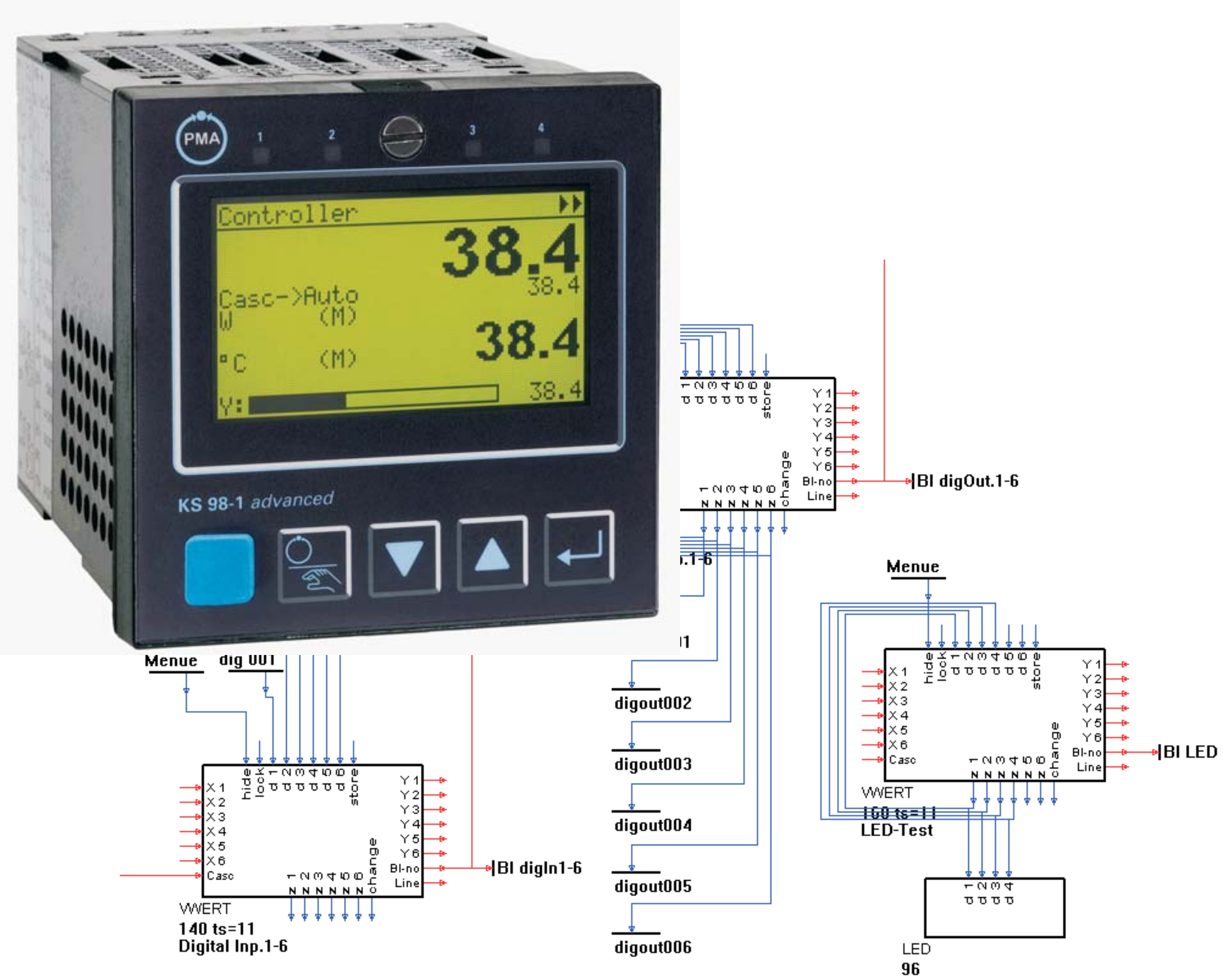
PMA Prozeß- und Maschinen-Automation GmbH

KS 98-1 Multi-function unit Engineering manual



Valid from: 1.02.2011
Order number : 9499-040-82711

A publication of:

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

Symbols used on the instrument

CE **EU conformity marking**

⚠ **Caution, Follow the operating instructions!**

Symbols in the text

⚠ **Danger of injury**

! Danger for the instrument, or of faulty function

⚡ Danger of destroying electronic components due to electrostatic discharge (ESD)

ℹ Additional information or reference to further sources of information.

☞ Important hint for avoiding frequent operator faults.

Liability and warranty

Any information and notes in these operating instructions were composed under consideration of the applicable regulations, the present state of the art and our extensive know-how and experience.

With special versions, additional ordering options or due to the latest technical modifications, the actual scope of delivery may vary from the descriptions and drawings in this manual.

For questions, please, contact the manufacturer.

⚠ **Before starting to work with the instrument and before commissioning, in particular, these operating instructions must be read carefully!**
The manufacturer cannot be held responsible for damage and trouble resulting from failure to comply with the information given in this manual.

This product may be subject to change due to improvements of the product features in the course of further development.

Copyright

This operating manual should be considered as confidential information, intended only for persons who work with the instrument.

Contraventions are subject to payment of damages. Further claims reserved.

Content

Preface

This manual consists of three parts.

I. ***Operating instructions***

II. ***Engineeringtool description***

III. ***Function block description***

Section I holds the required information for identification, mounting, connection and electrical commissioning of the unit under consideration of safety notes of the application and environmental conditions.

The basic principles of operation are explained: Controls and indicators, menu structure and navigation with the cursor, selection of sub-menus and properties as well as adjustment of e.g. s and parameters.

Section II comprises the handling of the engineering-tool, the building of a simple engineering and transmission to the KS 98-1.

Section III presents the particular fnction blocks in detail.

☞ For functional commissioning, additional descriptions are required; please, order them separately or load them from the PMA homepage: www.pma-online.de

☞ As the functions provided in KS 98-1 are composed individually for each application using an Engineering Tool ET/KS 98, entire comprehension of the operating functions requires the relevant Project description with the Engineering !

Supplementary documentation:

PROFIBUS-protocol (GB)	**9499-040-82811**
ISO 1745-protocol (GB)	**9499-040-82911**

I Operating instruction

I-1 Description

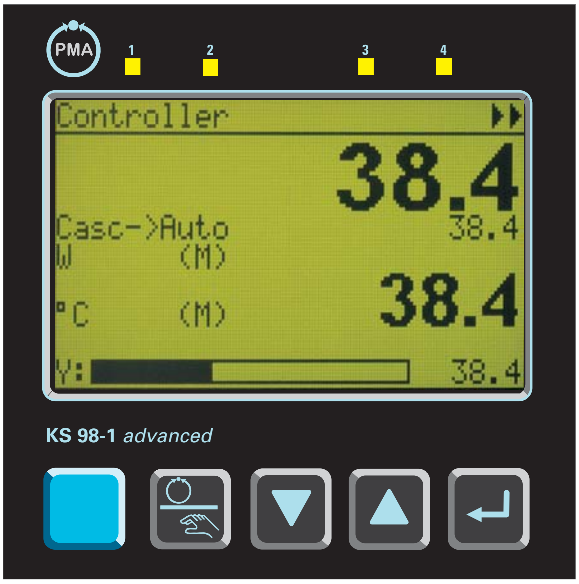


Fig. 1 Frontview

The instrument is a compact automation unit the function of which can be configured freely by means of function blocks. Each unit contains a comprehensive function library for selection, configuration, parameter setting and connection of max. 450 function blocks by means of an engineering tool.

I.e. complex mathematical calculations, multi-channel control structures and sequence controllers can be realized in a single unit.

Various operating pages are indicated via the front-panel LCD matrix display, e.g.

- Numeric input and output of analog and digital signals, values and parameters as well as
- full-graphics display of bargraphs, controllers, programmers and trends.
- Display colour red / green and direct / inverse display can be switched over dependent on event, or by operation dependent on the engineering.

Dependent on version, the basic unit is equipped with analog and digital inputs, outputs and relays.

Additional inputs and outputs are available either with option C or “modular option C”, which contains four sockets for various I/O modules.

Optionally, the unit may be equipped with 2 additional communication interfaces:

- Option B: serial TTL/RS422 interface or Profibus-DP
- Option CAN: CANopen-compliant interface port for the I/O extension with modular I/O system RM200

I-2 Safety notes

This section provides a survey of all important safety aspects: optimum protection of personnel and safe, trouble-free operation of the instrument.

Additionally, the individual chapters include specific safety notes for prevention of immediate hazards, which are marked with symbols. Moreover, the hints and warnings given on labels and inscriptions on the instruments must be followed and kept in readable condition continuously.

General

Software and hardware are programmed or developed in compliance with the state of the art applicable at the time of development, and considered as safe.

Before starting to work, any person in charge of work at the product must have read and understood the operating instructions.

The plant owner is recommended to request evidence for knowledge of the operating instructions by the personnel.

Correct use for intended application

The operating safety is only ensured when using the products correctly for the intended application. The instrument can be used as a multiple function controller for open and closed control loops in industrial areas within the limits of the specified technical data and environmental conditions.

Any application beyond these limits is prohibited and considered as non-compliant.

Claims of any kind against the manufacturer and /or his lawful agents, for damage resulting from non-compliant use of the instrument are precluded, liability is limited to the user.

User responsibility

The user is responsible:

- for keeping the operating manual in the immediate vicinity of the instrument and always accessible for the operating personnel.
- for using the instrument only in technically perfect and safe condition.

Apart from the work safety notes given in these operating instructions, compliance with the generally applicable regulations for safety, accident prevention and environment protection is compulsory.

The user and the personnel authorized by the user are responsible for perfect functioning of the instrument and for clear definition of competences related to instrument operation and maintenance. The information in the operating manual must be followed completely and without restrictions!

The user is responsible that the instrument is operated only by trained and authorized persons. Maintenance and repair may be done only by trained and specialized persons who are familiar with the related hazards.

Operation and maintenance of the instrument are limited to reliable persons. Any acts susceptible to impair the safety of persons or of the environment have to be omitted. Any persons who are under effect of drugs, alcohol or medication affecting reaction are precluded from operation of the instrument.

Instrument Safety

This instrument was built and tested according to VDE 0411 / EN61010-1 and was shipped in safe condition. The unit was tested before delivery and has passed the tests required in the test plan.

In order to maintain this condition and to ensure safe operation, the user must follow the hints and warnings given in these safety notes and operating instructions.

The unit is intended exclusively for use as a measuring and control instrument in technical installations.

The insulation meets standard EN 61010-1 with the values for overvoltage category, degree of contamination, operating voltage range and protection class specified in the operating instructions / data sheet.

The instrument may be operated within the specified environmental conditions (see data sheet) without impairing its safety.

The instrument is intended for mounting in an enclosure. Its contact safety is ensured by installation in a housing or switch cabinet.

Unpacking the instrument

Remove instrument and accessories from the packing. Enclosed standard accessories:

- – operating notes or operating instructions
- – fixing elements.

Check, if the shipment is correct and complete and if the instrument was damaged by improper handling during transport and storage.

! Warning!
If the instrument is so heavily damaged that safe operation seems impossible, the instrument must not be taken into operation.

We recommend to keep the original packing for shipment in case of maintenance or repair.

Mounting

Mounting is done in dustfree and dry rooms.

The ambient temperature at the place of installation must not exceed the permissible limits for specified accuracy given in the technical data. When mounting several units with high packing density, sufficient heat dissipation to ensure perfect operation is required.

For installation of the unit, use the fixing clamps delivered with the unit.

The sealing devices (e.g. sealing ring) required for the relevant protection type must also be fitted.

Electrical connections

All electrical wiring must conform to local standards (e.g. VDE 0100 in Germany). The input leads must be kept separate from signal and mains leads.

The protective earth must be connected to the relevant terminal (in the instrument carrier). The cable screening must be connected to the terminal for grounded measurement. In order to prevent stray electric interference, we recommend using twisted and screened input leads.

The electrical connections must be made according to the relevant connecting diagrams.

See electrical safety, page 23

Electrical safety

The insulation of the instrument meets standard EN 61 010-1 (VDE 0411-1) with contamination degree 2, overvoltage category III, working voltage 300 V r.m.s. and protection class I.

Galvanically isolated connection groups are marked by lines in the connecting diagram.

Commissioning

Before instrument switch- on, ensure that the rules given below were followed:

- Ensure that the supply voltage corresponds to the specification on the type label.
- All covers required for contact safety must be fitted.
- Before instrument switch- on, check if other equipment and / or facilities connected in the same signal loop is / are not affected. If necessary, appropriate measures must be taken.
- On instruments with protection class I, the protective earth must be connected conductingly with the relevant terminal in the instrument carrier.
- The instrument must be operated only when mounted in its enclosure.

Operation

Switch on the supply voltage. The instrument is now ready for operation. If necessary, a warm- up time of approx. 1.5 min. should be taken into account.

⚠ **Warning !**
Any interruption of the protective earth in the instrument carrier can impair the instrument safety. Purposeful interruption is not permissible.
If the instrument is damaged to an extent that safe operation seems impossible, shut it down and protect it against accidental operation.

Shut- down

For permanent shut- down, disconnect the instrument from all voltage sources and protect it against accidental operation.

Before instrument switch- off, check that other equipment and / or facilities connected in the same signal loop is / are not affected. If necessary, appropriate measures must be taken.

Maintenance and modification

The instrument needs no particular maintenance.

Modifications, maintenance and repair may be carried out only by trained, authorized persons. For this, the user is invited to contact the service.

For correct adjustment of wire-hook switches (page 21) and for installation of modular option C, the unit must be withdrawn from the housing.

⚠ **Warning!**
When opening the instruments, or when removing covers or components, live parts or terminals can be exposed.

Before carrying out such work, the instrument must be disconnected from all voltage sources.

After completing such work, re- shut the instrument and re-fit all covers and components. Check, if the specifications on the type label are still correct, and change them, if necessary.

Explosion protection

Non-intrinsically safe instruments must not be operated in explosion-hazarded areas. Moreover, the output and input circuits of the instrument / instrument carrier must not lead into explosion-hazarded areas.

I-3 Technical data

General

Housing

Plug-in module, inserted from front.

Material: Makrolon 9415, flame-retardant, self-extinguishing

Flammability class: UL 94 VO

Front-panel / display

160 x 80-dot LCD matrix display

4 LEDs, 4 foil-covered keys

Front interface port (standard)

Front-panel socket for PC adapter
(see „Accessories“ page 19).

Protection mode

(to EN 60 529, DINVDE 0470)

Front: IP 65

Housing: IP 20

Terminals: IP 00

Safety tests

Complies with EN 61 010-1

- Overvoltage category: III
- Contamination class: 2
- Working voltage range: 300 VAC
- Protection class: I

DIN EN 14597

The device may be used as temperature control and limiting equipment according to DIN EN 14597.

CE marking

Meets the EuropeanDirectives regarding „Electromagnetic Compatibility“ and „Low-voltage equipment“(→ page 23)

The unit meets the following European guidelines:

❑ Electromagnetic compatibility (EMC): 89/336/EEC (as amended by 93/97/EEC)

❑ Electrical apparatus (low-voltage guideline): 73/23/EEC (as amended by 93/68/EEC).

Evidence of conformity is provided by compliance with EN 61326-1 and EN 61010-1

cULus certification

cULus-certification

(Type 1, indoor use) File: E 208286

⚠ **RC protective circuitry must be fitted with inducti-ve load !**

Electrical connections

Screw terminals for conductor cross-section from 0,5 - 2,5 mm^2

Mounting method

Panel mounting with 4 fixing clamps at top/bottom

Mounting position

Not critical

Weight

Approx. 0.75 kg with all options

Environmental conditions

Permissible temperatures

For operation: 0...55°C

For specified accuracy: 0...60°C, optionC

For UL-Devices: Operation and protected operation 0...50°C

Storage and transport: –20...60 °C
Temperature effect: ≤ 0.15% / 10 K

Climatic category

KUF to DIN 40 040

Relative humidity: ≤ 75% yearly average, no condensation

Shock and vibration

Vibration test Fc: To DIN 68-2-6 (10...150 Hz)

Unit in operation: 1g or 0.075 mm,

Unit not in operation: 2g or 0.15 mm

Shock test Ea:To DIN IEC 68-2-27 (15g, 11 ms)

Connections

Depending on version and selected options, the following inputs and outputs are available:

	DI	DO	AI	AO
4 relays or	di1* di2*	OUT1 OUT2 OUT4 OUT5	INP1 INP5 INP6	–
2 relays + 2 AO	di1* di2*	OUT4 OUT5	INP1 INP5 INP6	OUT1 OUT2
OPTION B	di3 di4 di5 di6 di7	do1 do2 do3 do4	–	–
OPTION C* or	di8 di9 di10 di11 di12	do5 do6	INP3 INP4	OUT3
modular OPTION C*	Depending on module type			

* Not available with option CAN!

Inputs

Universal input INP1

Limiting frequency: fg = 1 Hz, Measurement cycle: 200 ms

Thermocouples

according to DIN IEC 584

Type	Range	Error	Resolution
L	–200...900°C	≤ 2K	0,05 K
J	–200...900°C	≤ 2 K	0,05 K
K	–200...1350°C	≤ 2 K	0,072 K
N	–200...1300°C	≤ 2 K	0,08 K
S	–50...1760°C	≤ 3 K	0,275 K
R	–50...1760°C	≤ 3 K	0,244 K
B 1)	(25)400...1820°C	≤ 3 K	0,132 K
T	–200.. .400°C	≤ 2 K	0,056 K
W(C)2)	0...2300°C	≤ 2 K	0,18 K
E	–200... 900°C	≤ 2 K	0,038 K

* 1) from 400 °C

* 2) W5Re/W26Re

With linearization (temperature-linear in °C or °F)

Input resistance: ≥ 1MΩ

Cold-junction compensation (CJC): built in

Input circuit monitor:

Current through sensor: ≤ 1 μA

Reverse-polarity monitor is triggered at 10 °C below span start.

Additional error of internal CJC

≤ 0.5 K per 10 K terminal temperature External temperature selectable: 0...60 °C or 32...140 °F

Resistance thermometer

Pt 100 to DIN IEC 751, and temperature difference 2 x Pt 100

Range	Error	Resolution
–200.0...250.0 °C	≤ 0.5 K	0.024 K
–200.0...850.0 °C	≤ 1.0 K	0.05 K
2 x –200.0...250.0 °C	≤ 0.5 K	0.024 K
2 x –200.0...250.0 °C	≤ 0.1 K	0.05 K

Linearization in °C or ° F

Two wire connection with compensation or three-wire connection without.

Two-wire connection with lead resistance adjustment.

Lead resistance: ≤ 30 Ω per lead

Sensor current: ≤ 1 mA

Input circuit monitoring for sensor/lead break, and lead short circuit.

Potentiometric transducer

R_{total} incl. 2 x R_L	Error	Resolution
0...500 Ω	≤ 0.1 %	≤ 0.02 Ω

Resistance-linear

Sensor current: ≤ 1 mA

Matching/scaling with sensor connected.

Input circuit monitoring for sensor/lead break, and lead short circuit.

Resistance input

Range	Error	Resolution
0...250 Ω	≤ 0.25 Ω	< 0.01 Ω
0...500 Ω	≤ 0.5 Ω	< 0.02 Ω

Direct current 0/4...20mA

Range	Error	Resolution
0/4...20 mA	≤ 0.1 %	≤ 0.8 mA

Input resistance: 50 Ω

Input circuit monitor with 4...20mA: triggered when I ≤ 2 mA

Direct voltage

Range	Error	Resolution
0/2...10 V	≤ 0.1 %	≤ 0.4 mV

Input resistance ≥ 100 kΩ

Signal input INP5

Differential amplifier input

Up to 6 controllers can be cascaded, if there is no other galvanic connection between them. If there is, only 2 inputs can be cascaded.

Direct current and voltage

Technical data as for INP1, except for:

Limiting frequency: = 0.25 Hz, Measurement cycle: 800 ms

Input resistance (voltage): ≥ 500 κΩ

Signal input INP6

Limiting frequency: = 0.5 Hz, Measurement cycle: 400 ms

Potentiometric transducer

Technical data as for INP1, except for:

R_{total} inkl. 2 x RL	Error	Resolution
0...1000 Ω	≤ 0.1 %	≤ 0.04 Ω

Direct current 0/4...20 mA

Technical data as for INP1

Signal inputs INP3, INP4 *(option C)*

Galvanically-isolated differential amplifier inputs.
Limiting frequency: fg = 1 Hz, Measurement cycle: 100 ms

Direct current

Technical data as for INP1, except Ri = 43 Ω

Control inputs di1...di12

di1, di2: standard

di3...d7: Option B

di8...di12: Option C

Opto-coupler

Nominal voltage: 24 VDC, external

Current sink (IEC 1131 Type 1)

Logic „0“ (Low): –3...5 V, Logic „1“ (High): 15...30 V

Current demand: approx. 6 mA (for galvanic connections and isolation see page 25).

Built-in transmitter supply (optional)

Can be used to energize a two-wire transmitter or up to 4 opto-coupler inputs.Galvanically isolated.

Output: ≥ 17.5 VDC, max. 22 mA

Short-circuit proof.

Factory setting

The transmitter supply is available at terminals A12 and A14

☞ See page 25, Configuration see page 21

Outputs

Relay outputs (OUT4, OUT5)

Relays have potential-free change-over contacts.

Max. contact rating:

500 VA, 250 V, 2 A with 48...62 Hz, cosφ ≥ 0,9

Minimum rating: 12 V, 10 mA AC/DC

Number of switching cycles electrical: for I = 1A/2A ≥ 800,000 / 500,000 (at ~ 250V / (resistive load))

! **When connecting a contactor to a relay output, RC protective circuitry to specification of the contactor manufacturer is required. Varistor protection is not recommended!**

Outputs OUT1, OUT2

Relay or current/logic signal, depending on version:

OUT1, OUT2 as current outputs

Galvanically isolated from the inputs

0/4...20 mA, selectable

Signal range: 0...22 mA

Resolution: ≤ 6 µA (12 bits)

Error: ≤ 0.5%

Load: ≤ 600 Ω

Load effect: < 0.1%

Limiting frequency: approx. 1 Hz Output cycle: 100ms

OUT1, OUT2 as logic signal

0 / ≥ 20 mA with a load ≥ 600 Ω

0 / > 12 V with a load ≥ 600 Ω

Output OUT3 (Option C)

Technical data as for OUT1, OUT2, as current output

Control outputs do1..do6

Galvanic isolated Opto-coupler outputs, galvanic isolation see page 25 and text.

Grounded load: common positive control voltage

Switch rating: 18...32 VDC; Imax ≤ 70 mA

Internal voltage drop: ≤ 0.7V with I_{max}

Protective circuit: thermal, switches off with overload.

Supply 24 V DC external

Residual ripple ≤ 5%$_{ss}$

Modular Option C

Each module has two channels which can be configured independently.

A/D-Converter

Resolution: 20,000 (50Hz) or 16,667 (60Hz) steps for the selected measuring range.

Conversion time: 20ms (50Hz) or 16.7ms (60Hz).

D/A-Converter

Resolution: 12 Bit

Refresh-Rate: 100 ms

Cut-off frequency

Analog: fg=10Hz

Digital: fg=2Hz

Measurement cycle: 100 ms per module

R_INP Resistive input module

(9407-998-0x201)

Connection method: 2, 3 or 4-wire circuit (with 3 und 4-wire connection, only one channel can be used).

Sensor current:

≤ 0,25mA

Resistance thermometer

Type	Range °C	Overall error	Resolution K/Digit
Pt100	-200...850°C	≤ 2 K	0,071
Pt100	-200...100°C	≤ 2 K	0,022
Pt1000	-200...850°C	≤ 2 K	0,071
Pt1000	-200...100°C	≤ 2 K	0,022
Ni100	-60...180°C	≤ 2 K	0,039
Ni1000	-60...180°C	≤ 2 K	0,039

Linearization: in °C oder °F

Lead resistance
Pt (-200...850°C): ≤ 30 Ω per lead
Pt (-200...100°C), Ni: ≤ 10 Ω per lead

Lead resistance compensation
3 and 4-wire connection: not necessary.

2-wire connection: compensation via the front with short-circuited sensor. The calibration values are stored in a non-volatile memory.

Lead resistance effect
3 and 4-wire connection: negligible

Sensor monitoring
Break: sensor or lead. Short-circuit: triggers at 20 K below measuring range.

Resistance / Potentiometer

Range R_{ges} / Ω	Overall error	Resolution Ω/Digit
0...160 Ω	≤ 1%	0,012
0...450 Ω	≤ 1%	0,025
0...1600 Ω	≤ 1%	0,089
0...4500 Ω	≤ 1%	0,025

Characteristic: resistance-linear

Lead resistance or 0%/100% compensation:
via the front with short-circuited sensor. The calibration values are stored in a non-volatile memory.

