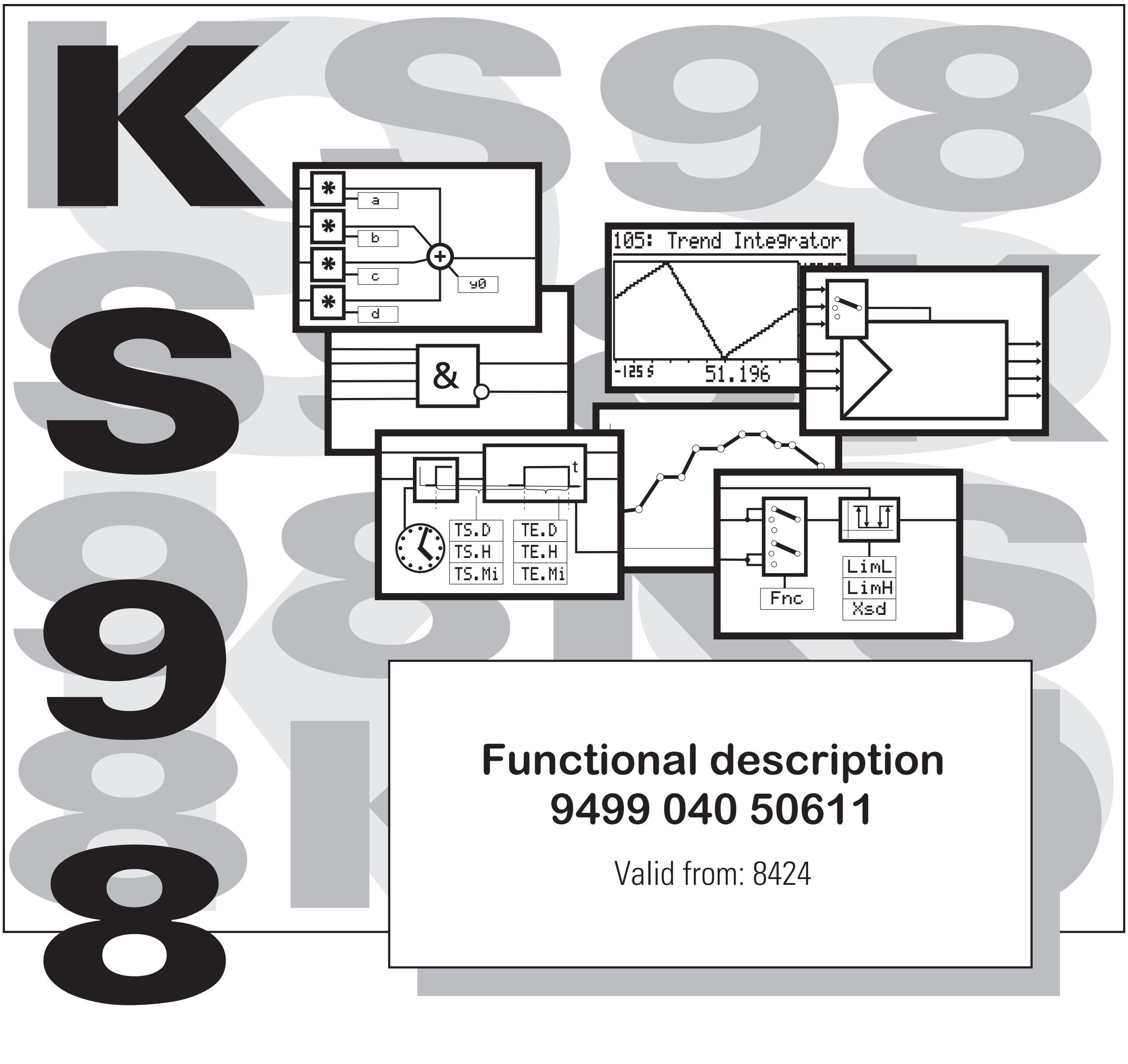
PMA Prozeß- und Maschinen-Automation GmbH

KS 98 and KS 98plus
Multi-function unit



Functional description
9499 040 50611

Valid from: 8424

Symbols on the instrument

EC-conformity marking

Caution, Follow the operating instructions!

A publication of:

PMA Prozeß- und Maschinen-Automation GmbH
P.O.Box 310 229
D-34058 Kassel
Germany

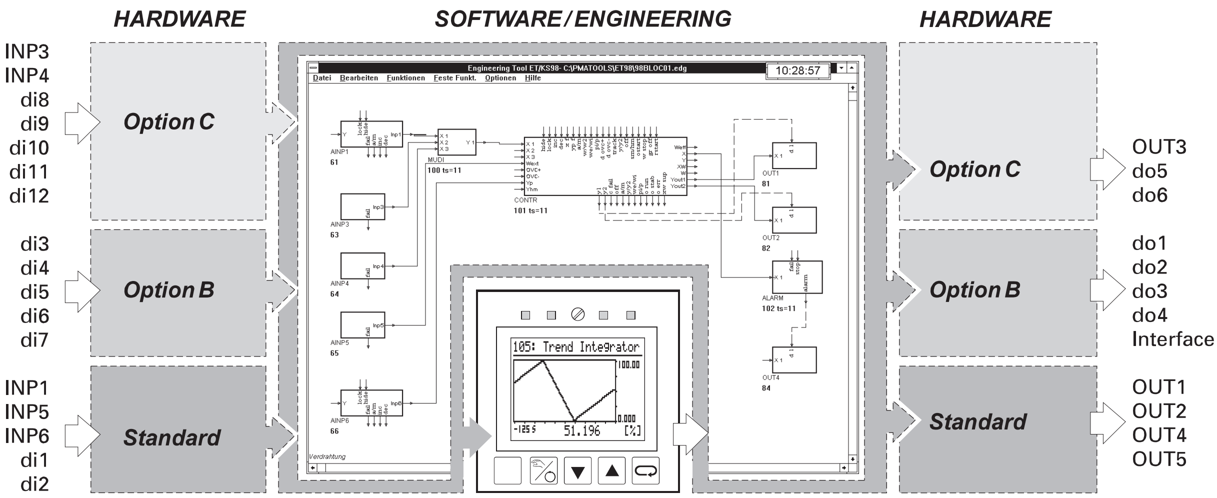
Content

1 Description

1.1 Construction

KS98 is a compact automation unit with freely configurable functionality using function blocks.

Each instrument contains a function library from which selection, configuration, parameter setting and combination of up to 150 function blocks is possible by means of an engineering tool. This permits realization of complex mathematical calculations, multi-channel control structures and sequencing in one instrument. Various pages are displayed by means of an LCD (64x128 dots): input and output for analog and digital signals, bargraphs and trends. Communication with other instruments and systems is possible via an optional digital interface.



Additionally, the introduction of a CANopen interface completes the basic version of KS98 multifunction unit by

- local I/O extensibility by means of the PMA RM 200 modular I/O system
- connection of PMA KS800 / KS816 multiple-channel temperature controllers with CANopren interface
- on-site data exchange with other KS98+ units (cross communication)

If necessary, connection of further sensors, actuators, transmitters, burner control systems or machine units, etc. with CANopen protocol is possible.

2 Important technical data

2.1 Analog inputs → sections 3 and 6.9

INP 1:	universal input, configurable for thermocouples, resistance thermometers, tem perature difference, resistance transducers, DC current and DC voltage
INP 3 and **INP 4**	(option C): DC current, INP 5: DC current and DC voltage
INP 6:	resistance transducer and DC current

2.2 Digital inputs

Opto-coupler for 24 V DC, current sink to IEC 1131 type 1, logic 0 = -3...5 V, logic 1 = 15...30 V, approx. 5 mA

di1 and di2:	in all versions,
di3...di7:	in option B,
di4...di12:	in option C.

2.3 Outputs → section 6.10

Relay contact rating: 500 VA, 250 V, 2 A at 48...62 Hz

OUT1, **OUT2**, **OUT4**, **OUT5**:	relay or current or logic dependent of version,
OUT 3	(option C): current

2.4 Control outputs

Opto-coupler, grounded load with common positive control voltage, power 18...32 V DC ≤ 100 mA.

do1...**do4**:	in option B,
do5 and **do6**:	in option C

2.5 Supply voltage

90...260V AC, 48...62 Hz, power consumption approx. 10 VA (equipped with all possible options)
For detailed technical data, see data sheet 9498 737 32133.

2.6 Further "modular" in- and outputs

Further analog and digital inputs and outputs can be provided on the "modular C card"
→ see KS98 I/O-extension modules on page 225 ff

2.7 Further external in- and outputs

Further analog and digital inputs and outputs can be connected via CANopen.
→ see KS98+ I/O extensions with CANopen page 117 ff

3 Versions

Order no. 9 4 0 7 - 9 _ _ - _ _ _ _ 1

		Code
Basic unit	standard	6
	with integrated supply voltage	7
	KS98+ with CANopen I/O [*1)]	8
Power supply and process outputs	90...250 V AC with 4 relays	3
	90...250 V AC with 2 relays + 2 current outputs	5
	24 V UC with 4 relays	7
	24 V UC with 2 relays + 2 current outputs	9
Option B	no option B	0
	TTL interface + di3...7 / do1...4	1
	RS 422 + di3...7 / do1...4 + real-time clock	2
	PROFIBUS-DP + di3...7 / do1...4	3
	INTERBUS + di3...7 / do1...4	4
Option C	no option C	0
	INP3 / INP 4 / OUT 3 / di8...12 / do5 / do6	1
	INP3 [*2)] / INP 4 / OUT 3 / di8...12 / do5 / do6	2
	modular option c basic card (not with KS98+) [*1)]	3
	modular option c basic card with modules [*1)]	4
Engineering	single-channel controller (basic unit)	0
	cascade controller	1
	flow controller	2
	program controller	3
	heat-energy counter	4
	flow calculator	5
Setting	standard setting	0
	setting to specification	9

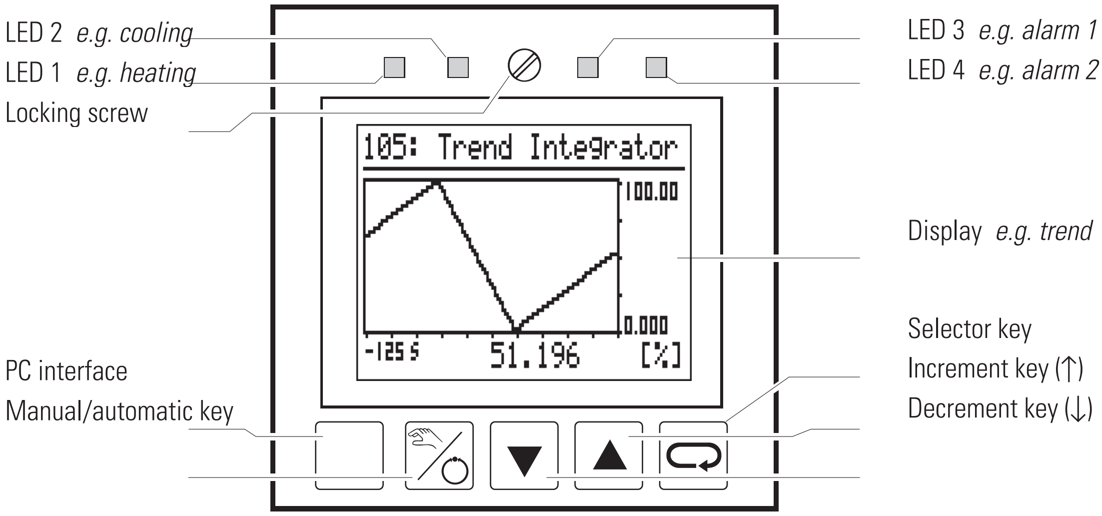
*1) *The combination KS98+ (CANopen I/O)* ***and*** *modular option c is* ***not*** *possible! Either KS98+ or modular option c!*
*2) *INP3: With type = 0...20 mA the input is scaled to -50...1300 mV. If output INP3 shall be used with this scaling, x0 must be set to -50 and x100 to 1300.*

3.1 I/O-modules - for units with modular option c basic card

Order no. 9 4 0 7 - 9 9 8 - 0 _ _ _ 1

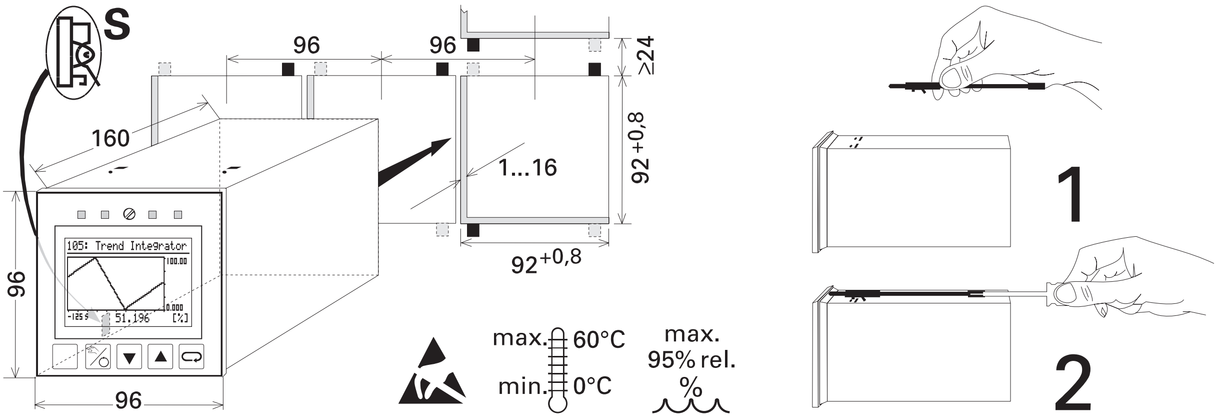
			Code
Plug position	Separate order		0
	Fitted on KS98 slot 1	galvanically connected	1
	Fitted on KS98 slot 2		2
	Fitted on KS98 slot 3	galvanically connected	3
	Fitted on KS98 slot 4		4
Module type Analog inputs	Pt100 / 1000, Ni 100 /1000, resistance, pot		2 0
	Thermocouple, mV, 0/4...20mA		2 1
	-50...1500mV, 0...10V		2 2
Module type- Analog outputs	0/2...10V, 0...±10V		3 0
	0/4...20mA, 0...±20mA		3 1
Module type Dig.inputs/outputs	Digital I/O (universal)		4 0
	Frequency/counter input		4 1

4 Front view

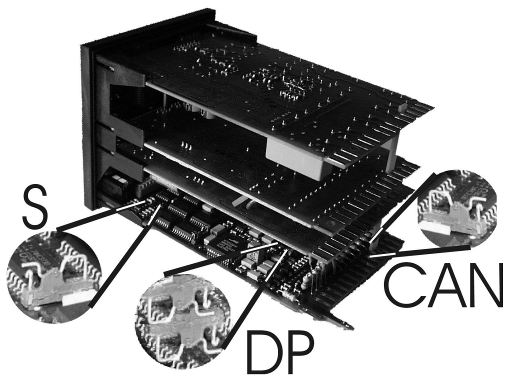


- Locking screw: locks the controller module in the housing.
- LEDs: indicate the statuses of the LED function (→ 219).
- Display: LCD dot matrix with (64x128 dots, back lighting). The relevant display is shown in sections 4 Menus, 6.12 Visualization, 6.14 Programmer and 6.15 Controller.
- Keys ▼▲: The relevant function is described in section 4 Menus.
- PC interface: PC connection for structuring/wiring/configuring/parameter setting/operating with the engineering tool.

5 Mounting



Mount the unit with min. 2 fixing clamps (diagonally at top and bottom).
Protection type IP65: use 4 fixing clamps. Insert the instrument module firmly and block it using the locking screw.
S.I.L. switch S: its switching status is signalled by function STATUS and can be used in the engineering. After delivery, the switch is open. For closing, release the fixing screw, withdraw the instrument module from the housing, close the S.I.L. switch. Insert the unit and lock it with the screw.



SIL switches DP: PROFIBUS terminating resistor
Every Profibus network must be terminated. This means that the first and the last node on the bus must be provided with a terminating resistor.
(→ see KS98 PROFIBUS-DP interface description 9499-040-52711).
In KS98, the bus terminating resistor can be activated by 2 SIL switches (DP). Both switches must always be closed or open (closed = terminating resistor is active).

SIL switch for CAN: for CANbus terminating resistor,
→ see chapter 22

Take care that the unit is tight!

Caution! The unit contains electrostatically sensitive components.

6 Electrical connections

6.1 Safety hints

Following the enclosed safety hints 9499 047 07101 is indispensable! The instrument insulation meets standard EN 61 010-1 (VDE 0411-1) with contamination degree 2, overvoltage category III, operating voltage range 300 V and protection class I.

With horizontal installation, the following rule is applicable additionally: with the instrument module withdrawn, a facility which prevents conducting parts from dropping into the open housing must be provided.

If the unit is switched to off-line, the outputs keep the status from the time of switch-over!!!

6.2 Electromagnetic compatibility

The unit meets European guideline 89/336/EEC. The following European generic standards are met:
Electromagnetic radiation: EN 50081-2 and Electromagnetic immunity: EN 50082-2. The unit is suitable for use in industrial areas (in residential areas, RF interference may occur). The electromagnetic radiation can be reduced decisively by installing the unit in a grounded metal switch cabinet.

6.3 Measurement earth (for grounding interference)

Interference voltages, e.g. high-frequency interference, acting on the unit from outside may cause functional trouble. For grounding the interference voltages and ensuring interference suppression, a measurement earth must be connected.

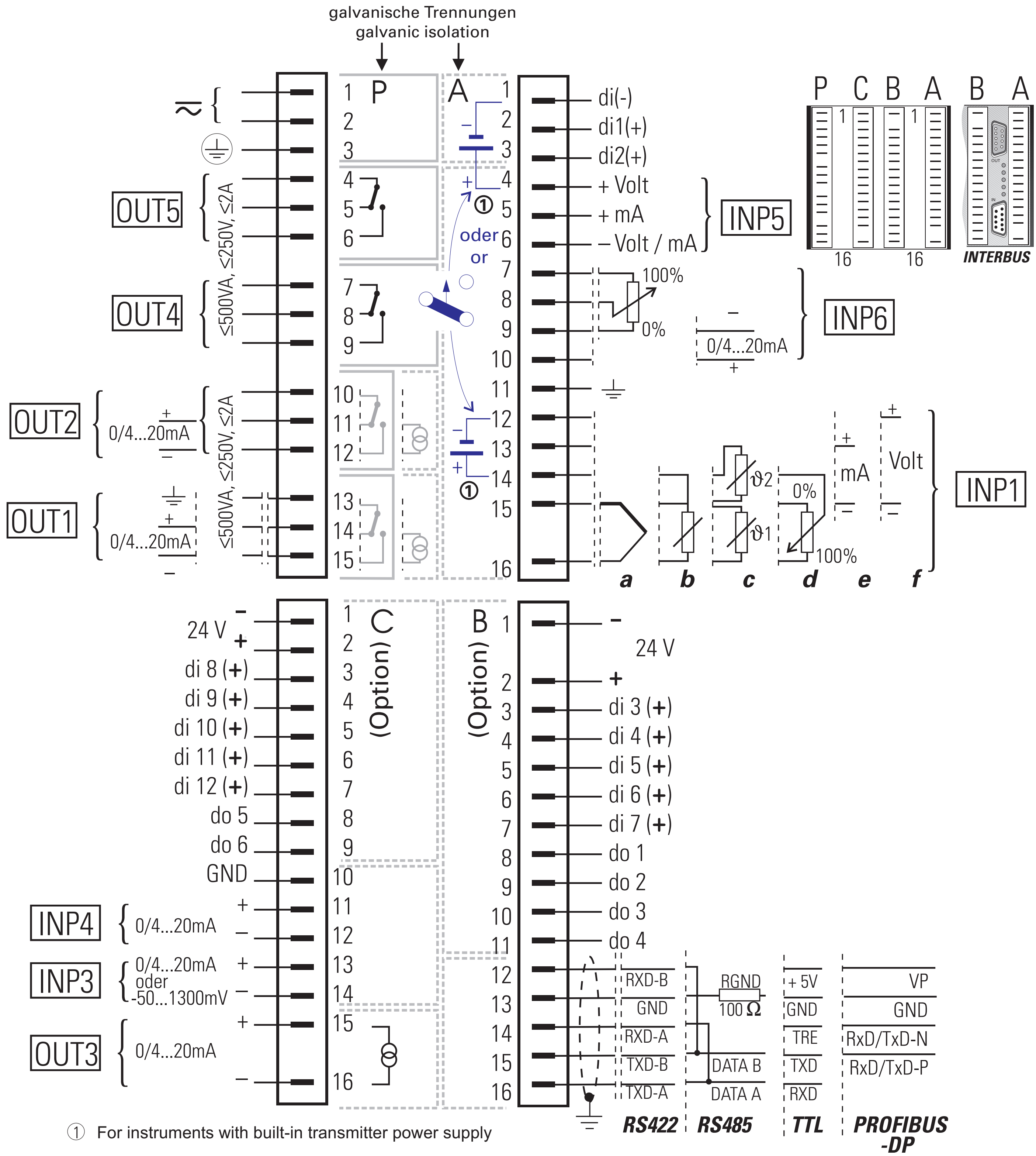
Terminal A11 must be connected to ground potential by means of a short cable (approx. 20 cm, e.g. at control cabinet ground)!

This cable must be kept separate from power supply cables. On instruments with current outputs at OUT1 and OUT2 proceed accordingly with terminal P13.

6.4 Connecting diagram

- Power supply cables must be kept **separate** from signal and measuring cables.
- We recommend using **twisted and screened measuring cables** (screening connected to measurement earth).
- Connected motor actuators must be provided with **protective circuitry** to manufacturer specifications. This avoids high voltage peaks, which may cause trouble to the instrument.
- The units must be protected with a fuse individually per unit or in common according to a max. power consumption of 10 VA (standard fuse ratings, min. 1 A)!

In measuring and signal circuits, the max. potential against ground may be 50 Veff,
Between power supply circuits, the max. potential may be 250 Veff.



① For instruments with built-in transmitter power supply

* with 24 V DC / AC connection of the protective earth is also necessary, the polarity it is uncritical.
For devices with modular option C see according chapter

6.5 Analog inputs (→ connecting diagram)

Thermocouples (a)

No lead resistance adjustment is necessary.
Internal temperature compensation: the compensating lead must be taken up to the instrument terminals.
In AINP1 STK = int.TC must be configured.
External temperature compensation: use a separate cold-junction reference with fixed reference temperature. The compensating lead is taken up to the cold-junction reference, whilst the lead between cold-junction reference and instrument can be of copper.
In AINP1 STK = ext.TC and with TKref = the reference temperature must be configured.

Resistance thermometer Pt 100 in 3-wire connection (b)

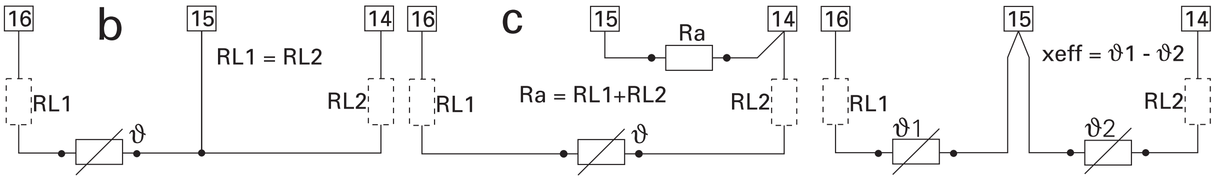
No lead resistance adjustment is necessary, provided that RL1 = RL2.

Resistance thermometer Pt 100 in 2-wire connection (c)

Lead resistance adjustment is necessary: make Ra equal to RL1 + RL2.

Two resistance thermometers Pt100 in difference connection (d)

Compensating the lead resistances: proceed as described → 24 .

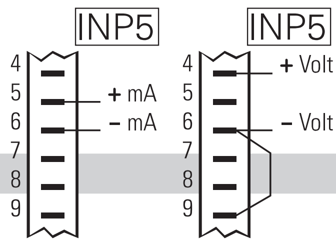


Resistance transducer (e)

Calibrating the measurement: proceed as described →24 .

Standard voltage signals 0/2...10V (g)

Input resistance: ≥ 100 kΩ, Configure scaling and digits behind the decimal point. INP5 is a difference input, the reference potential of which is connected to terminal A9. With voltage input, A6 must always be connected with A9.



Standard current signals 0/4...20 mA (f)

Input resistance: configure 50 Ω, scaling and digits behind the decimal point.

Supply voltage -50...1300mV DC

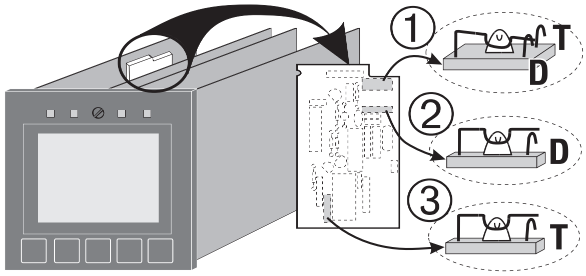
(only INP3 on instruments with order no. 9407-9xx-x2xx1):
With type =0...20mA, the input is designed for -50...1300 mV. For using the INP3 output with this scaling, set x0 to -50 and x100 to 1300.

Inputs INP1 / INP6 are interconnected. This must be taken into account, if both inputs must be used for standard current signals. If necesary, a galvanic isolation must be used.

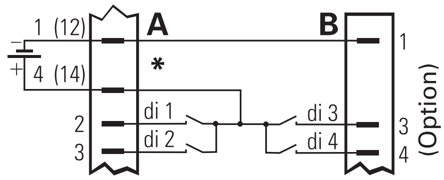
6.6 Versions with integrated supply voltage

The potential-free supply voltage can energize a 2-wire transmitter or max. 4 control inputs. The output connections can be selected with 3 S.I.L. switches:

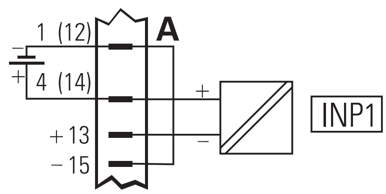
Connections		①	②	③	Remarks
14(+)	**12 (-)**	**T**	**open**	**closed**	Only available, if INP1 is configured for current or thermocouple
4 (+)	**1 (-)**	**D**	**closed**	**open**	The voltage input of INP5 is not available



Energization of digital inputs (e.g. di1...di4)



2-wire transmitter connection (e.g. INP1)



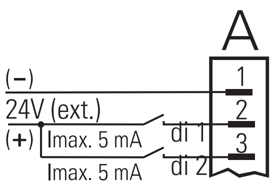
Factory setting: ① = T, ② = open, ③ = closed (T).
The unit must be withdrawn from the housing for changing the switch positions.
The S.I.L. switches are located on the circuit board shown right.

* If A14/A12 is used for di1/di2, A12 must be connected with A1

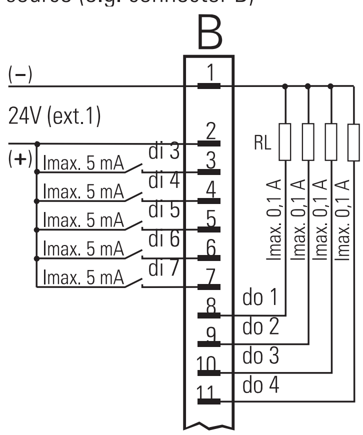
6.7 Digital inputs and outputs (→ connecting diagram)

The digital inputs and outputs must be energized from one or several 24 V DC sources. The current consumption is 5 mA per input. The max. load is 0,1 A per output. Examples:

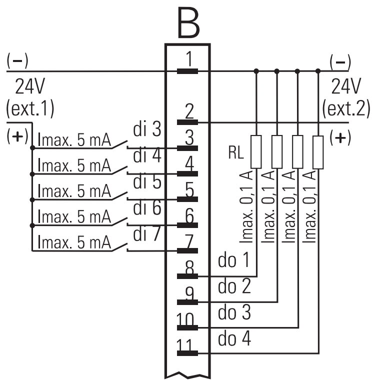
Digital inputs (connector A)



Digital inputs and outputs at one voltage source (e.g. connector B)



Digital inputs and outputs at two voltage sources (e.g. connector B)

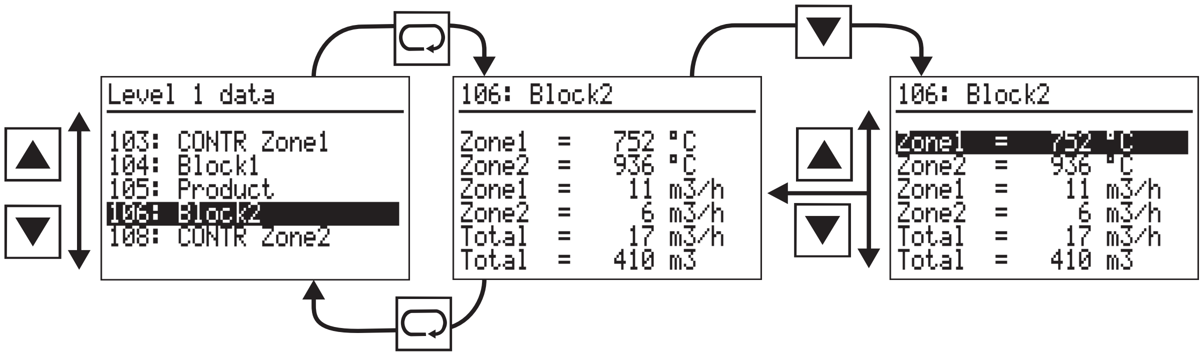


7 Menu

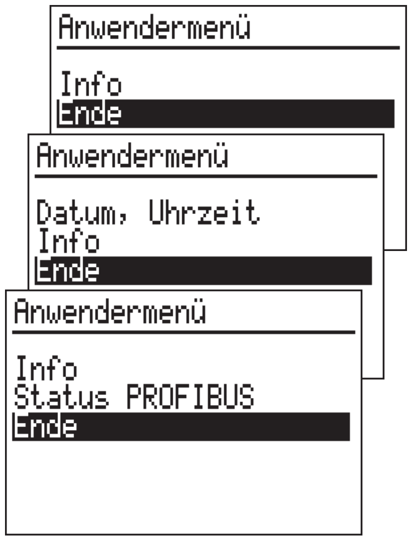
The instrument operation is menu-guided. Distinction between complete dialogue and short-form dialogue is made. In the complete dialogue, the main menu with its sub-menus is shown so that all permitted settings are selectable. In the short-form dialogue, the main menu is switched off so that accidental access is prevented. In this case, only the operating page menu with the permitted operating pages is selectable. The short-form dialogue is available from operating version 2.

7.1 Short-form dialogue

Available from operating version 2. The main menu is switched off via interface (`m-hide`) or function STATUS (`m-hide`). The operating page menu with the permitted operating pages is selectable. Selecting, marking lines and adjusting values are done as described further down in this manual.



When pressing key during > 3 s, a *user menu* is shown, which is different dependent of instrument version (standard / real-time clock / PROFIBUS):
Line `Info`: hardware order no., software order no., software version and operating version.
Line `Date, time`: display and adjustment of date and time.
Line `Status PROFIBUS`: bus access, parameter setting, configuration and data communication status.



7.2 Complete dialogue

A main menu for selecting the sub-menus, which can be used for selecting an instrument and application-dependent number of pages.

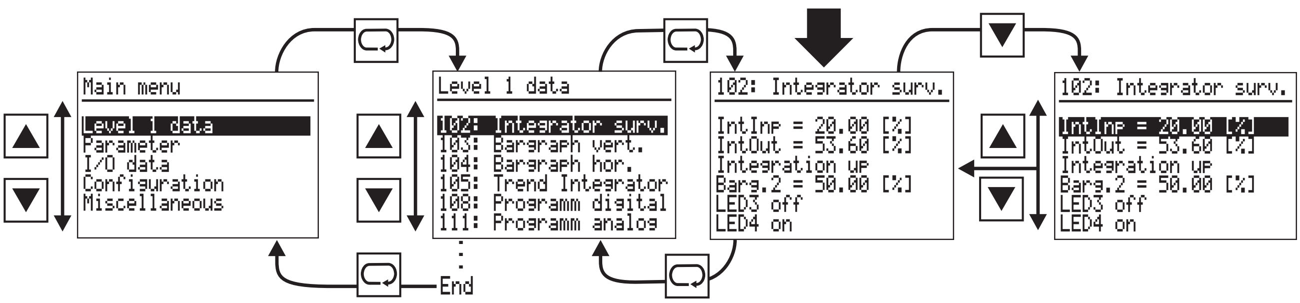
Sub-menu	Page contents
`Level 1 data`	The operating pages of VWERT, VPARA, VBAR, VTREND, APROG, DPROG, CONTR and CONTR+ are shown: display and adjustment of operating values.
`Parameter`	For each function used, which has adjustable parameters a separate page is prepared: parameter display and adjustment.
`I/O data`	There is a separate page for each used function: Input and output data display.
`Configuration`	There is a separate page for each used function, which must be configured: configuration data display and adjustment. For changing a configuration, the instrument must be set to ‘offline’. (→ operating modes).
`Miscellaneous`	Page `Date, Time`: date, time display and adjustment. ① Page `Device data`: interface, mains frequency, language display and adjustment. Page `Online/Offline`: online ↔ offline, configuration cancelation. Page `Calibration`: Display and calibration of signals to be calibrated. Page `Info`: hardware, software order no., software version no. display. ② Page `Status CAN-BUS`: State of eventually connected CAN-Nodes. ③ Page `Status PROFIBUS`: bus access, data communication status display. ④ Page `Status INTERBUS`: bus access, data communication status display. ⑤

Only with option B with built-in real-time clock
The operating version is also displayed from operating version 2
Only KS98+ (with CAN I/O extension (see page))
Only with option B with PROFIBUS
Only with option B with INTERBUS

Prior to operating version 2 `KS98:` was displayed additionally in the headers of main menu and five sub-menus. Example `KS98: main menu`

7.3 Selection (switch-on and operating pages)

After power supply switch-on, logo and `Main menu wait!` followed by the sub-menu during some seconds are displayed. Unless a selection is made during this time, the first operating page entered in the sub-menu without marked line is displayed. At each pressure of ▲ / ▼ a line (inverted display) is marked. When the page without marked line is reached again by pressing keys ▲ / ▼, return to the sub-menu is possible by pressing key ⮌. When the `End` is reached by pressing keys ▲ / ▼, return to the main menu is possible by pressing key ⮌.



Keys ▲ / ▼ scroll the marked line up to the menu start or down to the menu end. When pressing the key again, the marked line jumps from the start to the end or vice versa.

7.4 Language selection

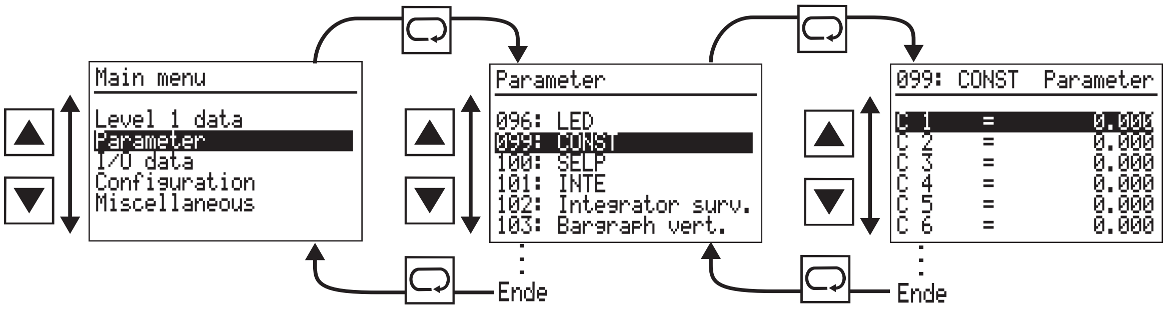
English: mark `Allgemeine Daten` → `Gerätedaten` → `Sprach = deutsch`.
Press ↵: `deutsch` blinks. Press ▲: `english` blinks. Press ↵: `Main menu` is displayed.

German: Mark `Miscellaneous` → `Device data` → `Langu. = english`.
Press ↵: `english` blinks. Press ▼: `deutsch` blinks. Press ↵: `Hauptmenü` is displayed.

7.5 Selection (other pages)

In the main menu, select the sub-menu (inverse display) with ▲▼ and display it with ↵. In the sub-menu, select the page with ▲▼ and open it with ↵. The first line is marked (inverse, → value adjustment). When the `End` is reached with ▲▼, returning to the sub-menu is possible with ↵. When the `End` is reached with ▲▼ in the sub-menu, return to the sub-menu is with ↵.

Example: Parameter

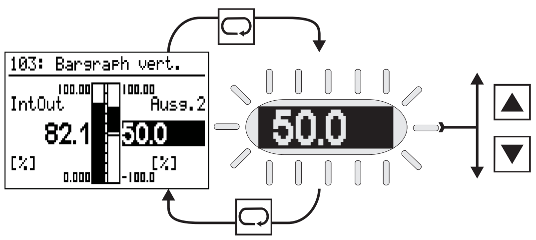


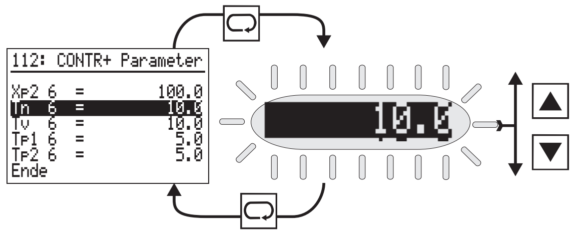
▲▼ scrolls the marked line to the menu start or end. When pressing again, the marked line changes from the start to the end, or vice versa.

7.6 Adjusting values

Values in marked lines of pages can be adjustable. Mark the required line or variable by pressing ▲▼ (inverted display). When confirming the value with ↵, it starts blinking and can be adjusted with ▲▼. When reaching the required value, it must be confirmed with ↵. Subsequently, another page can be marked with ▲▼.

Example: Bargraph vertical
Example: Parameter CONTR+





7.7 Calibrating

Select (▲▼) and open (⮐) Miscellaneous → Calibration. Select the bottommost line by pressing ▲ (inverted display, e.g. Quit). Then continue as follows:

Transducer input (INP1 or INP6)
Calibrating the transducer start and end:

① Set the transducer to start (→ section operating modes)
② Press ⮐ → Quit blinks *
③ Press ▲ → Set 0% blinks
④ Wait until the input has settled (min. 6 s)
⑤ Press ⮐ → 0% done is displayed
⑥ Set the transducer to end (→ Section Operating modes)
⑦ Press ⮐ → 0% done blinks
⑧ Press ▲ 3x → Set 100% blinks
⑨ Wait until the input has settled (min. 6 s)
⑩ Press ⮐ → 100% done is displayed

Calibration is finished.
For leaving the calibration press ▼ until nothing is marked and press ⮐.

2 Resistance thermometer in difference (INP1)
Calibrating the lead resistance effect:

① Short-circuit the thermometer in the connecting head
② Press ⮐ → Quit blinks *
③ Press ▲ → Set Dif blinks
④ Wait until the input has settled (min. 6 s)
⑤ Press ⮐ → Cal done is displayed

Lead resistance adjustment is finished.
Remove both short circuits. For leaving the calibration press ▼ until nothing is marked and press ⮐.

* If another word blinks, key ▲ or ▼ must be pressed, until the required word blinks.

7.8 Operating modes

❑ Online/Offline
For configuration changing, set the unit to ‘Offline’ and back to ‘Online’ (Miscellaneous, Online/Offline).

❑ Manual mode/automatic mode
When using controllers, it may happen that automatic or manual mode is requested by several units. The controller leaves the manual mode, when all control signals request automatic operation.
Example: INP6 is provided for transducers and connected accordingly (position feedback). When this input is calibrated, the controller can be switched to manual mode on the calibrating page (by means of [manual key], Man. is displayed on the bottom left). Now, pressing ▲ and ⮐ to mark line Y and pressing ▲ / ▼ to drive the actuator to its limits is possible. After calibration, the manual mode must be switched off again on this page. (press [manual key]).

8 Maintenance

8.1 Behaviour in case of trouble

The unit is maintenance-free. In case of trouble, the following points must be checked.

- The unit is in on-line mode.
- The supply voltage is connected correctly with correct voltage and frequency.
- All connections are in correct condition.
- Sensors and motor actuators operate correctly.
- The used engineering is the required one and is O.K.
- The unit is configured for the required operating mode.
- The adjusted parameters generate the required effect.

If the unit does not work perfectly after these checks, shut it down and replace it. A defective unit can be sent to the supplier for repair.

8.2 Shut-down

Switch off the supply voltage completely and protect it against accidental operation. As the unit is mostly connected in the same signal loop with other instruments, measures against the occurrence of undesired operating conditions due to the switch-off effects must be taken.

8.3 Cleaning

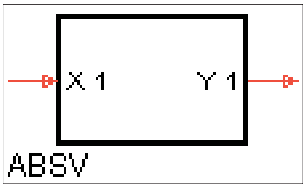
Housing and front panel can be cleaned using a dry, lint-free cloth. Do not use solvents or cleaning agents!

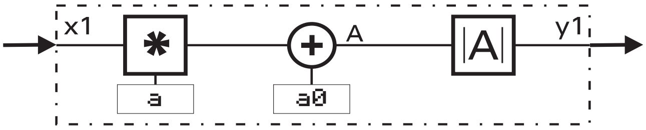
8.4 Further information

		Order no.
For a structured single-channel controller	operating instructions	9499 040 51001
For a structured cascade controller	operating instructions	9499 040 51101
For a structured flow controller	operating instructions	9499 040 51201
For a structured program controller	operating instructions	9499 040 51301
For a structured heat energy counter	operating instructions	9499 040 51401
For a structured flow calculator	operating instructions	9499 040 51501
For the engineering tool	operating instructions	9499 040 45741
For the digital interface (ISO1745)	interface description	9499 040 45111
For the PROFIBUS	interface description	9499 040 52711
For the INTERBUS	interface description	9499 040 57011
Engineering manual	manual	(in preparation)

9 Scaling and calculating functions

9.1 ABSV (absolute value)





$$y_1 = \left| a \cdot x_1 + a_0 \right|$$

The absolute value of a number is the number without polarity sign.

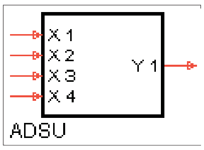
Input variable x1 is multiplied by factor a (parameter). Now, constant a0 is added. The absolute value of the resulting value is formed and output at y1 .

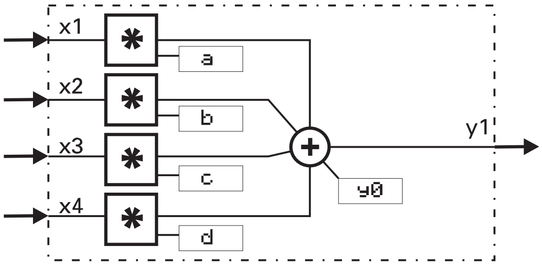
Example:

y1 = ABS(a • x1 + a0) a=5 x1 =2 a0 = +5 results in y1 = 15
y1 = ABS(a • x1 + a0) a=5 x1 =2 a0 = -20 results in y1 = 10

Parameter	Description	Range	Default
a	Multiplication factor	-29 999...999 999	1
a0	Offset	-29 999...999 999	0

9.2 ADSU (addition/subtraction)



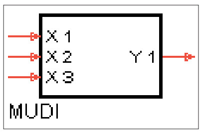


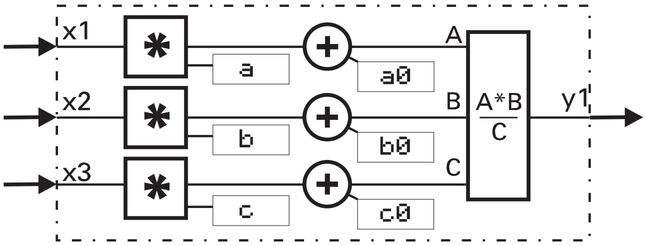
$$y_1 = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 + y_0$$

Input variables x1 ...x4 are multiplied by factors a...d. Constant y0 is added to the sum of evaluated inputs. Value "0" is assigned automatically to unused inputs.

Parameter	Description	Range	Default
a...d	Multiplication factors	-29 999...999 9990	1
y0	Offset	-29 999...999 999	0

9.3 MUDI (Multiplication / division)



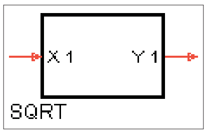


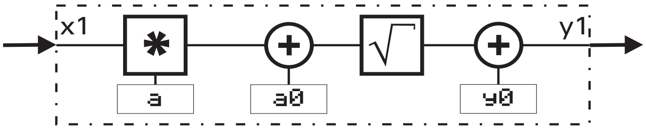
$$y_1 = \frac{A \cdot B}{C} = \frac{(a \cdot x_1 + a_0) \cdot (b \cdot x_2 + b_0)}{c \cdot x_3 + c_0}$$

Input variables x1 ...x3 are multiplied by factors a, b, c. The relevant constants a0, b0, c0 are added. The output variable corresponds to the product.
Value “1" is assigned automatically to unused inputs.
With divisions by “0" (C = c • x3 + c0 = 0) output y1 is set to $1.5 \cdot 10^{37}$.

Parameter	Description	Range	Default
a...c	Multiplication factors	-29 999...999 999	1
a0...c0	Offset	-29 999...999 999	0

9.4 SQRT (square root function)



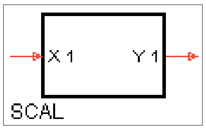


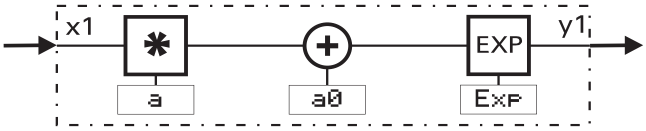
$$y_1 = \sqrt{a \cdot x_1 + a_0} + y_0$$

Constant a0 is added to input variable x1 multiplied by a. The result is subjected to square root extraction. Constant y0 is added to the result of square root extraction.
If the expression under the root is negative, the square root expression is set to 0. As a result: y1 = 0.
If the input is not connected, this is interpreted as x1 = 0.

Parameter	Description	Range	Default
a	Multiplication factors	-29 999...999 999	1
a0	Input offset	-29 999...999 999	0
Y0	Output offset	-29 999...999 999	0

9.5 SCAL (scaling)





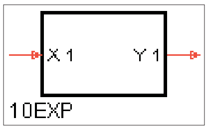
$$y_1 = (a \cdot x_1 + a_0)^{Exp}$$

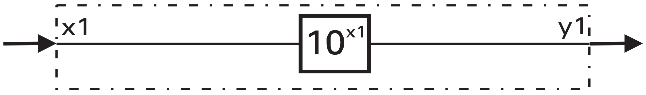
Input variable x1 is multiplied by factor a and added to constant a0.
The result (a • x1 + a0) is set to the power Exp.

If x1 is not used, this is interpreted as x1 =0. With Exp = 0 SCAL outputs 1.

Parameter	Description	Range	Default
a	Multiplication factor	-29 999...999 999	1
a0	Offset	-29 999...999 999	0
Exp	Exponent	-7...7	1

9.6 10EXP (10s exponent)





$$y_1 = 10^{x1}$$

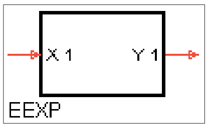
Input value **x1** is calculated according to formula $y_1 = 10^{x1}$ and output at the **y1** output.

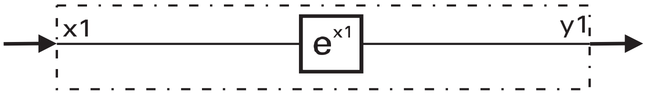
An unwired **x1** is interpreted as **x1** = 0 (in this case **y1** is 1).

If the value at input **x1** is higher than 36,7, an overflow may occur. In this case, output **y1** is set to $1.5_{w} 10^{37}$ rather than forming the power.

✎ ***Note:***
10EXP is the reversal function of function LG10.

9.7 EEXP (e-function)





$$y_1 = e^{x1}$$

The e-function is calculated.
If input signal **x1** is higher than 85, there may be an overflow. In this case, **y1** = 1,5 $_{w}$ 10^{37} is output rather than forming the power.

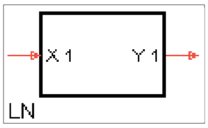
If **x1** is not wired, this is interpreted as **x1** = 0 and thus as **y1** = 1.

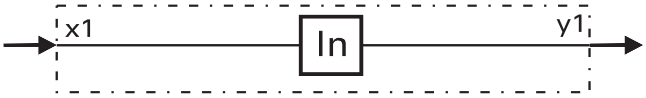
✎ ***Note:***
EEXP is the reversal function of function LN.

Examples:

With an input value of **x1** = 5, output value **y1** = 148,413159.
With an input value of **x1** = 0,69314718, output value **y1** = 2.

9.8 LN (natural logarithm)





$$y_1 = ln\,(\,x1)$$

The natural logarithm of input variable **x1** is formed.
The basis of natural logarithms is constant e (2,71828182845904).
If **x1** is not wired, this is interpreted as **x1** = 1. In this case **y1** is 0.

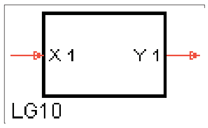
With a negative input variable **x1**, **y1** = -1,5 $_w$ 10^{37} is set.

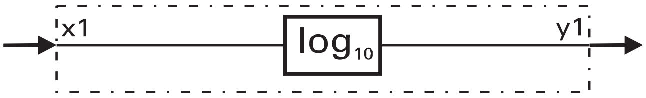
Note:
LN is the reversal function of function EEXP.

Examples:

The result of input value **x1** = 63 is an output value of **y1** = 4,143134726.
The result of input value **x1** = 2,71828182845904 is an output value of **y1** = 1.

9.9 LG10 (10s logarithm)





$$y_1 = log\,(\,x1)$$

The common logarithm of input variable x1 is formed.
LG10 provided the logarithm of a number to base 10.
If x1 is not wired, this is interpreted as x1 = 1. In this case, y1 is 0.

With a negative input variable x1 , y1 = -1,5 $_{w}$ 10^{37} is set.

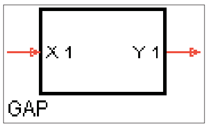
Note:
LG10 is the reversal function of function 10EXP.

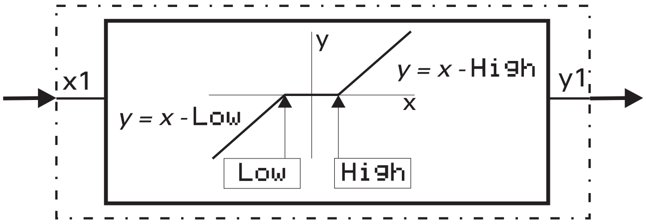
Examples:

The result of input value x1 = 63 is an output value of y1 = 1,799340549.
The result of an input value x1 = 2,71828182845904 is an output value of y1 = 1.

10 Non-linear functions

10.1 GAP (dead band)





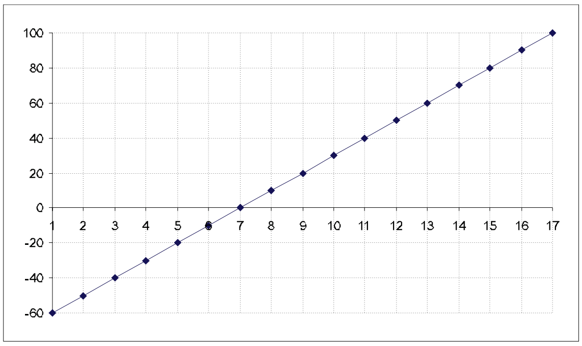
$y_1 = x1 - L$ bei x1 < L
$y_1 = 0$ bei x1 = L...H
$y_1 = x1 - H$ bei x1 > H

The range of the dead band is adjusted with parameters Low (lower limit) and High (upper limit). If input value x1 is within the dead band (Low ≤ x1 ≤ High), output value y1 = 0. If x1 is not used, this is interpreted as x1 = 0.

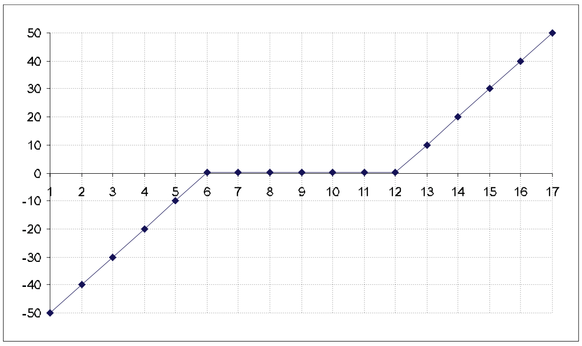
Example:

In the following example, -10 for Low and 50 for High was used.

Input x1

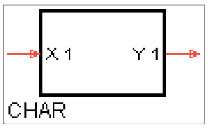


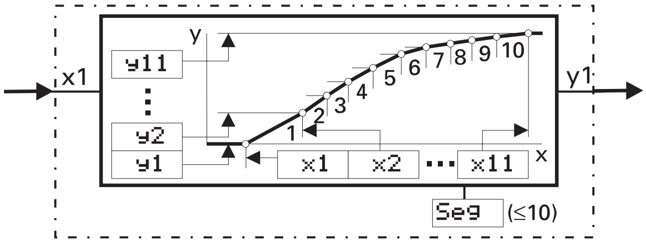
Output y1



Parameter	Description	Range	Default
Low	Lower limit value	-29 999...999 999	0
High	Upper limit value	-29 999...999 999	0

10.2 CHAR (function generator)



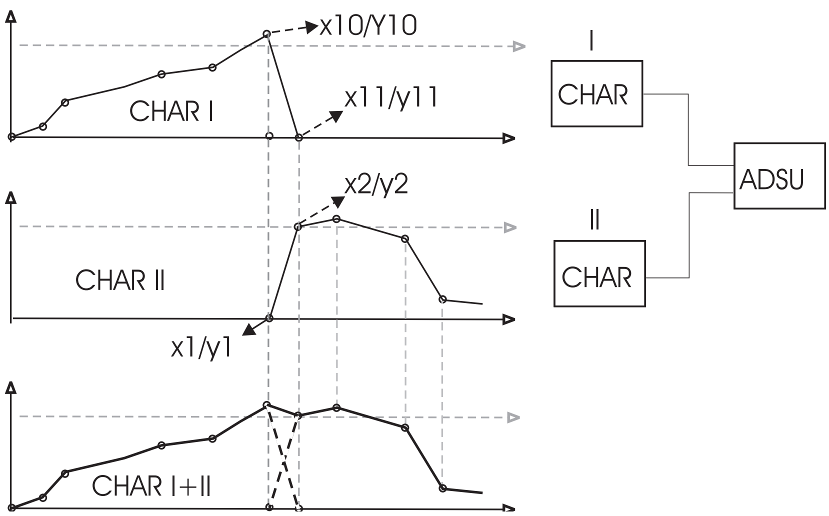


With max. 11 adjustable value pairs, non-linear functions can be simulated or linearized. Each value pair comprises input x(1) and output y(1). The number of value pairs is determined using configuration parameter Seg (number of segments +1 corresponds to the number of value pairs).
The value pairs are connected automatically with straight lines so that each input value x1 provides a defined output value y1.
If input value x1 is smaller than parameter x(1), the output value is equal to the value of y(1).
If input value x1 is higher than the highest parameter x(n) the output value is equal to the corresponding y(n) value.
During entry of the configuration parameters, the condition is that the assigned values stand in ascending order (x(1) < x(2) <...< x(11)).

Configuration	Description	Range	Default
Seg	Number of segments	1...10	2
x(1)...(11)	Input value for curve point	-29 999...999 999	0...10*
y(1)..(11)	Output value for curve point	-29 999...999 999	0...10*

* 0 for x(1) and y(1), 1 for x(2) and y(2) ... 10 for x(11) and y(11).

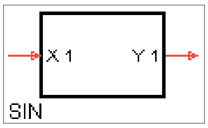
Unless one CHAR is sufficient, the following tip might be helpful:



whereby x10 of CHAR I = x1 of CHAR II and x11 of CHAR I = x2 of CHAR II

11 Trigonometric functions

11.1 SIN (sinus function)



$$y_1 = sin\,(\,x1)$$

The function provides the sinus of the input value, i.e. x1 is the angle the sinus of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

The calculation clarity can be reached by limiting the input signal (e.g. to the 1st or 4th quadrant (±90° / ± $^{0}/_{2}$)). Internal limiting is not provided. If input value x1 is out of the range in which the sinus function can still provide purposeful values, output y1 is set to 1,5 • 10^{37}.

Example degree of angle:
y1 = sin(x1) x1 = 30° ≙ y1 = 0,5

Example radian:
y1 = sin(x1) x1 = 90rad ≙ y1 = 0,89399666

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang. deg
	Unit: radian	Radian

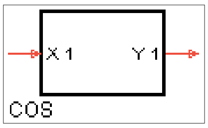
1 rad = 180°/π = 57,296°
1 ° = π/180° = 0,017453 rad

Control with the pocket calculator:
The function for the calculation in "rad" with the pocket calculator is limited to e.g. ±8 π.

→ 90/ π = 28,6479: sin (0,6479 • π) = 0,893996664

Also during input in "°" usually a limitation is effective in the pocket calculator (e.g. < 1440°)!

11.2 COS (cosinus function)



$$y_1 = \cos(x1)$$

The function provides the cosinus of the input value, i.e. x1 is the angle the cosinus of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

Calculation clarity can be reached by limiting the input signal (e.g. to the 1st and 2nd quadrant (0°...180° /0...π). Internal limiting is omitted. If input value x1 is out of the range in which the cosinus function can still provide purposeful values, output y1 is set to $1{,}5 \cdot 10^{37}$.

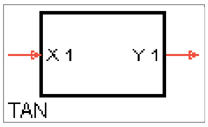
Example degree of angle:
y1 = cos(x1) x1 = 60° ≙ y1 = 0,5

Example radian:
y1 = cos(x1) x1 = 45rad ≙ y1 = 0,525321988

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

Important: When controlling with the pocket calculator see → **page 39**

11.3 TAN (tangent function)



$$y_1 = \tan(x1)$$
$$\text{valid for } x_1: -90° < x_1 < +90° \left(-\frac{\pi}{2} < x_1 < \frac{\pi}{2}\right)$$

The function provides the tangent of the input value, i.e. x1 is the angle the tangent of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

For calculation clarity the argument range is limited to the 1st or 4th quadrant (-90° ... 90° or -π/2 ... π/2). If input value x1 is out of this range, output y1 -1, • 10^{37} (x1 <-90 [-π/2]) or 1,5 • 10^{37} (x1 >90 [π/2]) is set.

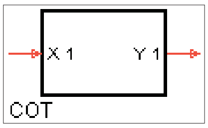
Example degree of angle:
y1 = tan(x1) x1 = 60° ≙ y1 = 1,73205

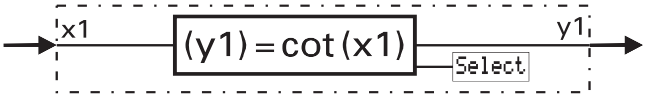
Example radian:
y1 = tan(x1) x1 = 1,53rad ≙ y1 = 24,498

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

Important: When controlling with the pocket calculator see → **page 39**

11.4 COT (cotangent function)





$$y_1 = cot\ (x1)$$
$$valid\ \ for\ x_1: 0 < x_1 < 180°(0 < x_1 < \pi)$$

The function provides the cotangent of the input value, i.e. x1 is the angle the cotangent of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

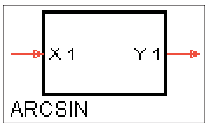
For calculation clarity, the range for the argument is limited to the 1st and 2nd quadrant (0° ... 180° or 0 ... π). If input value x1 is out of this range, output y1 is set to $1{,}5 \bullet 10^{37}$ (x1<0) or $1.5 \bullet 10^{37}$ (x1>180 [x1>π]).

Example degree of angle:
y1 = tan(x1) x1 = 45° ≙ y1 = 1

Example radian:
y1 = tan(x1) x1 = 0,1rad ≙ y1 = 9,967

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

11.5 ARCSIN (arcus sinus function)



$$y_1 = arcsin\,(x1)$$
$$valid\ \ for\, x_1:\ -1 \le x_1 \le +1$$

The function provides the arcus sinus of the input value, i.e. x1 is the angle the arcus sinus of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

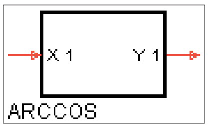
The calculation is output as degree of angle [-90° ... 90°] or as radian [-ó/2 ... ó/2]. With arguments out of the function validity range, output y1 is limited to $-1{,}5 \cdot 10^{37}$ (x1 <-1) or $1{,}5 \cdot 10^{37}$ (x1 >1).

Example degree of angle:
y1 = arcsin(x1) x1 = 0,5° ≙ y1 = 30

Example radian:
y1 = arcsin(x1) x1 = 1rad ≙ y1 = 1,571

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

11.6 ARCCOS (arcus cosinus function)



$$y_1 = \arccos(x1)$$
valid for x_1: $-1 \le x_1 \le +1$

The function provides the arcus sinus of the input value, i.e. **x1** is the angle the arcus sinus of which is calculated. Parameter **Select** is used to adjust, if the angle is provided in degree of angle [°] or in radian.

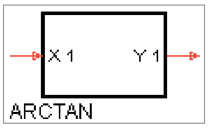
Calculation is either as degree of angle [0° ... 180°] or as radian [0...π]. With arguments out of the function validity range, output **y1** is set to $1{,}5 \cdot 10^{37}$ (**x1** <-1) or $-1{,}5 \cdot 10^{37}$ (**x1** >1).

Example degree of angle:
y1 = arccos(**x1**) **x1** = 0,5° ≙ **y1** = 60

Example radian:
y1 = arccos(**x1**) **x1** = 0,5rad ≙ **y1** = 1,047

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

11.7 ARCTAN (arcus tangent function)



$$y_1 = \arctan(x1)$$

The function provides the arcus tangent of the input value, i.e. x1 is the angle the arcus tangent of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

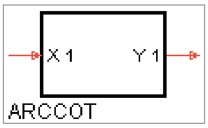
The calculation is output either as degree of angle [-90° ... 90°] or as radian [-$^{\Pi}/_{2}$ ... $^{\Pi}/_{2}$].

Example degree of angle:
y1 = arctan(x1) x1 = 1° ≙ y1 = 45

Example radian:
y1 = arctan(x1) x1 = 12rad ≙ y1 = 1,488

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

11.8 ARCCOT (arcus cotangent function)



$$y_1 = arccot\ (x1)$$

The function provides the arcus cotangent of the input value, i.e. x1 is the angle the arcus cotangent of which is calculated. Parameter Select is used to adjust, if the angle is provided in degree of angle [°] or in radian.

The calculation is output in degree of angle [0° ... 180°] and in radian [0 ... π].

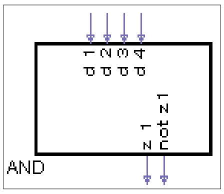
Example degree of angle:
y1 = arccot(x1) x1 = 45° ≙ y1 = 1,273

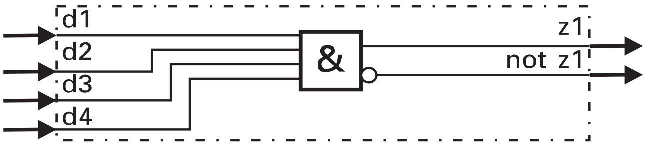
Example radian:
y1 = arccot(x1) x1 = -12rad ≙ y1 = 3,058

Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang.deg
	Unit: radian	Radian

12 Logic functions

12.1 AND (AND gate)



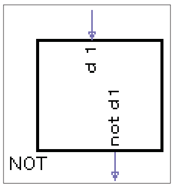


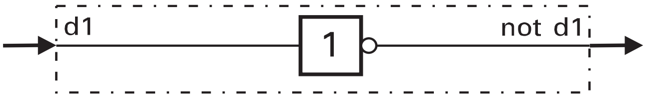
$$z_1 = d_1 \ AND \ d_2 \ AND \ d_3 \ AND \ d_4$$

Logic function AND combines inputs d1 ...d4 according to the truth table given below. Unused inputs are interpreted as logic 1.

d1	d2	d3	d4	z1	not z2
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	0	1

12.2 NOT (inverter)



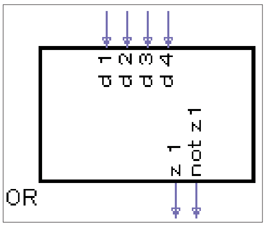


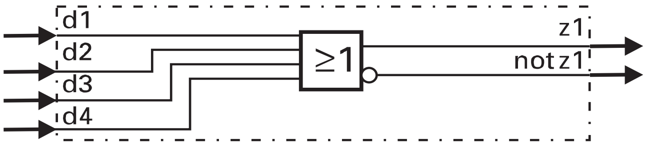
$$z_1 = \overline{d_1}$$

Logic input signal **d1** is output invertedly at **y1**. If **d1** is not wired, this is interpreted as logic 0.

d1	not z1
0	1
1	0

12.3 OR (OR gate)



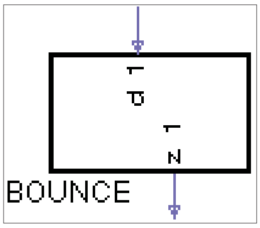


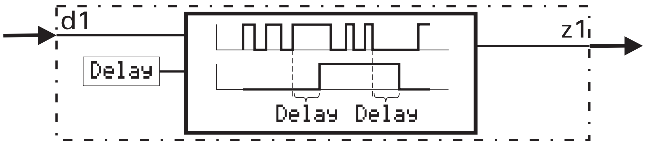
$$z_1 = d_1 \ OR \ d_2 \ OR \ d_3 \ OR \ d_4$$

Logic function OR combines inputs **d1** ...**d4** according to the truth table given below. Unused inputs are interpreted as logic 0.

d1	d2	d3	d4	z1	not z1
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	1	0
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	0

12.4 BOUNCE (debouncer)



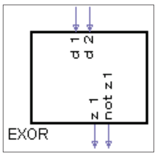


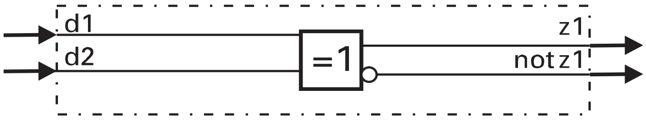
This function is used for de-bouncing a logic signal. The change of input signal d1 is transferred to output z1 only, when it remained constant for the time adjusted in parameter Delay. The time-out accuracy is dependent of the sampling interval assigned to the function.

Example:
Delay = 0,5s for assignment to
- sampling interval 100ms means that the signal is transferred only after ≥ 0,5s.
- Sampling interval 200ms means that the signal is transferred only after ≥ 0,6s.
- Sampling interval 400ms means that the signal is transferred only after ≥ 0,8s.
- Sampling interval 800ms means that the signal is transferred only after ≥ 0,8s.

Parameter	Description	Range	Default
Delay	Switch-on and off delay time	0...999 999 [s]	0

12.5 EXOR (exclusive OR gate)





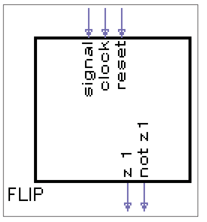
$$z_1 = d_1 \ \ EXOR \ \ d_2$$

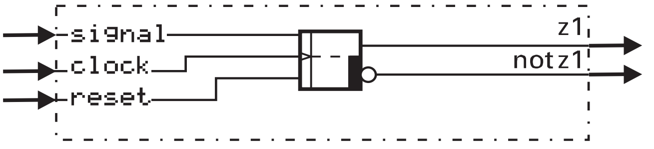
Logic inputs **d1** and **d2** are combined into **z1** according to the truth table given below. Unused inputs are interpreted as logic 0.

Output **z1** is 0, when the two inputs are equal (both 0 or both 1).

d1	d2	z1	not z1
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

12.6 FLIP (D flipflop)





The digital signal status at static input signal is transferred to output z1 when

- the signal at clock input clock changes from 0 to 1 (positive flank), and
- when input reset is logic 0.

With reset = 1, output z1 is forced to 0 independent of inputs signal and clock.

reset has priority!

Input signals signal, clock and reset must be available at least for the duration of sampling interval T_r selected for this block (100, 200, 400 or 800ms).
In the switch-on status (initial condition), z1 = 0!
Unused inputs are interpreted as logic 0.

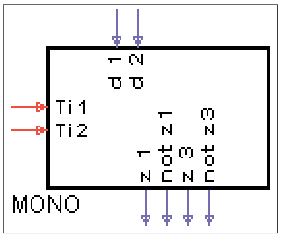
This function has a "memory". This means: after power-on, it continues operating with the statuses at z1 and not z1, which existed at power-off, provided that the RAM data are still unchanged.

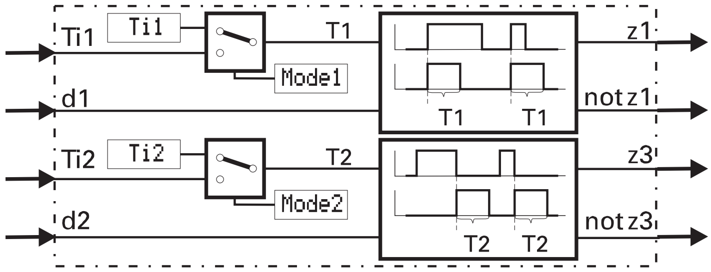
Inputs/outputs

Digital inputs	
signal	D input - This signal is output via z1 by the positive flank (0 → 1) of clock, when reset is not 1.
clock	Clock input - A positive flank transfers the instantaneous status at input Signal to output z1, when reset is not 1.
reset	Reset input - sets z1 to 0

Digital outputs	
z1	Flipflop output
not z1	Flipflop output NOT z1

12.7 MONO (monoflop)





The function generates a positive pulse of length Ti_1 at output z1, when a positive flank at trigger input d1 is detected. It generates a positive pulse of length Ti_2 at output z3, when a negative flank at trigger input d2 is detected.

Pulse duration Ti_1 is adjusted either as parameter Ti1 or read in via inputs Ti1. The origin of pulse duration is selected via parameter Mode 1. The duration of an output pulse is matched to the new values with changes at inputs Ti1/Ti2. With input values Ti1/Ti2 ß 0, the pulse is output for the duration of one scanning cycle.

The function is re-triggerable. I.e., if a new trigger condition is detected during a pulse output, the remaining pulse time to be output is prolonged to a full pulse length. The pulse duration accuracy is dependent of the sampling time, which is assigned to the function.

Example:
Ti = 0,9s for assignment to
- sampling interval 100ms means that the signal is output during = 0,9s.
- Sampling interval 200ms means that the signal is output during = 1,0s.
- Sampling time 400ms means that the signal is output during = 1,2s.
- Sampling interval 800ms means that the signal is output during = 1,6s.

