PENKO Engineering B.V.

Your Partner for Fully Engineered Factory Solutions

Protocol description:
PENKO 1020 PROFIBUS-DP
For firmware V1.5.1.9.0.6 and above

PENKO
an ETC Company

PENKO PROFIBUS-DP protocol

Table of Contents

PENKO PROFIBUS-DP protocol

Introduction

PROFIBUS (Process Field Bus) is a standard for fieldbus communication in automation technology. PROFIBUS is openly published as part of IEC 61158.

This document describes the PENKO PROFIBUS-DP implementation for the PENKO 1020 range.

1 Basics

The GSD information and PENKO devices that support the protocol.

1.1 GSD

This document describes the General Station Description, or GSD, for the 1020 indicator (revision 1.05) and the 1020 controller (revision 1.05)

GSD filename 1020 indicator: **PTEI0E02.GSD**

GSD filename 1020 controller: **PTEC0E02.GSD**

The PENKO 1020 can be an indicator or a controller with a predefined program for a check weigher, belt weigher or mono filler.

1.2 Devices

The following PENKO devices support PROFIBUS-DP:

Device	PROFIBUS-DP
1020 IND	Yes, starting at first firmware version*
1020 BLT	Yes, starting at first firmware version*
1020 CHK	Yes, starting at first firmware version*
1020 MFL	Yes, starting at first firmware version*

** This manual applies for firmware V1.5.1.9.0.6 and above, for older versions check comments in GSD file*

1.3 Supported baudrates

The following baudrates are supported:

Baudrate (bps)	Max Tsdr (Station delay of responder Time)
9k6	60
19k2	60
45k45	60
93.75	60
187k5	60
500k	100
1M5	150
3M	250
6M	450
12M	800

2 Addresses

The following addresses are used:

2.1 Inputs

1020 indicator:

Name	Data type	Word offset
Weight register	Double Word, 32 bit signed integer/float	00
Weigher status	Word	02
Weigher control	Byte	03 - high byte
Weight register selector	Byte	03 - low byte
Inputs 1…16*	Word	04
Outputs 201…216*	Word	05
Preset Tare	Double Word, 32 bit signed integer	06
Gross x 10	Double Word, 32 bit signed integer/float	08
Net x 10	Double Word, 32 bit signed integer/float	10
Tare x 10	Double Word, 32 bit signed integer/float	12
Multi-range weight	Double Word, 32 bit signed integer/float	14

** Inputs 4 - 16 are virtual inputs generated by the software*
** Outputs 205 - 216 are virtual outputs generated by the software*

1020 controller:

Name	Data type	Word offset
Weight register	Double Word, 32 bit signed integer/float	00
Weigher status	Word	02
Weigher control	Byte	03 - high byte
Weight register selector	Byte	03 - low byte
Inputs 1…16*	Word	04
Outputs 201…216*	Word	05
Markers 401…416	Word	06
Markers 417…432	Word	07
Ext register 1 - 5 - 9 - 13*	Double Word, 32 bit signed integer	08
Ext register 2 - 6 - 10 - 14*	Double Word, 32 bit signed integer	10
Ext register 3 - 7 - 11 - 15*	Double Word, 32 bit signed integer	12
Ext register 4 - 8 - 12 - 16*	Double Word, 32 bit signed integer	14

** Inputs 4 - 16 are virtual inputs generated by the software*
** Outputs 205 - 216 are virtual outputs generated by the software*
** Extended register number depend on channel (0, 1, 2, 3) set with control bit 7 and 8, see control bits*

2.2 Outputs

1020 indicator:

Name	Data type	Word offset
Weigher control	Byte	00 - high byte
Weight register selector	Byte	00 - low byte
Preset Tare*	Double Word, 32 bit signed integer	01
Level 1*	Double Word, 32 bit signed integer	03
Level 2*	Double Word, 32 bit signed integer	05
Level 3*	Double Word, 32 bit signed integer	07
Level 4*	Double Word, 32 bit signed integer	09

** At rising edge of control bit 4, preset tare is activated with the value in this register, see control bits*
** Levels can only be written when control bit 6 and 7 are set, see control bits*

1020 controller:

Name	Data type	Word offset
Weigher control	Byte	00 - high byte
Weight register selector	Byte	00 - low byte
Markers 969…984	Word	01
Markers 985…1000	Word	02
Ext register 85 - 89 - 93 - 97*	Double Word, 32 bit signed integer/float	03
Ext register 86 - 90 - 94 - 98*	Double Word, 32 bit signed integer/float	05
Ext register 87 - 91 - 95 - 99*	Double Word, 32 bit signed integer/float	07
Ext register 88 - 92 - 96 - 100*	Double Word, 32 bit signed integer/float	09

** Extended register number depend on channel (0, 1, 2, 3) set with control bit 6 and 7, see control bits*

2.3 Status bits

Status bit #	Description
0	Hardware overload detected
1	Overload detected
2	Stable signal
3	In stable range
4	Zero corrected
5	Center of zero
6	In zero range
7	Zero tracking possible
8	Tare active
9	Preset tare active
10	New sample available
11	Calibration invalid
12	Calibration enabled
13	User certified operation
14	Invalid weight
15	Register function mode indication

2.4 Control bits

Control bit #	Description
0	Zero reset command
1	Zero set command
2	Tare off
3	Tare on
4	Preset tare command
5	Freeze*
6	Indicator channel 2^0*
7	Indicator channel 2^1*
0 + 1	Rising edge changes mode to Register Functions. Reset of bit #0 or #1 changes mode back to normal operation.
6 + 7	Enable writing level 1 - 4

** Freeze weigher registers at rising edge of bit for selected weigher. If bit is 0 registers will be updated. Use this bit to read out all necessary weigher registers without any interruption of the weigher.*

** Bit 6 and 7 set the channel for the extended registers. This is only applicable for the controller. Channel 0 selects the first extended register number, channel 1 the second, channel 2 the third and channel 3 the fourth. See table:*

Bit 7	Bit 6	Channel
0	0	0
0	1	1
1	0	2
1	1	3

2.5 Weight register

The indicator shown in the weight register can be selected with the weight register selector.

