PENKO Engineering B.V.

Your Partner for Fully Engineered Factory Solutions

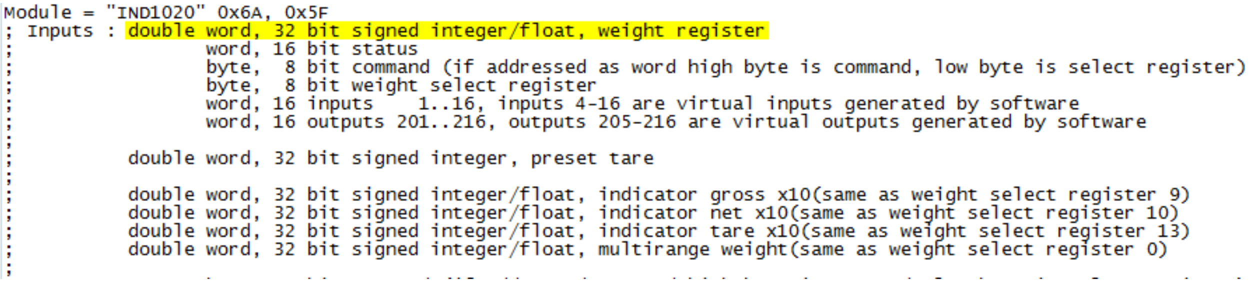
How to…
Calibrate a SGM740 or SGM840 with Profibus

PENKO
an ETC Company

Table of Contents

Read indicator weight

The weight of the indicator is located at the beginning of the inputs.



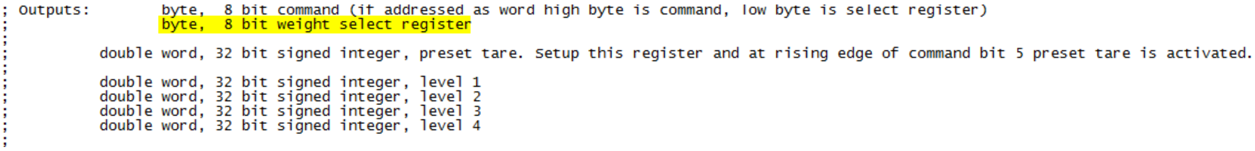
```
Module = "IND1020" 0x6A, 0x5F
; Inputs : double word, 32 bit signed integer/float, weight register
;                    word, 16 bit status
;                    byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                    byte,  8 bit weight select register
;                    word, 16 inputs    1..16, inputs 4-16 are virtual inputs generated by software
;                    word, 16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
;
;            double word, 32 bit signed integer, preset tare
;
;            double word, 32 bit signed integer/float, indicator gross x10(same as weight select register 9)
;            double word, 32 bit signed integer/float, indicator net x10(same as weight select register 10)
;            double word, 32 bit signed integer/float, indicator tare x10(same as weight select register 13)
;            double word, 32 bit signed integer/float, multirange weight(same as weight select register 0)
;
```

Change the weight indicator

On default the indicator will show the “display weight including the multi range interval step”, but it’s also possible to select a different weight value.

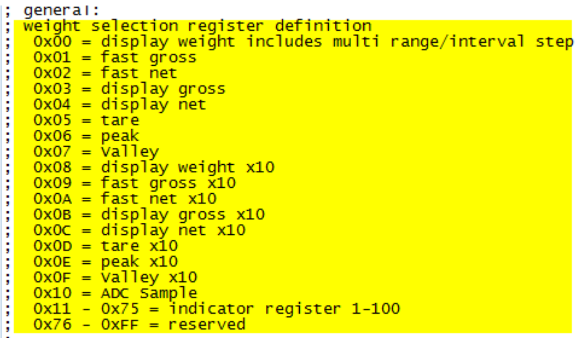
In the example below we will choose the “Display net”

Use the “weight select register” to select a weight value.



```
; Outputs:        byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;
;         double word, 32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
;
;         double word, 32 bit signed integer, level 1
;         double word, 32 bit signed integer, level 2
;         double word, 32 bit signed integer, level 3
;         double word, 32 bit signed integer, level 4
;
```

Choose which value you want below and set the hex value in “weight select register”.