Inputs/outputs

Digital inputs	
d1	Triggerinput: Pulse at z1 and not z1 with positive flank 0 → 1
d2	Triggerinput: Pulse at z3 and not z3 with positive flank 1 → 0

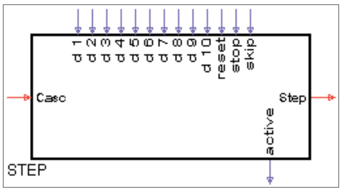
Analog inputs	
Ti1	Pulse duration Ti [s] of the pulse generated by d1, when Mode 1 = Para.Ti1.
Ti2	Pulse duration Ti [s] of the pulse generated by d2, when Mode 2 = Para.Ti2.

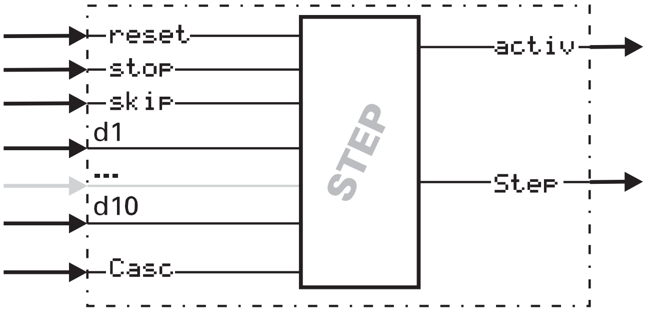
Digital outputs	
z1	Positive pulse of length Ti, when a positive flank at input d1 was detected.
not z1	Negative pulse of length Ti, when a positive flank at input d1 was detected.
z3	Positive pulse of length Ti, when a positve flank at input d2 was detected.
not z3	Negative pulse of length Ti, when a positive flank at input d2 was detected.

Parameters:

Parameter	Description		Range	Default
Mode 1	Source of pulse duration at z1	Parameter Ti1	Para.Ti1	←
		Input Ti1	Input Ti1	
Mode 2	Source of pulse duration at z3	Parameter Ti2	Para.Ti2	←
		Input Ti2	Input Ti2	
Ti1	Duration of the pulse generated by d1 ,when Mode 1 = Para.Ti1 is entered.		0,1...999 999 [s]	1
Ti2	Duration of the pulse generated by d2, when Mode 2 = Para.Ti2 is entered.		0,1...999 999 [s]	1

12.8 STEP (step function for sequencing)





The STEP function realizes the individual steps for sequencing.
The function starts with RESET at step 1 and remains at this step, until the relevant condition input d_1 or the skip input is set from 0 to 1. This is followed by switch-over to step 2. The procedure for all further steps is identical.
The step number is output as a value at output Step.

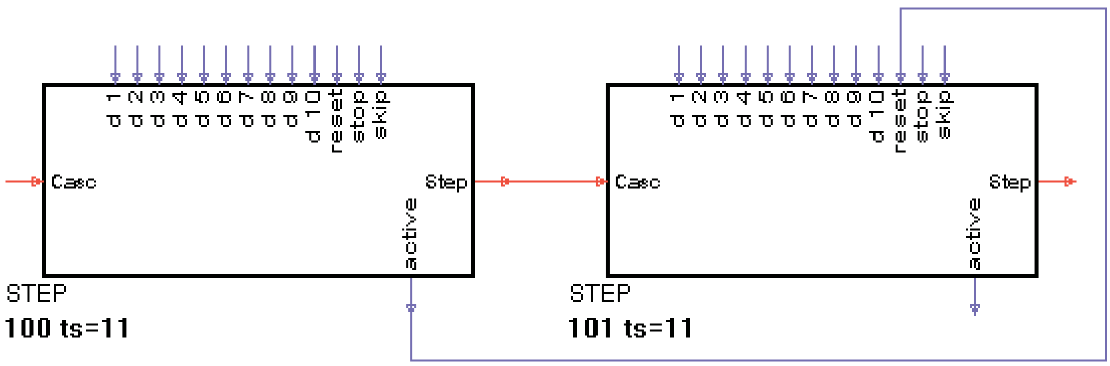
Example:
Switch-over from step 3 (Step = 3) to step 4 (Step = 4) is only after the condition at d3 was met (d3 = 1). The condition at d4 is checked only when calling up the function for the next time. Thus immediate switch-over is prevented. As long as d3 = 0, the value of output Step remains 3.
Alternatively, a positive flank at input skip also leads to switch-over to the next step (independent of the status at input d1 ..d10).

The function has a 'memory'. This means: after power-on, it continues operating with the step at power-off, provided that the RAM data are still unchanged.

When several switch-over conditions are 1 simultaneously (e.g. d1, d2, d3, d4 and d5), only the instantaneously effective input is handled. I.e. in each calculation cycle, switch-over is only by one step.

For realizing a sequencing with more than 10 steps, the STEP function can be cascaded:

The wiring example shows how 2 STEP functions are cascaded. With cascading, step number 1...n is output always as a value at output Step of the last follow-up step.



Inputs/outputs

Digital inputs	
d1...d10	Condition inputs for switching over to the next step
reset	With input reset = 1, output Step is set to 1 (only with individual function or at the first step of a cascade). With the follow-up steps of a cascade, output y = the Casc input is set. reset has the highest priority of all digital inputs.
Stop	With input Stop = 1, the function block remains in the instantaneous step (y1 and z1 remain unchanged, unless reset is switched to 1).
skip	This input reacts only to a positive flank, i.e. on a change from 0 to 1. At this flank the STEP function switches over to the next step without taking the status at the relevant d input into account.

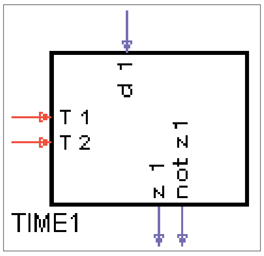
Analog input	
Casc	Used for STEP function cascading. At the first STEP function of a cascade

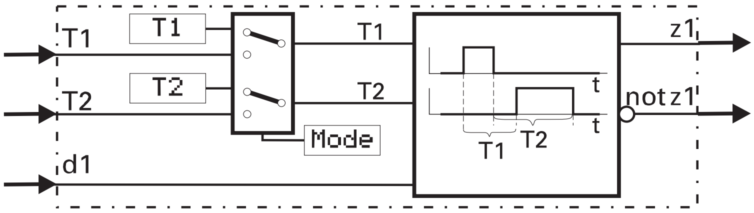
Digital output	
activ	activ =1 indicates that the STEP function is still in the active status or in reset. activ =0 indicates that the STEP function has elapsed.

Analog output	
Step	The value at Step indicates the current step of the STEP function. With cascading, the value at Casc is added to this value.

No parameters!

12.9 TIME1 (timer)





The function outputs the change of signal status at d1 with a delay at z1.
The delay time can be adjusted separately for each change direction of the signal status! (positive and negative flank).

With change from 0 to 1 at input d1, output z1 is switched to 1 with a delay of time T1. With change from 1 to 0 at input d1, output z1 is switched to 0 with a delay of time T2.

Time T1 is adjusted either as parameter T1 or read in via input T1.
Time T2 is adjusted either as parameter T2 or read in via input T2.
The time origin is selected via parameter Mode.

The pulse duration accuracy is dependent of the time group to which the function is assigned.
It is an integer multiple of the sampling interval adjusted for this block (100, 200, 400, 800ms).

Inputs/outputs

Digital input	
d1	This signal is output with a delay at output z1 and negated at output not z1.

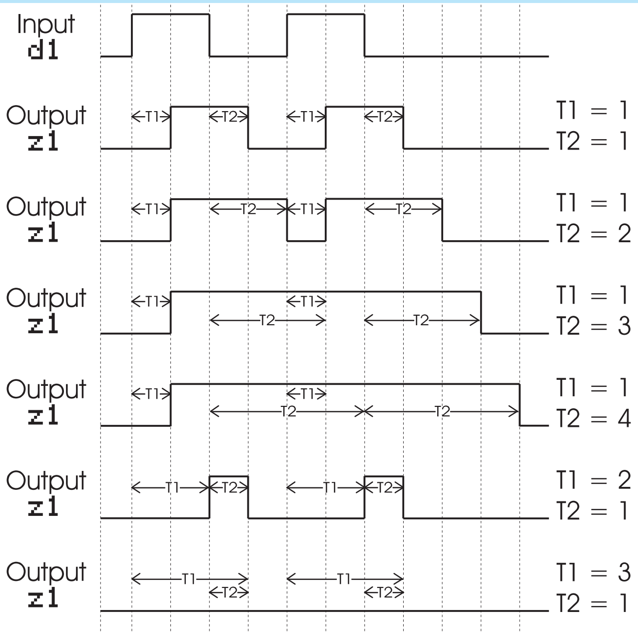
Analog inputs	
T1	Delay time T1 [s], by which the positive signal of d1 is delayed, when Mode = Inputs.
T2	Delay time T2 [s], by which the negative signal of d2 is delayed, when Mode = Inputs.

Digital outputs	
z1	Delayed input signal d1.
not z1	Inverted delayed input signal d1.

Configuration:

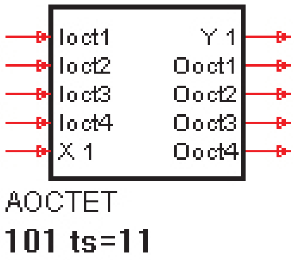
Configuration	Description		Range	Default
Mode	Source of delay times	parameters T1 and T2	Parameters	→
		Inputs T1 and T2	Inputs	

Example with different delay time T1 and T2



13 Signal converters

13.1 AOCTET data type conversion



Function AOCTET converts an analog value (X1) into the individual bytes (Ooct1-4) of a data type as used e.g. for transmission via the CAN bus (see CPREAD / CPWRIT). In the CAN notation, the bytes are transmitted in Intel format. Unless connected instruments are in compliance with this notation, word or bytewise echange of the bytes may be necessary.
The function works in both directions simultaneously (analog > bytes / bytes > analog) with separate data type adjustment in the parameters.

Analog inputs:

X1	analog input value
Ioct1..4	analog input byte value 1

Analog outputs:

Y 1	analog output value
Ooct1..4	analog output byte value 1

Parameters:

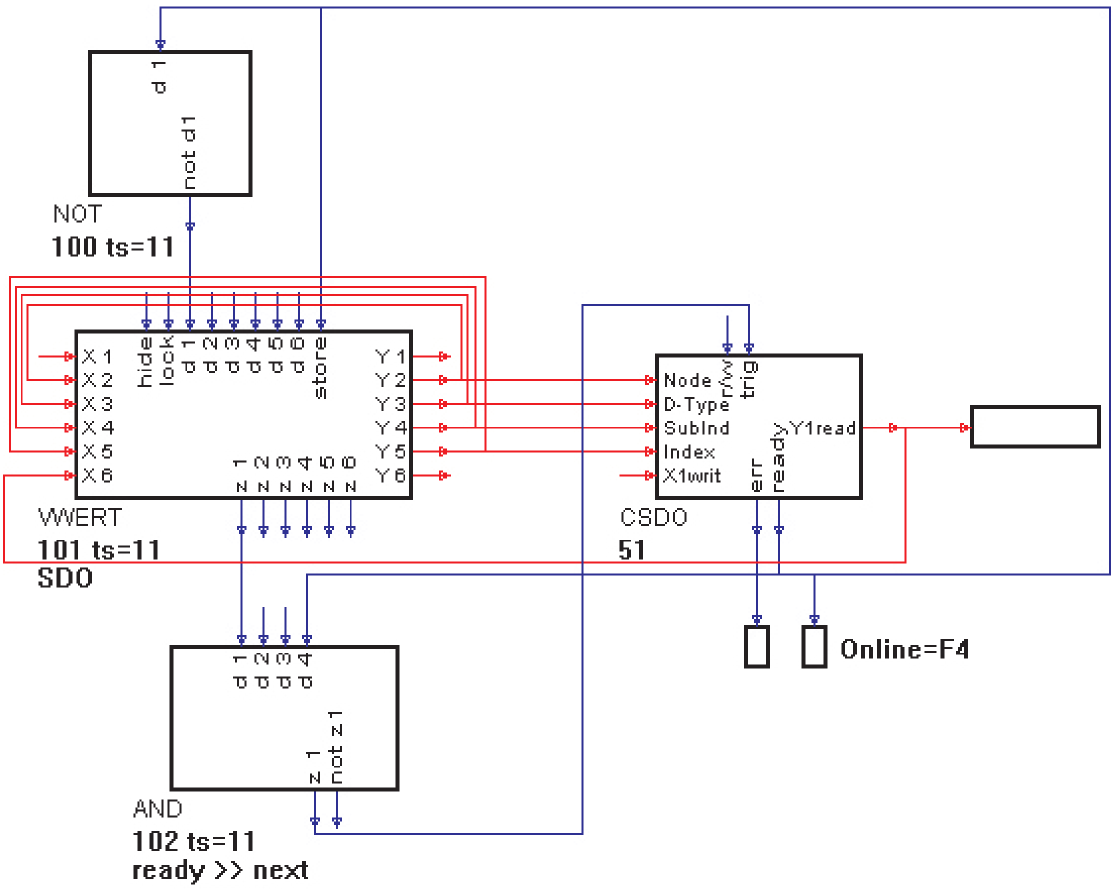
Ioct	data type of analog > byte conversion
Ooct	data type of byte > analog conversion

The following data types are available

0:	Uint8
1:	Int8
2:	Uint16
3:	Int16
4:	Uint32
5:	Int32
6:	Float

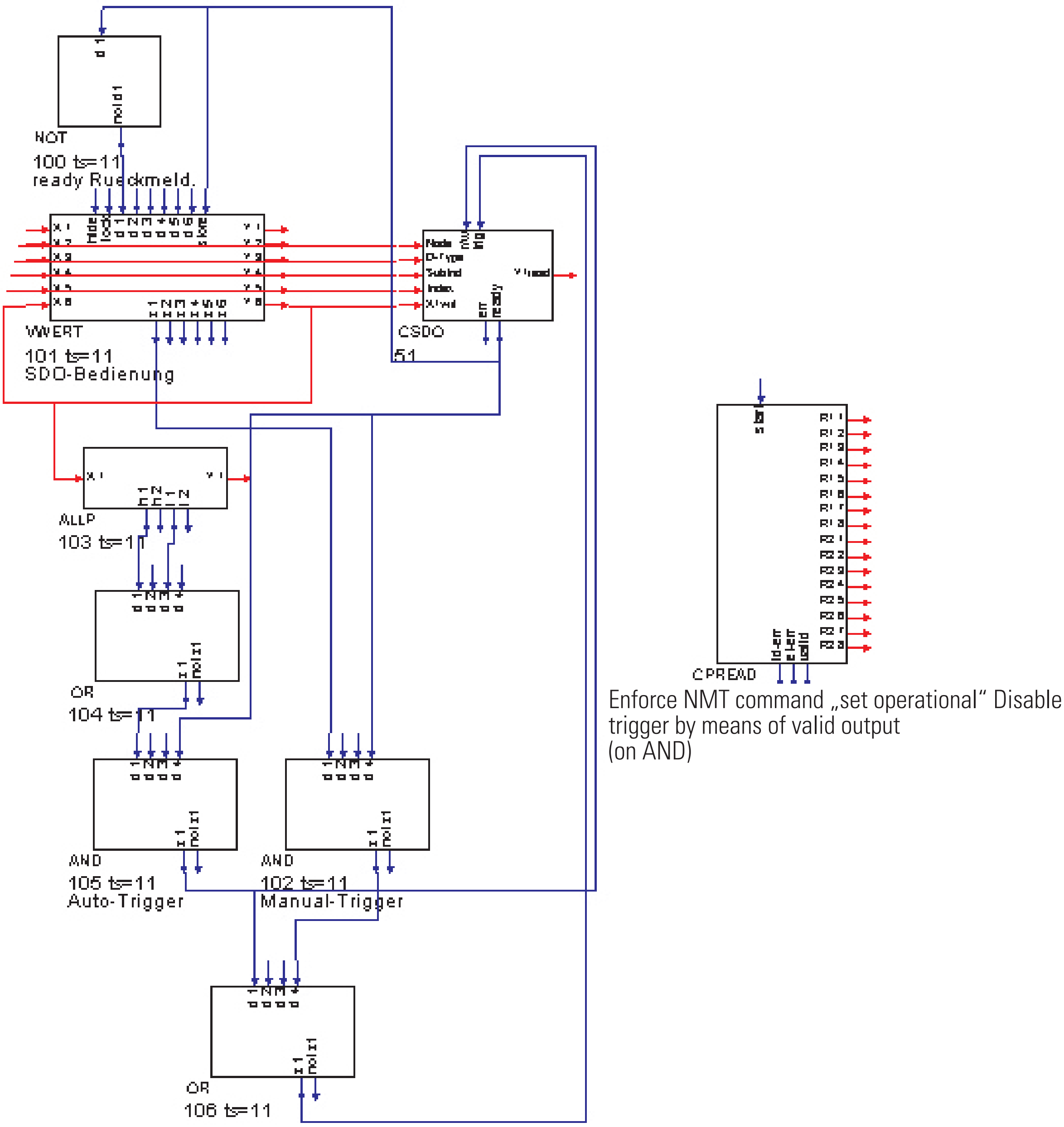
Engineering examples

SDO for data reading



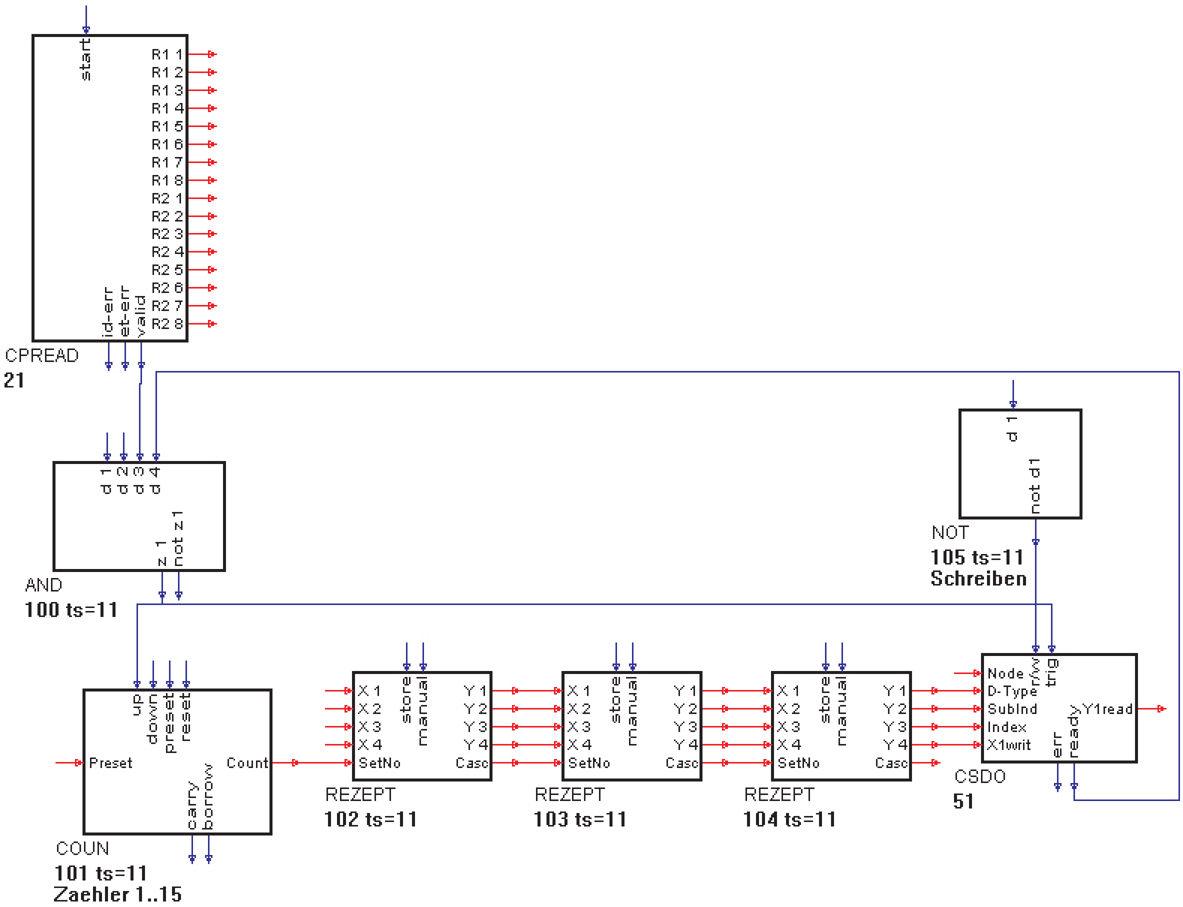
This example shows a possibility for data reading via an SDO access. Node address, data type, index and sub-index can be adjusted on an operating page. On the first line, a trigger bit which is reset by the following "ready" signal of the SDO block can be set. The engineering cannot be used to put a connected instrument into "operational" condition for PDO accesses. For this purpose, NMT commands must be used (see the example given below).

SDO for data read/write with node guarding and set operational



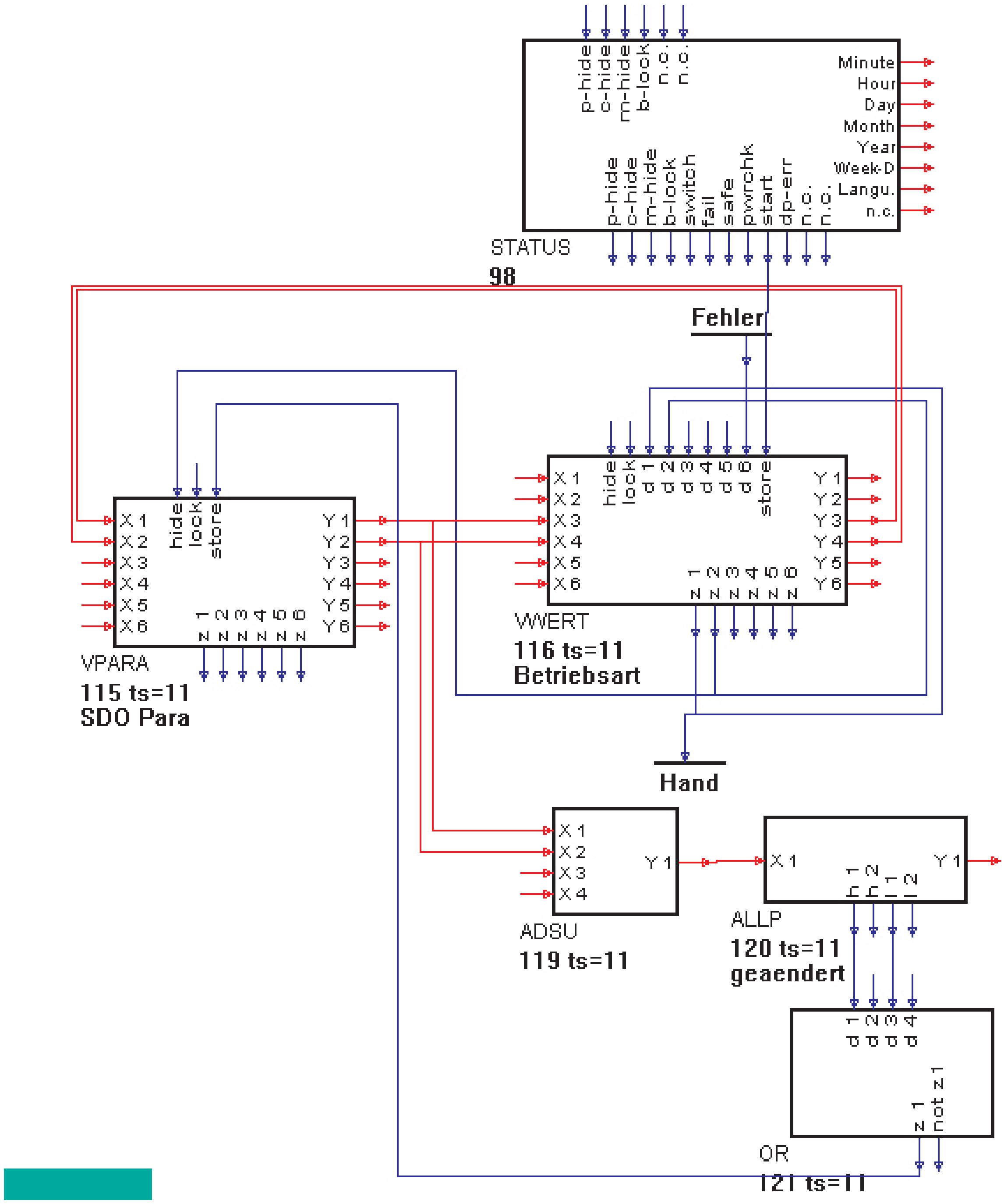
In this engineering example for data write and read via SDOs, a trigger can be set automatically when changing a value to be transmitted, or manually via the first line of the operating page. Function block CPREAD, which is used normally for reading PDOs can be used to realize node guarding for an adjustable node. Moreover, this block ensures that the selected node is set "operational". In this case, connecting the "valid" output on the AND gates may be purposeful to prevent triggering as long as the connected instrument is not ready for addressing.

Generating an SDO command sequence

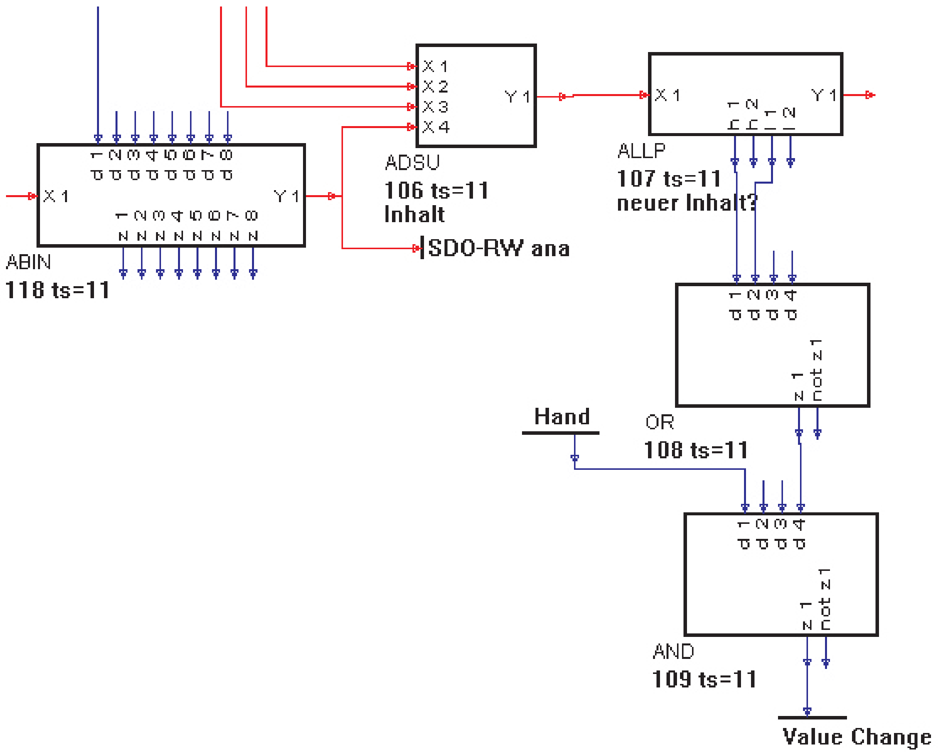


Engineering example SDO-SEQ.EDG shows the generation of an endless SDO command sequence. The values for D-type, sub-index, index and value are stored in the recipe blocks. The counter (COUN) counts from 1 to 15 continuously.

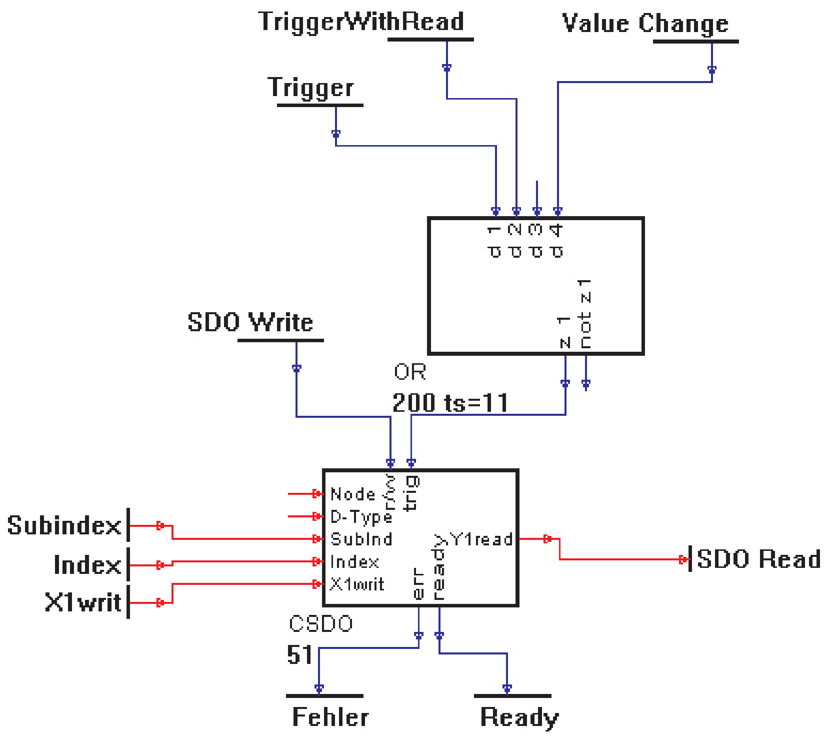
An extended engineering for advanced users SDO-SEQ2.EDG shows further functions and possibilities of KS98 engineerings in conjunction with command sequences.



This partial engineering shows the possibility of access to SDO block parameters via an operating page.

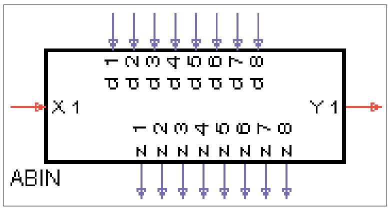


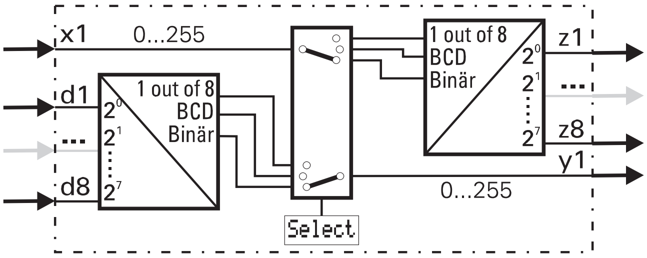
This partial function monitors the change of settings on the operating page and starts a pulse (value change) for storage in the recipe blocks.



Command triggering is subject to various conditions: when reading, after changing during manual mode and cyclically in automatic mode.

13.2 ABIN (analog ↔ binary conversion)





Analog input variable x1 is converted into a binary number, a BCD number or a selection "1 out of 8". Thereby, x1 is always rounded off (down for values < 0,5, up for values ≥ 0,5).
Simultaneously, binary input values d1 ...d8 (considered as a binary number or a BCD number) can be converted into an analog output variable.
The conversion mode is determined by configuration parameter Select.

Analog/binary conversion - binary/analog conversion (Select = ana<->bin)
Conversion analog value into binary number:
The analog input value at x1 is converted into a binary variable, which is output in binary form at outputs z1 ...z8 (z1 =2^0 ... z8=2^7). The range is within 0...255.
Out of the range, the output allocation is:

Input	z1	z2	z3	z4	z5	z6	z7	z8
x1 ≤ 0	0	0	0	0	0	0	0	0
x1 ≥ 255	1	1	1	1	1	1	1	1

Conversion binary number into analog value
A binary number at digital inputs d1 ...d8 (d1 =2^0...d8=2^7) is converted into an analog output variable and output at analog output y1 . The range is within 0...255.

BCD - conversion (Select = ana<->BCD) Converting a value into a BCD number
The analog input value at x1 (range 0...99) is output as a BCD number at outputs z8...z5 and z4...z1 .
Example: x1 = 83 ⟶ the output allocation is:

Input	z8	z7	z6	z5	z4	z3	z2	z1
x1 = 83	1	0	0	0	0	0	1	1
BCD	8				3			

Out of the range, the output allocation is:

Input	z8	z7	z6	z5	z4	z3	z2	z1
x1 ≥ 99	0	0	0	0	0	0	0	0
	0				0			
x1 ≤ 0	1	0	0	1	1	0	0	1
	9				9			

Converting a BCD number into an analog value
BCD input values at inputs d1...d4 and d5...d8 are converted into a floating point number and available at output y1.

With a BCD number > 9 at inputs d1...d4 or d5...d8, output variable y1 is limited to 9. Out of the range, the output allocation is:

Output	d8	d7	d6	d5	d4	d3	d2	d1
	0	0	0	0	0	0	0	0
y1 =	0				0			
	1	1	1	1	1	1	1	1
y1 =	9				9			

Converting a value into selection "1 out of 8" (Select = ana<->1/8)
An analog input value at x1 (range 0...8) selects none or one of the 8 outputs z1...z8.

Example for conversion value (x1 = 5) into selection:

Input	z1	z2	z3	z4	z5	z6	z7	z8
x1 = 5	0	0	0	0	1	0	0	0

Out of the range, the output allocation is:

Input	z1	z2	z3	z4	z5	z6	z7	z8
x1 ≤ 0	0	0	0	0	0	0	0	0
x1 ≥ 8	0	0	0	0	0	0	0	1

Conversion Selection "1 out of 8" into analog value (Select = ana<->1/8)
Individual digital input allocation d1...d8 result in an analog output variable at y1 according to the allocated input value.

Example for conversion value (x1 = 5) into selection:

Output	z1	z2	z3	z4	z5	z6	z7	z8
y1 = 5	0	0	0	0	1	0	0	0

If more than one of inputs d1...d8 is active, output variable y1 is set to 0.

Inputs/outputs

Digital inputs	
d1...d8	Digital inputs for binary value, BCD value or selection 1 out of 8.

Analog input	
x1	Analog input for binary value, BCD value or selection 1 out of 8.

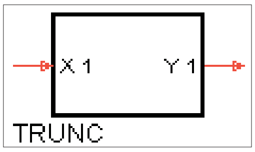
Digital outputs	
z1...z8	Converted binary value, BCD value or value selection.

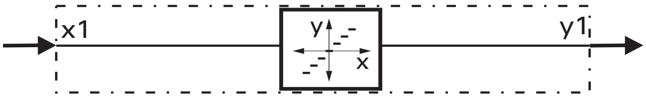
Analog output	
y1	Converted analog value.

Configuration:

Configuration	Description		Range	Default
Select	Mode of conversion	analog/binary conversion and binary/analog conversion	ana<->bin	←
		Analog/BCD conversion and BCD/analog conversion	ana<->BCD	
		Selection 1 out of 8	ana<->1/8	

13.3 TRUNC (integer portion)





$$y_1 = INT(x_1)$$

The function provides the integer portion (integer) of input variable x1 without rounding off at output y1.

Example:

x1 = 1,7 → y1 = 1,0
x1 = -1,7 → y1 = -1,0

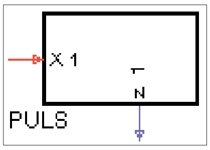
Inputs/outputs

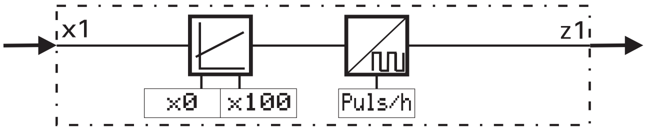
Analog input	
x1	Input variable to be handled

Analog output	
y1	Integer portion of x1

No parameters!

13.4 PULS (analog pulse conversion)





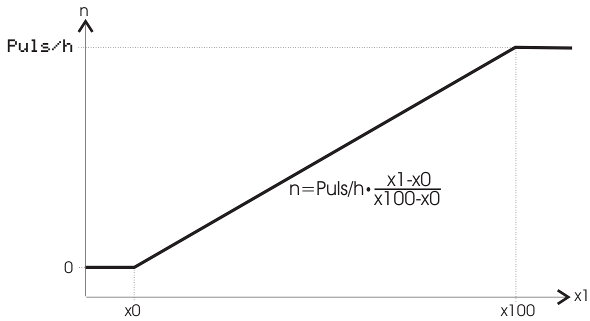
$$n = Puls/h \cdot \frac{x_1 - x_0}{x_{100} - x_0}$$

n = Number of pulses per hour
x0 = Parameter
x100 = Parameter
x1 = Analog input

Input variable x1 is converted into a number of pulses per hour. Parameter Puls/h is used for selecting the maximum number of pulses at x1 ? x100. For x1 ß x0 no pulses are output.

Within range x0 - x100 , input value x1 is converted linearly into pulses per hour.

Puls/h= maximum Pulse/h
x0 = 0% of Puls/h
x100 = 100% of Puls/h



The parameter settings result in a straight line between 0 and 3600 pulses /h according to input x1The pulse length corresponds to the sampling interval (100, 200, 400 or 800ms) adjusted for this block. The length of switch-off time between pulses is not always equal and dependent of the configured sampling interval.

The sampling interval allocation also determines the maximum number of pulses/hour, which can be realized. If higher values than can be output due to the sampling interval are entered in parameter Puls/h, limiting is to the maximum possible number of pulses.

Maximum number of pulses:		
100 ms	=	18 000 pulses/h
200 ms	=	9 000 pulses/h
400 ms	=	4 500 pulses/h
800 ms	=	2 250 pulses/h

Inputs/outputs

Analog input	
X1	Input variable to be converted

Digital output	
z1	Pulse output

No configuration parameters

Parameters:

Parameter	Description	Range	Default
x0	Span start (0 %)	-29 999...999 999	0
x100	Span end (100 %)	-29 999...999 999	1
Puls/h	Number of output pulses per hour for *x1 ? x100.*	0...18 000	

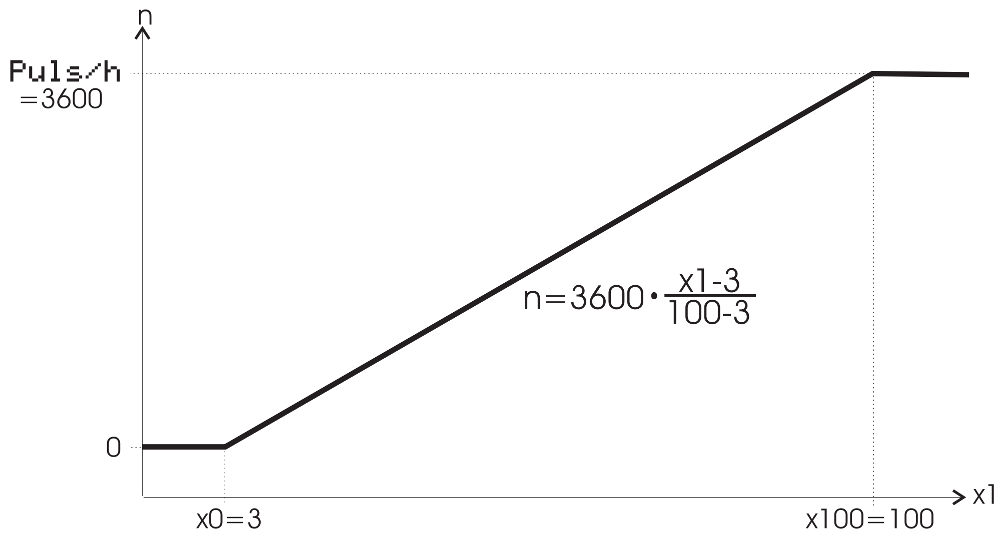
Equation for calculating the momentary impulse number of n per hour

$$n = Puls/h \cdot \frac{x_1 - x_0}{x_{100} - x_0}$$

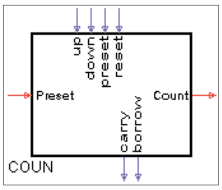
n	=	momentary impulse number/hour
x0	=	Parameter. With analog input x1 $\leq$ x0 no pulses are produced (area start, creeping flow suppression)
x100	=	If the analog input is x1 $\leq$ x100 n remains = constant = Puls/h
Puls/h	=	Parameter. Pulse number/hour for analog input $x_1 = x_{100}$

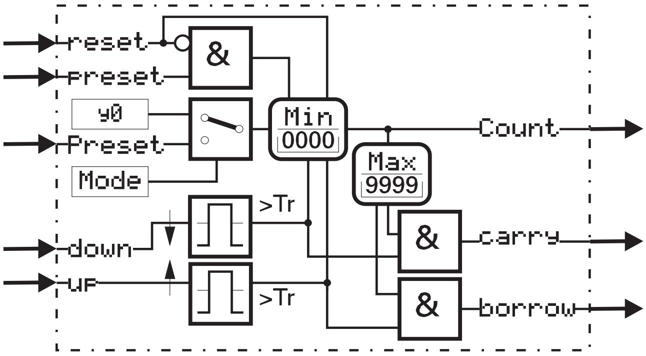
Example:
x1 = 3...100% ≙ 0...3600/h

$x_0 = 3$
$x_{100} = 100$
Puls/h = 3600
sampling period $\leq$ 400 ms



13.5 COUN (up/down counter)

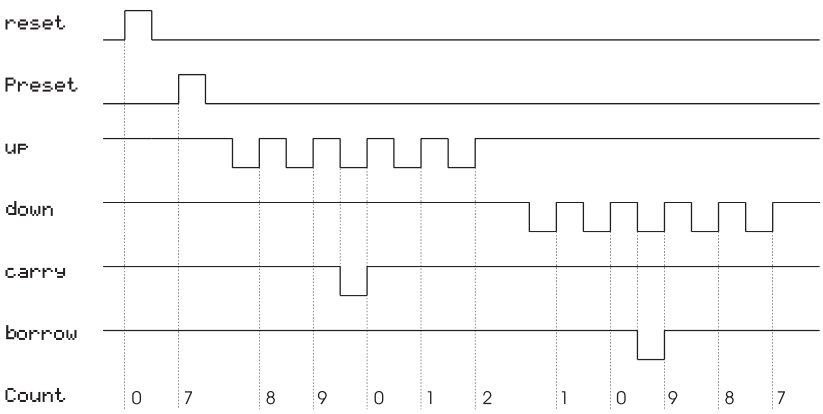




'COUN' is an up/down counter and counts the events at input up or down, which are available at the up or down input for at least the duration of the time group in which the function runs.

reset	preset	Mode
0	0	GO (default)
0	1	Preset
1	0	Reset (first run)
1	1	Reset (first run)

Pulse diagram of the up/down counter:



Example:
max. limit = 9; min. limit = 0; Preset = 7.
An unwired clock input is set to value 1 internally. If both clock inputs go from 0 to 1 signal simultaneously, counting is omitted. If one of clock inputs (up or down) are set from 0 to 1 signal, without the other one being already set to 1, counting is omitted.

If parameters for the min. or max. limit are changed during operation, the counter can be out of this new range. In order to prevent faulty functions, the counter must be set to a new, defined output status with 'reset' or 'preset'. The function has a 'memory'. This means: after power-on, it continues operating with counter state and internal states at power-off, provided that the RAM data are still unchanged.

Function up counter:

At each positive flank (0 ⟶ 1) at input up, output Count is increased by 1, until the max. limit is reached. Carry output carry is set to 0 for the duration of the applied pulse. With the next pulse, output Count returns to the min. value and continues counting with the next pulses.

If the down-input is wired, the up counter is prepared by signal 1 at input down. If not , counting is not possible. I.e. there must be a 1 signal at input down prior to input up, if the pulse shall be counted.

Function down counter:

With each positive flank (0 ⟶ 1) at input down, output Count is decreased by 1, until the min. limit is reached. Subsequently, borrow output borrow is set to 0 for the duration of the applied pulse 0. With the next pulse, output Count returns to the max. value and continues counting down with the next pulses.

If the up-input is wired, the down counter is prepared by signal 1 at input up. If not , counting is not possible. I.e. there must be a 1 signal at input up prior to input down, if the pulse shall be counted.

Function reset:

A 1 signal at input reset has priority over all other inputs. reset resets the count to the min. value.

Function preset:

A 1 signal at input preset has priority over inputs up and down. preset resets the count to the preset value.
The origin of the preset value is selected with parameter Mode.

- Mode = Para.y0 means that the preset value corresponds to parameter y 0.
- Mode = InpPreset means that the preset value corresponds to analog input Preset.

With a preset value higher than the max. limit, output Count is set to the max. limit. A preset value smaller than the min. limit is set to the min. limit. A preset value which is not an integer is rounded off.

Inputs/outputs

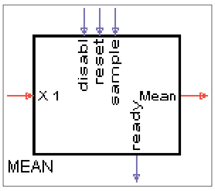
Digital inputs	
up	Input for clock up - pulse count up
down	Input for clock down - pulse count down
preset	Input for the preset mode - output Count goes to value Reset.
reset	Input for the reset mode - output Count goes to value Min.

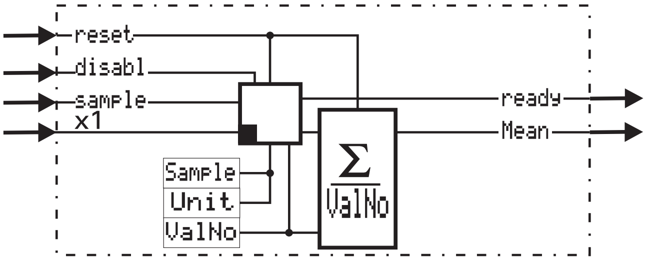
Analog input	
Preset	Analog input for external preset value

Digital outputs	
carry	Carry output (clock - up)
borrow	Borrow output (clock - down)

Analog output	
Count	Count output

13.6 MEAN (mean value formation)





General

Function MEAN forms the floating, arithmetic mean value of the number (ValNo) of the values detected last at input x1 for output at output y1.
The interval between the individual samplings (interval) is adjustable with Sample and Unit.
Unit is used to specify the measurement interval (sec = seconds, min = minutes or h = hours).
Sample is used to specify the number of 'Unit` intervals for measurement.

With the sample input wired, the adjusted sample and unit parameter are ineffective.
Only the sample-pulse is used

Example 1: mean value of the past minute with sampling per second.

Sample = 1 and Unit = sec → value sampling per second.
ValNo = 60 → the past 60 values form the mean value (1 minute).

Example 2: mean value of the past day with sampling per hour.

Sample = 1 and Unit = h → value sampling per hour.
ValNo = 24 → the past 24 values for the mean value (1 day).

Example 3: mean value of the past day with sampling per quarter of an hour.

Sample = 15 and Unit = min → value sampling at intervals of 15 minutes.
ValNo = 96 → the past 96 values form the mean value (1 day).

If the sample input is wired, sampling is triggered by a positive flank at this input.
The adjusted sampling interval is invalid.

disabl = 1 interrupts the sampling, reset = 1 deletes the mean value.

Internal calculation:

The number of input values entered in ValNo is stored, totalized and divided by the number.

$$y1 = \frac{value\_1 + value\_2 + value\_3 + ...value\_n}{n}$$

Example: ValNo = 5

11	24	58	72	12

$$y1 = \frac{11 + 24 + 58 + 72 + 12}{5} = 35,4$$

reset

Analog input Mean goes to value 0 for the duration of the applied reset signal.
The stored values are deleted.

Example:

ValNo = 5 output Mean at reset:

x_1=	x	x	x	x	x

Detection that no valid values are available is made. Value 0 is output at output y1.

ValNo = 5 1st sample after reset:

x1=	55	x	x	x	x

Detection that only one valid valid value is available is made. The only valid value y1 = 55 is available at output y1.

ValNo = 5 2nd sample after reset:

x1=	44	55	x	x	x

Detection that two valid values are available is made. The mean value of these valid values y1 =49,5 is output at output y1.

After all memory cells with a value are occupied (ValNr = 5), with every sample a new input value is added, the at this time oldest value subtracted and the result divided by ValNr. = 5. The input values are shifted (like with a shift register).

Inputs/outputs

Digital inputs	
disabl	The disable input interrupts sampling
reset	The reset input clears the memory and resets the mean value to 0.
sample	A positive flank (0 ⟶ 1) is used for sampling a new value.

Analog input	
x1	Process value, of which the mean value is formed.

Digital output	
ready	Display for an elapsed overall cycle

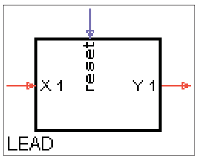
Analog output	
Mean	Calculated mean value

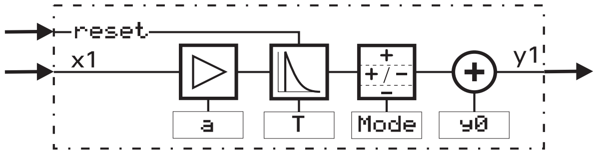
Configuration:

Parameter	Description		Value	Default
ValNo	Number of values which can be aquired		1...100	100
Unit	Unit of time for “Sample”	Seconds	sec	←
		Minutes	min	
		Hours	h	
Sample	Interval time for averaging		0,1...999 999	1

14 Time functions

14.1 LEAD (differentiator)





The differentiator forms the difference quotient according to equation:

$$y_1(t) = \frac{T}{T+T_s} \cdot \left[y_1(t-t_s) + a \cdot \{ x_1(t) - x_1(t-t_s) \} \right] + y_0$$

ts	sampling interval	x1(t)	instantaneous x1
T	time constant	x1(t-ts)	previous x1
a	gain	y1(t)	instantaneous y1
y0	output offset	y1(t-ts)	previous y1

$$C = \frac{T}{T+Ts} < 1$$ (differentiation constant)

The complex transfer function reads: $F_{(p)} = \frac{a \cdot T \cdot p}{T \cdot p + 1}$

Inputs/outputs:

Digital input	
reset	= 1 causes that y1= y0 and the difference quotient is set to 0. = 0 starts differentiation automatically.

Analog input	
x1	Input variable to be differentiated

Output	
y1	Differentiator output

Parameters:

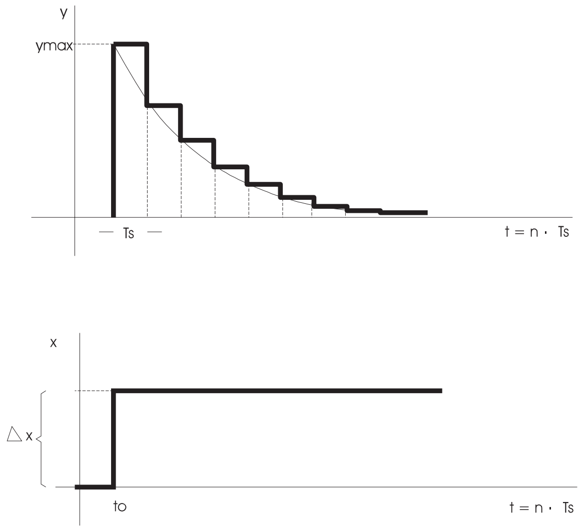
Parameters	**Description**	Range	**Default**
a	Gain factor	-29 999...999 999	1
y0	Output offset	-29 999...999 999	0
T	Time constant in s	0...199999	1

Configuration:

Configuration	**Description**		**Value**	**Default**
Mode	Differentiator operation	Differentiating all changes	0	0
		Differentiating only positive changes dx/dt >0	1	
		Differentiating only negative changes dx/dt <0	2	

Step response:

After a step change of input variable x1 by {x =xt-x(t-ts), the output changes to maximum value y max.



$$y_{max} = C \cdot a \cdot \Delta x + y0$$

and decays to 0 according to function

$$y(n.ts) = C_n \cdot \Delta.x + y0 = ymax \cdot C^{(n-1)}$$

Thereby, n is the number of calculation cycles ts after the input step change. Number n of required calculation cycles ts until output variable decaying to y(n*Ts) is

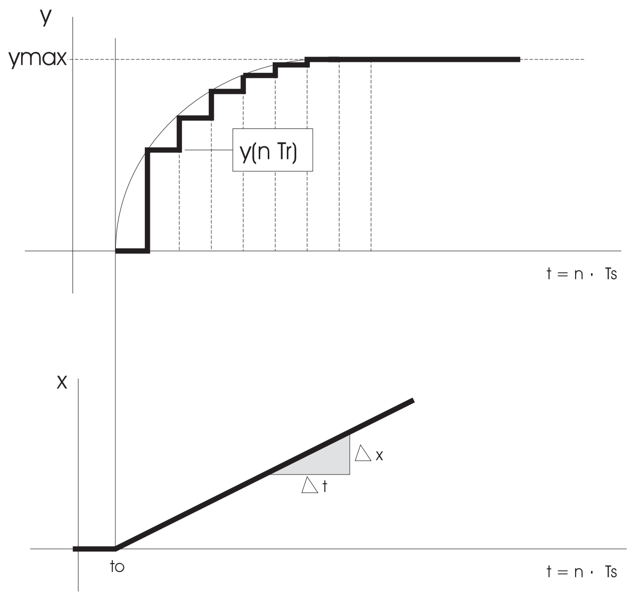
$$n = \frac{\lg \frac{y(n \cdot ts)}{ymax}}{\lg C} + 1$$

Surface area A under the decaying function is

$$A = y_{max} \cdot (T + ts)$$

Ramp response:

After ramp starting, output variable y runs towards the final value of differentiation quotient



$$y_{max} = m \cdot a \cdot T$$

according to function

$$y(n \cdot ts) = m \cdot a \cdot T \cdot (1 - C^n)$$

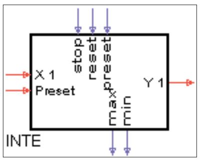
Thereby, $m = m = {}^{dx}\!/_{dt}$ is the gradient factor of the input function. Relative error F after n calculating cycles Ts referred to the final value is calculated as follows:

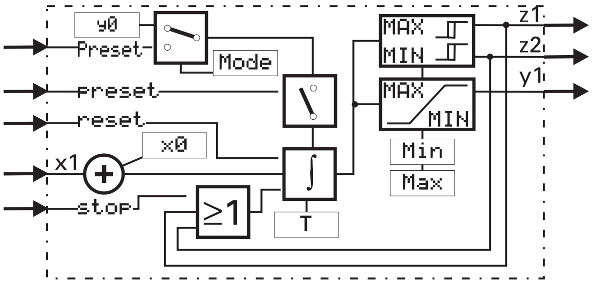
$$F = C^n$$

and the number of required calculating cycles, according to which function $y(n \cdot ts)$ approaches final value y= y_{max} to error F is

$$n = \frac{\lg F}{2 \cdot \lg C}$$

14.2 INTE (integrator)





The integrator forms the integral according to equation:

$$y_1(t) = y_1(t - t_s) + \frac{t_s}{T} \cdot \left[x_1(t) + x_0\right]$$

ts	sampling interval	x1(t)	instantaneous x1
T	Integration constant	y1(t)	y1 after t=n*ts
n	Number of calculation cycles	y1(t-ts)	previous y1
x0	Input offset		

The complex transfer function is:

$$F(p) = \frac{1}{T \cdot p}$$

Unused control inputs are interpreted as logic “0". With simultaneous input of several control commands:
reset = 1 has priority over preset and stop
preset = 1 priority over stop

Integrator output y1 is limited to the preset limits (Min, Max): Min ≤ y1 ≤ Max. When exceeding Min or Max, the integrator is stopped automatically and the relevant control output min or max is set to logic 1. Limit value monitoring uses a fixed hysteresis of 1 % referred to operating range (Max - Min)

Inputs/outputs

Digital inputs	
stop	= 1 The integrator is stopped for the duration of the stop command. Output *y1* does not change.
reset	= 1 The integration result is adjusted to lower limit (Min). After cancelation of reset, integration starts at lower limiting.
preset	= 1 The integration result is set either to a preset value y0 (Mode=0) or to a preset variable Preset (Mode= 1). After cancelation of the preset command, integration starts with the actually effective preset value.

Analog inputs	
x1	Input variable to be integrated
Preset	External preset value

Digital outputs	
max	= 1 exceeded with max. limiting
min	= 1 exceeded with min. limiting

Analog output	
y1	Integrator output after elapse of integration $t = n \cdot ts$

Parameters:

Parameter	Description	Range	Default
T	Time constant in s	0.1...999 999	60
x0	Constant	-29 999...999 999	0
y0	Preset value	-29999...999 999	0
Min	Min. limiting	-29999...999 999	1
Max	Max. limiting	-29999...999 999	0
Mode	Source of preset = Para y0	0	0
	Source of preset = InpPreset	1	

Ramp function:

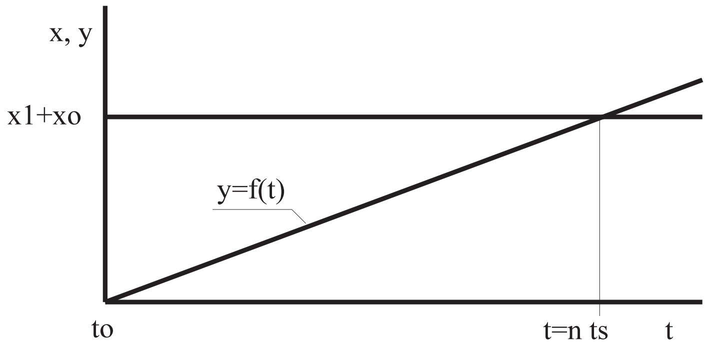
With constant input x1+x0, the applicable formulas are

$$y1(t) = y(t0) + n \cdot \frac{ts}{T} \cdot (x1 + x0)$$

$$t = n \cdot ts$$

t is the time required by the integrator for changing output y1 linearly by value x1 + x0 after integration start.

Ramp response:



The function has a 'memory'. This means: after power-on, it continues operating with values y1, z1 and z2, which existed at power-on, provided that the RAM data are still available.

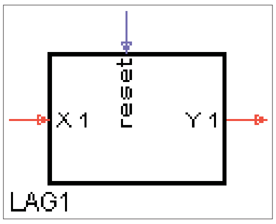
Example: Which is the value of output variable y after t=20s with a time constant of 100s, if a constant of x1 = 10 Volt is preset. Sampling interval ts is 100ms.

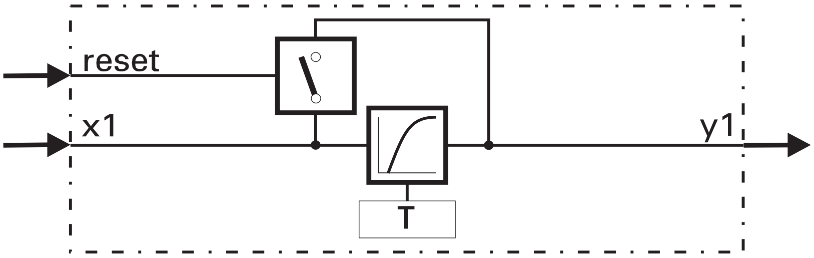
$$n = \frac{t}{t_s} \qquad n = \frac{20s}{0{,}1s} = 200s$$

$$y = 0 + 200 \cdot \frac{0{,}1}{100} \cdot 10 = 2 \quad \text{after} \quad 20s$$

This results in a gradient of 2V/20s or 0.1Volt /1s.

14.3 LAG1 (filter)





Dependent of control input reset, input variable x1 is passed on to output y1 with delay (reset= 0) or without delay (reset = 1). Delay is according to a 1st order e-function (1st order low pass) with time constant T(s). The output variable for reset= 0 is calculated according to the following equation:

$$y_1(t) = \frac{T}{T+t_s} \cdot y_1(t-t_s) + \frac{t_s}{T+t_s} \cdot x_1(t)$$

ts	sampling interval	x1 (t)	instantaneous x1
T	time constant	x1(t-ts)	y1 after t $= n \cdot$ ts
n	number of calculation cycles	y1(t-ts)	previous y1

The complex transfer function is:

$$F(p) = \frac{1}{1+p \cdot t}$$

Inputs/outputs:

Digital input	
reset	= 0 means that input signal *x1* is output without delay at output *y1*. = 1 means that input signal *x1* is output at output *y1* according to the calculated e-function.

Analog input	
x1	Input variable to be calculated

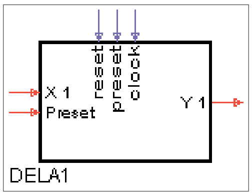
Analog output	
y1	Delayed output variable

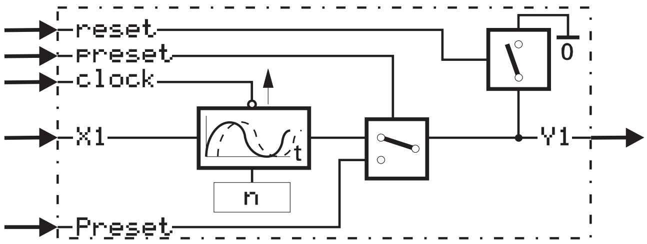
Parameter:

Parameter	Description	Range	Default
T	Time constant in s	0...199999	1

No configuration parameters!

14.4 DELA1 (delay time)





If the clock input is not wired, the function calculates y1(t) = x1 (t-n • ts).
(ts = sampling interval, Delay = delay factor n)

Unless clock input clock is wired, the following is applicable: input variable x1 is output with a delay by n times the amount of adjusted sampling interval ts (phase shift by n · ts). The effective delay time corresponds to integer multiples of the selected time group (sampling interval ts 100/200/400/800 ms). The delay time range covers n= 0 to 255 (0....255 · ts)

With clock input clock wired, DELA1 acts like a shift register with a length of max. 255=Parameter delay. This register can be switched on by one step by an external event preset .
Switching on is only with a positive flank (transition from 0 ⟶ 1) at the clock input plus the adjusted delay factor (parameter delay).

Example:
With delay = 4 change-over at output y is only after 4 flank changes from 0 ⟶ 1 at input clock

Preset: The output provides the value applied to Preset. After (n+1) positive flanks at clock or (n+1) sampling cycles t_S (if clock isn't wired), the first input value x1 appears at y1.

reset: The output provides value 0. After a positive flank at clock, value zero still is provided for the sampling interval ts.

The function has a ‘memory’. This means: after power-on, it continues operating with values y1, z1 and z2, which existed at power-on, provided that the RAM data are still unchanged.

Inputs/outputs

Digital inputs	
clock	= 0->1 clock for delaying
preset	= 1 The preset value is taken to the output
reset	= 1 Output y1 is set to zero

Analog inputs	
x1	Input variable to be delayed
Preset	Value output without delay by preset = 1

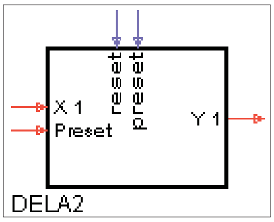
With several simultaneous control commands:
reset = 1 has priority over preset and stop
preset = 1 has priority over stop

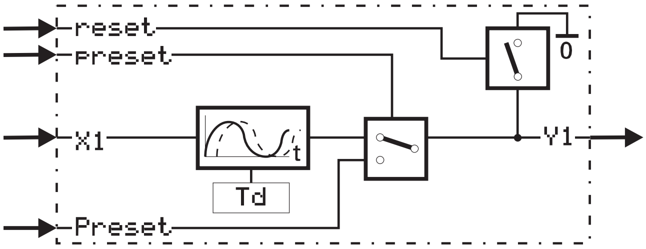
Analog output	
y1	Delayed output variable

Parameter:

Parameter	Description	Range	Default
Delay	Delay factor n	0/1/.....255	0

14.5 DELA2 (delay time)





The function provides calculation y1(t) = x1(t – Td)

Input variable x1 is output at y1 with delay by time Td. The accuracy Td is dependent of the time group (sampling interval ts) , to which the function is assigned.

Example:
Td = 0,7s with assignment
to time group 100ms means Td = 0,7s
to time group 200ms means Td = 0,8s
to time group 400ms means Td = 0,8s
to time group 800ms means Td = 0,8s
The possible delay time is dependent of the configured time slot (sampling interval ts).
Td max = 25,5s with ts = 100ms
Td max = 51,0s with ts = 200ms
Td max = 102,0s with ts = 400ms
Td max = 204,0s with ts = 800ms

Inputs/outputs

Digital input	
preset	= 1 The preset value is taken to the output
reset	= 1 Output y1 is set to zero

With several simultaneous control commands:
reset = 1 has priority over preset and stop
preset = 1 priority over stop

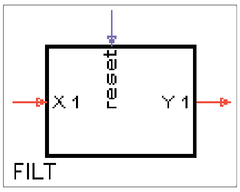
Analog input	
x1	Input variable to be delayed
Preset	Value output with delay by preset=1

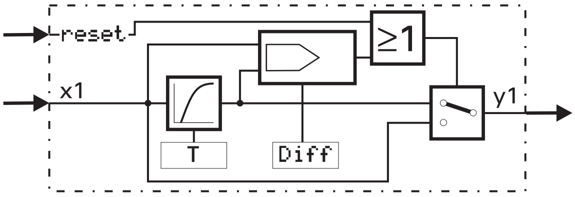
Analog output	
y1	Delayed output variable

Parameters:

Parameter	Description	Range	Default
Td	Delay in s	-29999.....199999	0

14.6 FILT (filter with tolerance band)





The complex transfer function of the filter within a tolerance band around the last output value ($| x1 - y1| <= \delta$) is:

$$F(p) = \frac{1}{1 + p \cdot T}$$

With a difference higher than **Diff** or **reset** = 1 between input x1 and output y2, the filter stage is switched off and the output follows the input directly.

With a difference of input x1 and output y1 smaller than **Diff** and **reset** = 0, the output follows an e-function with time constant T. The output variable is calculated according to the following equation:

$$y1(t) = \frac{T}{T + ts} \cdot y1\,(t - ts) + \frac{ts}{T + ts} \cdot x1(t)$$

ts	sampling interval	x(t)
T	time constant	x1(t-ts)

Inputs/outputs

Digital input		
reset	= 0 $\|x1 - y1\| <$ Diff	delay effective
	= 0 $\|x1 - y1\| >$ Diff	delay switched off
reset	= 1 $\|x1 - y1\| <=$ Diff	delay switched off
	= 1 $\|x1 - y1\| >$ Diff	delay switched off

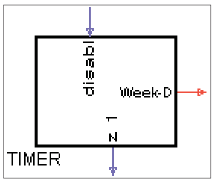
Analog input	
x1	Input variable to be delayed

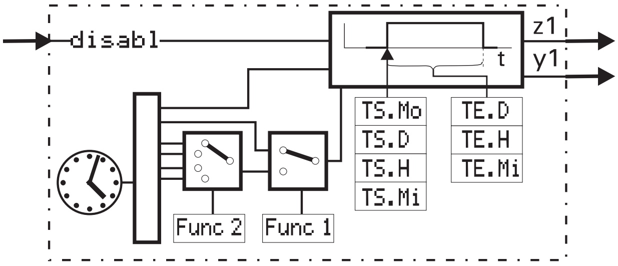
Analog output	
y1	Delayed output variable

Parameters:

Parameter	Description	Range	Default
T	Time constant in s	0...199999	1
Diff	Tolerance band ϑ	0...999999	1

14.7 TIMER (timer)





The function timer can only be used with real-time clock (9407-9xx-2xxx). Output z_1 is switched on at absolute time TS and switched off again after TE. This switching operation can be unique or cyclical (parameter adjustment). Output Week-D indicates the actual weekday (0...6 = Su...Sa) . TS.Mo = 0 and TS.D = 0 means actual day

When the time defined with TS.H and TS:Mi has elapsed, the 1st switching operation occurs on the following day. With TS.Mo = 0 and TS.D < actual day, the first switching operation occurs in the following month. With TS.Mo ≤ actual month and TS.D < actual day, the 1st switching operation occurs in the next year.

Inputs/outputs

Digital input	
disabl	= 0 output z1 active. Becomes 1 when the time was reached.
disabl	= 1 output z1 switched off. The output behaves like “time not yet reached”.

Digital output	
z1	z1 is logic 1 between the start and end time.

Analog output	
Week-D	indicates the actual weekday (0...6 = Su...Sa)

Parameters:

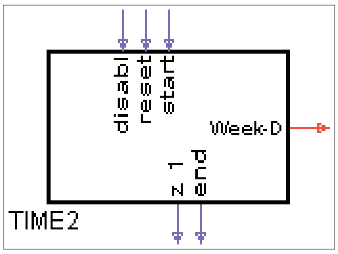
Parameter	Description	Range	Default
TS.Mo	Switch-on time month	0...12	0
TS.D	Switch-on time day	0...31	0
TS.H	Switch-on time hour	0...23	0
TS.Mi	Switch-on time minute	0...59	0
TE.D	Time duration days	0...255	0
TE.H	Time duration hours	0...23	0
TE.Mi	Time duration minutes	0...29	0

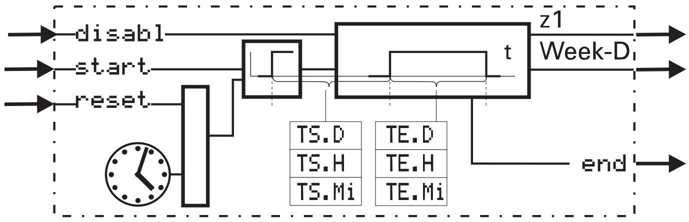
Configuration:

Configuration	Description	Value	Default
Func1	cyclical function runs cyclically	0	0
	once function runs once	1	
Func2	daily function runs daily	0	0
	Mo...Fr. Function runs from Monday to Friday	1	
	Mo... Sa function runs from Monday to Saturday	2	
	weekly function runs weekly	3	

* 1) with the engineering tool broken rational numbers can be used; however only the integral portion is taken over!

14.8 TIMER2 (timer)





The function timer2 can only be used with real-time clock (9407-9xx-2xxx).With a positive flank at start, TIMER2 is started and output z1 is switched to 1 after elapse of time TS and reset to 0 after elapse of time TE.
Example:
TS.D = 2, TS.H = 1, TS.Mi = 30 TE.D = 0, TE.H = 2, TE.Mi = 2
After the change from 0 to 1 at input start, output z1 is set to 1 after 2 days, 1 hour and 30 seconds and reset to 0 after 2 hours and 2 seconds. Cyclic switching operations can be realized by feed-back of the end output to the start input.

Inputs/outputs

Digital inputs	
disabl	= 1 suppresses the switching operation.
reset	= 1 finishes an instantaneously running switching operation immediately.
start	= 1 switch-on duration start

Digital outputs	
z1	= 1 switching operation running
end	= 1 switching operation end

Analog output	
Week-D	indicates the actual weekday (0...6 ≙ Su...Sa)

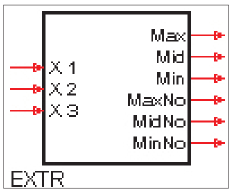
Parameters:

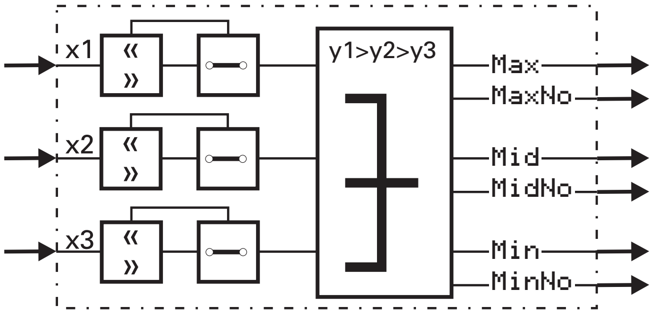
Parameter	Description	Range	Default
TS.D	Switch-on delay day	0...255	0
TS.H	Switch-on delay hour	0...23	0
TS.Mi	Switch-on delay minute	0...59	0
TE.D	Switch-on duration days	0...255	0
TE.H	Switch-on duration hours	0...23	0
TE.Mi	Switch-on duration minutes	0...29	0

* 1) with the engineering tool broken rational numbers can be used; however only the integral portion is taken over!