- Variable resistance (only 2-wire connection): compensation for 0%
- Potentiometer: compensation for 0% and 100%

Lead resistance effect

3 and 4-wire connection: negligible

Sensor monitoring:

break: sensor or lead

TC_INP Thermocouple, mV, mA Module

(9407-998-0x211)

Thermocouples

To DIN IEC 60584 (except L, W/C, D)

Type	Range	Overall error	K/Digit
L	-200...900°C	≤ 2 K	0,080
J	-200...900°C	≤ 2 K	0,082
K	-200...1350°C	≤ 2 K	0,114
N	-200...1300°C	≤ 2 K	0,129
S	-50...1760°C	≤ 3 K	0,132
R	-50...1760°C	≤ 3 K	0,117
B*	(25) 400...1820°C	≤ 3 K	0,184
T	-200...400°C	≤ 2 K	0,031
W(C)	0...2300°C	≤ 2 K	0,277
D	0...2300°C	≤ 2 K	0,260
E	-200...900°C	≤ 2 K	0,063

* (1) Values apply from 400°C

Linearization: in °C or °F
Linearity error: negligible

Input resistance: ≥1MΩ

Temp. compensation (CJC): built in

Error: ≤ 0.5K/10K

External CJC possible: 0...60 °C or. 32...140 °F

Source resistance effect: 1mV/kΩ

Sensor monitoring:

Sensor current: ≤ 1μA
Reversed polarity monitor: triggers at 10K below measuring range

mV input

Range	Overall error	Resolution
0...30 mV	≤ 45 mV	1,7 mV
0...100 mV	≤ 150 mV	5,6 mV
0...300 mV	≤ 450 mV	17 mV

Input resistance: ≥1MΩ

Break monitoring: built in. Sensor current: ≤1μA

mA input

Range	Overall error	Resolution
0/4...20 mA	≤ 40 μA	2 μA

Input resistance: 10 Ω

Break monitor:<2μA (only for 4...20 mA)

Over range monitor: >22mA

U_INP High-impedance voltage input

(9407-998-0x221)

Range	Overall error	Resolution mV/Digit
-50...1500 mV	≤ 1,5 mV	0,09
0...10 V	≤ 10 mV	0,56

Characteristic: voltage-linear

Input resistance: >1GΩ

Source resistance effect: ≤ 0,25 mV/MΩ

Sensor monitoring: none

U_OUT Voltage output module

(9407-998-0x301)

Signal ranges: 0/2...10V, -10...10V configurable

Resolution: approx.. 5.4 mV/digit

Load: ≥ 2 kΩ

Load effect : ≤ 0.1%

I_OUT Current output module

(9407-998-0x311)

Signal ranges: 0/4...20mA, -20...20mA configurable by channel

Resolution: ca. 11μA/Digit

Load: ≤ 400 Ω

Load effect: ≤ 0.1%/100Ω

DIDO Digital I/O Module

(9407-998-0x401)

Number of channels: 2 (configurable as input or output by channel)

Input

Current sink: to IEC 1131 Type 1)
Logic „0“: -3...5V, Logic „1“: 15...30V

Measurement cycle: 100 ms

Galvanic isolation : via opto-couplers

Nominal voltage: 24 VDC external

Input resistance: 5 kΩ

Output

Grounded load (common positive control voltage)

Switch rating: 18...32 VDC; ≤70 mA

Internal voltage drop: ≤ 0.7 V

Refresh-Rate: 100 ms

Galvanic isolation : via opto-couplers

Protective circuit: thermal, switches off with overload.

F_INP Frequency/counter Module

(9407-998-0x411)

Current sink: to IEC 1131 Type 1
Logic „0“: -3...5V
Logic „1“: 15...30V

Galvanic isolation: via Opto-couplers

Nominal voltage: 24 VDC external

Input resistance: 12 kΩ

Selectable functions:

- Control input (2 channels)
- Pulse counter (2 channels)
- Frequency counter (2 channels)
- Up/down counter (1 channel)
- Quadrature counter (1 channel)

Frequency range: ≤ 20 kHz

Signal shape: any (square 1:1 with 20kHz)

Gate time: 0.1...20s adjustable (only relevant with frequency measurement)

Influencing factors

Temperature effect: ≤ 0,1%/10K

Supply voltage: negligible

Common mode interference: negligible up to 50 Vrms

Series mode interference: negligible up to 300 mVrms (TC), 30 mVrms (RT), 10 Vrms (U), 5 Vrms (F)

CAN I/O extension

The unit offers a port conforming to CANopen for connection of the RM 200 system and of KS 800 or additional KS 98-1 units with max. five CAN nodes.

☞ Control inputs di1 and di2 are not available !

☞ Modular option C is not available

Power supply

Dependent on version:

Alternating current

90...250 VAC

Frequency: 48...62 Hz

Power consumption: approx. 17,1 VA; 9,7 W (max. configuration).

Universal current 24 V UC

24 VAC, 48...62 Hz/24 VDC

Tolerance: +10...–15% AC 18...31.2 VDC

Power consumption AC: approx. 14.1 VA; 9.5 Ω

DC: 9,1 W (max. Configuration).

Behaviour after power failure

Storage in non-volatile EEPROM for structure, configuration, parameter setting and adjusted setpoints.

Storage in a capacitor-buffered RAM (typ. > 15 minutes) for data of time functions (programmers, integrators, counters, ...)

Real-time clock(Option B, RS 422)

Buffer capacitor provides back-up for at least 2 days (typical).

Bus interface (Option B)

TTL and RS422 / 485-interface

Galvanically isolated, choice of TTL or
RS 422/485 operation.

Number of multi function unitsper bus:

With RS 422/485: 99

With TTL signals: 32 interface modules (9404 429 980x1). Address range ((00...99)See Documentation 9499-040-82911) .

PROFIBUS-DP interface

According to EN 50 170, Vol. 2.

Reading and writing of all process data, parameters, and configuration data.

Transmission speeds and cable lengths

automatic transm. speed detection, 9,6 kbit/s ...12 Mbit/s

Addresses

0...126 (factory setting: 126).Remote addressing is possible.

Other functions

Sync and Freeze

Terminating resistors

Internally selectable with wire-hook switches.

Cable

According to EN 50 170, Vol. 2 (DIN 19 245T3).

Required accessories

Engineering Set PROFIBUS-DP, consisting of:

- GSD file, Type file
- PROFIBUS manual (9499-040-82911)
- Function block(s) for Simatic S5/S7

Electromagnetic compatibility

Complies with EN 50 081-2 and EN 50 082-2.

Electrostatic discharge

Test to IEC 801-2,

8 kV air discharge

4 kV contact discharge

High-frequency interference

Test to ENV 50 140 (IEC 801-3)

80...1,000 MHz, 10 V/m

Effect: ≤ 1%

HF interference on leads

Test to ENV 50141 (IEC 801-6)

0.15...80 MHz, 10 V

Effect: ≤ 1%

Fast pulse trains (burst)

Test to IEC 801-4

2 kV applied to leads for supply voltage, and signal leads.
Effect: ≤ 5%resp. restart

High-energy single pulses (surge)

Test to IEC 801-5

1 kV symmetric or 2 kV asymmetric on leads for supply voltage.

0.5 kV symmetric or 1 kV asymmetric on signal leads.

I-4 Versions

The instrument version results from the combination of various variants from the following table.

Fig. 2

Please mind footnotes!
KS98-1 with screw terminals only!

K S 9 8 - 1 □ □ - □ □ 0 □ □ - □ 0 0

BASIC UNIT	KS 98 Standard	0
	KS 98 with transmitter power supply	1
	KS 98 with CANopen I/O [1]	2
POWER SUPPLY AND CONTROL OUTPUTS	90...250V, AC 4 relays	0
	24V UC, 4 relays	1
	90...250V AC, 2 relays+ 2 mA/logic	4
	24V UC, 2 relays+ 2 mA/logic	5
OPTION B INTERFACE	no interface	0
	TTL interface + di/do	1
	RS422 + di/do + clock	2
	PROFIBUS DP + di/do	3
*OPTION **C** (standard)*	no extensions	0
	INP3, INP4, OUT3, di/do	1
*OPTION **C** (modular)*	Motherboard without modules [2]	3
	Motherboard ordered modules inserted [2]	4
SETTING	Standard configuration	0
	Customer-specific configuration	9
	Operating instruction [3]	0
APPROVALS	Standard (CE certification)	0
	cULus approval	U
	DIN EN 14597 certified	D

[1] Not possible with Modular Option C!
RM 200 not included in cULus approval !

[2] Not possible with CANopen ! I/O modules must be ordered separately!
Mind possible combinations and power limitations; → Text!

[3] Detailled system manual can be ordered separately or downloaded (www.pma-online.de)

I-4.1 I/O-Modules

Can be installed on instruments with modular option C basic card.

Fig. 3 Table of I/O-module Versions

9 4 0 7 9 9 8 0 □ □ □ 1

	Singular order(separate delivery)	**0**
SOCKETS Module group 1	In KS 98-1 pinned to socket 1[3]	**1**
	In KS 98-1 pinned to socket 2[3]	**2**
Module group 2	In KS 98-1 pinned to socket 3[3]	**3**
	In KS 98-1 pinned to socket 4[3]	**4**
ANALOG INPUTS	R_INP: Pt100/1000, Ni100/1000, resistor	**2 0**
	TC_INP: Thermocouple, mV, 0/4...20mA	**2 1**
	U_INP: -50...1500mV (z.B. Lambda-probe), 0...10V	**2 2**
ANALOG OUTPUTS	U_OUT: Voltage output	**3 0**
	I_OUT: Current output [4]	**3 1**
DIGITAL SIGNALS	DIDO: digitale in-/outputs	**4 0**
	F_INP: Frequency-/counter inputs	**4 1**

3) Please note on your order: "mounted in KS98-1 in orderposition X"
4) Max. 1 current output module

I-4.2 Ex-factory setting

All delivered units permit operation, parameter setting and configuration via the front-panel keys.

Instruments with default setting are delivered with a test engineering which permits testing of the basic instrument inputs and outputs (no I/O extension) without auxiliary means.

☞ This engineering is not suitable for controlling a system. For this purpose, a customer-specific engineering is required (see versions, section: Setting)

Instruments with "setting to specification" are delivered complete with an engineering. Code no. KS 98-1-1xx-xx0**9**x-xxx is specified on the type label.

Accessories delivered with the instrument:

Operating manual, 4 fixing clamps

I-4.3 Auxiliary equipment

Engineering Tool ET/KS 98

Simulation SIM/KS 98-1

PC-Adapter

– Adapter cable for connecting a PC (Engineering Tool) to the front-panel interface socket of the KS 98-1.

– Updates and Demos on the PMA- Homepage (www.pma-online.de)

I-5 Mounting

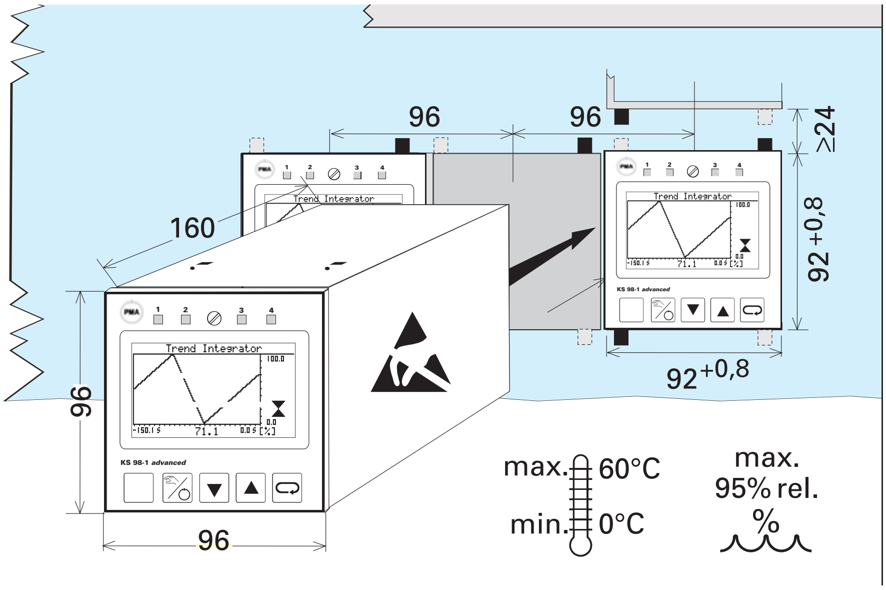
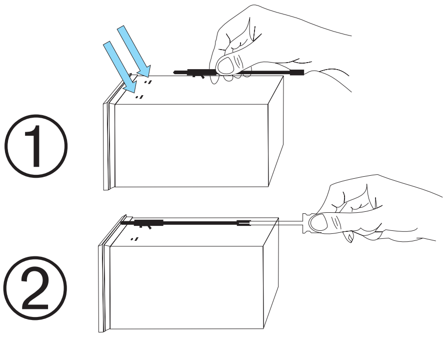


Fig. 4 Mounting

The instrument must be installed as described below. Required dimensions of the control cabinet cut-out and minimum clearances for installation of further units are shown in the drawing.

For mounting, insert the unit into the control cabinet or cabinet door cut-out. Push the instrument module fully home and mount it firmly by means of the locking screw. Four fixing clamps are delivered with the unit.

Fig. 5 Inserting the locking screws



① These clamps have to be fitted in the instrument from the control cabinet inside: 2 at top and bottom.

② Now, the threaded pins of the fixing clamps can be screwed against the control cabinet from inside.

! A rubber seal is fitted on the rear of the instrument front panel (in mounting direction).
This rubber seal must be in perfect condition, flush and cover the cut-out edges completely to ensure tightness!

cULus certification

☞ For compliance with cUus certificate, the technical data at the beginning must be taken into account (see technical data, page 12 .

I-5.1 Function of wire-hook switches

For closing the wire-hook switches, release the locking screw, withdraw the instrument module from the housing and close the wire-hook switch. Re-insert the instrument and lock it.

Ex-factory setting	
S	open
DP	open - terminating resistor not active
CAN	open - terminating resistor not active
TPS	A 14/12

The unit contains electrostatically sensitive components. Comply with rules for protection against ESD during mounting.

Wire-hook switch S:

The switching status is signalled by function STATUS and can be used in the engineering for blocking e.g. operating pages and other settings.

Wire-hook switch for PROFIBUS DP (Option B):

The bus terminating resistor can be connected by 2 wire-hook switches (DP) in KS98-1.Both wire-hook switches must always be open or closed .

Wire-hook switch for CAN bus: (only Option CANbus):

Both ends of the CAN bus must be terminated (wire-hook switch closed).

Wire-hook switch for transmitter supply

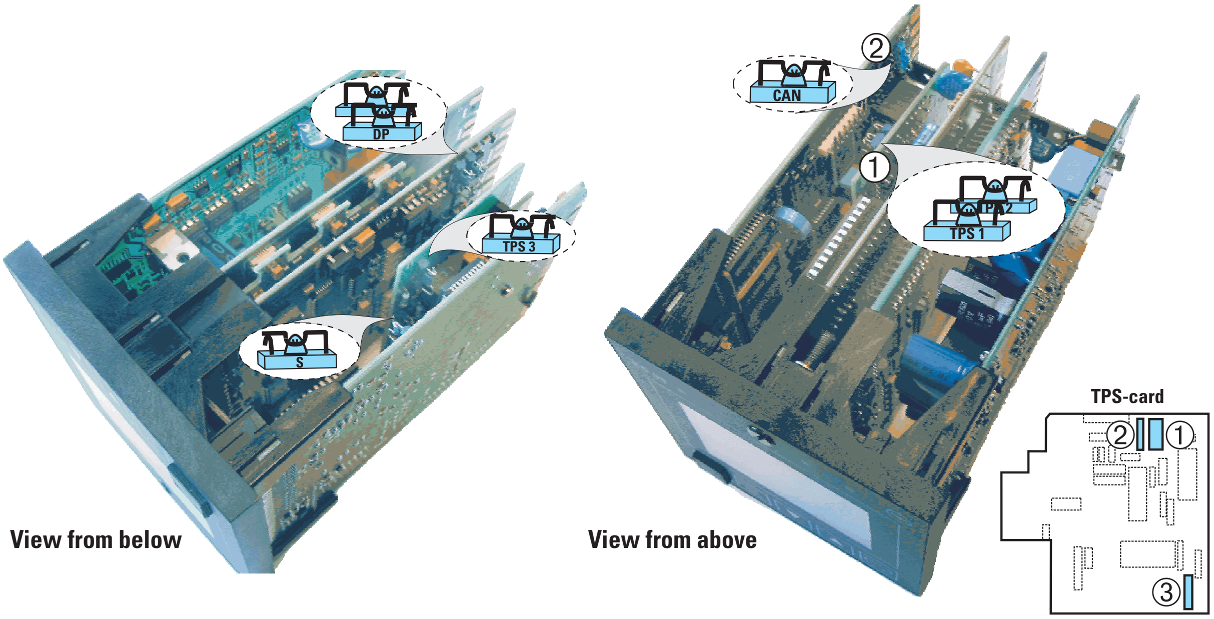
Versions (KS98-11x-xxxxx) with transmitter supply are provided with a potential-free supply voltage for energization of a 2-wire transmitter, or of max. 4 control inputs.

The output connections can be made to terminals A4(+) - A1(-) by means of 3 wire-hook switches.

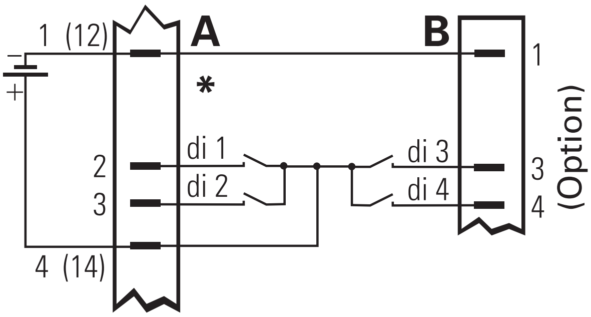
If A14/A12 is used for di1/di2, A12 must be linked with A1.

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

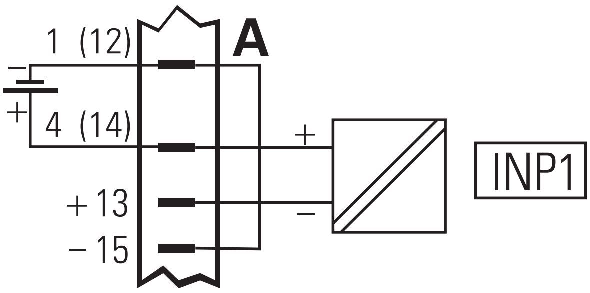
Fig. 6 Position of wire-hook switches



Energizing digital inputs (e.g. di1...di4)



Connection 2-wire-transducer (e.g. INP1)



I-5.2 Retro-fitting and modific. of I/O-ext. (*watch connecting diagram)

Only for instruments with modular option C!

The instrument contains electrostatically sensitive components. Original packing protects against electrostatic discharge (ESD), transport only in original packing.

– When mounting, rules for ESD protection must be taken into account.

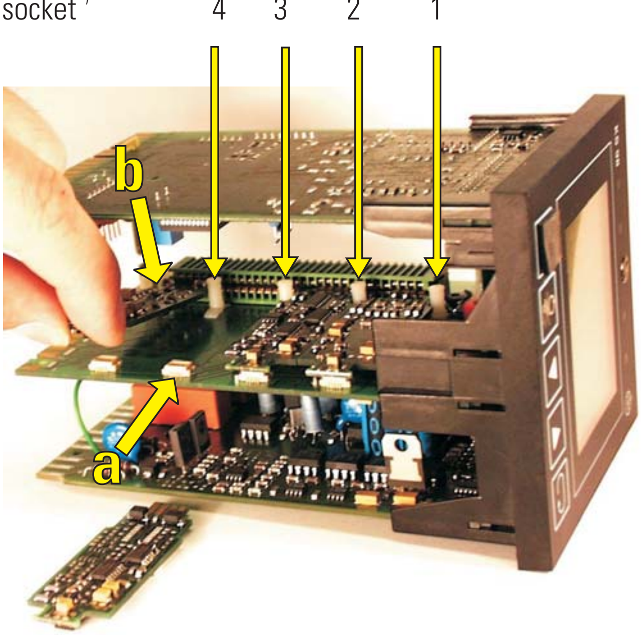
Connection:
KS 98-1 engineering must be taken into account, because it determines pin allocation and signification of connections.

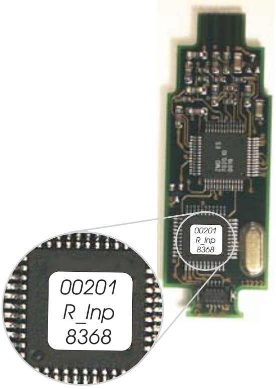
Moreover, the rules for the performance limits must be followed (see manual → 9499-040-82711).

Mounting
After releasing the locking screw, withdraw the KS 98-1 module from the housing.

a Insert the module into the required socket with the printed label pointing down-wards into the green connector and click it in position in the small, white contact ***b*** at the top.

Fig. 7 Mounting of i/o modules





The various modules can be identified by the printed label.
The last 5 digits of the order number are given in the upper line.

I-5.3 I/O extension with CANopen

The unit offers a CANopen-compliant interface port for connection of the RM 200 system, KS 800 or additional KS 98-1 units with max. five CAN nodes. See installation notes in the CANopen system manual (9499-040-62411).

I-6 Electrical connections - Safety hints

⚠ **Following the safety notes starting on page 6 is indispensable!**

The installation must be provided with a switch or power circuit breaker which must be marked accordingly. The switch must be located near the instrument and readily accessible for the operator.

⚠ **When the instrument module is withdrawn from the housing, protection against dropping of conducting parts into the open housing must be fitted.**

⚠ **The protective earth terminal (P3) must be connected to the cabinet ground. This applies also to 24V supply.**

I-6.1 Electromagnetic compatibility

European guideline 89/336/EEC. The following European standards are met: EN 61326-1.

The unit can be installed in industrial areas (risk of radio interference in residential areas). A considerable increase of the electromagnetic compatibility is possible by the following measures:

- Installation of the unit in an earthed metal control cabinet.
- Keeping power supply cables separate from signal and measurement cables.
- Using twisted and screened measurement and signal cables (connect the screening to measurement earth).
- Providing connected motor actuators with protective circuitry to manufacturer specifications. This measure prevents high voltage peaks which may cause trouble for the instrument.

I-6.2 Measurement earth connection

Measurement earth connection is required for protection against interference. External interference voltages (incl. radio frequencies) may affect the perfect function of the device.

To ensure protection against interference, the measurement earth must be connected to cabinet ground.

☞ Terminals A11(measurement earth) and P3 (protective earth) must be connected to cabinet ground via a short cable (approx. 20 cm)! The protective earth conductor of the power supply cable must also be connected to this earth potential (cabinet ground). → See also diagram on page 24

I-6.3 RC protective circuitry

Load current free connections between the ground potentials must be realized so that they are suitable both for the low-frequency range (safety of persons, etc.) and the high-frequency range (good EMC values). The connections must be made with low impedance.

All metal grounds of the components installed in the cabinet ① or in the cabinet door ② must be screwed directly to the sheet-metal grounding plate to ensure good and durable contact.

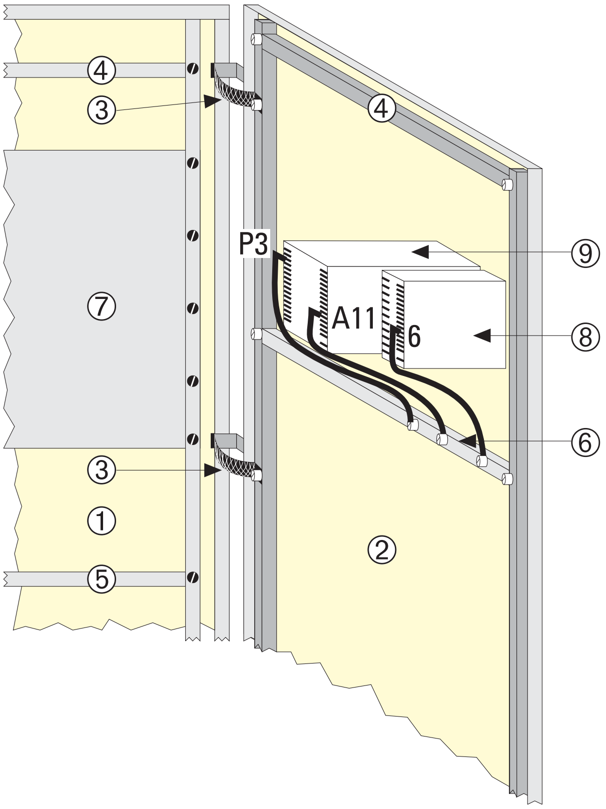
In particular, this applies to earthing rails ④, protective earth rail ⑤, mountingplates for switching units ⑦ and door earthing strips ⑥.Controllers KS40/50/90 ⑧ and KS98-1 ⑨ are shown as an example for earthing.

The max. length of connections is 20 cm (see relevant operating instructions).

Generally, the yellow/green protective earth is too long to provide a high-quality ground connection for high-frequency interferences.

Braided copper cables ③ provide a high frequency conducting, low-resistance ground connection, especially for connecting cabinet ① and cabinet door ②.

Fig. 8 Protective circuitry



Because of the skin effect, the surface rather than the cross section is decisive for low impedance. All connections must have large surfaces and good contact. Any lacquer on the connecting surfaces must be removed.

! Due to better HF properties, zinc-plated mounting plates and compartment walls are more suitable for large-surface grounding than chromated mounting plates.

I-6.4 Galvanic isolations

Galvanically isolated connection groups are marked by lines in the connecting diagram.

- Measuring and signal circuits: functional isolation up to a working voltage ≤ 33 VAC / 70 VDC against earth (to DIN 61010-1; dashed lines).
- Mains supply circuits 90...250 VAC, 24 VUC: safety isolation between circuits and against earth up to a working voltage ≤ 300 Vr.m.s. (to EN 61010-1; full lines).
- Instruments with I/O extension modules (KS98-1xx-x3xxx and KS98-1xx-x4xxx): sockets 1-2 and 3-4 are galvanically isolated from each other and from other signal inputs/outputs (functional isolation).

