PENKO Engineering B.V.

Your Partner for Fully Engineered Factory Solutions

Protocol description:
PENKO Buslink

PENKO
an ETC Company

PENKO Buslink protocol

Table of Contents

PENKO Buslink protocol

Introduction

The PENKO Buslink protocol is used to connect a PENKO device (controller, indicator or digitizer) to a PENKO FLEX controller and supports the following features:

- Read Indicator Data
- Read inputs/outputs/markers

By connecting a PENKO RIO700 or RIA700, the amount of digital and analog I/O of the PENKO FLEX controller can be extended

The Buslink protocol is supported over Ethernet:

- Ethernet Buslink (EBL)

When a device supports both interfaces, only one interface can be used at the time.

PENKO Buslink protocol

1 Basics

The PENKO devices that support the protocol and the used settings and addressing.

1.1 Devices

The following PENKO devices support the Buslink protocol:

Device Series	Buslink support
SGM7xx series	720 Ethernet Buslink || 730 CAN Buslink
SGM8xx series	820 Ethernet Buslink || 830 CAN Buslink
RIO700/RIA700	Ethernet Buslink || CAN Buslink
1020 series	Ethernet Buslink || CAN Buslink (optional)
Flex 2100	Ethernet Buslink || CAN Buslink
Flex	Ethernet Buslink || CAN Buslink
Flex Multichannel	Ethernet Buslink || CAN Buslink

1.2 Settings

Ethernet Buslink has no specific settings. CAN Buslink supports the following baud rates:

Baud rate
100k
125k
250k
500k

1.3 Addressing

Buslink works with 8 addresses and 5 sub addresses resulting in a total of 40 unique addresses.

		Sub addresses				
		1	2	3	4	5
Addresses	1	1-1	1-2	1-3	1-4	1-5
	2	2-1	2-2	2-3	2-4	2-5
	3	3-1	3-2	3-3	3-4	3-5
	4	4-1	4-2	4-3	4-4	4-5
	5	5-1	5-2	5-3	5-4	5-5
	6	6-1	6-2	6-3	6-4	6-5
	7	7-1	7-2	7-3	7-4	7-5
	8	8-1	8-2	8-3	8-4	8-5

2 Connecting

Connect the devices using Ethernet or CANbus.

2.1 Ethernet

Make sure the devices are in the same network and have a unique IP address and Buslink address.

Example:

Device	IP address	Subnet	Buslink address	Buslink sub address
FLEX	10.1.2.4	255.255.255.0	1	1
1020	10.1.2.5	255.255.255.0	1	2
SGM720	10.1.2.6	255.255.255.0	1	3
RIO700	10.1.2.7	255.255.255.0	1	4

3 Usage

How to use and setup the Buslink protocol for the various devices.

3.1 Extend digital I/O with RIO700

Connect a RIO700 to the FLEX over CAN or Ethernet.

FLEX Buslink address: 1 - 1

RIO700 Buslink address: 1 - 2

Consult the FLEX mapping in chapter 3.4:

The RIO700 is on FLEX address 1 - 2 so the 8 RIO700 inputs are mapped to FLEX inputs 1041 to 1048.

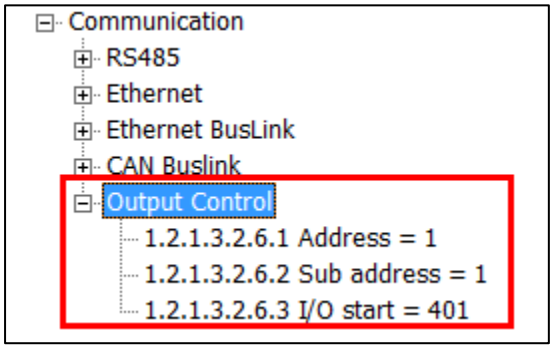
The 8 RIO700 outputs are mapped to FLEX outputs 1241 to 1248.

The RIO700 outputs are triggered by markers on the FLEX. Therefore a marker start address has to be set in the RIO700.

Up to 5 RIO700 devices can be controlled:

RIO700 I/O start	FLEX markers mapped to RIO700 outputs
401	401 - 408
409	409 - 416
417	417 - 424
425	425 - 432
433	433 - 440

The markers are always grouped by 8. Selecting I/O start 403 for example will still use marker 401 - 408.