15 Selecting and storage

15.1 EXTR (extreme value selection)





Analog inputs x1 , x2 and x3 are sorted according to their instantaneous values and provided at outputs Max, Mid and Min. Input value output is at Max for the highest one, at Mid for the medium one and at Min for the smallest one.
The number of the input with the highest value is output at MaxNo.
The number of the input with the medium value is provided at output MidNo.
The number of the input with the smallest value is provided at output MinNo.

With equality, the distribution is at random.

Inputs are <u>not</u> included into the extreme value selection, if:

-the input is not wired
-or the input value is higher than $1{,}5 \cdot 10^{37}$ or smaller than $-1{,}5 \cdot 10^{37}$.

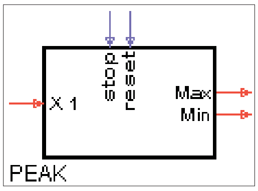
Number of failed inputs	Max	Mid	Min	MaxNo	MidNo	MinNo
0	x_{max}	x_{mid}	x_{min}	number of x_{max}	number of x_{mid}	number of x_{min}
1	x_{max}		x_{min}	number of x_{max}		number of x_{min}
2	the valid value			number of the valid value		
3	1,5 . 1037	1,5 . 1037	1,5 . 1037	0	0	0

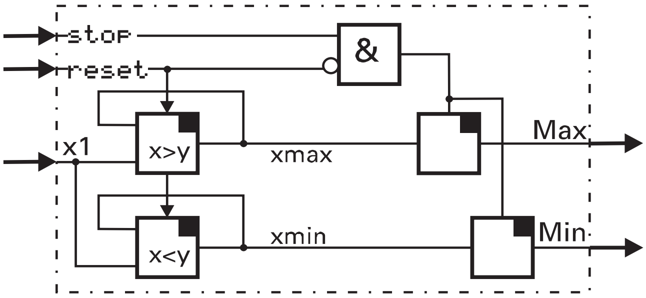
Inputs/outputs

Analog inputs	
x1 . . . x3	Input variables to be compared

Analog outputs	
Max	Maximum instantaneous input value
Mid	Mean instantaneous input value
Min	Minimum instantaneous input value
MaxNo	Number of maximum instantaneous input value (1 = x1 , 2= x2, 3= x3)
MidNo	Number of mean instantaneous input value (1 = x1 , 2= x2, 3= x3)
MinNo	Number of minimum instantaneous input value (1 = x1, 2= x2, 3= x3)

15.2 PEAK (peak value memory)





Maximum input value x_{max} and minimum input value x_{min} are determined, stored and output at Max and Min. With the stop input set to 1, the extreme values determined last remain unchanged.
If the reset input is set to 1, the extreme value memory and any applied stop command are cancelled. (x_{max} and x_{min} are set to the instantaneous x1 value and follow input x1, until the reset input returns to 0.)

Unused inputs are interpreted as 0 or logic 0.

The function has a 'memory'. This means: after power-on, it continues operating with the Min- and Max values which existed at power-off, provided that the RAM data are still unchanged.

No parameters!

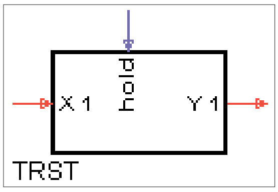
Inputs/outputs

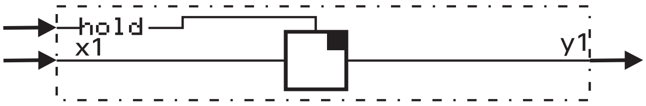
Digital inputs	
stop	With the stop input set to 1, instantaneous values Max and Min are unchanged.
reset	The reset input deletes the Min and Max values.

Analog inputs	
x1	Process value, the min and max values of which are output.

Analog outputs	
Max	Maximum value
Min	Minimum value

15.3 TRST (hold amplifier)





With control input hold set to 1, instantaneous input value x1 is stored and output at y1. With control input hold set to 0, output y1 follows input value x1.

The function has a ‘memory’. This means: after power-on, it continues operating with the y1 value which existed at power-off, provided that the RAM data are still unchanged.

No parameters!

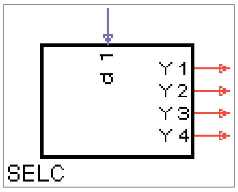
Inputs/outputs

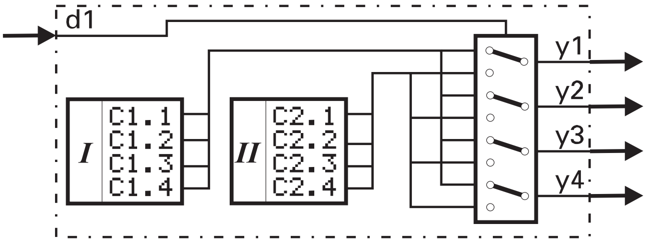
Digital input	
hold	Storage signal for the value

Analog input	
x1	Process value which can be output stored.

Analog output	
y1	Function output

15.4 SELC (Constant selection)





Dependent of control signal d1, the four preset parameters of group 1 or of group 2 are output.

Inputs/outputs

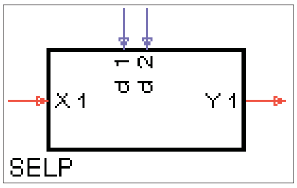
Digital input	
d1	Selecting the constant group (0 = group 1; 1= group 2)

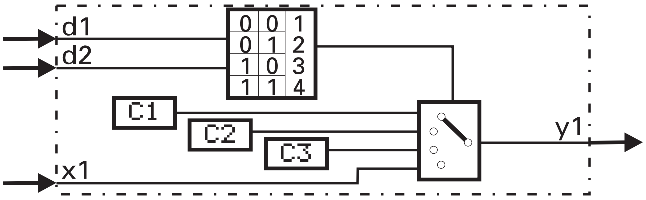
Analog outputs		
	d1 = 0 ≙ group 1	d1 =1 ≙ group 2
y1	c1.1	c2.1
y2	c1.2	c2.2
y3	c1.3	c2.3
y4	c1.4	c2.4

Parameters:

Parameter	Description	Range	Default
C1.1	1. Constant of group 1, output at y1 with d1 =0.	-29 999...999 999	0
C1.2	2. Constant of group 1, output at y2 with d1 =0.	-29 999...999 999	0
C1.3	3. Constant of group 1, output at y3 with d1 =0.	-29 999...999 999	0
C1.4	4. Constant of group 1, output at y4 with d1 =0.	-29 999...999 999	0
C2.1	1. Constant of group 2, output at y1 with d1 =1.	-29 999...999 999	1
C2.2	2. Constant of group 2, output at y2 with d1 =1.	-29 999...999 999	1
C2.3	3. Constant of group 2, output at y3 with d1 =1.	-29 999...999 999	1
C2.4	4. Constant of group 2, output at y4 with d1 =1.	-29 999...999 999	1

15.5 SELP (parameter selection)





Dependent of control signals d1 and d2, either one of the three preset parameters C1, C2, C3 or input variable x1 is connected with output y1. Unused inputs are interpreted as 0 or logic 0.

Inputs/outputs

Digital inputs	
d1	1st digital input for parameter selection
d2	2nd digital input for parameter selection

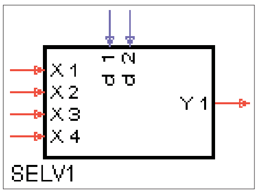
Analog input	
x1	Input is output at **y1** , when d1 = 1 and d2 =1

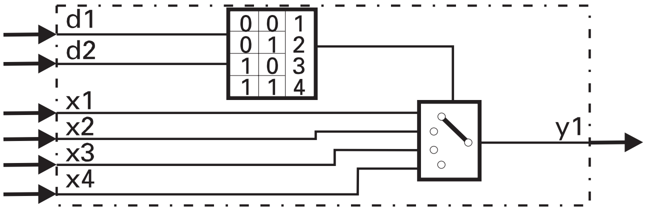
Analog outputs		
	d1	d2
y1 = C1	0	0
y1 = C2	0	1
y1 = C3	1	0
y1 = x1	1	1

Parameters

Parameter	Description	Range	Default
C1	1st constant, output at y1 with **d1** = 0 and **d2** = 0 .	-29 999...999 999	0
C2	2nd constant, output at y1 with **d1** = 0 and **d2** = 1 .	-29 999...999 999	0
C3	3rd constant, output at y1 with **d1** = 1 and **d2** = 0 .	-29 999...999 999	0

15.6 SELV1 (variable selection)





Dependent of control signals d1 and d2, one of four inputs x1...x4 is connected with output y1.
Unused inputs are interpreted as 0 or logic 0.

Inputs/outputs

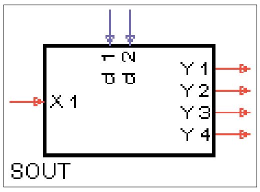
Digital inputs	
d1	1st digital input for parameter selection
d2	2nd digital input for parameter selection

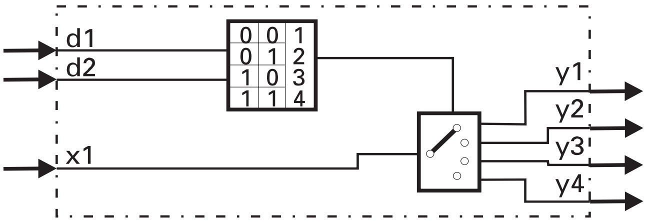
Analog inputs	
x1	Input is output at y1, when d1 = 0 and d2 =0
x2	Input is output at y1, when d1 = 0 and d2 =1
x3	Input is output at y1, when d1 = 1 and d2 =0
x4	Input is output at y1, when d1 = 1 and d2 =1

Analog outputs		
	d1	d2
y1 = x1	0	0
y1 = x2	0	1
y1 = x3	1	0
y1 = x4	1	1

No parameters!

15.7 SOUT (Selection of output)





Dependent of control signals d1 and d2,, input variable x1is connected to one of outputs y1, y2, y3 or y4.
Unused inputs are interpreted as 0 or logic 0.

Inputs/outputs

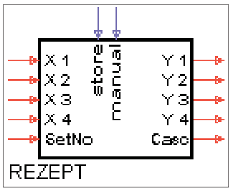
Digital inputs	
d1	1st digital input for output selection
d2	2nd digital input for output selection

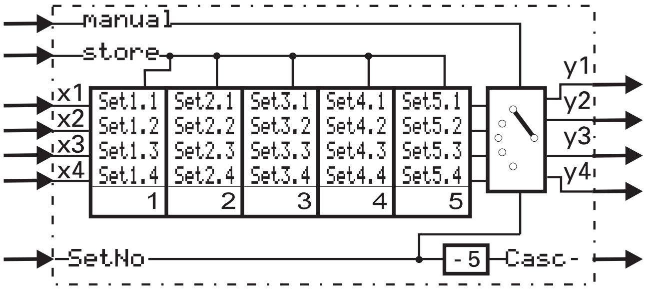
Analog input	
x1	Input is output at y1 , when d1 = 0 and d2 =0

Analog outputs		
	d1	d2
y1 = x1	0	0
y2 = x1	0	1
y3 = x1	1	0
y4 = x1	1	1

No parameters!

15.8 REZEPT (recipe management)

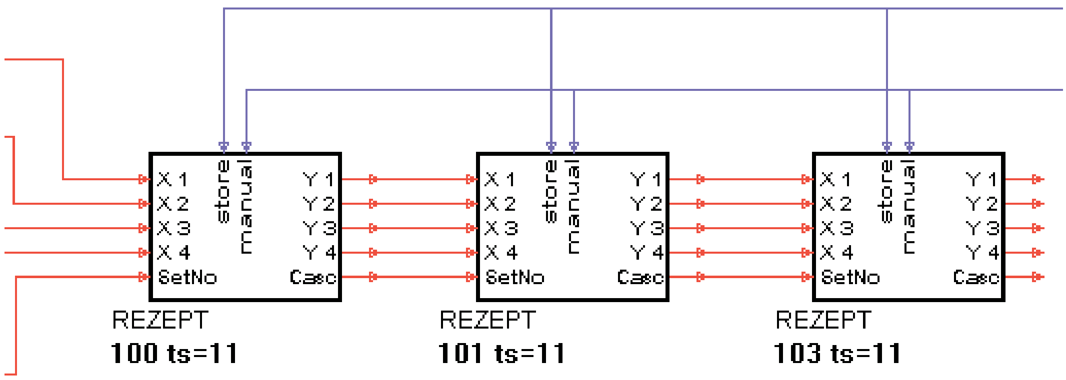




The function has 5 groups (recipe blocks) each with 4 memory locations. The recipes can be written via parameter setting and analog inputs. The function parameters are stored in EEPROM with back-up.
Selection which recipe block is output at y1...y4 is determined by the value applied to input SetNo.
In mode STORE (store = 1), the values applied to x1... x4 are written into the memory addresses of the recipe block selected with input SetNo.
During manual mode (manual = 1), the inputs are directly connected with the outputs.

If more than 5 recipes are required, a corresponding number of recipe functions are simply cascaded.

Example for 15 recipes



With cascading, the values for the overall recipe are available at outputs y1...y4 of the last stage.

Inputs/outputs

Digital inputs	
store dynamic	This input reacts only on a positive flank, i.e. on a change from 0 to 1. With this flank, input values x1...x4 are stored in the recipe block selected with SetNo. The values are stored in RAM and in EEPROM. ☞ **With store = 0 or permanent 1, storage is omitted.**
manual	manual = 0: automatic mode: recipe function active manual = 1: manual mode: the values of inputs x1...x4 are applied to y1...y4 directly.

Analog inputs	
x1...x4	In mode STORE (store =1), the values applied to ... are written into the memory locations of the group selected with SetNo. The inputs are connected with the outputs directly in manual mode (manual = 1) and also when the SetNo input is beyond range 1...5.
SetNo5	Selecting a recipe block: The value of SetNo determines, which one of the 5 recipe blocks is selected. Selection is valid for reading and storage (→ store). A recipe block is selected only with a value within 1...5 at SetNo. With SetNo out of range 1...5, the inputs are connected directly with the outputs (independent of the status at the A/M input manual). This is required for cascading.

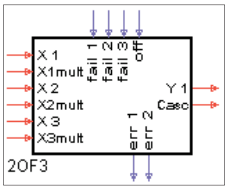
Analog outputs	
y1...y4	The values at y(i) correspond either to the recipe block selected with SetNo or to inputs x(i) in manual mode (store =1).
Casc	The value at output Casc is the value of input SetNo reduced by 5 and is used for cascading.

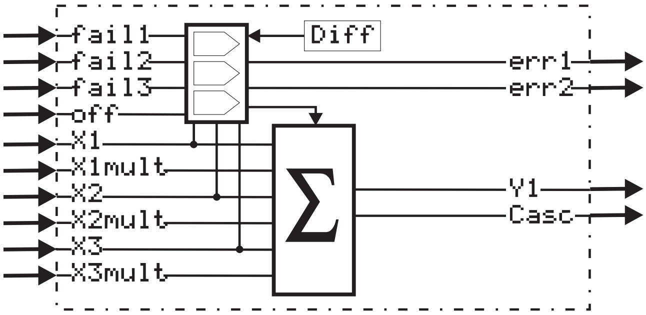
Parameters:

Via interface, 20 parameters (5 recipe blocks each with 4 values) can be preset:

Parameter	Description		Range	Default
Set1.1	**Recipe block 1**	Parameter 1 for recipe 1	-29 999...999 999	0
Set1.2		Parameter 2 for recipe 1	-29 999...999 999	0
Set1.3		Parameter 3 for recipe 1	-29 999...999 999	0
Set1.4		Parameter 4 for recipe 1	-29 999...999 999	0
Set2.1	**Recipe block 2**	Parameter 1 for recipe 2	-29 999...999 999	0
Set2.2		Parameter 2 for recipe 2	-29 999...999 999	0
Set2.31		Parameter 3 for recipe 2	-29 999...999 999	0
Set2.4		Parameter 4 for recipe 2	-29 999...999 999	0
Set3.1	**Recipe block 3**	Parameter 1 for recipe 3	-29 999...999 999	0
Set3.2		Parameter 2 for recipe 3	-29 999...999 999	0
Set3.3		Parameter 3 for recipe 3	-29 999...999 999	0
Set3.4		Parameter 4 for recipe 3	-29 999...999 999	0
Set4.1	**Recipe block 4**	Parameter 1 for recipe 4	-29 999...999 999	0
Set4.2		Parameter 2 for recipe 4	-29 999...999 999	0
Set4.3		Parameter 3 for recipe 4	-29 999...999 999	0
Set4.4		Parameter 4 for recipe 4	-29 999...999 999	0

15.9 2OF3 (2-out-of-3 selection with mean value formation)

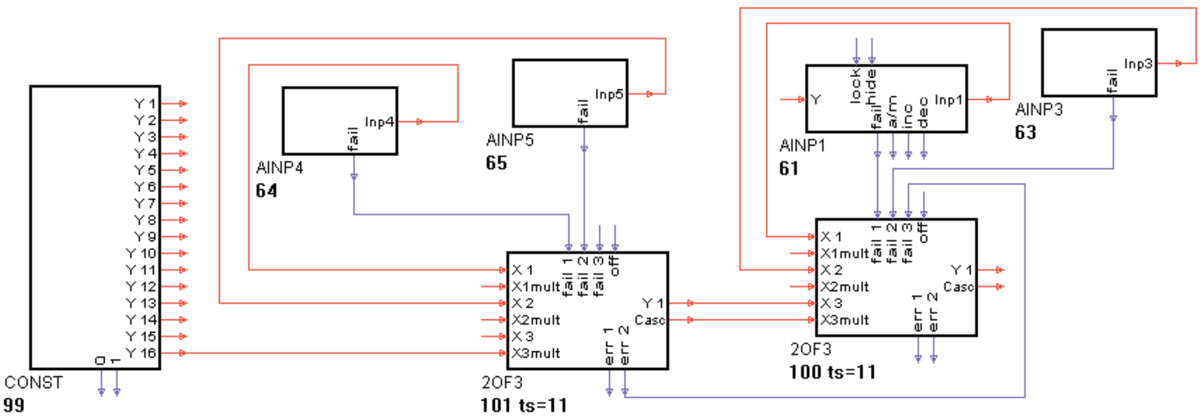




Function 2OF3 forms the arithmetic mean value of input variables x1 , x2 and x3.
The difference of x1 , x2 and x3 is formed and compared with parameter Diff. Inputs the value of which exceeds this limit value are not used for mean value formation. With 1 applied to fail1...fail3 (e.g. the fail signals of AINP), faulty inputs are not taken into account either for mean value formation. err1 = 1 indicates that 1 input failed and was not used for mean value formation. If at least 2 inputs do not participate in mean value formation, output err2 is set to 1. With input off set to 1 or if output err2 = 1 the x1 value is output at y1 .

With more than 3 input variables, function 2OF3 can be cascaded.
Output Casc indicates the number of values used for mean value formation. This is important with 2OF3 function cascading.
With unwired factor inputs (x1mult...x3mult) factor 1 is used automatically. If one of inputs x1 ...x3 is not used, the relevant x-mult must be set to 0.
The x-mult input of the following function block is wired with factor output Casc of the previous function block.

Example of cascading



In this example, CONST output y16 = 0 is set.
The following formulas are calculated:

The left 2OF3: $\frac{x1 \cdot 1 + x2 \cdot 1 + x3 \cdot 0}{2} = y1$ and the right 2OF3: $\frac{x1 \cdot 1 + x2 \cdot 1 + x3 \cdot 2}{4} = y1$

Inputs/outputs

Digital inputs	
fail1	Error message for input x1. With fail1 = 1, input x1 is not taken into account for mean value formation.
fail2	Error message for input x2. With fail2 = 1, input x2 is not taken into account for mean value formation.
fail3	Error message for input x3. With fail3 = 1, input x3 is not taken into account for mean value formation.
off	Function switch-off: with off = 1, input x1 is output at y1.

Analog inputs	
x1	Measurement input 1
x1mult	Factor input, pertaining to measurement input 1. Determination is made of how many measurement inputs the x1 consists (required with function block cascading or input not connected). Non-connected input x1mult is evaluated as value 1.
x2	Measuring input 2
x2mult	Factor input, pertains to measurement input 2. Determination is made of how many measurement inputs the x2 consists (required with function block cascading or input x2 not connected). Non-connected input x2mult is evaluated as value 1.
x3	Measurement input 3
x3mult	Factor input, pertains to measurement input 3. Determination is made of how many measurement inputs x3 consists (required with function block cascading or input x3 not connected). Non-connected input x3mult is evaluated as value 1.

Digital outputs	
err1	Error message: err1 = 1 indicates that at least one of inputs x1... x3 is not taken into account with mean value formation.
err2	Error message: err2 = 1 indicates that mean value formation is omitted. Either several inputs (fail or difference > Diff) are disturbed or function was switched off by input off.

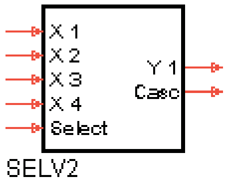
Analog outputs	
y1	arithmetic mean value or switching to x1 occurred (off = 1 or several inputs defective).
Casc	Factor: number of the values used for mean value formation. Casc = x1mult + x2mult + x3mult.

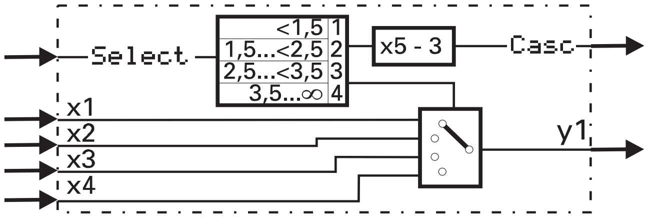
Parameter:

Parameter	Description	Range	Default
Diff	Limit value for comparison of differences between inputs x1 ... x3 for determination of faulty inputs.	0...999 999	1

No configuration parameters!

15.10 SELV2 (cascadable selection of variables)



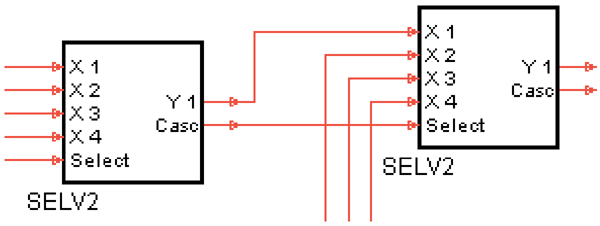


Dependent of input Select,, one of the four inputs x1...x4 is connected with output y1.

Unused inputs are interpreted as 0.
Output Casc = input Select -3.

The function can be cascaded as shown in the example given below. Dependent of input signal Select at the 1st SELV1, the corresponding variable is output at Y1 of the 2nd SELV2.

Cascading



SELV1	y 1output 2nd SELV1
Select< 1,5	x1 of 1st SELV1
1,5 < Select < 2,5	x2 of 1st SELV1
2,5 < Select <3,5	x3 of 1st SELV1
3,5 < Select < 4,5	x4 of 1st SELV1
4,5 < Select < 5,5	x2 of 2nd SELV1
5,5 < Select < 6,5	x3 of 2nd SELV1
< 6,5	x4 of 2nd SELV1

Inputs/outputs

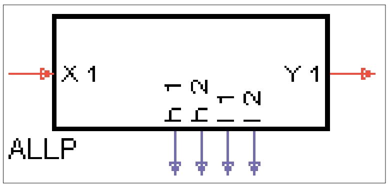
Analog inputs	
x1	Input is output at y1 with Select< 1,5.
x2	Input is output at y1 with 1,5 < Select < 2,5.
x3	Input is output at y1 with 2,5 < Select < 3,5.
x4	Input is output at y1 with Select < 3,5.
Select	Dependent of input value, the relevant variable is output at y1 .

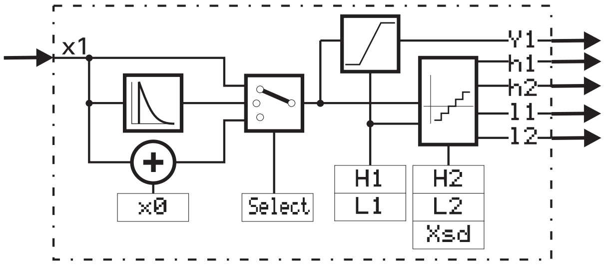
Analog outputs	
y1	According to the input value of Select, the relevant input variable is output.
Casc2	Cascade output = Select - 3

No parameters!

16 Limit value signalling and limiting

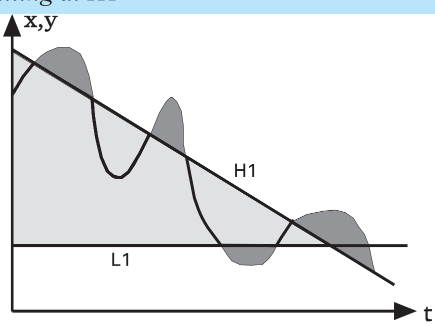
16.1 ALLP (alarm and limiting with fixed limits)





Signal limiting:
Parameter L1 determines the minimum, H1 the maximum limiting. y1 is limited to the range between L1 and H1. (L1 ≤ y1 ≤ H1).
With parameter H1 smaller than L1, a higher priority is allocated with H1. This means that y1 is ≤ H1.

Limiting at H1



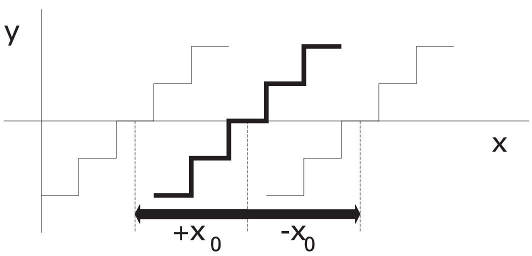
Limit signaller:

The limit signaller has two 2 low and high alarms (L1, L2, H1 and H2). Configuration parameter Select can be used to select the variable to be monitored (x1, dx1/dt, x1 - x0).
The limit values are freely adjustable as parameters and have an adjustable hysteresis of ? 0.
The smallest separation between a minimum and a maximum limit value is 0.
When an alarm is triggered, the corresponding output (L1, L2, H1 and H2) is logic “1".

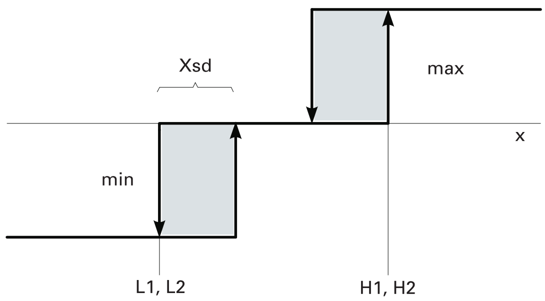
D -alarm (dx1/dt)
Value x1(t-1) measured one sampling interval before is subtracted from instantaneous value x1(t). This difference is divided by calculation cycle time Tr (100, 200, 400, 800ms).
Thus input variable x1 can be monitored for its rate of change.

Alarm with offset (x1 - x0):
x1 can be shifted by means of x0. This corresponds to the offset of the adjusted alarm limits (L1, L2, H1 and H2) in parallel to the x-axis

Offset of the alarm limits



Switching hysteresis and alarm limits:



Inputs/outputs

Analog input	
x1	Input value to be monitored

Digital outputs	
L1	Low alarm 1 - becomes logic 1, if x1 < L1
L2	Low alarm 2 - becomes logic 1, if x1 < L2
H1	High alarm 1 - becomes logic 1, if x1 <H1
H2	High alarm 2 - becomes logic 1, if x1 <H2

Analog output	
y1	Calculated and limited input signal .

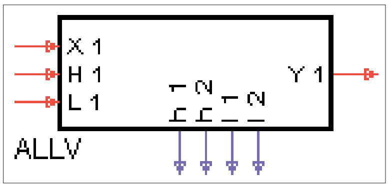
Configuration parameter:

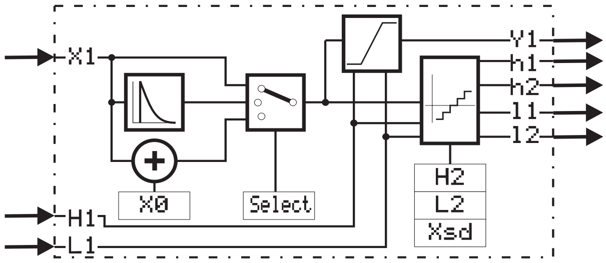
Parameter	Description		Range	Default
Select	Selection of the variable to be monitored	x1	x1	←
		D alarm	dx1/dt	
		Alarm with offset	x1-x0	

Parameters:

Parameter	Description	Range	Default
H1	High alarm 1	-29 999 ... 999 999	9999
H2	High alarm 2	-29 999 ... 999 999	9999
L1	Low alarm 1	-29 999 ... 999 999	-9999
L2	Low alarm 2	-29 999 ... 999 999	-9999
x0	Offset x0	-29 999 ... 999 999	0
Xsd	Switching hysteresis	0 ... 999 999	1

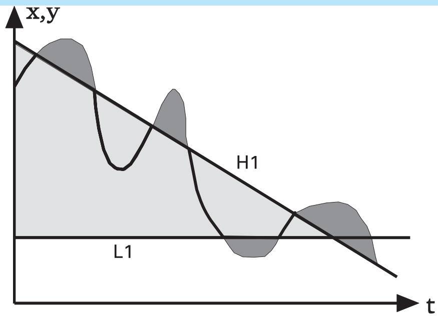
16.2 ALLV (alarm and limiting with variable limits)





Signal limiting:
Analog input H1determines the maximum limiting, L1 determines the minimum limiting. y1 is limited to the range between L1 and H1 (L1 ≤ y1 ≤ H1).
As both H1 and L1 come from analog inputs, H1 can be smaller than L1 . In this case, H1 is assigned a higher priority. This means that signal y1 is ≤ H1 !

Limiting at H1

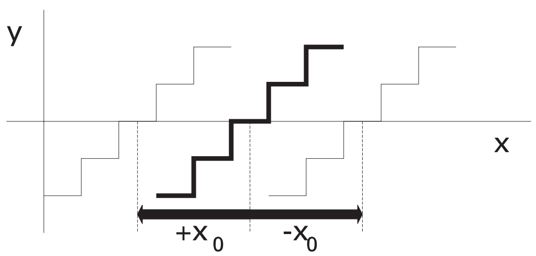


Limit signaller:
The limit signaller has 2 low and high alarms (L1, L2, H1 and H2). The variable to be monitored can be selected with configuration parameter Select (x1, dx1/dt, x1 - x0).
The limit values are freely adjustable via the analog inputs H1 and L1 and have an adjustable hysteresis of ≥ 0.
The smallest separation between a minimum and a maximum limit value is 0. With an alarm triggered, the relevant output (L1, L2, H1 and H2) is logic “1".

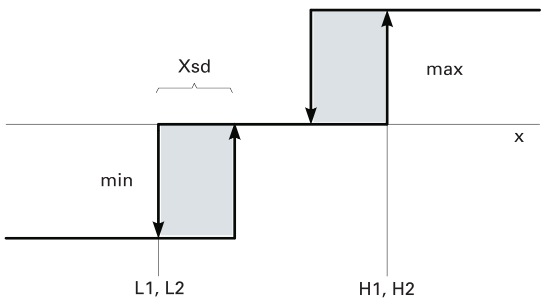
D alarm (dx1/dt)
Value x1(t-1) measured one sampling interval before is subtracted from instantaneous value x1(t). This difference is divided by calculation cycle time Tr (100, 200, 400, 800ms).
Thus input variable x1 can be monitored for rate of change.

Alarm with offset (x1 - x0):
x1 can be shifted by means of x0. This corresponds to the offset of alarm limits (L1, L2, H1 and H2) in parallel to the x-axis.

Offset of the alarm limits



Switching hysteresis and alarm limits:



Inputs/outputs

Analog inputs	
x1	Input value to be monitored
H1	High alarm 1
L1	Low alarm 1

Digital outputs	
L1	Low alarm 1 - is logic 1 with x1 <L1
L2	Low alarm 2 - is logic 1 with x1 <L2
H1	High alarm 1 - is logic 1 with x1 <H1
H2	High alarm 2 - is logic 1 with x1 <H2

Analog output	
y1	Calculated and limited input signal x1 .

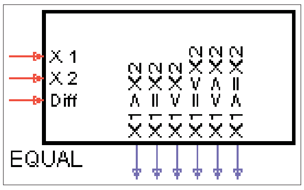
Configuration parameters:

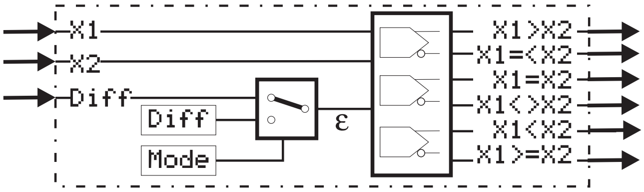
Parameter	Description		Range	Default
Select	Selection of variable to be monitored	x1	x1	←
		D alarm	dx1/dt	
		Alarm with offset	x1-x0	

Parameters:

Parameter	Description	Range	Default
H2	High alarm 2	-29 999 ... 999 999	9999
L2	Low alarm 2	-29 999 ... 999 999	-9999
x0	Offset x0	-29 999 ... 999 999	0
Xsd	Switching hysteresis	0 ... 999 999	1

16.3 EQUAL (comparison)





The function checks the two analog input values x1 and x2 for equality.
The values are equal, if the amount of their difference is smaller than oder equal to the preset tolerance.

Comparison conditions	z6	z5	z4	z3	z2	z1
x2 + Diff < x1	1	1	0	0	0	1
x2 - Diff ≤ x1 ≤ x2 + Diff	1	0	1	0	1	0
x2 − Diff > x1	0	1	1	1	0	0

The tolerance can be adjusted either as parameter Diff (Mode = Para.Diff) or entered at analog input Diff (Mode = Inp.Diff).

Inputs/outputs

Analog inputs	
x1	1st input value to be compared
x2	2nd input value
Diff	Tolerance for comparison operations

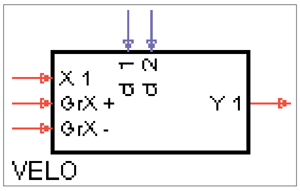
Digital outputs	
z1	z1 = 1 with x2 + Diff < x1
z2	z2 = 1 with x2 - Diff ≤ x1 ≤ x2 + Diff
z3	z3 = 1 with x2 - Diff > x1
z4	z4 = 1 with x2 + Diff x1
z5	z5 = 1 with x2 - Diff > x1 > x2 + Diff
z6	z6 = 1 with x2 - Diff ≤ x1

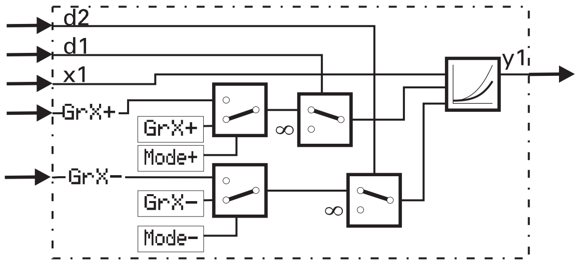
No configuration parameters!

Parameters:

Parameter	Description		Range	Default
Mode	Tolerance source	Parameter Diff	Para.Diff	←
		analog input Diff	Inp.Diff	
Diff	Tolerance for comparison operation		0 ... 999 999	0

16.4 VELO (rate-of-change limiting)





The function passes input variable x1 to output y1 and limits its rate of change dx1/dt to a positive and negative gradient.

The gradients can be adjusted either as parameter GrX+ and Grx− or preset at analog inputs GrX+ and Grx−. Switch-over between the gradient sources is by parameter Mode+ for the positive gradient and by Mode− for the negative gradient.
Via digital inputs d1 and d2, limiting can be switched off separately for positive and negative rates of change.
When using the analog inputs for gradient adjustment, the following is applicable:
GrX+ ≥ 0 or GrX− ≤ 0 otherwise the relevant gradient is set to 0.

The function has a 'memory'. This means: after power-on, it continues operating with the value of y1 which existed at power-off, provided that the RAM data are still unchanged.

Inputs/outputs

Digital inputs	
d1	Control of positive gradient 0 = the selected gradient is effective. 1= gradient =
d2	Control of negative gradient 0 = the selected gradient is effective. 1= gradient = -

Analog inputs	
x1	Input variable to be limited
GrX+	positive gradient [/] *with parameter Mode+ = Inp. GrX+*
GrX−	negative gradient [/] *with parameter Mode- = Inp. GrX-*

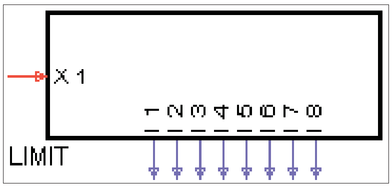
Analog output	
y1	Limited input value *x1*

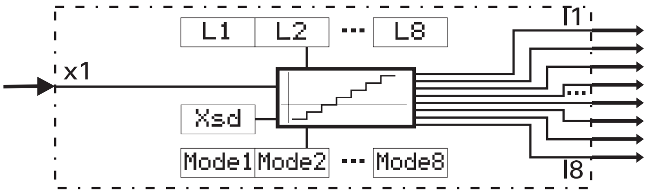
No configuration parameters!

Parameters:

Parameter	Description		Range	Default
Mode+	Source of positive gradient	parameter GrX+	Para.GrX+	←
		analog input GrX+	Inp. GrX+	
Mode−	Source of negative gradient	parameter GrX−	Para.GrX−	←
		analog input GrX−	Inp. GrX−	
Grx+	positive gradient [/] with parameter Mode+ = Para.GrX+		0 ... 999 999	0
Grx−	negative gradient [/] with parameter Mode− = Para.GrX−		-29 999 ... 0	0

16.5 LIMIT (multiple alarm)

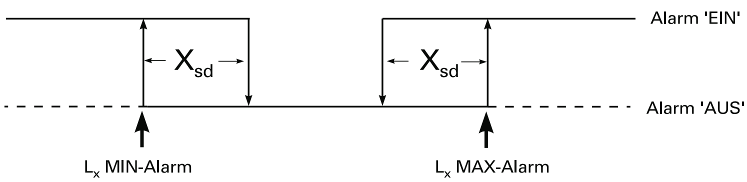




The function checks input variable x1 for 8 alarm values L1...L 8. Dependent of configuration by Mode 1 ... Mode 8, the relevant alarm value is evaluated as MAX or MIN alarm.

With MAX alarm configuration, the alarm is triggered when the input signal is higher than the alarm value and finished when it is lower than (alarm value - hysteresis Xsd).

With MIN alarm configuration, the alarm is triggered when the input signal is lower than the alarm value and finished when it is higher than (alarm value + hysteresis Xsd).



Inputs/outputs

Analog input	
x1	Input variable to be monitored

Digital outputs	
l1 ... l8	The alarm statuses of alarm 1 to alarm 8: 0 = no alarm; 1= alarm case

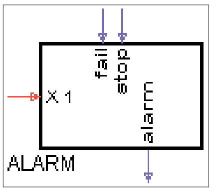
Configuration parameters:

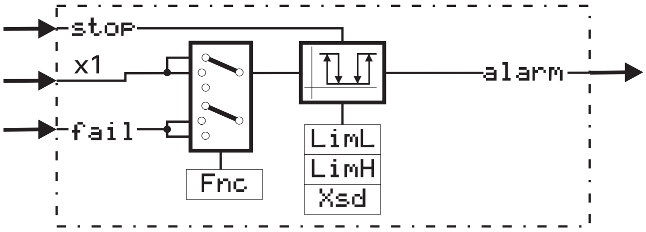
Parameter	Description		Range	Default
Mode1 ... Mode8	alarm functions of the 8 alarms	**max-alarm**	MAX-alarm	←
		min-alarm	MIN-alarm	

Parameters:

Parameter	Description	Range	Default
L1 ...L8	Alarm values of alarm 1 to alarm 8	-29 999 ... 999 999	0
Xsd	Switching hysteresis Xsd	0 ... 999 999	0

16.6 ALARM (alarm processing)

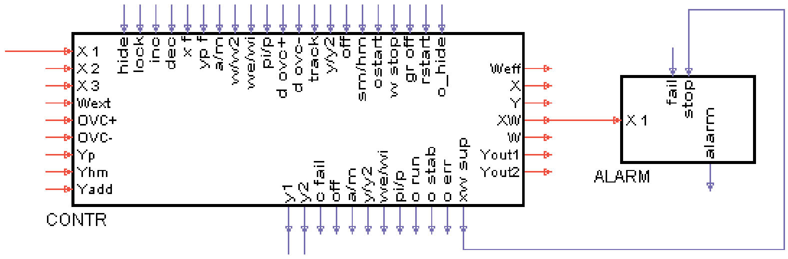




x1 is checked for a lower and an upper alarm value. Additionally, digital alarm input fail can be used. Configuration parameter Fnc can be used to select which signal shall be monitored (x1, x1 + fail or fail).
With input stop = 1, alarms (fail and x1) are suppressed. After removal of this signal, suppression lasts, until the monitored value is again within the limits. This can be used e.g. for suppressing an alarm message with set-point change.

Alarm suppression with set-point change

During set-point value change at the exit xw sup a pulse with the length of a scanning cycle Ts is sent.



Inputs/outputs

Digital inputs	
fail	Digital alarm signal e.g. fail signal of AINP
stop	stop = 1, alarms (fail and x1) are suppressed. After stop returned to 0, suppression lasts, until the monitored value is again within the limits.

Analog input	
x1	Input variable to be limited

Digital output	
alarm	Alarm status: 0 = no alarm; 1= alarm

Configuration parameter:

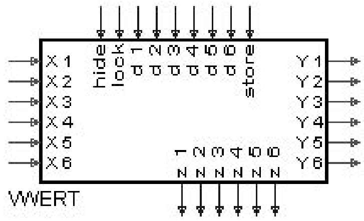
Parameter	Description		Range	Default
Fnc	Alarm function	**only x1 is monitored**	Mes.val.X1	←
		x1 and fail are monitored	X1 + fail	
		only fail is monitored	fail	

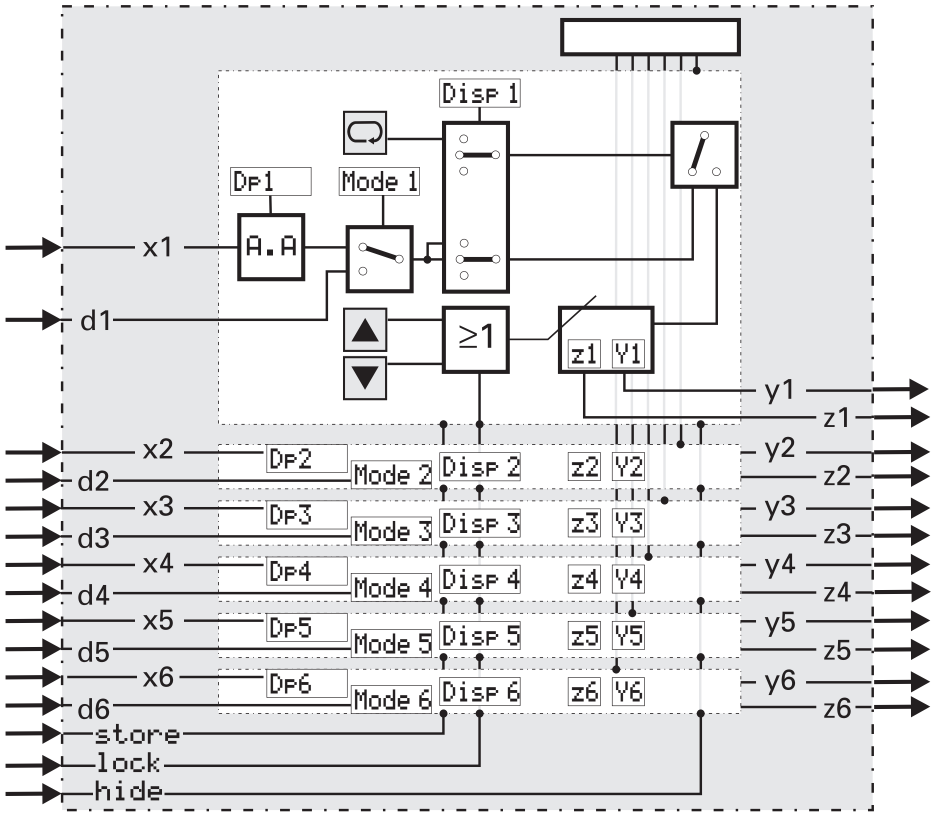
Parameters:

Parameter	Description	Range	Default
LimL	lower limit for the alarm	-29 999 ... 999 999	-10
LimH	upper limit for the alarm	-29 999 ... 999 999	10
Lxsd	Switching hysteresis Xsd	0 ... 999 999	10

17 Visualization

17.1 VWERT (display / definition of process values)





General

This function permits display or definition of 6 analog or digital process values in 6 display lines.

- Determination if the display line has digital or analog functions, or if it is switched off is made via configurations (generation of an empty line in the display).
- Normally, the values applied to the inputs are displayed.
- A value adjustable at the front panel is output at the relevant function output.
- The change of these values from the operating level can be switched off.
- Parameters z1 ... z6 or y1 ... y6 are used as initial value for the outputs at power-on.
- The output value is displayed only, if the output is fed back to the relevant input, or if the display for this value is in the adjustment mode.
- With a positive flank at the store input, the values applied to the remaining inputs are stored in parameters z1 ... z6 and y1 ... y6 and thus used as output values.

Value changes are stored as parameters z1 ... z6 or y1 ... y6 in non-volatile EEPROM. With digital input lock set, no values can be changed. With digital input hide set, the operating page cannot be displayed. The engineering tool can be used to configure a 16-digit text for the display header and further texts for identification of value and unit, or for the two digital statuses.

Inputs/outputs

Digital inputs:	
hide	Display suppression (with hide = 1 the page is not displayed in the operation.
lock	Adjustment locking (with lock = 1 the values are not adjustable by means of keys ▲▼).
d1 ... d6	Process statuses to be displayed. (Default = 0)
store	With a positive flank (0⟶1) the input values are used as output values.

Digital outputs:	
z1 ... z6	Valid process values

Analog inputs:	
x1 ... x6	Process values to be displayed (default = 0)

Analog outputs:	
y1 ... y6	Valid process values

Parameter and configuration data

Parameter	Description	Range	Default
z1 ... z6	Start values for digital outputs 1...6 at power-on	0 / 1	0
Y1 ... Y6	Start values for the analog outputs 1...6 at power on	-29999...999 999	0

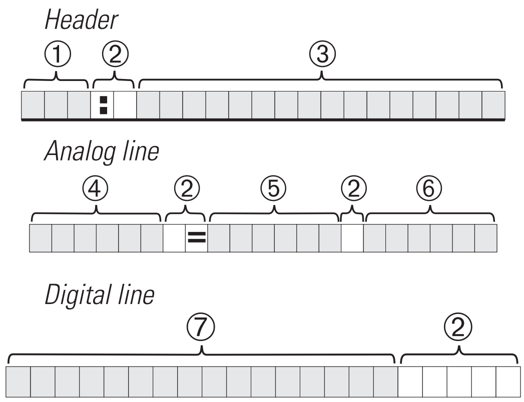
Configuration	Description		Values	Default
Disp1 ... Disp6	Function of display line 1...6	display line, value ajustable	adjustable	
		only display line	display	←
		line = empty line	empty	
Mode1 ... Mode6	Type of display line 1...6	display line analog	analog	←
		display line digital	digital	
Dp1 ... Dp6	Digits behind decimal point in analog line 1...6		0 ... 3	0

Entry and display of texts

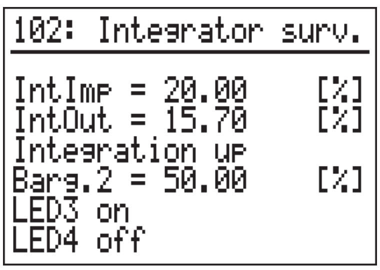
Changing the texts displayed in the unit is only possible in the engineering tool! Max. 16 characters can be entered into each text parameter. Depending of whether a line was configured as analog or digital line, all characters (Mode x = digital) or only the first 6 characters (Mode x = analog) are displayed in the instrument.

The following values or texts are displayed in the lines:

① Block number 3 digits
② Fixed text or spaces (no access)
③ Title (16 digits)
④ Parameter name (the first 6 characters of 'Text1 a ... Text6 a' dependent of line)
⑤ Value x1 ... x6 dependent of line
⑥ Unit (the first 6 characters of 'Text1 b ... Text6 b' dependent of line)
⑦ User texts (16 characters)
signal = 0: dependent of line of 'Text1 a ... Text6 a'
signal = 1: dependent of line of 'Text1 b ... Text6 b'

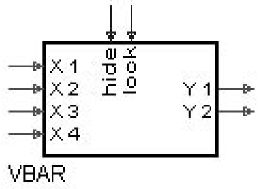


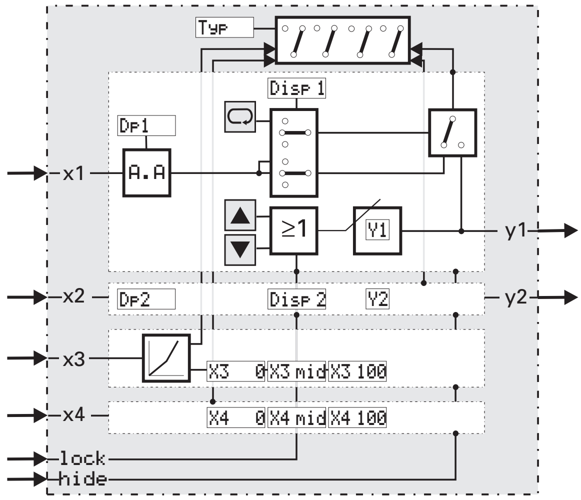
VWERT operating page



VWERT has an operating page, which can be selected in the operating page menu with input 'hide' not used.
To change the value of an input field, this value must be marked by means of ▲▼ (inverse display). If the value is acknowledged with ⏎, it starts blinking and can be adjusted with ▲▼. When the required value is reached, it must be acknowledged with ⏎. If a line is configured as display, the value of this line cannot be changed.

17.2 VBAR (bargraph display)





General

This function permits the display of 2 analog input signals as bargraphs, and of 2 analog input signals as numeric values. Moreover, two analog output signals can be defined.

- Determination of horizontal or vertical bargraph is via configuration.
- Determination, if the bargraphs/numeric displays are visible or switched off (generation of an individual bargraph) is via configuration.
- One-directional or two-directional bargraph operation is determined by configuring the start value.
- Normally, the values applied to the inputs are displayed.
- A value which is adjustable via the front panel is output at the relevant analog output.
  Changing these values from the operating level can be suppressed.
- Parameters Y1 / Y2.
- The output value is displayed only with the output fed back to the relevant input, or if the display for this value is in the adjustment mode.
- Value changes are stored as parameters Y1 / Y2 in non-volatile EEPROM.

With digital input lock set, no values can be changed. With digital input hide set, the operating page cannot be displayed during operation. A 16-digit text for the display header can be adjusted user-specifically via the engineering tool. The same is applicable for further texts for identification of value and unit.

Inputs/outputs

Digital inputs:	
hide	Display suppression (with hide = 1 the operating page is not displayed).
lock	Adjustment blocking (with lock = 1 these values cannot be adjusted by means of keys ▲▼).

Analog inputs:	
x1 / x2	Process values to be displayed as values (default = 0)
x3 / x4	Process values to be displayed as bargraph (default = 0)

Analog outputs:	
y1 / y2	Valid process values

Parameter and configuration data

Parameter	Description	Range	Default
Y1 / Y2	Start values at power-on	-29999...999 999	0

Configuration	Description		Values	Default
Disp1 Disp2	Function of numeric display 1 and 2	display x1 / x2, value adjustable	disp+adj	
		only display x1 / x2	display	←
		x1 / x2 = empty	empty	
Dp1 / Dp2	Digits behind the decimal point in numeric display 1 / 2		0 ... 3	0
Typ	Position of bargraphs	Both bargraphs horizontal	horizont	←
		Both bargraphs vertical	vertical	
X3 0	Display scaling bargraph 1, 0% (left or bottom end)		-29999...999 999	0
X3 100	Display scaling bargraph 1, 100% (right or upper end)		-29999...999 999	100
X3 mid	Display scaling bargraph 1, start value (middle)		-29999...999 999	0
X4 0	Display scaling bargraph 2, 0% (left or bottom end)		-29999...999 999	0
X4 100	Display scaling bargraph 2, 100% (right or upper end)		-29999...999 999	100
X4 mid	Display scaling bargraph 2, start value (middle)		-29999...999 999	0

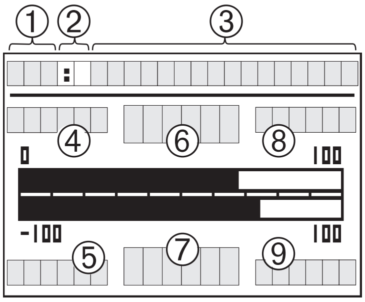
Input and display of texts

Changing the texts displayed in the unit is only possible in the engineering tool! Max.16 characters can be entered in each text parameter.

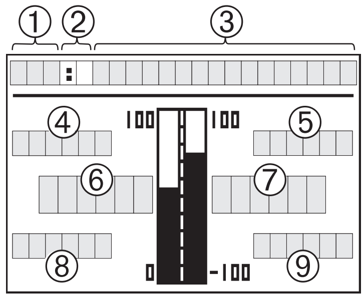
The following values or texts are displayed:

① Block number 3 digits
① Fixed text or spaces (no access)
① The 16 characters of 'title'
① Parameter name for x1 (first 6 characters of 'Name 1')
① Parameter name for x2 (first 6 characters of 'Name 2')
① Value x1
① Value x2
① Unit for x1 (first 6 characters of 'Unit 1')
① Unit for x2 (first 6 characters of 'Unit 2')

horizontal bargraphs

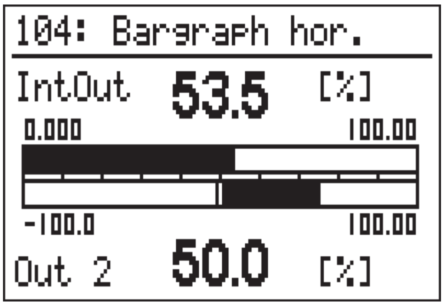


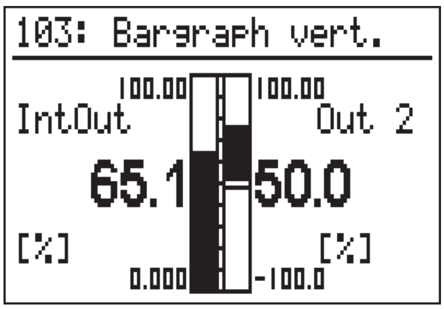
vertical bargraphs



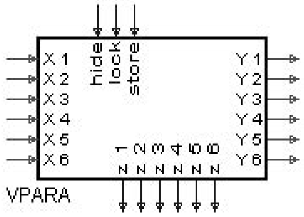
VBAR operating page

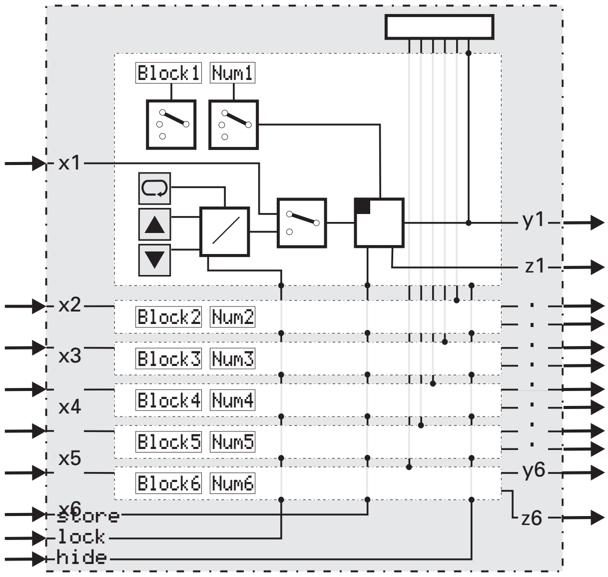
VBAR has an operating page, which can be selected in the operating page menu with the 'hide' input not used.
For changing the value of an input field, this value must be marked with ▲▼ (inverse display). When acknowledging the value with ⮐, it starts blinking and can be adjusted with ▲▼. When the required value was reached, it must be acknowledged with ⮐. A value configured as display cannot be changed.





17.3 VPARA (parameter operation)

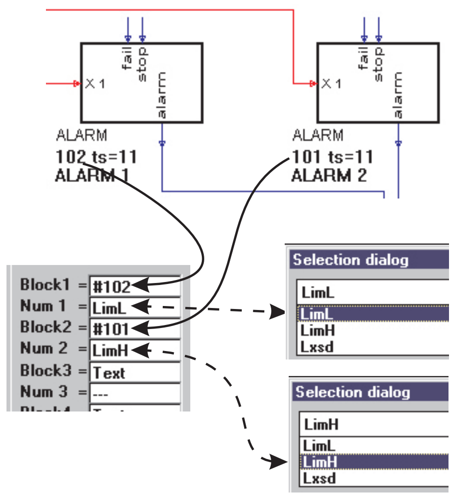




General

Function VPARA provides an operating page which can be used for changing max. 6 parameters of other function blocks available in the engineering from the operating level.

Each parameter to be displayed is made known to the display function with block number and parameter number by means of two configuration data. The engineering tool supports parameter setting by a special operating sequence in which the parameter numbers of the selected block are selected by means of the parameter descriptions (→ see figure opposite).
Additionally, an identifier and a unit text can be specified.



Values of the used analog inputs are stored as analog values, when a positive flank is detected at the `store` input. Activation of this input must be organized so that it occurs only with relevant input value changes. Too frequent storage can lead to EEPROM destruction.

Inputs/outputs

Digital inputs:	
`hide`	Display suppression (with `hide` = 1 the page is not displayed in the operation).
`lock`	Adjustment blocking (with `lock` = 1 the values are not adjustable by means of keys ▲▼).
`store`	With a positive flank (0→1) the input values are stored as parameter values.

Digital outputs:	
z1 ... z6	The outputs provide a status, which shows if the last storage of the values taken over from the inputs was successful (z1 ... z6 = 0). Errors may occur due to exceeded limits of the parameter value or due to non-existing parameters (z1 ... z6 = 1).

Analog inputs:	
x1 ... x6	Process values to be stored as parameter values (default = 0)

Analog outputs:	
y1 ... y6	The values of the 6 parameters are output at the analog outputs. Unused parameters provide value `0'.

Parameter and configuration data

Configuration	Description	Values	Default
Block1 ... Block6	Block number of parameter to be displayed	*	*
Num1 ... Num6	Parameter number	*	*

* To avoid confusions and thus operating errors, we recommend adjusting block numbers and parameters exclusively via the engineering tool, where the parameters with their short-form descriptions must also be specified. Text entry is only possible via the engineering tool.

Entry and display of texts

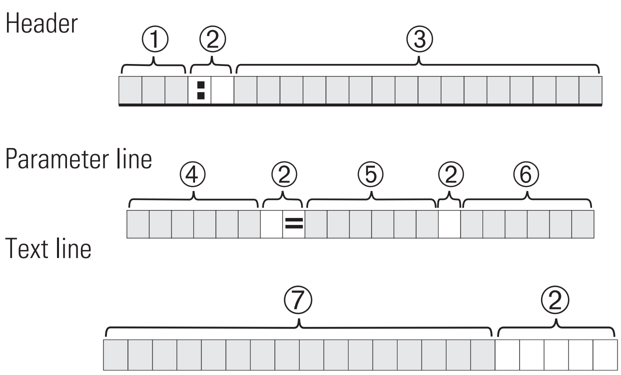
Changing the texts displayed in the unit is possible only in the engineering tool! Max. 16 characters can be entered in each text parameter. Dependent of whether a line is allocated to a block number or defined as a text line, all characters (Blockx = Text) or only the first 6 characters (Blockx = #xxx) are displayed in the unit. If parameter number (Numx) or block number (Blockx) are undefined, `??????' is displayed as a value.

Parameter allocation to the display lines: Block1; Num1; Text1; Unit 1 → line 1

....

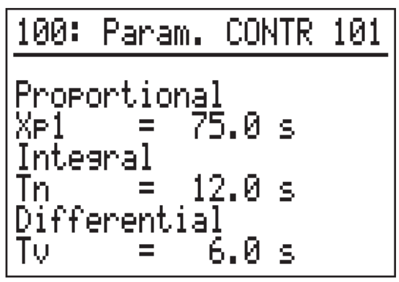
Block6; Num6; Text6; Unit6 → line 6

The following values or texts are displayed:
1. Block number 3 digits
2. Fixed text or spaces (no access)
3. The first 16 characters of 'title'
4. Parameter name (the first 6 characters of 'Text 1'...'Text 6' dependent of line)
5. the parameter values
6. Unit (the first 6 characters of 'Unit 1'...'Unit 6' dependent of line)
7. The first 16 characters of 'Text 1'...'Text 6' dependent of line



VPARA operating page

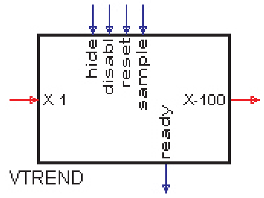
VPARA has an operating page, which can be selected in the operating menu with input 'hide' not used.

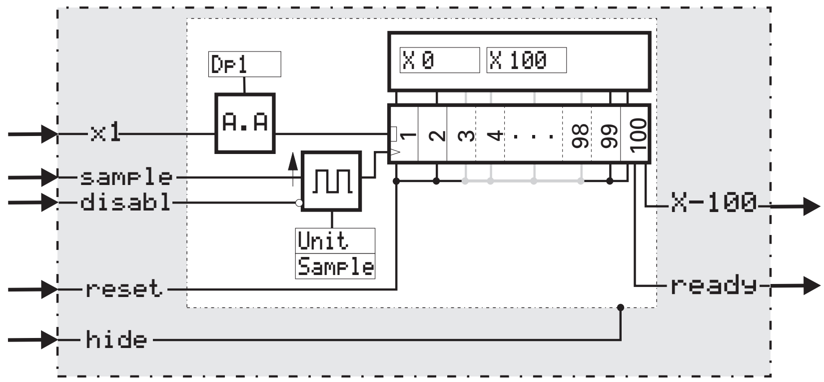


For changing the value of an input field, this value must be marked with ▲▼ (inverse display). When acknowledging the value with ⏎, it starts blinking and can be adjusted with ▲▼.

When reaching the required value, confirm it with ⏎.
If the analog inputs (x1 ... x6) are used by the engineering, operation (change) of this input field is not possible.

17.4 VTREND (trend display)





General

Function VTREND collects 100 values of the analog input 'x1' in a shift register and permits value display as a trend curve. When the shift register is filled with 100 values, the value 100 samples ago is overwritten by a new value. With input sample 'sample' not used, data recording is synchronous with the time units specified in the configuration. Trigger pulses at the 'sample' input permit asynchronous data recording.

With voltage failure, the sampled values remain unchanged.

Inputs/outputs

Digital inputs:	
hide	Display suppression (with hide = 1 the page in the operation is not displayed).
disable	The digital input can be used to interrupt automatic or triggered sampling (high-active).
reset	The digital input deletes the shift register and resets trend measurement.
sample	If the digital input is wired, sampling is triggered by a positive flank (0 → 1) at this input. In this case, the adjusted sampling interval (configuration) is not effective.

Digital outputs:	
ready	After filling the shift register with 100 values first, the digital output is set to high.

Analog inputs:	
x1	Process value to be displayed as trend (default = 0)

Analog outputs:	
X-100	The value of the shift register which is overwritten by the next sample value is provided at the analog output (value 100 samples ago).

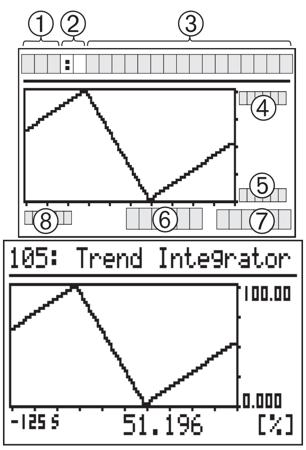
Configuration data

Configuration	Description		Values	Default
Unit	Unit of sampling interval	seconds (s)	sec.	←
		Minutes (m)	min.	
		Hours (h)	h	
Sample	Value of sampling interval in the unit defined with 'Unit'.		0,2...200000	1
Dp	Digits behind the decimal point for value displays		0 ... 3	0
X 0	Display scaling start value (0%)		-29999...200000	0
X100	Display scaling end value (100%)		-29999...200000	100

Text display and entry

Changing the texts displayed in the unit is possible only in the engineering tool!
Max. 16 characters can be entered into each text parameter.

Trend curve



The following values or texts are displayed:

1. Block number 3 digits
2. Fixed text or spaces (no access)
3. The first 16 characters of 'Title'
4. X 100
5. X 0
6. Last stored value
7. Unit for x1 (first 6 characters of 'Unit')
8. -100 * Sample, Unit

VTREND operating page

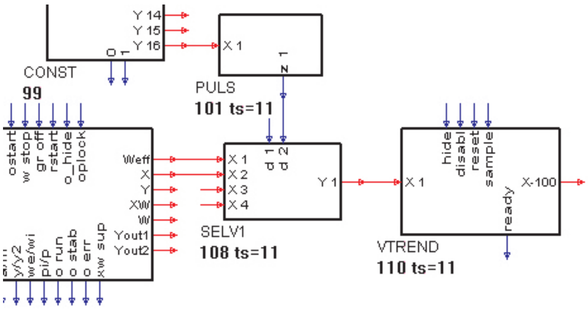
VPARA has an operating page, which can be selected in the operating page menu with input 'hide' not used. VTREND has no operating functions. The operating page is used exclusively for trend data display

Examples: Trend recording with 2 curves

Although distinction of different curves is not possible, display of two values on a trend page may be purposeful (e.g. controller set-point and process value, or one value and zero, in order to have a curve). In the example, a clock triggering switch-over between the values together with the SELV1 is generated by means of a pulse.

Unit	s
Dp	0
Sample	1.000

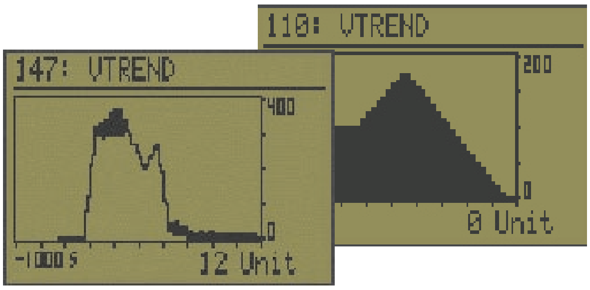
X 0	0.000
X 100	3600.00
Puls/h	3600.00



For making a record at intervals of a second e.g. in VTREND, Unit is set to s and Sample is set to 1.

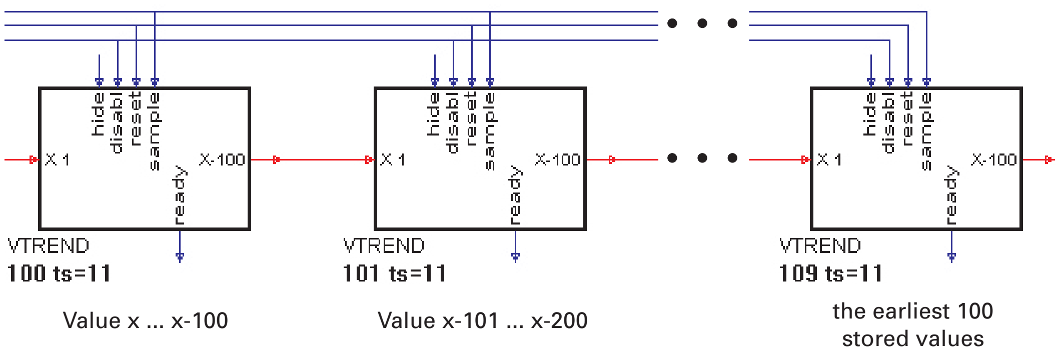
Settings: Unit = s and
Sample =1 ≙ 1/s = 3600/h

x0=0, x100 and pulse/h to 3600,
1/2 sample interval = 1800 must be applied to pulse input x1.



Cascading

Example of trend-/datarecording with n-values



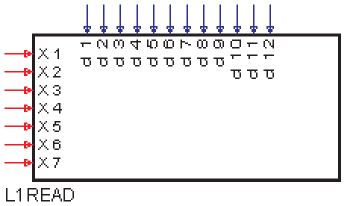
Trend or data recording with any number of values can be realized by cascading VTREND function blocks. Limiting refers only to the number of available block numbers and to the calculation time. The data sequence is dependent of VTREND function block wiring.

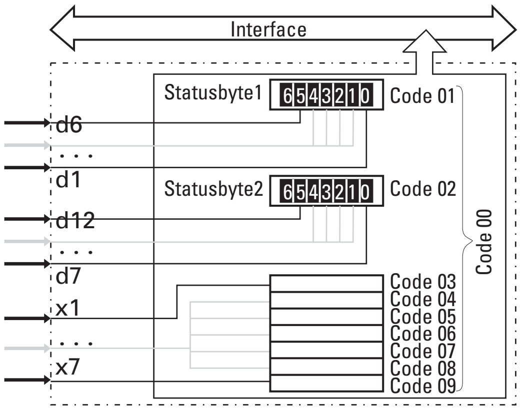
18 Communication

ISO 1745

In total, max. 20 L1READ and L1WRIT functions can be configured (blocks 1...20), any combination of functions is possible. Any number of data can be used in the functions.

18.1 L1READ (read level1 data)





General

Any 7 analog process values (x1...x7) and any 12 digital status informations (d1...d12) of the engineering are composed into a data set for the digital interface. The digital interface can read the data set as a complete block with code 00, function number 0, or the individual values with codes 01...09, function number 0.