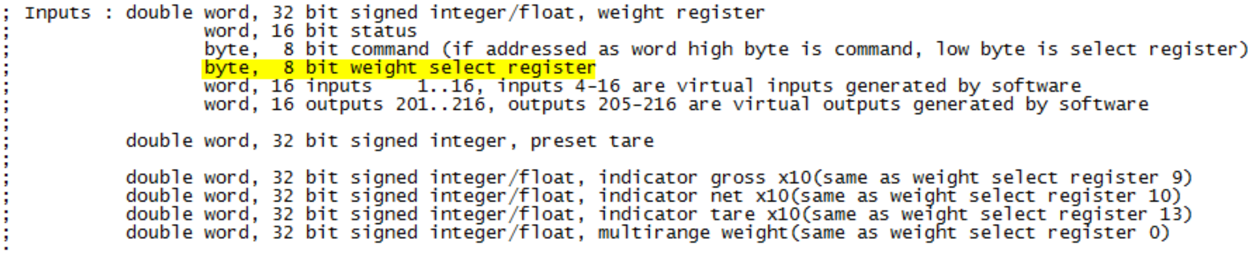


```
; general:
; weight selection register definition
;  0x00 = display weight includes multi range/interval step
;  0x01 = fast gross
;  0x02 = fast net
;  0x03 = display gross
;  0x04 = display net
;  0x05 = tare
;  0x06 = peak
;  0x07 = valley
;  0x08 = display weight x10
;  0x09 = fast gross x10
;  0x0A = fast net x10
;  0x0B = display gross x10
;  0x0C = display net x10
;  0x0D = tare x10
;  0x0E = peak x10
;  0x0F = valley x10
;  0x10 = ADC Sample
;  0x11 - 0x75 = indicator register 1-100
;  0x76 - 0xFF = reserved
```

To select “Display net”, set “4” in “weight select register”.

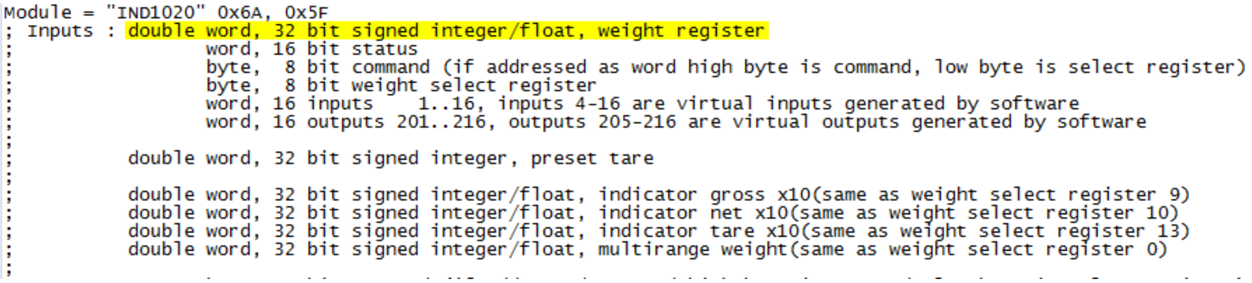
Read the weight select register

To make sure that we are looking at the correct weight, it's possible to readout the "weight select register". The "weight select register" should read "4" meaning that the "Display net" is chosen.



```
; Inputs : double word, 32 bit signed integer/float, weight register
;                 word, 16 bit status
;                 byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;                 word, 16 inputs    1..16, inputs 4-16 are virtual inputs generated by software
;                 word, 16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
;
;          double word, 32 bit signed integer, preset tare
;
;          double word, 32 bit signed integer/float, indicator gross x10(same as weight select register 9)
;          double word, 32 bit signed integer/float, indicator net x10(same as weight select register 10)
;          double word, 32 bit signed integer/float, indicator tare x10(same as weight select register 13)
;          double word, 32 bit signed integer/float, multirange weight(same as weight select register 0)
;
```

