index Tables

Index (hex)	Type	Size	Access	Function	
0x2000	Float	2	R	Gross Weight	This index returns the latest Gross value obtained from the device. The format is IEEE754 Single precision floating point format. The 32 bit data is obtained by reading 2 16-bit registers from index 0x2000. See also command description: **GG** get gros value
0x2002	Float	2	R	Net Weight	This index returns the latest Net value obtained from the device. The format is IEEE754 Single precision floating point format. The 32 bit data is obtained by reading 2 16-bit registers from index 0x2002. See also command description: **GN** get net value
0x2004	Float	2	R	Tare	This index returns the actual Tare weight. The format is IEEE754 Single precision floating point format. The 32 bit data is obtained by reading 2 16-bit registers from index 0x2004. See also command description: **GT** Get Tare
0x2008	Float	2	R	Average Weight	This index returns the latest average weight value obtained from the device. The format is IEEE754 Single precision floating point format. The 32 bit data is obtained by reading 2 16-bit registers from index 0x2008. See also command description: **GA** get average value
0x2020	Int32	2	R	Gross Weight	This index returns the Gross weight value obtained from the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2020. See also command description: **GG** get gros value
0x2022	Int32	2	R	Net Weight	This index returns the Net weight value obtained from the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2022. See also command description: **GN** get net value
0x2024	Int32	2	R	Tare	This index returns the actual Tare weight. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2024. See also command description: **GT** Get Tare
0x2028	Int32	2	R	Average Weight	This index returns the latest average weight value obtained from the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2028. See also command description: **GA** get average value
0x202A	Int32	2	R	A/D Sample	This index returns the current ADC value obtained from the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x202A. See also command description: **GS** get sample
0x202C	Int32	2	R	Device ID	This index returns the ID of the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x202C. The high word (0x202C) are 0(zero) and the low word (0x202D) should be split into two bytes: 1st byte is the minor ID, e.g. 0x90 (144d) if the device is a LDU 179 2nd byte is the major ID, e.g. 0x17 (23d) if the device is a LDU 179. See also command description: **ID** identification device
0x202E	Int32	2	R	Firmware Version	This index returns the current firmware version of the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x202E. The high word (0x202E) are 0 (zero) and the low word (0x202F) should be split into two bytes: 1st byte is the minor IV, e.g. 0x10 (16d). 2nd byte is the major IV, e.g. 0x01 (1d). See also command description: **IV** Firmware Version
0x2030	Int32	2	R	Device Status	This index returns the current status for the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2030. See also command description: **IS** Device Status
0x2034	Int32	2	R	Serial Number	This index returns the serial number of the device. The 32 bit integer data is obtained by reading 2 16-bit registers from index 0x2034. See also command description: **RS** Read Serial Number

0x2060	Int16	1	R	Qualifier	This index returns the latest qualifier value obtained from the device. The 16 bit integer data is obtained by reading one 16-bit registers at index 0x2060. The Qualifier bit values are: 0x0001 - Under range 0x0002 - Over range 0x0004 – Not used 0x0008 – Exactly zero 0x0010 - No motion, still stand, steady state 0x0020 - Tare set 0x0040 – Not used 0x0080 - Invalid weighing (wire-break, A/D ref. out of range) 0x0100 - Set-point 0 (source>limit) 0x0200 - Set-point 1 0x0400 - Set-point 2 0x0800 – Set-point 3 0x1000 – Not used 0x2000 – Not used 0x4000 – Not used 0x8000 – Not used
0x2061	Int16	1	W	Bit Commands	This index is used to set or reset Zero and Tare. The 16 bit integer data is accessed by writing one 16-bit register at index 0x2061. See also command description: 0x01: **RZ** Reset Zero 0x02: **SZ** Set Zero 0x04: **RT** Reset Tare 0x08: **ST** Set Tare
0x2062	Int16	1	W	Trigger	This index is used to trigger measurements in the device. The 16 bit integer data is accessed by writing one 16-bit register at index 0x2062. The value 0x0080 starts the triggered measurement. see also command description: **TR** Software Trigger
0x2066	Int16	1	W	Save in EEPROM	This index is used to initiate writing to the device EEPROM. The 16 bit integer data is accessed by writing one 16-bit register at index 0x2066. The values are: 0x0002: **CS** Save calibration 0x0004: **WP** Save general setup parameters 0x0010: **SS** Save set-point parameters 0x8000: **FD** Factory default