Inputs/outputs

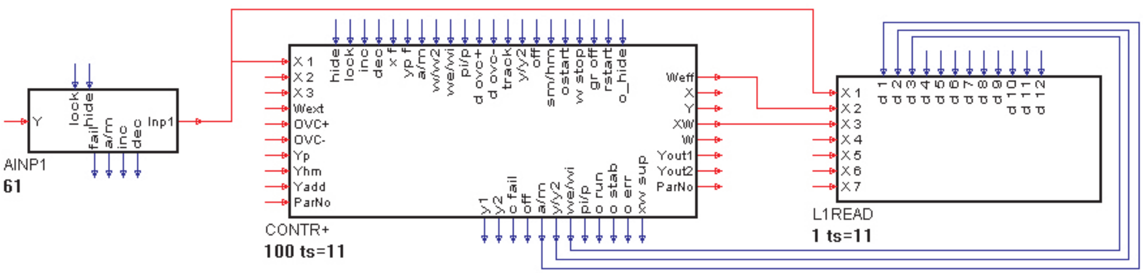
Digital inputs:	
d1 ... d6	Digital process values, which can be read via interface (status byte 1). (Default = 0)
d7 ... d12	Digital process values, which can be read via interface (status byte 2). (Default = 0)

Analog inputs:	
x1 ... x7	Analog process values, which can be read via interface. (Default = 0)

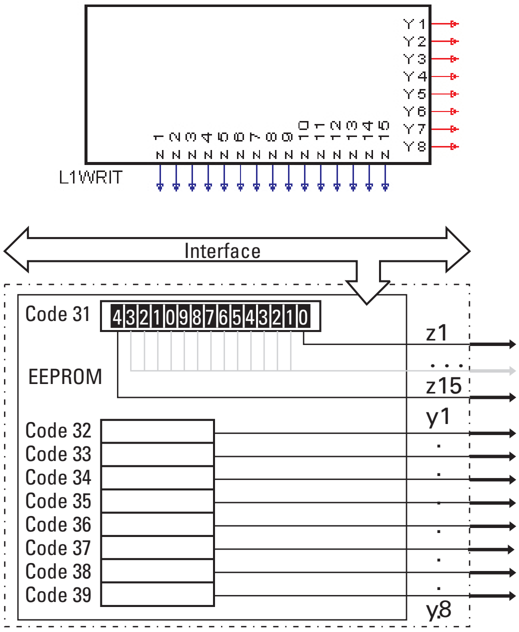
Engineering example

In the following example, several process data (process value, effective set-point and control deviation) and controller statuses (automatic/manual, Wint/Wext and y/Y2) are connected with the L1READ function block. Now, these data can be read in a message via interface.

Example for L1READ engineering



18.2 L1WRIT (write level1 data)



General

This function is used to provide a data set transmitted by the interface to the engineering. The digital interface describes EEPROM cells with codes 31...39, function number 0. The data set comprises 8 analog process values (y1...y8) and 15 digital control informations (z1...z15), which are provided to the engineering.

The transmitted data are stored in the EEPROM. After power failure, start is with the data rather than with the default values.

Inputs/outputs

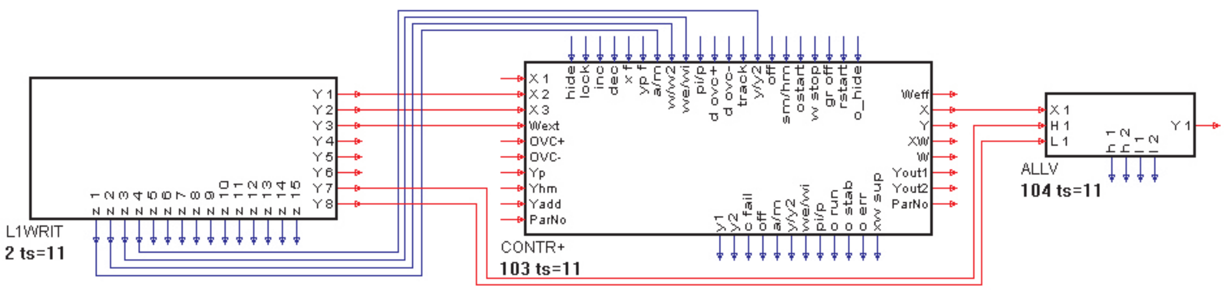
Digital outputs:	
z1 ... z12	Digital process values, which can be written via the interface (default = 0)

Analog outputs:	
y1 ... y8	Analog process values, which can be written via the interface (default = 0)

Engineering example

In the following example, the L1WRIT function block is used to make several process data (process values x2, x3, external set-point and two alarm limits) and the control information (automatic/manual, w/W2, Wint/Wext and y/Y2) available to the engineering. These data can be written in a message via interface.

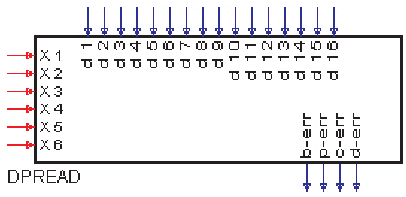
Example for L1WRIT engineering

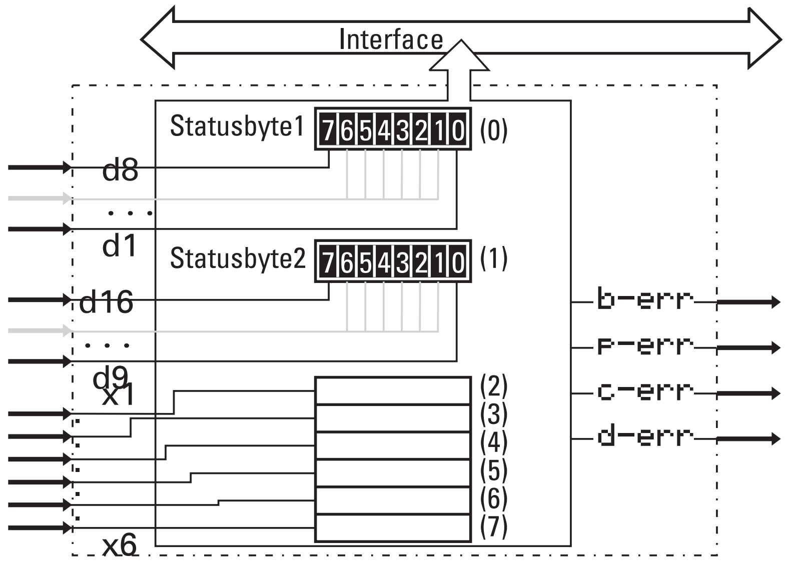


PROFIBUS

Max. 4 functions DPREAD and DPWRIT can be configured (blocks 1...4 or 11...14). Any combination of functions is possible. Any data can be used in the functions.

18.3 DPREAD (read level1 data via PROFIBUS)





General

Block numbers 1...4. Any 6 analog process values (x1...x6) and any 16 digital process values (d1...d16) of the engineerings are composed for scanning via a PROFIBUS data channel. Block number 1 provides the data for channel 1, block number 2 provides the data for channel 2, etc.

The PROFIBUS module reads the data of two channels at intervals of 100 ms. The digital outputs indicate the PROFIBUS status.

Further information on communication with PROFIBUS is given in the interface description (order no.: 9499 940 52711).

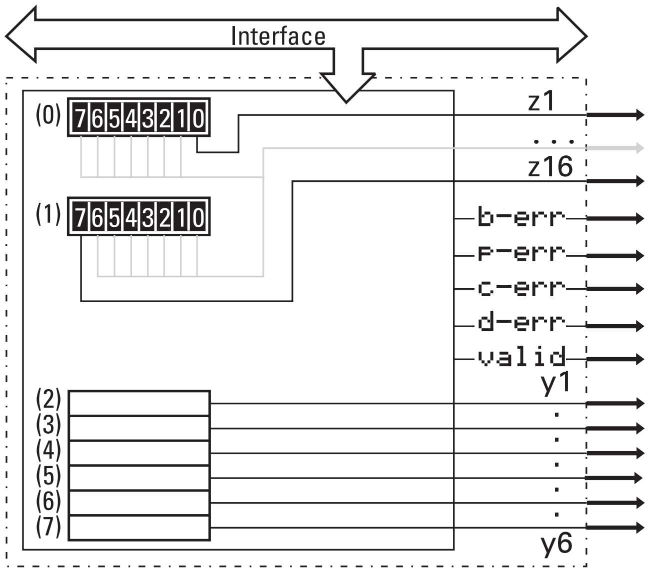
Inputs/outputs

Digital inputs:	
d1 ... d8	Digital process values, which can be read via the PROFIBUS (status byte 1)
d9 ... d16	Digitale process values, which can be read via the PROFIBUS (status byte 2)

Digital outputs:	
b-err	PROFIBUS status: 1 = bus access not successful
p-err	PROFIBUS status: 1 = faulty parameter setting
c-err	PROFIBUS status: 1 = faulty configuration
d-err	PROFIBUS status: 1 = no data communication

Analog inputs:	
x1 ... x6	Analog process values, which can be read via the PROFIBUS

18.4 DPWRIT (write level1 data via PROFIBUS)



General

Block numbers 11...14. The data of a PROFIBUS data channel are transmitted into the memory. Block number 11 transmits the data of channel 1, block number 12 transmits the data of channel 2, etc. The PROFIBUS module writes the data of two channels at intervals of 100ms. The data set comprises 6 analog process values (y1...y6) and 16 digital status informations (z1...z16), which are available to the engineering. The digital outputs (b-err, p-err, c-err, d-err and valid) indicate the PROFIBUS status.

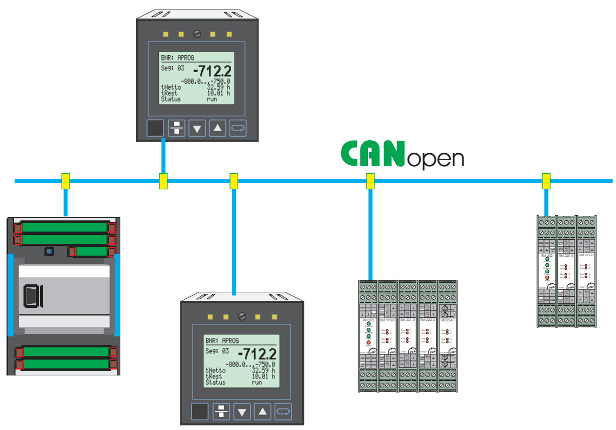
Further information on communication with PROFIBUS is given in the interface description (order no.: 9499 940 52711).

Inputs/outputs

Digital outputs:	
z1 ... z16	Digital process values, which can be written via the Profibus.
b-err	PROFIBUS status: 1 = bus access not successful
p-err	PROFIBUS status: 1 = faulty parameter setting
c-err	PROFIBUS status: 1 = faulty configuration
d-err	PROFIBUS status: 1 = no data communication
valid	PROFIBUS status: 1 = data o.k.

Analog outputs:	
y1 ... y6	Analog process values, which can be written via the Profibus.

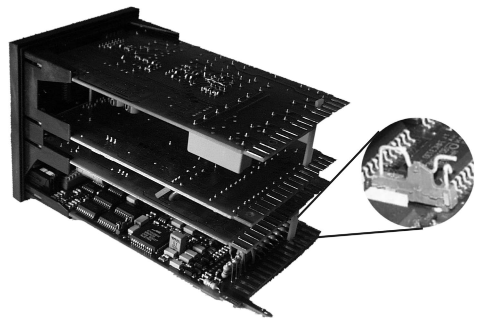
19 KS98+ I/O extensions with CANopen



The additional CANopen interface completes the functionality of the multifunction unit basic version by:

- local I/O extensibility using the PMA RM 200 modular I/O system.
- connection of the PMA multiple-channel temperature controllers with CANopen interface
- on-site data exchange with other KS98+ (cross communication)

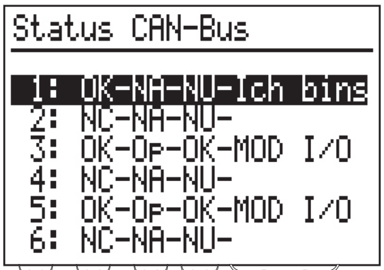
These functions are available only in KS98+ versions from operating version 5.



BUS terminating resistor

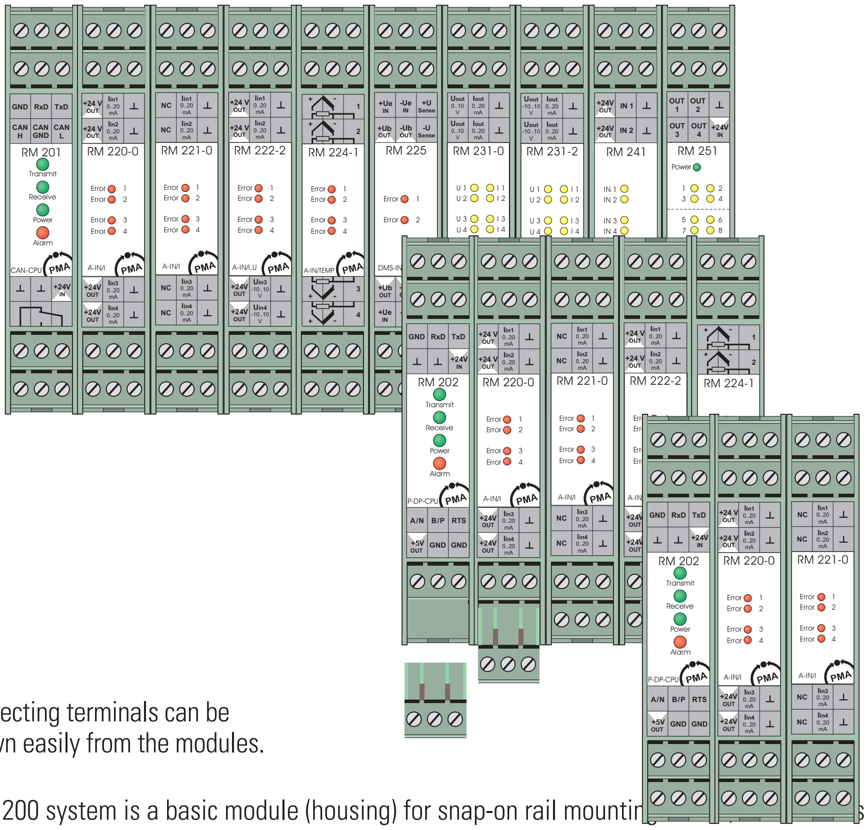
Both ends of the CANopen bus must be provided with a bus terminating resistor at (first and last node). For this, the bus terminating resistor provided in every KS98+ can be used. With the SIL switch closed, the terminating resistor is activated.
As default, the SIL switch is open
(see opposite).

Status display : CAN bus status



Value	Signification
1...42	Node number
NC	NoCheck: Node existence not checked so far / node not provided
Ck	Check: Node existence just being checked.
NR	NoResponse: No reply from this node, but node is required.
OK	Ready: Node has replied and is identified.
ES	EMStart: Node has provided an emergency message.
NA	NotAvailable: Node status is unknown.
PO	PreOperation: Node is in the PreOperational status.
Er	Error: Node is in error status.
Op	Operational: Node is in Operational status.
NU	NotUsed: Node is not required by an own lib function.
Wa	Waiting: Lib function waits for identification of this node.
Pa	Parameter setting: Lib function just setting the node parameters.
OK	Ready: Lib function has finished the parameter setting.
String	detemined node name

19.1 RM 211, RM212 and RM213 basic modules

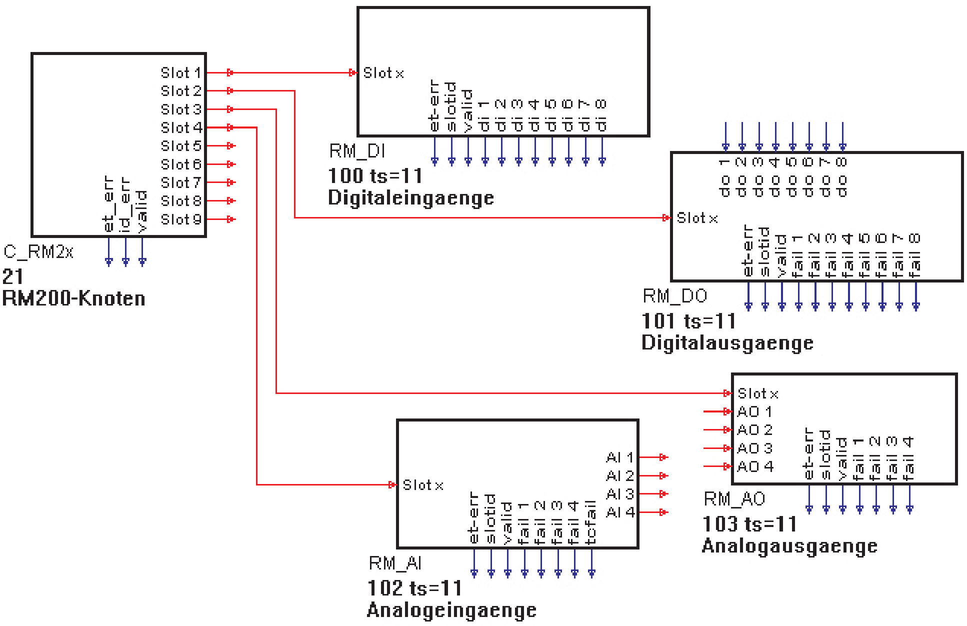


The connecting terminals can be withdrawn easily from the modules.

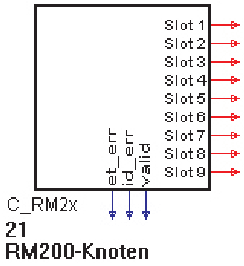
The RM 200 system is a basic module (housing) for snap-on rail mountin[illegible]s.
The left socket is always reserved for the RM 201 CANopen bus coupler module. Dependent of requirement, I/O modules or dummies can be plugged into the remaining sockets. The modules click in position in the basic module and can be released using simple tools for replacement (e.g. small screwdriver).

Don't insert or remove modules with the supply voltage switched on.

Partial engineering for communication with an RM200 node



19.2 C_RM2x (CANopen fieldbuscoupler RM 201)



Coupler module RM201 is fitted with an interface to the CAN bus and plugs into the first slot.
The other slots are provided for various I/O modules, which are polled cyclically via an internal bus.

Outputs

Analog Outputs	
Slot1 ... Slot9	Connection of RM modules RM_DI, RM_DO, RM_AI and RM_AO

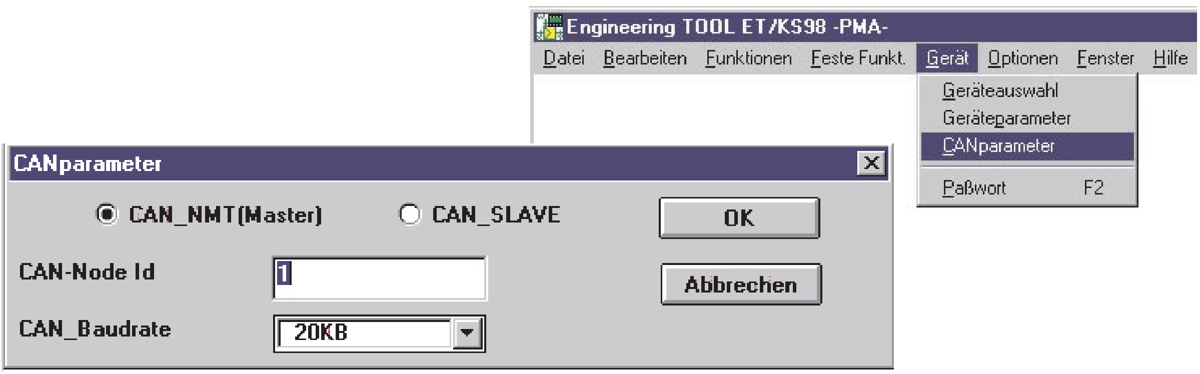
Digital Outputs		
et-err	**0 = no engineering error detected**	1 = reply from min. 2 nodes with identical node ID; → Change the addresses of connected instruments accordingly (e.g. DIP switches on RM 201).
id-err	**0 = correct node Id**	1 = wrong communication module ID no reply from any unit with the specified node ID; → Adjust the DIP switches on the connected RM 201 and on page "Parameter Dialog C_RM2x.
valid	0 = invalid data	1= data are valid

Unlike the other KS98 functions, only one data function may be connected to the analog outputs.

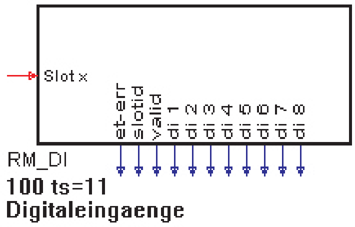
Parameters and configuration data

Parameter	**Beschreibung**	**Range**	**Default**
NodeId	RM201node address	2...42	32

Prerequisite for communication between KS98+ multifunction unit and CANopen field bus coupler RM 201 is that the CAN parameter setting is identical.
Adapt engineering tool settings and RM201 fieldbus coupler switch position.



19.3 RM_DI (RM 200 - (digital input module)



Function RM_DI handles the data from the connected digital input modules.

Inputs and outputs

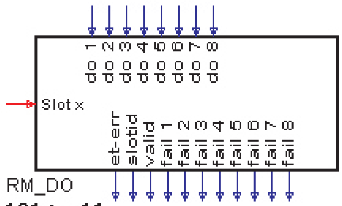
Analog input	
Slotx	Connection of one of the slot outputs of the RM200 node (C_RM2x)

Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions at a slot)
slotid	0 = correct slot assignment	1 = faulty slot assignment (wrong RM module inserted)
valid	0 = no data	1 = data could be received
di 1 ... di 8	1st to 8th digital input signal	

Parameter and configuration data

Configuration	Description	Value	Default
MTyp	Module type	0: RM241 = 4 x 24 VDC 1: RM242 = 8 x 24 VDC 2: RM243 = 4 x 243 VAC	0
Inv1 ... Inv8	Direct or inverse output of input signal 1? ... Direct or inverse output of input signal 8?	direct / inverse	direct

19.4 RM_DO (RM 200 - digital output module)



Function RM_DO handles the data from connected digital output modules.

Input and output modules

Analog input	
Slotx	Connection of one of the slot outputs of the RM200 node (C_RM2x)

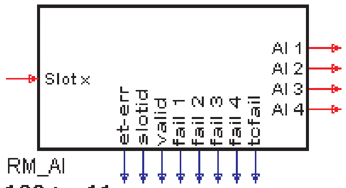
Digital inputs	
do 1 ... do 8	Set-points for digital inputs 1 to 8

Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions at a slot)
slotId	0 = correct slot assignment	1 = faulty slot assignment (faulty RM module fitted)
valid	0 = no data	1 = data could be received
di 1 ... di 8	1st to 8th digital input signal	

Parameter and configuration data

Configuration	Description	Range	Default
MTyp	Module type	0: RM251 = 8 x 24 VDC, 0,5 A 1: RM252 = 4 x relay (230 VDC) 2 A	0
Inv1 ... Inv8	Direct or inverse output of input signal 1? ... Direct or inverse output of input signal 8?	direct / inverse	direct
FMode1 ... FMode8	Output last signal or FState in case of communication failure?	no → no particular reaction / FStat value output	no
FState1 ... FState8	Output status in case of error	0/1	0

19.5 RM_AI (RM200 - analog input module)



Function RM_AI handles the data from connected analog input modules.

Inputs and outputs

Analog input	
Slotx	Connection of one of the slot outputs of the RM200 node (C_RM2x)

Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions at a slot)
slotId	0 = correct slot assignment	1 = faulty slot assignment (faulty RM module fitted)
valid	0 = no data	1 = data could be received
fail 1 ... fail 8	Measurement error at channel 1 to 4 (e.g. sensor break)	
tcfile	Temperature compensation error	

Analog outputs	
Ai 1...Ai 4	1st to 4th analog input signal

Parameter and configuration data

Configuration	Description	Range	Default
MTyp	Module type	0: RM221-0 = 4 x 0/4...20 mA 1: RM221-1 = 4 x -10/0...10 V 2: RM221-2 = 2 x 0/4...20 mA + 2 x -10/0...10 V 3: RM222-0 = 4 x 0/4...20 mA, TPS 4: RM222-1 = 4 x -10/0...10 V, potentiometer, TPS 5: RM222-2 = 2 x 0/4...20 mA + 2 x -10/0...10 V, potentiometer, TPS 6: RM224-1 = 4 x TC/Pt100, 16 bits 7: RM224-0 = 2 x TC, 16 bits 8: RM224-2 = 1 x -3...3V, 1x TC, 16 bits	
STyp 1 ... STyp 4	Input signal	1: type J = -120 ... 1200°C 2: type K = -130 ... 1370°C 3: type L = -120 ... 900°C 4: type E = -130 ... 1000°C 5: type T = -130 ... 400°C 6: type S = 12 ... 1760°C 7: type R = 13 ... 1760°C 8: type B = 50 ... 1820°C 9: type N = -109 ... 1300°C 10: type W = 50 ... 2300°C 30: Pt100 = -200 ... 850°C 40: standard signal = 0 ... 10V 41: standard signal = -10 ... 10V 50: standard signal = 4 ... 20mA 51: standard signal = 0 ... 20mA	
Unit 1 ... Unit 4	Temperature unit input 1 to 4 (only relevant with thermo-couple and Pt100 inputs)	0: unit = °C 1: unit = °F 2: unit = K	0
Tf 1 ... Tf 4	Filter time constant input 1 ... 4 in (s)	0 ... 999 999	0,5
x0 1 ... x0 4	Scaling start value input 1...input 4	-29 999 ... 999 999	0
x100 1 ... x100 4	Scaling end value input 1 ... input 4	-29 999 ... 999 999	100
Fail 1 ... Fail 4	Signal behaviour with sensor error at input 1...4	upscale downscale	←
X1in 1...4	Measured value correction input value Segment point 1 → input 1...4	-29 999 ... 999 999	0
X1out 1...4	Measured value correction output value Segment 1 → input 1...4	-29 999 ... 999 999	0
X2in 1...4	Measured value correction input value Segment point 2 → input 1...4	-29 999 ... 999 999	100
X2out 1...4	Measured value correction output value Segment point 2 → input 1...4	-29 999 ... 999 999	100

19.6 Potentiometer connection and calibration

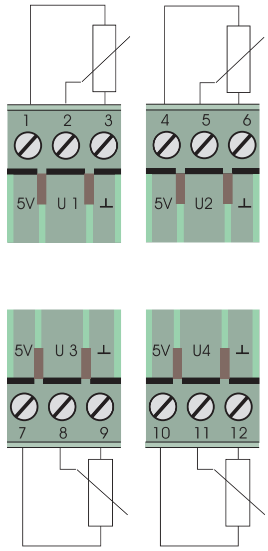
Connection:

Modules RM 222-1 and RM222-2 are also suitable for connection of potentiometers. Max. two potentiometers can be connected to module RM222-2 and max. four potentiometers can be connected to module RM 222-1.

For potentiometer measurement, a voltage divider circuit is used. The channels designed for voltage can be changed for potentiometer measurement pairwisely (by means of jumpers on the module circuit board)

only RM222-1

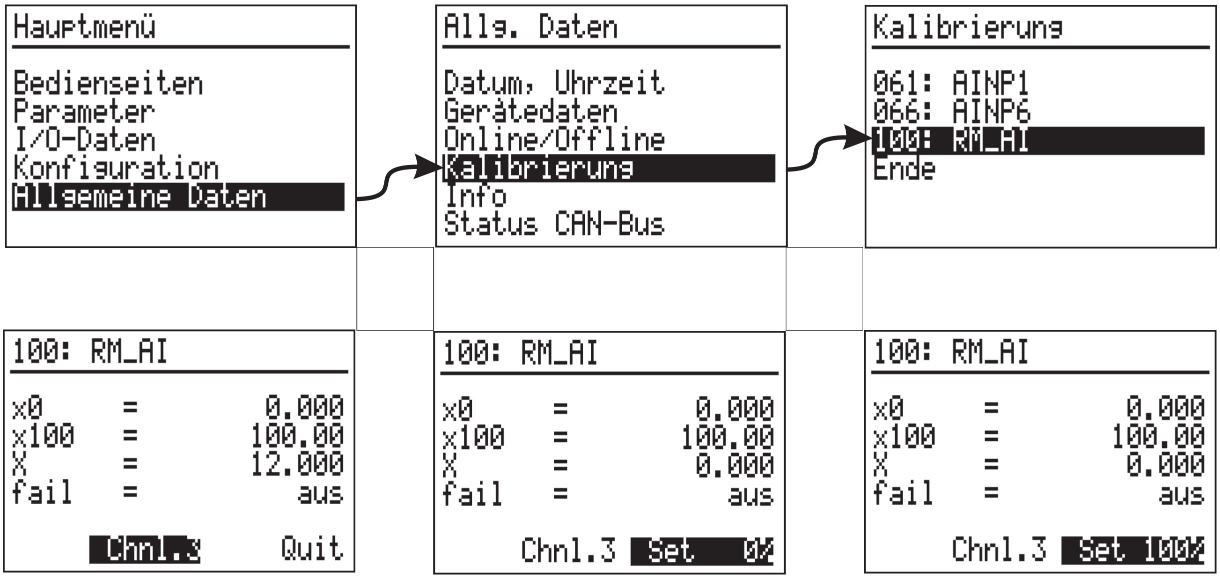
RM222-1 RM222-2



Uconst: Us = 5V DC (output instead of +24V OUT);
Short circuit proof current limiting: 20mA
Max. load: 4mA/channel; Σ I $\leq$ 20mA (can be distributed to the 4 module channels.
The min. resistance values must be
4 x 1000 Ω , 2 x 500 Ω or 1 x 250 Ω

Calibration:

In order to calibrate the potentiometer inputs, call up menu item Calibration.
For this, leave the operating menu, call up `Main menu` → `Miscellaneous`, select `Calibration`, and call up the module you wish to calibrate.

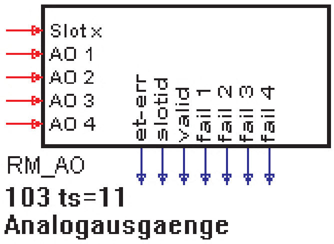


Start by selecting the channel you wish to calibrate.
Press key ▲ to select the channel number (`chnl.1`) and change it after acknowledgement via key ⏎

Subsequently, press `Quit` and change over to `Set 0%`. Press key ⏎. `Set 0%` starts flashing.
Bring the resistance value into the position for X0. The value valid for this channel appears on display `X.` Press key ⏎ again to store this value as `X0.`

Actuate `Set 0%` and change over to `Set 100%.` Press key ⏎. `Set 100%` starts flashing.
Now, bring the resistance value into the position for X100. The value valid for this channel appears on display `X`. Press key ⏎ again to store this actual value as X100.

19.7 RM_(RM200 - analog output module)



Function RM_AO handles the data from connected analog output modules.

Input and outputs

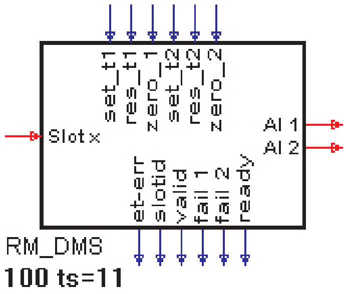
Analog inputs	
Slotx	Connection of one of the Slot outputs of the RM 200 node (C_RM2x)
AO 1...AO 4	1st to 4th analog output signal

Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions on a slot)
slotId	0 = correct slot assignment	1 = faulty slot assignment (wrong RM module fitted)
valid	0 = no data	1 = data could be received
fail 1 ... fail 4	Measurement error on channel 1 to 4 (e.g. sensor break)	

Parameter and configuration data

Configuration	Description	Range	Default
MTyp	Module type	0: RM231-0 = 4 x 0/4...20 mA / 4 x 0...10 V 1: RM231-1 = 4 x 0/4...20 mA / 2 x 0...10 V / 2 x -10...10 V 2: RM231-2 = 4 x 0/4...20 mA / 4 x -10...10 V	
OTyp 1 ... OTyp 4	Output signal	10: standard signal = 0 ... 10V 11: standard signal = -10 ... 10V 20: standard signal = 0 ... 20 mA 21: standard signal = 4 ... 20 mA	
x0 1 ... x0 4	Scaling start value input 1...input 4	-29 999 ... 999 999	0
x100 1 ... x100 4	Scaling end value input 1 ... input 4	-29 999 ... 999 999	100

19.8 RM_DMS strain gauge module



Function RM_DMS reads data from a special strain gauge module of KS98+ I/O extension with CANopen. Max. 2 strain gauges can be connected to the module. The measured values are available at outputs AI 1 and AI 2.

The two measurements can be influenced via digital command inputs, e.g. zero setting. Monitoring a new command (positive flank at one of the digital inputs) is restarted only when the “ready” output is “1”. The module position in the RM rack is determined by connection of analog input Slotx to the RM2xx node.

Important hint:

A special coupler module (RM201-1) must be used for operation of the strain gauge module. This coupler module cannot be combined with thermocouple modules. Moreover, the limitations as for coupler module RM201 (e.g. max. 4 analog input modules) are applicable.

Digital inputs:

- set_t1 Set tare strain gauge channel 1. The actual weight is not stored continuously as tare (packaging weight). The following measurements provide the net weight.
- res_t1 Reset tare strain gauge channel 1. The tare value is set to 0. Gross weight= net weight.
- zero_1 Zero setting of strain gauge channel 1 measured value. The actual measured value is stored as a zero value in a non-volatile memory.
- set_t2 Set tare strain gauge channel 2. The actual weight is buffered as tare (packaging weight). The following measurements provide net weight.
- res_t2 Reset tare strain gauge channel 2. The tare value is set to 0. Gross weight=net weight.
- zero_2 Zero setting of the strain gauge channel 2 measured value. The actual measured value is stored as zero in the non-volatile memory..

Digital outputs:

- et-err 0 = no engineering error
  1 = engineering error (several module blocks at a slot output).
  slots not connected.
- slotId 0 = correct slot allocation
  1 = faulty slot allocation (module type).
  Faulty coupler module
- valid 0 = no data
  1 = data could not be received

fail 1 faulty connection or measurement error on channel 1
fail 2 faulty connection or measurement error on channel 2
ready ready message after command handling

Analog inputs:

- connection of one of the slot outputs of the RM201-1-node block

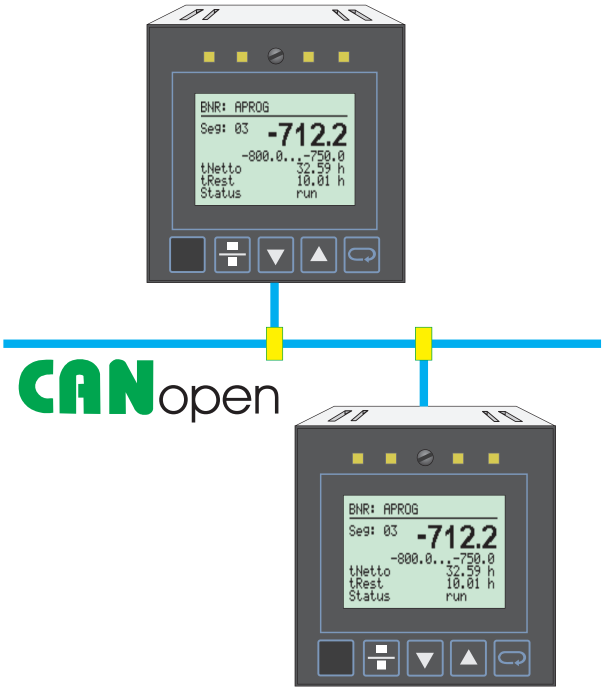
Analog outputs:

- AI 1st measured value of strain gauge channel 1
- AI 2nd measured value of strain gauge channel 2

Parameters:

MTyp 1/2	module type 0: RM225 = strain gauge	
STyp 1	/2 0: -4 +4mV/V	
Unit 1/2	mV/V	
Tf 1/2	filter time constant input 1 ... 2 in (s) 0 ... 999 999	(0,5)
x0 1/2	scaling start value input 1 ... 2 -29 999 ... 999 999 (0)	
x100 1/2	scaling end value input 1 ... 2 -29 999 ... 999 999	(100)
Fail 1/2	signal action in case of sensor error 0:upscale 1:downscale	
X1in 1/2	measured value correction input value segment point 1 > input 1...2 -29 999 ... 999 999	(0)
X1out 1/2	measured value correction output value segment point 1 > input 1...2 -29 999 ... 999 999	(0)
X2in 1/2	measured value correction input value segment point 2 > input 1...2 -29 999 ... 999 999	(100)
X2out 1/2	measured value correction output value segment point 2 > input 1...2 -29 999 ... 999 999	(100)

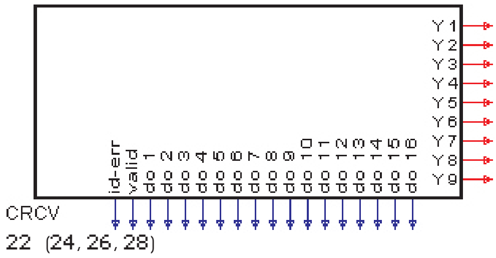
20 KS 98 cross communication



CROSS COMMUNICATION
Whilst data exchange between KS 98+ and RM200, KS800 or KS816 must be done exclusively via KS98+ as a master, direct "cross communication" is possible.
Data exchange between several KS 98+ of a CAN network is via send modules (CSEND; block numbers 21, 23, 25, 27) and receive modules (CRCV; block numbers 22, 24, 26, 28).
Max. 9 analog values and 16 digital statuses from the relevant engineering can be transmitted per send/receive module. The sender sends the data together with its node address and block number.
The receiver checks, if the messages correspond with the adjusted send address, and if the sender block number is by "1" lower than its own one.

For BUS terminating resistor, see page: 117

20.1 CRCV (receive mod. block no's 22,24,26,28 no.56)



Function CRCV can receive data from a different KS98+. The data of the other multifunction unit are made available by means of the CSEND function. Hereby, the CSEND block number is by 1 lower than the CRCV block number.

CRCV no. 22 reads the data of another KS98+ from CSEND no. 21
CRCV no. 24 reads the data of another KS98+ from CSEND no. 23
CRCV no. 26 reads the data of another KS98+ from CSEND no. 25
CRCV no. 28 reads the data of another KS98+ from CSEND no. 27

Outputs

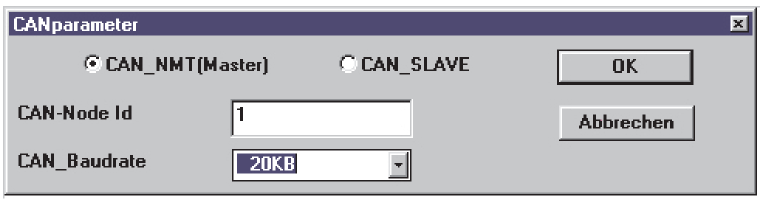
Analog outputs	
Y1...Y9	Analog output values 1 to 9

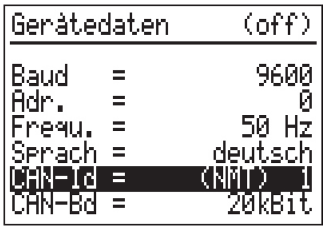
Digital outputs		
id-err	0 = correct node Id	1 = faulty node Id
valid	0 = no data	1 = data could be received
do 1 ... do 16	Status values 1 to 16	

Parameter and configuration data

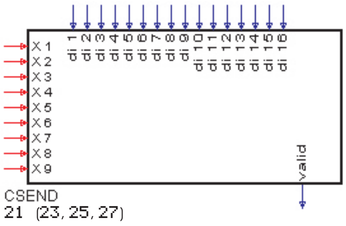
Configuration	Description	Range	Default
NodeId	Node address of the sending KS98+ (The sending KS98plus is adjusted accordingly in engineering tool window "CANparameter ".) → see *1		

* 1) The node address of the sending KS98plus is adjustable in engineering tool window "CANparameter" or via the instrument parameters on the front panel (during off-line mode).





20.2 CSEND (Send mod. blockno.'s 21, 23, 25, 27 - No. 57)



Function CSEND provides data for other KS98+ units on the CANopen bus. The data can be read by the other multifunction units using the CRCVfunction.

Inputs and outputs

Analog inputs	
X1...X9	**Analog values 1 to 9, which are sent.**

Digital inputs	
di1...di9	**Digital values 1 to 16, which are sent.**

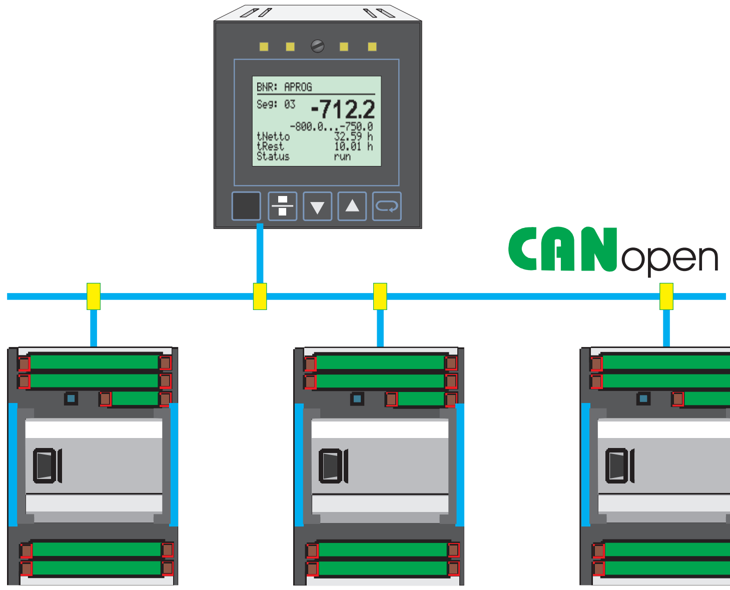
Digital output		
valid	**0 = invalid data (e.g. no KS98+ but only KS98)**	**1 = data could be received**

Parameter und Konfigurationsdaten

Configuration	Description	Range	Default
delta	**Change from which a new send operation is started.**	0,000...999 999	0,1

Transmission is at intervals of 200ms.
Note that there is a risk of data loss for values which are available only during 100 ms.

21 Connection of KS 800 and KS 816

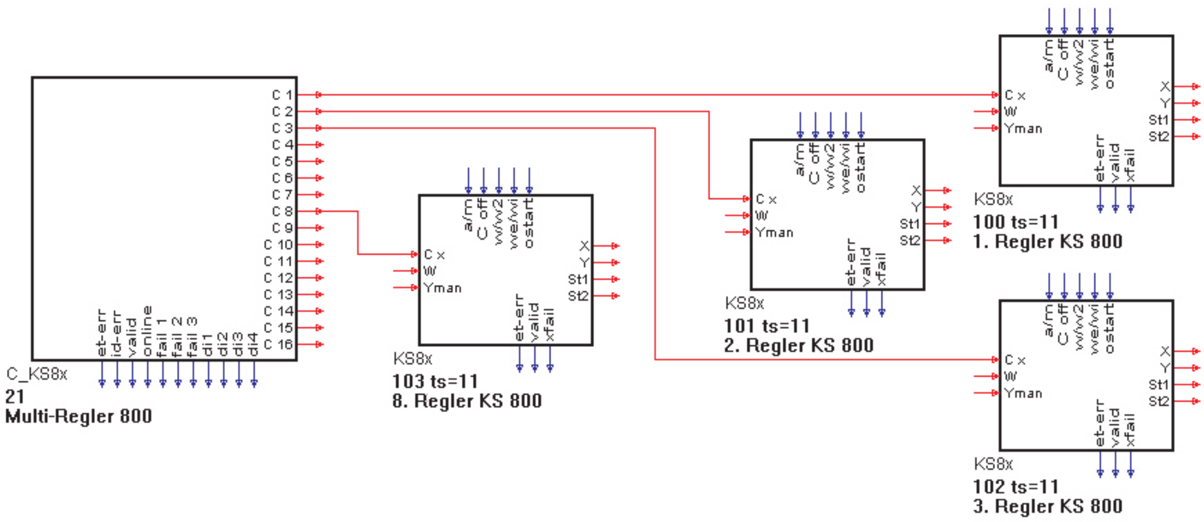


Function blocks C_KS8x and KS8x can be used for communication of multifunction unit KS98+ and multi-channel temperature controllers KS 800 and KS 816.
A node function C_KS8x is allocated to each KS 800 or KS816.

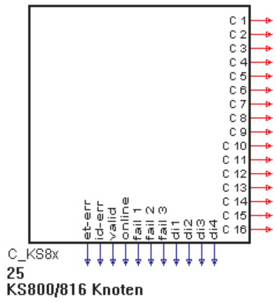
The KS8x functions are allocated to the various controllers of KS 800 (up to 8 controllers) or KS 816 (up to 16 controllers).

For BUS terminating resistor, see page 117

Partial engineering for communication with multiple channel temperature controllers KS800 and KS816



21.1 C_KS8x (KS 800 and KS 816 node function - no. 58)



Node function C_KS8x provides the interface to one of the multi-channel temperature controllers KS 800 or KS 816. Analog outputs C1 … C16 can be used to connect the KS8x functions, which represent each a controller of KS 800 (max. 8 controllers) or of KS 816 (max. 16 controllers).

Unlike the other KS98 functions, only one data function can be soft-wired to each analog output.
Prerequisite for communication of KS98+ multi-function unit and KS800 or KS816 is the complying adjustment of the CAN parameters. (→ see *1)).

Outputs

Analog inputs	
C1...C16	**Connection of the KS8x functions (single controllers in KS800 / KS816)**

Digital outputs		
et-err	0 = no engineering error	1 = engineering error (different node function at the same KS800)
id-err	0 = correct node	1 = faulty node Id (no KS800 / KS816 replied under the configured node ID)
valid	0 = no data	1 = data were received
online	0 = KS800/816 is off-line	1 = KS800/816 is on-line
fail 1	0 = no fail at do1...do12	1 = fail at do1...do12
fail 2	0 = no fail at do13...do16	1 = fail at do13...do16
fail 3	0 = no heating current short circuit	1 = heating current short circuit
di1	di1status	
di2	di2 status	
di3	di3 status	
di4	di4 status	

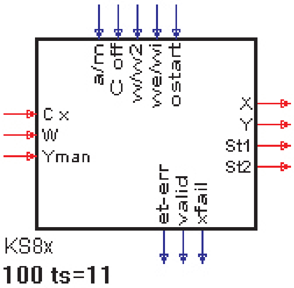
Parameter und Konfigurationsdaten

Configuration	Description	Range	Default
NodeId	KS800/KS816 node address	2...42	2

The data from the various controllers are read cyclically.
Maximally at intervals of 1.6 seconds (KS800) or 3.2 seconds (KS816), all data are updated.

* 1) The parameters for the CANopen bus are adjustable in engineering tool window "CANparameter" or via the instrument parameters on the front panel (ET98 → Device → CANparameter).

21.2 KS8x (KS 800/ KS 816 controller function - no. 59)



Each KS8x function handles a controller of KS 800 or KS 816. The analog and digital inputs can be used to send the control signals to the controller in KS800/16. The analog outputs provide the process and controller values.

Inputs and outputs

Analog inputs	
C x	**Connection to one of the C1...C16 outputs of node function C_KS8x**
W	**Controller set-point**
Yman	**Correcting variable in manual mode**

Digital inputs		
a/m	0 = controller is in automatic mode	1 = controller is in manual mode
C off	0 = controller is switched on	1 = controller is switched off
w/w2	0 = controller is in automatic mode	1 = 2nd set-point is active (safety set-point)
we/wi	0 = external set-point is active	1 = internal set-point is active
ostart	0 = don’t start self-tuning	1 = start self-tuning

Digital outputs		
et-err	0 = no engineering error	1 = engineering error (several KS8x controller functions on a controller channel)
valid	0 = no data	1 = data were received
xfail	0 = no sensor fail	1 = sensor fail

Analog outputs		
X	Controller process value	
Y	Controller correcting variable	
St1	Statusbyte 1	For an engineering example to evaluate St1 and St2, see the next page.
St2	Statusbyte 2	

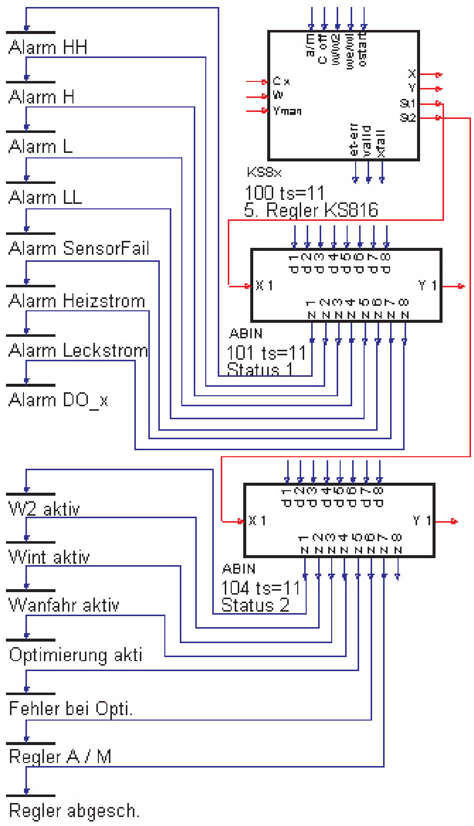
St.1 Statusbyte 1

Bit	Value	Description
0	1	HH alarm
1	2	H alarm
2	4	L alarm
3	8	LL alarm
4	16	sensor fail alarm
5	32	heating current alarm
6	64	leakage current alarm
7	128	alarm DOx

St.2 Statusbyte 2

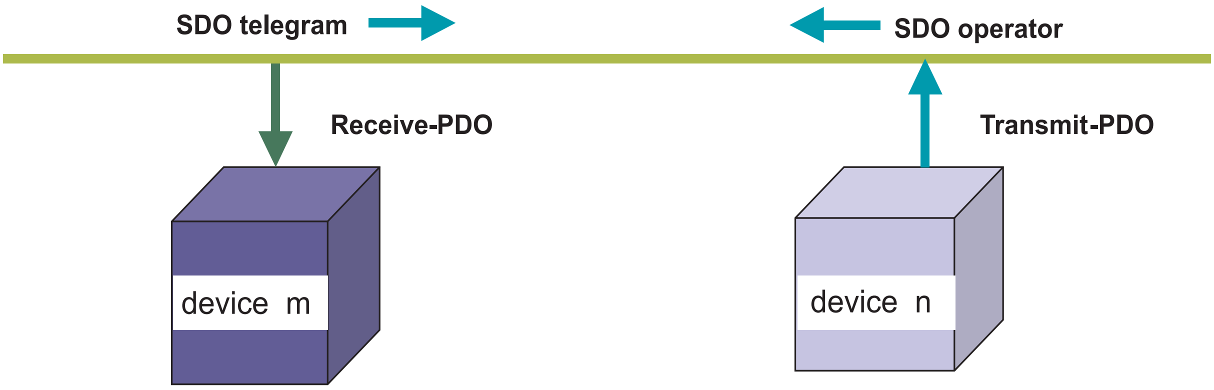
Bit	Value	Description
0	1	W2 active
1	2	Wint active
2	4	Wstart active
3	8	self-tuning active
4	16	self-tuning error
5	32	controller A / M
6	68	controller switched off
7	128	- - - -

Beispielengineering zur Auswertung St1/ St2



22 Description of KS98 CAN bus extension

There are various modes for KS98+ communication via the CAN bus. The unit can be master for handling the NMT services (NMT = Network ManagementT), or slave, can send or receive PDOs (PDO = process data object) cyclically or send SDO telegrams asynchronously (SDO = service data object). A KS98+ can contact any bus parties simultaneously with other KS98+, allocated remote IOs, KS800 multi-controllers and up to 40 sensors or actuators, and via asynchronous telegrams. Max. 42 CAN nodes can be addressed.



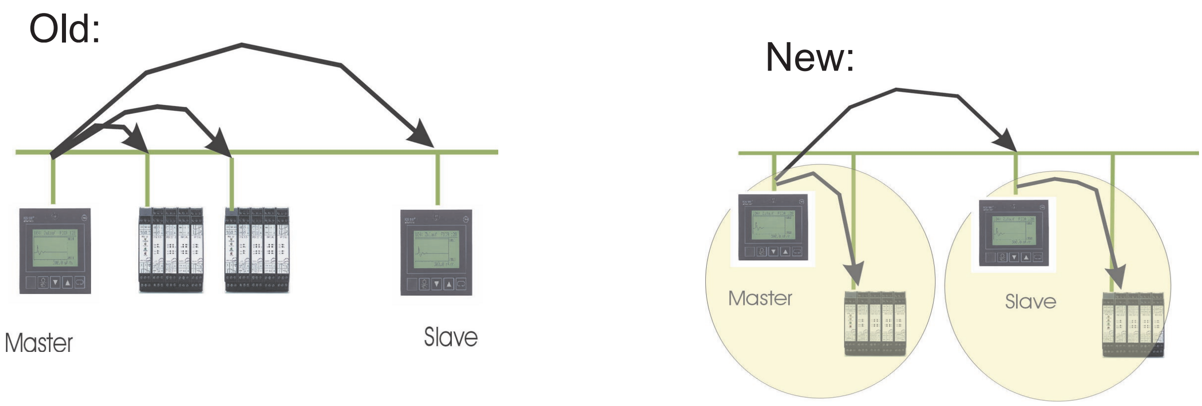
Data-receiver

Process Data Objects (PDO)
Process data for fast exchange
- one party sending = all parties can read
(Producer / consumer concept)
max. 8 bytes for data / message
non-confirmed message
snchronous or event-triggered
Priority control via address

Data-producer

Instrument-specific data contents
Service Data Objects (SDO)
for data without real-time requirement
asynchronous, confirmed messages
Repartition over several telegrams is possible
Data addressing via indexes in the object directory (index, subindex).

KS98 handles guarding tasks as a master or a slave with an own local RM node. Display is in the CAN status window.



However, there are limits to the performance of the bus parties and the bus itself. The dynamic operations of the bus can be evaluated only by statistics.
The resulting bus and interface load of an instrument is dependent on thedetails of the communication structure and can be estimated only, if the behaviour of the individual parties is known exactly. In the following, properties and effects of various bus parties are explained and figures and facts are presented. Information on the COB-IDs consumed internally at PMA is given in the annex. This information should be taken into account when adding instruments from other manufacturers.

KS98+ CAN communication features

Every message on the bus activates the KS98 interrupt handler and loads the processor. The message is analyzed and queued, if the destination of the message is the own address. This queue is handled in the idle task and during the cyclical system processing phase (at intervals of 100ms).
70% of the CPU capacity is reserved for the engineering. This time is considered as 100 % in the KS98 ET timing dialogue. I.e. Min. 30ms are available for general tasks and communication. Included are front and rear instrument interface processing and Profibus handling. However, these loads are insignificant, because, for example, front and rear interface can only receive one telegram per 100 ms. This means that the CAN communication causes the largest part of the CPU load.
The PDO handling program is activated, as soon as the processing phase for the engineering within a cycle is finished (idle-task). Hence, more than 30 % of the processor capacity may be available for CAN communication with small engineerings. The user can decide freely and at his own responsibility how to use these reserves can.

Receive PDOs

The interrupt handler requires approx. 0,16ms for each PDO.
The event queues comprise 4 x 80 items. There is a queue for all send messages, another one for all PDO receive messages, still another one for the network receive messages and still another one for the SDO receive messages.
The queues are handled at intervals of 100 ms and during the idle task.
This means that no more than 80 PDOs per 100ms may be received.
The PDO handling is a processor load of approx. 1,2 ms for each individual PDO.
Blockwise handling of 50 receive PDOs takes KS98 18 ms (19 ms, if the same number of PDOs for other receivers are rejected).
Although the load of the basic communication blocks (C_RM2X, CPREAD, ...) cannot be allocated to a time slot, it is assigned as a fixed value to the engineering portion automatically.

Send PDOs

The load for transmitted PDOs is nearly the same as for receive PDOs (18ms / 50 PDOs), however, sending is not cyclical.
PDOs are sent only, when a value has changed (threshold adjustable with CSEND, otherwise, there will be a change of accuracy of the transmitted data format). At the latest after 2 seconds, the values are sent again also if unchanged.
This reduces the output load by an unpredictable percentage. A filter can be used to reduce the transmission frequency of instable data.

Estimation of CAN bus activities of various instruments

For reducing the data traffic between PMA instruments, PDOs are transmitted only in case of data changes.
The changes are read with the accuracy of the used data format (LSB).

KS800 communication

Both synchronous and asynchronous communication are used for KS800 communication. By configuration, one PDO is defined as synchronous and one PDO is defined as asynchronous.

A Sync message is sent at intervals of 200ms.

This is followed by reception of a PDO containing the data of one controller channel by each KS800/816 . I.e. refreshing of 8 channels takes 1,6 seconds.
The internal KS800/816 cycle for handling a controller channel is 63,5 ms. If a channel status or correcting variable change occurs during this cycle time, KS800/81 sends 1 PDO asynchronously.

RM 200

Data transmission in both directions is asynchronous. Data are transmitted only if changed (only the related PDOs). Checking, if changes were made is dependent on the accuracy of the data format (LSB). In both directions, the min. refresh rate is 100 ms.

Max. 5 PDOs + 1status PDO are sent by the RM node dependent on the number of modules in the nodes.
Max. 5 PDOs are sent to the RM node by KS98.

KS98+cross communication

Data transmission is asynchronous. Data are transmitted only when changes occurred (only the related PDOs). The min. refresh rate is 200 ms.

Max. 5 PDOs are sent dependent on the quantity of data connected to CSEND.
Max. 5 PDOs are received by KS98.

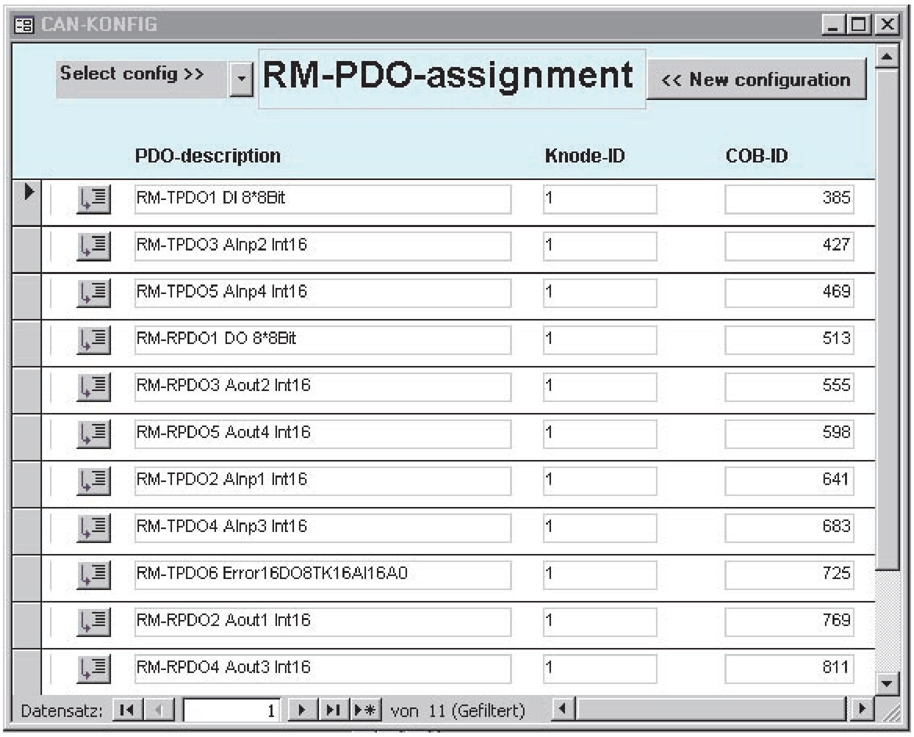
Instruments from other manufacturers

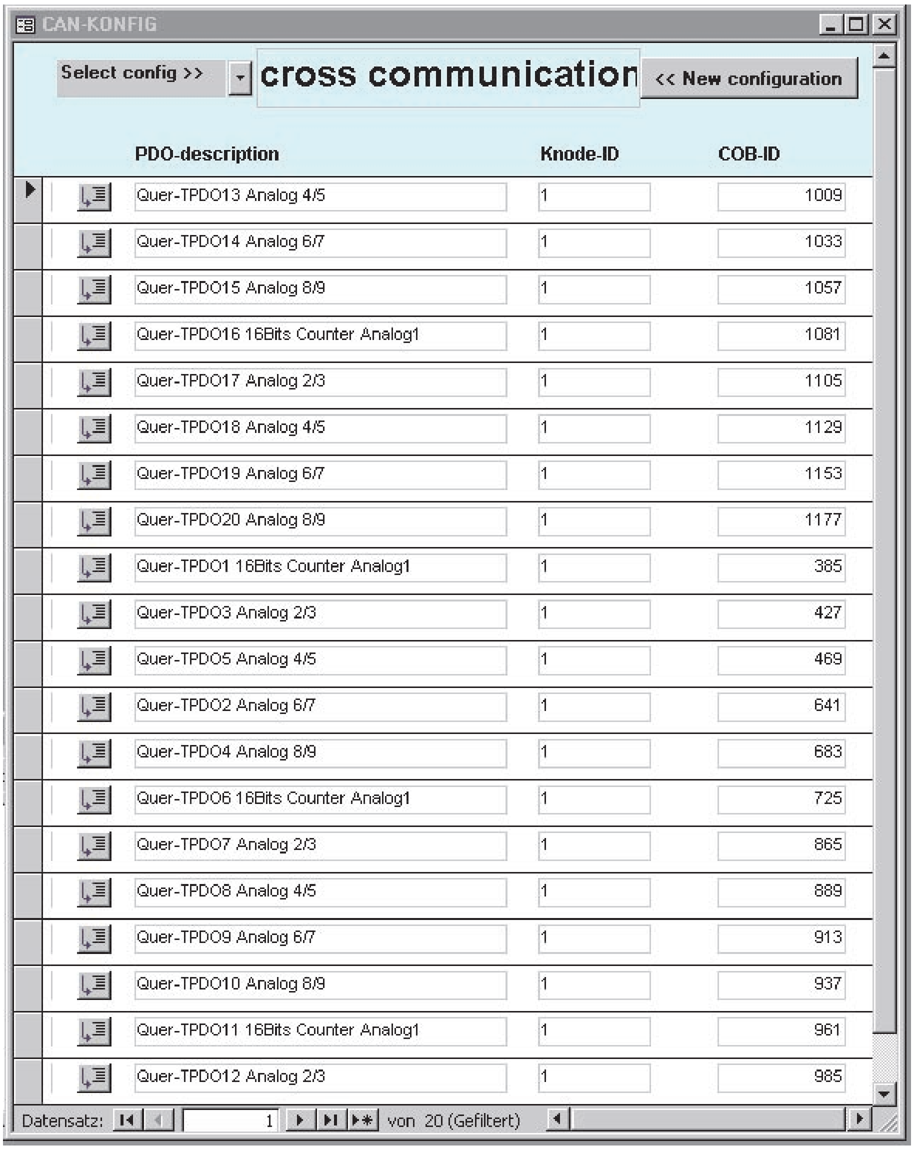
Instruments from other manufacturers - sensors / actuators – can be addressed via synchronous data communication (send and receive PDOs), or using asynchronous data communication via SDOs. For reduction of the bus activities, checking for data changes is done by the sending side.
PDO reception can be influenced only by increasing the “Inhibit time” on the sensor side, in order to prevent information from being sent more frequently than once per 100 ms (KS98 calculation cycle). Received data bytes can be converted into the internal format flexibly using function block AOCTET . The operating principle of the block for the sending side is equal.
The receive and send interfaces (CPREAD/CPWRIT) are handled at intervals of 100 ms.
In block number range 21-40, max. 40 PDO addresses (COB-ID=Communication OBject Identifier: basic address + node address) can be addressed.
The hearbeat protocol, which is offered by some manufacturers, is not supported.
The data definition according to DS301 V4.0 complies with the Intel notation.

Recommendation for safe operation:
Bus load limitation
≤ 100 telegrams / 100 ms
Baudrate ³ 250 kbit/s = 250m distance

Limitation of PDOs handled in the unit
≤ 50 telegrams / 100 ms (send/receive)

Send frequency for sensors ³ 100ms (inhibit time)
COB-ID allocation example for internal PMA CAN communication for node address 1:



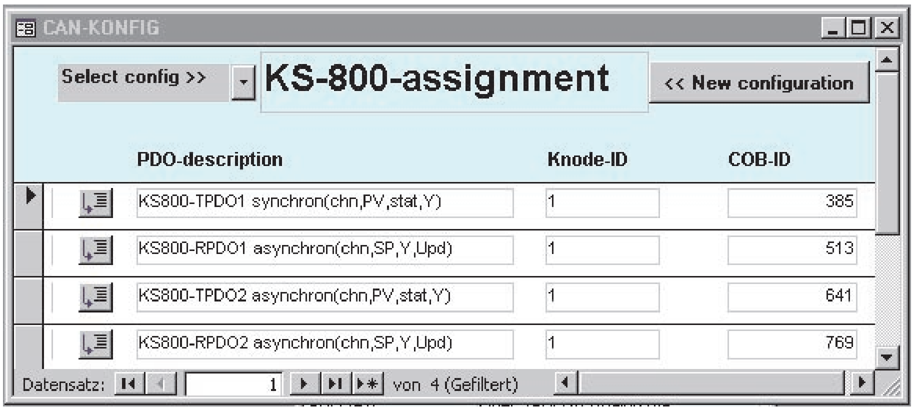


CAN-KONFIG

Select config >> cross communication << New configuration

PDO-description	Knode-ID	COB-ID
Quer-TPDO13 Analog 4/5	1	1009
Quer-TPDO14 Analog 6/7	1	1033
Quer-TPDO15 Analog 8/9	1	1057
Quer-TPDO16 16Bits Counter Analog1	1	1081
Quer-TPDO17 Analog 2/3	1	1105
Quer-TPDO18 Analog 4/5	1	1129
Quer-TPDO19 Analog 6/7	1	1153
Quer-TPDO20 Analog 8/9	1	1177
Quer-TPDO1 16Bits Counter Analog1	1	385
Quer-TPDO3 Analog 2/3	1	427
Quer-TPDO5 Analog 4/5	1	469
Quer-TPDO2 Analog 6/7	1	641
Quer-TPDO4 Analog 8/9	1	683
Quer-TPDO6 16Bits Counter Analog1	1	725
Quer-TPDO7 Analog 2/3	1	865
Quer-TPDO8 Analog 4/5	1	889
Quer-TPDO9 Analog 6/7	1	913
Quer-TPDO10 Analog 8/9	1	937
Quer-TPDO11 16Bits Counter Analog1	1	961
Quer-TPDO12 Analog 2/3	1	985

Datensatz: 1 von 20 (Gefiltert)



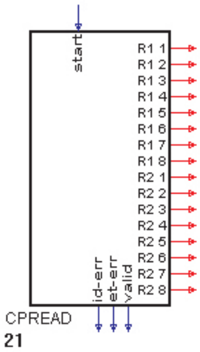
CAN-KONFIG

Select config >> KS-800-assignment << New configuration

PDO-description	Knode-ID	COB-ID
KS800-TPDO1 synchron(chn,PV,stat,Y)	1	385
KS800-RPDO1 asynchron(chn,SP,Y,Upd)	1	513
KS800-TPDO2 asynchron(chn,PV,stat,Y)	1	641
KS800-RPDO2 asynchron(chn,SP,Y,Upd)	1	769

Datensatz: 1 von 4 (Gefiltert)

22.1 CPREAD (CAN-PDO read function)



Function CPREAD is used for read access to instrument PDOs. Due to the normal quantity of min. 2 PDOs per instrument, the data quantity of 2 PDOs 2 with 2 COB-IDs was grouped in one block.
Node address and COB-ID (CAN-OBject IDentifier) parameter setting is in the block. Moreover, node guarding for monitoring the CAN communication to the specified node can be switched on.
Data provided by the instrument must be interpreted according to the instrument specification.
Groups of 4 transmitted bytes can be converted into different data types.
For this purpose, a conversion function for converting and inverting 1 to 4 bytes into a parameterizable data type (see function AOCTET) is available.
Examples: R1+R2 > Int16 / R1+R2+R3+R4 >Long

Important note: The heart beat protocol is not supported. If an instrument can be operated only via "heart beat", the guarding function must be switched off.

Digital inputs:

start The function is active with the input not connected, or if start=1 is connected.

Digital outputs:

et-err 0 = no engineering error
1 = no CAN-HW (KS98 type)
multiple node monitoring

id-err 0 = correct node id
1 = faulty node id or instrument does not reply
specify own node ID as "NodeId"
no free receive PDOs (RPDO)

valid Bit follows node status with the node guarding active
(0="preoperational", 1="operational") always 1 with node guarding switched off

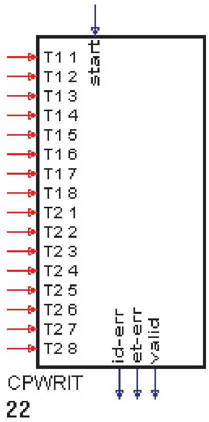
Analog outputs:

R1 1 ...R1 8 1st to 8th analog input value in byte format (8-bit) for COB-ID 1
R2 1 ...R2 8 1st to 8th analog input value in byte format (8-bit) for COB-ID 2

Configuration parameters (can be changed only during OFFLINE):

NodeId CAN node address
Guard node guarding off/on
COBID1 decimal ID of the first CAN object identifier
COBID2 decimal ID of the second CAN object identifier

22.2 CPWRIT (CAN-PDO write function)



Function CPWRITE is used for write access to instrument PDOs. Because of the normal quantity of min. 2 PDOs per instruments, the data quantity of 2 PDOs 2 with 2 COB-IDs was grouped in a block.
Node address and COB-ID (CAN-OBject IDentifier) parameter setting is in the block. Moreover, node guarding for monitoring the CAN communication to the specified node can be switched on.
Data sent to the instrument must be interpreted according to instrument specification. Groups of 4 transmitted bytes represent different data types.
To provide the bytes according to the required data type, a conversion function for transforming the value in the engineering into 1 to 4 bytes is available (see function AOCTET).
Examples: R1+R2 > Int16 / R1+R2+R3+R4 >Long

Important note: The heart beat protocol is not supported. If an instrument can be operated only via "heart beat", the guarding function must be switched off.

Digital inputs:

start The function is active, unless the input is connected, or if start=1 is connected.

Digital outputs:

et-err 0 = no engineering error
1 = no CAN-HW (KS98 type)
multiple node monitoring
id-err 0 = correct node id
1 = faulty node id or the instrument does not reply
own node ID was specified as "NodeId"
no free send PDOs (TPDO)
valid bit follows the node status with the node guarding active
(0="preoperational", 1="operational") always 1 with the node guarding switched off

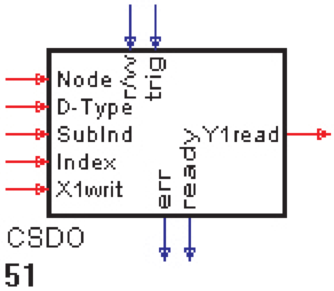
Analog outputs:

T1 1 ...T1 8 1st to 8th output value in byte format (8-bit) for COB-ID 1
T2 1 ...T2 8 1st to 8th analog output value in byte format (8-bit) for COB-ID 2

Configuration parameters (can be changed only during OFFLINE):

NodeId CAN node address
Guard node guarding off/on
COBID1 decimal ID of the first CAN object identifier
COBID2 decimal ID of the second CAN object identifier

22.3 CSDO CAN-SDO function



Function CSDO permits access to the CAN bus by means of SDOs (Service Data Objects). SDOs are used for asynchronous data exchange without real-time inquiry.