Selector Byte	Indicator	Description
0x00	WEIGHT	Multi range net weigher value
0x01	FAST GROSS	Unfiltered gross weigher value
0x02	FAST NET	Unfiltered net weigher value
0x03	DISPLAY GROSS	Filtered gross weigher value
0x04	DISPLAY NET	Filtered net weigher value
0x05	TARE	Tare value
0x06	PEAK	Highest reached weigher value
0x07	VALLEY	Lowest reached weigher value
0x08	HOLD*	Stores the weight value
0x09	WEIGHTx10	Multi range net weigher value shown with extra decimal
0x0A	FAST GROSSx10	Unfiltered gross weigher value shown with extra decimal
0x0B	FAST NETx10	Unfiltered net weigher value shown with extra decimal
0x0C	DISPLAY GROSSx10	Filtered gross weigher value shown with extra decimal
0x0D	DISPLAY NETx10	Filtered net weigher value shown with extra decimal
0x0E	TAREx10	Tare value shown with extra decimal
0x0F	PEAKx10	Highest reached weigher value shown with extra decimal
0x10	VALLEYx10	Lowest reached weigher value shown with extra decimal
0x11	HOLDx10*	Stores the weight value shown with extra decimal
0x12	mV signal	Direct mV value from the load cell(s)
0x13…0x77	INDICATOR REGISTERS	Indicator register 1…100
0x78…0xFF	RESERVED	Reserved

*Hold and Holdx10 are added in the 1020 firmware version V1.5.1.9.0.6 and above.

3 Register functions

By using register functions, all device parameters can be read and/or written.

The following parameters are used:

Parameter	Data type	Word offset
Input - parameter 1	Double Word	03
Input - parameter 2	Double Word	05
Input - parameter 3	Double Word	07
Input - parameter 4	Double Word	09
Output - result 1	Double Word	08
Output - result 2	Double Word	10
Output - result 3	Double Word	12
Output - result 4	Double Word	14
Enable register command mode	Byte	00 high byte (bit 0 + 1)
Status register command mode	Word	02 (bit 15)

Input:

Parameter	Data type	Description
1	Double Word	Low word = function code | High word = 0
2	Double Word	Input parameter, depending on function code
3	Double Word	Input parameter, depending on function code
4	Double Word	Input parameter, depending on function code

Output:

Result	Data type	Description
1	Double Word	Low word = function code | High word = error code
2	Double Word	Result, depending on function code
3	Double Word	Result, depending on function code
4	Double Word	Result, depending on function code

Activating:

The register command mode is activated at the rising edge of control bit 0 + 1. Status bit 15 indicates that the register command mode is active. Reset bit 0 or 1 to go back to normal operation mode.

Usage:

First write parameter 2, 3 and 4. Write parameter 1 as last because this parameter triggers the action.

3.1 Function codes

The following function codes are present:

Name	Code	Description
NOP	0	No Operation
CAL_ZERO	1	Calibrate zero by weight
CAL_SPAN	2	Calibrate span by weight
CAL_MV	3	Calibrate in mV/V
CAL_DEADLOAD	4	Calibrate dead load by measuring weight
CAL_INSERT	5	Calibrate multipoint insert by measuring weight
CAL_POINT	6	Calibrate multipoint read point at parameter index
CAL_DELETE	7	Calibrate multipoint delete point at parameter index
CAL_GEOGRAPHIC_ORIGIN_SET	8	Calibrate set geographic origin calibration
CAL_GEOGRAPHIC_ORIGIN_GET	9	Calibrate get geographic origin calibration
CAL_GEOGRAPHIC_LOCAL_SET	10	Calibrate set geographic local calibration
CAL_GEOGRAPHIC_LOCAL_GET	11	Calibrate get geographic local calibration
IND_MAXLOAD_SET	101	Indicator set maximum load
IND_MAXLOAD_GET	102	Indicator get maximum load
PDI_PATH_SET	201	PDI path set
PDI_PROPERTY_SET	202	PDI property set
PDI_PROPERTY_GET	203	PDI property get
PRINT	301	Print ticket or line to printer
PRINT_SUBTOTAL	302	Subtotals to printer
PRINT_TOTAL	303	Totals to printer
PRINT_DAYTOTAL	304	Day totals to printer
PRINT_BATCHTOTAL	305	Batch totals to printer
PRINT_LAYOUT	306	Custom total layout to printer not supported by SGM series and 1020
PRINT_ALIBI	307	Print to Alibi memory
PRINT_ALIBIMEMORY	308	print full alibi memory to printer
PRINT_EVENTMEMORY	309	print full event memory to printer
TOTAL_TOTALIZE	401	Totalize actual stable weight
TOTAL_SUBTOTAL	402	Read or reset actual subtotal
TOTAL_TOTAL	403	Read or reset actual totals
TOTAL_DAYTOTAL	404	Read or reset actual day totals
TOTAL_BATCHTOTAL	405	Read or reset actual batch totals
RFN_PROCESS_RECIPE_GET	501	Read MFL/CHK/BLT recipe
RFN_PROCESS_RECIPE_SET	502	Write MFL/CHK/BLT recipe

RFN_PROCESS_CONFIG_GET	601	Read MFL/CHK/BLT configuration
RFN_PROCESS_CONFIG_SET	602	Write MFL/CHK/BLT configuration
RFN_PROCESS_DATA	701	Read MFL/CHK/BLT process data

3.2 Error codes

The following error codes are present:

Name	Code	Description
SUCCES	**0**	**Function successful**
WRN_WARNING	**1000**	**System warnings:**
WRN_TIMEOUT	1001	Generic time-out warning
WRN_TOLOW	1002	Generic parameter to low warning
WRN_TOHIGH	1003	Generic parameter to high warning
WRN_ZERO	1004	Generic parameter/result is zero warning
WRN_NOTZERO	1005	Generic parameter/result is not zero warning
WRN_POSITIVE	1006	Generic parameter is positive warning
WRN_NEGATIVE	1007	Generic parameter is negative warning
WRN_FULL	1008	Generic something is full warning
WRN_EMPTY	1009	Generic something is empty warning
WRN_NOTFOUND	1010	Generic search not found warning
WER_WARNING	**1100**	**Weigher warnings:**
WER_NO_TARE	1101	Zero tare level, tare rst
ERR_ERROR	**2000**	**System errors:**
ERR_PARAMETER_INCORRECT	2001	Generic parameter error
ERR_TIMEOUT	2002	Generic time-out error
ERR_TOLOW	2003	Generic parameter to low error
ERR_TOHIGH	2004	Generic parameter to high error
ERR_ZERO	2005	Generic parameter/result is zero error
ERR_NOTZERO	2006	Generic parameter/result is not zero error
ERR_POSITIVE	2007	Generic parameter is positive error
ERR_NEGATIVE	2008	Generic parameter is negative error
ERR_FULL	2009	Generic something is full error
ERR_EMPTY	2010	Generic something is empty error
ERR_NOTFOUND	2011	Generic search not found error
ERR_FILE_NOT_FOUND	2012	Generic file not found error
WER_ERROR	**2100**	**Weigher errors:**
WER_NOT_STABLE	2101	Weigher not stable
WER_ABOVE_MAXLOAD	2102	Parameter above max load
WER_BELOW_ZERO	2103	Parameter below zero

WER_NOT_IN_ZERO_RANGE	2104	Not in zero range
WER_ARITHMIC_OVERFLOW	2105	Arrhythmic overflow occurred
WER_ADC_OVERFLOW	2106	A/D reads all 1's
WER_ADC_UNDERFLOW	2107	A/D reads all 0's
WER_GAIN_NEGATIVE	2108	Gain ref. < zero ref.
WER_GAIN_OVERFLOW	2109	Gain limit
WER_SAVE	2110	Save errors:
WER_SAVE_FLASH_EXHAUSTED	2111	Flash ROM exhausted
WER_SAVE_CREATE_HEADER	2112	Error on header creation
WER_SAVE_DATA_WRITE	2113	Error on data write
WER_SAVE_HEADER_VALIDATE	2114	Header validation failed
WER_SAVE_DEACTIVATE	2115	Deactivate old data fail
WER_LOAD	2116	Load errors
WER_LOAD_NOT_FOUND	2117	Item not found in store
WER_LOAD_DATA_ERROR	2118	Error in stored data
WER_BAD_CALIBRATION	2119	No calibration available
WER_NOT_ENABLED	2120	Action not enabled
WER_MCAL_NOT_FOUND	2121	Multi-point not found
WER_MCAL_OVERFLOW	2122	Calibration table full
WER_TARE_ACTIVE	2123	Not allowed, tare active
WER_NOT_ALLOWED	2124	Action is not allowed
WER_ADC_NOPOWER	2125	ADC has no power
ERR_DOSER	**2200**	**Doser errors**
ERR_POSITION	**2300**	**Position errors**
ERR_SPCAPP	**2400**	**SPC-application errors**
ERR_SCOPE	**2500**	**Scope errors**
ERR_INTERPRETER	**2600**	**Interpreter errors**
ERR_USB	**3000**	**USB errors - use USB routines for returning error texts**
ERR_FLASH	**3100**	**FLASH file system errors**

3.3 Calibration functions

This chapter describes the calibration function codes.

3.3.1 CAL_ZERO

Calibrate zero by weight. Function code = 1.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
1	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
1	Not used	Not used	Not used

3.3.2 CAL_SPAN

Calibrate span by weight. Function code = 2.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
2	Span weight	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
2	Not used	Not used	Not used

Example - calibrate span on 1.200kg:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
2	1200	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
2	Not used	Not used	Not used

Example - calibrate span without loading scale - will result in **error**:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
2	1200	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
138215426	Not used	Not used	Not used

The result is **138215426**

- The function code (low word) is **2**
- The error code (high word) is **2109 - WER_GAIN_OVERFLOW, Gain limit**

3.3.3 CAL_MV

Theoretic calibration by millivolts. Function code = 3.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
3	Fixed point mV/V value	Maximum weight at mV/V	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
3	Not used	Not used	Not used

Example - theoretic calibration of load cell 200kg @ 2.0012mV/V:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
3	20012	200	Not used

3.3.4 CAL_DEADLOAD

Calibrate dead load by measuring weight. Function code = 4.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
4	Actual weight on scale	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
4	Not used	Not used	Not used

Example - calibration of dead load with 12kg on the scale:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
4	12	Not used	Not used

3.3.5 CAL_INSERT

Multipoint calibration up to 10 points. Insert or replace a calibration point. Function code = 5.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
5	Actual weight on scale	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
5	Not used	Not used	Not used

Example - add calibration point of 10.000kg - if the point already exists, its ADC value is replaced:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
5	10000	Not used	Not used

3.3.6 CAL_POINT

Multipoint calibration up to 10 points. Read the calibration point at index (1...10). Function code = 6.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
6	Index (1...10)	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
6	Index	Calibration reference weight	Calibration in mV

Example - read calibration point 1 (10.000kg @ 9.9975mV/V):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
6	1	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
6	1	10000	9.9975

3.3.7 CAL_DELETE

Multipoint calibration up to 10 points. Delete the calibration point at index (1...10). Function code = 7.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
7	Index (1...10)	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
7	Index	Not used	Not used

Example - delete calibration point 1:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
7	1	Not used	Not used