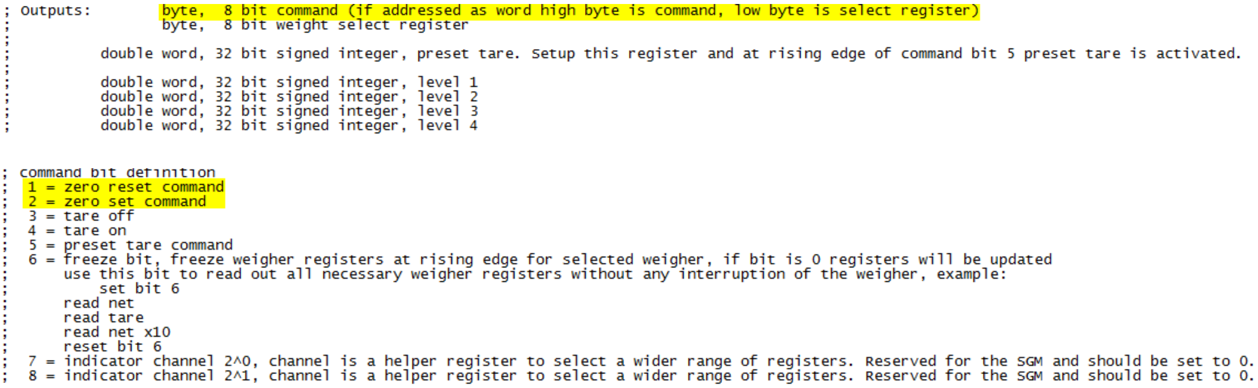
The weight register will now show the "Display net"



```
Module = "IND1020" 0x6A, 0x5F
; Inputs : double word, 32 bit signed integer/float, weight register
;                 word, 16 bit status
;                 byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;                 word, 16 inputs    1..16, inputs 4-16 are virtual inputs generated by software
;                 word, 16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
;
;          double word, 32 bit signed integer, preset tare
;
;          double word, 32 bit signed integer/float, indicator gross x10(same as weight select register 9)
;          double word, 32 bit signed integer/float, indicator net x10(same as weight select register 10)
;          double word, 32 bit signed integer/float, indicator tare x10(same as weight select register 13)
;          double word, 32 bit signed integer/float, multirange weight(same as weight select register 0)
;
```

Enter “Function mode”

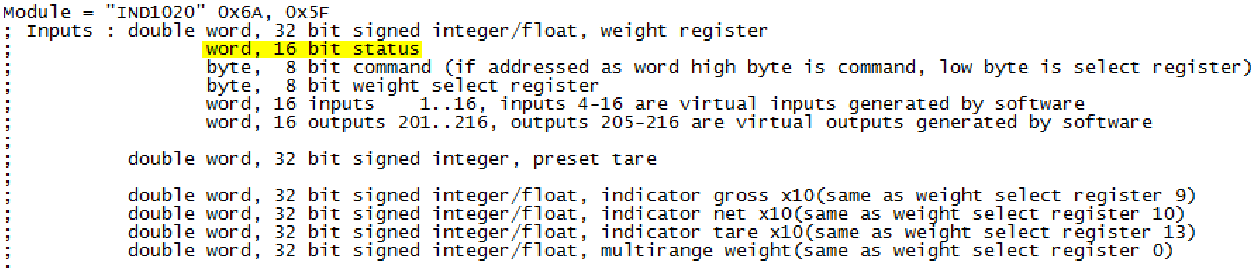
To enter the “Function mode” command bit 1 and 2 must be set high at the same time.



```
; Outputs:        byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;
;          double word, 32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
;
;          double word, 32 bit signed integer, level 1
;          double word, 32 bit signed integer, level 2
;          double word, 32 bit signed integer, level 3
;          double word, 32 bit signed integer, level 4


; command bit definition
;  1 = zero reset command
;  2 = zero set command
;  3 = tare off
;  4 = tare on
;  5 = preset tare command
;  6 = freeze bit, freeze weigher registers at rising edge 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, example:
;          set bit 6
;      read net
;      read tare
;      read net x10
;      reset bit 6
;  7 = indicator channel 2^0, channel is a helper register to select a wider range of registers. Reserved for the SGM and should be set to 0.
;  8 = indicator channel 2^1, channel is a helper register to select a wider range of registers. Reserved for the SGM and should be set to 0.
```

The status will show if “Function mode” is activated.