Transmission started by the trigger input is always confirmed by the receiver, possibly during data inquiry along with value transmission. Reception of the confirmation is indicated by a logic 1 at the “ready” output. A new command can be generated via the positive flank at trig only with “1” indicated by the “ready” output.-

Data required for command generation can be adjusted as parameters or connected as values to the inputs. As soon as a connection at an input was made, the relevant parameter looses its function. In this case, the value applied to the input is valid. Data (command) addressing in the connected instrument is done via indexes (index / sub-index), which is described in the CAN instrument documentation.

A value to be transmitted is connected to X1writ (or parameter “value”). A received value is output at Y1read. Y1read is set to 0 after power-on, after an error (“err” = 1) and after a data output.

With RM modules provided in the KS98 engineering, and for addressing the same nodes also via a CSDO block , the trigger should be interlocked with the valid bit of the RM-200 block. During access to RM nodes which are handled already by KS98 in the background, there may be start-up collisions the consequences of which are removed only by restarting KS98.

Important note: The heart beat protocol is not supported. If an instrument can be operated only via “heart beat”, the guarding function must be switched off.

Digitale Eingänge:

r/w Zugriffsart: 0 = lesen, 1 = schreiben

Analoge Eingänge:

Node dezimale CAN-Knotenadresse,1..42
(KS98+ bildet den CAN Object Identifier gemäß CiA DS301, Knoten ID + 600H)

D-Type Datentyp des angeschlossenen Wertes, 0..6. Folgende Datentypen stehen zur Verfügung

0:	Uint8
1:	Int8
2:	Uint16
3:	Int16
4:	Uint32
5:	Int32
6:	Float

SubInd Adressierung in Objektverzeichnis 1..255
Index Adressierung in Objektverzeichnis 1..65535
X1writ Datenwert –29999 ... 999999)

Digital outputs:

err	0 = no error 1 = error detected.

Possible errors:

- Faulty KS98 hardware. KS98+ expected.
- The trigger input is not connected.
- No reply or faulty reply from the instrument.
- Instrument replies an inquiry with an error message.
- Min. one parameter or connected value is out of limits.

ready	0 = transmission is being handled. So far, no confirmation was received. 1 = transmission completed. Ready for the next command.

Analog outputs:

T1 1 ...T1 8	1st to 8th analog output value in byte format (8-bit) for COB-ID 1
T2 1 ...T2 8	1st to 8th analog output value in byte format (8-bit) for COB-ID 2

Parameters (can be changed during operation):

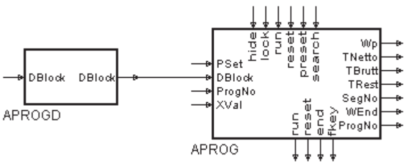
Access	access mode: 0 = read, 1 = write
NodeId	decimal CAN node address,1..42 (KS98+ forms the CAN Object Identifier according to CiA DS301, node ID + 600H)
D-Type	data type of the connected value, 0..6. The following data types are available

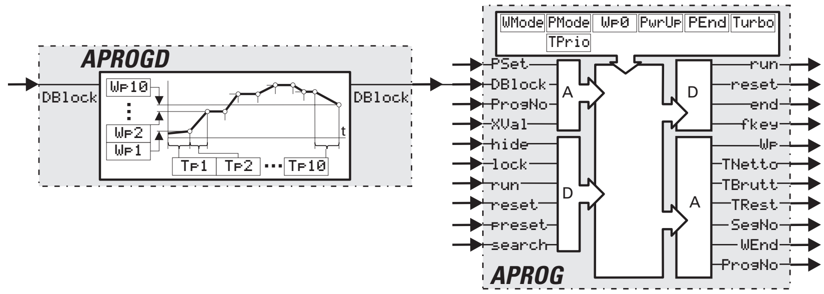
0:	Uint8
1:	Int8
2:	Uint16
3:	Int16
4:	Uint32
5:	Int32
6:	Float

SubInd	address in object directory 1..255
Index	address in object directory 1..65535
Wert	data value –29999 ... 999999)

23 Programmer

23.1 APROG (analog programmer) / APROGD (APROG data)



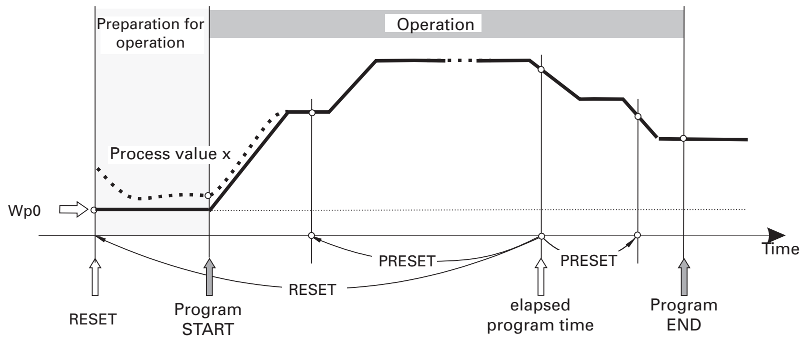


General

An analog programmer comprises a programmer (APROG) and min. one data block (APROD), whereby output DBlock of the APROGD is connected with input DBlock of the APROG. By connection of several of these cascadable functions (each with 10 segments), a programmer with any number of recipes with any number of segments can be realized. Limiting is only in the number of available block numbers and in the calculation time.
The data block has an analog output, at which the own block number is made available. This information is read-in by the programmer and used for segment data addressing. If an error with segment data addressing is detected, the reset value is output (status display on operating page: `Error').

After an engineering downlaod, Seg 0 is output (reset). If run is not connected, stop is used.

Definition of the analog programmer



Inputs/outputs

Digital inputs (APROG):		
hide	Display suppression (with hide = 1 the page is not displayed in the operation).	
lock	Adjustment blocking (with lock = 1 the values are not adjustable by means of keys ▲▼).	
run	Program stop/run (0 = stop, 1 = run)	reset has highest priority
reset	Program continue/reset (0 = continue, 1 = reset)	
preset	Program preset (1 = preset)	
search	Start programm search run (1 = search)	

Digital outputs (APROG):	
run	Status program stop/run (0 = program stop ; 1 = program run)
reset	Status program reset (1 = program reset)
end	Status program end (1 = program end reached)
fkey	Status A/M key / interface function `fkey' (:pressing key causes switch-over (0 or 1))

Analog inputs (APROG):	
PSet	Preset value for program
DBlock	Block number of 1st data function `APROGD'
ProgNo	Required program number (recipe)
XVal	Value for search run

Analog inputs (APROGD):	
DBlock	Block number of cascaded data function `APROGD'

Analog outputs (APROG):	
Wp	Programmer set-point
TNetto	Net program time (] Trun)
TBrutt	Gross program time (] Trun +] Tstop)
TRest	Programmer rest time
SegNo	Actual segment number
WEnd	End value of actual segment
ProgNo	Actual program number (recipe)

Analog outputs (APROGD):	
DBlock	Own block number

Parameter and configuration data

Parameter APROG	Description		Range	Default
WMode	Change mode:	Ramp	Ramp	←
		Step	Step	
PMode	Preset Mode:	Preset to segment	Pres.time	←
		Preset to time	Pres.seg.	
TPrio	Start mode in search run	Gradient has priority	Grad.prio	←
		Segment/time has priority	Time prio	
Wp0	Program set-point at reset		-29999 ... 999 999	0

Parameter APROGD	Description	Range ET	Range Unit	Default ET	Default Unit
Tp 1	Time for segment 1 (①)	0 ... 59 999	0:00...999:59	OFF	--:--
Wp 1	Segment end set-point in segment 1	-29999 ... 999 999		0	0
Tp 2	Time for segment 2 (①)	0 ... 59 999	0:00...999:59	OFF	--:--
Wp 2	Segment end set-point in segment 2	-29999 ... 999 999		0	0
...					
Tp 10	Time for segment 10 (①)	0 ... 59 999	0:00...999:59	OFF	--:--
Wp 10	Segment end set-point in segment 10	-29999 ... 999 999		0	0

Enter the time for a segment in seconds or minutes into the engineering tool dependent of configuration (Turbo). Entry into the unit is in Hrs:Min or Min:Sec. In addition to the range, a switch-off value can be entered (ET: OFF/-32000; unit: --:--). When reaching a segment with a switch-off value, 'End' is output.

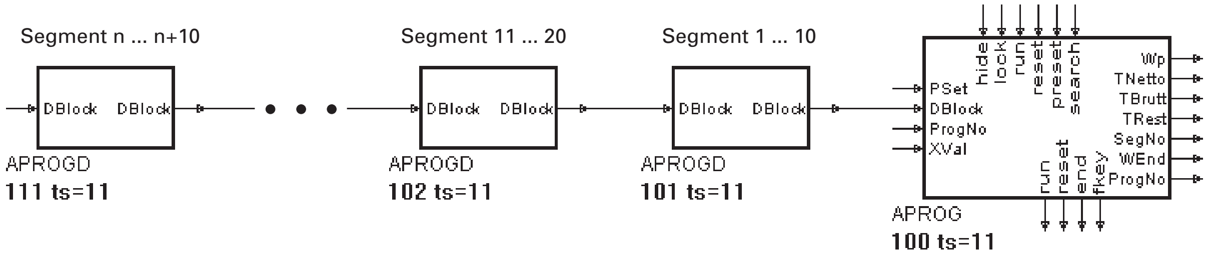
Configuration DPROG	Description		Values
PwrUp	Behaviour after mains recovery	Continue program (default)	Cont prog
		Search run in the actual segment	Cont seg
		Continue at actual time	Cont time
PEnd	Behaviour at program end	stop after program end (default)	Stop
		Reset after program end	Reset
Turbo	Time unit	Time = hours : minutes (default)	Hrs:Min
		Time = minutes : seconds	Min:Sec

Cascading

Cascading APROGD function blocks permits realization of a programmer with any number of segments. The segment sequence is dependent of the APROGD function block wiring (→ see below).

Recipes

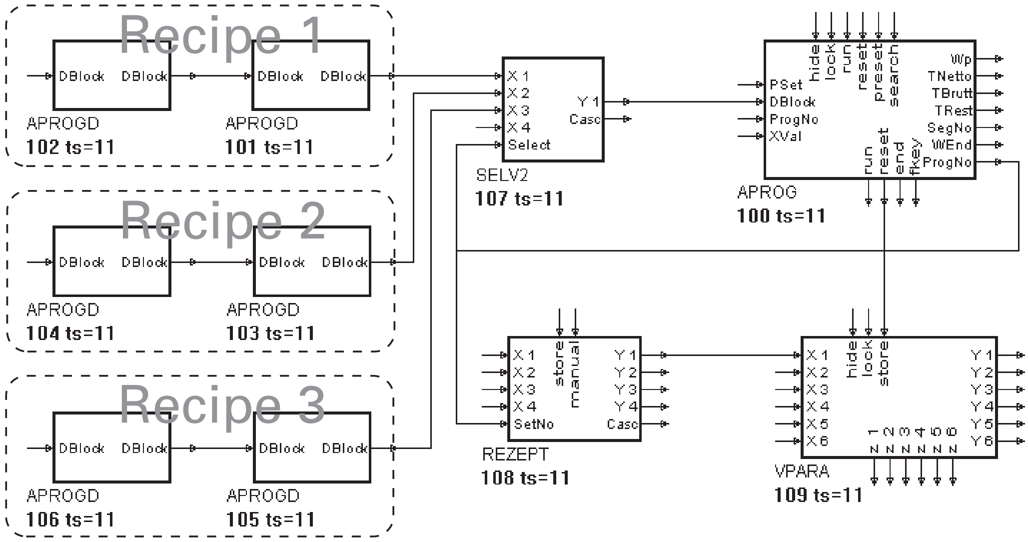
Example of an analog programmer with n segments



Analog output 'ProgNo', at which the actual recipe number is output, and one or several SELV2 function blocks can be used to select a recipe the block number of which is switched to the APROG input (→ see below). Selection of the required recipe is possible via analog input 'ProgNo' or recipe number, which can be entered via operation/interface.

Recipe switch-over (new ProgNo) is effective only after programmer reset.
Entry of the recipe number via operation/interface is possible only, if analog input ProgNo is not connected.

Example of an analog programmer with 3 recipes à 20 segments

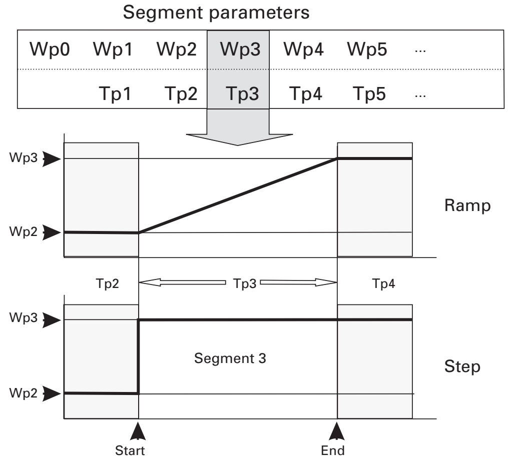


If each recipe shall have a separate reset set-point (Wp0), function blocks REZEPT and VPARA can be used as shown up. Note the calculation order (APROG → REZEPT → VPARA).

Change mode (ramp/step)

If the set-point shall change in a step or ramp is determined by a parameter (Wmode) valid for all segments of a recipe (default: ramp).
Ramp: The set-point changes linearly from the start (end value of previous segment) to the end value of the relevant segment in time Tp. For the first segment, the following gradient is applicable: (Wp1 - Wp0) / Tp1

Step: The set-point goes to value Wp immediately at segment start and maintains it during segment time Tp.



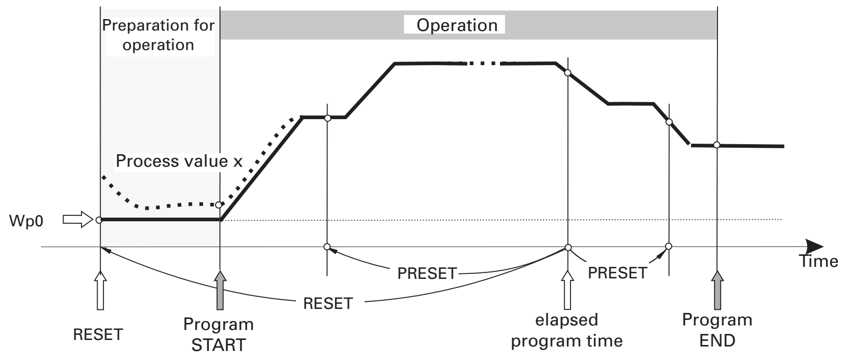
Operation preparation and end position

Each program starts at an initial position Wp0, which is used and maintained with reset or first programmer set-up. With program start from rest position, the first programmer segment runs from the instantaneous process value at the time of start command („ramp" with gradient (Wp1 - Wp0) / Tp1). With step change mode, the set-point of the first segment is activated immediately.

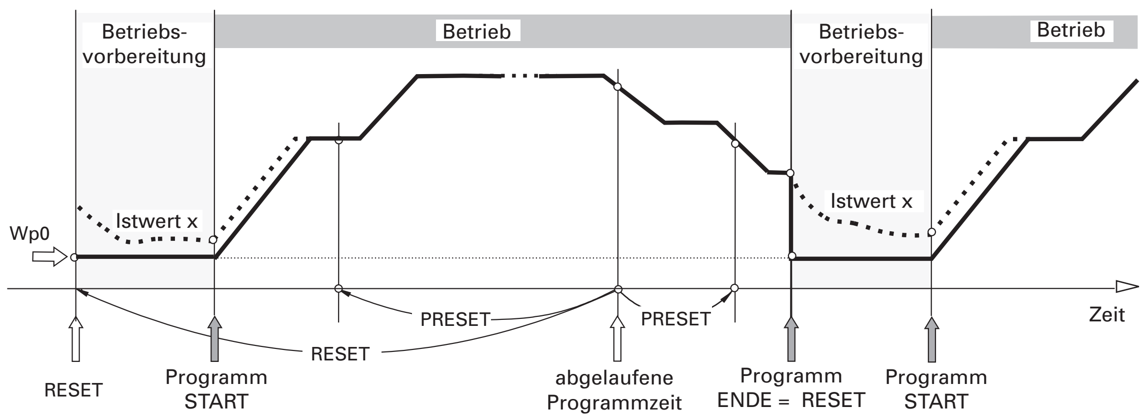
At program end, either

- the set-point of the last segment is maintained (→ see below),
- or the programmer goes to rest position Wp0 (→ see below) dependent of configuration (PEnd). The program can be started either by entry of run (control input switch-on or off or front-panel operation) or preset.

Profile with stop at end position



Profile with automatic reset at program end



Program sequence changes

During the running program, set-points and times (online) can be changed. Moreover, further segments, which did not exist so far, can be added. The actual segment number remains unchanged. Unless the actual segment is changed, the relative elapsed time also remains unchanged.

- Past changes
  A change of values and times of the past (already elapsed segments) are only effective after re-start (after previous reset).
- Future changes
  Changes of the future (segments which are not reached so far) are immediately effective. With changes of segment lines, the „rest time“ is re-calculated automatically.
- Present changes
  Changes of the actual segment time, which mean a return into the past (e.g. segment time Tp reduction to lower values than the relative time which has already elapsed in this segment) cause a branch to the start value of the next segment.
  Changes of the destination value of the actual segment cause unique re-calculation of the segment gradient for this program run, in order to reach the new destination value in the remaining time.
  Final re-calculation of the segment gradient is when starting a new batch (reset and start) or with preset to an earlier time.

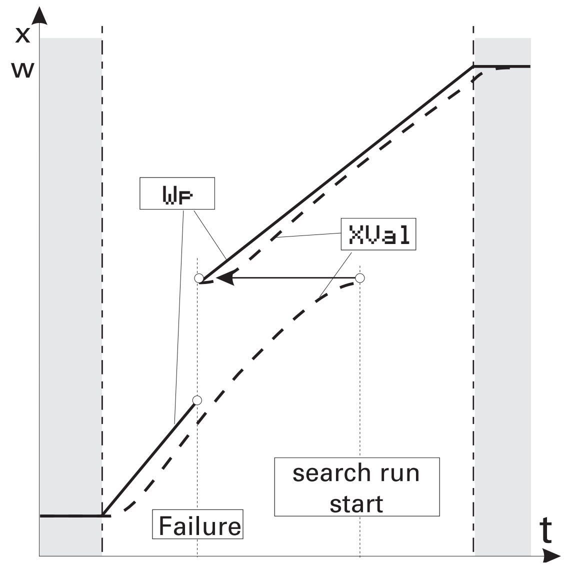
Search run

In the following cases, a search run is carried out:

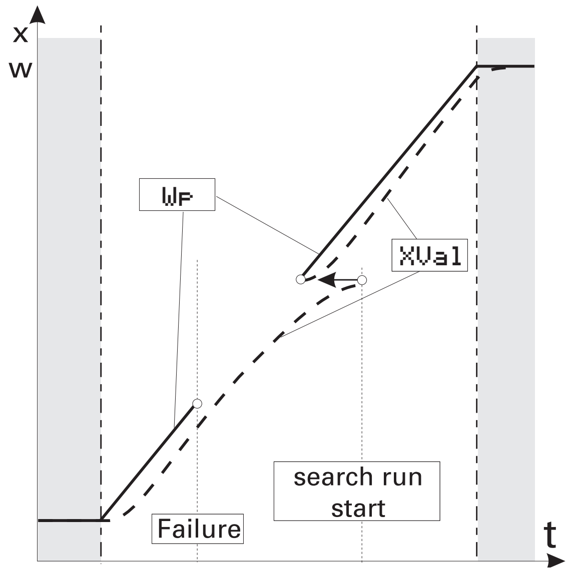
- Start via operation
- Start via interface
- Start with search = 1
- Program start after Reset
- After short power failure with PowerUp = Cont.seg. or Cont.time

When starting the search run, set-point Wp is set to the XVal value, from where it runs towards the segment end value with actual gradient (TPrio = Grad.prio) or in the actual segment rest time (TPrio = Time prio).
If the search value is out of the actual segment with TPrio = Grad.prio, the program is continued at the segment point next to the search value (actual segment start / end). With segment start value = segment end value, the program is continued at the segment start.

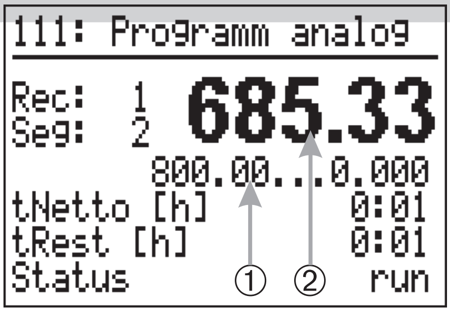
Search run with TPrio = Time Prio



Search run with TPrio = Grad.Prio



Analog programmer operating page



Analog programmer APROG has an operating page, which can be selected in the operating page menu with input '`hide`' not connected.
To change the value of an input field, mark this value by means of ▲▼ (inverse display). When acknowledging the value with ⏎, it starts blinking and can be adjusted with ▲▼. If the required value was reached, acknowledge it with ⏎.
If the FB inputs (function block inputs) allocated to the input fields in the following table are assigned to the engineering, operation (change) of this input field is not possible.

Input field		Operation	Display	FB input
Rec		If input ProgNo is wired, entering of the desired recipe number is not possible over the frontside!	indicates the actual recipe number.	ProgNo
Seg		If control input preset is wired, entering the desired segmentnumber is not possible over the frontside!	indicates the actual segment number	preset
①		No operation possible	start and end value of the actual segment	
②		No operation possible	display of actual set-point (Wp)	
tNetto		Entry of required programmer time (preset to time)	indicates the run time total (without pause)	preset
tRest		No operation possible	indicates the time until program end	
Status	stop	Stop the programmer	programmer stopped	run
	run	Start the programmer	the programmer was started	
	reset	The programmer is switched to segment 0 and'stop'	the programmer is switched to segment 0 and'stop'	reset
	quit	Leave the field without change		
	program	direct adjustment of segment parameters	segmentparameter	

Extended programmer functions

Valid from : **SIM/KS 98 version 2.1**
ET/KS 98 version 2.2

Set-point limits and decimal point (only APROG)

Parameter	Description	Range	Default
W0	Lower input limits for Wp0 ... Wpn	-29999 ... 999 999	0
W100	Upper input limits for Wp0 ... Wpn	-29999 ... 999 999	100
Dp	Number of digits behind the decimal point with input and display of segment parameters	0 ... 3	0

☞ **Note: When adjusting the segment parameters via the parameter level, these parameters are not effective!**

Direct programmer adjustment

Program set-points and segment times can be adjusted at the operating page directly via the instrument front panel, without calling up the parameter level. Direct access to parameter setting is enabled with control input p-show = „1" set at the function blocks of programmer APROG and DPROG.

Menu item Program can be selected in the status line. After confirmation, all segment parameters Tp and Wp pertaining to an effective recipe Rec can be displayed and adjusted in a scroll window (Fig.). Return to the normal operation is with End.

Scrolling is done over several data blocks (APROGD, DPROGD). „n" segment parameter (Wpn, Tn) indexing is with 3 digits. The segment parameters are distributed to the concerned data blocks automatically from left to right with ascending index (Fig.).

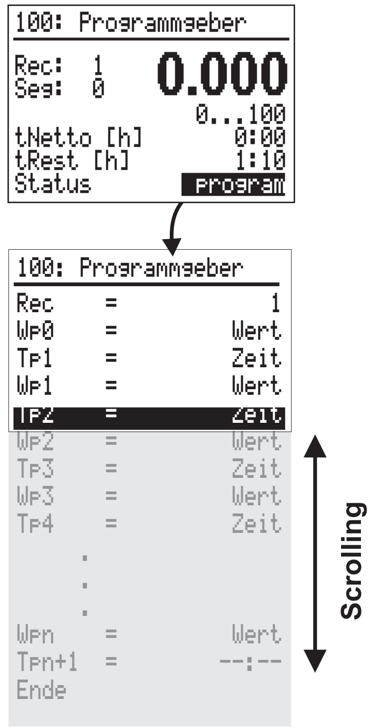
If the last segment time Tn is adjusted to a valid value, the next parameter Tn+1 is displayed automatically. = --:-- etc.

Thus an actual program can also be reduced by setting Tn+1. = --:-- in the required position. The following segments are suppressed in the program sequence.

However, the relevant segment parameters remain unchanged and are made effective again by entry of a valid value at the relevant point.

Adjusting the programmer via the parameter level remains possible. In this case, however, each data block APROGD or DPROGD must be selected separately. In this case, however, parameters W0, W100 and Dp pertaining to APROG are not effective.

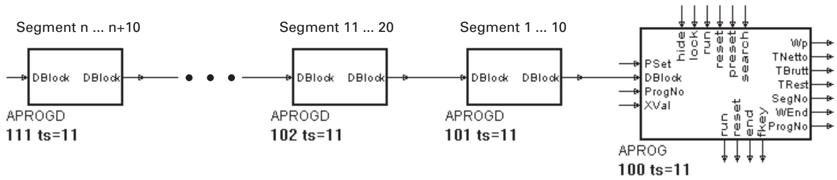
Scrolling window



Adjusting several recipes

In modes run, stop and preset, only the instantaneously effective recipe can be processed as described. Further recipes (Rec) can be changed only with the programmer at permanent reset. This is possible only with the reset input connected. For this, e.g. the automatic/manual key can be used (f-key → reset, see Fig.).

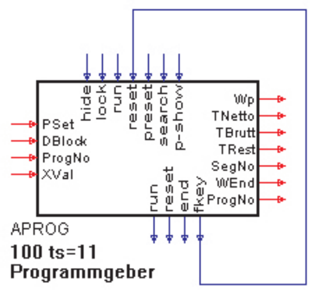
Automatic parameter distribution to data blocks



Compatibility with earlier engineerings

Earlier engineerings are converted automatically with „operating version 3" adjusted. Adjustment of new parameters W0, W100 and Dp is possible only then. For using the adjustment via the scroll window, however, control input p-show must be connected previously.

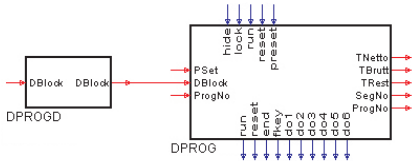
Fig.16: Perment reset

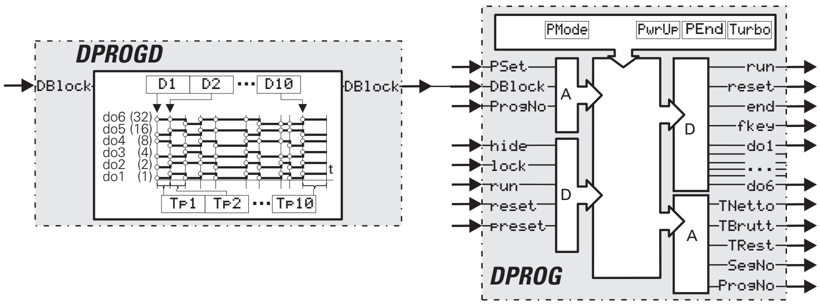


Downward compatibility

As the additional functions in earlier KS 98 firmware versions are not known, „Operating version 2" must be adjusted before transmission of the engineering. KS 98 with new firmware version (from V2.1) cannot be processed with earlier ET/KS 98 (≤ V2.1)!

23.2 DPROG (digital programmer) / DPROGD (DPROG data)



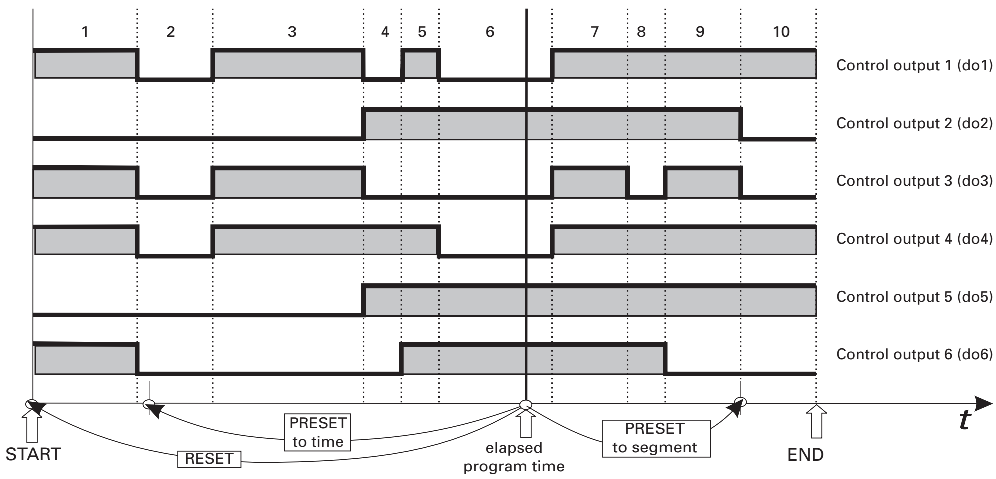


General

A digital programmer comprises a programmer (DPROG) and at least one data block (DPROD), whereby output `DBlock` of DPROGD is connected with input `DBlock` of the DPROG. Connection of several of these cascadable functions (each with 10 segments) permits realization of a programmer with any number of recipes and any number of segments. Limiting is only in the number of available block numbers and calculation time.
The data block has an analog output, at which it provides its own block number. This information is read-in by the programmer and used for segment data addressing. When an error in the segment data addresses is found, the reset value is output (status display on operating page: `Error`').

After an engineering download, `Seg 0` is output (reset). If `run` is not connected, `stop` is used.

Digital programmer definition



Inputs/outputs

Digital inputs (DPROG):		
hide	Display suppression (with hide = 1 the page is not displayed during operation).	
lock	Adjustment blocking (with lock = 1 the values are not adjustable by means of keys ▲▼).	
run	Program stop/run (0 = stop, 1 = run)	**reset has highest priority**
reset	Program continue/reset (0 = continue, 1 = reset)	
preset	Program preset (1 = preset)	

Digital outputs (DPROG):	
run	Status program stop/run (0 = program stop ; 1 = program run)
reset	Status program reset (1 = program reset)
end	Status program end (1 = program end reached)
fkey	Status A/M key / interface function `fkey' (:pressing key causes switch-over (0 or 1))
do1... do6	status of control outputs in the actual segment

Analog inputs (DPROG):	
PSet	Preset value for program
DBlock	Block number of 1st data function `DPROGD'
ProgNo	Required program number (recipe)

Analog inputs (DPROGD):	
DBlock	Block number of cascadable data function `DPROGD'

Analog outputs (DPROG):	
TNetto	Program time net (∫ T*run*)
TBrutt	Program time gross (∫ Trun + ∫ Tstop)
TRest	Programmer rest time
SegNo	Actual segment number
ProgNo	Actual program number (recipe)

Analog outputs (DPROGD):	
DBlock	Own block number

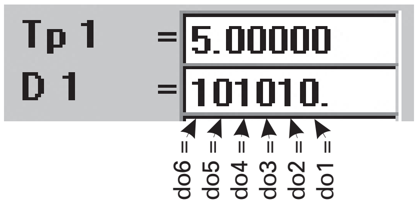
Parameter and configuration data

Parameter DPROG	Description		Range	Default
PMode	Preset Mode:	Preset to segment	Pres.time	←
		Preset to time	Pres.seg.	
D0	Status of control outputs 6...1 with reset		0 / 1 per output	000000

Parameter DPROGD	Description	Range ET	Range Unit	Default ET	Default Unit
Tp 1	**Time for segment 1 (①)**	0 ... 59 999	0:00...999:59	OFF	--:--
D 1	**Status of control output values in segment 1 (②)**	0 / 1 per output		000000	000000
Tp 2	**Time for segment 2 (①)**	0 ... 59 999	0:00...999:59	OFF	--:--
D 2	**Status of control output values in segment 2 (②)**	0 / 1 per output		000000	000000
...					
Tp 10	**Time for segment 10 (①)**	0 ... 59 999	0:00...999:59	OFF	--:--
D 10	**Status of control output values segment 10 (②)**	0 / 1 per output		000000	000000

The time for a segment is entered in the engineering tool in seconds or minutes dependent of configuration (Turbo), whereas entry into the unit is in Hrs:Min or Min:Sec. In addition to the range, a switch-off value can be entered (ET: OFF/-32000; unit: --:--). When reaching a segment with a switch-off value, 'End' is output.

With entry of control values in the engineering tool, the first digit before the decimal point corresponds to control output 1 (do1), the second digit before the decimal point corresponds to control output 2 (do2) etc. Entries behind the decimal point are interpreted as 0. Leading zeros are deleted.

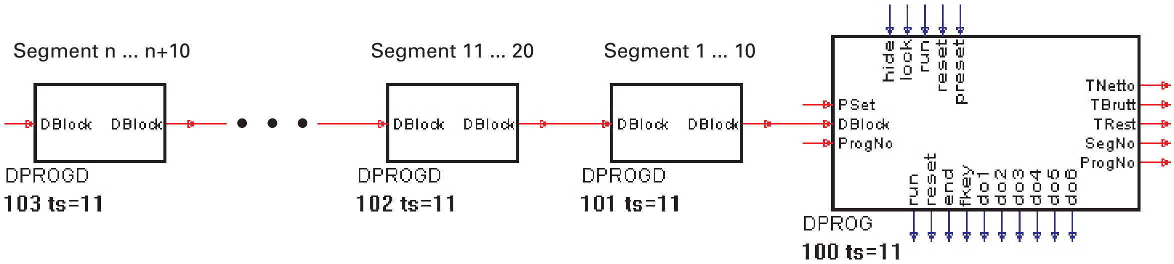


Configuration DPROG	Description		Values
PwrUp	Behaviour after mains recovery	Continue program (default)	Cont.prog
		Continue at actual time	Cont.time
PEnd	Behaviour at program end	Stop after program end (default)	Stop
		Reset after program end	Reset
Turbo	Time unit	Time = hours : minutes (default)	Hrs:Min
		Time = minutes : seconds	Min:Sec

Cascading

Cascading of DPROGD function blocks permits realization of a programmer with any number of segments. The segment sequence is dependent of DPROGD function block wiring (→ see Fig.:).

Example of a digital programmer with n segments



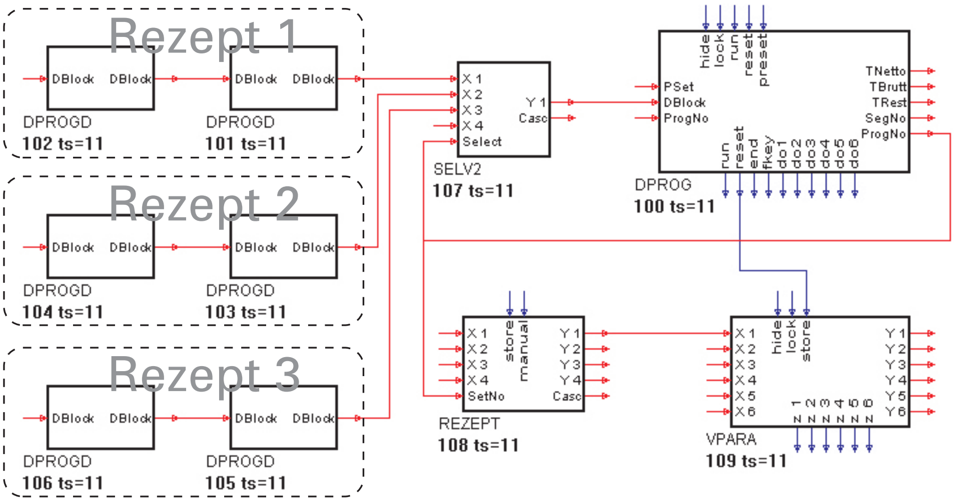
Recipes

Analog output 'ProgNo', at which the actual recipe number is output, and one or several SELV2 function blocks can be used for selecting a recipe the block number of which is switched to the DPROG input (→ see Fig.:). Selection of the required recipe is possible via analog input 'ProgNo' or recipe number, which is adjustable via operation/interface.

Recipe switch-over (new ProgNo) is effective only after programmer reset.
Entry of the recipe number via operation/interface is possible only with analog input ProgNo not connected.

Soll jedes Rezept einen eigenen Resetwert (D0) haben, können die Funktionsblöcke REZEPT und VPARA wie in Fig.: verwendet werden. Hierbei ist die Berechnungsreihenfolge (DPROG → REZEPT → VPARA) zu beachten.

Example of a digital programmer with 3 recipes each with 20 segments



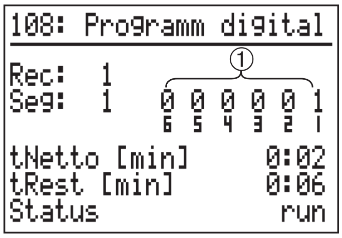
If each recipe shall have an own reset value (D0), function blocks REZEPT and VPARA can be used as shown in Fig.: . Hereby, calculation order (DPROG → REZEPT → VPARA) must be taken into account.

Program sequence changes

During the program sequence, set-points and times (online) can be changed. Moreover, further segments which did not exist so far can be added. The actual segment number remains unchanged. Unless the actual segment is changed, the relative elapsed segment time also remains unchanged.

- Past changes
  A change of values and times of the past (already elapsed segments) is only effective after restart (after previous reset).
- Future changes
  Changes of the future (segments which were not reached so far) are immediately effective. With segment time changes, **TRest** is recalculated automatically.
- Present changes
  Changes of the actual segment time which mean a return into the past (e.g. reduction of segment time **TP** to values lower than the relative time already elapsed in this segment) cause a branch to the start value of the following segment.
  When starting a new batch (reset and start) or with preset to an earlier time, the segment gradient is finally recalculated.

Digital programmer operating page



Digital programmer DPROG has an operating page, which can be selected in the operating page menu with input 'hide' not connected. For changing the value of an entry field, this value must be marked using ▲▼ (inverse display). When acknowledging the value with ⏎, it starts blinking and can be adjusted with ▲▼. When reaching the required value, it must be acknowledged with ⏎. If the FB inputs assigned to the input fields (function block inputs) in the following table are allocated to the engineering, operation (change) of this input field is not possible.

Input fields		Operation	Display	FB input
Rec		If input ProgNo is wired, entering of the desired recipe number is not possible over the frontside!	indicates the actual recipe number.	ProgNo
Seg		If control input preset is wired, entering the desired segmentnumber is not possible over the frontside!	indicates the actual segment number	preset
①		No operation possible	Status of control outputs (do1...do6) in the actual segment	
tNetto		Entry of required programmer time (preset to time)	indicates the total run time (without pause)	preset
tRest		No operation possible	indicates the time until program end.	
Status	stop	Stopping the programmer	the programmer was stopped	run
	run	Starting the programmer	the programmer was started	
	reset	The programmer is switched to segment 0 and 'stop'	the programmer is switched to segment 0 and'stop'	reset
	quit	Leaving the field without change		

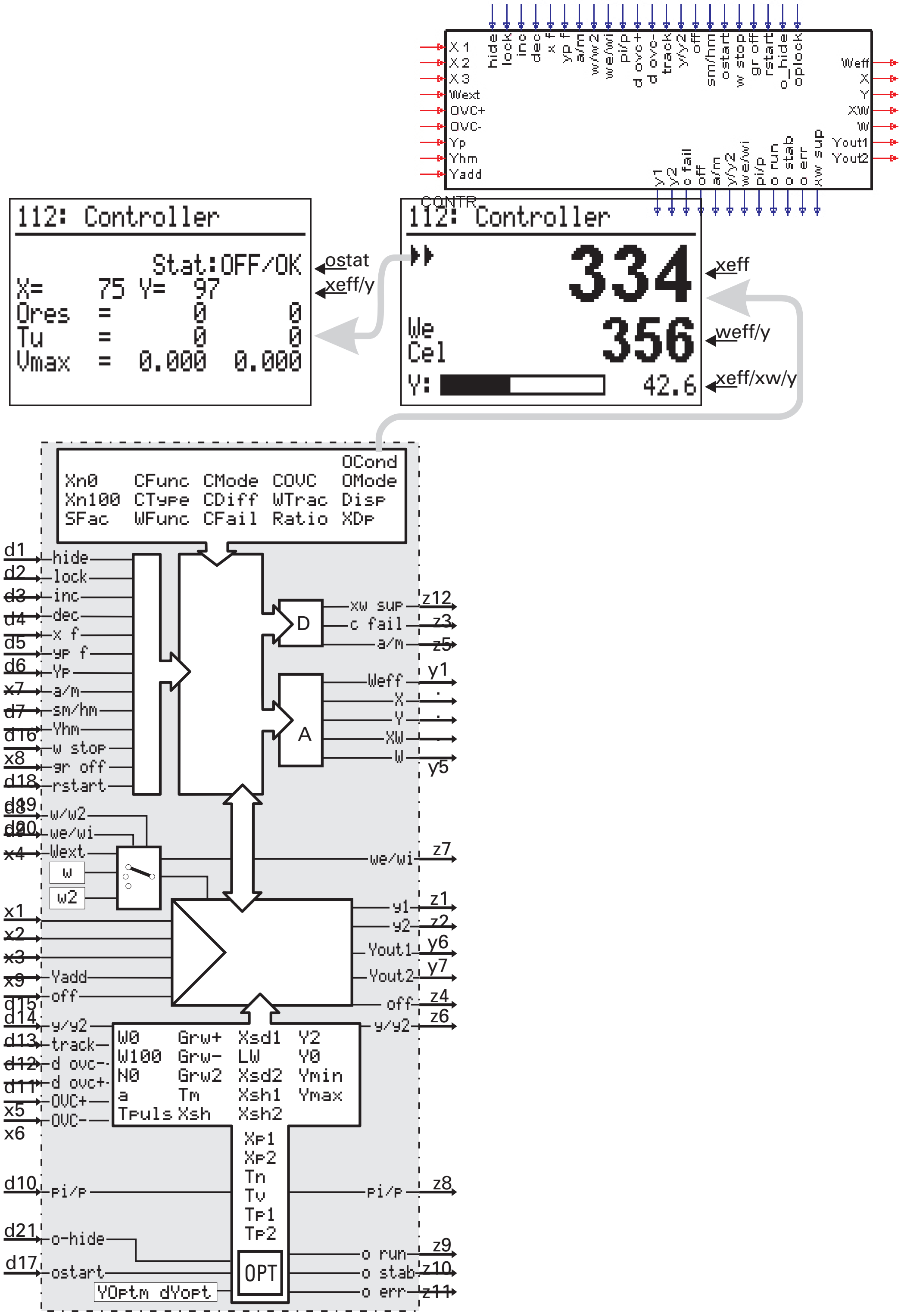
Distinction of preset (to segment 1) and reset

D0 is output with reset. With preset to segment 1, however, D1 is output, although both actions are realized at time t=0.

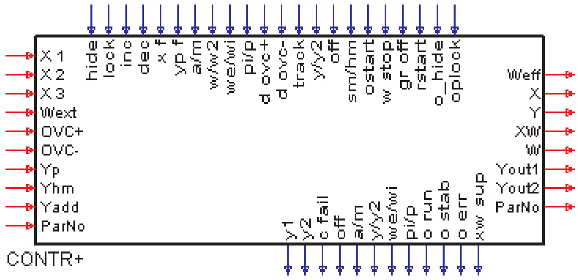
24 Controller

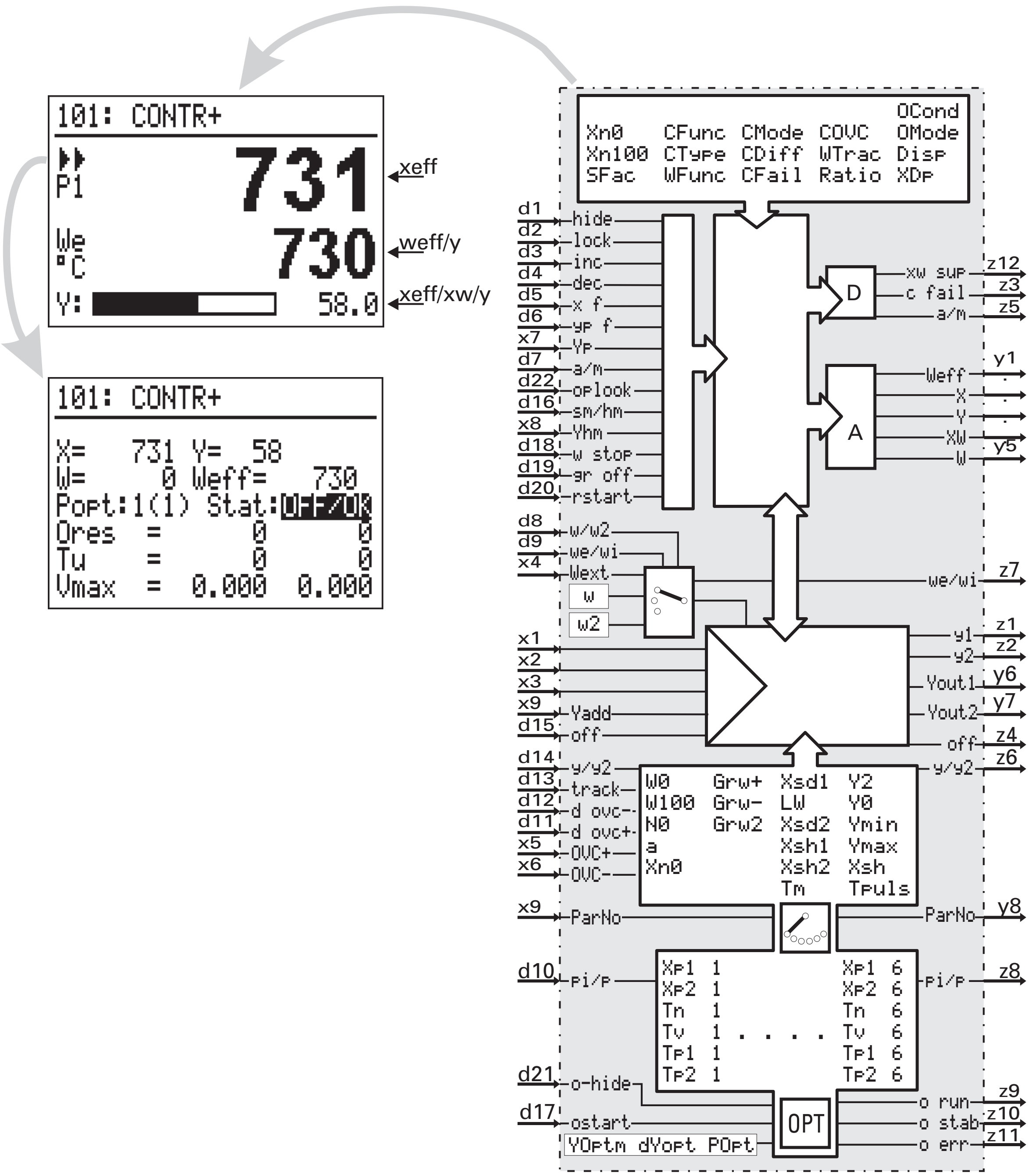
General: Function blocks CONTR and CONTR+ and PIDMA are complex control functions. Unlike CONTR, CONTR+ contains six selectable control parameter sets. PIDMA contains a special control algorithm and a different self-tuning method. A description of the functions available in the controllers starts on page 167. In case of differences in the PIDMA behaviour, a reference is made at the beginning of each section. Particularities are described at the end of each relevant section (see also PIDMA).

24.1 CONTR (Controllerfunction with one parameterset)



24.2 CONTR+ (Controllerfunction with six parametersets)

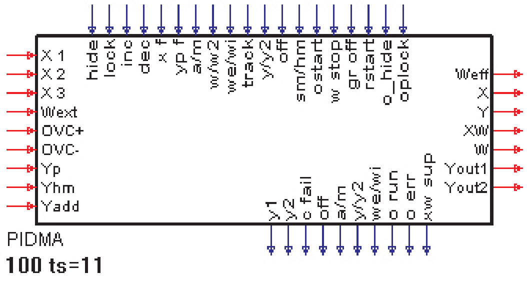


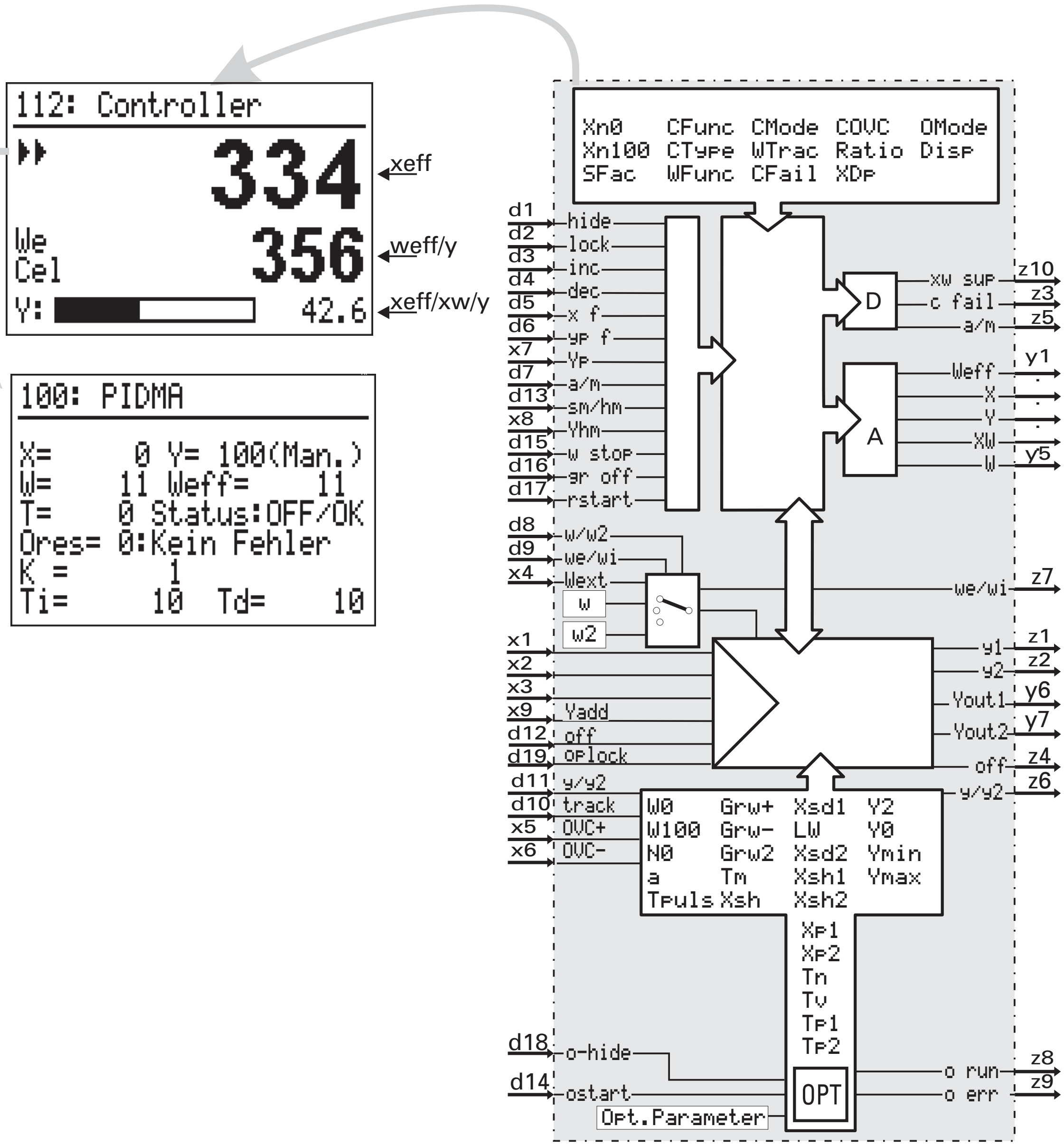


24.3 PIDMA

(Control function in parallel structure and special self-tuning method).

A special reference at the beginning of each of the following descriptions is made, if its content is only partly applicable to the PIDMA block, of if is not at all applicable. Separate descriptions of special characteristics are given at the end of the relevant section.





Inputs/outputs

Digital inputs:	
hide	Display suppression (with hide = 1 the operating page is not displayed).
lock	Adjustment locking (with lock = 1 the values are not adjustable by means of keys ▲▼).
inc	Increment for manual adjustment
dec	Decrement for manual adjustment
x f	Sensor error x1...x3
yp f	Sensor error Yp
a/m	0 = automatic 1 = manual
w/w2	0 = int./ext. set-point 1 = W2
we/wi	0 = external 1 = internal set-point
pi/p	0 = PI action; 1 = P action (→ page PI/P switch-over) (not applicable to PIDMA)
d ovc+	1 = override control + with 3-point stepping controller (→ page 201 ff) (not applicable to PIDMA)
d ovc-	1 = override control - with 3-point stepping controller(→ page 201ff) (not applicable to PIDMA)
track	0 = tracking function off; 1 = tracking function on(→ page 163, , 186)
y/y2	0 = output value Y1 = output value Y2
off	0 = controller switched on 1 = controller switched off
sm/hm	0 = soft manual 1 = hard manual
ostart	1 = self-tuning start → page 178
w stop	1 = effective set-point freeze
gr off	1 = set-point gradient suppression
rstart	1 = Start the set-point ramp → the set-point makes a step change towards the process value and goes to the adjusted process value according to GRW+ (GRW-). The rising flank (0→1) is evaluated.
o-hide	1 = self-tuning page display suppression
oplock	Blockage of key (with oplock = 1, switchover to manual by means of key is not possible).

Digital outputs:	
y1	Status of switching output Y1; 0 = off 1 = on
y2	Status of switching output Y2; 0 = off 1 = on
c fail	1 = controller in error handling
off	0 = controller switched off 1 = controller switched on
a/m	0 = automatic 1 = manual
y/y2	0 = output value Y 1 = output value Y2
we/wi	0 = external 1 = internal set-point
pi/p	feedback/integrator 0 = with 1 = without (not applicable to PIDMA)
o run	Self-tuning running
o stab	Process at rest (for self-tuning) (not applicable to PIDMA)
o err	Error during self-tuning
xw sup	Alarm suppression with set-point change

Analog inputs:	
x1	Main variable x1
x2	Auxiliary variable x2 e.g. for ratio control
x3	Auxiliary variable x3 e.g. for 3-element control
Wext	External set-point
OVC+	Override control + → page 201
OVC-	Override control - → page 201
Yp	Position feedback
Yhm	Output with hard manual
Yadd	Feed-forward control
ParNo	Only with CONTR+; required parameter set

Analog outputs:	
Weff	Effective set-point
X	Effective process value
Y	Effective output value
XW	Control deviation
W	Internal set-point
Yout1	Output value yout1 (heating)
Yout2	Correcting variable yout2 (cooling; only with continuous controller with split-range behaviour → CFunc= splitRange)
ParNo	Only with CONTR+; effective parameter set

24.4 Parameter und Konfiguration für CONTR, CONTR+ und PIDMA.

Parameter für CONTR und CONTR+

Parameter	Description	Range	Default ET	Unit
W0	Min. set-point limit (Weff)	-29999...999999	0	0
W100	Max. set-point limit (Weff)	-29999...999999	100	100
W2	Additional set-point	-29999...999999	100	100
Grw+ 3)	Set-point gradient plus unit/min	0,001...999999	Aus	----
Grw- 3)	Set-point gradient minus unit/min	0,001...999999	Aus	----
Grw2 3)	Set-point gradient for W2 unit/min	0,001...999999	Aus	----
N0	Zero offset with ratio control	-29999...999999	0	0
a	Factor a with 3-element control	-9,99...99,99	1	1
Xsh 2)	Trigger point separation (step controller)	0,2...20,0%	0,2	0,2
Tpuls	Minimum positioning step time (step controller)	0,1...2,0[s]	0,3	0,3
Tm	Actuator response time (step controller)	5...999999 [s]	30	30
Xsd1	Switching difference signaller	0,10...999999	1	1
LW	Trigger point separation additional contact (Signalgerät)	-29999...999999	Aus	----
Xsd2	Switching difference additional contact(Signalgerät)	0,10...999999	1	1
Xsh1 1)	Trigger point separation (PD) (three-point controller)	0,0...1000,0[%]	0	0
Xsh2 1)	Trigger point separation (PD) (three-point controller)	0,0...1000,0[%]	0	0
Y2	Additional output value (not with step controller)	-105,0...105,0[%]	0	0
Ymin	Min. output limiting (not with step controller)	-105,0...105,0[%]	0	0
Ymax	Max. output limiting (not with step controller)	-105,0...105,0[%]	100	100
Y0	Controller working point (not with step controller)	-105,0...105,0[%]	0	0
Y0ptm 4)	Output value with process at rest	-105,0...105,0[%]	0	0
dYopt 4)	Self-tuning step height	5...100[%]	100	100
P0pt 4)	Only with CONTR+; parameter set to be optimized	1...6	1	1
Xp1 1...6 1)	Proportional band 1	0,1...999,9[%]	100	100
Xp2 1...6 1)	Proportional band 2 (three-point and splitrange)	0,1...999,9[%]	100	100
Tn 1...6	Integral time (Tn = 0 → I-part not effektive)	0,0...999999[s]	10	10
Tv 1...6	Derivative time(Tv = 0 → D-part not effektive)	0,0...999999[s]	10	10
Tp1 1...6	Cycle time heating (three-point controller)	0,4...999,9[s]	5	5
Tp2 1...6	Cycle time cooling (three-point controller)	0,4...999,9[s]	5	5

1) %-refering to measuring range x_{n0} ... x_{n100}
2) neutral zone x_{sn} with three-point controllers is dependent to T_{puls}, T_m and x_{p1}
3) gradient control → page 192
4) self optimization → page187 ff

Parameters for PIDMA

Parameter	Description	Range	Default	Unit
PTyp	Process type (with compensation or integral)	comp. / integral	comp.	comp.
Drift	Drift compensation	switched off / switched on	off	off
CSpeed	Control dynamics	slow / normal / fast	normal	normal
W0	Min. setpoint limit (Weff)	-29999...999999	0	0
W100	Max. setpoint limit (Weff)	-29999...999999	100	100
W2	Additional setpoint	-29999...999999	100	100
Grw+ [2]	Setpoint gradient plus unit/min	0,001...999999	Aus	----
Grw- [2]	Setpoint gradient minus unit/min	0,001...999999	Aus	----
Grw2 [2]	Setpoint gradient for W2 unit/min	0,001...999999	Aus	----
N0	Zero offset with ratio control	-29999...999999	0	0
a	Factor a with 3-element control	-9,99...99,99	1	1
Xsh [1]	Trigger point separation (3-point stepping controller)	0,2...20,0%	0,2	0,2
Tpause	Min. pause time (3-point stepping controller)	0,1...999999[s]	0,1	0,1
Tpuls	Min. positioning step time (3-point stepping controller)	0,1...2,0[s]	0,3	0,3
Tm	Actuator response time (3-point stepping controller)	5...999999 [s]	30	30
thron	Switch-on threshold for OPEN and CLOSE (3-point stepping controller)	0,2...100%	0,2	0,2
throff	Switch-on threshold for OPEN and CLOSE (3-point stepping controller)	0,2...100%	0,2	0,2
Y2	Additional correcting value (not with 3-point stepping controller)	-105,0...105,0[%]	0	0
Ymin	Min. output limiting (not with 3-point stepping controller)	-105,0...105,0[%]	0	0
Ymax	Max. output limiting (not with 3-point stepping controller)	-105,0...105,0[%]	100	100
Y0	Controller working point (not with 3-point stepping controller)	-105,0...105,0[%]	0	0
dYopt [3]	Self-tuning step height	5...100[%]	100	100
Xlimit	Switch-off point for correcting variable step (process value change)	0,5...999999	1	1
Tdrift	Time window for drift determination (process value)	0...999999	30	30
Tnoise	Time window for noise determination (process value)	0...999999	30	30
Kp	Control amplification	0,1...999,9[%]	100	100
Tn 1	Integral time (Tn = 0 → I action is not effective)	0,0...999999[s]	10	10
Tv 1	Derivative time (Tv = 0 → D action is not effective)	0,0...999999[s]	10	10
Tp1 1	Cycle time heating (3-point controller)	0,4...999,9[s]	5	5
Tp2 1	Cycle time cooling (3-point controller)	0,4...999,9[s]	5	5
VD	Derivative gain (Td/T1)	1...999999	4	4
bW_p	Setpoint weighting factor proportional action	0...1	1	1
cW_d	Setpoint weighting factor derivative action	0...1	0	1
Tsat	Time constant for integral action in Y limiting (anti-reset wind-up)	1...999999	50	50
Xsh	Neutral zone for integral action	1...999999	0	0

[1] Neutral zone x_{sn} with 3-point stepping controllers is dependent on T_{puls}, T_m and x_{p1} (→ V. Hints for self-tuning).
[2] for gradient control → page 192
[3] for self-tuning → page 187 ff

Konfigurationsdaten CONTR, CONTR+ und PIDMA

Configuration	Description		Values	Default
CFunc	Control behaviour:	Signaller 1 output	Signal 1	
		Signaller 2 outputs	Signal 2	
		2-point controller	2-point	
		3-point controller (heating/cooling switching)	3-point	
		3-point controller (heat.continuous/cool.switching)	Cont/swi	
		3-point controller (heat.switching/cool.continuous)	Swi/Cont	
		triangle-star-off (d/Y-off)	2P-DSO	
		3-point stepping controller	Stepping	
		3-point-stepping controller with pos. feedback Yp	Step+Yp	
		Continuous controller	Cont	←
		Continuous controller with split-range operation	splitRang	
		Continuous controller with position feedback Yp	Cont Yp	
CType	Controller type	Standard controller → page	Standard	←
		Ratio controller → page 187	Ratio	
		3-element controller → page 198	3-elem	
WFunc	Set-point function	Set-point control → page 183	Set-point	←
		Set-point/cascade control → page 183	Sp/casc	
CMode	Output action	Inverse output action	Inverse	←
		Direct output action	Direct	
CDiff	Differentiation	Xw differentiation	Xw	←
		X differentiation	X	
CFail	Behaviour with sensor error	Neutral	Neutral	
		Ypid = Ymin (0%)	Ymin	←
		Ypid = Ymax (100%)	Ymax	
		Ypid = Y2 (no adjustment via front panel)	Y2	
		Ypid = Y2 (automatic) or Yman (manual operation)	Y2/Yman	
COVC	Output limiting	No override control	Off	←
		Override control +	OVC+	
		Override control -	OVC-	
		Override control + / -	OVC+/OVC-	
WTrac	Int. set-point tracking	No tracking of Wint	Off	←
		Set-point tracking	SP-track	
		Process value tracking	PV-track	
Ratio	Ratio controller function:	(x1 + N0) / x2 → page 1887	Type 1	←
		(x1 + N0) / (x1 + x2) → page 188	Type 2	
		(x2 -x1 + N0) / x2 → page 188	Type 3	
XDp	Digits behind the decimal point (process value)		0...3	0
Disp	Contents of bargraph line:	Output variable	Y	←
		Control deviation	XW	
		Xeff	Xeff	
OMode	Self-tuning mode:		Standard	←
OCond	Condition for process at rest:	grad = 0	grad=0	←
		grad <0 (controller inverse) grad >0 (controller direct)	grad<0/>0	
		grad <>0	grad<>0	
Xn0	Span start		-29999 ... 999 999	0
Xn100	Span end		-29999 ... 999 999	100
SFac	Factor stoichiom. ratio		0,01 ... 99,99	1,00

Fields shown with gray background are not available with the PIDMA block.

24.5 Small controller ABC

Some operating principles, which are realized in the controller (✔) or which are possible by means of an additional engineering are explained in the following section (✎). Cross references are shown in italics.

✓ *Anti-reset-wind-up*

Measure which prevents the controller integrator from saturation.

✓ *Working point (Y0)*

The working point of a P or a PD controller indicates the value output to the process with process value = set-point. Although this value is only important for P and PD controller, it can also be of interest for controllers with integrator (automatic working point).

✓ *Automatic operation*

Normal controller operation. The controller controls the process by means of the adjusted control parameters. Automatic operation is effective with a/m (di7) set to 0 (automatic) AND automatic selected via front-panel key AND sm/hm (di16) set to 0 (soft manual). Contrary: manual operation.

✓ *Cutback*

Reset of the integral action shortly before reaching the end setpoint with setpoint ramps.

✓ *Cycle time*

The duration of a switching cycle (pulse and pause) at 50 % power control of a 2-point controller.

✓ *Line-out to the target*

By early setpoint switch-over to the ramp end setpoint, the controller
is given a new target orientation for smooth line-out to the target.

✎ ***Bandwidth control***

With program control or gradient control, there may be a considerable control deviation if the process is slow. This can be prevented by monitoring the control deviation for an adjusted tolerance band by means of additional function blocks. With out-of-tolerance, the set-point change is stopped (w stop with controller or stop with program controller).

✓ *Three-element control*

Particularly suitable for processes in which load changes would be detected too late (e.g. level control for steam boilers). In this case, a disturbance variable is used at which the mass balance (steam removal, feed water) is evaluated, subtracted and added to the control variable (after differentiation, if necessary).

✎ ***Feed-forward control***

Especially suitable for processes with long delay time, e.g. pH-control. A disturbance variable is used, at which the evaluated, differentiated or delayed value of an analog input (YAdd) is added directly to the controller output for avoiding the controller time behaviour.

✓ *Gradient control*

Particularly suitable for processes in which energy shocks or quick set-point changes must be avoided. Set-point changes are bumpless in both directions, since the effective set-point always runs towards the changed set-point (destination set-point) by means of gradients Grw+ or Grw-. For the second set-point w2, gradient Grw2 acts in both directions, also with w → w2 switch-over.

✓ *Manual operation*

When switching over to manual operation, the automatic sequence in the control loop is interrupted. Modes *soft manual* and *hard manual* are available. Switch-over automatic → manual and vice versa are bumpless. Manual operation is effective with a/m (di7) set to 1 (manual) OR manual selected via front-panel key OR sm/hm (di16) set to 1 (hard manual). Contrary: *automatic*.

ⓘ If automatic remains selected via key, the controller changes to automatic after omission of the di7 signal. With manual selected additionally via key, the controller remains in manual mode after omission of the di7 signal.

✓ *Hard manual (sm/hm)*

Safety output value Yhm. The controller output goes to the preset value immediately, when hard manual is active (the controller is switched to *manual mode* directly). Keys ▲ / ▼ are without effect. Switch-over to *automatic mode* is bumpless.

✎ ***Cascade control***

Particularly suitable for temperature control in e.g. steam boilers. A continuous master controller (load controller) provides its output signal as an external set-point to a slave controller, which varies the output value.

✓ *Override control (OVC) → page 201*

Limiting of the min. (OVC-) or max. (OVC+) output value to the value of an analog input. Limitation by override control can be used e.g. with control continued by a different controller dependent of different conditions when reaching defined process statuses. The transitions from unlimited → limited output value and vice versa are bumpless.

✎ ***Program control***

The effective set-point follows the profile of a programmer (APROG with APROGD) connected to input Wext; the controller must be set to we (di9=0).

✓ *Process at rest*

For a clear optimization attempt during *self-tuning*, the control variable must be in a stable position. Various rest conditions can be selected:

Process behaviour with constant output value	Recommended setting	Stability PIR_H is reached, if
A constant process value is reached in relatively short time (standard process).	grad=0	**the process value is constant during 1 minute.**
After a relatively long time, a constant process value is reached (slow process).	grad<0/>0	**the process value decreases constantly during 1 minute (controller inverse) or increases constantly during 1 minute (controller direct).**
The process is affected from outside.	grad<>0	**the process change is constant during 1 minute. The output action is unimportant.**

✓ *Ramp function*

Set-point changes in ramps rather than in steps. See gradient control.

✓ *Control parameters*

For controller optimization, the controller must be matched to the process characteristics (→ 5 Optimizing help, → 6 Self-tuning). The effective parameters are Xp1, Tn, Tv and Y0.
Dependent of controller operating principle, the following additional parameters are possible:
Tp1 (with *2-point/3-point controllers*), Xp2 and Tp2 (with *3-point controllers*),
Xsh and Tpuls and Tm (with *3-point stepping controllers*).

✓ *Control behaviour*

Generally, fast line-out to the set-point without overshoot is required. Dependent of process, various control behaviours are desirable for this process:

- easily controllable processes (k < 10%) can be controlled with PD controllers,
- processes with medium controllability (k 10...22%) using PID controllers and
- badly controllable processes (k > 22%) with PI controllers.

✎ ***Controller OFF (off)***

With input off =1, there are no pulses at the switching output and the continuous outputs are 0%.

✓ *Self-tuning*

For optimum process control, the controller must be matched to the process requirements. The time required for this purpose can be reduced considerably by self-tuning (→ 6 Self-tuning). During self-tuning, the controller makes one adaptation attempt during which the *control parameters* are determined automatically from the process characteristics for fast line-out to the set-point without overshoot.

✓ *Soft-Manual*

Usual *manual operation*: with *automatic* ⟶ *manual* change-over, the last output value remains active and can be adjusted via keys ▲ / ▼. Transitions *automatic* ⟶ *manual* and vice versa are bumpless.

✓ *Set-point switch-over*

In principle, the following set-points are possible: internal set-point wi , second internal set-point w2 and external set-point we. With program control, external set-point we must be selected. The analog set-point comes from APROG and is applied to input Wext.

✎ ***y feedback control***

Particularly suitable for processes in which load changes lead to process value drops. A load-dependent change to set-point (preferable) or process value is made. The evaluated and filtered output value is added to the set-point in a separate function block. Use the Wext input and set the controller to we.

✎ ***PI/P switch-over***

When optimizing slow processes, e.g. big furnaces, the controller I action can cause problems: if starting up was optimized, line-out can be slow; with optimization of the disturbance behaviour, there may be an important overshoot. This effect is prevented by switching off the I action during start-up or with high control deviations (e.g. by applying a limit contact to the control deviation) and switching it on again only when the process value approaches the set-point. To prevent permanent control deviations, the limit contact must be further away from the set-point than the permanent control deviations.

✓ *Tracking*

During switch-over from external or program set-point to internal set-point, set-point or output value step changes may occur. By means of the tracking function, the transition is bumpless. Process value tracking: During switch-over, the effective process value is used as internal set-point. Set-point tracking: During switch-over, the external or program set-point used so far is taken over as internal set-point.

✓ *Behaviour with fail (configuration of the controller behaviour with sensor failure, xf)*

Selected behaviour	Effect with 3-point stepping controllers	Effect with other controllers
Neutral	No output pulses	No output pulses or 0%
Ymin	Actuator is closed	Ymin (≙ limiting)
Ymax	Actuator is opened	Ymax (≙ limiting)
Y2	Not selectable	Y2 fixed, also with manual operation
Y2/Yman	Not selectable	Y2, adjustable in manual mode with ▲▼

✓ *Ratio control*

Particularly suitable for controlling mixtures, e.g. fuel-air mixture for ideal or stoichiometric combustion. For taking e.g. the atomizer air into account, zero offset N0 can be added.

✓ *x/xw differentiation*

Dynamic changes of process value or set-point affect control differently. x-differentiation: Process value changes (disturbances) are used dynamically to permit better control results. xw-differentiation: Changes of process value (disturbances) and set-point are used dynamically to permit a better control result. In this case, the improvement is dependent of both disturbance and control behaviour.