3.3.8 CAL_GEOGRAPHIC_ORIGIN_SET

Geographic correction. Set the origin calibration location. Function code = 8.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
8	Fixed point latitude degrees	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
8	Not used	Not used	Not used

Example - set origin latitude to 50.00 degrees:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
8	5000	Not used	Not used

3.3.9 CAL_GEOGRAPHIC_ORIGIN_GET

Geographic correction. Get the origin calibration location. Function code = 9.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
9	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
9	Fixed point latitude degrees	Not used	Not used

Example - get origin latitude (50.00 degrees):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
9	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
9	5000	Not used	Not used

3.3.10 CAL_GEOGRAPHIC_LOCAL_SET

Geographic correction. Set the actual scale location. Function code = 10.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
10	Fixed point latitude degrees	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
10	Not used	Not used	Not used

Example - set actual latitude to 50.00 degrees:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
10	5000	Not used	Not used

3.3.11 CAL_GEOGRAPHIC_LOCAL_GET

Geographic correction. Get the actual scale location. Function code = 11.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
11	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
11	Fixed point latitude degrees	Not used	Not used

Example - get location latitude (50.00 degrees):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
11	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
11	5000	Not used	Not used

3.4 Indicator functions

This chapter describes the indicator function codes.

3.4.1 IND_MAXLOAD_SET

Set the indicator maximum load. Function code = 101.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
101	Max load	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
101	Not used	Not used	Not used

Example - set the maximum load to 10.020kg:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
101	10020	Not used	Not used

3.4.2 IND_MAXLOAD_GET

Get the indicator maximum load. Function code = 102.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
102	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
102	Max load	Not used	Not used

Example - get the maximum load (10.020kg):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
102	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
102	10020	Not used	Not used

3.5 PDI functions

This chapter describes the PDO function codes.

3.5.1 PDI_PATH_SET

Set the PDI path to perform the action on. Function code = 201.

Request:

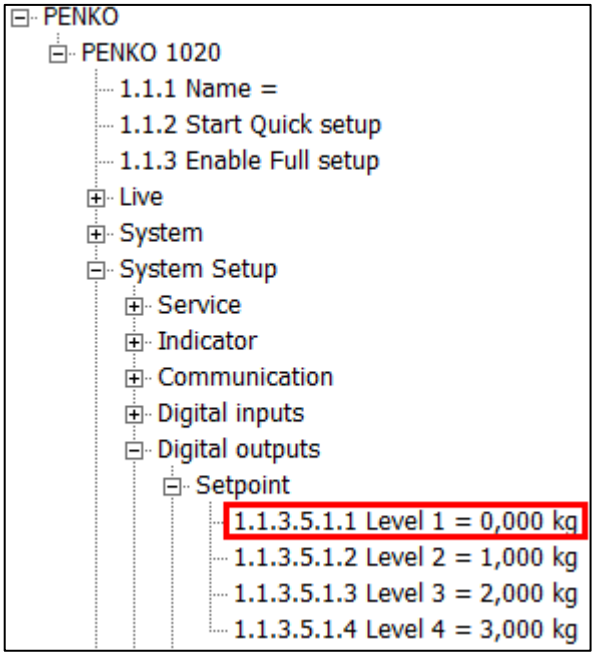
Parameter 1	Parameter 2	Parameter 3	Parameter 4
201	Path no. 1,2,3,4	Path no. 5,6,7,8	Path no. 9,10,11,12

Reply:

Result 1	Result 2	Result 3	Result 4
201	Path no. 1,2,3,4	Path no. 5,6,7,8	Path no. 9,10,11,12

PDI (PENKO Device Interface) represents the device configuration in a tree structure. Every property has its own unique path number. The tree is used in the PENKO configuration tools Pi Mach II and PDI Client, both available at www.penko.com/software

For example, a part of the 1020 looks like this:



Setpoint 1 has path number 1.1.3.5.1.1. This results in the following input parameters:

Fill the path with zeros to make 12 numbers: 1.1.3.5.1.1.0.0.0.0.0.0

Parameter 1	PDI_PATH_SET	201
Parameter 2	0x01 0x01 0x03 0x05	16843525
Parameter 3	0x01 0x01 0x00 0x00	16842752
Parameter 4	0x00 0x00 0x00 0x00	0

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
201	16843525	16842752	0

Reply:

Result 1	Result 2	Result 3	Result 4
201	16843525	16842752	0

The PDI path is now set.

If the path is not found, all zeros are returned.

3.5.2 PDI_PROPERTY_SET

Set a PDI property for the selected PDI path. Function code = 202.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
202	Property value	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
202	Not used	Not used	Not used

Example - set setpoint 1 to 0.500kg (path must be selected with PDI_PATH_SET):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
202	500	Not used	Not used

3.5.3 PDI_PROPERTY_GET

Get a PDI property from the selected PDI path. Function code = 203.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
203	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
203	Property value integer/string	Property value string optional	Property value string optional

Example - get setpoint 1 (0.500kg) (path must be selected with PDI_PATH_SET):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
203	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
203	500	Not used	Not used

Example - get the software version number (1.4.3.9.0.1) (path must be selected with PDI_PATH_SET):

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
203	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
203	825111598	858667310	808333568

The result is a string containing the version number:

Result 2	825111598	0x31 0x2E 0x34 0x2E	1.4.
Result 3	858667310	0x33 0x2E 0x39 0x2E	3.9.
Result 4	808333568	0x30 0x2E 0x31 0x00	0.1

1.4.3.9.0.1

3.6 Printer functions

This chapter describes the printer function codes.