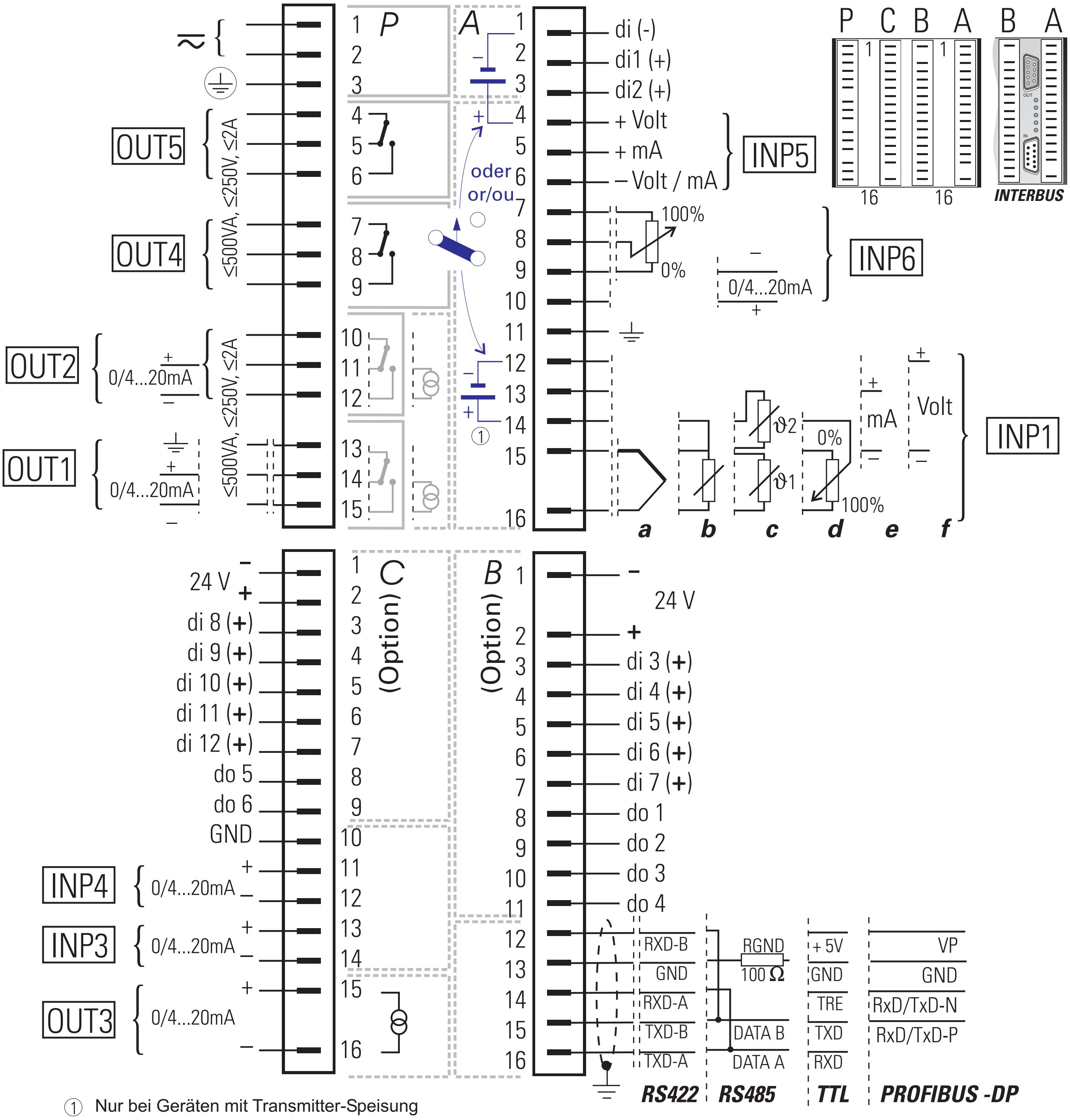
I-6.5 General connecting diagram

⚠ **The max. permissible working voltage on input and signal circuits is 33 VAC / 70 VDC against earth ! Otherwise, the circuits must be isolated and marked with warning label for “contact hazard”.**

⚠ **The max. permissible working voltage on mains supply circuits may be 250 VAC against earth and against each other ! Only on versions with transmitter power supply (factory setting: connection across terminals A12-A14)**

⚠ **Additionally, the units must be protected by fuses for a max. power consumption of 12,3VA/7,1W per instrument individually or in common (standard fuse ratings, min. 1A)!**

Fig. 9 connecting diagram



① Nur bei Geräten mit Transmitter-Speisung

① For instruments with built-in transmitter power supply

① Seulement pour les appareils avec alimentation transmetteur

☞ For instruments with modular option C → see connecting diagram on page 28

⚠ **The protective earth must be connected also with 24 V DC / AC supply (see safety notes on page 23). The polarity is uncritical.**

⚠ **Only on versions with transmitter power supply (factory setting: connection to terminals A12-A14). Connection of the transmitter power supply is determined by the wire-hook switch → page 21.**

I-6.6 Analog inputs

Thermocouples

see general connecting diagram on page 25. No lead resistance adjustment.

Internal temperature compensation:

compensating lead up to the instrument terminals.

With AINP1, `STK = int.CJC` must be configured.

External temperature compensation:

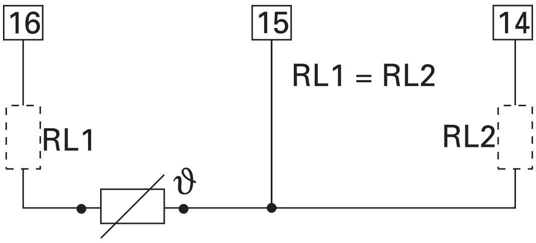
Use separate cold junction reference with fixed reference temperature.

Compensating lead is used up to the cold junction reference. Copper lead between reference and instrument. With AINP1, `STK = ext.CJC` and `TKref =` reference temperature must be configured.

Resistance thermometer

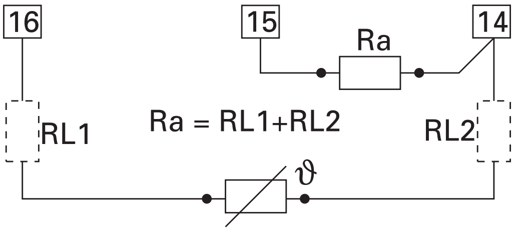
Pt 100 in 3-wire connection.

Lead resistance adjustment is not necessary, if RL1 is equal RL2.



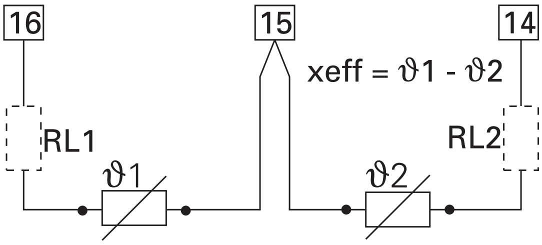
Resistance thermometer

Pt 100 in 2-wire connection. Lead resistance adjustment is necessary: Ra must be equal to RL1 + RL2.



Two Resistance thermometer

Pt100 for difference. Lead resistance compensation: proceed as described in chapter calibration page 27 - I-9.4.



Resistance transducer

Measurement calibration → proceed as described in chapter calibration page 27 - I-9.4.

Standard current signals 0/4...20 mA

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

Standard voltage signals 0/2...10V

Input resistance: ≥ 100 kΩ
(Voltage input module U_INP: >1 GΩ),
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 to A9.

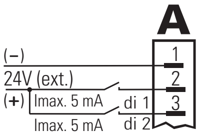
⚠ **The inputs INP1 / INP6 are interconnected (common reference potential). This must be taken into account if both inputs must be used for standard current signal. If necessary, galvanic isolation should be applied.**

I-6.7 Digital in- and outputs

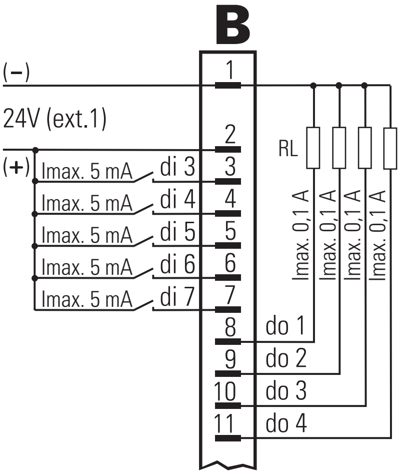
The digital inputs and outputs must be energized from one or several external 24 V DC sources. Power consumption is 5 mA per input. The max. load is 70 m A per output.

Examples:

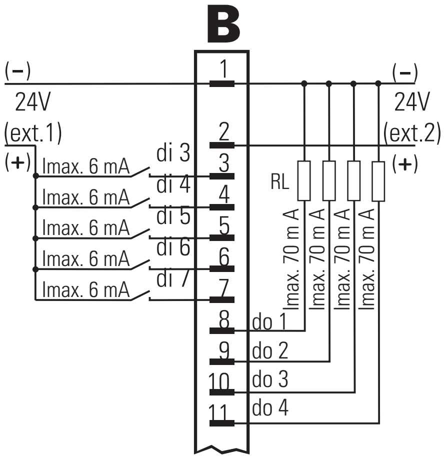
Digital inputs (connector A)



Digital inputs and outputs at one voltage source (e.g. connector B) 70mA!



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



I-6.8 Connecting diagram i/o-modules

(Modular Option C)

☞ CAN and modular option C are mutually precluding.

The inputs and outputs of multi-function unit KS98-1 can be adapted to the individual application by means of "modu - lar optionC". The supporting card is firmly installed in the unit.

The card contains four sockets for various I/O modules which can be combined, whereby the positions of the various connection types are dependent on engineering.

The KS98-1 programming personnel must provide a connecting diagram corresponding to the block diagram (→ page) for installation of the device.

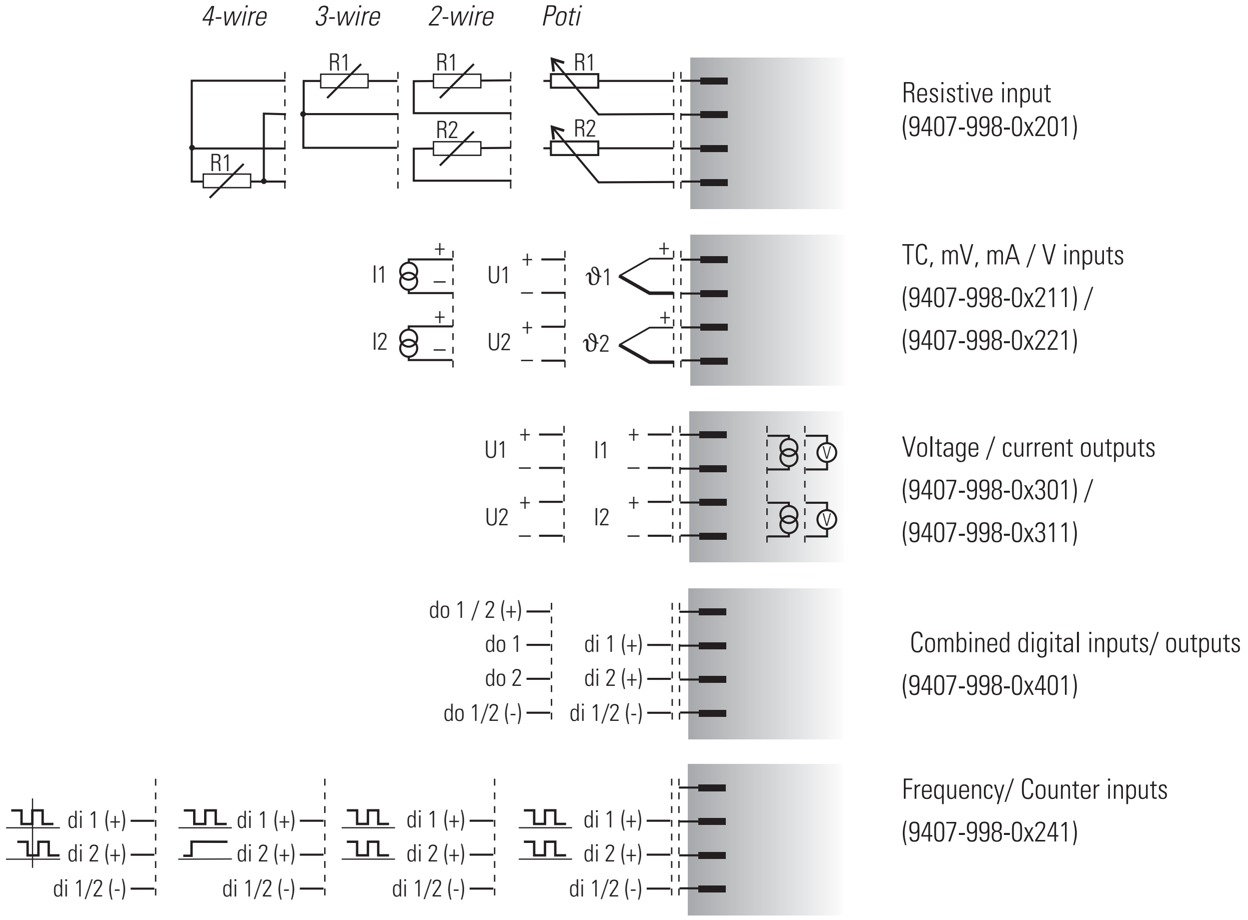


Fig. 10 Quadruple counter up/down counter 2 x counter and 2 x frequency

I-7 Commissioning

Before switching on the instrument, ensure that the following points are taken into account:

- The supply voltage must correspond to the specification on the type label!
- All covers required for contact protection must be fitted.
- Before operation start, check that other equipment in the same signal loop is not affected. If necessary, appropriate measures must be taken.
- The unit is freely configurable. For this reason, the input and output behaviour is determined by the loaded engineering. Before commissioning, make sure that the right instructions for system and instrument commissioning are available.

Unless an application-specific engineering was loaded, the unit is equipped with the IO test engineering described on page 49.

⚠ **Before instrument switch-on, the plant-specific input and output signal types must be adjusted on the instrument, in order to avoid damage to the plant and to the device.**

On instruments without default setting, partial I/O signal testing is possible.

⚠ **The effect on connected equipment must be taken into account.**

After supply voltage switch-on, *start-up logo* and `Main menu wait!` are displayed, followed by display of the main menu during several seconds .

Unless a selection is made during this time, the first operating page (e.g. a controller) is displayed automatically, without marking a line or a field.

I-8 Operation

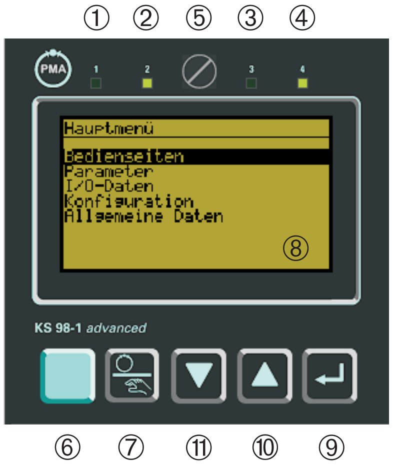
The operation of the device is menu-guided. The menu is divided into several levels which can all be influenced via the engineering, i.e. the final scope of the menu is dependent on the engineering.

This manual describes the operating functions which are independent of the engineering.

I-8.1 Front view

LEDs (①②③④):
for indication of conditions determined by the engineering, e.g. alarms or switching states.

Fig. 11



✋▼▲⮐ **keys (⑦⑪⑩⑨):**
Operation is by four keys for selection of pages and input on pages.

▼▲

The two functions of keys up/down are:

– Navigation in menus and on pages

– Changing input values (e.g.)

⮐

The two significations of the selector key are dependent on the selected field:

– Pressing the selector key (confirmation / Enter): starts page changing,

– starts value alteration via the up/down keys and confirms the adjustment subsequently (→ page 32).

✋

The functions of the auto/manual key are dependent on operating page, i.e. this key is sometimes called function key.

– Controller: auto/manual switchover

– Programmer: programmer control

– Adjustment of digital values.

⑤ **Locking screw:**
for locking the instrument module in the housing

⑥ **PC interface:**
PC interface for engineering tool ET/KS98 and BlueControl. The tools permit structuring/soft-wiring/configuration/parameter setting/operation.

⑧ **Display/operating page:**

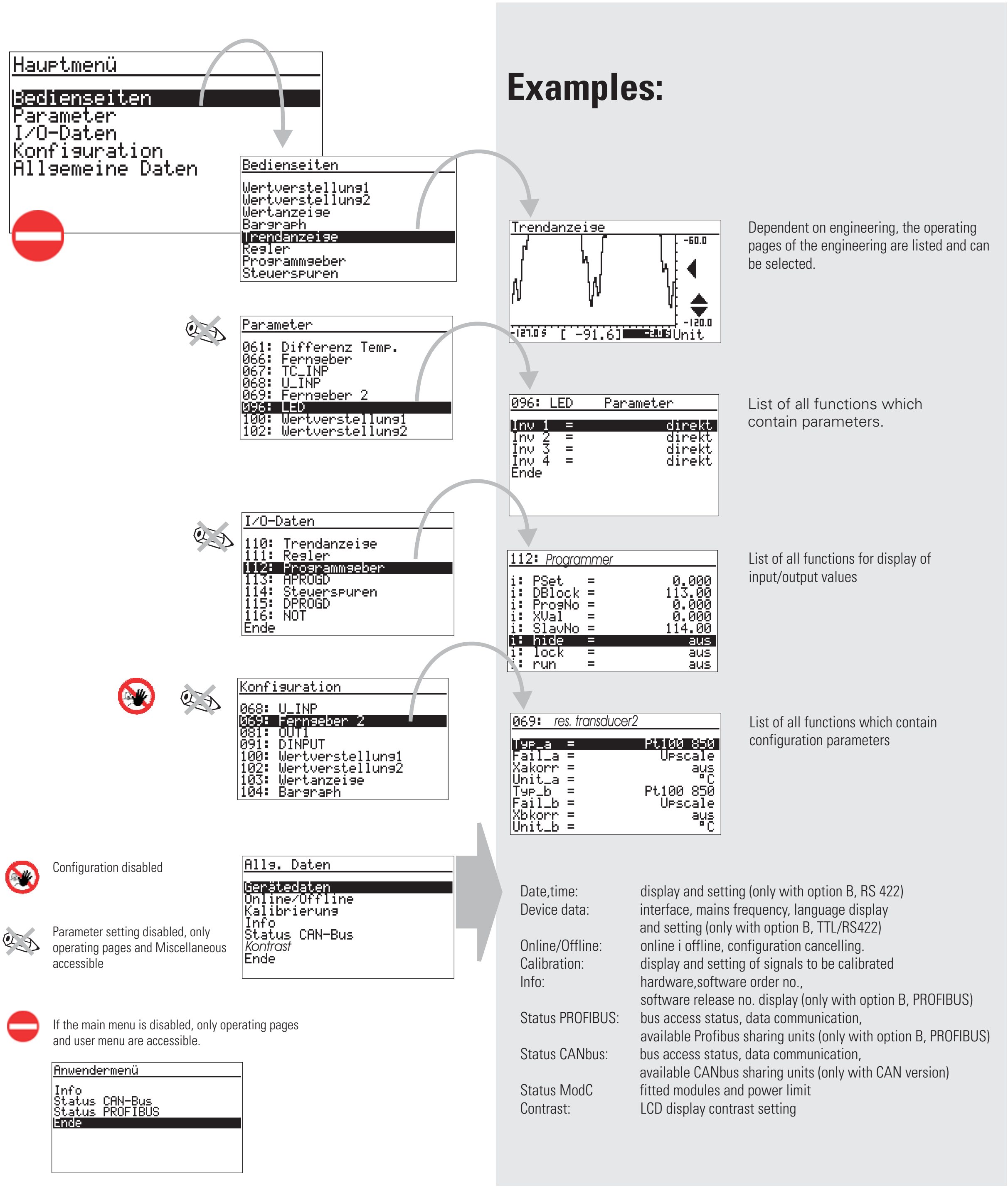
– LCD point matrix (160 x 80 dots),

– changeable green/red back lighting, direct/inverse display mode.

Display is dependent on configured functions.

I-8.2 Menu structure

The main menu is the uppermost menu level. Its structure is fixed and independent from the engineering.



```
Hauptmenü
Bedienseiten
Parameter
I/O-Daten
Konfiguration
Allgemeine Daten
```

```
Bedienseiten
Wertverstellung1
Wertverstellung2
Wertanzeige
Bargraph
Trendanzeige
Regler
Programmgeber
Steuerspuren
```

```
Parameter
061: Differenz Temp.
066: Ferngeber
067: TC_INP
068: U_INP
069: Ferngeber 2
096: LED
100: Wertverstellung1
102: Wertverstellung2
```

```
I/O-Daten
110: Trendanzeige
111: Regler
112: Programmgeber
113: APROGD
114: Steuerspuren
115: DPROGD
116: NOT
Ende
```

```
Konfiguration
068: U_INP
069: Ferngeber 2
081: OUT1
091: DINPUT
100: Wertverstellung1
102: Wertverstellung2
103: Wertanzeige
104: Bargraph
```

```
Allg. Daten
Gerätedaten
Online/Offline
Kalibrierung
Info
Status CAN-Bus
Kontrast
Ende
```

Configuration disabled

Parameter setting disabled, only operating pages and Miscellaneous accessible

If the main menu is disabled, only operating pages and user menu are accessible.

```
Anwendermenü
Info
Status CAN-Bus
Status PROFIBUS
Ende
```

Examples:

```
Trendanzeige
                              -60.0

                             -120.0
-127.0 s  [ -91.6]  -2.0 s Unit
```

Dependent on engineering, the operating pages of the engineering are listed and can be selected.

```
096: LED     Parameter
Inv 1   =        direkt
Inv 2   =        direkt
Inv 3   =        direkt
Inv 4   =        direkt
Ende
```

List of all functions which contain parameters.

```
112: Programmer
i: PSet    =      0.000
i: DBlock  =     113.00
i: ProgNo  =      0.000
i: XVal    =      0.000
i: SlavNo  =     114.00
i: hide    =        aus
i: lock    =        aus
i: run     =        aus
```

List of all functions for display of input/output values

```
069:  res. transducer2
Typ_a   =    Pt100 850
Fail_a  =      Upscale
Xakorr  =          aus
Unit_a  =           °C
Typ_b   =    Pt100 850
Fail_b  =      Upscale
Xbkorr  =          aus
Unit_b  =           °C
```

List of all functions which contain configuration parameters

Date,time:	display and setting (only with option B, RS 422)
Device data:	interface, mains frequency, language display and setting (only with option B, TTL/RS422)
Online/Offline:	online i offline, configuration cancelling.
Calibration:	display and setting of signals to be calibrated
Info:	hardware,software order no., software release no. display (only with option B, PROFIBUS)
Status PROFIBUS:	bus access status, data communication, available Profibus sharing units (only with option B, PROFIBUS)
Status CANbus:	bus access status, data communication, available CANbus sharing units (only with CAN version)
Status ModC	fitted modules and power limit
Contrast:	LCD display contrast setting

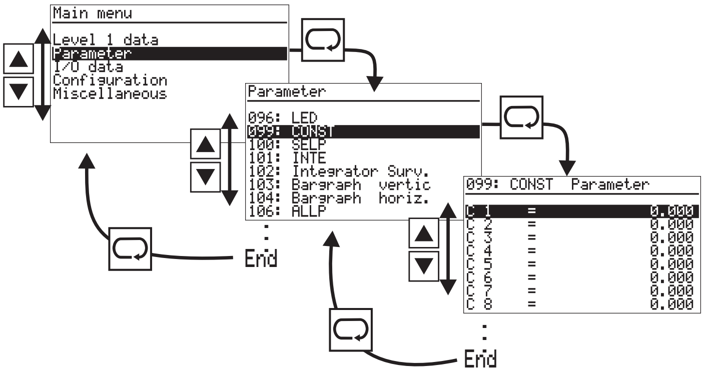
Fig. 12

I-8.3 Navigation, page selection

Operation of the device is by keys ↵ and ▲▼. After pressing key ↵ during 3 seconds, the main menu is always displayed.

☞ When the main menu is disabled the user menu is displayed.

Fig. 13 Example: Parameters



Procedure

① Press ▲▼ to select the input field or the line (the selected item is shown inversely).

② Confirm the entry with ↵ (for selecting).

③ **a)** If the selected item is a page, the page is opened and navigation can be continued with the ▲▼ keys.

③ **b)** If an input field was selected, the field starts blinking after pressing key ↵ and the required change can be entered with the ▲▼ keys. Confirm with key ↵. The input field stops blinking and the alteration is saved .

④ For exit from a page, scroll to menu item “End” at the bottom of the list. When selecting (↵), the next higher menu level is displayed.

☞ Scrolling up is possible.
When exceeding the uppermost menu item, menu item ”End” is displayed.
Unless display of a page is inverse despite actuation of keys ▲▼, the items were disabled (e.g. via engineering.
An inverse menu item which cannot be altered is also disabled.

Operating pages:

These pages offer an additional navigation function:

- Continuation or previous pages (marked by an arrow at the bottom (▼) or top (▲) of the page) can be activated by selecting and pressing key ↵.
- Items marked with ▶▶ open another operating page when selecting (▲▼) and confirming with key ↵.

☞ Menu item “End” is not provided on operating pages. Scroll until nothing is selected any more (no input field/line is shown inversely) and press key ↵ to go to the next higher menu level.