✓ *Controller operating principle*

The static operating principles for controllers with P or PD behaviour with adjustable *working point* Y0 are shown. On controllers with I action, the *working point* is shifted automatically. The outputs (⊖→) are described with **h** („heating“), **c** („cooling“), („open“) and („close“).

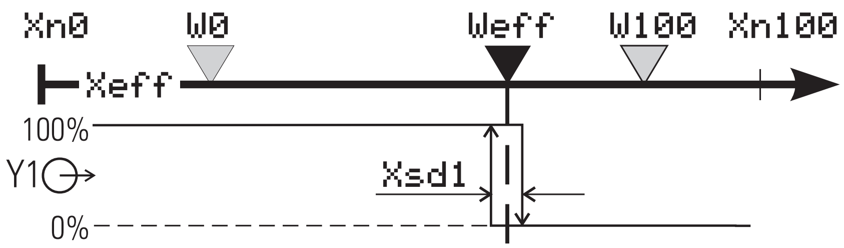
24.6 Controller behaviour

Signaller, 1 output (not available with PIDMA):

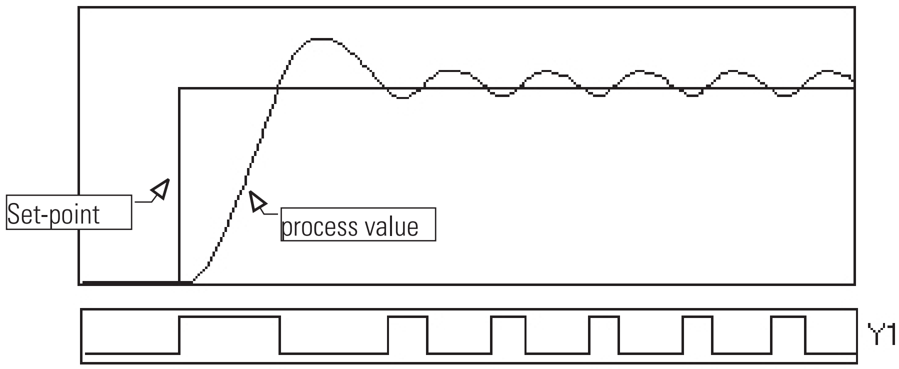
The signaller is suitable for processes with small T_u and low v_{max} .
The advantage is in the low switching frequency. Switch-on is always at a fixed value below the set-point, switch-off is always at a fixed value above the set-point.
The control variable oscillation band is determined as a result of :

$$X_0 = x_{max} \bullet \frac{T_u}{T_g} + X_{Sd} = v_{max} \bullet T_u + X_{Sd}$$

The signal function corresponds to limit signalling, whereby the set-point is the limit value. The trigger point is symmetrical to the set-point; hysteresis X_{sd1} is adjustable.



Static operating principle of the signalling function of a signaller, 1 output



Configuration	Effective controller parameters of a signaller with one output			
CFunc = signaller, 1 output	W0	1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100	1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2	1)	Additional set-point	-29 999 ...999 999
	Grw+	2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw-	2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2	2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0		Zero offset (only effective with CType=ratio controller)	-29 999 ... 999 999
	a		Factor a (only effective with CType=3-element control)	-9,99 ... 99,99
	Xsd1	1)	Signaller switching difference	0,1 … 999 999
	Titel		Title of controller page (only display)	16 characters
	Einh.X		Process value unit (only display)	6 Zeichen
	W int		Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999

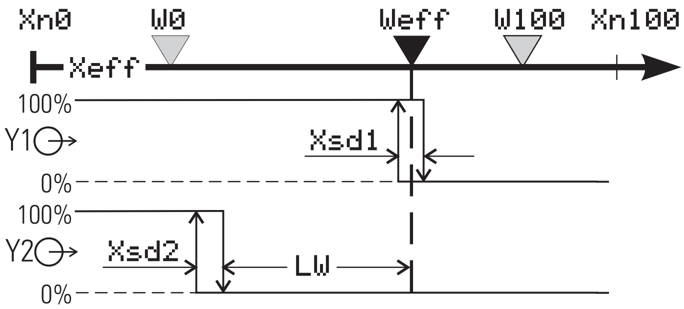
1) The values are specified in the process value unit - e.g. [°C, °F, bar, %, etc.]
2) The rate of change must be specified in units /minute (e.g. °C/min).
→ see gradient control page 192.

Signaller, 2 outputs (not available with PIDMA):

The signaller is suitable for processes with small T_u and low v_{max} .
The advantage is in the low switching frequency. Switch-on is always at a fixed value below the set-point, whereas switch-off is always at a fixed point above the set-point. The control variable oscillation band is determined as a result of :

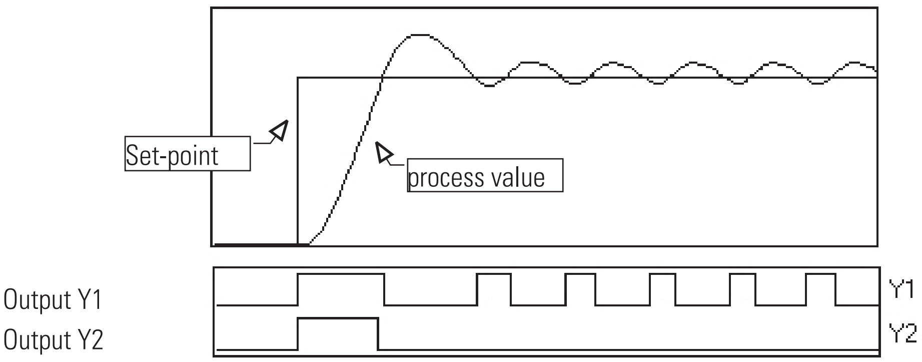
$$X_0 = x_{max} \bullet \frac{T_u}{T_g} + X_{Sd} = v_{max} \bullet T_u + X_{Sd}$$

The signalling function provides alarm signalling, whereby the set-point is the limit value. The trigger point is symmetrical to the set-point; hysteresis X_{sd1} is adjustable.
The signaller with two outputs has an additional "limit contact". Its difference from the set-point is adjustable in parameter LW (including polarity sign).



Static operating principle of the signalling function Signaller, 2 outputs

LW is shown as a negative value in the example (e.g. -20)



Configuration	Effective controller parameters of a signaller with two outputs			
CFunc = Signaller, 2 outputs	W0	1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100	1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2	1)	Additional set-point	-29 999 ...999 999
	Grw+	2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw-	2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2	2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0		Zero offset (only effective with CType=ratio controller)	-29 999 ... 999 999
	a		Factor a (only effective with CType=3element control)	-9,99 ... 99,99
	Xsd1	1)	Signaller switching difference	0,1 … 999999
	LW	1)	Trigger point separation of additional contact OFF ≙ the additional contact is switched off	-29 999 … 999 999 -32 000 = AUS
	Xsd2	1)	Switching difference of additional contact	0,1 … 999 999
	Titel		Controller page title (only display)	16 characters
	Einh.X		Unit of the process value (only display)	6 characters
	W int		Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999

1) The values are specified in the process value unit - e.g. [°C, °F, bar, %, etc.]
2) The rate of change must be specified in units / minute (e.g. °C/min).
→ see gradient control page 192.

Two-point controller

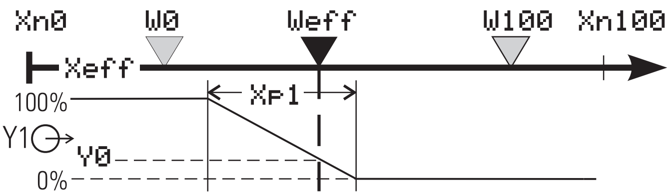
Switching controller with two switching statuses:

1. Heating switched on; → output Y1 = 1
2. Heating switched off; → output Y1 = 0

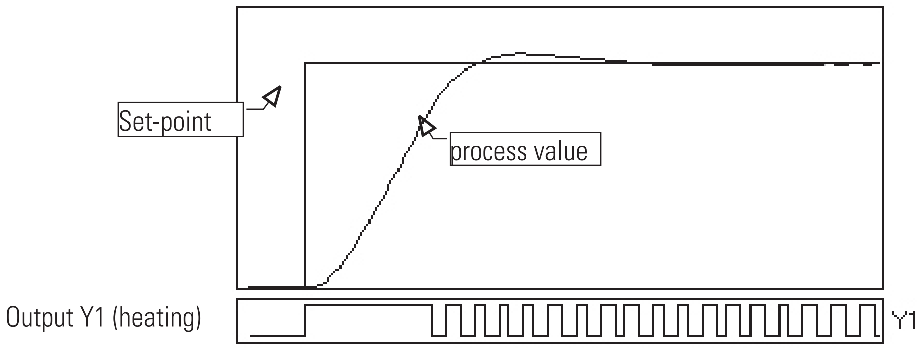
E.g. for temperature control with electrical heating (inverse operation) or cooling (direct operation)

Adjust cycle time Tp1 as follows: Tp1<= 0,25• Tu

With higher Tp1, oscillations must be expected. Tp1 corresponds to the minimum cycle time (time in seconds) at 50 % duty cycle.



Static operating principle of a two-point controller



PD action (**Tn** = 0 ≙ switched off Tn = ∞)
The working point is in the middle of proportional band X_{p1} at 50 % duty cycle. In order to keep the control variable constant, a defined quantity of energy dependent of set-point is required. This energy causes a permanent control deviation, which increases with growing X_{p1} .

DPID action
By means of the I action, line-out is without permanent control deviation.
The static characteristic of a two-point controller is identical to the one of a continuous controller, with the difference that a duty cycle instead of a linearly variable current signal is output (relay contact, logic signal 0/20mA or control output 0/24V).
Working point Y_0 and cycle time Tp1 at 50% are adjustable.
The shortest switch-on or switch-off time is 100ms.

Configuration	Effective controller parameters of a two-point controller			
CFunc = 2-Punkt	Popt		Parameter set for self-tuning (only with CONTR+)	1...6
	W0	1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100	1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2	1)	Additional set-point	-29 999 ...999 999
	Grw+	2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw-	2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2	2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0		Zero offset (only effective with CType=ratio controller)	-29 999 ...999 999
	a		Factor a (only effective with CType=3-element control)	-9,99 ... 99,99
	Y2		Additional correcting variable	0 … 100 [%]
	Ymin		Min. correcting variable limiting	0 … 100 [%]
	Ymax		Max. correcting variable limiting	0 … 100 [%]
	Y0		Correcting variable working point (start-up correcting variable)	0...100 [%]
	YOptm		Correcting variable during process at rest (not with PIDMA)	0...100 [%]
	dYopt		Self-tuning step change height	5...100 [%]
	Xp1(1...6)	3) 4)	Proportional band 1	0,1 … 999,9 [%]
	Tn1(1...6)	4)	Integral action time	0 … 999 999 [s]
	Tv1(1...6)	4)	Derivative action time	0 … 999 999 [s]
	Tp1(1...6)	4)	Cycle time heating	0,4 … 999,9 [s]
	Titel		Title of controller page (only display)	16 characters
	Einh.X		Unit of the process value (only display)	6 characters
	W int	1)	Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999
	A/H		Controller status after transmission of the engineering to KS98	0 or 1

1) The values are specified in the process value units - e.g. [°C, °F, bar, %, etc.]
2) Specifiy the rate of change in units / minute (e.g. °C/min).
→ see gradient control page 192.
3) % specifications are related to measuring range Xn100 - Xn0 .
They are not related to values W0 and W100.
4) (1 ...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Additional controller parameters for PIDMA

(for description, see section Controller characteristics)

Parameter	Beschreibung	Wertebereich
Kp	Control amplification (replaces Xp1/Xp2 of CONTR)	0,1...999,9[%]
VD	Derivative gain (Td/DT1)	1...999999
bW_p	Setpoint weighting factor proportional action	0...1
cW_d	Setpoint weighting factor derivative action	0...1
Tsat	Time constant for integral action in Y limiting (anti- reset wind-up)	1...999999
dYopt	Step width with self-tuning using different range	-100 … 100
xsh	Neutral zone in which the integral action is held	0 … 999999

Three-point controller

Switching controller with three switching statuses:
1. Heating switched on; → output Y1 = 1, Y2 = 0
2. Heating and cooling switched off; → outputs Y1 = 0, Y2 = 0
3. Cooling switched on; → outputs Y1 = 0, Y2 = 1

E.g. for temperature control with electrical heating (h) and cooling (c).
Adjust cycle time **Tp1** and **Tp2** as follows:

Tp1<= 0,25• Tu (h) Tp2<= 0,25• Tu (c).

With higher **Tp1**/**Tp2**, oscillations have to be expected. Cycle times **Tp1** and **Tp2** are the minimum cycle times at 50% duty cycle.

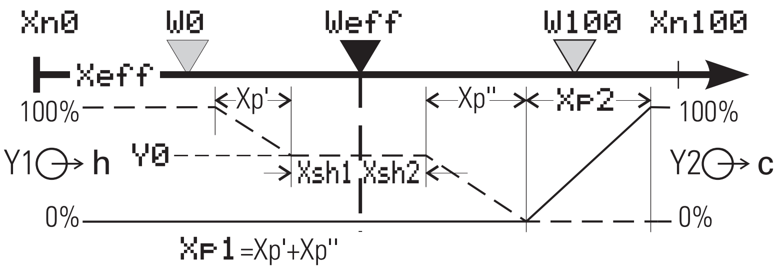
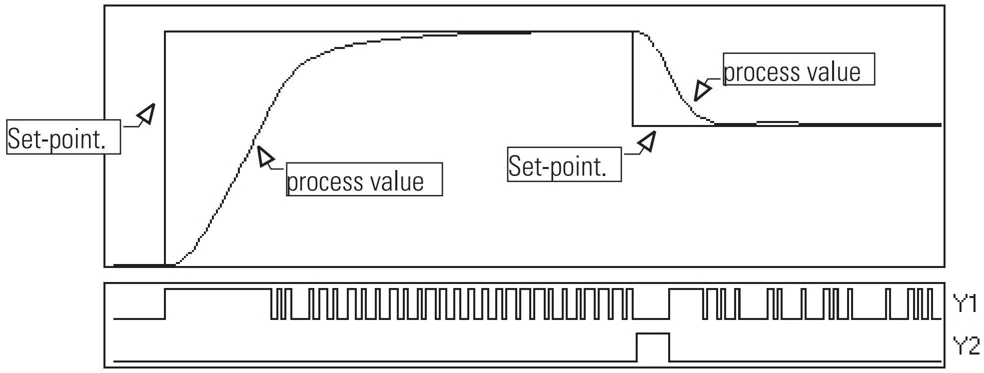


Fig.: 1
Static operating principle of a three-point controller



Output Y1 (heating)
Output Y2 (cooling)

PD/PD action (**Tn** = 0 ≙ switched off Tn = ∞)
The positioning range reaches from 100 % heating (Y1) to 100 % cooling (Y2).
The proportional bands must be adapted to the various heating and cooling power values. In order to keep the control variable constant, a defined amount of energy dependent of set-point is required. This causes a permanent control deviation, which increases with growing $X_{p(1,2).}$

DPID/DPID action
By means of the I action, line-out without permanent control deviation is possible.
Transition from trigger point 1 (heating) to trigger point 2 (cooling) is without neutral zone. The proportional bands must be adapted to the various heating and cooling power values.

Fig.: 1 shows the static characteristic for inverse output action.
Direct/inverse switchover only causes exchanging of the outputs for "heating/cooling".
Expressions "heating" and "cooling" may also mean similar processes (dosing acid/lye, ...).
The neutral zone is adjustable separately for the trigger points (X_{sh1}, X_{sh2}) i.e. it need not be symmetrical to the set-point.

The type of positioning signals is selectable:

CFunc = 3-point heating switching, cooling switching
CFunc = cont/switchheating continuous, cooling switching
CFunc = switch/contheating switching, cooling continuous

Combination "heating continuous" and "cooling continuous" is covered by "splitRange - continuous controller with split-range behaviour". → see also "continuous controller" page: 174.

Configuration	Effective controller parameters with two-point controller		
CFunc = 2-Punkt	Popt	Parameter set for self-tun ing (only with CONTR+)	1...6
	W0 1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100 1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2 1)	Additional set-point	-29 999 ...999 999
	Grw+ 2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw- 2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2 2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0	Zero offset (effective only with CType=ratio controller)	-29 999 ...999 999
	a	Factor a (effective only with CType=3-element control)	-9,99 ... 99,99
	Xsh1 3)	Neutral zone (Xw > 0)	0,0 ... 1000 [%]
	Xsh2 3)	Neutral zone (Xw < 0)	0,0 ... 1000 [%]
	Y2	Additional positioning value	0 … 100 [%]
	Ymin	Min. output limiting	0 … 100 [%]
	Ymax	Max. output limiting	0 … 100 [%]
	Y0	Correcting variable working point (start-up corr. variable)	0...100 [%]
	YOptm	Positioning value during process at rest	0...100 [%]
	dYopt	Step height during self-tuning	5...100 [%]
	Xp1(1...6) 3) 5)	Proportional band 1	0,1 … 999,9 [%]
	Xp2(1...6) 3) 5)	Proportional band 2	0,1 … 999,9 [%]
	Tn1(1...6) 5)	Integral action time	0 … 999 999 [s]
	Tv1(1...6) 5)	Derivative action time	0 … 999 999 [s]
	Tp1(1...6) 5)	Cycle time heating	0,4 … 999,9 [s]
	Tp2(1...6) 5)	Cycle time heating	0,4 … 999,9 [s]
	Titel	Title of controller page (only display)	16 characters
	Einh.X	Unit of process value (only display)	6 characters
	W int 1)	Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999
	A/H	Status of controller after transmission of the engineering to KS98	0 or 1

1) The values are specified in the process value unit - e.g. [°C, °F, bar, %, etc.]
2) The rate of change must be specified in units/minute (e.g. °C/min).
→ see gradient control page 192.
3) % specifications are related to measuring range Xn100 - Xn0.
There is no relation to values W0 and W100.
4) As default, value Ymin is set to 0. In this case, output Y1 cannot switch!
5) (1 ...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Additional controller parameters for PIDMA (for description, see section Controller characteristics)

Parameter	Description	Value range
Kp	Control amplification (replaces Xp1/Xp2 of CONTR)	0,1...999,9[%]
VD	Derivative gain (Td/DT1)	1...999999
bW_p	Setpoint weighting factor proportional action	0...1
cW_d	Setpoint weighting factor derivative action	0...1
Tsat	Time constant for integral action in Y limiting (anti- reset wind-up)	1...999999
xsh	Neutral zone in which the integral action is held	0...999999
dYopt	Step width with self-tuning using different range	-100 … 100

D / Y / off (not available for PIDMA

The principle is identical to the control behaviour of a 2-point controller with additional contact.

Output Y2 is used for switchover of the connected circuit between "Δ" and "Y". Output Y1 switches the heating energy on and off.
E.g. for temperature control with electrical heating (inverse operation) or cooling (direct operation).
Cycle time Tp1 must be adjusted as follows: Tp1<= 0,25•Tu With higher Tp1, oscillations must be expected. Tp1 corresponds to the minimum cycle time (time in seconds) at 50 % duty cycle.

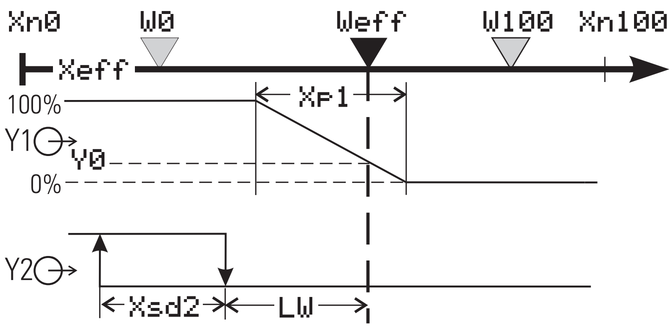
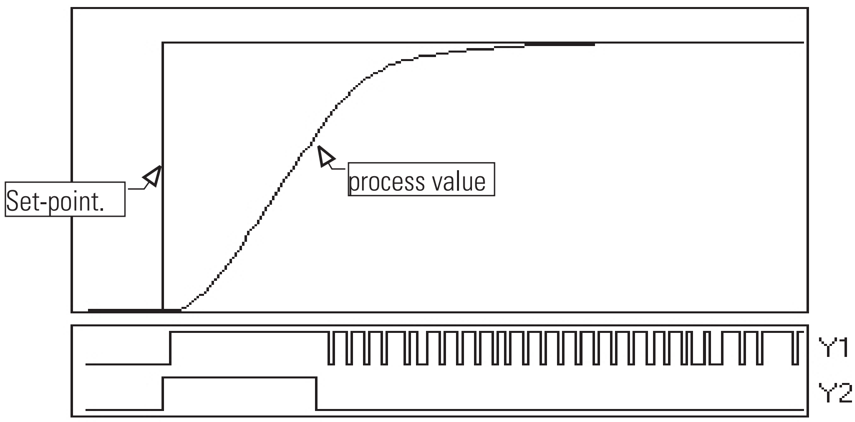


Fig.: 2 Static operating principle of the Δ / Y / off function



Output Y1 (heating)

Output Y2 (additional contact)

PD action (**Tn** = 0 ≙ switched off Tn = ∞)
The working point is in the middle of the proportional band X_{p1} at 50 % duty cycle.
For keeping the control variable constant, a defined amount of energy dependent of set-point is required. This causes a permanent control deviation, which increases with higher X_{p1}.

DPID action
By means of the I action, line-out without permanent control deviation is possible.
The static characteristic of a two-point controller is identical to the one of a continuous controller. The difference is that a duty cycle instead of a linearly variable current signal is output (relay contact, logic signal 0/20mA or control output 0/24V).
Working point Y_0 and cycle time Tp1 of the cycle at 50% are adjustable.
The shortest switch-on or off time is 100ms.

Configuration	Effective controller parameters with / Y / off controller			
CFunc = 2-P+Zusatz	Popt		Parameter set for self-tuning (only with CONTR+)	1...6
	W0	1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100	1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2	1)	Additional set-point	-29 999 ...999 999
	Grw+	2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw-	2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2	2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0		Zero offset (effective only with CType=ratio controller)	-29 999 ...999 999
	a		Factor a (effective only with CType=3-element control)	-9,99 ... 99,99
	LW	1)	Trigger point separation of additional contact OFF ≙ the additional contact is switched off	-29 999 … 999 999 -32 000 = OFF
	Xsd2	1)	Switching difference of additional contact	0,1 … 999 999
	Y2		Additional positioning value	0 … 100 [%]
	Ymin		Min. output limiting	0 … 100 [%]
	Ymax		Max. output limiting	0 … 100 [%]
	Y0		Working point of correcting variable (start-up correcting variable)	0...100 [%]
	YOptm		Positioning value during process at rest	0...100 [%]
	dYopt		Self-tuning step height	5...100 [%]
	Xp1(1...6)	3) 4)	Proportional band 1	0,1 … 999,9 [%]
	Tn1(1...6)	4)	Integral time	0 … 999 999 [s]
	Tv1(1...6)	4)	Derivative time	0 … 999 999 [s]
	Tp1(1...6)	4)	Cycle time heating	0,4 … 999,9 [s]
	Titel		Title of controller page (only display)	16 characters
	Einh.X		Unit of process value (only display)	6 characters
	W int	1)	Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999
	A/H		Controller status after transmission of the engineering to KS98	0 oder 1

1) The values are specified in the process value unit - e.g. [°C, °F, bar, %, etc.]
2) The rate of change must be specified in units/minute (e.g. °C/min).
→ see gradient control page 192.
3) % specifications are related to measuring range Xn100 - Xn0.
There is no relation to values W0 and W100.
4) (1...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Three-point stepping controller

Switching controller for control of a valve (e.g. temperature control by means of motorized valve and gas-air mixture)
1. Open valve; → outputs Y1 = 1, Y2 = 0
2. Don't move valve; → outputs Y1 = 0, Y2 = 0
3. Close valve; → outputs Y1 = 0, Y2 = 1

To validate the adjusted X_{p1} for the actuator response time, response time T_m must be adjusted. The smallest positioning step is 100ms.

Adjusting the neutral zone
Neutral zone X_{sh} can be increased in case of excessive output switching. However, note that an increase the neutral zone will reduce the control sensitivity.
For this reason, we recommend optimizing switching frequency (actuator wear) and control sensitivity.

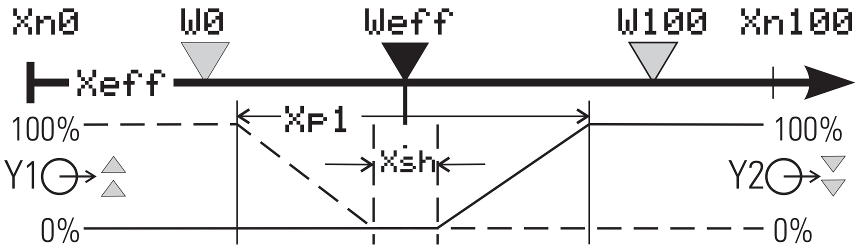
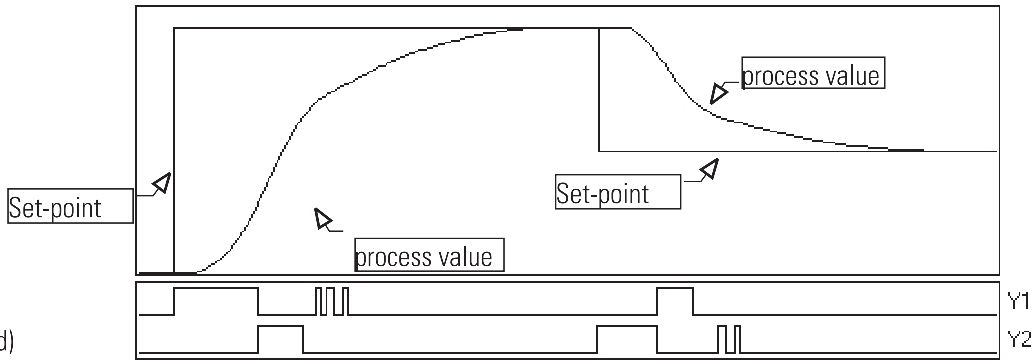


Fig.: 3 Static operating principle of a three-point stepping controller



Three-point stepping controllers can be operated with or without position feedback Yp .

Schritt 3-point stepping controller
SchrittYP 3-point stepping controller with position feedback

whereby Yp is not used for control. The static characteristic of a three-point stepping controller is shown in Fig. : 3 .

The hysteresis shown in this diagram is practically unimportant, but can be calculated from the adjustable min. pulse length $T_{puls} \geq 100$ms.

$$X_{sh} = \frac{\text{Tpuls}}{2} \cdot 0{,}1 \cdot \frac{\text{Xp}}{\text{Tm}}$$

With Tpuls switched off, the shortest positioning step Tpuls is dependent of Tm, Xsh and Xp. By varying Xsh , a required min. pulse length Tpuls can be realized:

$$X_{sh} = 12{,}5 \cdot \text{Xp} \cdot \frac{\text{Tpuls'}}{\text{Tm}} - 0{,}75$$

Configuration	Effective controller parameters with three-point stepping controller			
CFunc = Schritt Schritt Yp	Popt		Parameter set for self-tuning (only with CONTR+)	1...6
	W0	1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100	1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2	1)	Additional set-point	-29 999 ...999 999
	Grw+	2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw-	2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2	2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0		Zero offset (effective only with CType=ratio controller)	-29 999 ...999 999
	a		Factor a (effective only with CType=3-element control)	-9,99 ... 99,99
	Xsh	3)	Trigger point separation	0,2 … 20 [%]
	Tpuls		Min. positioning step time	0,1 … 2 [s]
	Tm		Actuator response time	5 … 999 999 [s]
	Y2		Additional positioning value (only with step Yp → with position feedback)	0 … 100 [%]
	YOptm		Positioning value during process at rest (not with PIDMA)	0...100 [%]
	dYopt		Self-tuning step height	5...100 [%]
	Xp1(1...6)	3) 4)	Proportional band 1	0,1 … 999,9 [%]
	Tn1(1...6)	4)	Integral action time	0 … 999 999 [s]
	Tv1(1...6)	4)	Derivative action time	0 … 999 999 [s]
	Titel		Title of controller page (only display)	16 characters
	Einh.X		Unit of the process value (only display)	6 characters
	W int	1)	Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999
	A/H		Controller status after transmission of the engineering to KS98	0 or 1

1) The values must be specified in the process value unit - e.g. [°C, °F, bar, %, etc.]

2) The rate of change must be specified in units/minute (e.g. °C/min).
→ see gradient control page 192.

3) % specifications are related to measuring range Xn100 - Xn0 .
There is no relationship to values W0 and W100.

4) (1 ...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Additional controller parameters for PIDMA
(for description, see section Controller characteristics)

Parameter	Description	Wertebereich
Tpause	Minimum positioning pause time (stepping controller)	0,1...999999[s]
thron	Switch-on threshold for OPEN and CLOSE	0,2...100%
throff	Switch-off threshold for OPEN and CLOSE	0,2...100%
Kp	Control gain (replaces Xp1/Xp2 of CONTR)	0,1...999,9[%]
VD	Derivative gain (Td/DT1)	1...999999
bW_p	Setpoint weighting factor of proportional action	0...1
cW_d	Setpoint weighting factor of D action	0...1
Tsat	Time constant for I action in Y limiting (anti-reset wind-up)	1...999999
xsh	Neutral zone in which the integral action is held	0 … 999999
dYopt	Step width with self-tuning using different range	-100 … 100

Continuous controller

Continuous controller
An analog value is provided as correcting variable by output Yout1 , e.g. temperature control with electrical heating and thyristor power regulator.
A continuous controller in 'split-range' operation is comparable with a three-point controller. The neutral zone is also separately adjustable.

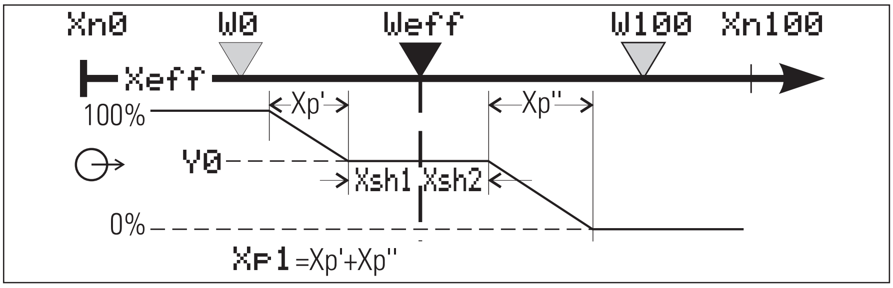
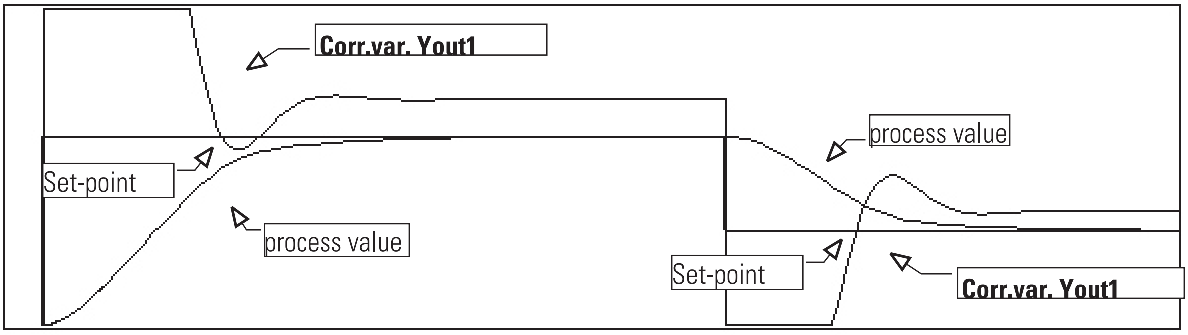


Fig.: 4
Operating principle of the proportional part of the continous controller

Innerhalb der Grenzen Xsh1 und Xsh2 wird die Regelabweichung zur Berechnung der Reglerreaktion zu Null gesetzt. Ein reiner P-Regler verändert innerhalb dieser Grenzen die Stellgröße nicht mehr. Ein PID-Regler hat ein dynamisches Verhalten, das auch bei Erreichen von “Regelabweichung = 0” nicht unbedingt abgeklungen ist. Sowohl der D- als auch der I-Teil können auf Grund einer vorausgehenden Störung oder eines Sollwertsprunges entsprechend der mit Tv festgelegten Charakteristik nachwirken. Das kann soweit gehen, dass der Bereich Xsh1/Xsh2 wieder verlassen wird, sodass der P-Teil noch einmal aktiviert wird, um endgültig in die neutrale Zone zu gelangen.



Selection from the following continuous controllers is possible:

1.) CFunc = continuous → continuous controller

2.) CFunc = splitRang → continuous controller with split-range operation
The continuous output is split on outputs Yout1 and Yout2 .

3.) CFunc = continuous Yp → continuous controller w ith position feedback.
The actually flowing positioning current can be displayed via input Yp .Yp is not included in the control operation.

Configuration	Effective controller parameters of a continuous controller		
CFunc = Stetig SplitRange	Popt	Parameter set for self-tuning (only with CONTR+)	1...6
	W0 1)	Lower set-point limit for Weff	-29 999 ...999 999
	W100 1)	Upper set-point limit for Weff	-29 999 ...999 999
	W2 1)	Additional set-point	-29 999 ...999 999
	Grw+ 2)	Set-point gradient plus	off / 0,001 ... 999 999
	Grw- 2)	Set-point gradient minus	off / 0,001 ... 999 999
	Grw2 2)	Set-point gradient for W2	off / 0,001 ... 999 999
	N0	Zero offset (effective only with CType=ratio controller)	-29 999 ...999 999
	a	Factor a (effective only with CType=3-element control)	-9,99 ... 99,99
	Xsh1 3)	Neutral zone (Xw > 0) (not with PIDMA)	0,0 … 1000 [%]
	Xsh2 3)	Neutral zone (Xw < 0) (not with PIDMA)	0,0 … 1000 [%]
	Y2	Additional positioning value	0 … 100 [%]
	Ymin	Min. output limiting	(-100) 0 … 100 [%]
	Ymax	Max. output limiting	(-100) 0 … 100 [%]
	Y0	Correcting variable working point(start-up correcting variable)	-100...100 [%]
	YOptm	Positioning value during process at rest (not with PIDMA)	0...100 [%]
	dYopt	Self-tuning step height	5...100 [%]
	Xp1(1...6) 3) 4)	Proportional band 1	0,1 … 999,9 [%]
	Xp2(1...6) 3) 4)	Proportional band 2 (only with continuous controller split range)	0,1 … 999,9 [%]
	Tn1(1...6) 4)	Integral action time	0 … 999 999 [s]
	Tv1(1...6) 4)	Derivative action time	0 … 999 999 [s]
	Titel	Title of controller page (only display)	16 characters
	Einh.X	Process value unit (only display)	6 characters
	W int 1)	Internal set-point after transmission of the engineering to KS98	-29 999 ...999 999
	A/H	Controller status after transmission of the engineering to KS98	0 or 1

1) The values must be specified in the process value unit, e.g. [°C, °F, bar, %, etc.]
2) The rate of change must be specified in units/minute e.g. °C/min).
$\rightarrow$ see gradient control page 192.
3) % specifications are related to measuring range Xn100 - Xn0 .
There is no relationship to values W0 and W100.
4) (1 ...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Additional controller parameters for PIDMA

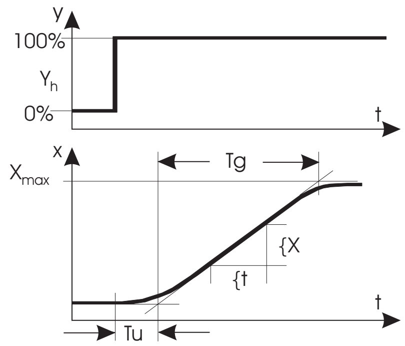
(for description, see section Controller characteristics)

Parameter	Beschreibung	Wertebereich
Kp	Control gain (replaces Xp1/Xp2 of CONTR)	0,1...999,9[%]
VD	Derivative gain (Td/DT1)	1...999999
bW_p	Setpoint weighting factor in proportional action	0...1
cW_d	Setpoint weighing factor in D action	0...1
Tsat	Time constant for I action in Y limiting (anti-reset wind-up)	1...999999
xsh	Neutral zone, in which the I action is held	0 .. 999999
dYopt	Height of step change during self-tuning	-100 … 100

24.7 Optimizing the controller

Process characteristics

characteristics are determined automatically by the controller during self-tuning and converted into control parameters. In exceptional cases, however, manual determination of these process characteristics may be necessary. For this, the response of process variable x after a step change of correcting variable y can be used (see Fig.: 10).
Usually, it is not possible to plot the complete response curve (0 to 100 %), as the process must be kept within certain limits.
The maximum rate of increase v_{max} can be determined from the values T_g and x_{max} (complete step response) or Δt and Δx (partial step response).



$$K = \frac{Vmax}{Xh} \cdot Tu \cdot 100\%$$

y = correcting variable
Y_h = control range
Tu = delay time (s)
Tg = recovery time (s)

$$V_{max} = \frac{X_{max}}{Tg} = \frac{\Delta x}{\Delta t} \triangleq \text{ max. rate of increase of process value}$$

X_{max} = maximum process value
X_h = control range ≙ x100 - x0

Characteristic values of the controllers

Generally, quick line-out to the set-point without oscillation is required.
Dependent of process, different control actions should be used.

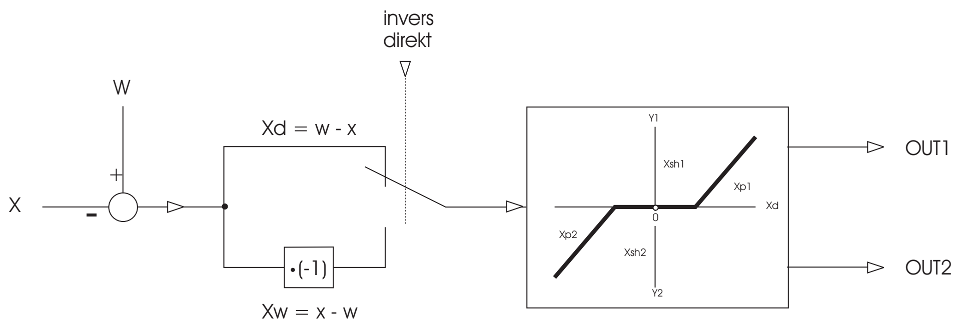
- Processes with good controllability (K < 10%) can be controlled by means of PD controllers,
- Processes with medium controllability (K = 10...22%) can be controlled with PID controllers and
- Processes with bad controllability (K > 22%) can be controlled with PI controllers.

The control parameters can be determined from the calculated values of delay time T_u, max. rate of inrease v_{max}, control range X_h and characteristic value K according to the formulas. For more exact adjustment, see the hints given in the table of parameter adjustment effects. Increase Xp if line-out oscillates.

Formulas			
Action	Xp[%]	Tv[s]	Tn[s]
(D)PID	1,7 K	2 Tu	2 Tu
PD	0,5 K	Tu	∞ = 0000
PI	2,6 K	0	6 Tu
P	K	0	∞ = 0000
3-point stepping controller PID			
	1,7 K	Tu	2 Tu

Parameter adjustment effects				
Setting		Control	Line-out of disturbances	Start-up behaviour
Xp	higher	Increased damping	Slower line-out	Slower reduction pf energy
	lower	Reduced damping	Faster line-out	Faster reduction of energy
Tv	higher	Reduced damping	Faster response	Earlier reduction of energy
	lower	Increased damping	Slower response	Later reduction of energy
Tn	higher	Increased damping	Slower line-out	Slower reduction of energy
	lower	Reduced damping	Faster line-out	Faster reduction of energy

Fig.: 5 Principle of direct / inverse switchover



Direct / inverse switchover is always possible in configuration parameter CMode (action).
The principle is shown in Fig.: 5.

24.8 Self-tuning → controller adaptation to the process

For determination of optimal parameters a self-optimization can be accomplished.
This is applicable for controlled systems with reconciliation and none dominating dead time and K ≤ 30%. After start by the operator the controller initiates an adaptation cycle in order to determine the line characteristic values Tu and Vmax. It calculates by it the control parameters for fast, overshoot-free correction to the set-point (Xp1, Xp2, Tn, Tv, Tp1, Tp2, depending upon kind of controller).

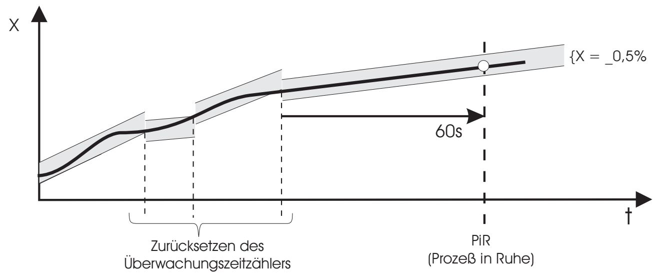
Preparation

- Set the desired controller behaviour
  P-controller: Tn = 0.0 Tv = 0.0
  PD-controller: Tn = 0.0 Tv > 0.0
  PI-controller: Tn > 0.0 Tv = 0.0
  PID-controller: Tn > 0.0 Tv > 0.0

  The parameters Tn and/or Tv can be switched off, by being adjusted to the value = 0,0.
  Thus they do not participate in the self-optimization.
- With the automatic controller CONTR+ is to be selected, which parameter set is to be optimized (POPt=1... 6).
- Configure conditions for process at rest (OCond)
  The condition designates, when "Prozess at rest" is to be recognized (PIR_H):
  grad=0, grad<0/>0 or grad0 (→ also see process at rest, page 177).
- The correcting variable YOPtm is to be specified. This is, in automatic running, the correcting variable, which is output with the start of self-optimization in order to generate "Prozess at rest".
- The step of the correcting variable dYOPt is to be specified. dYOPt is the amount the correcting variable jumps, from the initial value YOPtm and/or in manual operation from the original correcting variable.
- Consider the set-point reserve
  (→ also see set-point reserve, page 178)

‘Process-at-rest’ monitoring (PiR):

‘Process-at-rest’ monitoring is done at any time. The proces is at rest, when the process value is within a tolerance band of ±ΔX = 0.5% during more than 60 seconds. When the process value is out of this band, the monitoring timeout counter is reset to zero. With detection of PiR e.g. during control operation and output of a widely deviating stable correcting variable YOPtm at self-tuning start, waiting until the full PiR time has elapsed is required.



With extended monitoring, monitoring is for a constantly varying instead of a constant process variable!

Configuration word OCond can be used to determine ‘Process at rest’detection. One of the following modes can be selected:

grad(x) = 0:	Process at rest is detected, when x is constant.
grad(x) <0/>0:	Process at rest is detected when x decreases constantly with a controller with inverse output action. Process at rest is detected, when x increases constantly with a controller with direct output action.
grad(x) <> 0:	Process at rest is detected with constantly changing x. In this case, continuation of this constant change over the duration of identification must be ensured.

Set-point reserve:

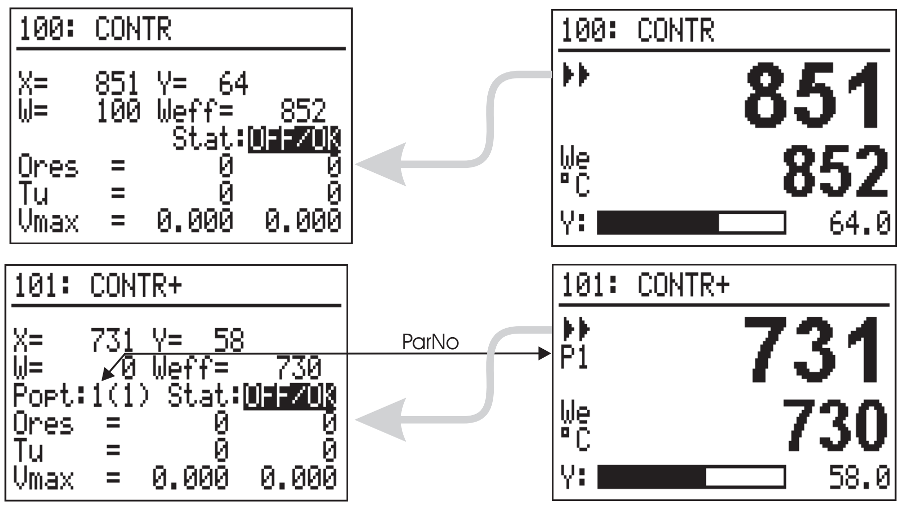
In order to make self-tuning possible, the separation between set-point and process value before the output step change must be higher than 10 % of W0...W100. The set-point reserve is provided either automatically by reducing the correcting variable during the PiRphase, or by changing the set-point or the process value manually (manual mode).
With inverse controllers, the set-point must exceed the process value by at least the set-point reserve.
With direct controllers, the set-point must be smaller than the process value by at least the set-point reserve. This is necessary, as the set-point is a limit which should not be exceeded during self-tuning.

Self-tuning start

Self-tuning can be started or stopped from automatic or from manual mode on the self-tuning page (→ see "Start from automatic mode" page 188 and "Start from manual mode" page 182).
Selec the self-tuning page by marking the two arrows ▸▸ followed by configuration. Select function `Stat: OFF/OK` (inverse display) and confirm it by ⏎
`Stat: OFF/OK` blinks and can be switched over to `Stat: Start` by pressing ▲ .
Press key ⏎ to start the self-tuning attempt. Set-point adjustment is always possible.

Self-tuning cancelling

A self-tuning attempt can always be cancelled.
Self-tuning can always be stopped by pressing automatic/manual key on the controller front panel, provided that key was not disabled (1-signal on input `oPlook`).
Moreover, cancellation is possible from the self-tuning page of the required controller. For this, press key ▲ on the self-tuning page to select the `Stat:` line (inverse display), press ⏎⑮ `Stat:` line blinks. Press ▲ until `Stat: Stop` blinks. Press ⏎, the self-tuning attempt was stopped and the controller continues operating in automatic mode.



Start from automatic mode:

After self-tuning start, stable correcting variable YOptm is output. When ‘Process at rest’ (PiR) is detected and a sufficient set-point reserve (→ see page 178) is provided, the correcting variable is changed by output step dYOpt (boosted with indirect controller, lowered with direct controller).
The self-tuning procedure is realized using the varying process value.

Fig.: 6 self optimization [grad(x)=0]

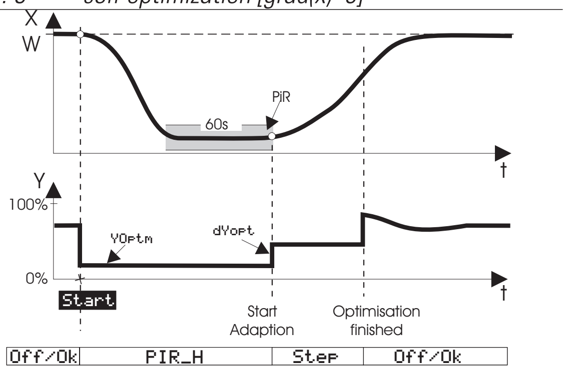
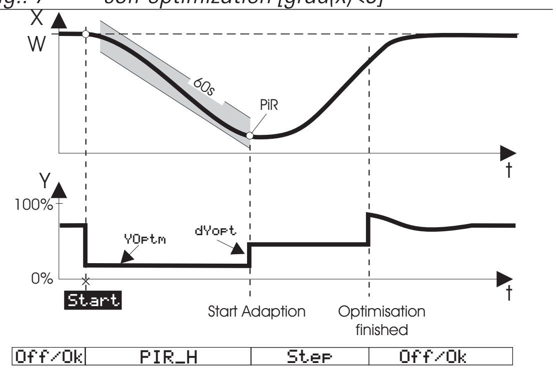


Fig.: 7 self optimization [grad(x)<0]



After successful self-tuning, the controller goes to the automatic mode and controls the set-point using the new parameters. Parameter Ores indicates the self-tuning result (→ see page 189) .

If self-tuning is finished with an error (Ada_Err), the stable correcting variable is output, until self-tuning is finished by the operator via the system menu, front-panel key , or via the interface.

Start from manual mode

To start self-tuning from manual mode, switch the controller to manual. During transition to manual mode, the correcting variable output last is taken over as manual correcting variable. At self-tuning start, this correcting variable is taken over and output as temporary stable correcting variable. Like in automatic mode, the set-point can be changed at any time.

With ‘Process at rest’ (PiR) detection and a sufficient set-point reserve (→ see page 178), the correcting variable is changed by the correcting variable step dYOpt (boosted with indirect controller, lowered with direct controller).
`*Process at Rest'* (PiR) can be reached at starting time, i.e. the normal 60 s waiting time can be omitted.
The self-tuning procedure is realized using the varying process value.

Fig.: 8 Start by boosting the set-point

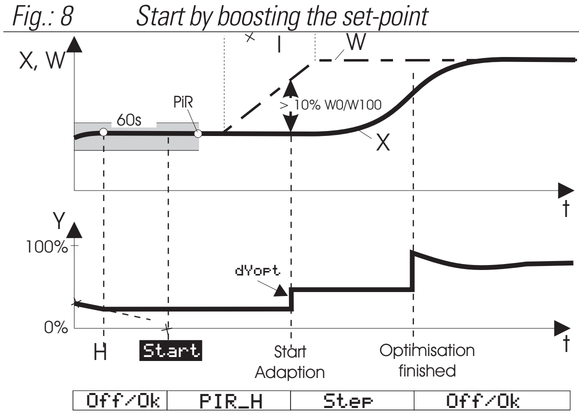
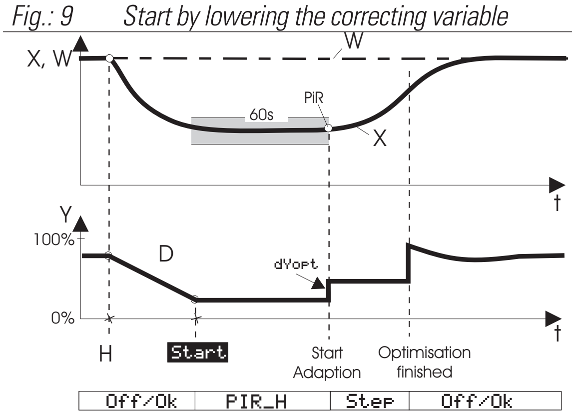


Fig.: 9 Start by lowering the correcting variable



After successful self-tuning, the controller goes to the automatic mode and controls the set-point using the new parameters. Parameter Ores indicates the self-tuning results (→ see page 189) .

When finishing self-tuning with an error (Ada_Err), the stable correcting variable is output, until self-tuning is finished by the operator via system menu, front-panel key , or via interface.

Self-tuning procedure with heating:

(2-point, 3-point stepping, continuous controller)
After reaching ‘Process in rest’, the process is stimulated by means of an output step change and the process response is used to determine Tu1 and Vmax1 at the step response reversal point, if possible.

Self-tuning procedure with heating and cooling processes:

(3-point / split-range controller)
Self-tuning starts as with a “heating” process. After self-tuning end, the controller settings based on the calculated parameters are made. This is followed by line-out at the pre-defined set-point, until PiR is reached again. Subsequently, a step to cooling is made to determine the “cooling” parameters, in order to determine Tu2 and Vmax2 using the step response. Based on these characteristics, the controller settings for the cooling process are made. When cancelling the cooling attempt, the parameters for “heating” are also taken over for cooling. No error message (Ada_Err) is output.

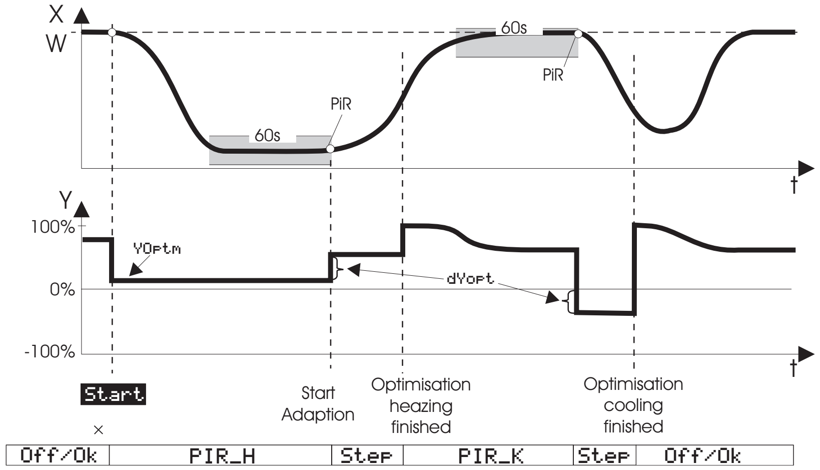


Fig.: 10
Self-tuning with heating and cooling

With 3-point stepping controllers, the motor actuator is closed first after starting and opening to YOptm will occur only then. This calibration procedure (Stat: Absl.) is not shown in the figures.

For maintaining a safe process condition, monitoring for an exceeded set-point is done continuously.

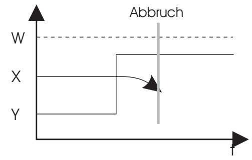
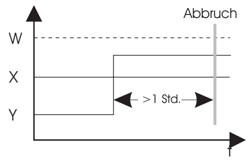
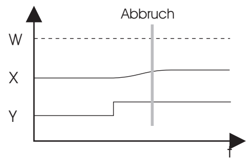
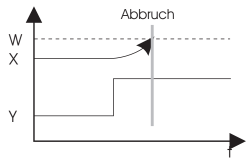
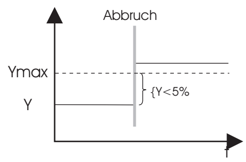
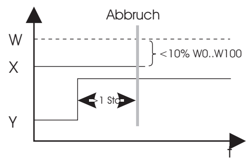
During self-tuning, the control’ function is switched off! I.e.: Ypid is within the limits of Ymin and Ymax.

With △/⅄/ Off controllers, self-tuning is using the ⅄ function, i.e.Y2 = 0.

Controlled adaptation

For defined applications, adaptation of the control parameter set to the current process condition is purposeful. For this, the Contr+ is provided with 6 control parameter sets, which can be selected via analog input ParNo .

Significaion of self-tuning messages ORes1/ORes2

ORes1/2	Signification or trouble cause	Possible solution
0	No attempt was made or attempt cancelled by Stat: Stop or switchover to manual mode (key) .	
1	**Cancellation:** Faulty correcting variable output action, X does not change in the direction of W. 	Change controller output action.
2	**Finished**: self-tuning was completed successfully (reversal point found, safe estimation)	
3	**Cancellation:** The process does not respond or responds too slowly (change of ΔX smaller than 1% in 1 hour) 	Close control loop.
4	**Completed**, withou6 AdaErr: Successful attempt, process has a low reversal point	Optimum result with low reversal point
	Cancellation, with AdaErr: Attempt failed, process stimulation low (Reversal point found, but estimatio n is unsafe) 	Increase output step change dYOpt.
5	**Cancellation:** Self-tuning cancelled because of exceeded set-point hazard. 	Increase separation of process value (X) and set-point (W) when starting, or decrease YOptm .
6	**Completed**: attempt successful, but self-tuning cancelled due to exceeded set-point hazard. (Reversal point not reached so far; safe estimation).	
7	**Cancellation**: Output step change too small, ΔY < 5%. 	Increase Ymax or set YOptm to a smaller value.
8	**Cancellation:** Set-point reserve too small, or exceeded set-point whilst PiR monitoring is busy. 	Vary stable correcting variable YOptm.

Unless control is as required despite self-tuning, proceed additionally as described in section "Empirical self-tuning"(→ page 182 , Hints for self-tuning, Hints for adjustment), and follow the hints given on further parameters.

24.9 Controller characteristics and self-tuning with PIDMA

As opposed to CONTR and CONTR+, the PIDMA includes a modified parallel controller structure, which is taken into account in the following additional parameters.

Additional parameters for PIDMA

Parameter	Beschreibung	Wertebereich
PType	Process type (a-priori information)	1: with compensation 2: without A.(integral)
Drift	Drift compensation des Istwertes zu Beginn der Selbstoptimierung	0: off 1: on
CSpeed	Required control loop dynamics	1: slow 2: normal 3:fast
Tpause	Minimum positioning step time (stepping controller)	0,1...999999[s]
thron	Switch-on threshold for OPEN and CLOSE (stepping controller)	0,2...100%
throff	Switch-off threshold for OPEN and CLOSE (stepping controller)	0,2...100%
Xlimit	Switch-off point for output step change (process value change)	0,5...999999
Tdrift	Time window for process value drift determination	0...999999
Tnoise	Time window for process value noise determination	0...999999
Kp	Control gain (replaces Xp1;/Xp2 of CONTR)	0,001...999,9[%]
VD	Derivative gain (Td/T1)	1...999999
bW_p	Setpoint weighting factor of proportional action	0...1
cW_d	Setpoint weighting factor of D action	0...1
Tsat	Time constant for I action in Y limiting (anti-reset wind-up)	1...999999
xsh	Neutrale Zone, in dem der I-Teilfestgehalten wird	0 … 999999

Three point stepping(Yp):

Tpause , thron and throff complete the effective parameters for stepping motor control.

Tpause permits adjustment of the minimum pause in addition to minimum pulse limiting by means of Tpuls.

thron and **throff** define a deviation of the actuator position from an internal setting (position feedback required) from which a pulse to the actuator is output or canceled. In connection with Xsh, these parameters can be used to influence the actuator switching frequency and fine adjustment. Xsh determines the neutral zone of control deviation in the master controller. Within this zone, the controller I action is held.

With setting 3-point stepping Yp (three-point stepping with position feedback, the PIDMA function block comprises two controllers: the master controller controls the process value and provides a required actuator position to an integrated position controller). By means of the position feedback, this position controller ensures that the actuator goes to the required position.

Self-tuning:

PType, Drift, Cspeed, Xlimit, Tdrift and Tnoise complete parameter dYopt which is also effective with CONTR. These parameters define the conditions during self-tuning.

Ptype determines, if the process is without compensation (the new process value after a correcting variable pulse is higher, e.g. level of a container without outlet or well- insulated furnace). An even decrease or increase of the process value before self-tuning can be detected by means of **drift** monitoring and taken into account when self-tuning is done for the next time.

CSpeed can be used to determine if, during subsequent operation, the controller should reach the setpoint quickly, with a slight overshoot, or slowly with gentle approach to the setpoint. Using **CSpeed** , the parameter can be switched over also after self-tuning, provided that the controller parameters were not changed manually.

After self-tuning start, timer **Tdrift** for process value drift detection and timer **Tnoise** for noise detection (variations independent of the correcting variable) are started. Dependent on process, the timers should be long enough to permit detection of an interference-independent drift and multiple "ups" and "downs" of interference effects.

After elapse of these timers, the actual correcting variable is increased by **dYopt** . When the process value has increased by more than **Xlimit** under consideration of drift and noise, the correcting variable is reset to the initial value. However, self-tuning is completed only, when the process value has decreased to nearly half of the initial value after exceeding the maximum. During decrease after the correcting variable pulse, the estimated remaining time until self-tuning end is displayed continuously. After completion of self-tuning, the determined parameters K, Ti and Td are displayed on the self-tuning page, taken over into the function block together with parameters VD, BW_p and CW_d and activated for the running process.

Control parameters of PIDMA:

Unlike CONTR, PIDMA does not have separate parameters for heating and cooling. Parameter **K** which is valid for both ranges determines the control gain of a parallel controller structure.

Further parameters permit independent weighting of individual controller components:

VD: In addition to the control gain, the derivative gain (Td/T1) permits an increase or reduction of the derivative action.

BW_p: Setpoint weighting factor of proportional action

CW_d: Setpoint weighting factor of derivative action.

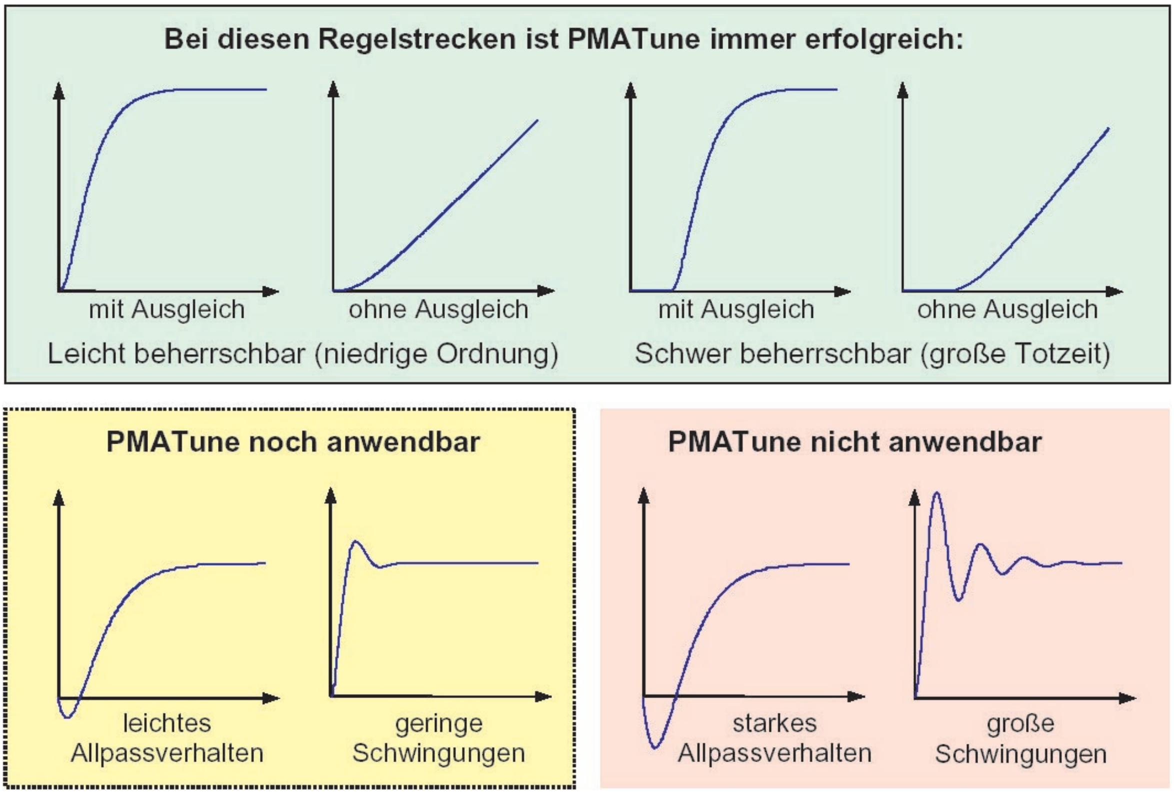
Parameters BW_p and CW_d can reduce the effect of a setpoint change on the controller reaction. I.e. different controller behaviour after setpoint changes (control behaviour) or process value changes (disturbance behaviour) can be selected. A factor within 0 and 1 can be applied to the setpoint effect.

In the course of dynamic process control, the control algorithm can temporarily determine values below 0 or above 100 for the correcting variable. If necessary, however, these values can be reset to the limits by means of accelerated integral behaviour (Tsat).

Tsat time constant for integral action in Y limiting (anti-reset wind-up).

Self-tuning → controller adaptation to the process (PIDMA)

Self-tuning can be started to determine the optimum parameters for a process. The function is applicable for the following processes.



Preparation

- Adjusting the required control behaviour.
  P-controller: Tn = 0.0
  Tv = 0.0
  PD-controller: Tn = 0.0
  Tv > 0.0
  PI-controller: Tn > 0.0
  Tv = 0.0
  PID-controller: Tn > 0.0
  Tv > 0.0
  Parameters Tn or Tv can be switched off by setting = 0.0. I.e. these parameters do not participate in self-tuning.
- Correcting variable step change dYopt must be determined. This is the value by which the correcting variable changes from the actual value. The step change can be positive or negative.
- Xlimit must be determined. It should be set to roughly half of the expected process value change.

"Process at rest" control:

The PIDMA doesn't control the "process at rest". The comissioner may choose the appropriate starting point.
Best results are received, if the process is lined out (all dynamic proceedings have faded away). In some cases when determining of parameters fails because of fading dynamic proceedings, the algorithm will deliver the error message "start new".

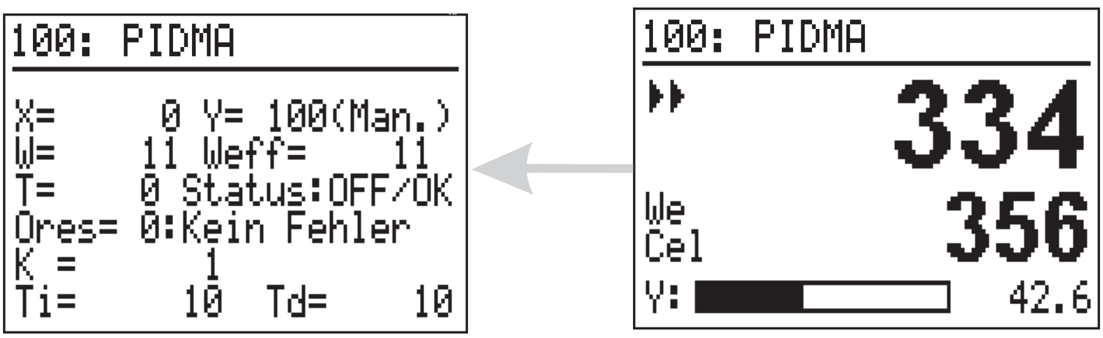
Self-tuning start

Self-tuning can be started and stopped from automatic or manual operation on the self-tuning page.
Select the self-tuning page by marking and confirming the two arrows. Select function Stat: OFF/OK (inverse display) and confirm it with .
Stat: OFF/OK blinks and can be switched over to Stat: Start by pressing I.
Pressing key starts the self-tuning attempt. The setpoint can be changed at any time. However, this is not necessary as opposed to CONTR. A change after starting from automatic mode would even cause faulty process evaluation.

Self-tuning cancelation

Self-tuning can be stopped at any time by pressing key manual/automatic on the controller front, provided that key H was not disabled (1-signal on input oplook).
Moreover, cancelation is possible on the self-tuning page of the required controller. For this purpose, select the Stat: line (inverse display) on the self-tuning page, press M, Stat: line blinks. Press I until Stat: Stop blinks. Press M, the self-tuning attempt is stopped and the controller continues operating in automatic mode.

Self-tuning page

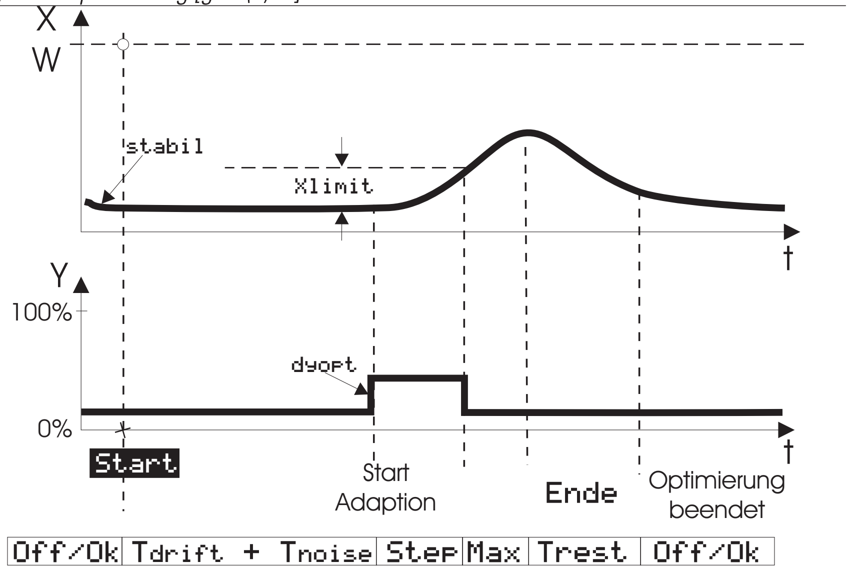


Start in manual mode or in automatic mode :

Basically, the PIDMA self-tuning algorithm does not distinguish between these two start conditions. In both cases, the operator must ensure that the process conditions are stable. In automatic mode, however, the PIDMA works with the non-optimized parameters until start of the correcting variable pulse. This means that, in the majority of cases, better stability of process conditions, i.e. better self-tuning results, are possible in manual mode. When changing to manual mode, the correcting variable output last is taken over as manual correcting variable and used during estimation.

After self-tuning start, the estimation timer for drift detection and noise detection is started at first. In the second phase, the correcting variable is changed by correcting variable step change dYOpt. When the process value has changed by more than Xlimit, the correcting variable is reset to the original value. In the third phase, the PIDMA waits for the maximum value of the increasing process value. Subsequently, it monitors the decaying process value in the fourth phase. During this time, an estimation of the remaining time until completion of the self-tuning attempt is output.
After a successful self-tuning attempt, the controller goes to the automatic mode and controls the setpoint using the new parameters. Parameter Ores indicates the result of self-tuning completion (→ see page 189) .

FigSelbstoptimierung [grad(x)=0]



When self-tuning is finished with an error (Ada_Err), the stable correcting variable is output, until self-tuning is finished by the operator via the system menu, front panel key H, or via the interface.

Self-tuning procedure with heating and cooling processes:

(3-point / split-range controller and mixed controllers)
With PIDMA, different control gains for heating and cooling cannot be specified. For this reason, the 2-step self-tuning attempt is omitted.

Signification of self-tuning messages ORes

ORes	Signification or error cause	Possible solution
0	No attempt was made	
1	Xlimit too small	Step change threshold too small: compared to the process noise, the step change threshold is too small. Start a new attempt with a higher positioning pulse.
2	dYopt too high	Positioning pulse too high: the correcting variable would exceed the positioning limits when the selected pulse height is output. Start a new attempt with smaller positioning pulse or reduce the correcting variable in manual mode previously.
3	Re-start	No rest. The autotuner has detected that the process is probably not at rest. Please wait, until reaching the rest condition. Another possibility is to activate the drift compensation or to increase the positioning pulse. Note: With pulse width modulated (PWM) control outputs (2 and 3-point controller), oscillations of process value PV are susceptible to occur even during manual mode, if the corresponding cycle time t1 (t2) is too long. In this case, the controller cycle times should be as low as possible.
4	dYopt small	Positioning pulse too small: the step response is hidden by process noise. Start a new attempt with a higher positioning pulse, or take measures to reduce the noise (e.g. filter).
5	No extreme	Max. detection failed: after output of the positioning pulse, no maximum / minimum in the process value curve was detected. The settings for the process type (with / without compensation) should be checked.
6	Positioning limit	Positioning limits during self-tuning were exceeded. During the attempt, correcting variable MV has exceeded the positioning limits. Repeat the attempt using a smaller positioning pulse or a reduced correcting variable during manual mode.
7	Controller type	No self-tuning result for the specified combination P/I/D can be found.
8	Monotony	Process not monotonous: the process has a strong all-pass behaviour (temporarily, the process value runs in opposite direction) or serious trouble during the attempt.
9	Estimation error	Extrapolation failed: after the positioning pulse end, no process value decrease was detected because of excessive noise. Increase the positioning pulse or attenuate the noise.
10	No result	Result useless: excessive noise, or the determined process parameters do not correspond to the description of a process with dead band. Start a new attempt with a higher positioning pulse or attenuate the noise.
11	Man. cancelation	The self-tuning attempt was canceled manually by the operator with „*STOP*" .
12	Output action	Faulty output action: the expected output action of the step response is opposed to the correcting variable. Cause can be faulty setting of the output action, or e.g. inverting actuators. Change the controller output action.