3.6.1 PRINT

Print ticket or line layout, depending on set layout in device. Function code = 301.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
301	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
301	Gross weight	Net weight	Tare weight

Example - print:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
301	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
301	699	620	79

Device printer layout setting: **ticket**

```
---------------------------------------
DATE                        03-09-14
TIME                        11:02.51
TICKET NUMBER:                    42

NET                           0,620 kg
Tare                          0,079 kg
                           --------- +
GROSS                         0,699 kg
---------------------------------------
```

Device printer layout setting: **line**

```
---------------------------------------
NR       (PRESET)TARE kg       NET kg
 75              0,079         0,620
 76              0,079         0,620
 77              0,079         0,620
```

3.6.2 PRINT_SUBTOTAL

Print subtotal to printer. Function code = 302.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
302	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
302	Subtotal gross weight	Subtotal net weight	Subtotal tare weight

Example - print subtotal:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
302	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
302	3078	2742	336

Printed ticket:

```
----------------------------------------
DATE                        03-09-14
TIME                        13:53.25
TICKET NUMBER:                     3

SUBTOTAL NET                   2,742 kg
SUBTOTAL TARE                  0,336 kg
                             --------- +
SUBTOTAL GROSS                 3,078 kg
----------------------------------------
```

3.6.3 PRINT_TOTAL

Print total to printer. Function code = 303.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
303	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
303	Total gross weight	Total net weight	Total tare weight

Example - print total:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
303	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
303	7182	6398	784

Printed ticket:

```
--------------------------------------
DATE                        03-09-14
TIME                        14:02.04
TICKET NUMBER:                     7

TOTAL NET                      6,398 kg
TOTAL TARE                     0,784 kg
                           --------- +
TOTAL GROSS                    7,182 kg
--------------------------------------
```

3.6.4 PRINT_DAYTOTAL

Print day total to printer. Function code = 304.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
304	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
304	Day total gross weight	Day total net weight	Day total tare weight

Example - print day total:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
304	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
304	3454	3118	336

Printed ticket:

```
--------------------------------------
DATE                        03-09-14
TIME                        14:09.36
TICKET NUMBER:                     3

DAY TOTAL NET                3,118 kg
DAY TOTAL TARE               0,336 kg
                           --------- +
DAY TOTAL GROSS              3,454 kg
--------------------------------------
```

3.6.5 PRINT_BATCHTOTAL

Print batch total to printer. Function code = 305.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
305	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
305	Batch total gross weight	Batch total net weight	Batch total tare weight

Example - print batch total:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
305	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
305	10636	9516	1120

Printed ticket:

```
--------------------------------------
DATE                        03-09-14
TIME                        14:12.08
TICKET NUMBER:                    10

BATCH TOTAL NET              9,516 kg
BATCH TOTAL TARE             1,120 kg
                           --------- +
BATCH TOTAL GROSS           10,636 kg
--------------------------------------
```

3.6.6 PRINT_LAYOUT

Print to Printer function custom layout 1-n. Function code = 306.

Not supported by 1020 and SGM series.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
306	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
306	Layout number 1-n	Not used	Not used

Example - print with custom layout:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
306	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
306	1	Not used	Not used

3.6.7 PRINT_ALIBI

Print to Alibi memory function. Store the actual stable weight in Alibi memory. Function code = 307.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
307	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
307	UID*	Gross/Net	Preset (Tare)

** UID can exceed the maximum positive value of the LONG data type (signed). Use the DWORD data type (unsigned).*

Example - write to Alibi memory, with active tare:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
307	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
307	1944985600	1315	112

Corresponding Alibi records:

Entry	Code	Date/Value	Time/Unit	UID
00001/00004	Alibi 001	03-09-14	15:00:46	**1944985600**
00002/00004	Net	**1.315**	kg	3803586561
00003/00004	Tare	**0.112**	kg	1269178371
00004/00004	Gross	1.427	kg	0718544901

Example - write to Alibi memory, without active tare:

Reply:

Result 1	Result 2	Result 3	Result 4
307	1660428288	1711	0

Corresponding Alibi records:

Entry	Code	Date/Value	Time/Unit	UID
00001/00002	Alibi 001	03-09-14	15:02:34	**1660428288**
00002/00002	Gross	**1.711**	kg	1133518849

3.6.8 PRINT_ALIBIMEMORY

Print the complete Alibi memory to a printer. Function code = 308.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
308	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
308	Not used	Not used	Not used

Example - print the complete Alibi memory:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
308	Not used	Not used	Not used

Printed ticket:

```
Device: 1020
Serial Number FFFFFFFF
Date : 03-09-14   Time : 15:19:19
Alibi Memory
Number UID        Code               Date/Value      Time/Unit
     1 1660428288 Alibi 001            03-09-14      15:11:28
     2 1133518849 Gross                   1.711      kg
     3 1941708803 Alibi 001            03-09-14      15:19:08
     4 3786547204 Net                     1.162      kg
     5 3158056966 Tare                    0.350      kg
     6 0510926856 Gross                   1.512      kg
     7 1941708810 Alibi 001            03-09-14      15:19:08
     8 3786547211 Net                     1.162      kg
     9 3158056973 Tare                    0.350      kg
    10 0510926863 Gross                   1.512      kg
```

3.6.9 PRINT_EVENTMEMORY

Print the complete Event log to a printer. Function code = 309.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
309	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
309	Not used	Not used	Not used

Example - print the complete Event log:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
309	Not used	Not used	Not used

Printer ticket:

```
Device: 1020
Serial Number FFFFFFFF
Date : 03-09-14   Time : 15:33:59
Event Log
Number UID        Code               Date/Value      Time/Unit
     1 0841613312 TAC Changed          30-06-14      11:43:48
     2 1371668481 Events Cleared       30-06-14      11:43:48
     3 1251344386 System Default       30-06-14      12:08:50
     4 1182662659 SoftwareUpdate       30-06-14      12:10:10
     5 4005953540 Alibi Cleared        30-06-14      13:50:20
     6 4005953541 Alibi Cleared        30-06-14      13:50:20
     7 2012479494 CAL Changed          30-06-14      13:58:34
     8 1503395847 CAL Changed          30-06-14      13:58:42
     9 4230086664 Alibi Cleared        30-06-14      14:29:58
    10 3949592585 SoftwareUpdate       31-07-14      09:11:28
```

3.7 Total functions

This chapter describes the total function codes.