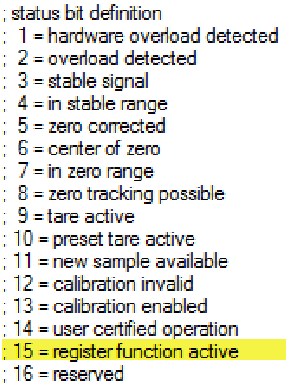


```
Module = "IND1020" 0x6A, 0x5F
; Inputs : double word, 32 bit signed integer/float, weight register
;                 word, 16 bit status
;                 byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;                 word, 16 inputs    1..16, inputs 4-16 are virtual inputs generated by software
;                 word, 16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
;
;          double word, 32 bit signed integer, preset tare
;
;          double word, 32 bit signed integer/float, indicator gross x10(same as weight select register 9)
;          double word, 32 bit signed integer/float, indicator net x10(same as weight select register 10)
;          double word, 32 bit signed integer/float, indicator tare x10(same as weight select register 13)
;          double word, 32 bit signed integer/float, multirange weight(same as weight select register 0)
;
```

Bit 15 will indicate if “Function mode” is activated.

If bit 15 is high, Function mode is activated.

If bit 15 is low, normal operation is activated.



```
; status bit definition
;  1 = hardware overload detected
;  2 = overload detected
;  3 = stable signal
;  4 = in stable range
;  5 = zero corrected
;  6 = center of zero
;  7 = in zero range
;  8 = zero tracking possible
;  9 = tare active
; 10 = preset tare active
; 11 = new sample available
; 12 = calibration invalid
; 13 = calibration enabled
; 14 = user certified operation
; 15 = register function active
; 16 = reserved
```

Calibrate zero

Calibrate zero can only be done when in “Function mode”

Send request

To calibrate zero, set value 1 in “write extended register n+0” (low word). The moment that the value 1 is set in the register, the zero calibration is executed.

Input parameters
Parameter 1 CAL_ZERO(=1)
Parameter 2 not used
Parameter 3 not used
Parameter 4 not used

Profibus outputs

Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	Byte 8 bit weight select register
Double word	32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
Double word	32 bit signed integer, if bit #7 + #8 is set write level 1 write extended register n+0
Double word	32 bit signed integer, if bit #7 + #8 is set write level 2 write extended register n+1
Double word	32 bit signed integer, if bit #7 + #8 is set write level 3 write extended register n+2
Double word	32 bit signed integer, if bit #7 + #8 is set write level 4 write extended register n+3

Receive request

The following data will be received.

Output parameters
Result 1 CAL_ZERO
Result 2 not used
Result 3 not used
Result 4 not used

The 4 parameters can be read out in the highlighted extended registers listed below.

Profibus inputs

Double word	32 bit signed integer/float, weight register
Word	16 bit status field
Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	8 bit weight select register
Word	16 inputs 1..16, inputs 4-16 are virtual inputs generated by software
Word	16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
Double word	32 bit signed integer, preset tare
Double word	32 bit signed integer/float, indicator gross x10(same as weight select register 9) read extended register m+0 in register function mode
Double word	32 bit signed integer/float, indicator net x10(same as weight select register 10) read extended register m+1 in register function mode
Double word	32 bit signed integer/float, indicator tare x10(same as weight select register 13) read extended register m+2 in register function mode
Double word	32 bit signed integer/float, (multi-range) mweight(same as weight select register 0) read extended register m+3 in register function mode

When calibrate zero is finished, send 0 in “write extended register n+0”

Calibrate Span

Place the weight on the scale, in the example below we use 500,0kg

Send request

To calibrate span, first set the weight (5000) in parameter 2 (extended register n+1). After that set value 2 in "write extended register n+0" (low word). The moment that the value 2 is set in the register, the span calibration is executed.

Input parameters
Parameter 1 CAL_SPAN(=2)
Parameter 2 Span weight (5000)
Parameter 2 not used
Parameter 3 not used

Profibus outputs

Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	Byte 8 bit weight select register
Double word	32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
Double word	32 bit signed integer, if bit #7 + #8 is set write level 1 write extended register n+0
Double word	32 bit signed integer, if bit #7 + #8 is set write level 2 write extended register n+1
Double word	32 bit signed integer, if bit #7 + #8 is set write level 3 write extended register n+2
Double word	32 bit signed integer, if bit #7 + #8 is set write level 4 write extended register n+3

Receive request

The following data will be received.

Output parameters
Result 1 CAL_SPAN (=2)
Result 2 not used
Result 3 not used
Result 4 not used

The 4 parameters can be read out in the highlighted extended registers listed below.