After successful self-tuning, parameter CSpeed can be used to increase or reduce the attenuation, when self-tuning was done with the setting for CSpeed = "normal" . Moreover, only an increase or reduction of Kp should be considered. After manual change of the controller parameters, the Cspeed switch-over stops being effective.

24.10 Empirical optimization

With missing distance data can empirically be optimized by means of the self-optimization or in manual attempts.With the attempts for empirical optimization the following is to be considered:

- It is to be guaranteed that correcting variable and controlled variable take never forbidden values!!!
- The conditions for the attempts should be always alike, in order to win comparable statements.
- The test sequence must be oriented at the goal of the optimization: Leadership- or interference behaviour.
- The operating point of the controller must be alike with the attempts.

The control parameters are to be set as follows with their first use:
Xp maximum: to the largest adjustable value,
Tv relatively large: time max., which the controlled system needs for distinct beginning of the reaction.
Tn large:time max., which the controlled system needs for the entire process.

The time requirement for an empirical optimization is large. In order to achieve an useful result in relatively short time , the following is recommended for appropriate procedure results:

Adjust Tn=Tv=0 and Xp largest possible(p-controller). The Xp is reduced from attempt to attempt, as long as the control is sufficiently stable. If it becomes too unstable, then the Xp is to be increased and next step is ②.

Measure lasting offset: If it is sufficiently small, then the optimization is successfully terminated (P). If it is too large, then the controlled system is better regulated with PD (adjust Tv relatively large and next step is ③).

Reduce Xp from attempt to attempt, as long as the control is sufficiently stable. If it becomes too unstable, then the next step is ④.

Tv is to be made smaller and determined whether the regulation can be sufficiently stabilized again. If, then it the next step is ③, if not, then Xp is to increase and the next step is ⑤ .

Determine whether with the procedures ③ and ④ the Xp was substantially made smaller. If, then the next step is ⑥, if not, then the controlled system better is pi-regulated (Tv set to 0 and the next step is >).

Measure lasting offset. If it is sufficiently small, then the optimization is successfully terminated (PD). If it is too large, then the controlled system is better PID-regulated (no longer change Xp and Tv and the next step is ⑦).

Tn is adjusted largely and reduced from attempt to attempt, as long as the control is sufficiently stable. If it becomes too unstable, then the Xp is to be increased, and the optimization is successfully terminated (PID or pi).

ⓘ For the controlled variable (process value X) the empirical optimization is substantially improved with a writer (or trend function of the engineering tool) in time requirement and quality, and evaluation of the test results is clearly simplified.

ⓘ The procedure mentioned can only with restrictions be generalized and does not lead to a clear improvement of the behavior with all controlled systems.

ⓘ Changes of the operating point (Y0), the switching point distance (Xsh) and the lasting switching period (Tp1 and Tp2) lead to results, which can be better or worse. With 3 - Point - step controllers TM must be adjusted to the real running time of the conncted actuator.

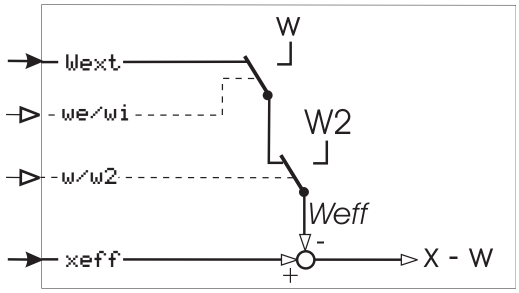
24.11 Set-point functions

Terminology

w internal set-point
we external set-point
w2 second (internal) set-point
Weff effective set-point
xw control deviation (x-w → process value - set-point)

General

Several possible set-points are available.
For the priorities, see the drawing shown opposite. "Safety set-point" W2 is given priority over the other set-points. Switchover between set-points is possible via interface or via the digital inputs of the conroller block.
If gradient control was activated, a set-point change will be made effective continuously instead of being made effective by a step → gradient control page 192.
By activating digital input w stop , the instantaneously effective set-point is maintained. In this case, neither set-point change nor switchover to another set-point becomes effective.



Set-point / set-point / cascade

Configuration word WFunc can be used to select, if switch-over to the external set-point (set-point/cascade) is also possible in addition to the internal set-point.

Set-point

(WFunc = set-point) Set-point control means that the set-point is firmly predefined by the internal set-point w.

Set-point/cascade

(WFunc =Fest/Folg) Set-point/cascade control permits switchover between internal set-point W and external set-point We . Switchover is via digital input we/wi or via the interface. Unless this iput is connected, or if a 0 signal is applied, the external set-point is used as effective set-point. Unless digital input we/wi as well as analog input wext are connected, the controller invariably uses the internal set-point.

W2 - safety set-point

The second set-point W2 can be activated at any time and has highest priority. The change-over between internal set-point and W2 can be triggered via interface or the digital control input w/w2. In order to make the W2 effective, on w/w2 is a logic 1 to be attached. If the internal set-point is to be active, a logic 0 must be given on we/wi.
In the past W2 was designated as "safety set-point". Whether W2 takes over safety functions or only a pre-defined starting position in certain process conditions, becomes determined only by the kind of the use and integration into an automation concept.

External set-point Wext

Switching between the internal set-point (wi) and the external set-point (we) is possible only if the parameter WFunc is adjusted to fixedfollow. The change-over can be triggered via interface or the digital control input we/wi. In order to make the internal set-point effective, on we/wi must be attached a logic 1. If the external set-point is to be active, a logic 0 must be given on we/wi.
If the digital control input we/wi is not wired, the external set-point is effective.

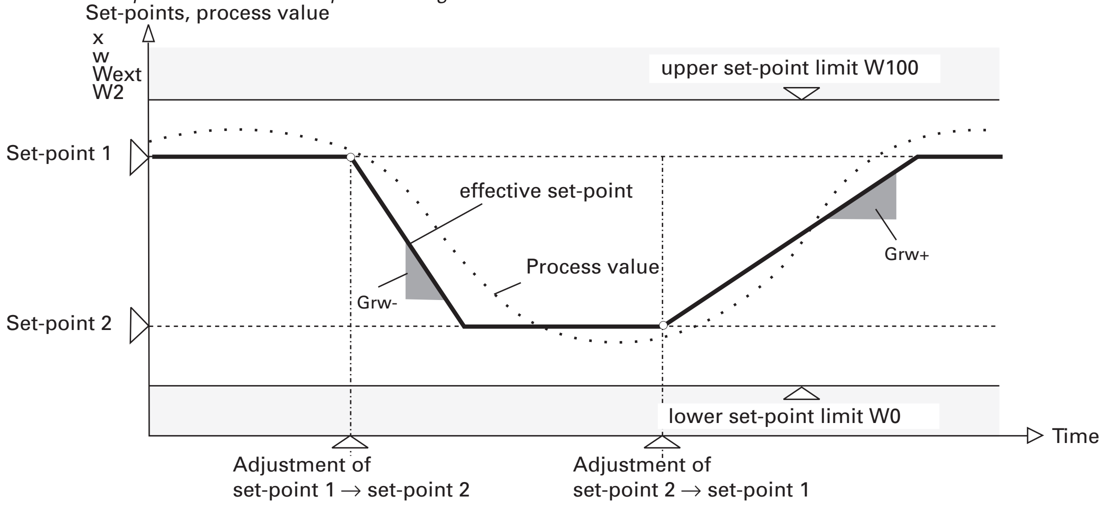
The internal set-point W is evaluated with priority. If in a place (interface or the digital control input we/wi) is switched to internal set-point it is not possible to switch over to the external set-point Wext the other place.

Gradient control - set-point changes with gradients

Normally, set-point changes occur stepwisely. Unless this behaviour is required, a gradient can be set-up using parameters Grw+ and Grw- or Grw2.

If these parameters are set, the set-point changes are made bumplessly. With digital input 'gr_off' not set, effective set-point Weff runs linearly towards the changed set-point (target value), whereby the slope is determined by gradients Grw+ and Grw- adjustable at parameter setting level (→ see Fig.: 12). For the second set-point W2, an independent gradient Grw2 was introduced, which is valid for both change directions and for w → W2 switchover. The gradient function is switched off when setting Grw+ and Grw- or Grw2 to "----" (engineering tool = off), or with digital input Gr off set to 1.

Fig.:12 Ramp function with set-point change

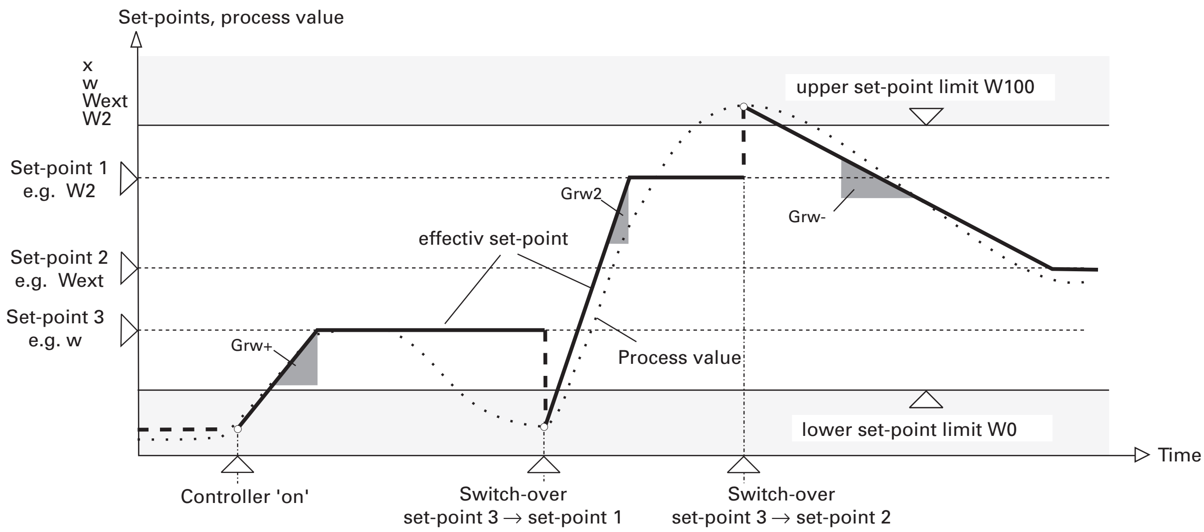


Set-point switch-over with gradients (W→W2, W→Wext, controller"on")

The new set-point is linear started outgoing from the momentary process-value. The slope of the ramp is determined related to the direction of Grw+, Grw- and/or Grw2.

This principle applies, even if the process-value is outside the adjustable set-point range W0/W100 at switch-over time (e.g. when starting).

Fig.:13 Rampfunction with controller "on" and set-point switch-over

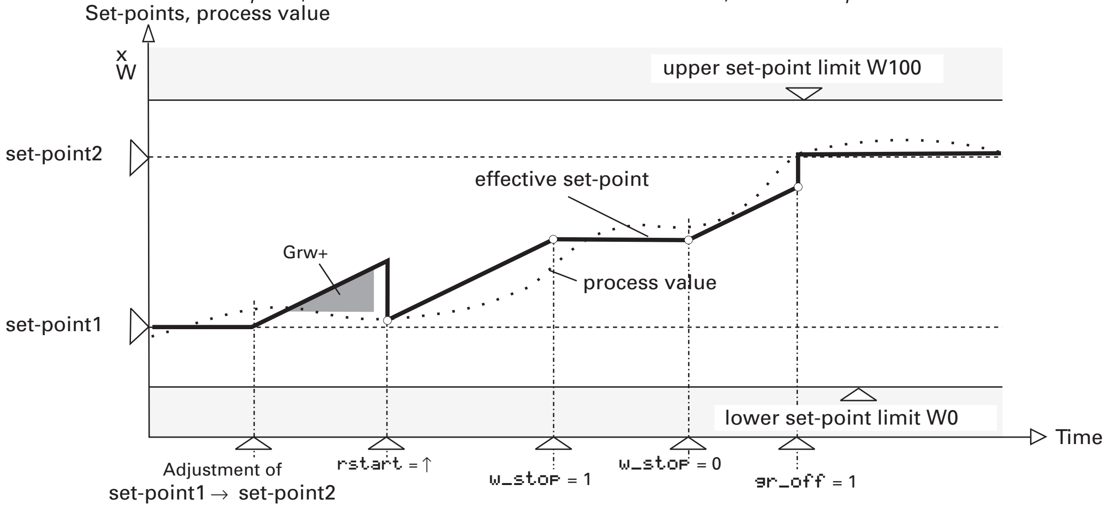


Controlling the set-point

The digital input rstart reacts to a positive signal slope and sets the effective set-point to the process value. The new goal set-point is started on the basis of the controlled variable xeff .

Such a ramp can only be started with activated gradient function (Grw+, Grw -, Grw2 and digital input gr_off not set). The digital input w_stop freezes the effective set-point Weff , i.e., the effective set-point is held to the current value, even if the effective set-point straight approaches a new goal set-point or a new goal set-point is selected.

Fig.: 14 Effect of control inputs (rstart, w-stop und gr_off) on the set-point



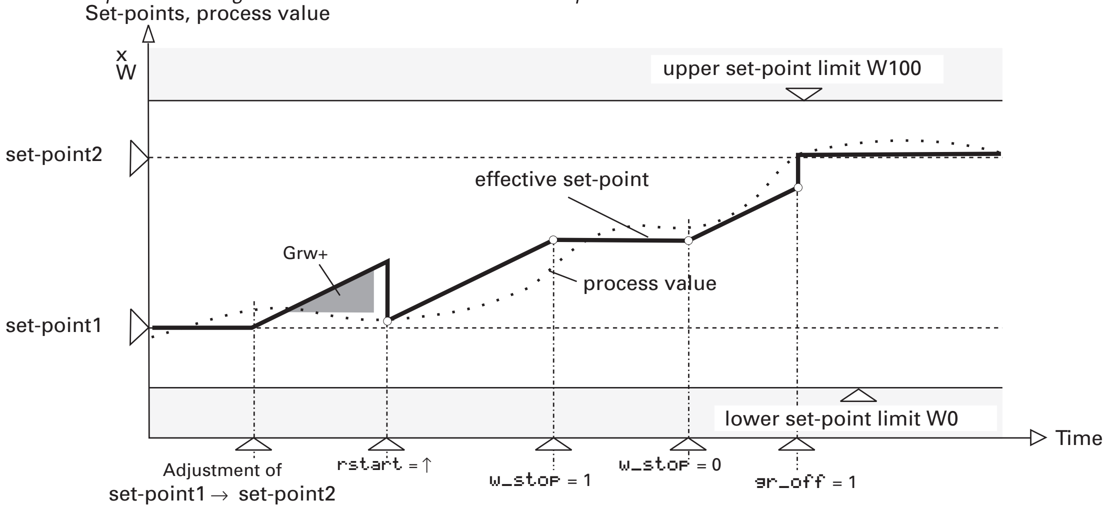
Set-point tracking

During the change-over of Wext → W it can come to unwanted set-point jumps. To aviod these jumps set-point tracking exists.

Set-point tracking takes care that in switch-over of Wext → W, the past Wext is taken over as int. set-point W.

When shifting back (W → Wext) Wext is started with the attitude of Grw+/-. If the controller shall follow process value or set-point tracking is determined with the configurationword Wtrac. Tracking can be activated via interface or digital input track .Tracking is evaluated with priority.

Fig.: 15 Set-point tracking with switch-over to internal set-point



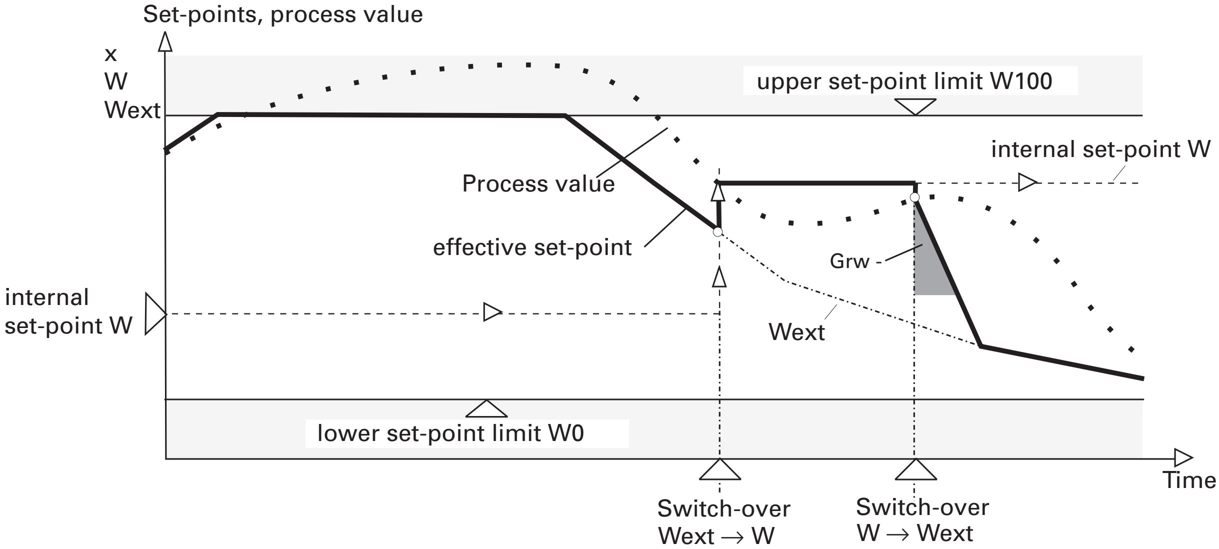
If in a place (interface or digital input) is switched to tracking, switching in another place is not possible!

Process-value tracking

It can occur that the set-point is far distant from the momentary process-value (e.g. when starting a plant).
In order to prevent the jump developing here, the function process-value tracking can be used. Process-value tracking causes a take-over of the process-value on the internal set-point, when changing over Wext → W. When shifting (W back → Wext) Wext is started with the attitude of Grw+/- (see → Figure below).
If the controller shall follow process value or set-point tracking is determined with the configurationword Wtrac. Tracking can be activated via interface or digital input track .Tracking is evaluated with priority.

If in a place (interface or digital input) is switched to tracking, switching in another place is not possible!

Fig.: 16 Process-value tracking when changing-over to the internal set-point W



Setpoint and correcting variable behaviour after setpoint switch-over

After setpoint and correcting variable switch-over, control behaviour or start-up behaviour has priority. The PID characteristic must be partly suppressed. The previous history which is important for the integral action and especially for the derivative action is largely insignificant with setpoint change due to the new target setpoint.

Switch-over operations which might affect the control behaviour are:

1	Manual -> automatic	Switch-over from manual to automatic mode
2	Off -> start-up	Start-up after off-line (power failurel/configuring)
3	W_{alt} -> W_{neu}	Setpoint change
4	W -> W2	Switch-over to 2nd setpoint
5	W2 -> W	Switch-over from 2nd setpoint to normal setpoint
6	We -> Wi, without tracking	Switch-over from external to internal setpoint without tracking
7	Wi -> We	Switch-over from internal to external setpoint
8	We -> Wi with tracking	Switch-over from external to internal setpoint with tracking

The approach to a new setpoint may be affected by further parameters. Parameters Grw+ (positive setpoint gradient), Grw- (negative setpoint gradient) and Grw2 (setpoint gradient during the approach to W2) can be used for gradual approach to a new target setpoint via a ramp function. Unless a gradient is defined (Grw = off), approach to the new setpoint starts with a step change at the previous setpoint or at the actual process value. To influence the correcting variable when switching over, any after-effect of the derivative action is eliminated internally or the integral action is adapted to avoid correcting variable bumps. The following table gives a survey of the controller switch-over behaviour implemented from operating version 8.

Controller-internal operations during switch-over with CONTR, CONTR+ and PIDMA

Switch-over	Without gradient function	With gradient function
1	After correcting variable adaptation with deletion of a still effective derivative action, the approach to the setpoint is bumpless	The effective setpoint ramp continues running in the background during manual mode. After switching over to automatic, the correcting variableis adapted and the derivative action is deleted and the setpoint is set to the actually reached ramp setpoint (bumpless).
2	The effective setpoint is set to the process value first and after deleting a still effective derivative action, a setpoint step change to the target is made. During this step change, the PID parameters are effective. The derivative action is a result of the step change (not bumpless).	At first, the effective setpoint is set to the process value. After deleting the derivative action, the approach to the target setpoint is via a ramp. During this transition, the PID parameters are effective (bumpless starting with 0).
3	After deleting a still effective derivative action, a step change from the instantaneous to the target setpoint is made. During this step change, the PID parameters are effective. The derivative action is a result of the new step change (not bumpless).	After deleting the derivative action and adapting the correcting variable, changing from the old to the new target setpoint is done via a ramp (bumpless) .
4, 5, 6, 7	After deleting a remaining derivative action, a setpoint step change from the instantaneous to the target setpoint is made. During this step change, the PID parameters are effective. The derivative action is only a result of the new step change (not bumpless).	The effective setpoint is set to the process value. After deleting the derivative action and adapting the correcting variable, setpoint changing from the process value to the target setpoint is done via a ramp (bumpless).
8	The internal target setpoint is set to the actual process value or to the external setpoint. Subsequently, any still effective derivative action is deleted and the correcting variable is adapted (bumpless).	The internal target setpoint is set to the actual process value or to the external setpoint. Subsequently, any still effective derivative action is deleted and the correcting variable is adapted (bumpless).

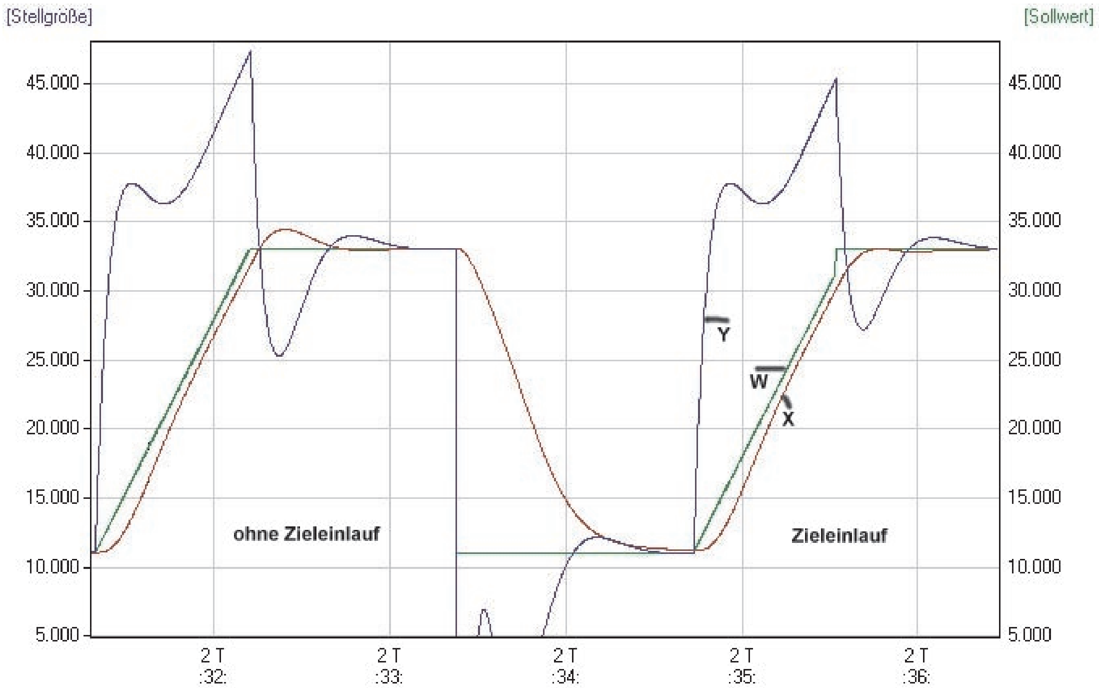
Gentle line-out to the target setpoint with ramps

When using a setpoint ramp, a process value overshoot at the ramp end may occur. Due to the difference between setpoint and process value in the course of the ramp, an integral action is built up and must be removed after the end of the ramp. The longer the ramp, the higher the integral action. And the more exact the process value follows the setpoint, the higher the probability that any integral action will cause an overshoot.

The target line-out function is used to adapt the integral action to the actual PD action at an adjustable distance before reaching the ramp end value, the D-dynamics is initialized and the setpoint is set to the ramp end value. Now the controller dynamics re-starts bumplessly related to the new setpoint.

Controller parameter “a” can be used to define at which distance to the final setpoint the target orientation should be switched over to the final setpoint. The target line-out function is activated under the following conditions :

1. W < Wend
2. W > Wend-2a
3. X > Wend-a



Marginal conditions / restrictions:

With internal setpoint ramps, the controller knows the future target setpoint. When using external setpoints with ramp function (programmer), the ramp end value must be bound to input X3 of the controller block. When the internal ramp is active, line-out to the target setpoint is always related to the internal ramp end value, and the value at X3 is ineffective.

Target line-out is activated only, if the external ramp setpoint changes continuously.

The function can be used both with differentiation of control deviation (XW) and differentiation of process value (X).

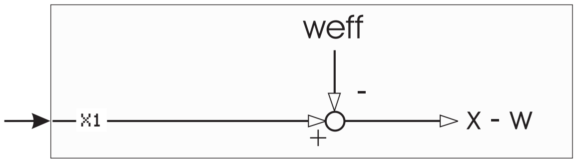
With 3-element control, target line-out is omitted. The signification of parameter “a” is different and connection of an external end setpoint is not possible.

With ratio control, target line-out is only restricted with fixed distance (1 in units of the physical quantity). The signification of parameter a is different.

24.12 Process value calculation

Standard controller

The process variable measured via analog input X1 is used as process value by the controller.



Ratio controller

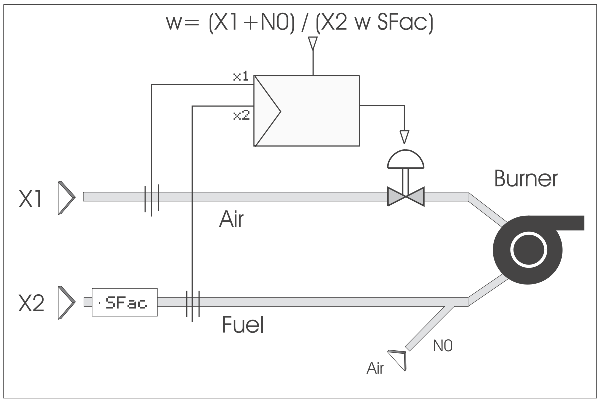
Process control frequently requires various components to be mixed into a product. These components must be mixed according to a given ratio. The main component is measured and used as reference for the other components. With increasing flow of the main component, the flow of the other components will increase accordingly. This means that process value x used by the controller is determined by the ratio of two input variables rather than being measured as one process variable.
For optimum combustion, the fuel-air ratio must be controlled. With stoichiometric combustion, the ratio is selected so that there are no inflammable residues in the waste gas. In this case, the relative rather than the physical ratio is displayed as process value and adjusted as l.
If the transmitters used by the controller are designed with a stoichiometric ratio, = 1 is met exactly with restless combustion. With a process value display of 1,05, the instantaneous air excess is clearly 5%. The amount of air required for atomizing is taken into account by constant 'N0'. For selecting a ratio controller, CType = Ratio must be selected. Moreover, configuration word 'Ratio' must be taken into account.

With ratio controller,
note that parameters Xn0 and Xn100 must be set to the input range of connector X1.

Example of standard ratio control:

Standard ratio control at the example of stoichiometric combustion. Analog input INP1 is configured to 4...20 mA with physical unit m^3/h (air). Values 0 and 100 are allocated to input variables 4 mA (x0) and 20 mA (x100). Atomizing air N0 is added to this input.
E.g. INP5 is selected as second ratio input. This input is also configured for 4...20 mA and m^3 /h (gas). x0 and x100 values 0 and 100 are allocated to the input variables.

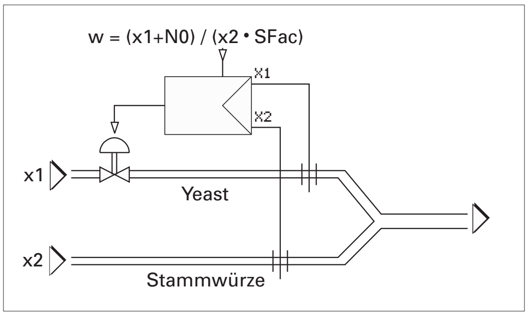
Set-point Weff effective as relative ratio is multiplied by stoichiometric factor SFac (e.g. SFac = 10), i.e. a „stoichiometric" flow ratio can be used for calculation of the control deviation. The instantaneous (controlled) process value is calculated from the physical ratio, multiplied by 1/SFac and displayed as relative value.



Material batching and mixing

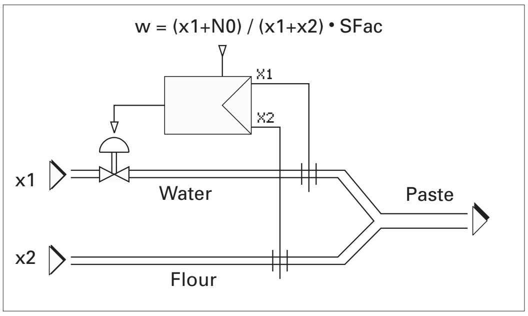
The following examples are intended to show that various control possibilities can be used. This is necessary, since the materials to be mixed (e.g. paste) are not always directly measurable due to their consistency. Other cases may require a component to be controlled in relation to a total rather than to another component.

`Ratio` = `Type 1` $W = \frac{X1 + N0}{X2 \cdot SFact}$



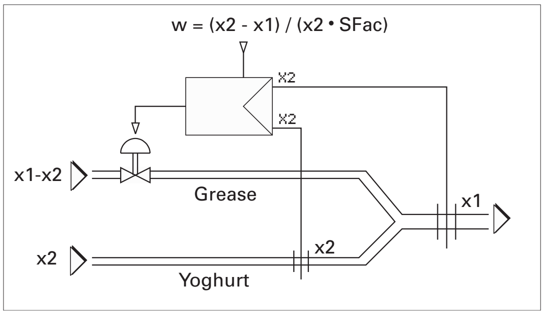
The first case is obvious. Almost everybody knows what happens during brewing.
Yeast (x1) must be batched in relation to the original wort (x2). The set-point is adjusted in '%yeast', e.g. W= 3%. The ratio inputs are scaled in equal units. The control deviation is multiplied by '`SFac` = 0,01' and calculated according to equation xw = (x1 + N0) - 0,03 w x2, so that exactly 3% of yeast are batched with xw = 0. Process value display is also in %. Constant `N0` is without importance (`N0` = 0)

`Ratio` = `Type 2` $W = \frac{X1 + N0}{(X1 + X2) \cdot SFact}$



In this example, water (x1) must be batched as a percentage of the total (paste; x1+x2). As the paste quantity is not available directly as a measurement signal, the total is calculated internally from x1 and x2. N0 = 0 must also be adjusted in this case.

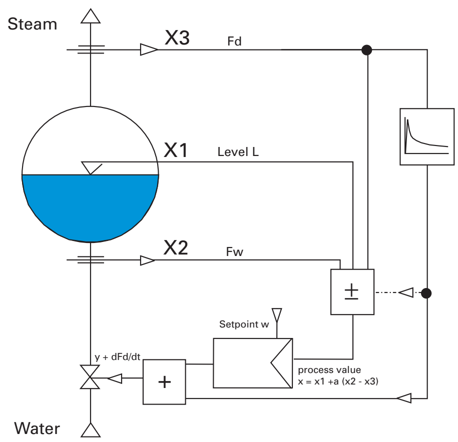
`Ratio` = `Type 3` $W = \frac{X2 - X1 + N0}{X2 \cdot SFact}$



Unlike the previous examples, yoghurt (x2) and the final product (x1) are measured in this case.

Three-element control

With three-element control, process value calculation is according to equation $x_{eff} = X1 + a \cdot (X2 \backslash X3)$ whereby term $(X2 \backslash X3)$ is the difference between the steam and water flow rates. Factor b for flow range matching used so far is omitted, because the mA signals are converted directly into physical units during input value conditioning (x0, x100). The calculated process value is displayed on the process value display.

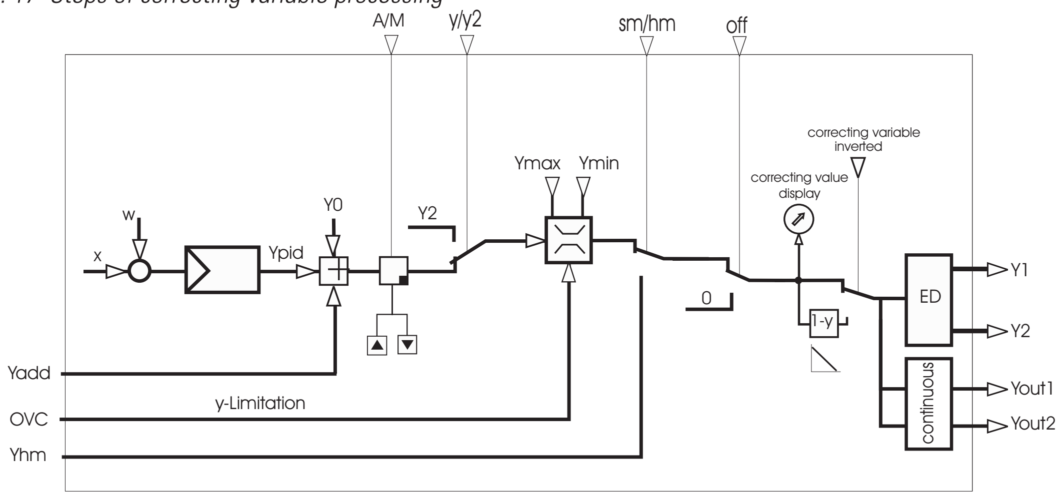


For selecting a three-element controller, '`CFunc` = `3-elem.`' must be entered in the configuration.

Correcting variable processing

The following considerations related to correcting variable processing are applicable to continuous controllers, two-point, three-point and three-point stepping controllers with position feedback. The diagram opposite shows the functions and interactions of correcting variable processing.

Fig.: 17 Steps of correcting variable processing



Second correcting value

Similar to set-point processing, switch-over to a second preset correcting value Y2 is possible. Switching over is done via digital input y/y2.
Whether Y2 has safety functions, or whether it is only a pre-defined start position in defined process conditions is determined only by the use and integration into an automation concept.

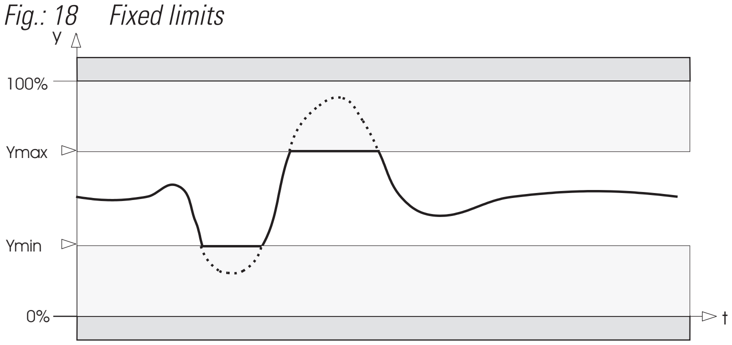
Second correcting value Y2 is evaluated with priority. When switching over to Y2 is done at one point (interface or digital control input 'y/y2'), switching over at the other point is not possible.

Correcting variable limits

Parameters Ymin and Ymax determine the limits of the correcting variable range within 0...100 %. With three-point and continuous controller „split range", the correcting variable limits are within -100 ... +100 %.

Parameters Ymin and Ymax are used to specify fixed correcting variable limits.

Fig.: 18 Fixed limits



External correcting variable limiting (override control)

Dependent of 'COVC' setting, the lowest (OVC−), the highest (OVC+) or lowest and highest correcting value (OVC+/OVC−) can be limited by analog input signals.

Override control is used where bumpless switch-over to another controller when reaching defined process conditions and mainly according to other criteria is required. The basic principle is that two controllers act on the same motor actuator.

Fig.:19 Maximum limit

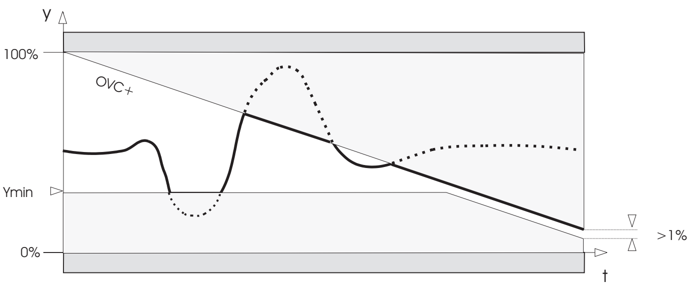


Fig.:20 Minimum limit

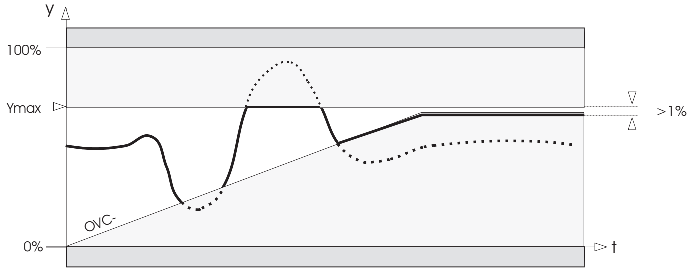
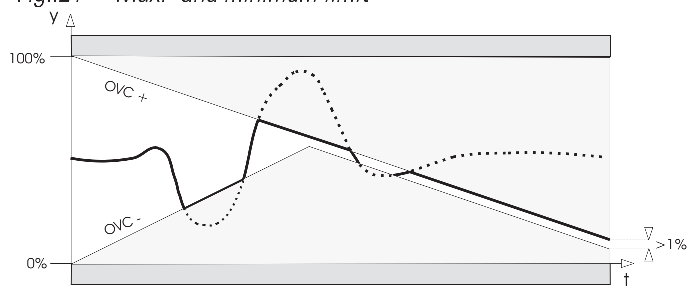


Fig.:21 Maxi- and minimum limit



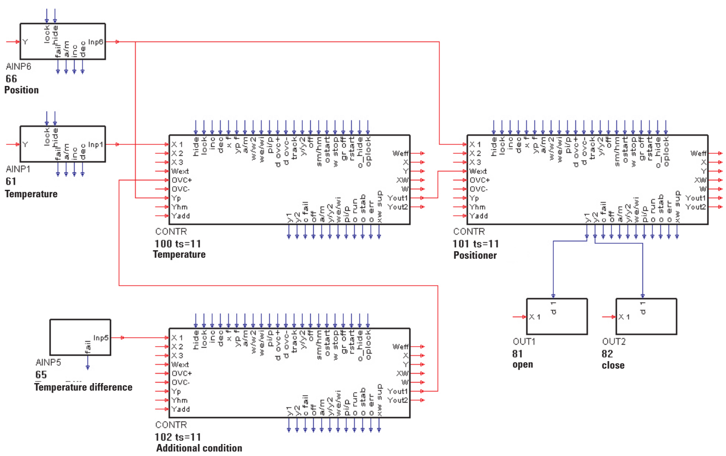
Override control

Override control with continuous output

Override control with three-point stepping output can be realized by a continuous controller with the OVC function.

The correcting variable defined by the continuous controller is adjusted by a position controller (three-point stepping).

Fig.: 22 Override control with continuous controller

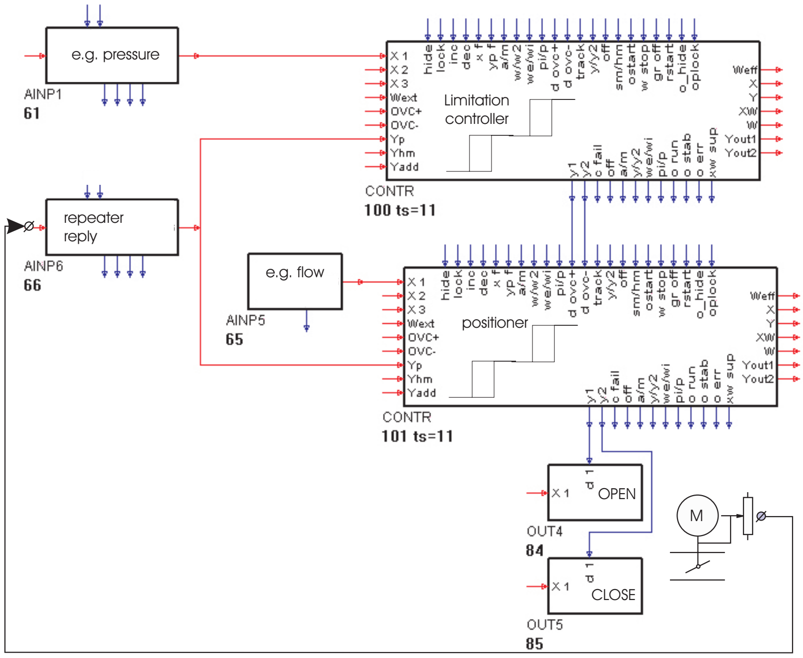


Override (limiting) control using a three-point stepping output

Override control is also possible by means of a classical three-point stepping controller. The positioning signals of the limiting controller must be connected as shown in the example Fig.: 23.

Which one of the two controllers influences the process is decided in the slave controller logic. The first "close" pulse coming from the limiting controller switches over to override control. The limited controller automatically retrieves the positioning authority, when it first tries to close the actuator further.

Fig.:23 Override control using 3-point stepping controllers



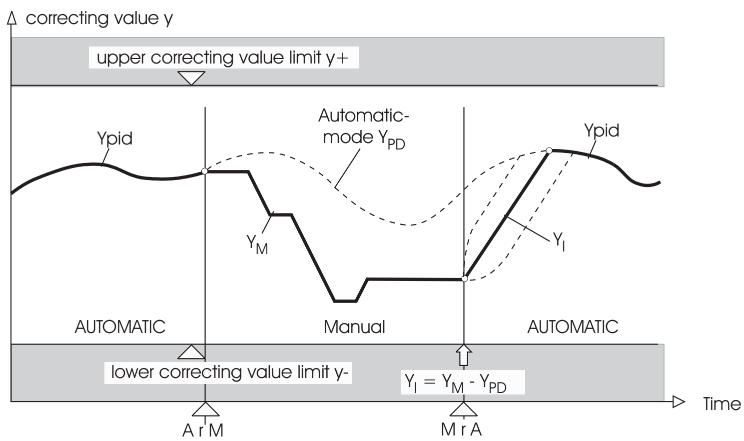
Bumpless auto/manual switch-over

Sudden process interventions by control mode switch-over are usually not desired. Excepted is purposeful switch-over y → Y2.

A → M switch-over is always bumpless; the last correcting value is frozen and can be changed manually. M → A switch-over is different. Correcting value differences are compensated as follows: when switching over, the controller integral action is set to correcting value Y_M output last plus correcting variable portions of the controller P and D action running in the background

($Y_I = Y_M + Y_{PD}$). Now, only the integrator, which adapts the correcting variable slowly to the stationary value according to the actual control deviation, is active. Until the D action has decayed completely, the adaptation can be delayed or accelerated

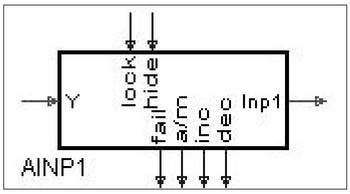
Fig.:24 Bumpless switchover

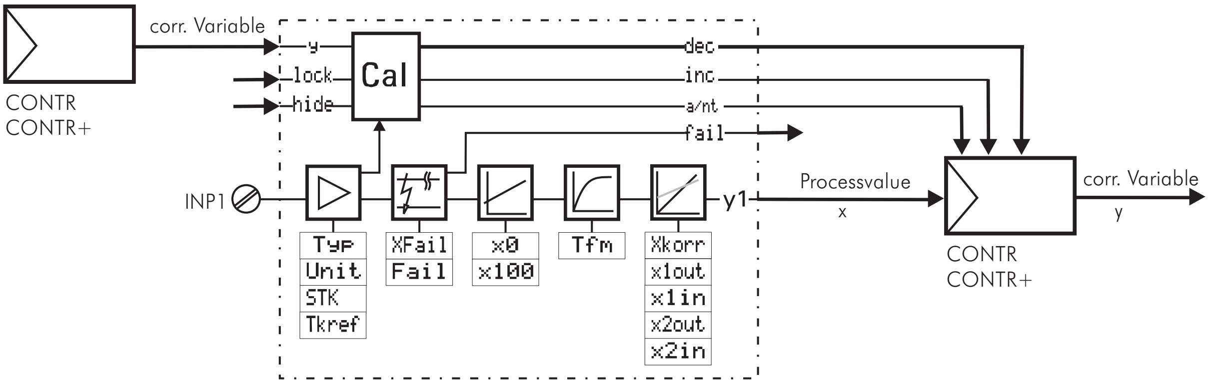


25 Inputs

25.1 AINP1 (analog input 1)

For direct connection of temperature sensors, for potentiometric transducers and standard signals





General

Function ‘AINP1’ is used for configuration and parameter setting of analog input INP1. It is firmly allocated to block number 61 and is calculated firmly in each time slot. The function provides a corrected measurement value and a measurement value status signal at its outputs.

Inputs/outputs

Digital inputs:	
lock	Calibration locked (with lock = 1 calibration is locked)
hide	Display suppression (with hide = 1 the calibration page is not displayed)

Digital outputs:	
fail	Signals an input error (short-circuit, polarity error, ..)
a/m	Manual signal
inc	Increment signal
dec	Decrement signal

Analog inputs:	
Y	Output variable

Analog outputs:	
Inp1	Signal input

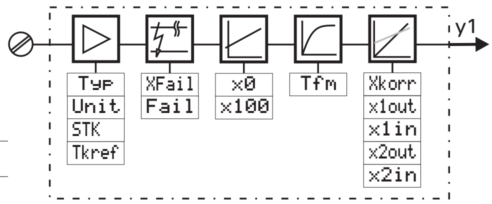
Parameter and configuration data

Parameter	Description	Values	Default
x1in	Measured value correction P1, input	-29999 ... 999 999	0
x1out	Measured value correction P1, output	-29999 ... 999 999	0
x2in	Measured value correction P2, input	-29999 ... 999 999	100
x2out	Measured value correction P2, output	-29999 ... 200000	100

Configuration	Description	Values	Default
Type	Type L -200...900 °C	Type L	
	Type J -200...900 °C	Type J	
	Type K -200...1350 °C	Type K	
	Type N -200...1300 °C	Type N	
	Type S -50...1760 °C	Type S	
	Type R -50...1760 °C	Type R	
	Type T -200...400 °C	Type T	
	Type W 0...2300 °C	Type W	
	Type E -200...900 °C	Type E	
	Type B 0...1820 °C	Type B	
	Pt 100 -99,9...850,0 °C	Pt100 850	
	Pt 100 -99,9...250,0 °C	Pt100 250	
	2x Pt 100 -99,9...850 °C	2Pt100 85	
	2x Pt 100 -99,9...250,0 °C	2Pt100 25	
	0...20 mA	0...20mA	←
	4...20 mA	4...20mA	
	0...10 V	0...10V	
	2...10 V	2...10V	
	Transducer 0...500 Ω	Pot.trans	
	Resistance 0...500 Ω linear	0..500Ohm	
	Resistance 0...250 Ω linear	0..250Ohm	
Fail	Fail function off	disabled	
	Digital output fail = 1, y1 = x100	Upscale	←
	Digital output fail = 1, y1 = x0	Downscale	
	Digital output fail = 1, y1 = XFail	Subst.val.	
Xkorr	Measured value correction off	off	←
	Measured value correction adjustable	on	
Unit	Unit = °C — only effective with thermocouple and Pt100 setting	°C	←
	Unit = °F	°F	
STK	Internal temperature compens. — only effective with thermocouple	int.CJC	←
	External temperature compens.	ext.CJC	
x0	Physical value at 0% — only effective with standard signals (0/4..20mA or 0/2..10V)	-29999 ... 999 999	0
x100	Physical value at 100%	-29999 ... 999 999	100
XFail	Substitute value with sensor error	-29999 ... 999 999	0
Tfm	Filtertime constant [s]	0 ... 999 999	0,5
Tkref	Reference temperature at STK = ext.TK	0 ... 140	0

Measured value conditioning

Before the pre-filtered (time constant ...; limiting frequency ...) analog input signals are available as digitized measurement values with physical quantiy, they are subjected to extensive measured value conditioning.



Input circuit monitor

- ❑ Thermocouples
  The input circuit monitor monitors the thermocouple for break and polarity error. An error is determined if the measured thermovoltage signals a value which is by more than 30 K below the span start.

- ❑ Pt100 measurements and transducers are monitored for break and short-circuit.

- ❑ Current and voltage signals
  With current (0/4...20mA) and voltage signals (0/2...10V), monitoring for out-of-range (I > 21,5 mA or U > 10,75 V) and for short circuit (I < 2 mA or U < 1 V) with “life zero” signals is provided.

Sensor errors are output as digital output (fail). In case of error, statuses ‘Upscale’, ‘Downscale’ or ‘Subst.val.’ defined in the configuration can be preset for the input circuit.

Linearization

Thermocouples and Pt100 are always measured over the overall physical measuring range according to data sheet and linearized according to the allocation table. Linearization is realized with up to 28 segment points by error curve approximation.

Scaling

mA and V standard signals are always scaled according to the physical measuring range of the transmitter (x0, x100).

With transducer measurements, “calibration” is according to the proven method. Bring the transducer to start and then to end position and “calibrate” it to 0 % or 100 % by key pressure. In principle, calibration corresponds to a scaling, whereby gradient and zero offset are calculateed automatically by the firmware.

Additional measurements

Dependent of configured sensor type, additional and corrective measurements are required.

The amplifier zero is checked with all measurement types and included into the measurement value. The lead resistances with Pt100 and transducer, and the cold-junction reference temperature (internal TC) are measured additionally.

Filter

A 1st order filter is adjustable in addition to the analog part filtering of each input signal.

Sampling intervals

The sampling interval for the INP1 is 200ms.

Linearization error

Thermocouples and Pt100 are linearized over the overall physical measuring range. Linearization is with up to 28 segments, which are placed optimally on the error curve by a computer program and thus compensate the linearity errors. As error curve approximation is provided only by segments (polygons) rather than by an nth order polynomial, there are points on the characteristic in which the residual error is zero. Between these “zero points”, however, the residual error has very small, but measurable values. For the reproducibility, however, this error is irrelevant, because it would repeat itself in exactly the same point and amount, if the measurement would be repeated under identical conditions.

Temperature compensation

Measurement of the cold-junction reference temperature is using a PTC resistor. The temperature error thus determined is converted into mV of the relevant thermocouple type, linearized and added to the measured value as corrective value with correct polarity. The remaining error with varying cold-junction reference temperature is approx. 0,5K/10K, i.e. about one tenth of the error which would occur without compensation. Better results are possible

with controlled external TC, which is adjustable within 0...+140°C at the cold junction reference dependent of controlled temperature. With cold-junction reference measurements for “reproducibility” assessment, however, utmost care must be taken that constant environmental conditions are not exceeded when working with internal TC. An air draft at the PTC resistor of the cold junction reference can be sufficient to falsify the measurement result.

Measured value correction

The measurement can be corrected in various ways using the measured value correction.

Pre-requisite: configuration XKorr = ein

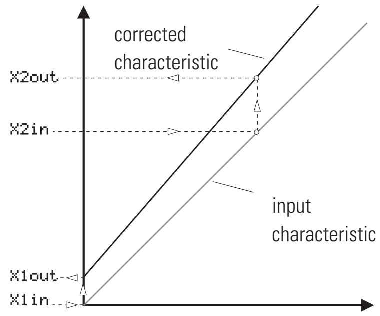
In most cases, the relative accuracy and reproducibility rather than the absolute one are of interest, e.g.:

-the compensation of measurement errors in one working point (set-point control)
-minimization of linearity errors in a limited operating range (variable set-point)
-correspondence with other measuring facilities (recorders, indicators, PLCs, ...)
-compensation of sample differences of sensors, transmitters, etc.

Measured value correction is designed for zero offset, gain matching and for both. It corresponds to scaling mx+b, with the difference that the KS98 firmware calculates gain m and zero offset b from the value pairs for process value (x1in; x2in) and set-point (x1out; x2out) of two reference points.

Example 1:
Zero offset
x1in = 100°C x1out = 100°C + 1,5°C
x2in = 300°C x2out = 300°C + 1,5°C

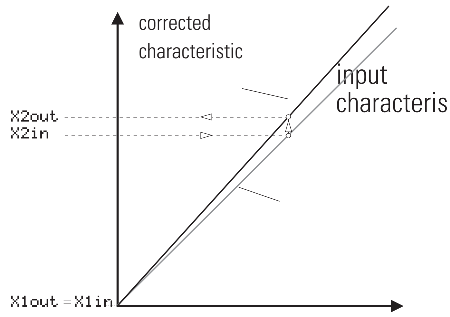
The corrected values are shifted evenly with reference to the input values over the complete range.



Example 2:
Gain change (rotation around the coordinate origin)

x1in = 0°C x1out = 0°C
x2in = 300°C x2out = 300°C + 1,5°C

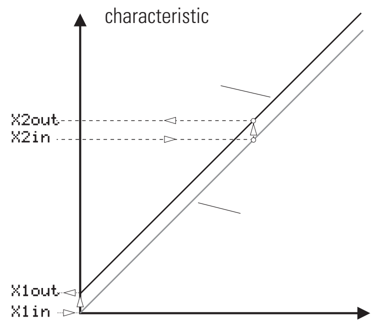
The corrected values diverge despite equality with the input values at x1in and x1out.



Example 3:
zero and gain matching

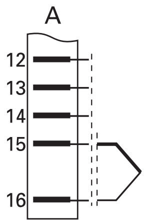
x1in = 100°C x1out = 100°C - 2,0°C
x2in = 300°C x2out = 300°C + 1,5°C

The corrected values are shifted already at values x1in and x1out and diverge additionally.



Sensor types

The input sensor type can be determined as thermocouple, resistance thermometer, potentiometric transducer or as standard signal (current and voltage). The physical quantity is freely selectable.



Input thermocouple

The following thermocouple types are configurable as standard:

Type L, J, K, N, S, R, T, W, E and B according to IEC584.

The signal behaviour can be affected by the configuration of the following points. Distinction between internal and external temperature compensation (→ STK) is made.

- Internal compensation:
  the compensating lead must be taken up to the multi-function unit connecting terminals. No lead resistance adjustment is required.
- External temperature compensation:
  A separate cold-junction reference with a fixed reference temperature must be used (between 0 and 140°C configurable) (→ Tkref)
  The compensating lead must be taken only up to the cold-junction reference, the cable between reference and multi-function unit terminals can be of copper. No lead resistance adjustment is necessary.
- The action of the built-in TC break protection can be configured for upscale (set-point << process value) or downscale (set-point >> process value) or fixed to a substitute value (→ Fail).
- For measured value processing, a filter time constant with a numeric value between 0,0 and 200000 is adjustable (→ Tfm).
- A process value correction is configurable (→ Xkorr).

Resistance thermometer input

Resistance thermometer, temperature difference

With a resistance thermometer, the signal behaviour with sensor break can be determined (→ Fail). No temperature compensation is required and is therefore switched off. With temperature difference measurement, calibration by means of short circuit is required. If lead resistance adjustment is necessary, it can be realized by means of a 10 calibrating resistor (order no. 9404 209 10101). Dependent of signal source type, the unit is configured for one of the following inputs:

- resistance thermometer Pt 100 with linearization
- temperature difference with 2 x Pt 100 and linearization
- linear potentiometric transducer

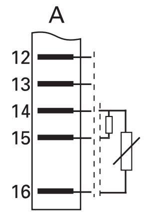
For measured value processing, a filter time constant with a numeric value within 0 and 999 999 can be adjusted (→ Tfm). Process value correction can be configured (→ Xkorr).

Resistance thermometer Pt 100

The two ranges -200,0...+250,0 °C and -200,0...+850,0 °C are configurable (→ type).

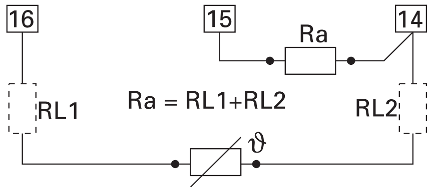
Connection in two or three-wire circuit is possible. Copper lead must be used for measurement. The input circuit monitoring responds at -130°C (sensor or lead break). The action is configurable for:

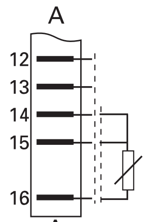
Upscale (set-point << process value)
Downscale (set-point >> process value)
Substitute value (the entered value is used as measured value in case of failure.



Resistance thermometer in 2-wire connection:

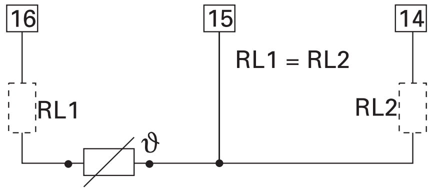
For lead resistance adjustment, disconnect the input leads from the multi-function unit terminals and short circuit them in the connecting head of the resistance thermometer. Measure the lead resistance by means of a resistance bridge and change lead adjusting resistor (Ra) so that its value is equal.

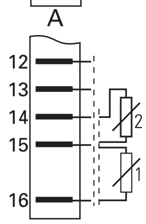




Resistance thermometer in 3-wire connection:

The resistance of each input lead must not exceed 30 Ω. No lead resistance adjustment is necessary, provided that the resistances of the input leads R_a are equal. If necessary, they must be equalized by means of a calibrating resistor.





Temperature difference 2 x Pt100

Range 850°C: X0 = -950°C; X100 = 950°C (Type = 2Pt100 85)

Range 250°C: X0 = -250°C; X100 = 250°C (Type = 2Pt100 25)

For lead resistance adjustment, the two thermocouples must be short-circuited in the connecting head. Select calibration according to Fig.: 25. With blinking Set Dif, wait until the input has settled (minimum 6 s). Press ⏎ → Cal done is displayed → Lead resistance adjustment is finished. Remove the two short-circuits.

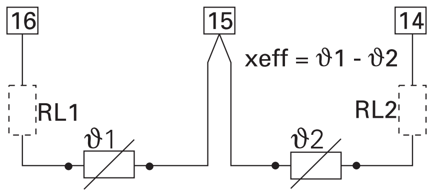
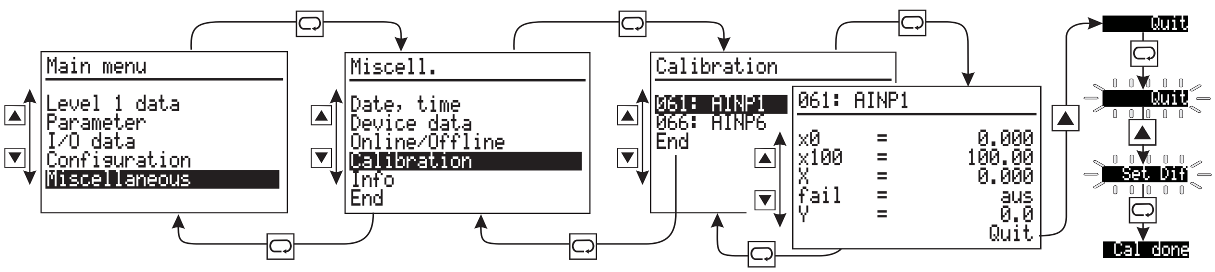


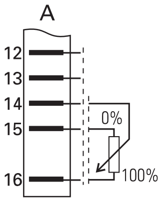
Fig.: 25 Selection of calibrating page



Potentiometric transducer

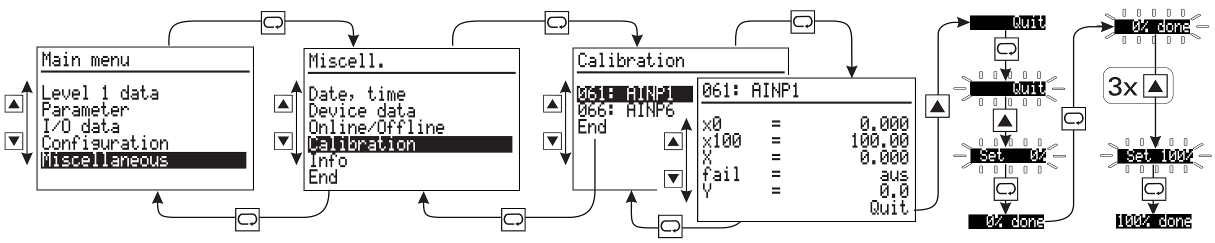
Overall resistance ≤ 500 Ω incl. 2 • RL.
Calibration or scaling is with the sensor connected.

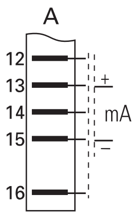
☞ **Before calibration, the mains frequency required for operation must be adjusted.**
Main menu → `Miscellaneous` → `Device data` → `Freque`.



Calibration is as follows.
Potentiometric transducer calibration is possible via interface and front panel operation.
Select `set 0%` as shown in Fig.: 26. After pressing the selector key, `set 0%` starts blinking. Bring the transducer into the position pertaining to X_0 (mostly lower end position). The instantaneously valid value for INP1 is displayed '`X`'. Press the selector key again to store this actual value as X_0.
Select `set 100%`. After pressing the selector key, `set 100%` starts blinking. Bring the transducer into the position pertaining to X_{100} (mostly upper end position). The instantaneously valid value for INP1 is displayed '`X`'. After pressing the selector key again, this actual value is stored as X_{100}.

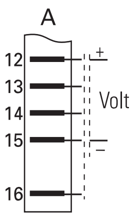
Fig.: 26 Selection of calibrating page





Standard 0/4...20 mA current input

The input resistance is 50 Ω
During configuration, distinction of 0...20 mA and 4...20 mA is made. For 4 ... 20 mA standard signal, the signal behaviour with sensor break can be determined (`Fail`). Additionally, physical input signal scaling using a defined value of `X0` and `X100` is possible. For measured value processing, a filter time constant with a numeric value between 0,0 and 200000 can be adjusted (→ `Tfm`)

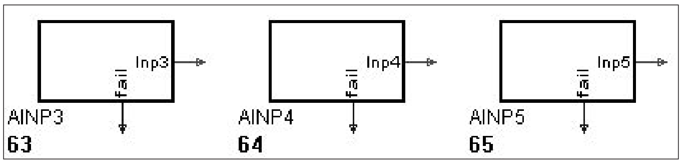


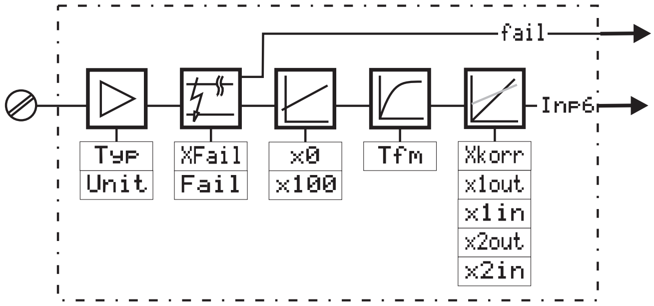
0/2...10 V voltage signal input

The input resistance is ≥ 100 kΩ
During configuration, distinction of 0...10 V and 2...10 V is made. For 2 ... 10 V standard signal, the signal behaviour with sensor break can be determined (`Fail`). Additionally, physical input signal scaling with a defined value of `X0` and `X100` is possible. For measured value processsing, a filter time constant with a numeric value within 0,0 and 200000 can be adjusted (→ `Tfm`)

25.2 AINP3...AINP5 (analog inputs 3...5)

For standard signal connection





General

Functions 'AINP3...AINP5' are used for configuration and parameter setting of analog inputs INP3...INP5. They are firmly allocated to block number 63...65 and are calculated in each time slot. The functions provide corrected measurement values and measurement value statuses at their outputs.

Inputs / outputs

Digital outputs:	
fail	Signals an input error (short-circuit, wrong polarity, ..)

Analog outputs:	
Inp1	Signal input

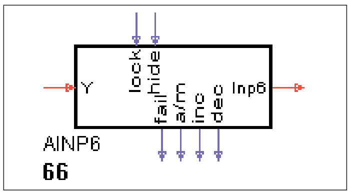
Parameter and configuration data

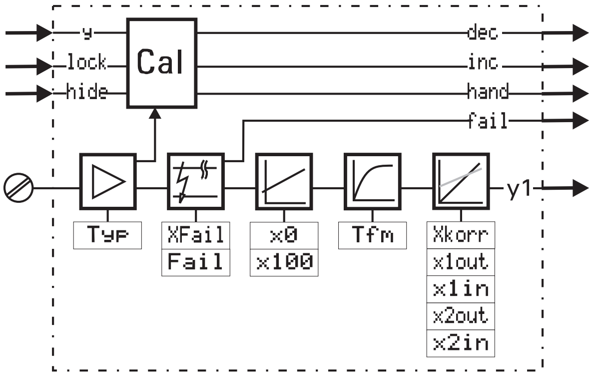
Parameter	Description	Values	Default
x1in	Measured value correction P1, input	-29999 ... 999 999	0
x1out	Measured value correction P1, output	-29999 ... 999 999	0
x2in	Measured value correction P2, input	-29999 ... 999 999	100
x2out	Measured value correction P2, output	-29999 ... 999 999	100

Configuration	Description	Values	Default
Type	0...20 mA 4...20 mA 0...10 V 2...10 V	0...20mA 4...20mA 0...10V 2...10V	←
Fail	Fail function off Digital output fail = 1, y1 = x100 Digital output fail = 1, y1 = x0 Digital output fail = 1, y1 = XFail	disabled Upscale Downscale Subst.val.	←
Xkorr	Measured value correction off Measured value correction adjustable	off on	←
x0 x100	Physical value at 0% Physical value at 100% — effective only with standard signals (0/4..20mA or 0/2..10V)	-29999 ... 999 999 -29999 ... 999 999	0 100
Tfm	Filter time constant [s]	0 ... 999 999	0,5

25.3 AINP6 (analog input 6)

For direct connection of potentiometric transducer and standard signal





General

Function 'AINP6' is used for configuration and parameter setting of analog input INP6. It is firmly allocated to block number 66 and is calculated in each time slot. The function provides a corrected measurement value and a measured value status signal at its outputs.

Inputs / outputs

Digital inputs:	
lock	Calibration disabled (with **lock** = 1 calibration is disabled)
hide	Display suppression (with **hide** = 1 the calibration page is disabled)

Digital outputs:	
fail	Signals an input error (short circuit, wrong polarity, ..)
a/m	Manual signal
inc	Increment signal
dec	Decrement signal

Analog inputs:	
Y	Output variable

Analog outputs:	
Inp1	Signal input

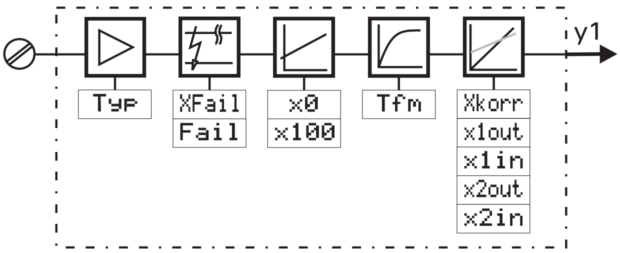
Parameter and configuration data

Parameter	Description	Values	Default
x1in	Measured value correction P1, input	-29999 ... 999 999	0
x1out	Measured value correction P1, output	-29999 ... 999 999	0
x2in	Measured value correction P2, input	-29999 ... 999 999	100
x2out	Measured value correction P2, output	-29999 ... 999 999	100

Configuration	Description	Values	Default
Type	0...20 mA 4...20 mA Transducer 0...1000 Ω	0...20mA 4...20mA Pot.trans.	←
Fail	Fail function off Digital output fail = 1, y1 = x100 Digital output fail = 1, y1 = x0 Digital output fail = 1, y1 = XFail	disabled Upscale Downscale Subst.val.	←
Xkorr	Measured value correction off Measured value correction adjustable	off on	←
x0 x100	Physical value at 0% only effective with standard signals Physical value at 100% (0/4..20mA or 0/2..10V)	-29999 ... 999 999 -29999 ... 999 999	0 100
XFail	Substitute value with sensor error	-29999 ... 999 999	0
Tfm	Filter time constant [s]	0 ... 999 999	0,5

Input value conditioning

Before the pre-filtered (time constant ...; limiting frequency ...) analog input signals are available as digitized measurement values with physical quantity, they are subjected to extensive input value processing.



Input circuit monitor

- Transducers are monitored for break and short circuit.
- Current signals Out-of-range monitoring (I > 21,5 mA) with current signals (0/4...20 mA) and short-circuit monitoring (I < 2 mA) with "life zero" signals are provided.

Sensor errors are output as digital output (fail). In case of error, statuses 'Upscale', 'Downscale' or 'Subst.val' defined in the configuration (Fail) can be used for the input circuit.

Scaling

The mA standard signals are scaled according the the physical measuring range of the transmitter (x0, x100). With potentiometric transducer measurements, "calibration" is according to the proven method. Bring the transducer to start and then to end position and "calibrate" to 0 % or 100 % by key pressure. The calibration principle corresponds to scaling, whereby gradient and zero offset are calculated automatically by the firmware.

Filter

A 1st order filter is adjustable in addition to filtering in the analog part of each input signal.

Sampling intervals

The sampling interval for INP6 is 400ms.

Measured value correction

Measured value correction can be used for various types of measurement correction.

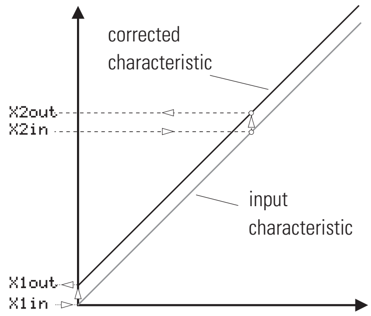
Pre-requisite: configuration XKorr = ein
In most cases, the relative rather than the absolute accuracy and reproducibility are of interest, e.g.:

-measurement error compensation in a working point (set-point control)
-minimization of linearity errors within a limited operating range (variable set-point)
-correspondence with other measuring facilities (recorders, indicators, PLCs, ...)
-compensation of sensor, transmitter, etc. sample differences.

The measured value correction is designed both for zero offset, gain matching and for both. It corresponds to scaling mx+b, with the difference that the KS98 firmware calculates gain m and zero offset b from the defined value pairs for process value (x1in; x2in) and set-point (x1out; x2out) of two reference points.