3.7.1 TOTAL_TOTALIZE

Totalize actual stable weight. Function code = 401.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
401	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
401	Added gross weight	Added net weight	Added tare weight

Example - add actual stable weight to total:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
401	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
401	1512	1162	350

3.7.2 TOTAL_SUBTOTAL

Get the subtotal weights. Function code = 402.

Leave parameter 2 empty to read the subtotal weights.

Set parameter 2 to **0x55AA55AA** to reset the subtotal weights.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
402	Optional	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
402	Subtotal gross weight	Subtotal net weight	Subtotal tare weight

Example - read the subtotal weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
402	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
402	12096	9296	2800

Example - reset the subtotal weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
402	1437226410	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
402	12096	9296	2800

When reading again, all subtotal weights are 0.

3.7.3 TOTAL_TOTAL

Get the total weights. Function code = 403.

Leave parameter 2 empty to read the total weights.

Set parameter 2 to **0x55AA55AA** to reset the total weights.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
403	Optional	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
403	Total gross weight	Total net weight	Total tare weight

Example - read the total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
403	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
403	12096	9296	2800

Example - reset the total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
403	1437226410	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
403	12096	9296	2800

When reading again, all total weights are 0.

3.7.4 TOTAL_DAYTOTAL

Get the day total weights. Function code = 404.

Leave parameter 2 empty to read the day total weights.

Set parameter 2 to **0x55AA55AA** to reset the day total weights.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
404	Optional	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
404	Day total gross weight	Day total net weight	Day total tare weight

Example - read the day total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
404	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
404	12096	9296	2800

Example - reset the day total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
404	1437226410	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
404	12096	9296	2800

When reading again, all day total weights are 0.

3.7.5 TOTAL_BATCHTOTAL

Get the batch total weights. Function code = 405.

Leave parameter 2 empty to read the batch total weights.

Set parameter 2 to **0x55AA55AA** to reset the batch total weights.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
405	Optional	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
405	Batch total gross weight	Batch total net weight	Batch total tare weight

Example - read the batch total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
405	Not used	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
405	12096	9296	2800

Example - reset the batch total weights:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
405	1437226410	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
405	12096	9296	2800

When reading again, all batch total weights are 0.

3.8 Controller functions

This chapter describes the controller functions for the belt weigher, check weigher and mono filler.

3.8.1 RFN_PROCESS_RECIPE_GET

Get the value of the selected recipe parameter. Function code = 501.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
501	Recipe param	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
501	Recipe param	Value	Not used

Example - get the value of recipe parameter 1:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
501	1	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
501	1	2000	Not used

3.8.2 RFN_PROCESS_RECIPE_SET

Set the value of the selected recipe parameter. Function code = 502.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
502	Recipe param	Value	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
502	Recipe param	Not used	Not used

Example - set the value of recipe parameter 2 to 500:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
502	2	500	Not used

3.8.3 RFN_PROCESS_CONFIG_GET

Get the value of the selected configuration parameter. Function code = 601.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
601	Config param	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
601	Config param	Value	Not used

Example - get the value of configuration parameter 1:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
601	1	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
601	1	2000	Not used

3.8.4 RFN_PROCESS_CONFIG_SET

Set the value of the selected configuration parameter. Function code = 602.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
602	Config param	Value	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
602	Config param	Not used	Not used

Example - set the value of configuration parameter 2 to 500:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
602	2	500	Not used

3.8.5 RFN_PROCESS_DATA

Get the value of the selected process data parameter. Function code = 701.

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
701	Process param	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
701	Process param	Value	Not used

Example - get the value of process parameter 1:

Request:

Parameter 1	Parameter 2	Parameter 3	Parameter 4
701	1	Not used	Not used

Reply:

Result 1	Result 2	Result 3	Result 4
701	1	2000	Not used

4 Examples

These examples are made with an Omron CJ2M-CPU31 PLC and CJ1W-PRM21 Profibus module using the CX-One software suite. The outputs start at address CIO3200. The inputs start at address CIO3300.

4.1 Inputs

Read weight register

The weight registers is located at word offset 0, so in this case address CIO3300. The data type is double word, 32 bit signed integer. The weight register selector is located at word offset 3, low byte.

Selector 0x00 = multi range net weigher value. Value is 1156:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	1156	8396	0	8	771	0	0	0	11561
CIO3310	0	11561	0	0	0	1156	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

Selector 0x05 = tare value. Value is 452:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	452	8652	5	8	769	0	0	0	10146
CIO3310	0	5622	0	4524	0	562	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

Read weigher status

The weigher status is located at word offset 2, so in this case address CIO3302.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3302	0	0	1	0	0	0	0	1	1	1	0	0	1	1	0	0	21CC
CIO3303	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005
CIO3304	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0008
CIO3305	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0301

This represents the following status:

Status bit #	Description
2	Stable signal
3	In stable range
6	In zero range
7	Zero tracking possible
8	Tare active
13	User certified operation

Read inputs

The device inputs are located at word offset 4, so in this case address CIO3304. Input 1 - 3 are the physical inputs. Inputs 4 - 16 are virtual inputs generated by the software.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3304	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	000B
CIO3305	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0303
CIO3306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Physical input 1 and 2 are active.

Read outputs

The device outputs are located at word offset 5, so in this case address CIO3305. Output 1 - 4 are the physical outputs. Outputs 5 - 16 are virtual outputs generated by the software.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3305	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0303
CIO3306	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3307	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Physical output 1 and 2 are active.