Profibus inputs

Double word	32 bit signed integer/float, weight register
Word	16 bit status field
Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	8 bit weight select register
Word	16 inputs 1..16, inputs 4-16 are virtual inputs generated by software
Word	16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
Double word	32 bit signed integer, preset tare
Double word	32 bit signed integer/float, indicator gross x10(same as weight select register 9) read extended register m+0 in register function mode
Double word	32 bit signed integer/float, indicator net x10(same as weight select register 10) read extended register m+1 in register function mode
Double word	32 bit signed integer/float, indicator tare x10(same as weight select register 13) read extended register m+2 in register function mode
Double word	32 bit signed integer/float, (multi-range) mweight(same as weight select register 0) read extended register m+3 in register function mode

When calibrate span is finished, send 0 in "write extended register n+0"

Profibus outputs

Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	Byte 8 bit weight select register
Double word	32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
Double word	32 bit signed integer, if bit #7 + #8 is set write level 1 write extended register n+0
Double word	32 bit signed integer, if bit #7 + #8 is set write level 2 write extended register n+1
Double word	32 bit signed integer, if bit #7 + #8 is set write level 3 write extended register n+2
Double word	32 bit signed integer, if bit #7 + #8 is set write level 4 write extended register n+3

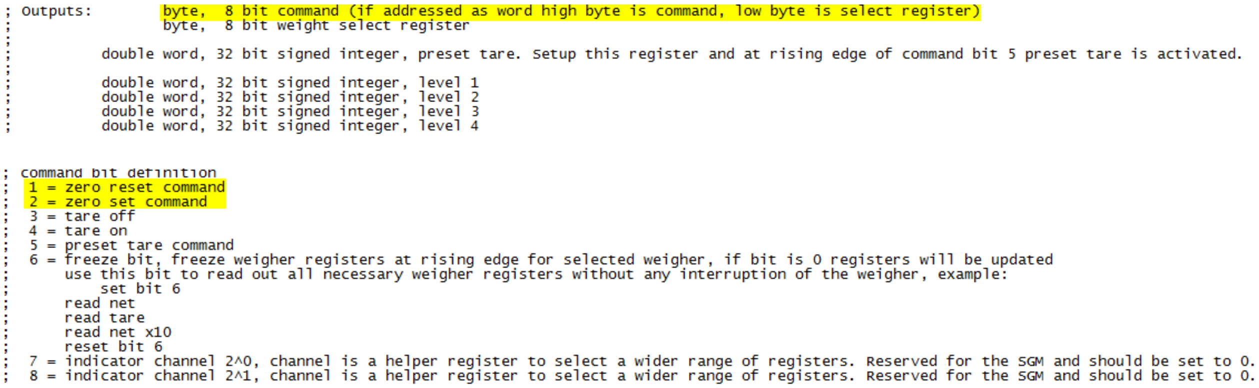
Now 0 will show in "read extended register m+0".

Profibus inputs

Double word	32 bit signed integer/float, weight register
Word	16 bit status field
Byte	8 bit command (if addressed as word high byte is command, low byte is select register)
Byte	8 bit weight select register
Word	16 inputs 1..16, inputs 4-16 are virtual inputs generated by software
Word	16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
Double word	32 bit signed integer, preset tare
Double word	32 bit signed integer/float, indicator gross x10(same as weight select register 9) read extended register m+0 in register function mode
Double word	32 bit signed integer/float, indicator net x10(same as weight select register 10) read extended register m+1 in register function mode
Double word	32 bit signed integer/float, indicator tare x10(same as weight select register 13) read extended register m+2 in register function mode
Double word	32 bit signed integer/float, (multi-range) mweight(same as weight select register 0) read extended register m+3 in register function mode

Leave “Function mode”