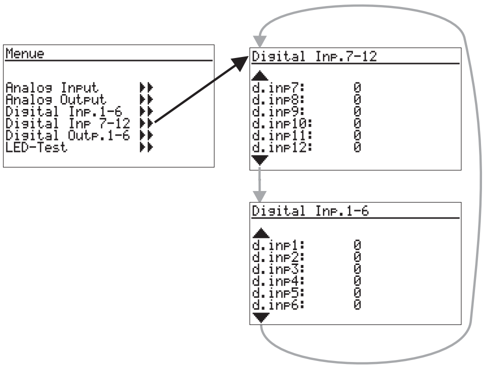


Fig. 14

I-8.4 Adjusting values

The menu operating pages include various types of fields for adjustment of values:

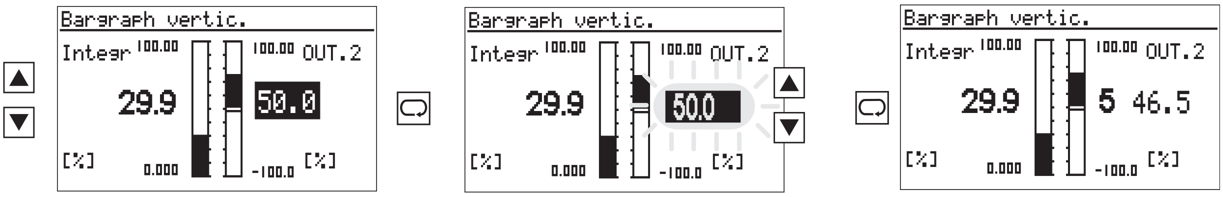
- analog values, digital values
- selection lists
- times
- on/off switches
- push-buttons
- selector switches (radio button)

Adjustment procedure

Select the value to be altered with keys ▲▼.

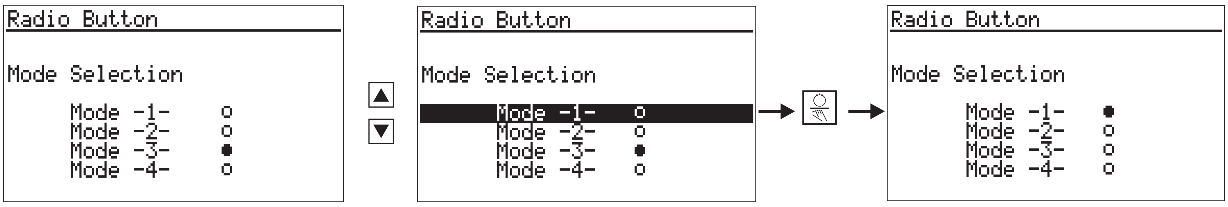
a) Press key ⏎ to start value changing (field blinks). Change the value with keys ▲▼.
Press ⏎ to store the change (field stops blinking).

Fig. 15 Example: Value adjustment bargraph



b) The longer keys up/down are pressed, the higher is the acceleration. When releasing, the adjustment speed decreases accordingly.

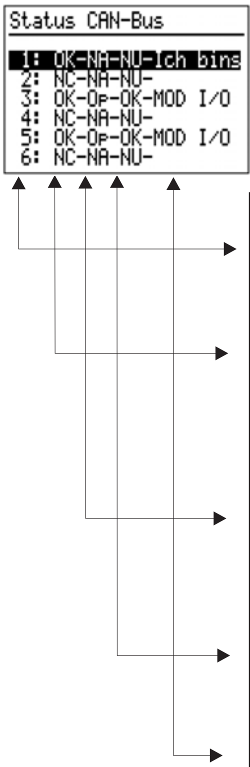
Fig. 16 Key ☝. This mode of adjustment is for switches, push-buttons and selector switches.



I-9 Instrument settings in the main menu

I-9.1 CAN-Status

The CAN bus status with the connected units is displayed.



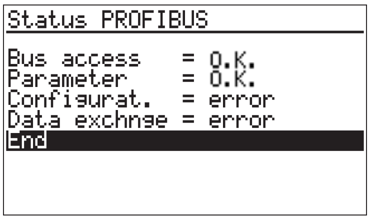
```
Status CAN-Bus

 1: OK-NA-NU-Ich bins
 2: NC-NA-NU-
 3: OK-Op-OK-MOD I/O
 4: NC-NA-NU-
 5: OK-Op-OK-MOD I/O
 6: NC-NA-NU-
```

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	Parametrierung: Lib function just setting the node parameters
OK	Ready: Lib function has finished the parameter setting.
String	detemined node name

I-9.2 Profibus-Status

The Profibus status page provides information on the Profibus connection status. The following error statuses are displayed:

- Bus access not successful
- Faulty parameter setting
- Faulty configuration
- No data communication

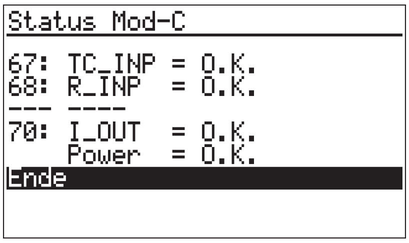


```
Status PROFIBUS

Bus access   = O.K.
Parameter    = O.K.
Configurat.  = error
Data exchnge = error
End
```

I-9.3 ModC-Status

The status page of the modular C card provides information on correct installation. Possible faulty installations are displayed:

- Difference of configured and fitted module type
- Power limits exceeded

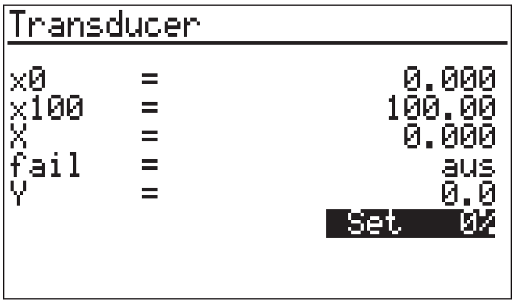


```
Status Mod-C

67: TC_INP = O.K.
68: R_INP  = O.K.
--- ----
70: I_OUT  = O.K.
    Power  = O.K.
Ende
```

I-9.4 Calibration

Press ▲▼ to select the input and ⏎ to open the calibration page.

Transducer input:
Adjusting the transducer start and end:

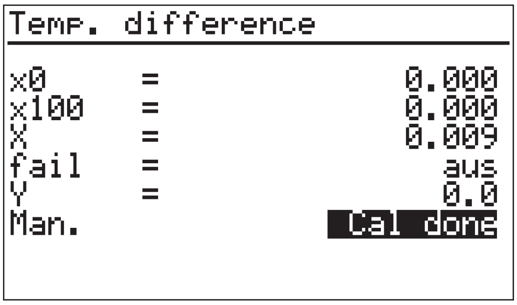


```
Transducer
x0       =        0.000
x100     =       100.00
X        =        0.000
fail     =          aus
Y        =          0.0
                 Set 0%
```

① Select `Quit` Set transducer to start
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 transducer to end
⑦ Press ⏎ → `0% done` blinks
⑧ Press ▲ 3x → `Set 100%` blinks
⑨ Wait until the input has settled
⑩ Press ⏎ → `100% done` is displayed.

Calibration is finished. For exit from calibration press ▼ until nothing is marked and then press ⏎.

Two resistance thermometers:
Calibration of lead resistance effect



```
Temp. difference
x0       =        0.000
x100     =        0.000
X        =        0.009
fail     =          aus
Y        =          0.0
Man.           Cal done
```

① Select `Quit` .
Short-circuit both thermometers 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 exit from calibration press ▼ until nothing is marked and then press ⏎.

I-9.5 Online/Offline

For configuration changing, switch the unit to ‘Offline’ and back to ‘Online’ after alteration.

⚠ **When switching the unit to status off-line, the outputs will remain in the status at switch-over time !!!**

☞ By switching over to on-line, all data is saved.

☞ When terminating the off-line mode by cancellation (Escape config.) the data saved last is loaded back into the working memory.

I-10 Operating pages

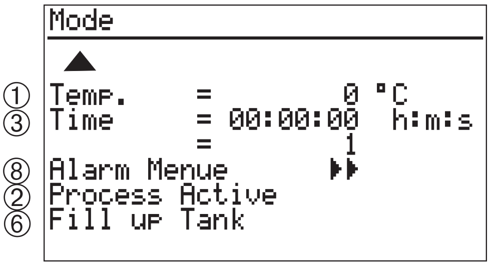
The engineering determines the scope of available operating pages. All available pages are listed in the operating page menu. The various types of pages are explained below.

I-10.1 List display

The operating page list is intended for display/input of process values and parameters.

Apart from digital, analog and time values, values of type radio button, switch and push-button can be defined in the value listing (→ page 33).

The value signification is determined by the engineering. The displayed values can be input fields.



```
Mode
 ▲
Temp.     =          0 °C
Time      = 00:00:00  h:m:s
          =          1
Alarm Menue        ▸▸
Process Active
Fill up Tank
```

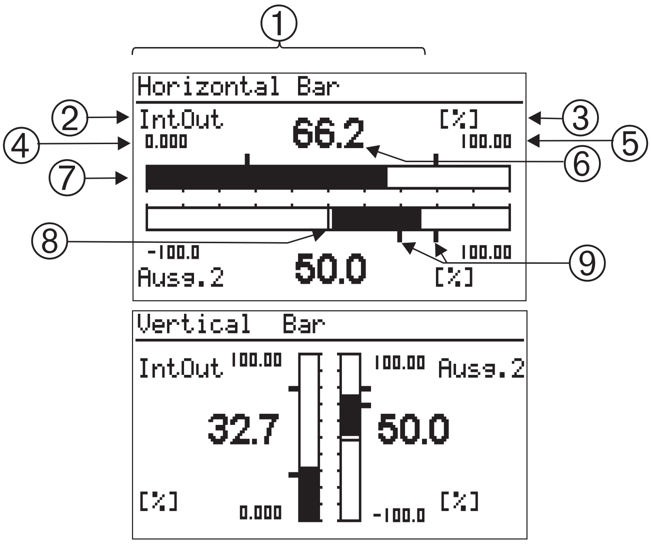
I-10.2 Bargraph display

The bargraph page is used for display of two analog variables as a bargraph.

Two further variables can be displayed and changed numerically and need not correspond with the bargraph values.

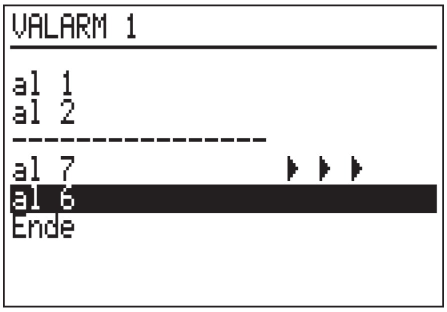
Four further analog inputs can be used to position two markers at the bargraph side, e.g. for indication of the alarm limit or reference values. When exceeding the limits, an arrow ▼ is displayed at the top and bottom end of the bargraph ▼ (see page 49).

Fig. 17



①	Title
②	Name for value
③	Unit for value
④	+⑤ Scale end values
⑥	Display/input field for value
⑦	Bargraph
⑧	Bargraph origin
⑨	Limit value markers for bargraph

I-10.3 Alarm display



```
VALARM 1

al 1
al 2
----------------
al 7            ▶▶▶
al 6
Ende
```

Alarm display is in the order of occurrence on a list.

One alarm per line is displayed:

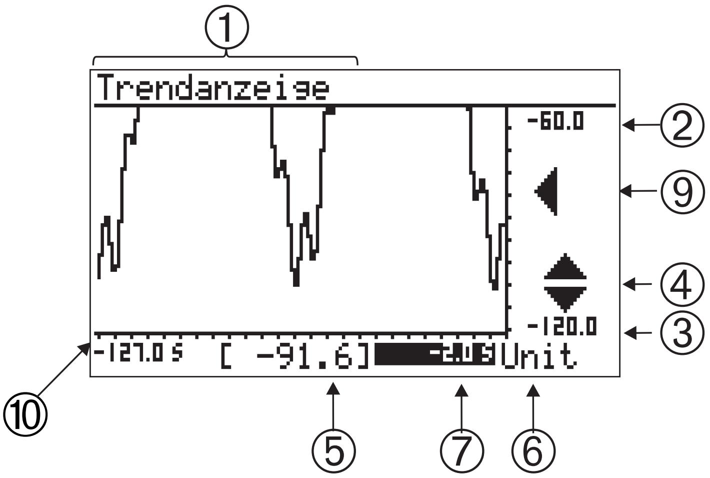
Alarm active	alarm text blinks
Alarm active and ackn.	Alarm text
Alarm not active any more and not acknowledged	Alarm text ▶▶▶
Alarm not active any more	--------------------

Acknowledging an alarm
Select an active alarm for acknowledging with ▲▼ and acknowledge it with ⏎. New alarms are displayed only when rebuilding the page, which is done by pressing key ⚇.

I-10.4 Graphic trend curve

The time curve of a process is displayed graphically on the trend page.

Fig. 18



① Title
② ☞③ Scale end values
④ Zoom switchover
⑤ Value at time ⑦ / actual input value
⑥ Unit of value
⑦ Origin of time axis related to the actual value
(=0) shift of time axis (scrolling into the past)
⑨ Axis shift signalling
⑩ End of time axis / earliest value in the displayed trend

Zoom value scale
The value axis can be zoomed by factor 1:4 (cut-out magnification).

Select the "zoom" field ④, press ⏎, the zoom symbol changes. Now, scaling can be changed by means of keys ▲▼. The scaling is shifted in steps of 12,5% via field ③.

Shift of time axis:
Earlier values than those visible in the actual window are also displayed by the trend function (Shift). Values left of the time axis are earlier values. These values can be displayed by changing the origin of the time axis.

Select field ⑦ with ▲▼ and shift the scale origin by changing the value.

☞ Symbol ◀ (⑨) indicates the shift.

When resetting the time scale to 0, the shift is switched off.

I-10.5 Programmer

- A programmer controls the process sequence of a plant.
- Programmers are configurable freely in structure and scope by means of the engineering.
- A programmer is composed of any number of s (analog values) and control values (digital control bits).
- Any number of programs (recipes) can be stored for a programmer.
- The program is divided into a defined number of segments (program segments).
- The maximum number of segments is determined in the engineering.
- The maximum scope is defined in the engineering.

The actual status of a running program is displayed on the programmer operating page. Dependent on programming, status (run/stop, auto/manual), segment number, net time and the actual (during manual operation) can be changed.

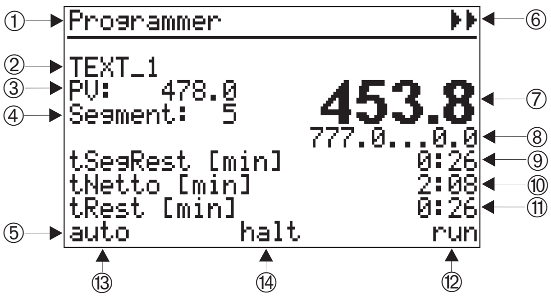
Programmer operation is divided into:

- Program control and monitoring
- Program (recipe) selection
- Adjustment of setpoints/control bits during manual operation
- Parameter setting for program

☞ Dependent on engineering, parts of this operation can be changed or disabled.

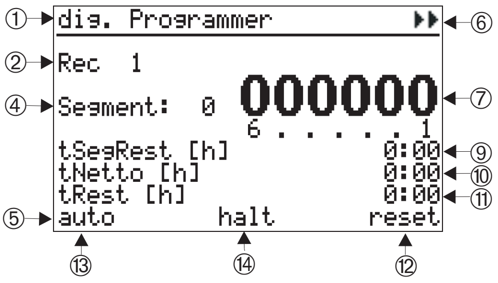
Display of the operating page is always related to one programmer output, whereby analog s and digital control bits are distinguished. Change to the next programmer output is via field ⑥ ▶▶ in the title.

Fig. 20



```
Programmer                 ▶▶
TEXT_1
PV:    478.0         453.8
Segment:   5
                 777.0...0.0
tSegRest [min]          0:26
tNetto [min]            2:08
tRest [min]             0:26
auto         halt        run
```

Fig. 19



```
dig. Programmer          ▶▶
Rec  1
Segment:  0       000000
                  6 . . . . . 1
tSegRest [h]            0:00
tNetto [h]              0:00
tRest [h]               0:00
auto        halt       reset
```

① Name of operating page
② Program name/no. (recipe)
③ [Process value]
④ Actual segment no.
⑤ Status line
⑥ Programmer output switchover
⑦ /control value
⑧ Setpoint from...to in the current segment
⑨ Remaining segment time
⑩ Elapsed program time
⑪ Remaining program time
⑫ Program status (stop,run,reset,search,program,quit,error)
⑬ auto/manual
⑭ halt, end

Selecting a program

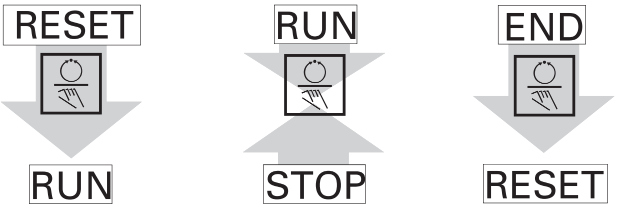
Selection of a program is by alteration of recipe field② . Dependent on engineering, selection is from a text list or by entry of a number.

☞ Program selection is possible only in status "reset".

Controlling a program

Press key to control the program sequence:

The time curve can be controlled also by changing the elapsed time ⑩ or segment number ④ (preset).



☞ Dependent on engineering, parts of this operation may be changed or disabled.

Program parameter setting

Select the program for editing via field "Rec" ②. Call up the relevant s/control values, segment times and types with menu item “program” in the status line (field ⑫).

A page on which the selected program is displayed as “RecEdt” is opened. The parameters are listed in the order of segments.

The data blocks are displayed dependent on engineering. The type of individual segments can be changed dependent on data block type.

Selection of all programs including the inactive ones is possible on line RecEdt in any programmer status.

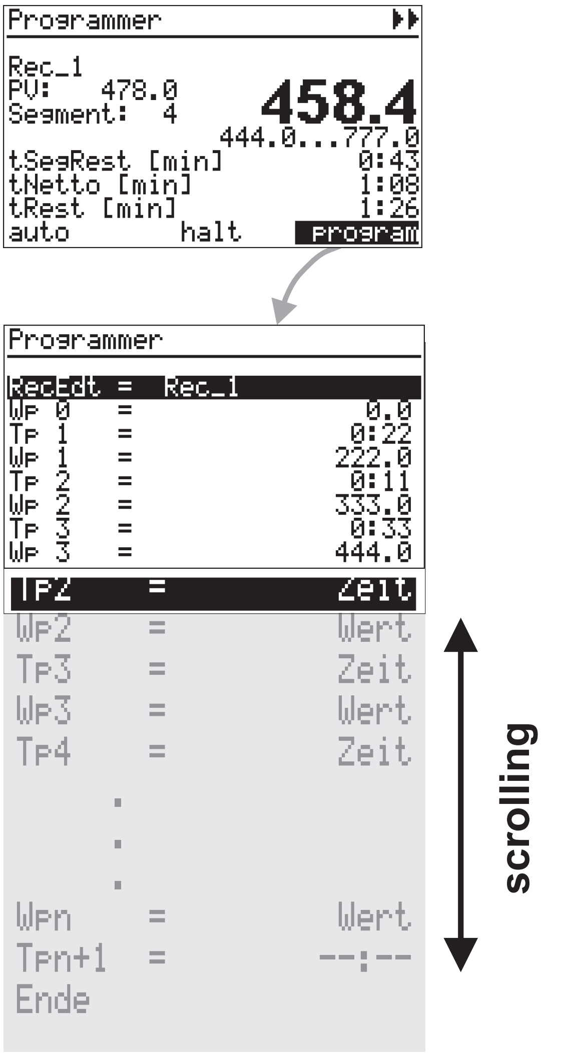
When using recipe names, these names are displayed on the editing page. Switching over to the parameters of a different recipe can be done by altering the recipe name. This is possible at any time and does not cause switchover of the active recipe.

A segment list is completed with end identification --:-- in parameter Tp_n of the last segment. When setting the last segment time T_n to a valid value (higher or equal to 0), the next parameter is displayed automatically T_{n+1}. = --:-- etc.

This procedure permits shortening of a current program by adjusting a value < 0 in the required position for T_n = --:-- with key ▼.

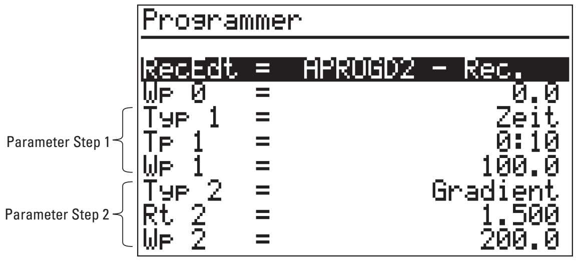
The following segments will be suppressed in the program. The relevant segment parameters remain unchanged and can be re-activated by input of a valid value for $T_{n.}$

Fig. 21



```
Programmer                ▸▸

Rec_1
PV:    478.0
Segment:   4        458.4
             444.0...777.0
tSegRest [min]        0:43
tNetto [min]          1:08
tRest [min]           1:26
auto        halt   program
```

```
Programmer

RecEdt =   Rec_1
Wp 0    =          0.0
Tp 1    =         0:22
Wp 1    =        222.0
Tp 2    =         0:11
Wp 2    =        333.0
Tp 3    =         0:33
Wp 3    =        444.0
Tp2     =         Zeit
Wp2     =         Wert
Tp3     =         Zeit
Wp3     =         Wert
Tp4     =         Zeit
      .
      .
      .
Wpn     =         Wert
Tpn+1   =        --:--
Ende
```



```
Programmer

RecEdt =   APROGD2 - Rec.
Wp 0    =          0.0
Typ 1   =         Zeit
Tp 1    =         0:10
Wp 1    =        100.0
Typ 2   =     Gradient
Rt 2    =        1.500
Wp 2    =        200.0
```

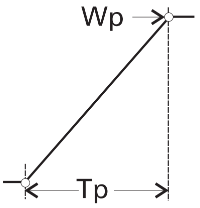
Segment types

Dependent on segment type, the following parameters can be altered:

Wp i	Target setpoint
D i	Control value in segment i
Tp i	Segment duration
Rt i	Segment gradient
Typ i	Segment type

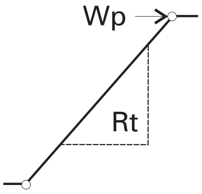
Ramp segment (time)

With a ramp segment (time), the runs linearly from the start value (end of previous segment) towards the target (Wp) of the relevant segment during time Tp (segment duration).



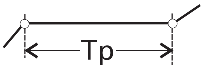
Ramp segment (gradient)

With a ramp segment (gradient), the runs linearly from the start value (end value of previous segment) towards the target value (Wp) of the relevant segment. The gradient is determined by parameter Rt.



Hold segment

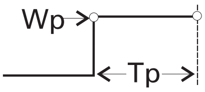
With a hold segment, the end of the previous segment is output constantly during a defined time which is determined by parameter Tp.



Step segment

With a step segment, the program goes directly to the value specified in parameter Wp.

The reached due to the step change is kept constant during the time determined in parameter Tp.



Waiting and operator call

All segment types can be combined with "Wait at the end and operator call".

If a segment with combination "wait" was configured, the programmer goes to stop mode at the segment end. Now, the programmer can be restarted by pressing the □-key.

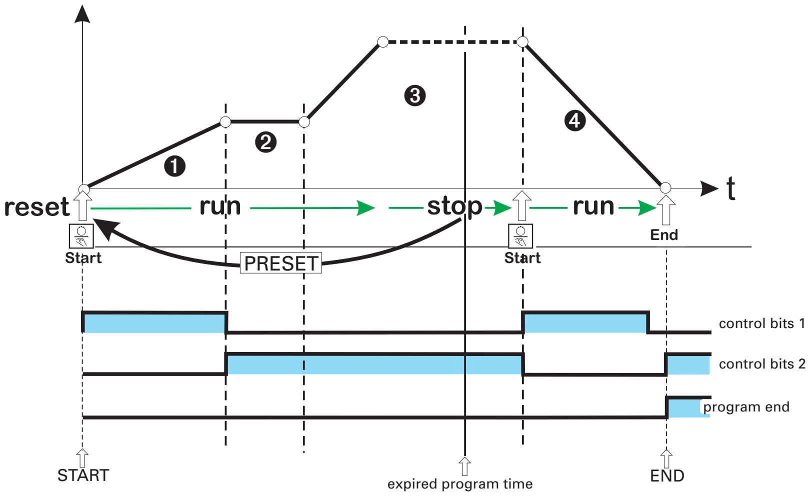


Fig. 22
❶ Segment type = Time
❷ Segment type = Hold
❸ Segment type = Time and wait
❹ Segment type = Gradient

Manual mode

The programmer output can be overwritten for each page.
For this, the relevant page must be switched off to "manual"⑬. In this mode, the or control value can be overwritten ⑦. The control value is changed separately for each control bit. Press ↵ to continue.

Field ⑬ permits returning to the automatic mode(→ page 38).