0x2067	Int16	1	RW	Set Point Selection	This index is used to select Set point in the device. The 16 bit integer data is accessed by writing one 16-bit register at index 0x2067. The values are 0 (zero) to 3 (dependent on the device type) and selects the set point acted upon by indexes: 0x2068, 0x206A and 0x206C This index act as the 'n' parameter for the A'n', H'n' and S'n' commands See also command description: **A'n'** Assign action, **H'n'** Hysteresis, **S'n'** Set point
0x2068	Int32	2	RW	Set Point Source	This Index is used to select Set point Source in a device. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2068. 0 = Gross weight as set point source 1 = Net weight as set point source 2 = Average value as set point source See also command description: **A'n'** (n = 0, 1)
0x206A	Int32	2	RW	Set Point Hysteresis	This index is used to get or set the Set point Hysteresis in a device. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x206A. See also command description: **H'n'** (n = 0, 1)
0x206C	Int32	2	RW	Set Point Value	This index is used to get or set the Set point limit in a device. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x206C. See also command description: **S'n'** (n = 0, 1)
0x2072	Int16	1	RW	Interface selector	This index is used as an interface selector for the communication parameters. Zero (0) select the serial channel and 1 select the CAN interface. See also command description: **NS'x', 'y', 'z'** Network settings.
0x2073	Int16	1	RW	Parameter selector	This index is used as a parameter selector for the communication parameters. See also command description: **NS'x', 'y', 'z'** Network settings.
0x207A	Int32	2	RW	Communication parameters	This index is used to get or set the communication parameters in a device. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x207A. See also command description: **NS'x', 'y', 'z'** Network settings.

0x2106	Int32	2	RW	Filter Setting	This index is used to select the filter setting. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2106. The values of the low pass filters are 0 to 8. See also command description: **FL** Filter Value
0x210A	Int32	2	RW	Logic Output	This index Reads/Modify the status of the physical output signals. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x210A. See also command description: **IO** Status of the logic Output
0x210C	Int32	2	R	Logic Input	This index reads the status of the physical input signals. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x210C. See also command description: **IN** Read status of logic Input
0x2110	Int32	2	RW	Filter Mode	This index chooses the filter mode; permitted values are "0" for IIR and "1" for FIR. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2110. See also command description: **FM** Filter Mode
0x2112	Int32	2	RW	No Motion Range	This index Reads/Modify the maximum number of counts allowed as no motion. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2112. See also command description: **NR** No Motion Range
0x2114	Int32	2	RW	No Motion Time	This index Reads/Modify the minimum time the weight must stay within NR to be no motion. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2114. See also command description: **NT** No Motion Time
0x2116	Int32	2	RW	Logic Output Mask	This index Reads/Modify the mask of the logic Outputs. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2116. See also command description: **OM** Control of logic Outputs
0x2118	Int32	2	R	Tare Value	This Index Reads the Tare value. The 32 bit integer data accessed by reading 2 16-bit registers from index 0x2118. See also command description: **GT** Get tare value
0x2120	Int32	2	RW	Update Rate	This index chooses Average after the filter by 2 exp. UR samples. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2120. The values of the update rate are 0 to 7 (2 exp 0 = 1, 2 exp 7 = 128). See also command description: **UR** Update Rate
0x2122	Int32	2	RW	Zero Tracking	This index enables or disables the zero tracking (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2122. The values of Zero Tracking function are 0 to 255. See also command description: **ZT** Zero Tracking
0x212A	Int32	2	RW	Pre-Filter	Turns the digital pre-filter on or off. See also command description: **PF** Pre Filter
0x212C	Int32	2	RW	Set Preset Tare	See also command description: **SP** Set Preset Tare
0x2200	Int32	2	RW	Absolute Gain Calibration	This index Reads/Modify the absolute gain point (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2200. See also command description: **AG** Absolute Gain
0x2202	Int32	2	RW	Absolute Zero Calibration	This index Reads/Modify the absolute zero point (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2202. See also command description: **AZ** Absolute Zero
0x2204	Int32	2	RW	Calibrate Enable	This index sets the calibration functions to the enabled state. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2204. See also command description: **CE** Calibrate Enable
0x2206	Int32	2	RW	Calibrate Gain	This index sets the calibration gain (span) value (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2206. See also command description: **CG** Calibrate Gain