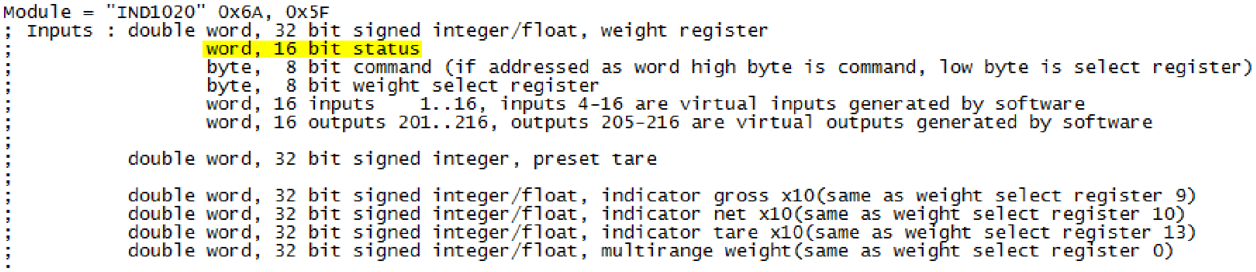
To leave the “Function mode” reset command bit 1 and 2.



```
; outputs:        byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                 byte,  8 bit weight select register
;
;         double word, 32 bit signed integer, preset tare. Setup this register and at rising edge of command bit 5 preset tare is activated.
;
;         double word, 32 bit signed integer, level 1
;         double word, 32 bit signed integer, level 2
;         double word, 32 bit signed integer, level 3
;         double word, 32 bit signed integer, level 4


; command bit definition
;  1 = zero reset command
;  2 = zero set command
;  3 = tare off
;  4 = tare on
;  5 = preset tare command
;  6 = freeze bit, freeze weigher registers at rising edge 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, example:
;          set bit 6
;      read net
;      read tare
;      read net x10
;      reset bit 6
;  7 = indicator channel 2^0, channel is a helper register to select a wider range of registers. Reserved for the SGM and should be set to 0.
;  8 = indicator channel 2^1, channel is a helper register to select a wider range of registers. Reserved for the SGM and should be set to 0.
```

The status will show if “Function mode” is activated.

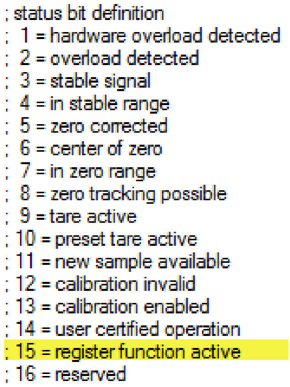


```
Module = "IND1020" 0x6A, 0x5F
; Inputs : double word, 32 bit signed integer/float, weight register
;                word, 16 bit status
;                byte,  8 bit command (if addressed as word high byte is command, low byte is select register)
;                byte,  8 bit weight select register
;                word, 16 inputs    1..16, inputs 4-16 are virtual inputs generated by software
;                word, 16 outputs 201..216, outputs 205-216 are virtual outputs generated by software
;
;         double word, 32 bit signed integer, preset tare
;
;         double word, 32 bit signed integer/float, indicator gross x10(same as weight select register 9)
;         double word, 32 bit signed integer/float, indicator net x10(same as weight select register 10)
;         double word, 32 bit signed integer/float, indicator tare x10(same as weight select register 13)
;         double word, 32 bit signed integer/float, multirange weight(same as weight select register 0)
;
```

Bit 15 will indicate if “Function mode” is activated.

If bit 15 is high, Function mode is activated.

If bit 15 is low, normal operation is activated.



```
; status bit definition
;  1 = hardware overload detected
;  2 = overload detected
;  3 = stable signal
;  4 = in stable range
;  5 = zero corrected
;  6 = center of zero
;  7 = in zero range
;  8 = zero tracking possible
;  9 = tare active
; 10 = preset tare active
; 11 = new sample available
; 12 = calibration invalid
; 13 = calibration enabled
; 14 = user certified operation
; 15 = register function active
; 16 = reserved
```

About PENKO

Our design expertise include systems for manufacturing plants, bulk weighing, check weighing, force measuring and process control. For over 35 years, PENKO Engineering B.V. has been at the forefront of development and production of high-accuracy, high-speed weighing systems and our solutions continue to help cut costs, increase ROI and drive profits for some of the largest global brands, such as Cargill, Sara Lee, Heinz, Kraft Foods and Unilever to name but a few.

Whether you are looking for a simple stand-alone weighing system or a high-speed weighing and dosing controller for a complex automated production line, PENKO has a comprehensive range of standard solutions you can rely on.

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. A schedule of training sessions is found on: **www.penko.com/training**

PENKO Alliances

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