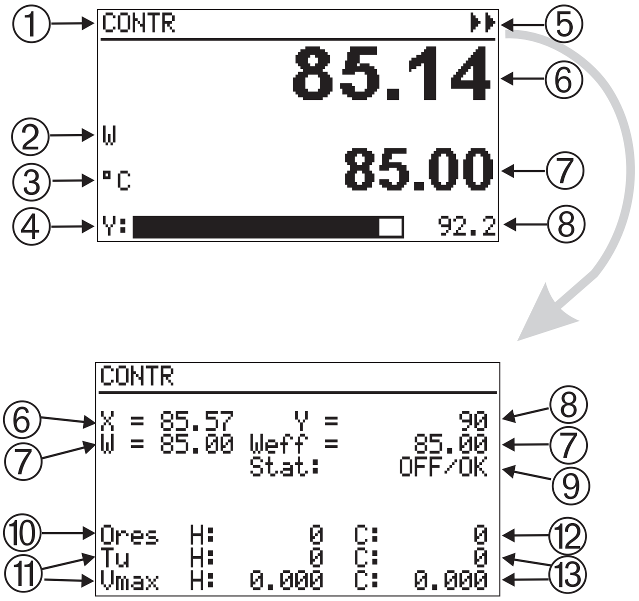
☞ The program run is not interrupted by the manual mode.

I-10.6 Controller

The controller page permits intervention into process control loops. Input fields (, source, correcting variable during manual mode, parameter set switchover) are selected via the ▲▼ key, pure display fields are skipped.

☞ Dependent on engineering, the input fields can be disabled.

Fig.23: Controller operation



```
CONTR                  ▸▸
                   85.14
W
°C                 85.00
Y:█████████████▯    92.2
```

```
CONTR
X = 85.57    Y =        90
W = 85.00 Weff =     85.00
          Stat:     OFF/OK

Ores  H:      0  C:      0
Tu    H:      0  C:      0
Vmax  H:  0.000  C:  0.000
```

① Page title
② source (Wint, Wext, W2)
③ Physical unit
④ Bargraph of correcting variable Y or XW or Xeff
⑤ Entrance into the self-tuning page
⑥ Effective process value
⑦ Controller
⑧ Value of correcting variable Y or XW or Xeff
⑨ Self-tuning/command input status
⑩ Self-tuning result heating
⑪ Process characteristics heating
⑫ Self-tuning result cooling
⑬ Process characteristics cooling

Apart from entries and switchover operations, further actions can be started: Switching over to manual operation is done via key ✋ and field - ⑤ provides access to the controller self-tuning page.

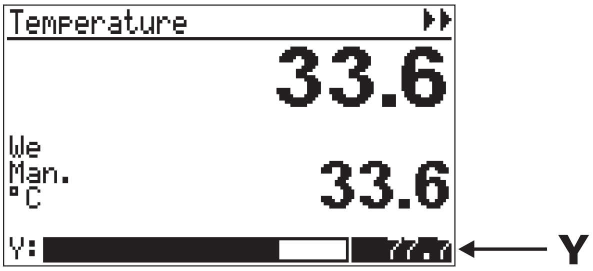
Input fields on the operating page

Manual value adjustment

This field can be used to adjust the correcting variable during manual operation. Adjustment is enabled only during manual operation. Unless manual operation is active, the field cannot be selected.

When changing over to manual operation, the bargraph display is always switched over to Y display (correcting variable), also with X1 or XW defined in the configuration. The actual correcting variable is displayed right beside the bargraph.

Fig. 24:Correcting variable adjustment via the front panel



```
Temperature            ▸▸
                    33.6
We
Man.
°C                  33.6
Y:███████████▯     77.7
```

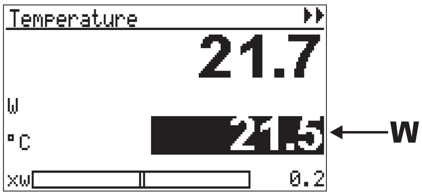
Manual correcting variable

Alteration of manual correcting variable ⑧ by means of keys ▲▼ is at three speeds.

Press the key to start the adjustment at a speed of 1% / sec. After 3 sec., switchover to 2.5% / sec occurs and switchover to 10%/sec. is after another 3 sec.

The internal can be altered at any time,
also when another is active.

Fig.25 : adjustment via the front panel



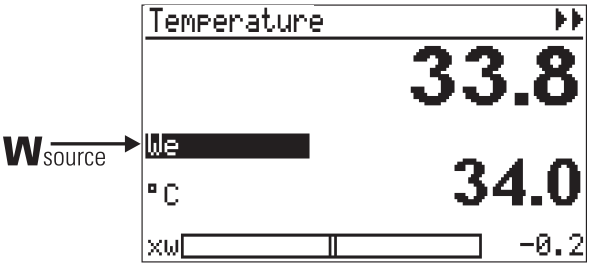
```
Temperature          ▸▸
                   21.7
W
°C                 21.5
xw                  0.2
```

source

 source switchover is possible via a selection field on the controller page.

Dependent on controller configuration, selection of Wint, Wext and W2 is possible.
The field can be left with Quit, if no switchover is needed,

Fig. 26 Front-panel switchover



```
Temperature          ▸▸
                   33.8
We
°C                 34.0
xw                 -0.2
```

Self tuning

Determination of the optimum process parameters is possible by self-tuning. Self-tuning is available for processes with compensation and no delay time.

Dependent on controller type, parameters Xp1, Xp2, T_n, Tv, Tp1, Tp2 are determined.

Preparation

- Adjust the desired controller behaviour.
- e = 0.0.
-

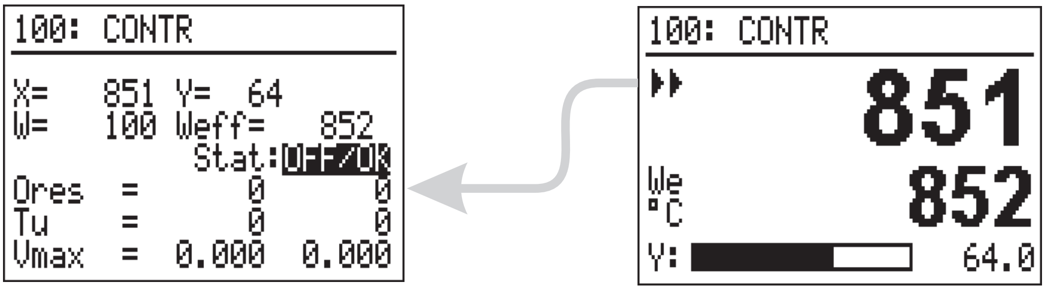
P- controller:	Tn = 0.0	Tv = 0.0
PD- controller:	Tn = 0.0	Tv > 0.0
PI- controller:	Tn > 0.0	Tv = 0.0
PID- controller:	Tn > 0.0	Tv > 0.0

- If a controller has several parameter sets, selection of the parameter sets to be optimized is required ((POpt =1...6). If necessary, these settings must be made available when creating the engineering).
- Switch the controller to manual mode (key ✋). Alter the correcting variable to reach the working point.

The process must be in a stable condition. Self-tuning starts only, when process value oscillations are smaller than 0.5% of the control range during one minute (controller display:, process at rest' (PiR)).

☞ If necessary, other control loops in the plant must be set to manual mode as well.

Fig. 27 Calling up the self-tuning page



```
100: CONTR
X=    851 Y=   64
W=    100 Weff=    852
          Stat:OFF/OK
Ores   =        0     0
Tu     =        0     0
Vmax   =   0.000  0.000
```

```
100: CONTR
▸▸
                 851
We
°C               852
Y:              64.0
```

reserve:

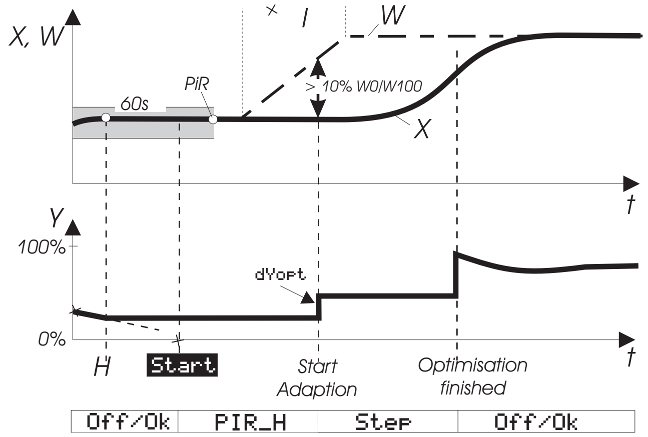
To permit self-tuning, the distance between setpoint and process value must be higher than 10% of the setpoint range before self-tuning start.

With inverse controllers, the must be higher than the process value. With direct controllers, it must be smaller.
The determines a limit which is not exceeded during self-tuning.

Self-tuning start

Select function Stat: OFF/OK and confirm it with ⊡. Stat:OFF/OK blinks and is switched over to Stat: Start by pressing ▲.

Fig.: 26 Self-tuning with heating and cooling



Pressing key ⊡ starts the self-tuning attempt. The can be changed also subsequently. After successful self-tuning, the controller changes to automatic operation and controls the with the new parameters.

When 'Process at rest' (PiR) is detected and a sufficient reserve is provided, the correcting variable is changed by output step (increased with indirect controller, decreased with direct controller).

! The size of the output step change is set to 100% as standard. In critical processes, this value (parameter dYopt) may have to be reduced to prevent damage to the process. The parameter is adjustable in the engineering, or via the para - meter dialogue of the main menu, if the engineering is known . In case of doubt, contact the programming engineer.

When self-tuning is finished with an error (Ada_Err or 0err on the controller page), the initial correcting variable is output until self-tuning is finished by pressing key ⊡.

Self-tuning procedure with heating and cooling processes:

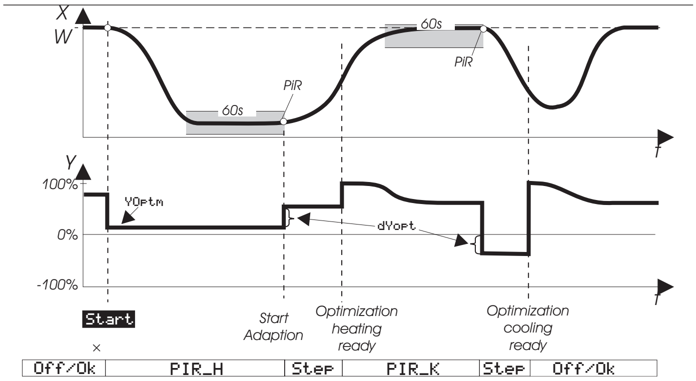
(3-point / split-range controller)

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 , until PiR is reached again.

Subsequently, a step to cooling is made to determine the "cooling" parameters. When cancelling the cooling attempt, the parameters for "heating" are also taken over for cooling. No error message (Ada_Err) is output.

Fig.29:



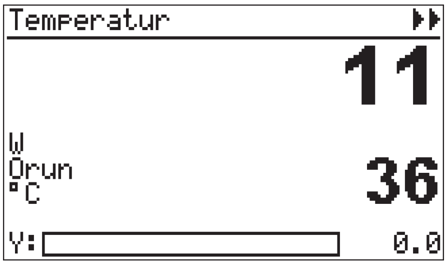
⚠ **Whilst self-tuning is active, the control function is switched off!**

The self-tuning statuses are indicated with priority in the display field for manual operation.

- Self-tuning running, display: ORun
- Self-tuning faulty, display: OErr

Self-tuning completed with an error is finished by pressing key ⚙ twice.

Fig.30: Controller page with started self tuning

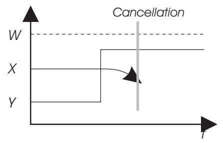
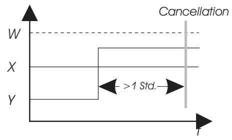
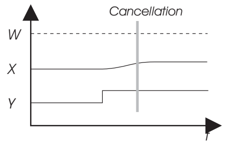
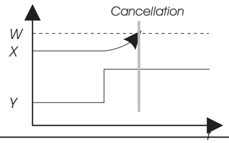
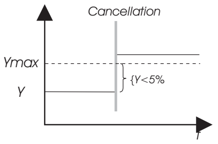
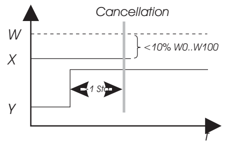


Temperatur ▸▸
11
W
Orun
°C
36
Y: 0.0

Self-tuning cancelation

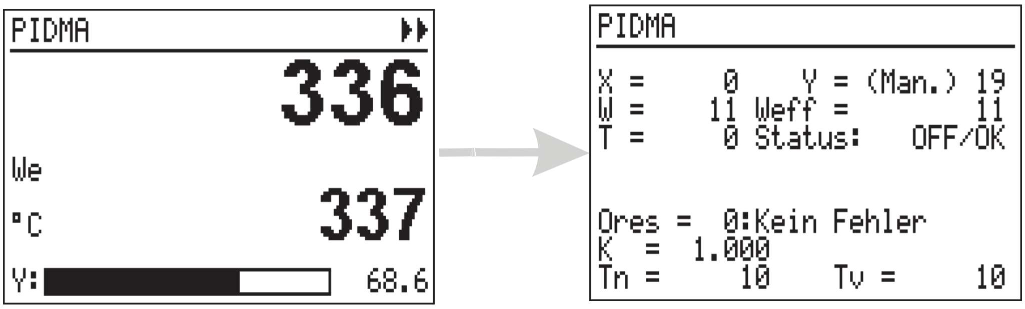
Self-tuning can be stopped at any time by pressing key ⚙, or by selecting Stop in the Stat field (status).

Signification of self-tuning messages ORes1/ORes2 for controller type CONTR and CONTR+

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 hazard. 	Increase separation of process value (X) and (W) when starting, or decrease YOptm .
6	Completed: attempt successful, but self-tuning cancelled due to exceeded 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: reserve too small, or exceeded whilst PiR monitoring is busy. 	Vary stable correcting variable YOptm.

The controller type PIDMA offers the following self-tuning page.

Fig. 31 Optimization page



```
PIDMA                 ▸▸
                     336
We
°C                   337
Y:                  68.6
```

```
PIDMA
X =      0    Y = (Man.) 19
W =     11 Weff =        11
T =      0 Status:   OFF/OK

Ores =  0:Kein Fehler
K  =  1.000
Tn =      10    Tv =     10
```

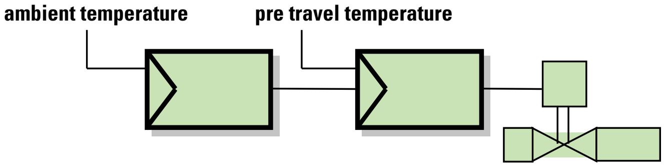
For self-tuning preparation, parameters must be adjusted dependent on process and engineering. For this purpose, special knowledge of the applicable function block is required, i.e. it should be done by the programming engineer. Self-tuning start is as described above.

Signification of self-tuning messages ORes for PIDMA

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 large	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	Start again	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 peak	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	Output sat	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	Extrapolation	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	Bad 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. break	The self-tuning attempt was canceled manually by the operator with „*STOP*" .
12	Direction	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.

I-10.7 Cascade controller

With cascade control, two coupled controllers act on a common actuator. A process value for the master and a process value for the slave controller are required.



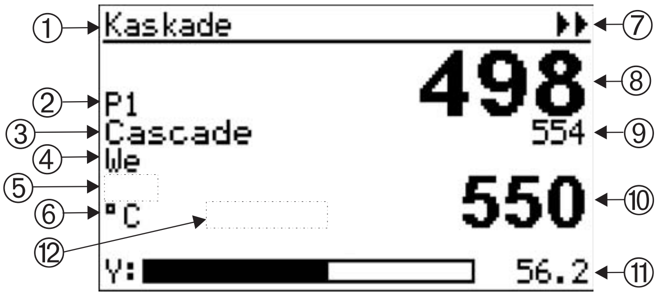
The slave is determined via the external by the master. Cascade operation is possible in the following statuses:

Automatic mode
In a cascade, master and slave operate automatically during automatic mode. The master and process value are the relevant variables for process control. The master is adjustable. The process value of the slave ⑨ is displayed additionally.

"**Cascade**" is displayed.

① Operating page title
② Parameter set selection, if available
③ Switchover field cascade mode (open/closed)
④ source of master (W_{int}, W_{ext}, W_2)
⑤ Display field for manual mode (otherwise empty)
⑥ Physical unit (master block parameter)
⑦ Entry into self-tuning
⑧ Master process value
⑨ Slave process value
⑩ (from master in automatic mode, from slave with open cascade)
⑪ Bargraph and display
(Y from slave or X/XW from master)
⑫ Display of slave selection with open cascade (otherwise empty)

Fig. 32 Operating page of a cascade controller in automatic


Kaskade
498
P1
Cascade
554
We
550
°C
Y: 56.2

Cascade opened

For opening the cascade and control by means of the slave controller (see note text “Slave” on the operating page), switchover field ③ is switched to “Casc- Open”.

☞ "**Casc-open**" is displayed.

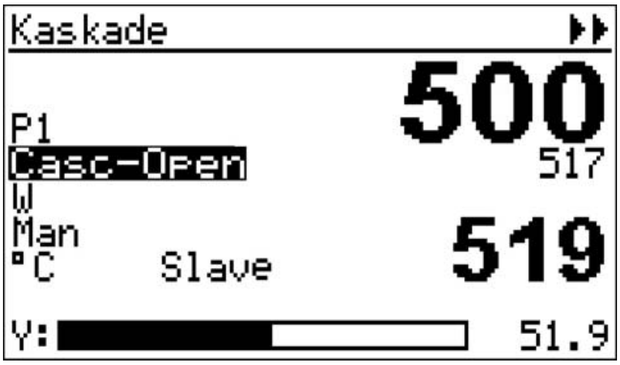
The slave is displayed now.

Now, the slave controller becomes the variable used for process control and can be adjusted.

The process value of the master control loop is set by the cascade loop rather than being controlled. Switchover between operation by master or slave is always possible.

In cascade mode, the master information is displayed in the fields for , source, physical unit and X/XW bargraph. With open cascade (display "Slave"), the slave information is displayed.

Fig. 33 Cascade controller with open cascade



```
Kaskade                ▶▶
                      500
P1
Casc-Open             517
W
Man                   519
°C     Slave
Y:                   51.9
```

Manual mode

Switchover to manual is via key ✋ (display in field⑤). The cascade status (open/closed) is not affected. In manual mode the process is controlled directly with the slave correcting variable. The slave correcting variable can be adjusted during manual operation.

☞ "**Man**" is displayed.

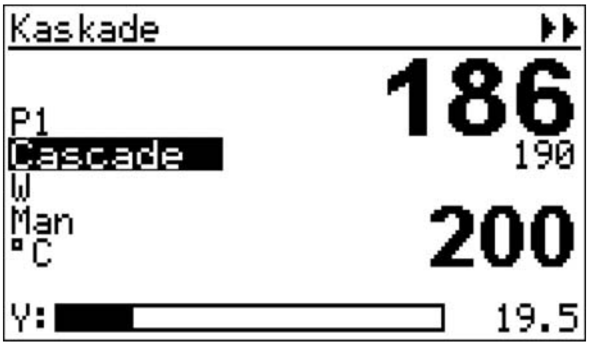
Cascade optimization

In a cascade, the slave controller and then the master must be optimized.

The self-tuning entry of the cascade operating page ▶▶ relates always to the slave!

☞ For optimizing the master controller, the master must be selected purposefully via the operating menu. For this, the project description must be used.

Fig. 34 Cascade controller in manual mode



```
Kaskade                ▶▶
                      186
P1
Cascade               190
W
Man                   200
°C
Y:                   19.5
```

I-11 Maintenance, test, trouble shooting

I-11.1 Cleaning

Housing and front panel can be cleaned using a dry, lint-free cloth. No use of solvents or cleansing agents!

☞ Avoid using solvents or cleansing agents!

I-11.2 Behaviour in case of trouble

The unit needs no maintenance. In case of trouble, check:

- Is the unit in on-line mode ?
- Is the supply voltage connected correctly? Are voltage and frequency within the tolerances?
- Were all connections made correctly ?
- Do the sensors and actuators work properly?
- Is the engineering OK?
- Is the unit configured for the required operating principle?
- Do the adjusted parameters have the required effect?
- Are the I/O extension modules plugged in and clicked in position correctly (modular option C)?
- Is a terminating resistor activated (can be required dependent on the instrument position in the bus topology with CANopen and PROFIBUS DP)?
- Were the required EMC measures carried out (screened cables, earthings, protective circuits, etc.)?
- Does the diagnostic page of the test engineering indicate an error?

If the unit does not function correctly after these checks, it must be shut down and replaced. A defective unit can be returned to the supplier for repair.

I-11.3 Shut-down

⚠ **Disconnect the supply voltage completely and protect the unit against accidental operation. As the instrument is mostly connected with other facilities in the control loop, consider the effects before switching off and take measures to prevent the occurrence of undesired operating conditions!**

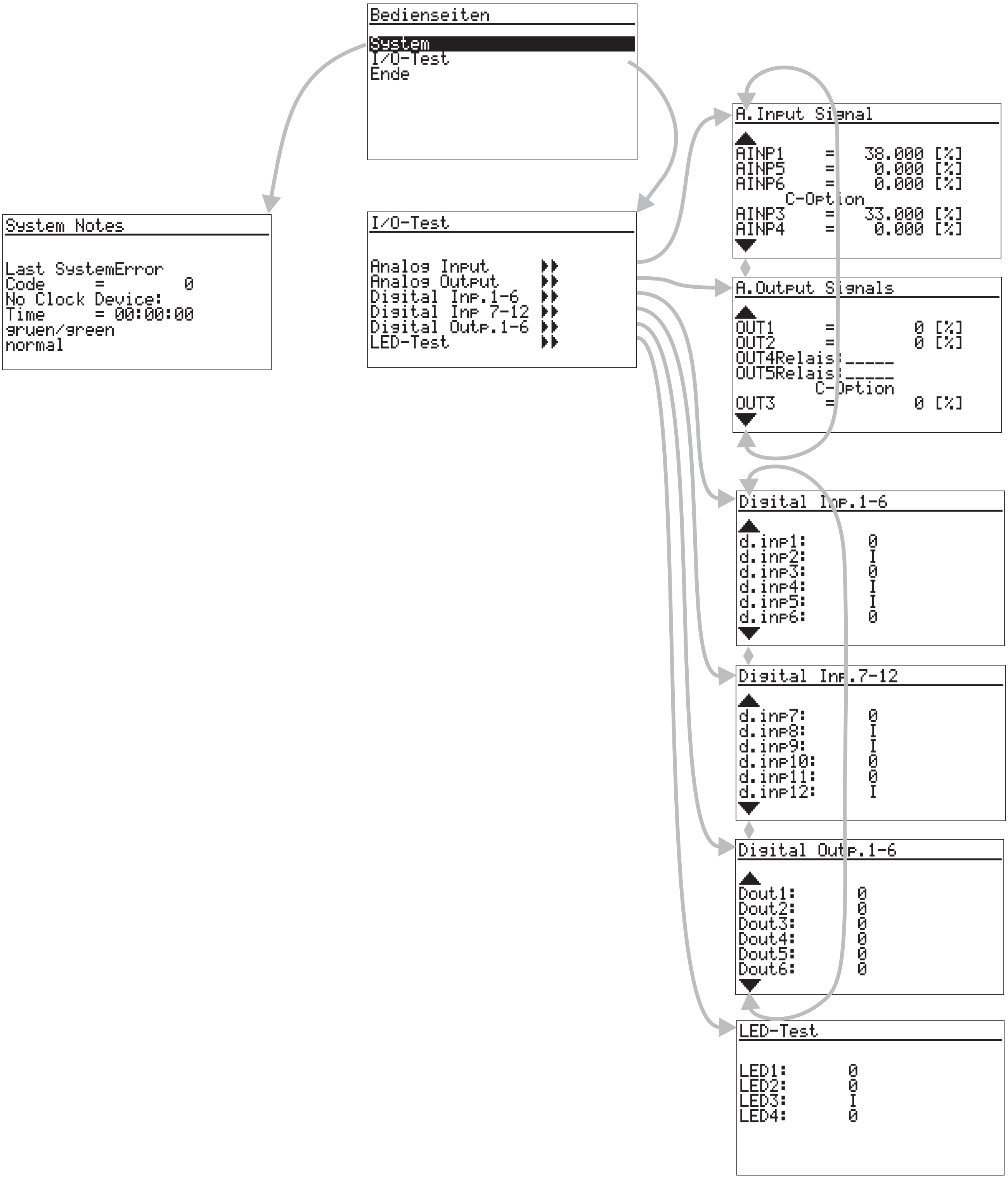
I-11.4 Test engineering as basic equipment

KS98-1 is factory set for a test engineering ***IO-test.edg,*** which ensures checking of the possible inputs and outputs of the extended basic unit (standard + option B + option C). If KS98-1 is provided with a customer-specific engineering, the relevant description is applicable.

! If KS98-1 is provided with customer-specific engineering, the engineering description is applicable.

A diagnostic page is provided to indicate *system errors* in case of start-up problems. The availability of the real-time clock is also displayed. The display colour can be changed green/red and normal/inverse.