0x220C	Int32	2	RW	Calibrate Max	This index Reads/Modify the maximum allowable output value (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x220C. See also command description: **CM** (CM1) Maximum Output
0x220E	Int32	2	RW	Calibrate Min	This index Reads/Modify the minimum allowable output value (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x220E. See also command description: **CI** Minimum Output
0x2212	Int32	2	W	Calibrate Zero	This index sets the calibration zero point (TAC protected). The 32 bit integer data accessed by writing 2 16-bit registers from index 0x2212. See also command description: **CZ** Calibrate Zero
0x2214	Int32	2	RW	Decimal Point	This index Reads/Modify the decimal point position (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2214. See also command description: **DP** Decimal Point
0x2216	Int32	2	RW	Display Step Size	This index Reads/Modify the the display step size (TAC protected). The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2216. See also command description: **DS** Display Step Size
0x2218	Int32	2	RW	Select Multi Range	See also command description: **MR** MRange
0x221A	Int32	2	RW	CM2 max display value	See also command description: **CM2** Maximum2

0x221C	Int32	2	RW	CM3 max display value	See also command description: **CM3** Maximum3
0x221E	Int32	2	RW	Initial Zero @ Power ON	This index enables or disables the initial zero function @ power ON. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x221E. See also command description: **ZI** Initial Zero ON/OFF
0x2220	Int32	2	RW	Zero Range	This index selects the zero range. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x221E. A value of 0 enables the standard zero range of ± 2% of maximum. See also command description: **ZR** Zero Range
0x2222	Int32	2	RW	Tare Mode	See also command description **TM** Tare Mode
0x2224	Int32	2	RW	Store Tare value @ Power OFF	This index enables or disables the tare storing in EEPROM @ power OFF. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2224. See also command description: **TN** Store Zero Value
0x2226	Int32	2	RW	Store Zero Value @ Power OFF	This index enables or disables the zero storing inh EEPROM @ power OFF. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2226. See also command description: **ZN** Store Zero Value
0x2400	Int32	2	RW	Trigger Level	This index selects the trigger level. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2400. See also command description: **TL** Trigger Level
0x2402	Int32	2	RW	Trigger Egde	This index selects rising or falling slope trigger. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2402. See also command description: **TE** Trigger Edge
0x2404	Int32	2	RW	Re-trigger Window	This index selects the re-trigger window size. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2404. See also command description: : **RW** Re-trigger Window
0x2406	Int32	2	RW	Re-trigger Time	This index selects the re-trigger time. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2406. See also command description: : **TT** Re-trigger Time
0x2408	Int32	2	RW	Hold Time	This index Reads/Modify the Hold time for the set points. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2408. See also command description: **HT** Hold Time
0x240A	Int32	2	RW	Tare Window	This index Reads/Modify the tare window. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x240A. See also command description: **TW** Tare Window
0x240C	Int32	2	RW	Tare Time	This index Reads/Modify the tare time. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x240C. See also command description: **TI** Tare Time
0X240E	Int32	2	RW	Re-trig Stop	This index selects the re-trig stop threshold in increments. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x240E. See also command description: : **TS** Re-trig Stop
0x2410	Int32	2	RW	Measuring Time	This index Reads/Modify the time over which the average value will be built. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2410. See also command description: **MT** Measure Time
0x2412	Int32	2	RW	Start Delay	This index Reads/Modify the delay between falling/rising edge of the trigger pulse and start of the measurement. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2412. See also command description: **SD** Start Delay
0x2414	Int32	2	RW	Delta Time	This index selects the time in milli seconds for sub-averaging. The 32 bit integer data accessed by reading or writing 2 16-bit registers from index 0x2414. See also command description: : **DT** Delta Time
0x3300	Int32 Int32 Int16	5	R	Combined result, integer	This index reads the gross weight, the net weight and the qualifier. The data is accessed by reading five 16-bit registers from index 0x3300. See also command description: **GW** Get data string "net, gross and status"
0x3500	Float Float Int16	5	R	Combined result, floating point	This index reads the gross weight, the net weight and the qualifier. The data is accessed by reading five 16-bit registers from index 0x3500. See also command description: **GW** Get data string "net, gross and status"