This RIO700 setting means the RIO700 will listen to address 1 -1 (the FLEX) and the 8 outputs react on FLEX marker 401 - 408.

3.2 Extend analog I/O with RIA700

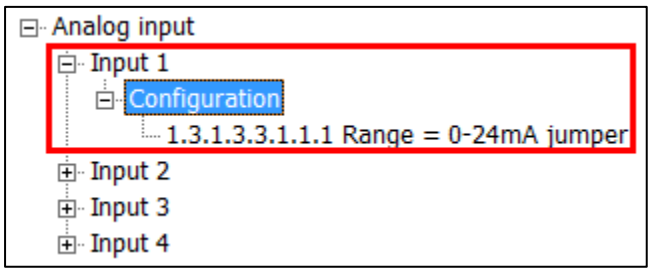
Connect a RIA700 to the FLEX over CAN or Ethernet.

FLEX Buslink address: 1 - 1

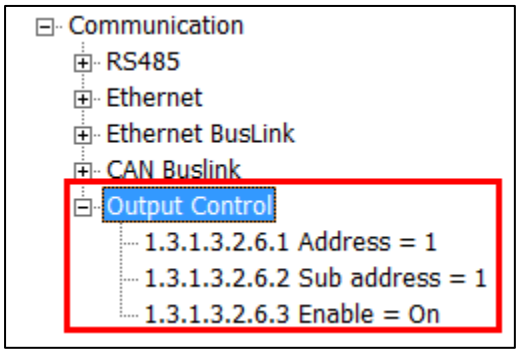
RIA700 Buslink address: 1 - 2

Consult the FLEX mapping in chapter 3.4:

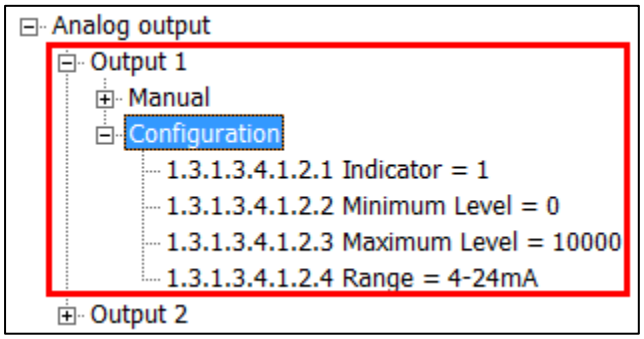
The 4 RIA700 inputs are mapped to FLEX indicators 116 - 119. The range of each RIA700 input can be configured:



The 2 RIA700 outputs react on FLEX indicators. To enable this set the RIA700 output control to the address of the FLEX and set Enable to ON:



Configure the FLEX indicator number, levels and range for each RIA700 output:



3.3 Connect an indicator or controller

Connect a PENKO indicator or controller to the FLEX over CAN or Ethernet.

FLEX Buslink address: 1 - 1

PENKO device Buslink address: 1 - 2

Consult the FLEX mapping in chapter 3.4:

The indicators are mapped to FLEX indicators 116 - 130.

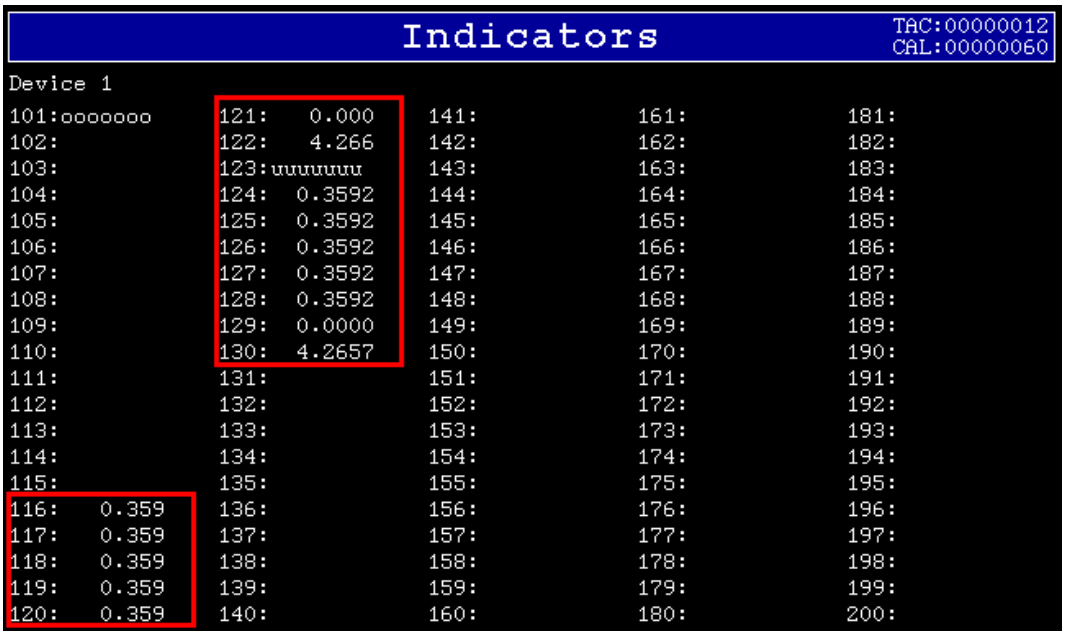
The inputs are mapped to FLEX inputs 1041 - 1080.