Read predefined weigher values (indicator only)

The following values are available:

Name	Data type	Word offset
Preset Tare	Double Word, 32 bit signed integer	06 -> CIO3306
Gross x 10	Double Word, 32 bit signed integer/float	08 -> CIO3308
Net x 10	Double Word, 32 bit signed integer/float	10 -> CIO3310
Tare x 10	Double Word, 32 bit signed integer/float	12 -> CIO3312
Multi-range weight	Double Word, 32 bit signed integer/float	14 -> CIO3314

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3306	0	0	0	22941	0	18417	0	4524	0	1842
CIO3316	0	0	0	0	0	0	0	0	0	0
CIO3326	0	0	0	0	0	0	0	0	0	0
CIO3336	0	0	0	0	0	0	0	0	0	0

CIO3306 = Preset Tare 0
CIO3308 = Gross x 10 22941
CIO3310 = Net x 10 18417
CIO3312 = Tare x 10 4524
CIO3314 = Multi-range weight 1842

Read markers (controller only)
The device markers are located at word offset 6 and 7, so in this case address CIO3306 and CIO3307.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3306	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0810
CIO3307	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0008
CIO3308	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

CIO3306 = markers 401…416 marker 5 and 12 are active

CIO3307 = markers 417…432 marker 20 is active

Read extended registers (controller only)
The following extended registers are available:

Name	Data type	Word offset
Ext register 1 - 5 - 9 - 13	Double Word, 32 bit signed integer	08 -> CIO3308
Ext register 2 - 6 - 10 - 14	Double Word, 32 bit signed integer	10 -> CIO3310
Ext register 3 - 7 - 11 - 15	Double Word, 32 bit signed integer	12 -> CIO3312
Ext register 4 - 8 - 12 - 16	Double Word, 32 bit signed integer	14 -> CIO3314

The number depends on the channel set with control bits 6 and 7. The control bits are located at word offset 3, high byte.

Bit 7	Bit 6	Channel
0	0	0
0	1	1
1	0	2
1	1	3

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	2294	8332	16384	11	516	2048	8	0	5555
CIO3310	0	6666	0	7777	0	8888	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

Bit 6 is set to 1 so channel 1 is selected.

CIO3308 = Extended register 5 5555
CIO3310 = Extended register 6 6666
CIO3312 = Extended register 7 7777
CIO3314 = Extended register 8 8888

4.2 Outputs

Control weigher
The control bits are located at word offset 0, high byte, so in this case address CIO3200.

Zero is set at the rising edge of control bit 1:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0200
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Set bit 6 and 7. For an indicator this enables writing the levels. For a controller this selects channel 3 for writing or reading extended registers:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

The register functions are enabled on the rising edge of bit 0 and 1:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0300
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Select weight register
The weight register select bits are located at word offset 0, low byte, so in this case address CIO3200.

Select preset tare = 0x05:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Set preset tare value (indicator only)
The preset tare value is located at word offset 1, so in this case address CIO3201.

Set preset tare value to 200:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	0	200	0	0	0	0	0	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

This value is activated at the rising edge of control bit 4, the preset tare command.

Write levels (indicator only)
The following levels are available:

Name	Data type	Word offset
Level 1	Double Word, 32 bit signed integer	03 -> CIO3203
Level 2	Double Word, 32 bit signed integer	05 -> CIO3205
Level 3	Double Word, 32 bit signed integer	07 -> CIO3207
Level 4	Double Word, 32 bit signed integer	09 -> CIO3209

Control bit 6 and 7 must be set to enable writing the levels:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Set level 1 to 200:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	0	0	0	0	200	0	0	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

Write markers (controller only)
The device markers are located at word offset 1 and 2, so in this case address CIO3201 and CIO3202.

CIO3301 = markers 969…984 set marker 969 and 973:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3201	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0011
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Write extended registers (controller only)
The following extended registers are available:

Name	Data type	Word offset
Ext register 85 - 89 - 93 - 97	Double Word, 32 bit signed integer/float	03 -> CIO3203
Ext register 86 - 90 - 94 - 98	Double Word, 32 bit signed integer/float	05 -> CIO3205
Ext register 87 - 91 - 95 - 99	Double Word, 32 bit signed integer/float	07 -> CIO3207
Ext register 88 - 92 - 96 - 100	Double Word, 32 bit signed integer/float	09 -> CIO3209

The number depends on the channel set with control bits 6 and 7. The control bits are located at word offset 0, high byte.

Bit 7	Bit 6	Channel
0	0	0
0	1	1
1	0	2
1	1	3

Set channel to 2 (control bit 7) to set extended register 43 to value 200 and 44 to value 300:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	32768	0	0	0	200	0	300	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

4.3 Register functions

To enable the register function mode, set control bit 0 and 1 at the same time:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3200	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0300
CIO3201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3202	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
CIO3203	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Status bit 15 indicates the register function mode is active:

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex
CIO3302	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000
CIO3303	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0300
CIO3304	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004
CIO3305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000

Reset control bit 0 or 1 to disable the register function mode and return to normal mode.

Use the following registers:

Parameter	Data type	Word offset
Input - parameter 1	Double Word	03 -> CIO3203
Input - parameter 2	Double Word	05 -> CIO3205
Input - parameter 3	Double Word	07 -> CIO3207
Input - parameter 4	Double Word	09 -> CIO3209
Output - result 1	Double Word	08 -> CIO3308
Output - result 2	Double Word	10 -> CIO3310
Output - result 3	Double Word	12 -> CIO3312
Output - result 4	Double Word	14 -> CIO3314

Calibrate zero

The function code is 1. The other parameters are not used.