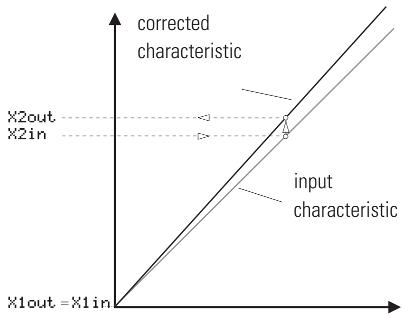
Example 1:
Zero offset
x1in = 100 x1out = 100 + 1,5
x2in = 300 x2out = 300 + 1,5

The corrected values are shifted evenly with reference to the input values over the complete range.



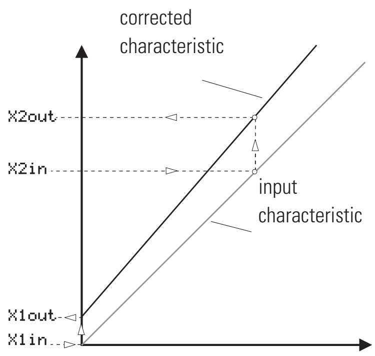
Example 2:
Gain change (rotation around the coordinate origin)
x1in = 0 x1out = 0
x2in = 300 x2out = 300 + 1,5

The corrected values diverge despite equality with the input values at x1in and x1out.



Example 3:
Zero and gain matching
x1in = 100 x1out = 100 - 2,0
x2in = 300 x2out = 300 + 1,5

The corrected values are already shifted at input values x1in and x1out and diverge additionally.



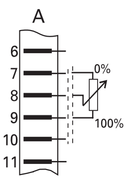
Sensor types

The input sensor type can be defined as potentiometric transducer or as standard current signal.

Potentiometric transducer

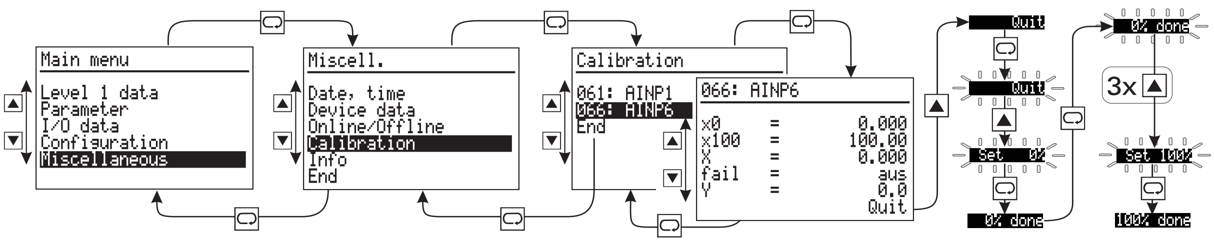
Overall resistance ≤ 1000 Ω incl. 2 • RL.
Calibration or scaling are with the sensor connected.

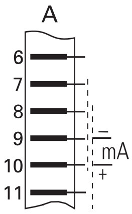
☞ **Before calibration, the mains frequency required during operation must be adjusted.**
Main menu → Miscellaneous → Device data → Freque.



Calibration is as follows.
Transducer calibration is possible via interface and front-panel operation.
Select set 0% as shown in Fig.: 26. After pressing the selector key, set 0% starts blinking. Bring the transducer into the position corresponding to X0 (usually lower end position). The instantaneously valid value for INP6 is displayed 'X'. Press the selector key again to store this actual value as X_0.
Select set 100%. After pressing the selector key set 100% starts blinking. Now, bring the transducer into the position corresponding to X_{100} (mostly upper end position). The instantaneously valid value for INP6 is displayed 'X'. Press the selector key again to store this actual value as X_{100}.

Fig .: 27 Selection of calibrating page

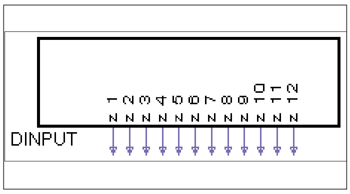


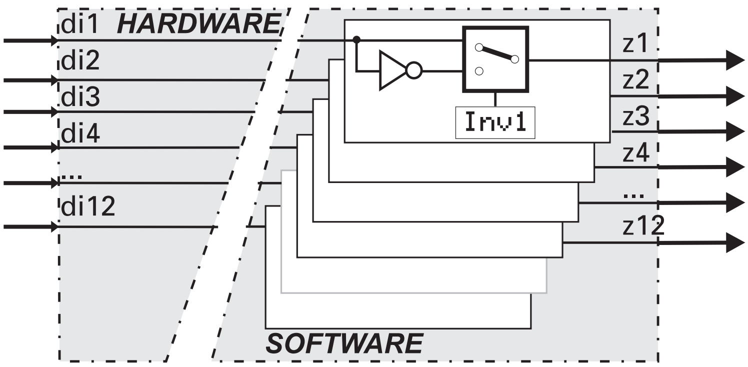


Standard 0/4...20 mA current input

The input resistance is 50 Ω
During configuration, distinction between 0...20 mA and 4...20 mA is made. For standard 4 ... 20 mA signal, the signal behaviour with sensor break can be defined (Fail). Additionally, physical input signal scaling using a defined value of X0 and X100 is possible. For measured value processing, a filter time constant with a numeric value within 0,0 and 200000 can be adjusted (→ Tfm)

25.4 DINPUT (digital inputs)





Function ‘DINPUT’ is used for digital input configuration and parameter setting. The function is assigned firmly to block number 91 and is calculated invariably in each time slot. Inversion of each individual signal can be configured. If inputs di3...di12 are provided is dependent of the KS98 hardware option.

Outputs

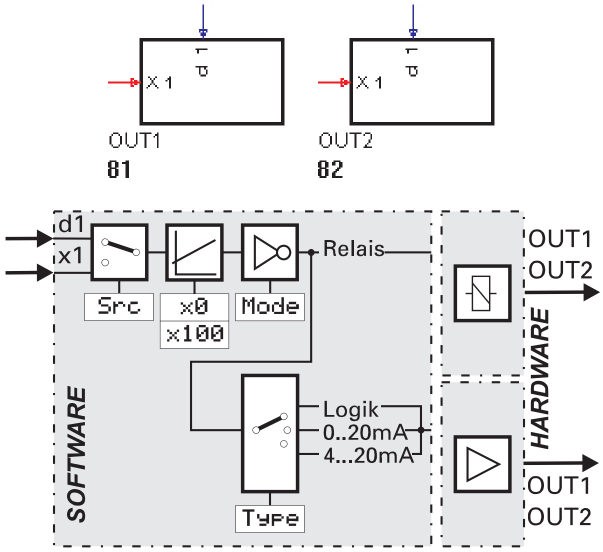
Digital outputs:	
z1...z2	Signal at digital input di1 or di2. (digital inputs di1 and di2 are available in each unit also without options).
z3...z7	Signal at digital input di1 or di2. (Digital inputs di3 to di7 are only provided with option B).
z8...z12	Signal at digital input di1 or di2. (Digital inputs di8 to di12 are only provided with option C).

Parameter and configuration data

Parameter	Description		Values	Default
Inv1	Transfer behaviour	direct output	direct	←
		inverted output	inverse	
Inv2	Transfer behaviour	direct output	direct	←
		inverted output	inverse	
:	:	:	:	:
Inv12	Transfer behaviour	direct output	direct	←
		inverted output	inverse	

26 Outputs

26.1 OUT1 and OUT2 (process outputs 1 and 2)



Functions OUT1 and OUT2 are used for process output OUT1 and OUT2 configuration and parameter setting. Dependent of hardware, the outputs can be analog or relay outputs. Function OUT1 is firmly allocated to block number 81, function OUT2 is firmly allocated to block number 82. They are calculated invariably in each time slot.

With digital input d1 used as signal source, it is switched to the digital output as specified in Mode on an instrument with relay output. With continuous output, switch-over is between 0 and 20 mA as with a logic output.

Analog input x1 used as signal source is taken to the continuous output linearly between x0 and x100 dependent of configuration. With switching output (relay or logic), switching from x0 to x100 is from 50% (hysteresis = 1%).

Inputs / outputs

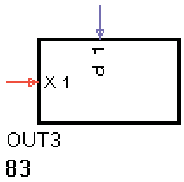
Digital input:	
d1	Input signal with digital signal conversion

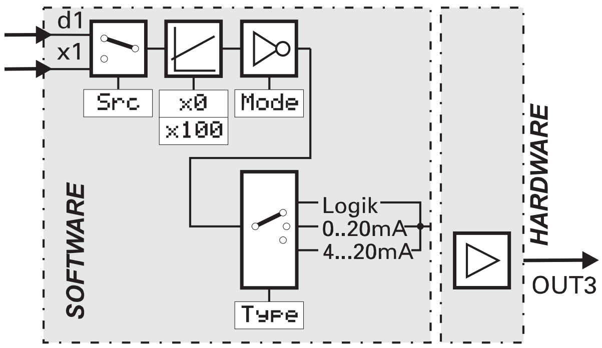
Analog input:	
x1	Input signal with analog signal conversion

Configuration parameters:

Parameter	Description		Values	Default
Src	Signal source	Digital input d1	Digital	←
		Analog input x1	Analog	
Mode	Signal source action	direct/normally open	direct	←
		inverse/normally closed	inverse	
Type	Function of the continuous output	logic 0/20 mA	logic	
		0...20mA	0...20mA	←
		4...20mA	4...20mA	
x0	Analog input value x1 at 0%		-29 999 ... 999 999	0
x100	Analog input value x1 at 100%		-29 999 ... 999 999	100

26.2 OUT3 (process output 3)





Function OUT3 is used for process output OUT3 configuration and parameter setting.
This analog output is provided only with hardware option C.
The function is firmly allocated to block number 83, it is calculated invariably in each time slot.

With digital input **d1** used as signal source, it is switched over between 0 and 20mA as a logic input.
Analog input **x1** used as signal source is taken to the continuous output linearly between **x0** and **x100** according to configuration.

Inputs / outputs

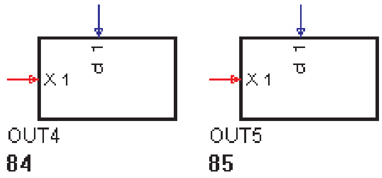
Digital input:	
d1	Input signal with digital signal conversion

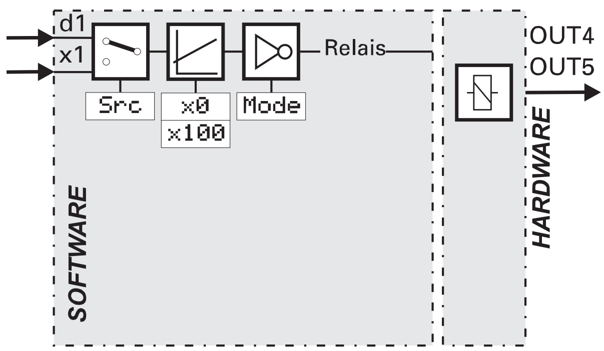
Analog input:	
x1	Input signal with analog signal conversion

Configuration parameters:

Parameter	Description		Values	Default
Src	Signal source	Digital input **d1**	Digital	
		Analog input **x1**	Analog	←
Mode	Signal source action	direct/normally open	direct	←
		inverse/normally closed	inverse	
Type	Continuous output function	logic 0/20 mA	logic	
		0...20mA	0...20mA	←
		4...20mA	4...20mA	
x0	Analog input value **x1** at 0%		-29 999 ... 999 999	0
x100	Analog input value **x1** at 100%		-29 999 ... 999 999	100

26.3 OUT4 and OUT5 (process outputs 4 and 5)





Functions OUT4 and OUT5 are used for process output OUT4 and OUT5 configuration and parameter setting. These two relay outputs are always provided as standard. Function OUT4 is firmly allocated to block number 84, function OUT5 is firmly allocated to block number 85. They are calculated firmly in each time slot.

With digital input d1 used as signal source, it is switched to the relay output as specified in Mode. If analog input x1 is used as signal source, switching from x0 to x100 is from 50% (hysteresis = 1%).

Inputs / outputs

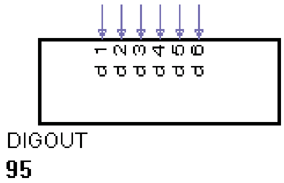
Digital input:	
d1	Standard signal with digital signal conversion

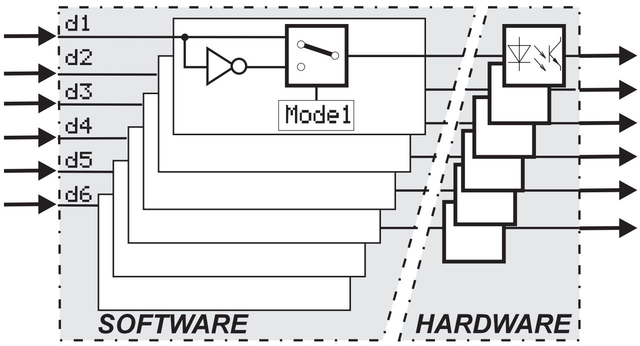
Analog input:	
x1	Input signal with analog signal conversion

Configuration parameter:

Parameter	Description		Values	Default
Src	Signal source	Digital input d1	Digital	←
		Analog input x1	Analog	
Mode	Signal source action	direct/normally open	direct	←
		invers/normally closed	inverse	
x0	Analog input value x1 at 0%		-29 999 ... 999 999	0
x100	Analog input value x1 at 100%		-29 999 ... 999 999	100

26.4 DIGOUT (digital outputs)





Function ‘DIGOUT’ is used for digital output configuration and parameter setting. It is firmly allocated to block number 95 and is calculated invariably in each time slot. Inversion of each individual signal can be configured. If all digital outputs are provided is dependent the KS98 hardware option.

Inputs

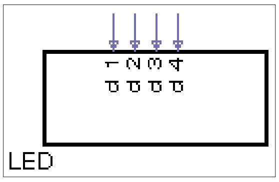
Digital inputs:	
d1...d4	Signal sources for control of digital outputs do1 to do4. (Digital outputs do1 to do4 are provided only in units with hardware option B).
d1...d4	Signal sources for control of digital outputs do5 and do6. (Digital outputs do5 and do6 are provided only in units with hardware option C).

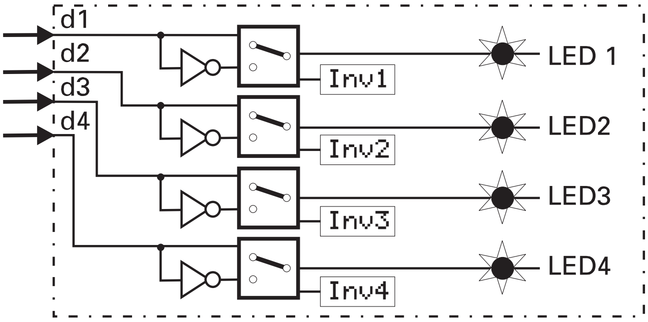
Parameter and configuration data

Parameter	Description		Values	Default
Inv1	Transfer behaviour for d1	direct output	direct	←
		inverted output	inverse	
Inv2	Transfer behaviour for d2	direct output	direct	←
		inverted output	inverse	
:	:	:	:	:
Inv6	Transfer behaviour for d6	direct output	direct	←
		inverted output	inverse	

27 Additional functions

27.1 LED (LED display)





Function LED is used for control of the 4 LEDs. It is firmly allocated to block number 96 and is calculated in each time slot. The statuses of digital inputs d1 ...d4 are output to LED 1...4. The statuses can be inverted via parameter Inv.

Inputs:

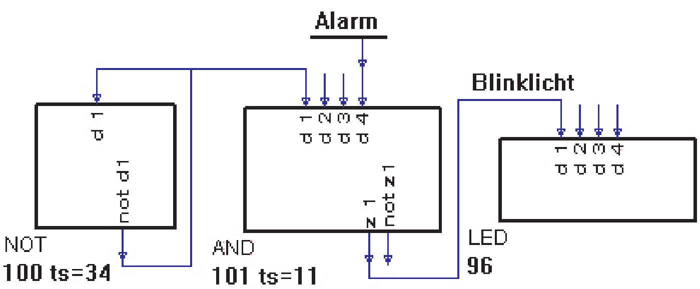
Input	Description
d1	LED 1
d2	LED 2
d3	LED 3
d4	LED 4

Parameters:

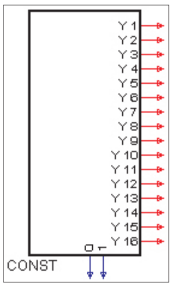
Parameter	Description	Range	Default
Inv 1	Inv1= 0 ≙ d1=1 LED1 is lit Inv1 = 1 ≙ d1= 0 LED1 is lit	0...1	0
Inv 2	Inv2= 0 ≙ d2=1 LED2 is lit Inv2 = 1 ≙ d2= 0 LED2 is lit	0...1	0
Inv 3	Inv3= 0 ≙ d3=1 LED3 is lit Inv3 = 1 ≙ d3= 0 LED3 is lit	0...1	0
Inv 4	Inv4= 0 ≙ d4=1 LED4 is lit Inv4 = 1 ≙ d4= 0 LED4 is lit	0...1	0

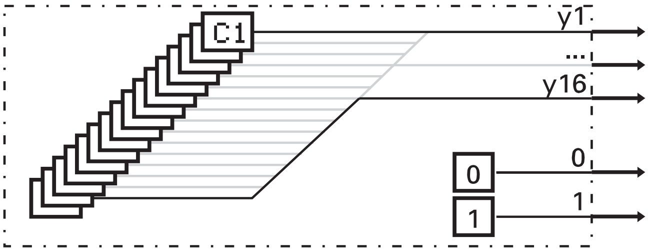
Example:

If a simple flashing function is to be produced, this is possible with the following example.
The sampling-time period code of the NOT-function indicates the flash frequency.



27.2 CONST (constant function)





16 analog constants at output y1...y16 and logic statuses 0 and 1 are made available. The block number is firmly configured with 99.

Outputs:

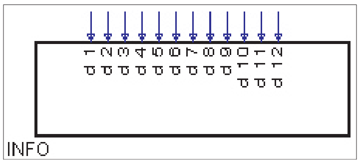
Digital outputs	
0	Logic 0 is always output at this output.
1	Logic 1 is always output at this output.

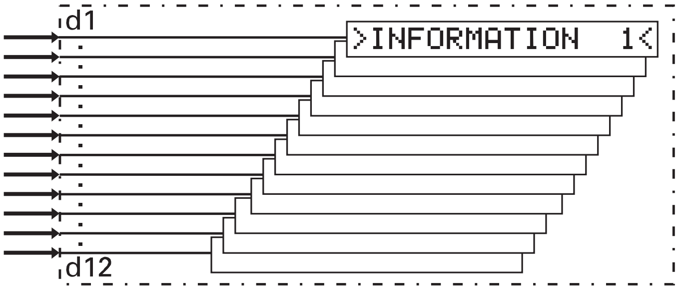
Analog outputs	
y1	Constant C1 is output.
y2	Constant C2 is output.
y3	Constant C3 is output.
y4	Constant C4 is output.
y5	Constant C5 is output.
y6	Constant C6 is output.
y7	Constant C7 is output.
y8	Constant C8 is output.
y9	Constant C9 is output.
y10	Constant C10 is output.
y11	Constant C11 is output.
y12	Constant C12 is output.
y13	Constant C13 is output.
y14	Constant C14 is output.
y15	Constant C15 is output.
y16	Constant C16 is output.

Parameters:

Parameter	Description	Range	Default
C1..C16	Analog constants	-29999...999 999	0

27.3 INFO (information function)





This function can be used for display of 12 user texts with max. 16 characters each by setting the relevant input d1...d12. The information is displayed in the "header" of operating pages (level 1 data) in alternation with the description of the called up operating page. If several texts are available simultaneously, they are displayed successively.
The block number is fixed to 97 and calculated once per time slot.

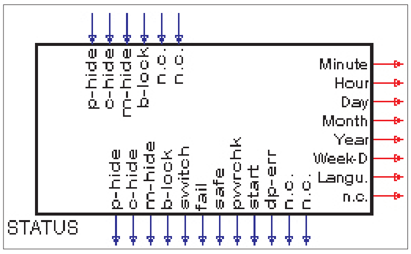
Inputs:

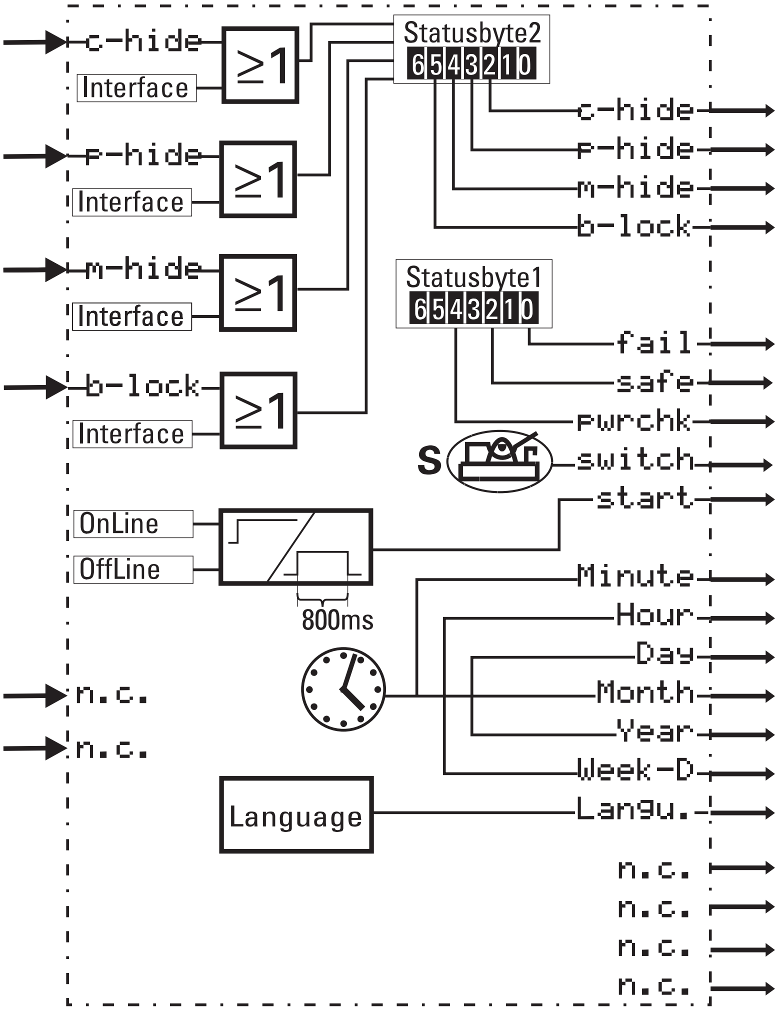
Digital inputs	
d1	=1 → the information configured in Text 1 is displayed.
d2	=1 → the information configured in Text 2 is displayed.
d3	=1 → the information configured in Text 3 is displayed.
d4	=1 → the information configured in Text 4 is displayed.
d5	=1 → the information configured in Text 5 is displayed.
d6	=1 → the information configured in Text 6 is displayed.
d7	=1 → the information configured in Text 7 is displayed.
d8	=1 → the information configured in Text 8 is displayed.
d9	=1 → the information configured in Text 9 is displayed.
d10	=1 → the information configured in Text 10 is displayed.
d11	=1 → the information configured in Text 11 is displayed.
d12	=1 → the information configured in Text 12 is displayed.

Parameters:

Parameter	Description	Range	Default
Text1 ... Text12	User text with max. 16 characters each	alphanumeric characters	>INFORMATION 1 < ... >INFORMATION 1 <

27.4 STATUS (status function)





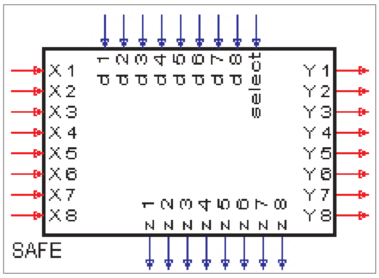
The function provides information from the KS98 instrument status byte at its digital outputs. The block is fixed to 98 and updated per time slot.

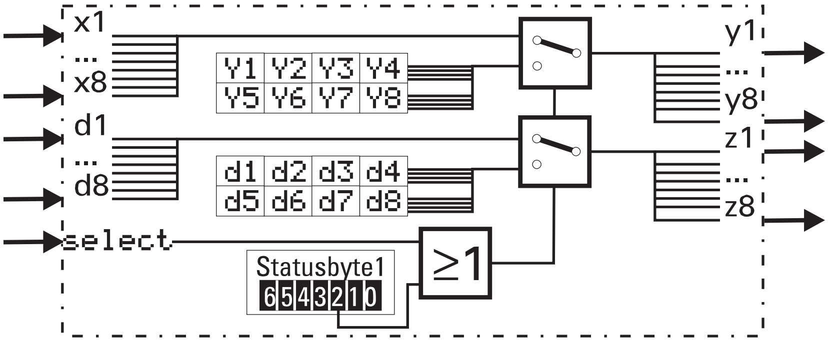
Input	Description
c-hide	With c-hide = 1, a configuration change via operation is disabled.
p-hide	= 1 parameters/configurations via operation disabled
m-hide	= 1 The main menu is not displayed, operating pages are displayed only during online mode.
b-block	= 1 The use of the bus interface is blocked.

Output	Description
c-hide	= 1 configuration change disabled
p-hide	= 1 parameters/configurations disabled
m-hide	= 1 The main menu is not displayed, the operating pages are displayed only during online mode
b-block	= 1 the use of the bus interface is blocked
fail	= 1 common message sensor error of inputs AINP1...AINP6
safe	= 1 safety status set via interface with code 22, Fbno. 0, Fctno. 0
pwrchk	Power-fail check. This value is always at reset(0) after power-on. It can be activated(1) by an interface message and permits a response to any power failure.
switch	S.I.L. switch open = 0 closed = 1. This information permits blocking via the hardware.
start	With change from offline to online, start is 1 during 800 ms. During this time, all time groups were calculated at least once.
Minute	Minute of the real-time clock 0...59
Hour	Hour of the real-time clock 0...23
Day	Day of the real-time clock 0...31
Month	Month of the real-time clock 1...12
Year	Year of the real-time clock 1970....2069
Week-D	Weekday of the real-time clock 0...6 ≙ Su...Sa
Langu	Language German = 0 language English = 1 Language selection is in Miscellaneous,Device data

With missing real-time clock option,, these outputs provide = 0

27.5 SAFE (safety function)





Function SAFE is used for generation of defined analog output values and digital statuses dependent of digital input select or of the status received via the interface. In the normal case select = 0 and status = 0, the values applied to the inputs are switched through to the outputs without change. For select = 1 or status = 1, configured data **z1** ...**z8** and **y1** ...**y8** are switched through to the outputs.

28 KS98 I/O extension modules

Can be used in KS98: 9407 - 9xx - x3xx1 and 9407 - 9xx - x4xx1.

Safety hints

ESD !

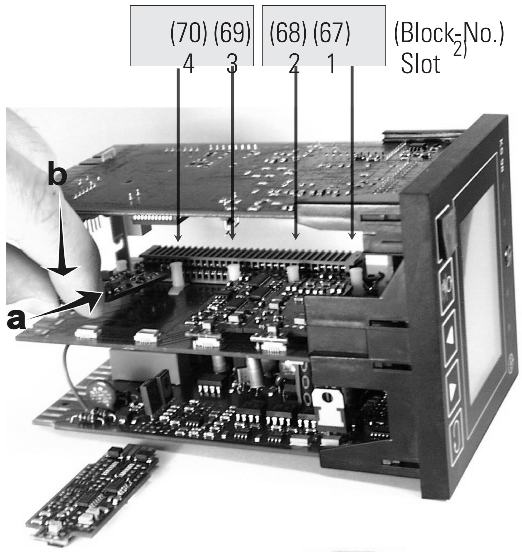
- Contains electrostatically sensitive components
- Original packaging protects against electrostatic discharge (ESD)
- Transport only in the original packaging
- During mounting, follow the rules for protection against ESD.

Connection:
Note the KS98 engineering! It determines the allocation of connector positions and the signification of connections.

Maintenance:
The instruments do not require any particular maintenance.

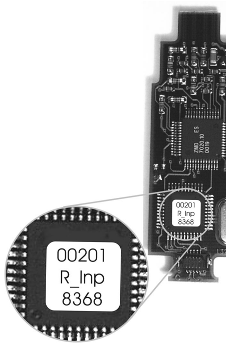
When opening the instruments, live parts may be exposed. The instruments must be completely de-energized before any work is done. The instruments contain electrostatically sensitive components.

I/O modules	**9407 - 998 -**	**00**	**xx1**
Module type:			
Analog inputs:			
Pt 100 / 1000, Ni 100 /1000, Resistance , potentiometer			20
Thermocouple, mV, 0/4..20mA			21
-50...1500mV, 0...10V			22
Analog outputs:[1]			
0/2...10V, 0...±10V			30
0/4...20mA, 0...±20mA			31
Digital inputs/output:			
Digital I/O (universal)			40
Frequency/counter			41



The various modules are distinguished by label inscription. The 5 last digits of the order number are given in the upper line.

Mounting
After releasing the locking screw, withdraw the KS98 module from the housing. (a) - Insert the module into the required socket with the printed label pointing downwards into the green connector and click it in position in the small, white contact (b) at the top.



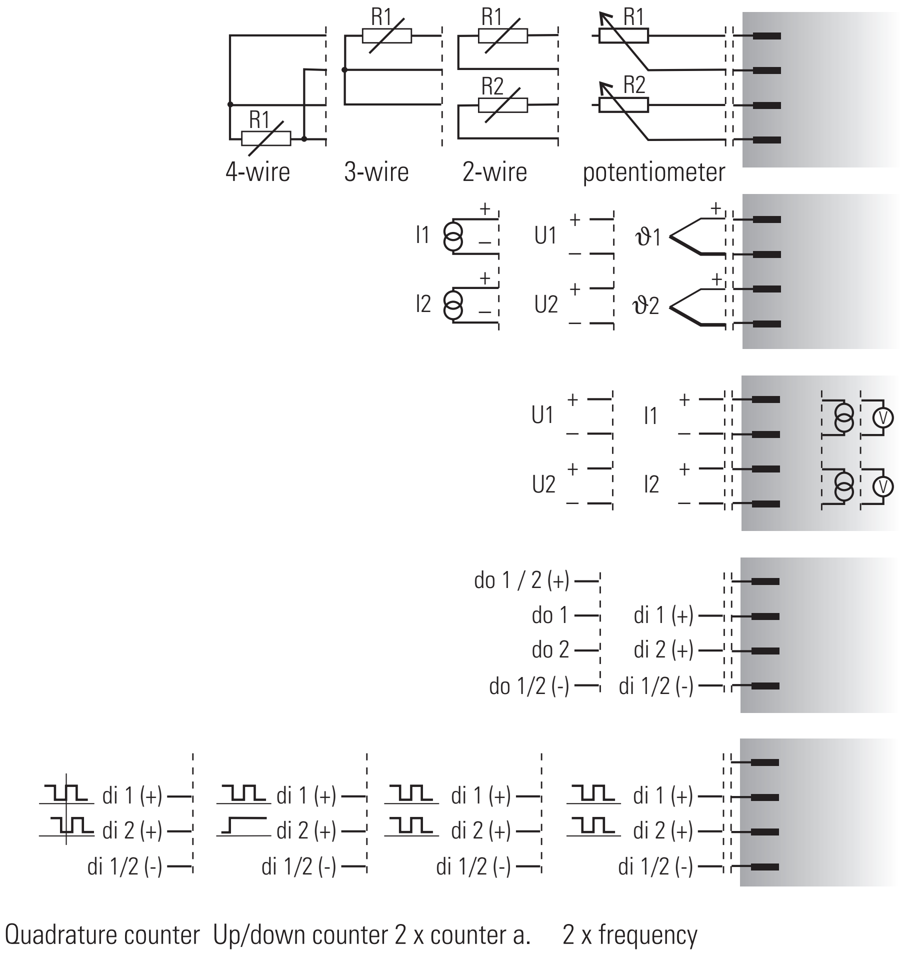
1)

PERFORMANCE LIMITS
Due to the maximum permissible self-heating, the number of analog output modules per KS98 is limited:
max. one current output module! Max.one voltage output module, if there is already a current output module (but in different, galvanically isolated module groups)! The total of performance factors (P-factor, → Technical data must not exceed 100% ! Exceeded performance limits are displayed in the engineering tool. Unless a current output module is used, all sockets can be provided with any modules. Max. 1 current output module (on any socket)! Max 1 current and max. 1 voltage output module, but on galvanically isolated sockets!

Example: current output module on slot 1 or 2 and voltage output module on slot 3 or 4.
The total of P factors is 95%. I.e. 1 more resistance or 1 TC/mV/mA module can be fitted.

[2] **Galvanic isolation:** Slots 1-2 are galvanically isolated from 3-4.

Electrical connections of modular option C



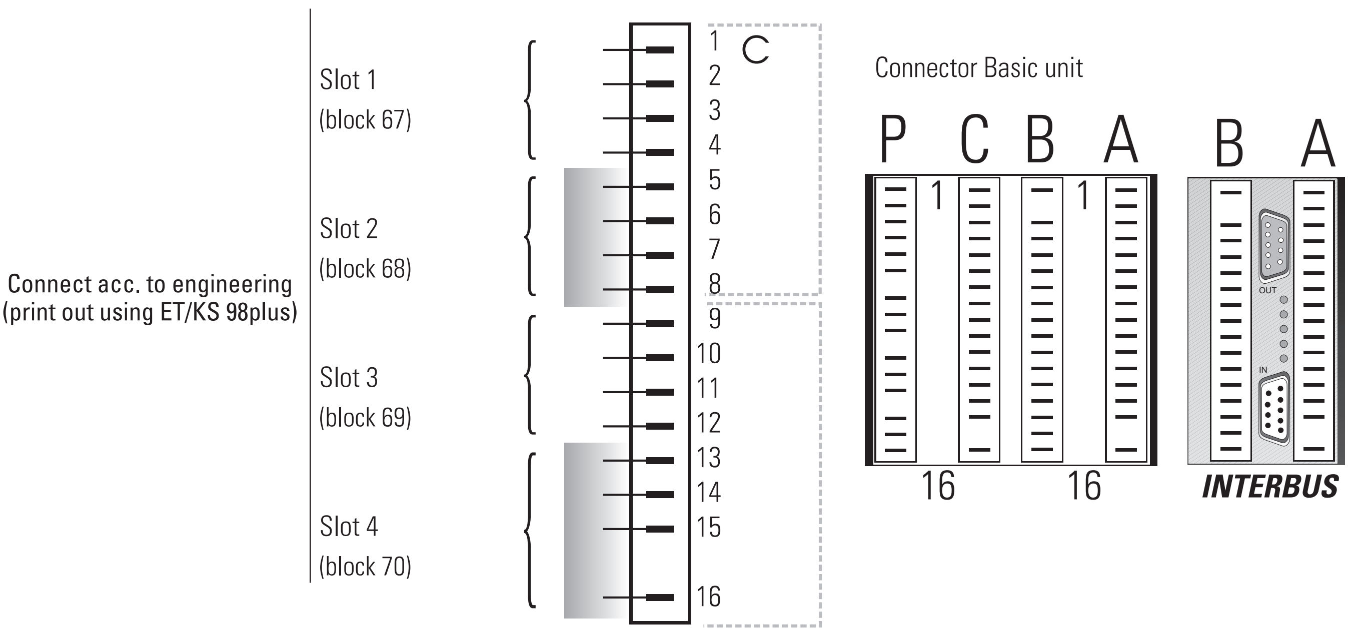
Resistance input
(9407-998-0x201)

TC, mV, mA / V inputs
(9407-998-0x211) /
(9407-998-0x221)

Voltage /
current outputs
(9407-998-0x301) /
(9407-998-0x311)

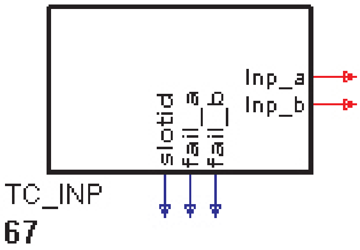
Combined
digital inputs/ outputs
(9407-998-0x401)

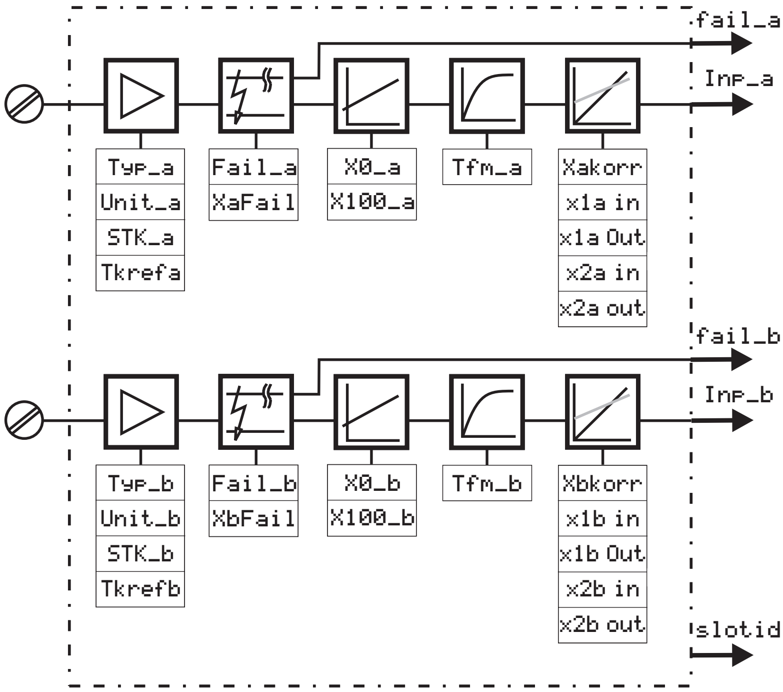
Frequency / counter inputs
(9407-998-0x241



29 Modular I/O - extension-modules

29.1 TC_INP (analog input card TC, mV, mA)





Analog input, plugs into modular options card C
For configuration and parameter setting of the analog inputs R_INP.
Calculation of the inputs is fixed to once per time slot.

Digital outputs:

slotid
0 = correct module fitted
1 = wrong module fitted

fail_a
0 = no measurement error at channel a detected
1 = measurement error at channel a detected; e.g. sensor break

fail_b
0 = no measurement error at channel b detected
1 = measurement error at channel b detected; e.g. sensor break

Analog outputs:

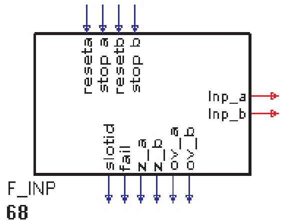
Inp_a → measurement value channel a
Inp_b → measurement value channel b

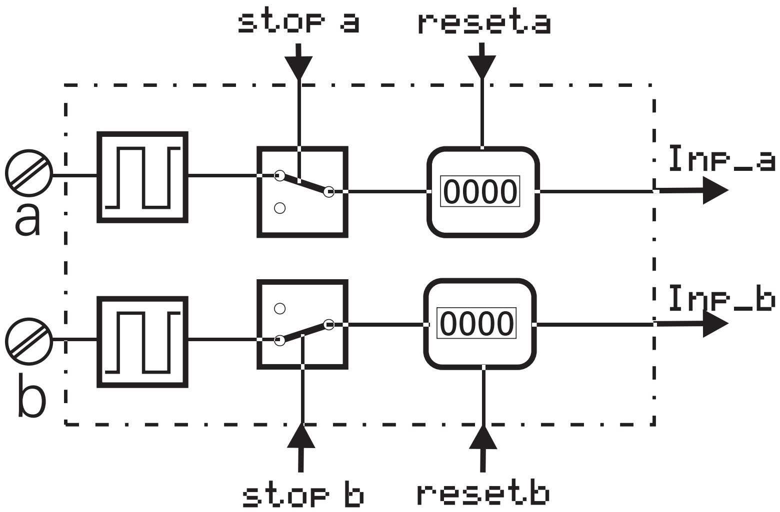
Parameter	Beschreibung	Values	Default
x1a in	Measured value correction Inp_a, P1 input value	Real	0
x1aOut	Measured value correction Inp_a, P1 output value		0
x2a in	Measured value correction Inp_a, P2 input value		100
x2aOut	Measured value correction Inp_a, P2 output value		100
x1b in	Measured value correction Inp_b, P1 input value		0
x1bOut	Measured value correction Inp_b, P1 output value		0
x2b in	Measured value correction Inp_b, P2 input value		100
x2bOut	Measured value correction Inp_b, P2 output value		100

Configuration	Description	Values	Default
Typ_a Typ_b	Type L -200...900 °C	00	30
	Type J -200...900 °C	01	
	Type K -200...1350 °C	02	
	Type N -200...1300 °C	03	
	Type S -50...1760 °C	04	
	Type R -50...1760 °C	05	
	Type T -200...400 °C	06	
	Type W(C) 0...2300 °C	07	
	Type E -200...900 °C	08	
	Type B 0...1820 °C	09	
	Type D 0...2300 °C	10	
	Voltage 0...30mV	27	
	Voltage 0...100mV	28	
	Voltage 0...300mV	29	
	Standard signal 0...20mA	30	
	Standard signal 4...20mA	31	
Fail_a Fail_b	Switched off	0	1
	Upscale, Inp_a (Inp_b) = x100_a (x100_b)	1	
	Downscale, Inp_a (Inp_b) = x0_a (x0_b)	2	
	Substitute value, Inp_a (Inp_b) = XaFail (XbFail)	3	
Xakorr Xbkorr	Measured value correction Inp_a (b) switched off	0	0
	Measured value correction Inp_a (b) effective	1	
Unit_a Unit_b	Unit of the measured value of Inp_a (b) = °C	1	1
	Unit of the measured value of Inp_a (b) = °F	2	
STK_a STK_b	Internal temperature compensation	1	1
	External temperature compensation	2	
x0_a(b)	Physical value Inp_a (Inp_b) at 0%	Real	0
x100_a(b)	Physical value Inp_a (Inp_b) at 100%	Real	100
Xa(b)Fail	Physical value Inp_a (Inp_b) at 0%	Real	0
Tfm_a(b)	Filter time constant of _a (Inp_b) in seconds	Real	0,5
Tkrefa(b)	Reference temperature for Inp_a(b) at STK_a(b)	Real	0

29.2 F_Inp (frequency/counter input)

The frequency/counter input plugs into the modular options card C.





For configuration and parameter setting of input F_INP.
Input calculation is fixed to once per time slot.

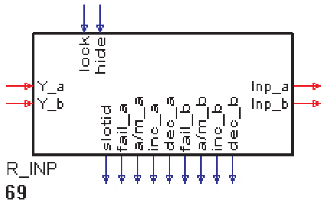
Digital inputs:
reseta →1 = the value for Inp_a is reset to 0.
stop a →1 = the instantaneous value for Inp_a remains unchanged.
resetb →1 = the value for Inp_b is reset to 0.
stop b →1 = the instantaneous value for Inp_b remains unchanged.

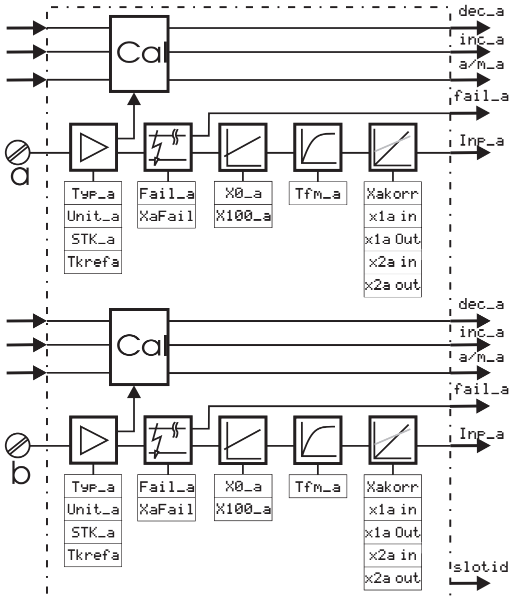
Digital outputs:
slotid → 0 = correct module fitted 1 = wrong module fitted
fail →1 = inserted module is detected, but no communication to the module.
z_a → signal status of HW input a
z_b → signal status of HW input b
ov_a →1 = frequency at HW input a exceeds the maximum permissible 20kHz
ov_b →1 = frequency at HW input b exceeds the maximum permissible 20kHz

Analog outputs
Inp_a →output value for channel a
Inp_b →output value for channel b

Configuration	Description	Values	Default
Func_a	DigInput → control input	0	1
	Count_1 → up counter	1	
	Count_2 → up/down counter	2	
	Count_3 → up/down counter with direction signal	3	
	Count_4 → quadrature signal	4	
	Frequenz → frequency measuring	5	
Func_b	DigInput → control input	0	1
	Count_1 → up counter	1	
	Frequenz → frequency measuring	5	
Time	for frequency measuring in seconds	0,1...20	10

29.3 R_Inp (analog input card)





Analog input card for Pt100/1000, Ni 100/1000, resistance and potentiometer

Analog input, plugs into modular options card C
For configuration and parameter setting of analog inputs R_INP.
Input calculation is fixed to once per time slot.

Digital inputs:

lock =1 ⟶ calibration disabled
hide =1 ⟶ calibration display suppressed

Digital outputs:

slotid
0 = correct module inserted
1 = wrong module inserted

fail_a(b)
0 = no measurement error at channel a (b) detected
1 = measurement error at channel a (b) ; e.g. sensor break

a/m_a(b)
Status of manual key ⟶ 0 = automatic
Status of manual key ⟶ 1 = manual

inc_a(b) =1 ⟶▲-key pressed
dec_a(b) =1 ⟶▼-key pressed

Analog inputs:

Y_a(b) ⟶ position feedback

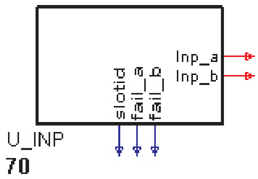
Analog outputs:

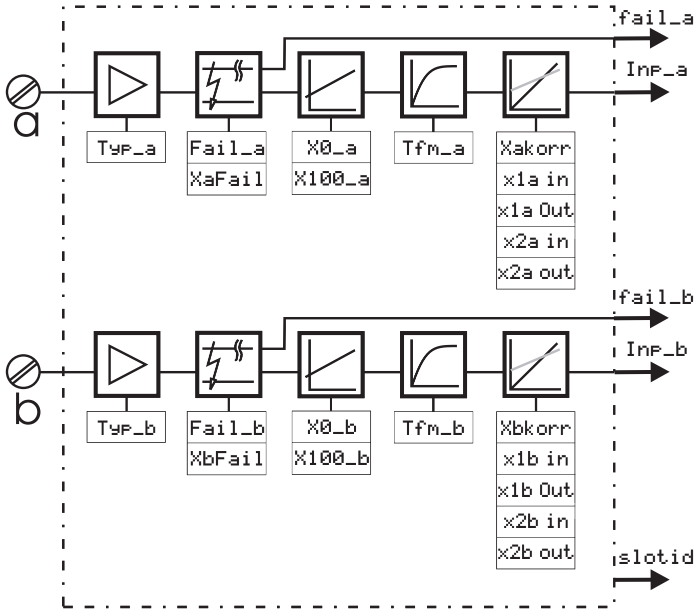
Inp_a ⟶ measured value channel a
Inp_b ⟶ measured value channel b

Parameter	Description	Value	Default
x1a in	Measured value correction Inp_a, P1 input value	Real	0
x1aOut	Measured value correction Inp_a, P1 output value		0
x2a in	Measured value correction Inp_a, P2 input value		100
x2aOut	Measured value correction Inp_a, P2 output value		100
x1b in	Measured value correction Inp_b, P1 input value		0
x1bOut	Measured value correction Inp_b, P1 output value		0
x2b in	Measured value correction Inp_b, P2 input value		100
x2bOut	Measured value correction Inp_b, P2 output value		100

Configuration	Description	Value	Default
Typ_a Typ_b	Pt100 (850) -200 ... 850 °C	00	0
	Pt100 (100) -200 ... 100 °C	01	
	Pt1000 (-1) -200 ... 850 °C	02	
	Pt1000 (-2) -200 ... 100 °C	03	
	Ni100 -60 ... 180 °C	04	
	Ni1000 -60 ... 180 °C	05	
	R160 resistance 0 ... 160 Ohm	06	
	R450 resistance 0 ... 450 Ohm	07	
	R1600 resistance 0 ... 1600 Ohm	08	
	R4500 resistance 0 ... 4500 Ohm	09	
	Potentiometer 160 Potentiometer 0 ... 160 Ohm	10	
	Potentiometer 450 Potentiometer 0 ... 450 Ohm	11	
	Potentiometer 1600 Potentiometer 0 ... 1600 Ohm	12	
	Potentiometer 4500 Potentiometer 0 ... 4500 Ohm	13	
Fail_a Fail_b	Switched off	0	1
	Upscale, Inp_a (Inp_b) = x100_a (x100_b)	1	
	Downscale, Inp_a (Inp_b) = x0_a (x0_b)	2	
	Substitute value, Inp_a (Inp_b) = XaFail (XbFail)	3	
Xakorr Xbkorr	Measured value correction Inp_a (b) switched off	0	0
	Measured value correction Inp_a (b) effective	1	
Unit_a Unit_b	Unit of the measured value of Inp_a (b) = °C	1	1
	Unit of the measured value of Inp_a (b) = °F	2	
Mode	Inp_a and Inp_b: 2-wire connection	0	0
	Inp_a: 3-wire connection no Inp_b	1	
	Inp_a: 4-wire connection no Inp_b	2	
x0_a(b)	Physical value Inp_a (Inp_b) at 0%	Real	0
x100_a(b)	Physical value Inp_a (Inp_b) at 100%	Real	100
Xa(b)Fail	Substitute value with sensor error at Inp_a(b)	Real	0
Tfm_a(b)	Filter time constant of _a (Inp_b) in seconds	Real	0,5
Kal_1a(b)	1st calibration value Inp_a(b) (read only)	Real	0
Kal_2a(b)	2nd calibration value Inp_a(b) (read only)	Real	100

29.4 U_INP (analog input card -50...1500mV, 0...10V)





Analog input, plugs into modular options card
For configuration and parameter setting of the analog input U_INP.
Input calculation is fixed to once per time slot.

Digital outputs:

slotid
0 = correct module fitted
1 = wrong module fitted

fail_a
0 = no measurement error at channel a detected
1 = measurement error at channel a detected; e.g. sensor break

fail_b
0 = no measurement error at channel b detected
1 = measurement error at channel b detected; e.g. sensor break

Analog outputs:

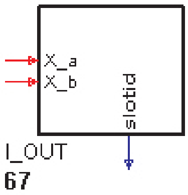
Inp_a → measured value channel a
Inp_b → measured value channel b

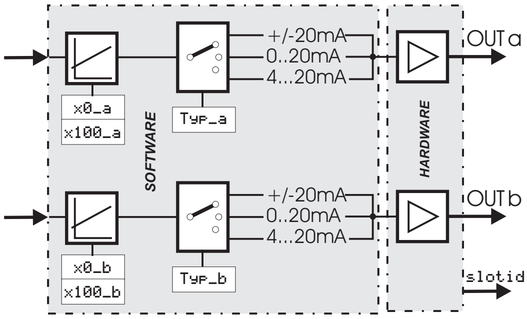
Parameter	Description	Value	Default
x1a in	Measured value correction Inp_a, P1 input value	Real	0
x1aOut	Measured value correction Inp_a, P1 output value		0
x2a in	Measured value correction Inp_a, P2 input value		100
x2aOut	Measured value correction Inp_a, P2 output value		100
x1b in	Measured value correction Inp_b, P1 input value		0
x1bOut	Measured value correction Inp_b, P1 output value		0
x2b in	Measured value correction Inp_b, P2 input value		100
x2bOut	Measured value correction Inp_b, P2 output value		100

Configuration	Description	Value	Default
Typ_a	Voltage 0...10V Voltage -50...1500mV	0 1	0
Fail_a	Switched off Upscale, Inp_a = x100_a Downscale, Inp_a = x0_a Substitute value, Inp_a = XaFail	0 1 2 3	1
Xakorr	Measured value correction Inp_a switched off Measured value correction Inp_a effective	0 1	0
Typ_b	Voltage 0...10V Voltage -50...1500mV	0 1	0
Fail_b	Switched off Upscale, Inp_b = x100_b Downscale, Inp_b= x0_b Substitute value, Inp_b = XbFail	0 1 2 3	1
Xbkorr	Measured value correction Inp_b switched off Measured value correction Inp_b effective	0 1	0
a0_a	Physical value Inp_a at 0%	Real	0
x100_a	Physical value Inp_a at 100%	Real	100
XaFail	Substitute value with sensor error at Inp_a	Real	0
Tfm_a	Filter time constant of Inp_a in seconds	Real	0,5
x0_b	Physical value Inp_b at 0%	Real	0
x100_b	Physical value Inp_b at 100%	Real	100
XbFail	Substitute value with sensor error at Inp_b	Real	0
Tfm_b	Filter time constant of Inp_b in seconds	Real	0,5

29.5 I_OUT (analog output card 0/4...20mA, +/-20mA)

Analog output, plugs into modular options card C





For configuration and parameter setting of analog output I_OUT.
Output calculation is fixed to once per time slot.

Digital output:
slotid
0 = correct module fitted
1 = wrong module fitted

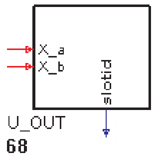
Analog inputs:

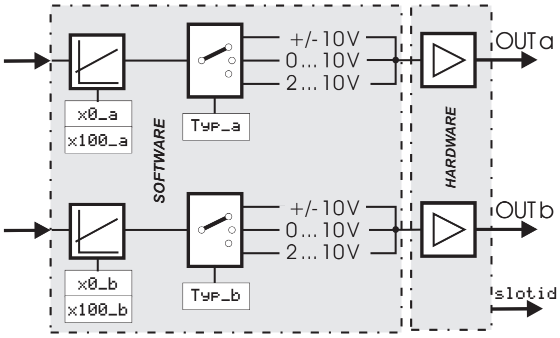
X_a → output value for channel a
X_b → output value for channel b

Configuration	Description	Values	Default
Typ_a	0...20mA 4...20mA +/-20mA	0 1 2	0
x0_a	Physical value Inp_a at 0%	Real	0
x100_a	Physical value Inp_a at 100%	Real	100
Typ_b	0...20mA 4...20mA +/-20mA	0 1 2	0
x0_b	Physical value Inp_b at 0%	Real	0
x100_b	Physical value Inp_b at 100%	Real	100

29.6 U_OUT (analog output card 0/2...10V, +/-10V)

Analog output, plugs into modular options card C





For configuration and parameter setting of analog output U_OUT.
Output calculation is fixed to once per time slot.

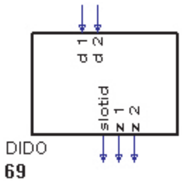
Configuration	Description	Values	Default
Typ_a	0...10V 2...10V +/-10V	0 1 2	0
x0_a	Physical value Inp_a at 0%	Real	0
x100_a	Physical value Inp_a at 100%	Real	100
Typ_b	0...10V 2...10V +/-10V	0 1 2	0
x0_b	Physical value Inp_b at 0%	Real	0
x100_b	Physical value Inp_b at 100%	Real	100

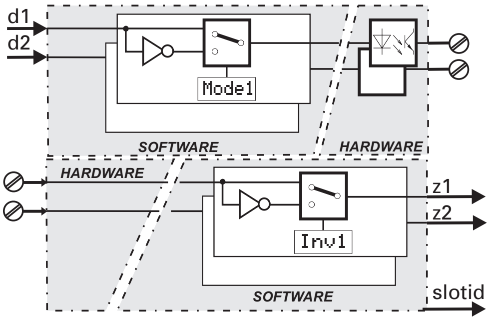
Digital output:
slotid
0 = correct module fitted
1 = wrong module fitted

Analog inputs:
X_a → output value for channel a
X_b → output value for channel b

29.7 DIDO (digital input/output card)

Digital input/output card, plugs into modular options card C





For configuration and parameter setting of digital inputs/outputs DIDO.
Function block calculation is fixed to once per time slot.

Digital inputs:

d1 → if configured as an output: hardware output a
d2 → if configured as an output: hardware output b

Digital outputs:

slotid

0 = correct module fitted
1 = wrong module fitted

z1 → status of hardware input a; if configured as an output: the output value read back
z2 → status of hardware input b; if configured as an output: the output value read back

Configuration	Description	Values	Default
Inv_Ia	direct - HW input di1 direct at z1	0	0
	inverse - HW input di1inverted at z1	1	
Inv_Ib	direct - HW input di2 direct at z2	0	0
	inverse - HW input di2 inverted at z2	1	
Inv_	direct - d1 direct on HW output do1	0	0
	inverse - d1 inverted at HW output do1	1	
Inv_Ob	direct - d1 direct on HWoutput do1	0	0
	inverse - d2 inverted on HW output do2	1	
Mode_a	Input - only HW input d1 at z1	0	0
	Output - d1 at HW output do1 with feedback at z1	1	
Mode_b	Input - only HW input d2 at z2	0	0
	Output - d2 at HW output do2 with feedback at z2	1	

30 Function management

Max. 450 function blocks can be used. Each function requires a defined portion of the working memory and a defined calculation time.
The used up resources can be examined in the engineering Tool under .

Memory requirement and calculation time

30.1

Function	Time %	Memory %
Scaling and calculating functions		
ABSV	0,4	0,1
ADSU	0,9	0,2
MUDI	0,9	0,2
SQRT	1,3	0,1
SCAL	3,2	0,1
10EXP	3,0	0,1
EEXP	1,6	0,1
LN	1,6	0,1
LG10	1,6	0,1
Non-linear functions		
GAP	0,3	0,1
CHAR	0,9	0,4
Trigonometric functions		
SIN	1,4	0,1
COS	2,0	0,1
TAN	1,4	0,1
COT	2,9	0,1
ARCSIN	2,4	0,1
ARCCOS	2,4	0,1
ARCTAN	1,8	0,1
ARCCOT	1,9	0,1
Logic functions		
AND	0,2	0,1
NOT	0,2	0,1
OR	0,2	0,1
EXOR	0,2	0,1
BOUNCE	0,3	0,1
FLIP	0,2	0,1
MONO	1,0	0,2
STEP	0,8	0,2
TIME1	1,2	0,2
Signal converters		
ABIN	1,5	0,2
TRUNC	0,3	0,1
PULS	0,9	0,1
COUN	0,4	0,2
MEAN	0,9	0,1
Controller		
CONTR	10,0	3,3
CONTR+	10,0	3,6

Function	Time %	Memory %
Time functions		
LEAD	0,7	0,2
INTE	0,6	0,2
LAG1	0,5	0,1
DELA1	0,9	2,2
DELA2	0,9	2,2
FILT	0,6	0,1
TIMER	0,5	0,2
TIME2	0,5	0,2
Selecting and storage		
EXTR	0,5	0,2
PEAK	0,3	0,1
TRST	0,3	0,1
SELC	0,3	0,2
SELP	0,3	0,2
SELV1	0,3	0,1
SOUT	0,3	0,1
REZEPT	0,7	0,4
2OF3	1,4	0,2
SELV2	0,4	0,1
Limit value signalling / limiting		
ALLP	0,8	0,2
ALLV	0,8	0,2
EQUAL	0,6	0,1
VELO	0,5	0,2
LIMIT	1,4	0,3
ALARM	0,4	0,2
Inputs		
AINP1	0,5	0,5
AINP3	0,4	0,2
AINP4	0,4	0,2
AINP5	0,4	0,2
AINP6	0,5	0,5
DINPUT	0,3	0,2
Outputs		
OUT1	0,9	0,1
OUT2	0,9	0,1
OUT3	0,9	0,1
OUT4	0,9	0,1
OUT5	0,9	0,1
DIGOUT	0,2	0,2

Function	Time %	Memory %
Additional functions		
LED	0,2	0,1
INFO	0,2	0,9
STATUS	0,4	0,2
CONST	0,2	0,4
SAFE	0,3	0,4
Visualization		
VWERT	0,4	1,8
VBAR	0,3	0,8
VPARA	2,5	1,1
VTREND	0,8	1,4
Communication		
L1READ	0,3	0,2
L1WRIT	0,3	0,4
DPREAD	0,5	0,3
DPWRIT	0,5	0,3
Programmer		
APROG	3,6	0,8
APROGD	0,9	0,4
DPROG	3,6	0,7
DPROGD	0,9	0,4
KS98+ CANopen		
C_RM2x	3,0	1,0
RM_DI	0,5	0,3
RM_DO	0,5	0,5
RM_AI	0,5	0,7
RM_AO	0,5	0,5
CRCV	4,0	0,3
CSEND	5,0	0,5
C_KS8x	3,0	0,8
KS8x	0,3	0,3
Modular Option C		
TC_Imp	0,5	0,5
F_Imp	0,9	0,2
R_Imp	0,9	0,7
U_Imp	0,9	0,4
I_Out	0,5	0,2
U_Out	0,5	0,2
DIDO	0,5	0,2

30.2 Sampling intervals

The table opposite shows the sampling intervals for conversion of the input signals into internal values and conversion of the internal values into output signals (hardware conversion).
The sampling interval for software calculation of function blocks AINP1, AINP3...AINP6, DINPUT, STATUS, CONST, LED, INFO, OUT1...OUT5 and DIGOUT is 100 ms.

Input or output	Sampling interval
INP1	at intervals of 200 ms
INP3 / INP4	at intervals of 100 ms
INP5	at intervals of 800 ms
INP6	at intervals of 400 ms
di1...di12	at intervals of 100 ms
OUT1...OUT5 / do1...do6	at intervals of 100 ms

Calculation of the other function blocks is at equal intervals according to their allocation to the 8 time slots of 100 ms each.
Allocation of a block to one or several time slots (at intervals of 100, 200, 400 or 800 ms) is in the engineering. For each block, the engineering tool provides an identification (ts) which can be used to determine the allocation from the table opposite.
The total of calculation times of all required function blocks must be < 100 % for each time slot.

	Time slot								
ts	1	2	3	4	5	6	7	8	Sampling interval
11	✘	✘	✘	✘	✘	✘	✘	✘	at intervals of 100 ms
21	✘		✘		✘		✘		at intervals of 200 ms
22		✘		✘		✘		✘	at intervals of 200 ms
31	✘				✘				at intervals of 400 ms
32		✘				✘			at intervals of 400 ms
33			✘				✘		at intervals of 400 ms
34				✘				✘	at intervals of 400 ms
41	✘								at intervals of 800 ms
42		✘							at intervals of 800 ms
43			✘						at intervals of 800 ms
44				✘					at intervals of 800 ms
45					✘				at intervals of 800 ms
46						✘			at intervals of 800 ms
47							✘		at intervals of 800 ms
48								✘	at intervals of 800 ms

30.3 EEPROM data

Data are stored in non-volatile EEPROM. The manufacturers specify approx. 100 000 permissible write/read cycles per EEPROM address, in reality, however, this value can mostly be exceeded by a multiple. If parameters and configurations are changed exclusively manually, exceeding the max. number of write/read cycles is almost precluded. With digital interface or automatic parameter changes, however, taking the maximum number of write/read cycles into account is indispensable, and measures against excessively frequent parameter writing must be take

31 Examples

During installation of the engineering tools, several examples were included. These are in path:
C:\Pmatools\Et98\prj\example

31.1 Useful small engineerings

Cascaded counter with pulse generator(ZAEHLER.EDG)
An INTE is used for generating pulses.
Max. parameter =1, time constant to 3600 sec.
An input value at x1 of e.g. 20 weighted via the MUDI generates 20 pulses per hour.
The first counter counts to 1000, the second counter counts the (1000s) overflows.

Simple password function (PASSWORT.EDG)
A VWERT is used for password entry. The output is not fed back to the input, for suppressing display of the entered value after pressing the enter key.
The current hour of the status block is used as a password (only with clock). The EQUAL block determines the condition for disabling the parameter level.

Password from the CONST block (PASSWORD.EDG)
A VWERT is used for password entry. The output is not fed back to the input, for suppression of the display of the entered value after pressing the enter key. A value of the constant block is used as a password. The EQUAL block determines the condition for disabling the parameter level and display suppression of the VWERT page.

Macro for dynamic alarm processing (ALARMSEL.EDG)
ASELV2 can be used to select one of 4 values for alarm processing.
An ALLV compares the value with an upper and a lower limit definable via a VWERT.
The alarms are displayed at the second VWERT and output to a relay via an OR. Each of the two VWERT can define or display two further alarm limits. Therefore, the configuration can be extended by another ALLV. As an example, possible alarm acknowledgement via a flipflop is provided. Alarms are held in the LED display and the alarm line until acknowledgment via the VWERT (alarms) .

Alarm acknowledgement of 5 alarm bits (ALAMQUIT.EDG)
The flipflops hold the alarms individually until acknowledgement via the VWERT. The acknowledge output is fed back to the Store input instead of the corresponding input bit.
Thereby, the acknowledge bit is reset automatically.

Alarm acknowledgement of 5 alarm bits,
which are not lost also after prolonged power failure (ALQITSAV.EDG)
Flipflops are also used for storing. In this case, the status change of the flipflop must be stored in non-volatile recipe blocks. Moreover, the flipflops must be loaded with the content of the recipe block for restoring the last status after power recovery. In VWERT, the alarms are displayed and acknowledged, if necessary.
Further display via LED, DIGOUT and INFO.

Parameter number display via texts (PRNRE.EDG)
The current parameter number (variable in VWERT) is compared with constants via EQUAL.
With equality, a bit at VWERT is set, whereby a digital text is displayed.

Two-point operation of a programmer (RUNFLIP1.EDG)
As entry of commands via the operating page is not possible with a programmer, if the relevant digital inputs were connected, the toggle key (fkey:a/m) must be used for realizing the Run/Stop order on the operating page. A monoflop generates a short pulse on the positive and negative flank. The external command (key or switch) from the control panel via d1 is also taken via a monoflop. With a key, only d1 (positive flank) is connected, with a switch, d1 and d2 are connected (positive and negative flank). The pulses are taken to a flipflop, which switches over between Run and Stop.

Weekly timer for a switch-on and a switch-off time (SCHALTUHR.EDG)
Prerequisite: options card B with clock. 3 ADSUs convert the day, hour, minute information from the status block and the switch-on/switch-off time from VWERT into a minute value. If the time from the status block is higher than the switch-on time, the flipflop is set, if the time is higher than the switch-off time, the flipflop is reset.

Recipe input via VWERT (REZEPT2.EDG)
Three configuration examples with different restrictions for operation.

The VWERT displays its own outputs, but not the actually selected recipe. Editing of an existing recipe is not possible.

The VWERT displays the selected recipe, but only, when storage was done after editing. The current values disappear after pressing the Enter key. VWERT has an additional edit function. This bit was applied to the manual input of the recipe block for output of the currently changed values on the display via the operating page. When storing and switching over to the next the recipe number (ALLP), the edit mode is reset automatically via OR and AND (due to the handling order).

31.2 Controller applications

Minimum controller configuration (C_SINGL.EDG)

Ratio controller with split-range or three-point stepping controller with position feedback (C_V_SPL.EDG)
The position feedback input is defined as a potentiometric transducer (which can be calibrated) and linked to the controller with its fail, a/m, inc, dec outputs. The use of process outputs can be configured at the controller and OUT1/OUT2.

Slave controller for testing the start of internal switching functions (C_SW_SL.EDG)

Circuit proposal for cascade configurations (KASK.EDG)
The master controller correcting variable must follow the slave set-point or process value, when the slave is switched to internal or manual mode, in order to ensure bumpless return to the automatic mode.

Programmer fragments

Analog output with 4 recipes (2*20 segments 2*10 segments) (PROG.EDG)
Selection of the recipe/program no. is via the VWERT and cannot be selected any more via the programmer operating page. The ALLP limits the input range.

Caution: the display is correct, however, the edit buffer contains the last output value, which may be too high. Entry of the preset time is via the programmer operating page. For input of the preset time via a VWERT, the digital connection (PRESET) must be provided.

Programmer with coupled outputs (PROG2.EDG)
The programmer blocks are coupled for program number, elapsed net time and RUN / RESET commands.

Programmer output with 10 programs with 20 segments (PROGRAMM.EDG)

31.3 Standard engineerings

A detailed description of the standard engineerings is available on request.

Single controller (96xxx001.edg / C9800014.edg)
Versions with standard setting
Basic versions
9407-963-00001 (switching)
- Signallers, 2-pnt., 3-pnt., 3-pnt. stepping
- Process value preprocessing (filter and characterizer)
- 2 alarms (selectable: x, xw,weff,y)
- Trend display for x, xw and weff
- Bargraph display of x and weff
- Programmer with 4 recipes each with 20 segments

9407-965-00001 (continuous) as switching version, but:
- Continuous controllers incl. split range, switching controllers with logic output
- Analog output for x, xw,weff or y2

Versions with option B function as basic version, plus:
- Operation disabling via control inputs
- Output of the programmer output signals
- Weekly timer for programmer start/stop (option B with clock)

Versions with option C function as basic version, plus:
- 3-element controller
- Override control +, or hard manual
- Galvanically isolated ratio control (x1 ® x2)
- Output of the programmer set-point
- Two additional control outputs for the programmer

Cascade controller (96xxx101.edg / C9800009edg)
Master controller
- Process value input INP5
- Process value preprocessing (filter and characterizer)
- Trends for control deviation and process value

Slave controller
- As master controller, but process value input INP1
- Adjustable controller types dependent of switching/continuous version as basic versions
- Position feedback via INP6

Flow controller (96xxx201.edg / C9800007.edg)
· Temperature and pressure-corrected flow measurement for mass flow (with or without square root extraction)
- Cycle 100 ms
- with or without control function
- trend display for flow and control deviation
- Totalization with creep flow cut-off
- Flow display up to 99.999.999
- Total reset via front panel after entry of a code number
- Counter pulse output (ones, thousands)
- Adjustable controller types dependent of switching/continuous version as basic versions

Programmer with 10 recipes (96xxx301.edg / C9800008.edg)
- 10 recipes with 20 segments
- 2 analog output
- 6 control outputs
- one controller per analog output
- operation via the analog programmer front panel
- operation disable via control input

Calorific counter (only with option B) (96xxx401.edg / C9800006 .edg)
- Flow and heating and refrigerating quantity totalizing
- Creep flow cut-off
- Pulse generation for flow and heating/refrigerating quantity (OUT4,5)
- Flow and heating quantity output as a 0/4..20mA signal
- Galvanically isolated flow outputs (option C)
- Temperature, flow alarm monitoring (option C)

Flowcalculator (possible without option B and C) (96xxx501.edg / C9800015.edg)
- Temperature and pressure-corrected flow measurement for mass flow (with or without square root extraction)
- Cycle 400 ms
- Trend display for flow, pressure and corrected flow
- Creep flow cut-off
- Pulse generation (OUT4)
- Temperature, pressure or flow monitoring (OUT5)

Index

P.O.B. 310 229, D-34058 Kassel, Germany
Printed in Germany 9499 040 50611 (05/2004)
www.pma-online.de
Subject to alterations without notice

A4