The outputs are mapped to FLEX outputs 1241 - 1280.

The markers are mapped to FLEX markers 1441 - 1480.

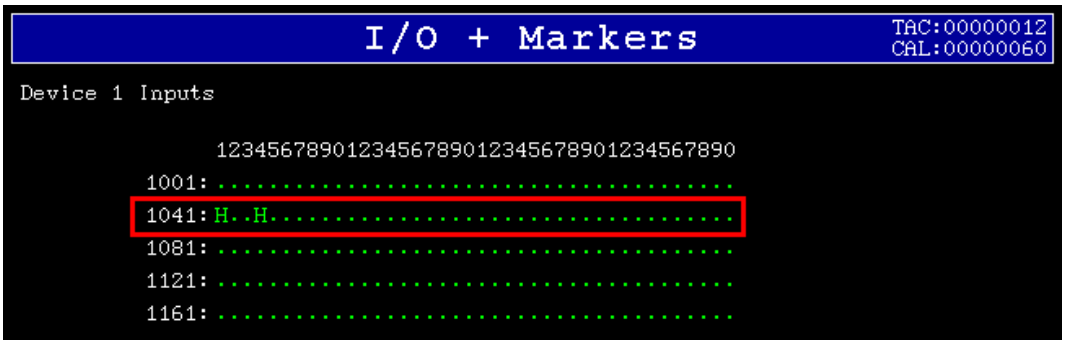
FLEX -> Menu -> Status

Indicators example:

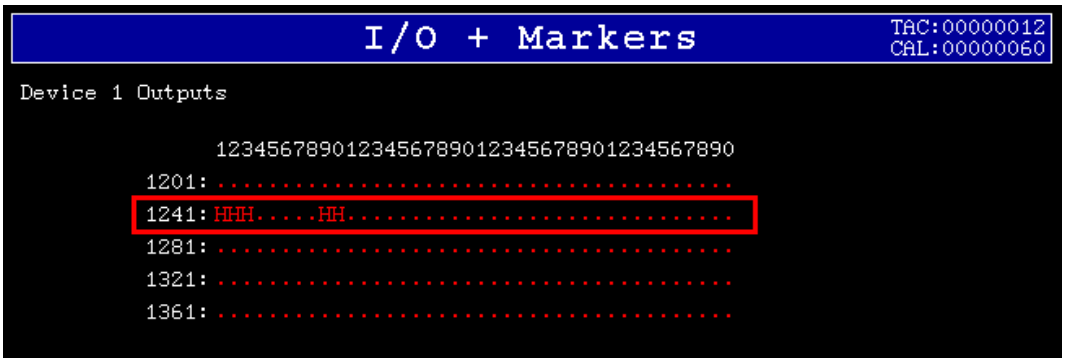


PENKO Buslink protocol

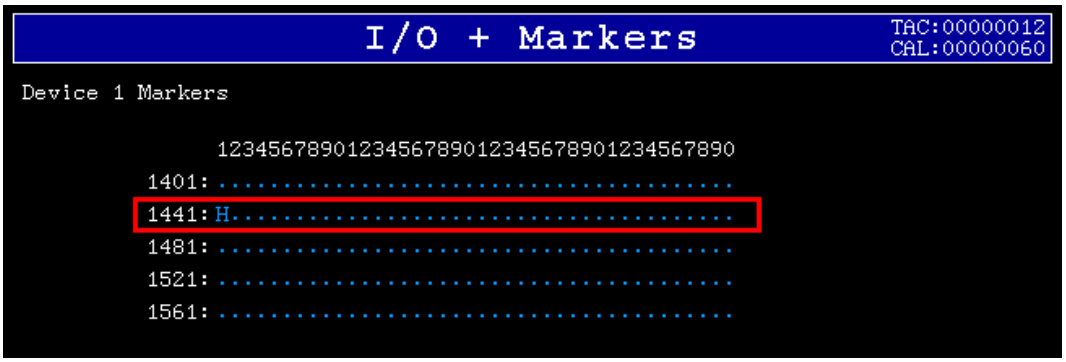
Inputs example:



Outputs example:



Markers example:



3.4 FLEX mapping

Buslink		I/O/M/Ind	Mapping	
1	1	Inputs	1001	1040
		Outputs	1201	1240
		Markers	1401	1440
		Indicators	101	115
1	2	Inputs	1041	1080
		Outputs	1241	1280
		Markers	1441	1480
		Indicators	116	130
1	3	Inputs	1081	1120
		Outputs	1281	1320
		Markers	1481	1520
		Indicators	131	145
1	4	Inputs	1121	1160
		Outputs	1321	1360
		Markers	1521	1560
		Indicators	146	160
1	5	Inputs	1161	1200
		Outputs	1361	1400
		Markers	1561	1600
		Indicators	161	175

Buslink		I/O/M/Ind	Mapping	
2	1	Inputs	2001	2040
		Outputs	2201	2240
		Markers	2401	2440
		Indicators	201	215
2	2	Inputs	2041	2080
		Outputs	2241	2280
		Markers	2441	2480
		Indicators	216	230
2	3	Inputs	2081	2120
		Outputs	2281	2320
		Markers	2481	2520
		Indicators	231	245
2	4	Inputs	2121	2160
		Outputs	2321	2360
		Markers	2521	2560
		Indicators	246	260
2	5	Inputs	2161	2200
		Outputs	2361	2400
		Markers	2561	2600
		Indicators	261	275

Buslink		I/O/M/Ind	Mapping	
3	1	Inputs	3001	3040
		Outputs	3201	3240
		Markers	3401	3440
		Indicators	301	315
3	2	Inputs	3041	3080
		Outputs	3241	3280
		Markers	3441	3480
		Indicators	316	330
3	3	Inputs	3081	3120
		Outputs	3281	3320
		Markers	3481	3520
		Indicators	331	345
3	4	Inputs	3121	3160
		Outputs	3321	3360
		Markers	3521	3560
		Indicators	346	360
3	5	Inputs	3161	3200
		Outputs	3361	3400
		Markers	3561	3600
		Indicators	361	375

Buslink		I/O/M/Ind	Mapping	
4	1	Inputs	4001	4040
		Outputs	4201	4240
		Markers	4401	4440
		Indicators	401	415
4	2	Inputs	4041	4080
		Outputs	4241	4280
		Markers	4441	4480
		Indicators	416	430
4	3	Inputs	4081	4120
		Outputs	4281	4320
		Markers	4481	4520
		Indicators	431	445
4	4	Inputs	4121	4160
		Outputs	4321	4360
		Markers	4521	4560
		Indicators	446	460
4	5	Inputs	4161	4200
		Outputs	4361	4400
		Markers	4561	4600
		Indicators	461	475

Buslink		I/O/M/Ind	Mapping	
5	1	Inputs	5001	5040
		Outputs	5201	5240
		Markers	5401	5440
		Indicators	501	515
5	2	Inputs	5041	5080
		Outputs	5241	5280
		Markers	5441	5480
		Indicators	516	530
5	3	Inputs	5081	5120
		Outputs	5281	5320
		Markers	5481	5520
		Indicators	531	545
5	4	Inputs	5121	5160
		Outputs	5321	5360
		Markers	5521	5560
		Indicators	546	560
5	5	Inputs	5161	5200
		Outputs	5361	5400
		Markers	5561	5600
		Indicators	561	575