Fig. 35 Menustructure of the test-engineering



```
Bedienseiten

System
I/O-Test
Ende
```

```
System Notes

Last SystemError
Code      =        0
No Clock Device:
Time      = 00:00:00
gruen/green
normal
```

```
I/O-Test

Analog Input      ▸▸
Analog Output     ▸▸
Digital Inp.1-6   ▸▸
Digital Inp 7-12  ▸▸
Digital Outp.1-6  ▸▸
LED-Test          ▸▸
```

```
A.Input Signal
▲
AINP1    =   38.000 [%]
AINP5    =    0.000 [%]
AINP6    =    0.000 [%]
      C-Option
AINP3    =   33.000 [%]
AINP4    =    0.000 [%]
▼
```

```
A.Output Signals
▲
OUT1     =         0 [%]
OUT2     =         0 [%]
OUT4Relais:_____
OUT5Relais:_____
         C-Option
OUT3     =         0 [%]
▼
```

```
Digital Inp.1-6
▲
d.inp1:       0
d.inp2:       I
d.inp3:       0
d.inp4:       I
d.inp5:       I
d.inp6:       0
▼
```

```
Digital Inp.7-12
▲
d.inp7:       0
d.inp8:       I
d.inp9:       I
d.inp10:      0
d.inp11:      0
d.inp12:      I
▼
```

```
Digital Outp.1-6
▲
Dout1:       0
Dout2:       0
Dout3:       0
Dout4:       0
Dout5:       0
Dout6:       0
▼
```

```
LED-Test

LED1:      0
LED2:      0
LED3:      I
LED4:      0
```

I-11.5 I/O-Test

Input and output type and measuring/signal range are configurable.

For this, switch the unit to OFFLINE(→ page 35) first after starting up. All inputs and outputs are preset to 0 .. 20mA and 0-100% value range.

Before commissioning, the inputs and outputs to be connected must be configured for the required sensor type via main menu “Configuration”.

After adjusting the correct type, the unit must be switched back to ONLINE!

Now, KS98-1 is ready for the first input/output test.

Possible settings:

- AINP1: thermocouple types; Pt100; 2*Pt100; 0/4 .. 20mA; 0/2 .. 10V; transducer 500 Ω; resistance 500Ω 250Ω
- AINP3 (option C): 0/4 .. 20mA
- AINP4 (option C): 0/4 .. 20mA
- AINP5 : 0/4 .. 20mA
- AINP6 : 0/4 .. 20mA
- OUT1 : 0/4 .. 20mA or relay
- OUT2 : 0/4 .. 20mA or relay
- OUT3 (option C) : 0/4 .. 20mA
- OUT4 : relay
- OUT5 : relay

Dependent on instrument selection, outputs OUT1 and OUT2 can be relay or current output. Accordingly, they must be controlled digitally or analoguely in the engineering.

As all outputs were defined as analog outputs in the present engineering, a value below 50% (corresponds to logic “0”) and a value above 50% (corresponds to logic “1”) must be set for relay output testing.

Outputs OUT4 and OUT5 are always relays, i.e. they are controlled digitally on the relevant operating page. (Adjusting values → Navigation page 32).

Adjusting the output values and selecting the pages are done as described. Continuation pages are selected using keys ▲▼ via the menu lines (▲, ▼) and called up with ⏎.

This engineering is not suitable for controlling a plant. For this purpose, a customer-specific engineering is required (see versions, section: Adjustment on page 18).

Faulty settings can cause damage to instrument and plant!

II Engineering-Tool

II-1 Survey

The engineering tool for KS 98-1 enables the user to make an engineering which is specially tailored for his application. The engineering tool mainly comprises a function block editor supported by the IEC 1131-3 standards.

The engineering tool offers the following functions:

- By selection from a menu, functions can be selected and placed in the working window.
- Outputs and inputs can be connected graphically.
- When shifting functions, the connections are dragged.
- Function configuration and parameter setting.
- Downloading of the engineering into KS 98-1.
- Adjustment management.
- Long-term storage of various engineerings on hard disk or floppy.

Coupling the PC with the KS 98-1 multifunction unit is done via an RS232/TTL adaptor cable, which can be ordered separately (order number : 9407 998 00001).

II-1.1 Scope of delivery

The engineering tool scope of delivery comprises the following components:

- One CD-ROM for English, French, and German version.
- the operating instructions
- licence conditions
- registration with licence number

II-2 Installation

II-2.1 Hardware and software prerequisites

For using the engineering tool, the following system prerequistes must be met:

- min. 486 IBM-compatible PC,
- at least 8 MB working memory
- VGA card and suitable monitor (min. screen resolution 800 * 600 pixels)
- hard disk with min. 2,5 MB free memory
- Diskette drive / CD drive (CD-ROM can be copied onto diskettes)
- MS-Windows from version 3.1
  (tested: Windows 3.1, Windows for Workgroups 3.11 and Windows 95)
- a free serial interface (COM1 - COM4)
- mouse required, adjustment as a double-key mouse in standard mode for right-handed operators

II-2.2 Software installation

Installation from CD

Place the CD into the CD drive and select the relevant path to install the required software. The CD drive is assumed to be drive “D” in the example below.

To install ET 98:	D:\install\ET98\cd\Setup.exe
To install ET 98plus:	D:\install\ET98plus\cd\Setup.exe
To update ET 98:	D:\install\ET98plus.UPD\cd\Setup.exe
To upgrade ET 98plus:	D:\install\ET98plus.UPG\cd\Setup.exe

Installation from diskette

Generating the installation diskettes.

The software is supplied on a CD-ROM. Installation diskettes can be generated simply by copying the corresponding folders from the CD to diskettes.

The CD drive is assumed to be drive “D” in the example below.

Generating ET 98 diskettes:
Copy the complete contents of D:\install\ET98\disk1 to ...\disk4 onto four diskettes.

Generating ET 98plus diskettes:
Copy the complete contents of D:\install\ET98plus\disk1 to ...\disk6 onto six diskettes.

Generating an ET 98 update diskette:
Copy the complete contents of D:\install\ET98plus.UPD\disk1 onto a diskette.

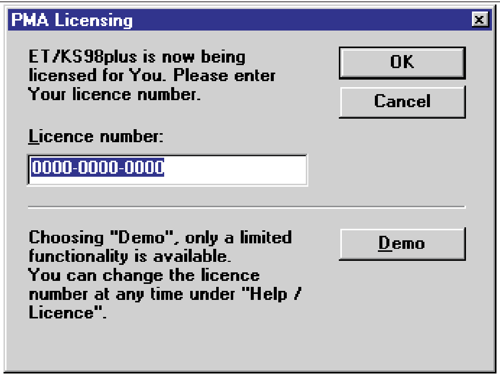
Generating ET 98plus upgrade diskettes:
Copy the complete contents of D:\install\ET98plus.UPG\disk1 to ...\disk6 onto six diskettes.

Installation: Place the first diskette into the drive. Click on “Start” (or open Program Manager) and select “Execute…” (open “File”). Enter ”A:SETUP” (or ”B:SETUP” if your diskette drive is “B:”), and follow the instructions on the screen. Depending on the language you are working with, not all diskettes might be needed.

II-2.3 Licencing

During initial installation of the engineering tool, an input mask (→ Fig.:36) for entry of the licence number is displayed. Unless a licence number is entered, the engineering tool starts only as a demonstration version with limited functions (saving and downloading of an engineering into KS 98-1 is not possible in the demonstration version). The licence number is given on the enclosed registration form. Please, keep the registration form carefully. You need the licence number in case of re-installation and when you have to make use of the technical support. Please, complete the registration sheet immediately and send it to the specified address via fax or as a copy via mail. In return, you will get technical support and regular information on product updates.

Fig.: 36


PMA Licensing

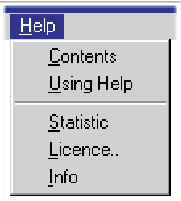
ET/KS98plus is now being licensed for You. Please enter Your licence number.

Licence number:

0000-0000-0000

Choosing "Demo", only a limited functionality is available. You can change the licence number at any time under "Help / Licence".

Fig.: 37



Please, note the PMA licence conditions for software products. Successful installations are only possible onto a hard disk, but not on a network drive (only on request).

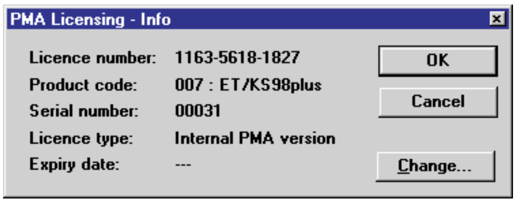
Updates

The licence number is stored in the system and need not be re-entered in case of an update.

Changing the licence number

A change of the licence number or licensing of a demonstration version (conversion into a full version) can be done via the menu bar (→ Fig.:37)"Lizenz". In the window (→ Fig.:38) displayed after clicking on Licence, input mask 'PMA licensing' (→ Fig.:36) can be called up via *Edit*. Now, the new licence number can be entered in this input mask.

Fig.: 38


PMA Licensing - Info

Licence number: 1163-5618-1827
Product code: 007 : ET/KS98plus
Serial number: 00031
Licence type: Internal PMA version
Expiry date: ---

II-2.4 Software start

Starting software "engineering tool KS 98-1" is by double-clicking on the icon created by the installation program in program group "PMA Tools".

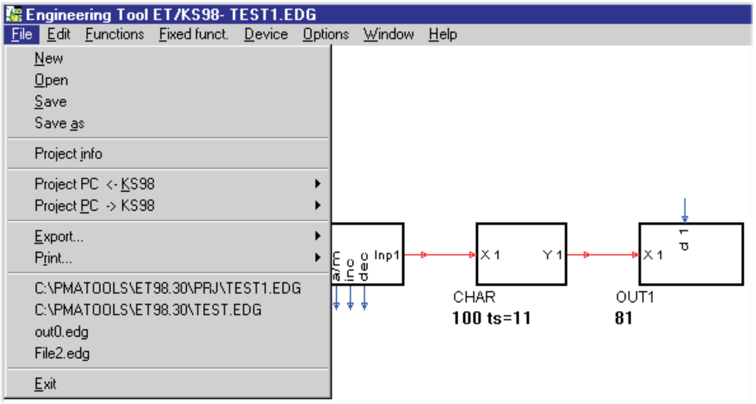
ET-KS98 V3

II-3 Menu reference to the engineering tool

II-3.1 Menu 'File'

This menu item permits standard data handling functions which are also known from other Windows programs (→ see Fig.: 39). Via this menu, e.g. finishing the program is possible.

Fig.: 39



New

Select command "New..." in the file menu, if you want to open an empty engineering without title. Working width / height and scroll bars are set to standard values. The existing engineering is removed from the working memory.

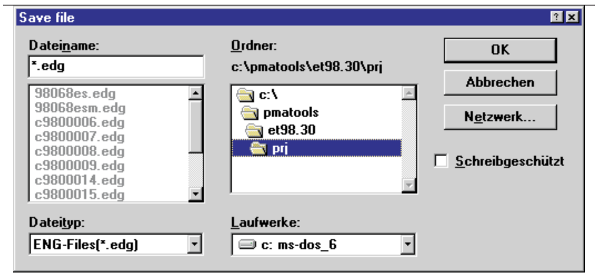
Open

This function can be used for reading in already created engineerings. After selection of this command, a standard dialogue box (→ Fig.: 40) in which the relevant drive as well as the required path and file name are selected is displayed.

Via drop-down list "File format", adjustment which kinds of files are displayed in the file list is possible.

After confirmation with OK, the file is loaded.

Fig.: 40



Dateiname: *.edg

98068es.edg
98068esm.edg
c9800006.edg
c9800007.edg
c9800008.edg
c9800009.edg
c9800014.edg
c9800015.edg

Ordner: c:\pmatools\et98.30\prj

c:\
pmatools
et98.30
prj

□ Schreibgeschützt

Dateityp: ENG-Files(*.edg)

Laufwerke: c: ms-dos_6

This command can be used for loading a stored engineering for editing. If write-protected is selected, a new name is always required when saving (Save as).

Save

This function can be used for saving an engineering created by you as a file. Storage is with the file name used when reading in. If no name exists (new Engineering), select the required path, enter a valid name (with omission of the file extenstion, standard extension .EDG is used automatically) and acknowledge with OK. If there is already a file with the same name, a message is displayed. When saving this file repeatedly during an operation, re-selection of this menu item is sufficient, whereby the name need not be entered repeatedly.

Save as

This function permits saving of an already loaded project under a different name. For this, enter a new name into the field provided for this purpose. If the file extension is omitted, the file is saved automatically with extension .EDG.

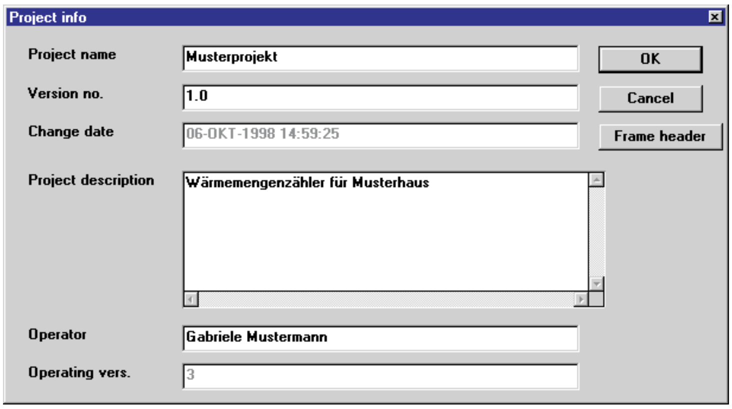
Project info

Execution of this command is followed by display of an input mask for specification of general information on the project. Date of modification and operating version are entered automatically.

The following parts of the project info are stored in KS 98-1:

the first line '*Project name*' (max. 45 characters can be edited freely) the modification data and the operating version

Fig. 41



Project info

Project name: Musterprojekt

Version no.: 1.0

Change date: 06-OKT-1998 14:59:25

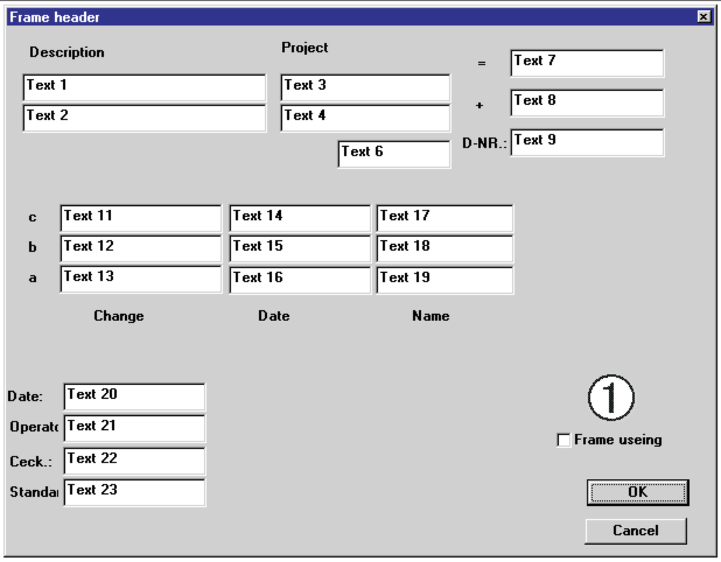
Project description: Wärmemengenzähler für Musterhaus

Operator: Gabriele Mustermann

Operating vers.: 3

After clicking on button Frame header a window for text entry for the drawing header is opened (→ Fig.:42).

Fig.: 42



Frame header

Description: Text 1, Text 2

Project: Text 3, Text 4, Text 6

= Text 7

+ Text 8

D-NR.: Text 9

	Change	Date	Name
c	Text 11	Text 14	Text 17
b	Text 12	Text 15	Text 18
a	Text 13	Text 16	Text 19

Date: Text 20

Operat(Text 21

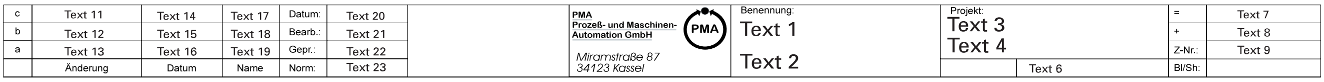
Ceck.: Text 22

Standa(Text 23

□ Frame useing

For print-out with drawing header, tick box 'Use frame' (→ see ⓘ Fig.:42).

Fig.: 43 Frame header



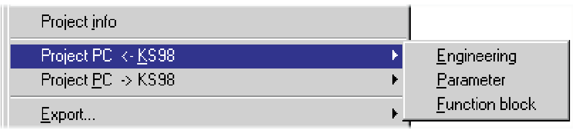
c	Text 11	Text 14	Text 17	Datum:	Text 20		PMA Prozeß- und Maschinen-Automation GmbH Miramstraße 87 34123 Kassel	Benennung: Text 1 Text 2	Projekt: Text 3 Text 4		=	Text 7
b	Text 12	Text 15	Text 18	Bearb.:	Text 21						+	Text 8
a	Text 13	Text 16	Text 19	Gepr.:	Text 22						Z-Nr.:	Text 9
	Änderung	Datum	Name	Norm:	Text 23					Text 6	Bl/Sh:	

(Modifying the frame header; see also the section "Print → Graphic print-out with frame header")

Project PC ← KS 98-1

After calling up this menu item, an additional selection (see Fig. 44) is displayed.

Fig.: 44



Engineering

Complete engineering read-in.

Para/Config

Configuration and parameter data read-in. To ensure correct data read-in, the engineering in the instrument and in the engineering tool must be equal.

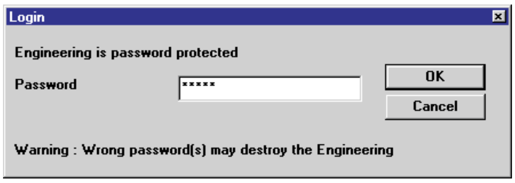
Function block

Configuration and parameter data read-in of a function block marked in the engineering.

☞ With password protection of an engineering, dialog box “Log in” (see Fig.: 45) is displayed. This dialog box asks you to enter the password for the existing engineering.

☞ Exceeding the “number of permitted faulty attempts” (see Fig.: 51) deletes the engineering in the multi-function unit

Fig.: 45



Login

Engineering is password protected

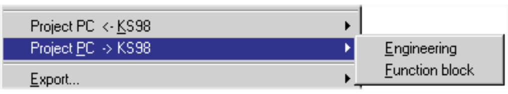
Password: *****

Warning : Wrong password(s) may destroy the Engineering

Project PC → KS 98-1

Function block
Transmission of configuration and parameter data of a function block marked in the engineering.

Fig.: 46



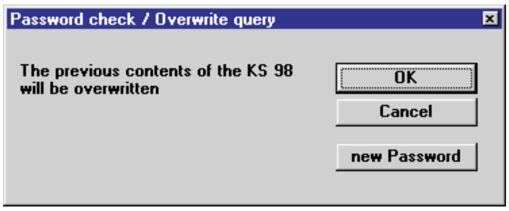
After calling up this menu item, an additional selection is displayed (→ Fig.: 46).

Engineering

After selecting this menu item, a dialogue box is displayed (→ Fig.:32).

When clicking on the OK button, the actual engineering is transmitted into KS 98-1 without password protection.

Fig.: 47

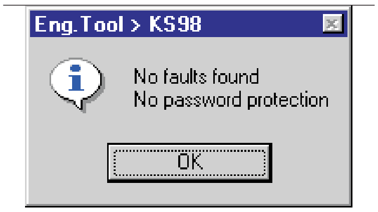


Password check / Overwrite query

The previous contents of the KS 98 will be overwritten

The project stored in the instrument so far is overwritten. The possible messages are shown below.

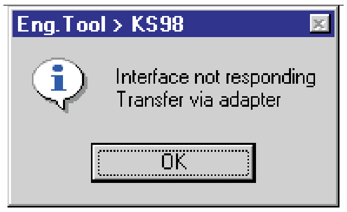
Fig.: 48



Eng.Tool > KS98

No faults found
No password protection

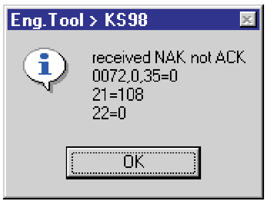
Fig.: 49



Eng.Tool > KS98

Interface not responding
Transfer via adapter

Fig.: 50



Eng.Tool > KS98

received NAK not ACK
0072,0,35=0
21=108
22=0

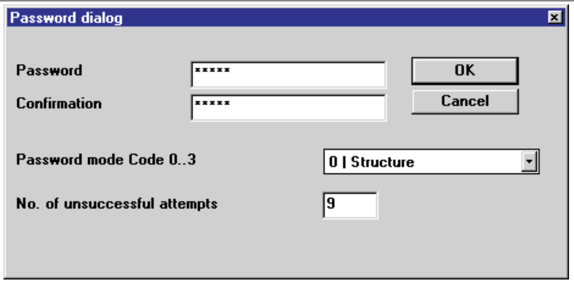
When clicking on button “new password” in the dialogue box, the password dialogue is opened (see Fig.: 51).

Enter password, password mode and number of permitted faulty attempts.

When clicking on button OK, the current engineering is transmitted into KS 98-1 with password protection.

Thereby, the project stored in the unit so far is overwritten.

Fig.: 51



Password dialog

Password: *****

Confirmation: *****

Password mode Code 0..3: 0 | Structure

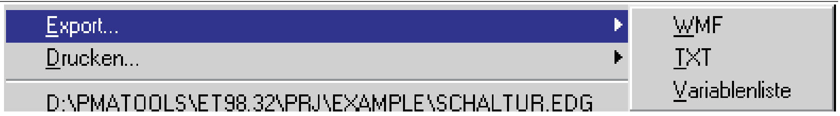
No. of unsuccessful attempts: 9

The message shown in Fig.:51 indicates an error in the transmitted data and is used for trouble shooting in case of technical questions.
The password mode determines the access privilege (≥ see following table) to the KS 98-1 data via the interface.

Export

After calling this instruction, you must define whether the parameters and configuration data of the individual function blocks (text file ⟶ see Fig.: 52), the graphical engineering (graphics in .wmf format) or the variables list are to be exported.

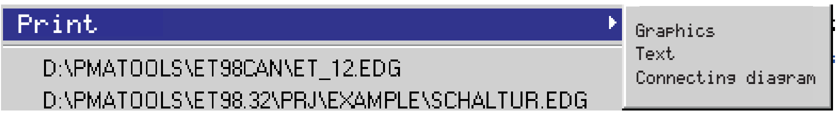
Fig.: 52



Print

After calling up this menu item, an additional selection is displayed (⟶ Fig.:53).

Fig.: 53


Print
D:\PMATOOLS\ET98CAN\ET_12.EDG
D:\PMATOOLS\ET98.32\PRJ\EXAMPLE\SCHALTUR.EDG
Graphics
Text
Connecting diagram

Graphics

Engineering print-out

Text

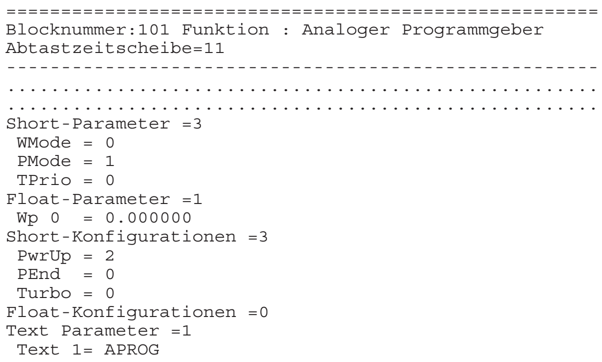
Print-out of parameter and configuration data of individual function blocks (text output ⟶ see Fig.:54)

Connecting diagram

Print-out of the connecting diagram (see page55 for connecting diagram)

Subsequently, the standard mask for adjusting printer functions under Windows is displayed. The possible settings are described in detail in the Windows documentation.

Fig.: 54



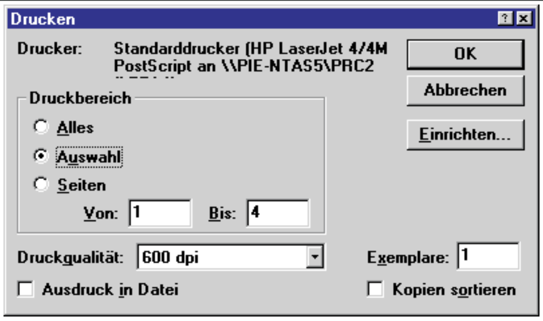
```
=======================================================
Blocknummer:101 Funktion : Analoger Programmgeber
Abtastzeitscheibe=11
-------------------------------------------------------
.......................................................
.......................................................
Short-Parameter =3
 WMode = 0
 PMode = 1
 TPrio = 0
Float-Parameter =1
 Wp 0  = 0.000000
Short-Konfigurationen =3
 PwrUp = 2
 PEnd  = 0
 Turbo = 0
Float-Konfigurationen =0
Text Parameter =1
 Text 1= APROG
```

The data of the actual project are output in a standard form on the connected printer. The current standard printer adjusted under Windows with standard font is used for this purpose (MS Sans Serif 2,8mm).

Print-out of a section

For printing parts of an engineering, the part to be printed must be marked in the survey mode. In the standard printer mask (see Fig.:55), click on "Mark" ("Mark" with Windows 95) before starting the print-out. This is only possible with graphic print-out. Printing out a section with drawing header is not possible.

Fig.: 55



Drucken

Drucker: Standarddrucker (HP LaserJet 4/4M PostScript an \\PIE-NTAS5\PRC2

Druckbereich

○ Alles
● Auswahl
○ Seiten

Von: 1 Bis: 4

Druckqualität: 600 dpi

Exemplare: 1

□ Ausdruck in Datei

□ Kopien sortieren

Graphic print-out with frame header

Graphic print-out is possible with or without frame header. As standard, printing is without frame header. Selection is in input mask 'Drawing header' (→ p. 55 Fig.: 42)

In the drawing header, additional information such as operator, date, version no. etc. can be entered. On the left beside the PMA logo, there is free space for a customer-specific logo. The PMA logo can be removed or replaced by a customer-specific logo.