Request:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	768	0	0	0	1	0	0	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

Reply:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	0	24812	768	8	0	0	0	0	1
CIO3310	0	0	0	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

Calibrate span
The function code is 2. Parameter 2 is used for the span weight. Calibrate span at 2000:

Request:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	768	0	0	0	2	0	2000	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

Reply:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	2000	24716	768	8	0	16	0	0	2
CIO3310	0	0	0	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

In case an error occurs (for example calibrate span without entering a value) the error code is shown in the high word of result register 1:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	0	24812	768	8	0	0	0	2108	2
CIO3310	0	0	0	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

In this case the error is 2108: *The gain reference is smaller than the zero reference.*

Get indicator max load
Get the indicator maximum load. The function code is 102. The value is returned in reply register 2.

Request:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	768	0	0	0	102	0	0	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

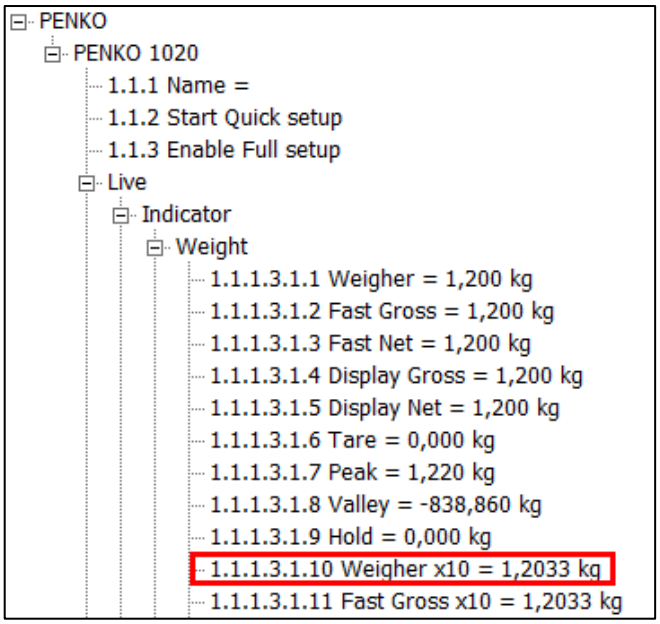
Reply:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	0	24812	768	8	0	16	0	0	102
CIO3310	0	10008	0	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

The max load is 10008.

Get weigher x 10 value using PDI
PDI (PENKO Device Interface) represents the device configuration in a tree structure. Every property has its own unique path number. The tree is used in the PENKO configuration tools Pi Mach II and PDI Client, both available at www.penko.com/software

Screenshot of PENKO Pi Mach II software with 1020 PDI configuration:



Weigher x 10 has path number 1.1.1.3.1.10. This results in the following input parameters:

Fill the path with zeros to make 12 numbers: 1.1.1.3.1.10.0.0.0.0.0.0

Parameter 1	PDI_PATH_SET	201
Parameter 2	0x01 0x01 0x01 0x03	16843011
Parameter 3	0x01 0x0A 0x00 0x00	17432576
Parameter 4	0x00 0x00 0x00 0x00	0

Request:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	768	0	0	0	201	257	259	266	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

Reply:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	1201	41164	768	8	771	0	0	0	201
CIO3310	257	259	266	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

The PDI path is now set. If the path is not found, all zeros are returned.

Use function code 203 to get the PDI property at the selected path:

Request:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3200	768	0	0	0	203	0	0	0	0	0
CIO3210	0	0	0	0	0	0	0	0	0	0
CIO3220	0	0	0	0	0	0	0	0	0	0
CIO3230	0	0	0	0	0	0	0	0	0	0

Reply:

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
CIO3300	0	364	24780	768	8	0	16	0	0	203
CIO3310	0	3644	0	0	0	0	0	0	0	0
CIO3320	0	0	0	0	0	0	0	0	0	0
CIO3330	0	0	0	0	0	0	0	0	0	0

The weigher x 10 value is 3644.

About PENKO

At PENKO Engineering we specialize in weighing. Weighing is inherently chemically correct, independent of consistency, type or temperature of the raw material. This means that weighing any kind of material guaranties consistency and thus, it is essential to sustainable revenue generation in any industry. As a well-established and proven solution provider, we strive for the ultimate satisfaction of custom design and/or standard applications, increasing your efficiencies and saving you time, saving you money.

Whether we are weighing raw materials, components in batching, ingredients for mixing or dosing processes, - or weighing of static containers and silos, or - in-motion weighing of railway wagons or trucks, by whatever means required during a process, we are essentially forming vital linkages between processes and businesses, anywhere at any time. We design, develop and manufacture state of the art technologically advanced systems in accordance with your strategy and vision. From the initial design brief, we take a fresh approach and a holistic view of every project, managing, supporting and/or implementing your system every step of the way. Curious to know how we do it? www.penko.com

Certifications

PENKO sets high standards for its products and product performance which are tested, certified and approved by independent expert and government organizations to ensure they meet – and even – exceed metrology industry guidelines. A library of testing certificates is available for reference on:
http://penko.com/nl/publications_certificates.html

PENKO Professional Services

PENKO is committed to ensuring every system is installed, tested, programmed, commissioned and operational to client specifications. Our engineers, at our weighing center in Ede, Netherlands, as well as our distributors around the world, strive to solve most weighing-system issues within the same day. On a monthly basis PENKO offers free training classes to anyone interested in exploring modern, high-speed weighing instruments and solutions. Training sessions on request: **www.penko.com/training**

PENKO Alliances

PENKO's worldwide network: Australia, Brazil, China, Denmark, Germany, Egypt, Finland, France, India, Italy, Netherlands, Norway, Poland, Portugal, Slovakia, Spain, Syria, Turkey, United Kingdom, South Africa, Slovakia Sweden and Switzerland, Singapore.
A complete overview you will find on: **www.penko.com/dealers**

PENKO Engineering B.V. ▪ Schutterweg 35, NL 6718XC Ede ▪ Tel +31 (0) 318525630 ▪ Fax +31 (0) 31852971 ▪ info@penko.com
Web ▪ www.penko.com ▪ 7600L1096 EN PENKO 1020 Profibus Protocol R4