Buslink		I/O/M/Ind	Mapping	
6	1	Inputs	6001	6040
		Outputs	6201	6240
		Markers	6401	6440
		Indicators	601	615
6	2	Inputs	6041	6080
		Outputs	6241	6280
		Markers	6441	6480
		Indicators	616	630
6	3	Inputs	6081	6120
		Outputs	6281	6320
		Markers	6481	6520
		Indicators	631	645
6	4	Inputs	6121	6160
		Outputs	6321	6360
		Markers	6521	6560
		Indicators	646	660
6	5	Inputs	6161	6200
		Outputs	6361	6400
		Markers	6561	6600
		Indicators	661	675

Buslink		I/O/M/Ind	Mapping	
7	1	Inputs	7001	7040
		Outputs	7201	7240
		Markers	7401	7440
		Indicators	701	715
7	2	Inputs	7041	7080
		Outputs	7241	7280
		Markers	7441	7480
		Indicators	716	730
7	3	Inputs	7081	7120
		Outputs	7281	7320
		Markers	7481	7520
		Indicators	731	745
7	4	Inputs	7121	7160
		Outputs	7321	7360
		Markers	7521	7560
		Indicators	746	760
7	5	Inputs	7161	7200
		Outputs	7361	7400
		Markers	7561	7600
		Indicators	761	775

Buslink		I/O/M/Ind	Mapping	
8	1	Inputs	8001	8040
		Outputs	8201	8240
		Markers	8401	8440
		Indicators	801	815
8	2	Inputs	8041	8080
		Outputs	8241	8280
		Markers	8441	8480
		Indicators	816	830
8	3	Inputs	8081	8120
		Outputs	8281	8320
		Markers	8481	8520
		Indicators	831	845
8	4	Inputs	8121	8160
		Outputs	8321	8360
		Markers	8521	8560
		Indicators	846	860
8	5	Inputs	8161	8200
		Outputs	8361	8400
		Markers	8561	8600
		Indicators	861	875

4 Technical implementation

How the Buslink protocol is implemented.

4.1 CAN baud rates

The following baud rates are supported:

Baud rate
100k
125k
250k
500k

4.2 Identifier + Address

The PENKO CAN Buslink is an auto transmitter. Every instrument has its own address and identifier.

The CAN base identifier is 0x15550000. The frame data type Y and address is added to the base identifier.

The address is built in two parts; a “Base address” and a “Sub address”. The formula for the Identifier is as follows:

0x15550000 + 5x (“Base address”-1) + “Sub address”

Base address	Sub address	Identifier
1	1	0x15550Y01
1	2	0x15550Y02
1	3	0x15550Y03
1	4	0x15550Y04
1	5	0x15550Y05
2	1	0x15550Y06
2	2	0x15550Y07
2	3	0x15550Y08
2	4	0x15550Y09
2	5	0x15550Y0A
3	1	0x15550Y0B
:	:	0x15550Yxx
:	:	0x15550Yxx
:	:	0x15550Yxx
8	4	0x15550Y27
8	5	0x15550Y28

4.3 Identifier type Y

Y	RDA Data[1][2][3][4]	RDB Data[5][6][7][8]
0	Inputs	Markers
1	Markers	Outputs
2	Indicator 1	Indicator 2
3	Indicator 3	Indicator 4
4	Indicator 5	Indicator 6
5	Indicator 7	Indicator 8
6	Indicator 9	Indicator 10
7	Indicator 11	Indicator 12
8	Indicator 13	Indicator 14
9	Indicator 15	0

Not all Inputs, Outputs or Markers are available on each device.

4.4 Identifier Examples

Base address	Sub address	Frame type Y	Identifier
1	1	Y=4, indicator 5 + 6	0x15550401
1	4	Y=0, Inputs + Markers	0x15550004
2	4	Y=8, indicator 13 +14	0x15550809
8	6	Y=9, indicator 15	0x15550927

4.5 Frame type Y detail

Y=0	Bit 32-25	Bit 24-17	Bit 16-9	Bit 8-1
RDA	Data[4]	Data[3]	Data[2]	Data[1]
	Input 25-32	Input 17-24	Input 9 -16	Input 1 -8
RDB	Data[8]	Data[7]	Data[6]	Data[5]
	Marker 417-424	Marker 409-416	Marker 401-408	Input 33 -40