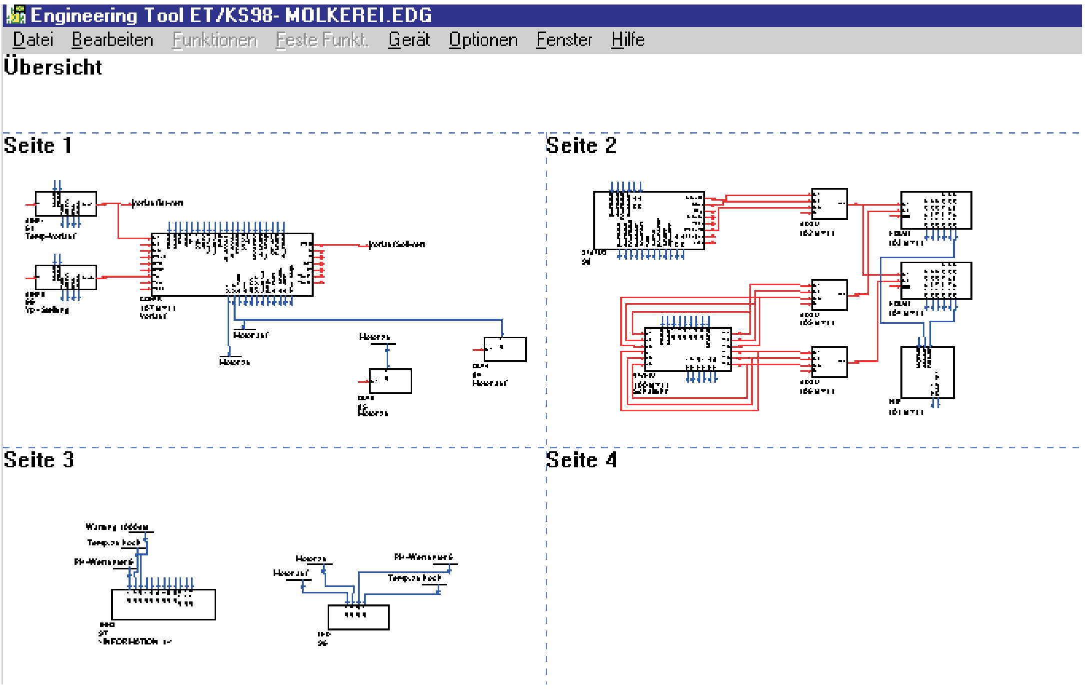
The frame header is in .wmf format (in ...\PMATools\Et98.xxx\Framexd.wmf) and can be edited with normal graphic programs(e.g. CorelDraw). However, fields and frames should not be changed to prevent faulty positioning of the texts entered in the engineering tool.

Page grid in the engineering survey

An engineering can be printed out either completely on only one page, or as a marked section.
By mouse-clicking (left key) in any position of the engineering with Ctrl key pressed, a page grid on which the engineering can be printed out is included into the survey (→ Fig.:56). The pages are numbered linewisely from left to right and from top to bottom in the print-out and can be printed out with or without frame header. Pagewise print-out is prepared by ticking 'pages' in the standard printer mask (see Fig.:55) Pressing Ctrl key + clicking with the mouse in the top corner of the survey removes the grid.

In the normal view, the page borders are shown as broken lines.

Fig.: 56 Setting up a page grid



C:\PMATOOLS\ET98.20D\PRJ\xx.edg (last 4 projects)

The last 4 handled or stored projects are displayed. The project is loaded after clicking on it.

Exit

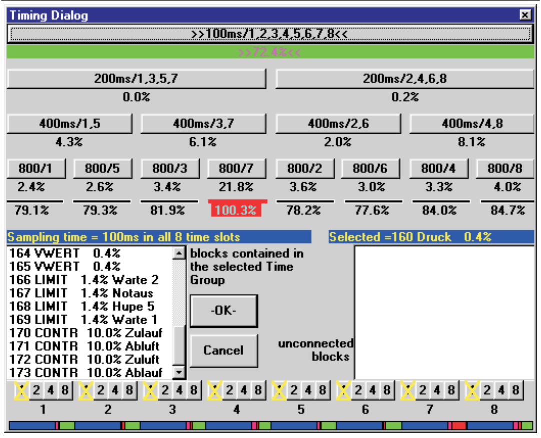
This command finishes working with the engineering tool. Moreover, finishing the program via the system menu field is possible, as in every Windows-supported program. For this, select option "Exit" accordingly. Before finishing, you may be asked, if you want to save the changes in the project handled last. If you deny, the changes are canceled, otherwise, they are stored. Select "Cancel", or press key "ESC" for exit from the dialogue box and returning to the current project.

II-3.2 Menu 'Edit'

Timing

This command calls up the timing dialogue in the normal view. This dialogue can be used for determination of the time slot assignment for each function block. Time slot assignment can be done also from the parameter dialogue (→ Fig.: 59). In the survey, a handling simulation which indicates the order in which the function blocks are calculated is called up.

Fig.: 57



164 VWERT 0.4%
165 VWERT 0.4%
166 LIMIT 1.4% Warte 2
167 LIMIT 1.4% Notaus
168 LIMIT 1.4% Hupe 5
169 LIMIT 1.4% Warte 1
170 CONTR 10.0% Zulauf
171 CONTR 10.0% Abluft
172 CONTR 10.0% Zuluft
173 CONTR 10.0% Ablauf

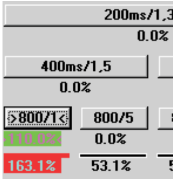
The timing dialogue shall indicate the time consumed by the function blocks in the time slots, as well as the handling sequence. Therefore, two different time slot selection functions are available. The timing sequence is displayed in the bottommost line, whilst the individual time slots are displayed at the top. If a scanning time of a time time slot is selected, all function blocks which are calculated in this time slot part are displayed in the left box. Thereby, the order corresponds to the timing sequence.

The right box contains all function blocks which are not allocated to a time slot so far.

If a function block is selected, it can be shifted into this time slot by mouse clicking on the relevant field. Clicking outside a time slot will remove it from a time group without re-allocating it.

If several function blocks are to be moved simultaneously, they can be selected by means of or   with the left mouse key. As default, the engineering tool assigns the newly placed function blocks to the time slot with 100 ms scanning time. The total of calculation times of all function blocks per time slot must not exceed 100%.
A calculation time of a time slot exceeding 100% is displayed by a colour change (red) in the timing dialogue (see Fig.:58).

Fig.: 58



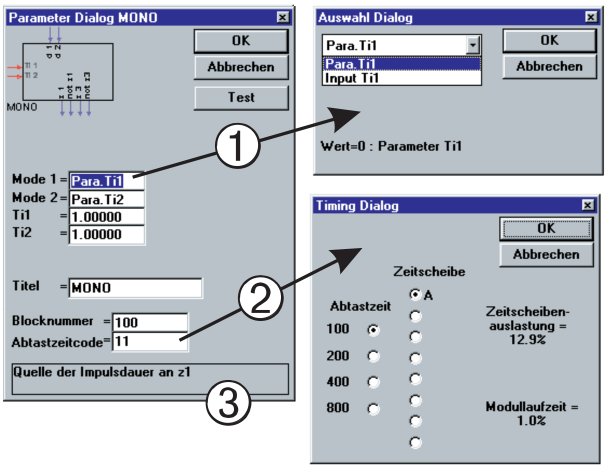
In this case, the function block repartition must be changed.

The survey display shows a simulation that indicates in which sequence the function blocks are calculated. The sequence is either shown automatically, or is defined individually by the user via the keys v (= forwards) and r (= backwards).

Parameters

This command can be used for calling up the parameter dialogue of a selected function (→ Fig.: 59). Its function is equivalent to clicking with the right mouse key with the function marked. In the parameter dialogue, the parameter and configuration data of the function blocks are adjusted. With parameter or configuration data with which the required setting can be selected from a text group, an additional dialogue in which the possible settings are offered via a drop-down item is called up ① at the moment at which the text is changed. A help text concerning the relevant parameter is displayed in a frame ③ .

Fig.: 59



Mode 1 = Para.Ti1
Mode 2 = Para.Ti2
Ti1 = 1.00000
Ti2 = 1.00000

Titel = MONO

Blocknummer = 100
Abtastzeitcode = 11

Quelle der Impulsdauer an z1

Para.Ti1
Para.Ti1
Input Ti1

Wert=0 : Parameter Ti1

Zeitscheibe

Abtastzeit

100 ●
200 ○
400 ○
800 ○

● A
○
○
○
○
○
○
○

Zeitscheiben-auslastung = 12.9%

Modullaufzeit = 1.0%

Moreover, the block number and the time group allocation (scanning time code ②) can be changed, if necessary. The block number determines the handling sequence within a time slot. The block number can be changed into any value required for this function. Normal calculating functions can be set to block numbers ranging within 100 and 450. If the new block number is already used, all other functions are shifted upwards by one, until a free block number is reached. Unless a free block number can be found, allocation of block numbers is refused

Delete

Deletes the marked function or connection (Del key)

Wiring / Positioning

Toggles between the wiring and the placement mode. Switching over can be done also by double-clicking with the left mouse key.

In the wiring mode, connections can be created, changed and deleted.
The placement mode permits function placement, shifting and parameter setting.

Standard connection

A connection which was changed manually can be re-calculated automatically via this command.

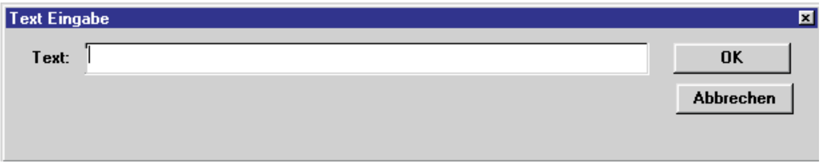
Survey (Key ‘a’)

Display of the overall engineering or switch-over to normal view (wiring mode).When clicking in the survey with the right mouse key, the engineering tool changes to the clicked position in the normal view. An area marked in the survey can be printed out selectively (see page 58).

Add text

If this instruction is called, a window is opened, in which a single-line text block with max. 78 characters can be entered (see Fig.: 60). This text serves as a brief explanation at any point in the engineering. The text can be moved and deleted like any other block.

Fig. 60

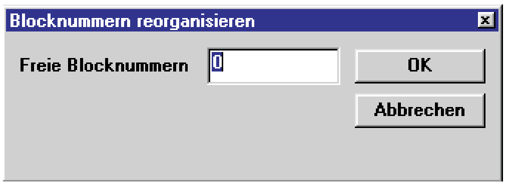


Reorg Block No.

Subsequent deletion of functions produces “gaps” in the list of assigned block numbers. The menu item “Reorg Block No.” opens a dialog box (see Fig.:61).

After confirmation with “OK”, all the block numbers are renumbered consecutively. If the default value “0" of the parameter ”Free block numbers" is changed (e.g. =10), all block numbers are increased by the specified value, and free spaces are eliminated.

Fig.:61



Blocknummern reorganisieren

Freie Blocknummern: 0

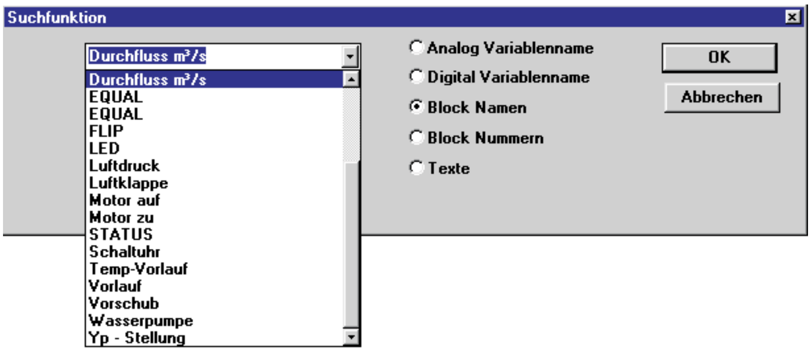
Search

Selecting this menu item opens the window “Search function” (see Fig.: 62).

It is possible to search for existing elements in the different groups.

All possible items are displayed in a list box, where they can be selected. If the search is successful, the corresponding part of the engineering is displayed, whereby the identified element appears inverse. Fig.:62 shows examples of block names.

Fig.:62



Suchfunktion

Durchfluss m^3/s

- Durchfluss m^3/s
- EQUAL
- EQUAL
- FLIP
- LED
- Luftdruck
- Luftklappe
- Motor auf
- Motor zu
- STATUS
- Schaltuhr
- Temp-Vorlauf
- Vorlauf
- Vorschub
- Wasserpumpe
- Yp - Stellung

○ Analog Variablenname
○ Digital Variablenname
● Block Namen
○ Block Nummern
○ Texte

Enlarge sheet

With a very extensive engineering, increasing the window greater may be necessary.

Move

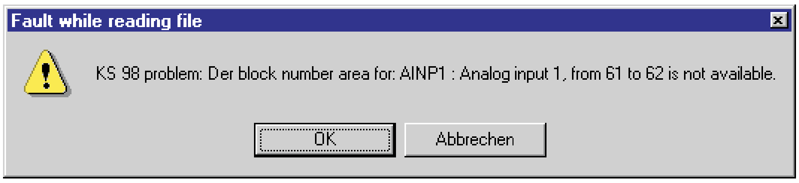
Shifts the overall engineering horizontally and vertically.

The x value shifts in horizontal and the y value shifts in vertical direction. Negative values shift left or up. A value of 100 shifts by approx. the size of a small function block.

Add

This command can be used for adding saved engineerings, inclusive of all adjusted parameters, to the one loaded instantaneously. Unless further block number ranges for defined functions are free in the current engineering, an error message (→ see Fig.:63) is displayed.

Fig.: 63



Fault while reading file

KS 98 problem: Der block number area for: AINP1 : Analog input 1, from 61 to 62 is not available.

Recurrent individual parts can be composed quickly into new engineerings by saving them (e.g. programmer, controllers with particular parameter settings, etc.)

Undo (Ctrl + Z)

This command can be used for canceling the last action.

II-3.3 Menu 'Functions'

Menu 'Funktionen' permits selection of all KS 98-1 software functions with variable block number. After clicking on 'Functions', a list of the functions classified in groups is displayed. When clicking on a function group (e.g. scaling and computing 2) the software functions belonging to this group are displayed as block diagrams
(→ Fig.:64). Click on a block diagram to select this function. The selected function is displayed in the bottom left status field and remains selected, until it is replaced by another function.

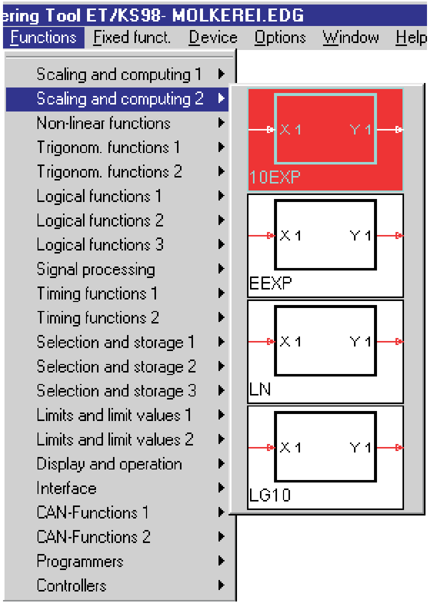
Selection of a function existing in the engineering selects this function for positioning.

Clicking with the right mouse key places the selected function in the mouse pointer position.

Further functions can be selected under menu 'Fixed funct.'.

Depending on the selected operating version, the displayed functions can vary.

Fig.: 64



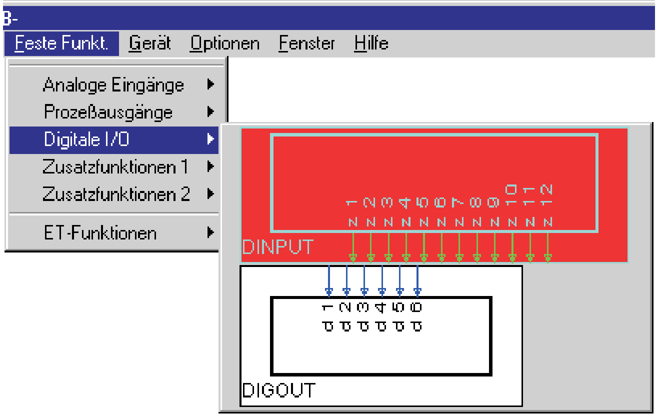
II-3.4 Menu 'Fixed funct.'

Menu 'Fixed functions' permits selection of all KS 98-1 software function with fixed block number. After clicking on 'Fixed functions', a list of the functions classified in groups is displayed. When clicking on a function group (e.g. Digital I/O) the software functions belonging to this group are displayed as block diagrams
(→ Fig.: 65). This function is selected by clicking on a block diagram. The selected function is displayed in the bottom left status field and remains selected, until it is replaced by another function.

Clicking with the right mouse key places the selected function in the mouse pointer position.

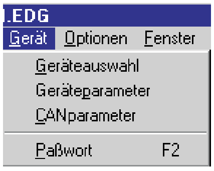
Fixed functions can always be selected only once. These functions are e.g. for inputs or outputs, or status functions. All these functions are sorted automatically in the block number range below 100.

Fig.: 65



II-3.5 Menu 'Device'

Device selection



This command is used for selecting the KS 98-1 industrial controller version.

The mask displayed after execution of the command is displayed in Fig.:66.

Selection of the instrument version is possible via the drop-down items.

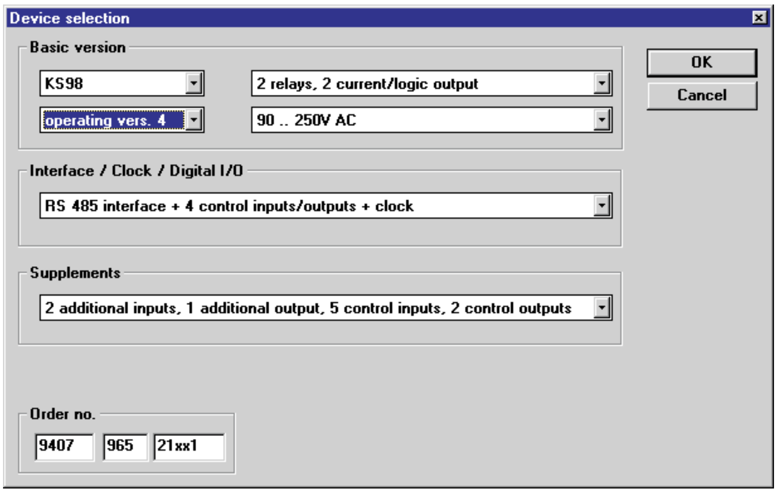
The order number resulting from the version selection is displayed on the bottom left. The inverse procedure

(entry of order number ≥ display of instrument version) is also possible.

For storage of the selected setting, the selection must be validated by clicking on "OK".

Clicking on button "Cancel" will cancel the current selection.

Fig.: 66



Basic version
KS98
2 relays, 2 current/logic output
operating vers. 4
90 .. 250V AC

Interface / Clock / Digital I/O
RS 485 interface + 4 control inputs/outputs + clock

Supplements
2 additional inputs, 1 additional output, 5 control inputs, 2 control outputs

Order no.
9407 965 21xx1

Instrument parameters

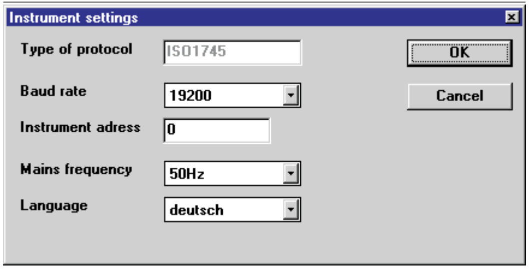
In the input mask (⟶ see Fig.: 67) the settings of KS 98-1 instrument data, address, Baudrate, mains frequency and language are entered. These data are transmitted to KS 98-1, provided that the instrument version is correct.

Freezing the outputs for download

With this field enabled, the engineering is prepared in such a way that during the next download to the multi-function unit, the outputs are frozen in their present status. With the field disabled, the outputs are switched off during this time.

This means that the selected function only becomes effective after the next download.

Fig.: 67



Type of protocol ISO1745
Baud rate 19200
Instrument adress 0
Mains frequency 50Hz
Language deutsch

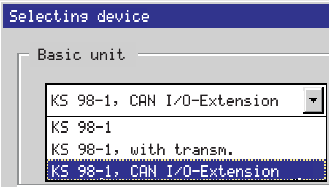
CANparameter

The menu item “CANparameter” can only be selected, if “KS 98-1, CAN I/O extension” has been enabled during device selection (see Fig.:68).

The window “CANparameter” (see Fig.:69) is used to define whether the device is a CAN_NMT (Master) or a CAN_SLAVE. Make sure that the CAN_Baudrate has been adjusted to the same transmission speed in the entire CAN network.

Various speeds between 10 kbytes and 1000 kbytes are selectable (default value is 20 kbytes).

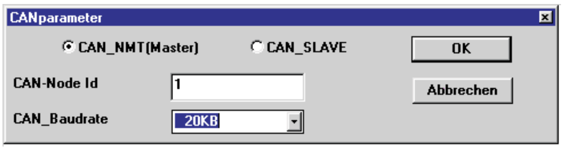
Fig.:68



Selecting device

Basic unit: KS 98-1, CAN I/O-Extension

Fig.: 69



CANparameter

● CAN_NMT(Master) ○ CAN_SLAVE

CAN-Node Id: 1

CAN_Baudrate: 20KB

Communication between KS 98-1 and RM 200 modules, KS 800 or KS 816

KS 98-1 communicating with RM 200 modules, KS 800 or KS 816, must be defined as master. In this case, the CAN node ID “1" is automatically assigned to the CAN_NMT (master).

Cross-communication with other KS 98-1 multifunction units

For the cross-communication between several KS 98-1, it is necessary to configure one KS 98-1 as the master, and the other units as slaves. In this case, the CAN node ID “1" is automatically assigned to the CAN_NMT (master). The addresses 2...24 can then be assigned to the CAN_SLAVEs.

Password (F2)

The password can be adjusted and changed via this menu item. Password entry during transmission is possible additionally.

II-3.6 Menu ‘Options’

Communication

This menu item can be used for selecting the interface via which the interface cable is connected to the PC for communication with Industrial Controller KS 98-1. Baud rate and device address are also adjustable. The other communication parameters are fixed and displayed only for information (→ see Fig.: 70). The function “Transmit address and bit rate” refers to the communication parameters defined in the window “Device parameters” (see Fig.:66).

For storage of the selected setting, the selection must be acknowledged by clicking on “OK”. Clicking on button “Cancel” will cancel the current selection.

Fig.:70

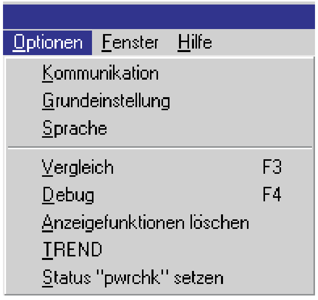
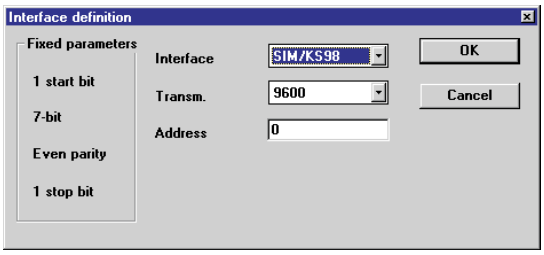


Fig.: 71



Interface definition

Fixed parameters:
1 start bit
7-bit
Even parity
1 stop bit

Interface: SIM/KS98

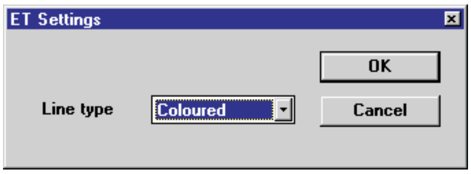
Transm.: 9600

Address: 0

Basic setting

Two display modes are possible. The hatched mode is recommended for print-out on black-and-white printers or displays (e.g. laptop). In all other cases, the colour setting should be used.

Fig.: 73



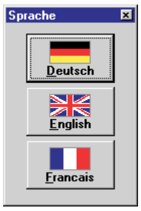
ET Settings

Line type: Coloured

Language

The engineering tool language can be selected.

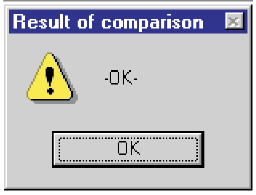
Fig.: 72



Comparison (F3)

Permits comparison of the active engineering with the contents of the connected KS 98-1 or of the simulation. The operation is started immediately after clicking on this item. The comparison result is displayed in a message window. Possible messages are shown in Fig.: 74 to Fig.: 76. (close window with key “x”).

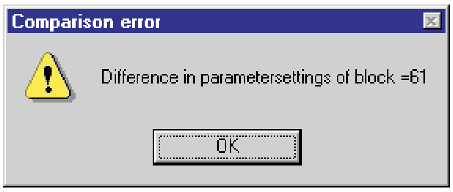
Fig.: 74



Result of comparison

-OK-

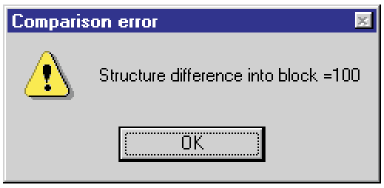
Fig.: 76



Comparison error

Difference in parametersettings of block =61

Fig.: 75



Comparison error

Structure difference into block =100

Debug (F4)

Activates cyclic display of signals in the additionally created monitor blocks (see also chapter 3.5).

Deleting display functions

As display functions are not transmitted to KS 98-1, deleting all these functions at the end of the debug phase may make sense.