Y=1	Bit 32-25	Bit 24-17	Bit 16-9	Bit 8-1
RDA	Data[4]	Data[3]	Data[2]	Data[1]
	Output 209-216	Output 201-208	Marker 433-440	Marker 425-432
RDB	Data[8]	Data[7]	Data[6]	Data[5]
	0	Output 233-240	Output 225-232	Output 217-224

Y= 2,3,4,5,6,7,8	Bit 32-25	Bit 24-17	Bit 16-9	Bit 8-1
RDA	Data[4]	Data[3]	Data[2]	Data[1]
	Indicator[2Y-3].data[4]	Indicator[2Y-3].data[3]	Indicator[2Y-3].data[2]	Indicator[2Y-3].data[1]
RDB	Data[8]	Data[7]	Data[6]	Data[5]
	Indicator[2Y-2].data[4]	Indicator[2Y-2].data[3]	Indicator[2Y-2].data[2]	Indicator[2Y-2].data[1]

Y= 9	Bit 32-25	Bit 24-17	Bit 16-9	Bit 8-1
RDA	Data[4]	Data[3]	Data[2]	Data[1]
	Indicator[15].data[4]	Indicator[15].data[3]	Indicator[15].data[2]	Indicator[15].data[1]
RDB	Data[8]	Data[7]	Data[6]	Data[5]
	0	0	0	0

4.6 Frame type Indicator detail

Bit #	Data[4]	Data[3]	Data[2]	Data[1]
8	AVAILABLE	Value Sign bit	Value 2^{15}	Value 2^{7}
7	ERROR	Value 2^{22}	Value 2^{14}	Value 2^{6}
6	ZERO	Value 2^{21}	Value 2^{13}	Value 2^{5}
5	STABLE	Value 2^{20}	Value 2^{12}	Value 2^{4}
4	TARE	Value 2^{19}	Value 2^{11}	Value 2^{3}
3	FORMAT_1	Value 2^{18}	Value 2^{10}	Value 2^{2}
2	FORMAT_2	Value 2^{17}	Value 2^{9}	Value 2^{1}
1	FORMAT_3	Value 2^{16}	Value 2^{8}	Value 2^{0}

4.7 Status bits

Bit #	Data[4]	Description
8	AVAILABLE	Indicator value is available, value is invalid
7	ERROR	Indicator value is out of range
6	ZERO	Zero active
5	STABLE	Indicator value is stable
4	TARE	Tare active

4.8 Format bits

Data[4] bit 3	Data[4 bit 2]	Data[4] bit 1	Description
0	0	0	No decimal point, format 000000
0	0	1	Decimal point 1, format 000000.0
0	1	0	Decimal point 2, format 00000.00
0	1	1	Decimal point 3, format 0000.000
1	0	0	Decimal point 4, format 000.0000
1	0	1	Decimal point 5, format 00.00000
1	1	0	Decimal point 6, format 0.000000
1	1	1	Raw data, not a weigh signal

4.9 Error conditions

The CAN controller will stop sending if the bus has too many failures. Reset instrument or reconfigure the CAN baud rate to restart sending.

4.10 Example code, indicator reconstruction

```
CONST
        DivTable[0..6]= 10, 100, 1000, 10000, 100000, 1000000, 10000000 /* weight is standard x10 */



Status= Data[4] AND 0xF8 /* extract status information bits*/
Format = Data[4] AND 0x07  /* extract format bits */


/* convert data bytes to long */
IF Data[3] AND 0x80 THEN /* Detect sign bit */
        Value= 0xFF /* make value negative, sign extension*/
ELSE
        Value= 0x00 /* Value is positive */
Value= Value SHIFT_LEFT 8
Value= Value + Data[3]
Value= Value SHIFT_LEFT 8
Value= Value + Data[2]
Value= Value SHIFT_LEFT 8
Value= Value + Data[1]


IF (Status AND 0xC0) == 0x80 THEN /* detect available and not error */
        /* Value is valid */
        IF Format<7 THEN
                Weight= Value / DivTable[Format] /* divide Value and convert to float */
        ELSE
                Number= Value /* Value is not a weight */
ELSE
        /* Value is not available or not valid, don't usage Value ! */
```

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

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 ▪