Trend

See description of ‘Universal trend function’ (chapter 3.6 from page 72)

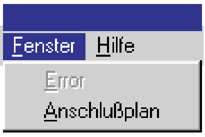
Set status “pwrchk”

The STATUS function provides a digital output “pwrchk” (power fail check), which goes to “0" when the power returns after a failure.

By means of the function “Set status pwrchk” this output can be reset to “1", thus enabling a functional test of the engineering to be triggered after a power failure.

II-3.7 Menu ‘Window’

Error



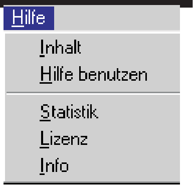
The menu item “Error” enables an error message window to be displayed or hidden, if errors occur during loading or when switching to another operating version.

Menu ‘Window’ permits selection of an error window, or returning to the background, if an error occurred during loading or operating version switch-over.

Connecting diagram

Depending on the selected hardware version and the wired inputs/outputs, a connecting diagram is generated interactively, which also contains the assigned block names.

II-3.8 Menu ‘Help’

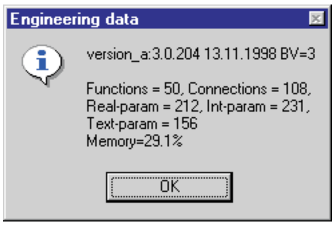


The engineering tool is delivered with an on-line help which is available at any time. How to handle the Windows help system is described in detail in the documentation of the Windows operating system.

Engineering INFO

After execution of this command, the following window with general engineering information is displayed (→ see Fig.:77).

Fig.: 77


Engineering data

version_a:3.0.204 13.11.1998 BV=3

Functions = 50, Connections = 108,
Real-param = 212, Int-param = 231,
Text-param = 156
Memory=29.1%

Licence

Information on your entered licence number required with questions you may have is displayed in this information window. A new licence number can be entered via button Change...
(→ page 53).

Info

When selecting command Info in menu Help, an information window containing the version number and a general information on the engineering tool is displayed. You should
be able to specify this version number, whenever you should have any questions.

II-4 Engineering tool operation

II-4.1 Fundamentals of the engineering tool operation

For operation of the engineering tool, basic knowledge of the Windows operating system is an advantage. Unless you have this knowledge, please, make familiar with Windows first before executing the program for the first time. This is possible using the Windows manual, the Windows on-line help and the Windows learning program.

❑ For all questions in connection with the engineering tool, the KS 98-1 ONLINE HELP (key F1) is available (→ Menu Help).
Press the F1 key with the function marked or with the parameter dialogue opened to start the on-line operating instructions (only possible with the Acrobat reader on the computer installed)

Before making changes to the engineering, the engineering must be stored, since changes cannot be canceled automatically!

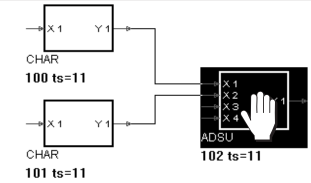
The two operating modes are distinguished by the mouse pointer display (hand symbol / arrow symbol). These two modes are described in the following sections. Switch-over is by double-click with the left mouse key or via 'Menu' → 'Edit' → 'Positioning'/'Wiring'.

II-4.2 Function block placement

A function can be selected either via menu bar 'Functions' or 'Fixed functions', or by entry of the function name in capital letters. With the engineering tool in the placement mode (mouse pointer is a hand symbol), the name of the instantaneously selected function is displayed in the bottommost status line.

By clicking with the right mouse key, the selected function is placed in the actual mouse pointer position.

Fig.: 78



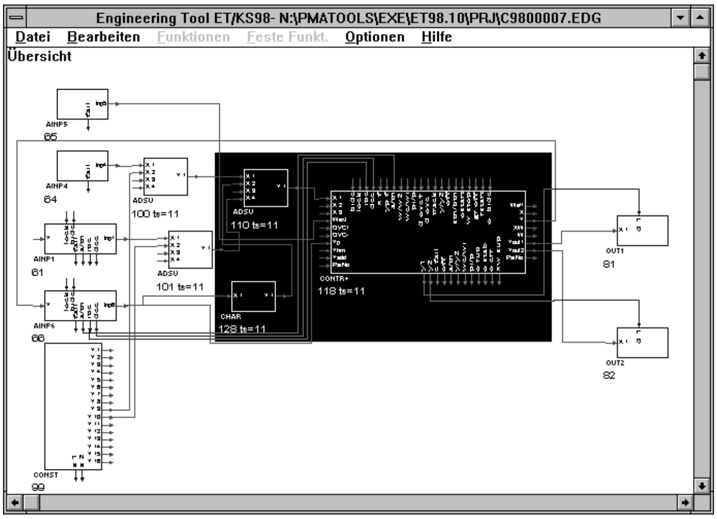
II-4.3 Function block shifting

After clicking with the mouse pointer on a function block, the mouse pointer is displayed inversely (→ see Fig.:78) and can now be shifted either by means of the cursor keys or using the mouse (keep left mouse key pressed). The linked connection lines are dragged. Clicking with the right mouse key with inverse display of the function block opens the parameter dialogue of this function (→ see also page 61). Shifting several function blocks simultaneously is possible only in the survey display ('Edit' → 'Survey' or by pressing key 'a').

Procedure:

Calling up the survey (key 'a').
Use the mouse pointer to draw a window across the function blocks which are to be shifted.
Use the cursor keys or left mouse key to shift the selected area.
Only the function blocks which are fully covered by the marking will be shifted.

Fig.: 79



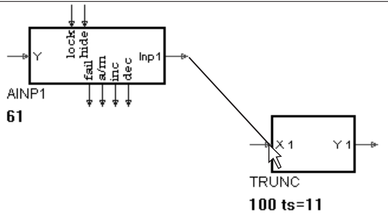
II-4.4 Creating connections

Connections can be created only between analog outputs and inputs or digital outputs and inputs. Connecting digital outputs and analog inputs and vice versa is not possible!

Procedure:

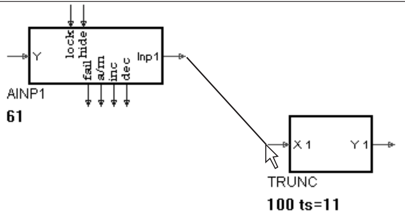
Click on the end point of the output arrow with the left mouse key.
With the mouse key kept pressed, draw the connection to the arrow start of the required input.
Release the mouse key.

Fig.: 80



false

Fig.: 81



correct

An output can be connected with several inputs. However, connecting an input with several outputs is not possible. The connection is displaced according to a standard algorithm.

Connecting additional inputs

If a connection was selected, an additional input can be connected directly by clicking with the left mouse key with key CTRL (Strg) pressed. Select a defined segment of a connection to determine at which segment the new connection is made (see Fig: 82/ 86 and 87 /83).

Fig.: 82

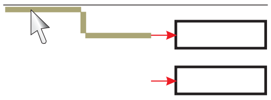


Fig.: 86

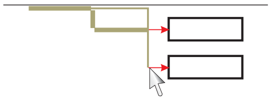


Fig.: 87

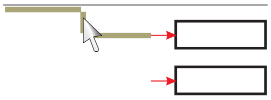
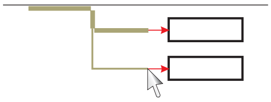


Fig.: 83



Handling connections

For increasing the graphic display clarity, subsequent handling of the connection lines is possible. The left mouse key can be used to select a line in the wiring mode (arrow symbol). The selected line is now displayed in a different colour and thickness. If this line belongs to a network (an output is connected with several inputs), the relevant lines are displayed in the colour of the selected line, but in normal thickness. Now, the individual segments of the selected line can be shifted by positioning the mouse pointer on a segment and with the left mouse key kept pressed. The segments can also be displaced using the cursor keys.

If further line segments are required, the last segment can be prolonged and displaced as a new segment (→ see Fig.: 85 and 84). In this way, max. seven variable segments can be created.

Function ‘Standard connection’ can be used for returning the selected line into its standard form (Function key F11).

Fig.: 84

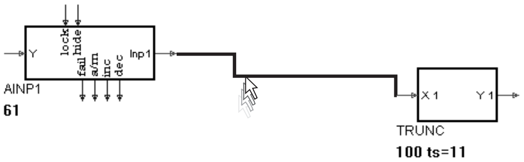
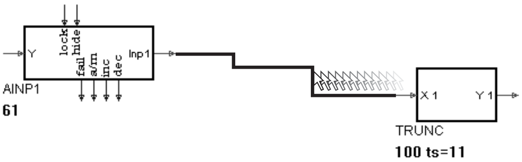


Fig.: 85



Changing signal source connections

In order to prevent the necessity to delete all connections and to remake them manually to the new source when changing the network wiring for another signal source, connecting a complete network to another source is possible. This is done automatically by clicking on the signal source (output) and simply clicking on the new source with key Ctrl (Strg) pressed. Thereby, all inputs are connected to the new source automatically.

Overlapping lines

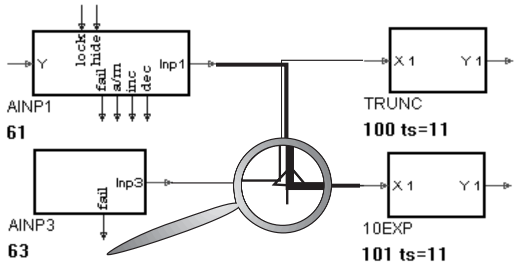
Extensive engineerings frequently feature overlapping lines which do not belong to a network.

Function key F5 can be pressed for searching the engineering for overlapping lines. The first line which is found will be marked.

Pressing function key F6 will search in the engineering and count how many overlappings are found. The line found last will be marked.

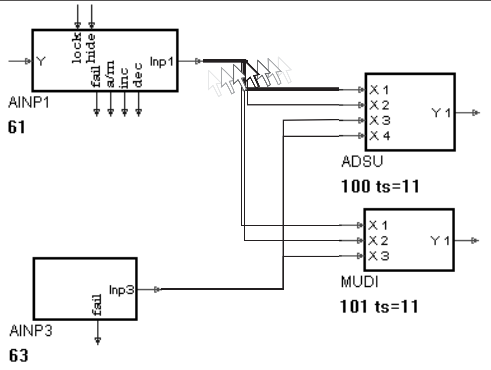
To ensure the clarity of the engineering, these lines shall be shifted apart, until line marking after pressing keys F5 or F6 stops.

Fig.: 88



With networks, concentration of lines running in parallel is desirable. For this, seize a line at the segment which is passed by all lines to be concentrated, and move the marked segment across all network lines (→ see Fig.:89) with the Shift key or the switch-over key pressed. Segment shifting can be done also using the cursor keys. Concentration of the segments belonging to a network is also caused by pressing key F7. Note, however, that the catching area is limited with this function!

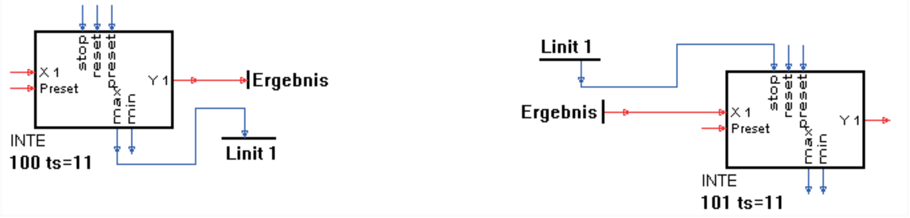
Fig.: 89



Variable editor and virtual links

Via menu item 'Fixed functions' '*ET functions*'(→ p. 63) data sources and destinations (analog and digital) can be selected and inserted into the engineering as special block (→ Fig.: 90). These sources can be assigned to variable names as with all other functions in the parameter dialogue. In the parameter dialogue of data destinations, the already defined variables are displayed in a listbox from which the required variable can be selected and assigned. These 'virtual' links are interpreted as 'full' lines in KS 98-1. Thus e.g. auxiliary functions can be placed at the periphery of the engineering without the need to draw confusing lines through the whole engineering, whereby clarity and readability are improved considerably.
However, these special blocks and their variable names are not stored in KS 98-1 and cannot be reconstructed when reading out directly from a KS 98-1. They are rather shown as full lines.

Fig.: 90

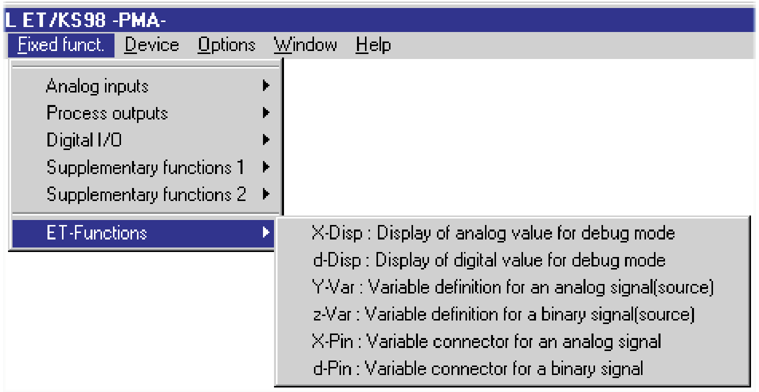


II-4.5 Online operation

Display blocks (analog and digital)

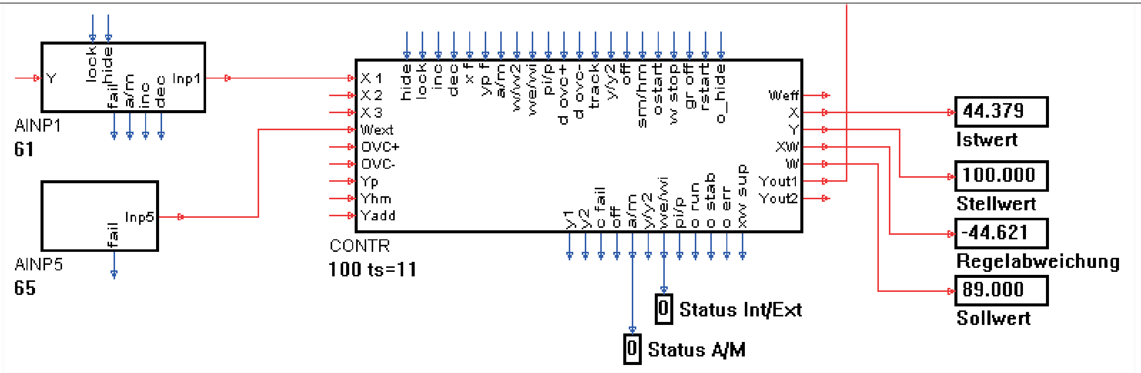
In 'Fixed functions' ⟶ '*ET functions*' (⟶ Fig.: 91) display blocks (X-Disp and d-Disp) can be selected and inserted into the engineering as special block (⟶ Fig.: 92).

Fig.: 91


L ET/KS98 -PMA-
Fixed funct.
Device
Options
Window
Help
Analog inputs
Process outputs
Digital I/O
Supplementary functions 1
Supplementary functions 2
ET-Functions
X-Disp : Display of analog value for debug mode
d-Disp : Display of digital value for debug mode
Y-Var : Variable definition for an analog signal(source)
z-Var : Variable definition for a binary signal(source)
X-Pin : Variable connector for an analog signal
d-Pin : Variable connector for a binary signal

Like all other functions, these blocks can be given names in the parameter dialogue. Via 'Options' ⟶ '*Delete display functions*', all display blocks can be deleted simultaneously e.g. after finishing the engineering test.

Fig.: 92



Debug

The debug mode can be enabled/disabled either via the menu item "Options ⟶ Debug" or directly with F4. Operating data are then exchanged cyclically (approx. 0,5s) with the KS 98-1 or SIM/KS 98-1. The generated display blocks will show the relevant values. Too many display blocks will extend the cycle time.

Parameters can be changed online using the parameter dialog. After confirmation with "OK", they are transmitted to the KS 98-1. The results are displayed immediately.

Parameter changes via the KS 98-1 front panel are not transmitted to the PC during the debug mode.

II-4.6 Trend function

Survey of the characteristics

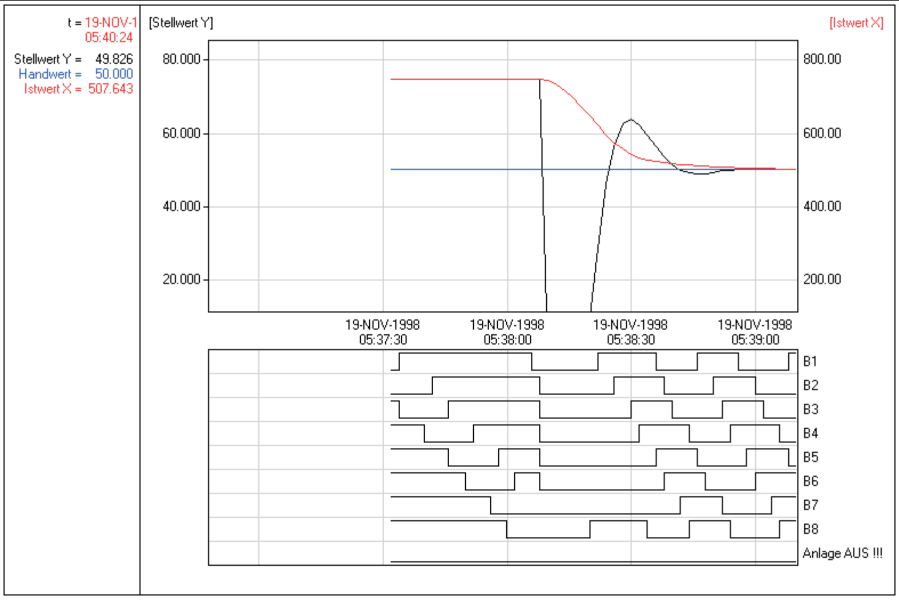
Apart from the trend function specially designed for “Controllers” (CONTR, CONTR+) of the simulation package SIM/KS 98-1, additional trend windows can be implemented. Each trend is able to display 7 analog values and 12 logic states from the Engineering against a time axis. Several independent trend displays can be shown simultaneously.

As soon as the trend function(s) have been called (L1READ), and the Engineering has been transmitted to the KS 98-1 or SIM/KS 98-1, the trend dialog window is opened via the menu item <Options><TREND>.

- Trend dialog window
  This window is used to select the required trend function, its duration, and the time interval. Clicking the START button initiates the trend recording function, and a display window is opened for the selected trend curve. In this way, it is possible to open several display windows.

Display window
Two different scales can be assigned to any pair of measurement values, which simplifies interpretation of the trend curves.
The length of the time axis is adjusted by means of “Sampling cycle time” and “Number of measurement values” (samples) for each trend display. The time axis is either “absolute” with date and time (hh:mm:ss) or “relative” (changeable during the recording).
The exact analog values are shown numerically at top left in the display window. If the cursor line is active, the analog values can be read precisely at any point of time. Trend display can be stopped (frozen) and resumed; the measurement continues in the background!
By holding down the left mouse key, the cursor can be used to mark an area for enlargement.
This zoom function is canceled via <View><Complete display>.

Fig.: 93 Trend display



t = 19-NOV-1
05:40:24

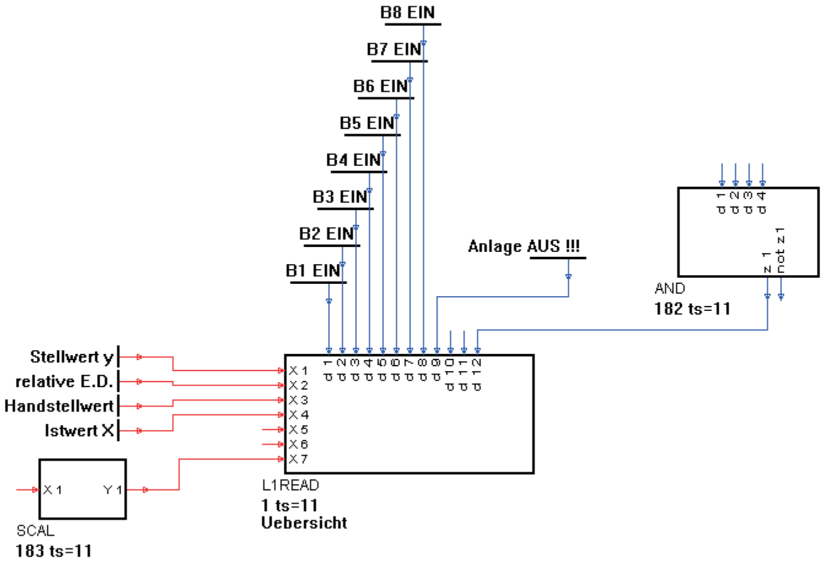
Stellwert Y = 49.826
Handwert = 50.000
Istwert X = 507.643

Preparations in the ET/KS 98

The trend function is an application that runs independently of the Engineering Tool. The values to be displayed are received directly from the KS 98-1 or from the simulation software SIM/KS 98-1. Data trans-mission is executed via the communication modules L1READ (blocks 1...20), which must first be configured. For each L1READ, 7 analog values and 12 logic statuses from the Engineering can be “connected”. Usually, a single L1READ is enough to display the characteristic values of an application in relation to each other. However, up to 20 display windows can be provided. Please note that the com-munication load depends on the transmission cycle adjusted in each display window, and can cause problems. Make sure that the following limits are not exceeded:

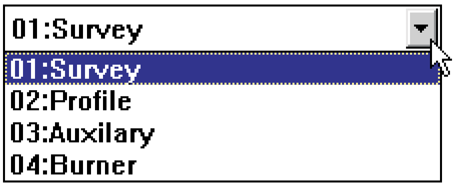
Übertragungszyklus	Anzahl Trendfenster
1s	≤ 2
2s	≤ 4
4s	≤ 8
8s	≤ 16

Fig.: 94 Preparations in the KS 98-1 Engineering



Implementing the trend function

Fig.: 95 Implemented L1READ functions

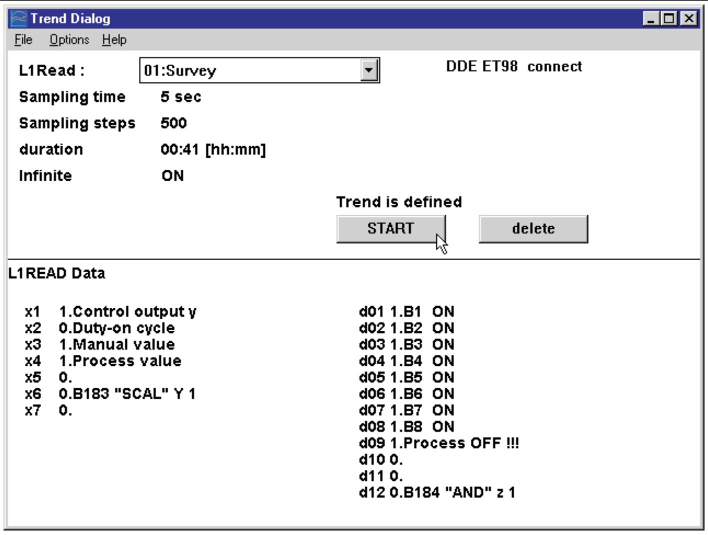


When the preparations in the Engineering have been completed, and it has been transmitted to the KS 98-1 or SIM/KS 98-1, the trend dialog box is opened directly from the Engineering Tool ET/KS 98 by means of <Options><Trend>.

All the L1READ functions implemented in the Engineering, are displayed in a list box with their assigned “names”, from where they can be selected for display.

The lower part of the trend dialog box shows all the signals connected to the selected L1READ function, complete with block number, block title, and connection description or variable name.

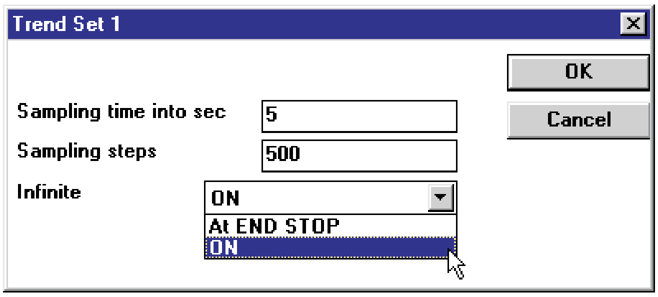
Fig.: 96 Trend dialog box (trend display not yet started)



```
L1Read :         01:Survey                  DDE ET98  connect
Sampling time    5 sec
Sampling steps   500
duration         00:41 [hh:mm]
Infinite         ON
                                 Trend is defined

L1READ Data

x1  1.Control output y               d01 1.B1  ON
x2  0.Duty-on cycle                  d02 1.B2  ON
x3  1.Manual value                   d03 1.B3  ON
x4  1.Process value                  d04 1.B4  ON
x5  0.                               d05 1.B5  ON
x6  0.B183 "SCAL" Y 1                d06 1.B6  ON
x7  0.                               d07 1.B7  ON
                                     d08 1.B8  ON
                                     d09 1.Process OFF !!!
                                     d10 0.
                                     d11 0.
                                     d12 0.B184 "AND" z 1
```

Trend recording of the selected L1READ function can now be started directly with the “Start” button. The trend parameters can be changed previously by means of the “Change” button. The duration of the trend curve (visible time axis) is determined by the product of “Sampling time x sampling steps”. Furthermore, you can define what happens after the selected recording duration has expired: Either the recording is terminated (“Stop at end”) or continued (“Ring memory/moving”). With the second option, previous values are deleted!

Fig.: 97 Behaviour when recording time has expired



```
Sampling time into sec   5
Sampling steps           500
Infinite                 ON
```

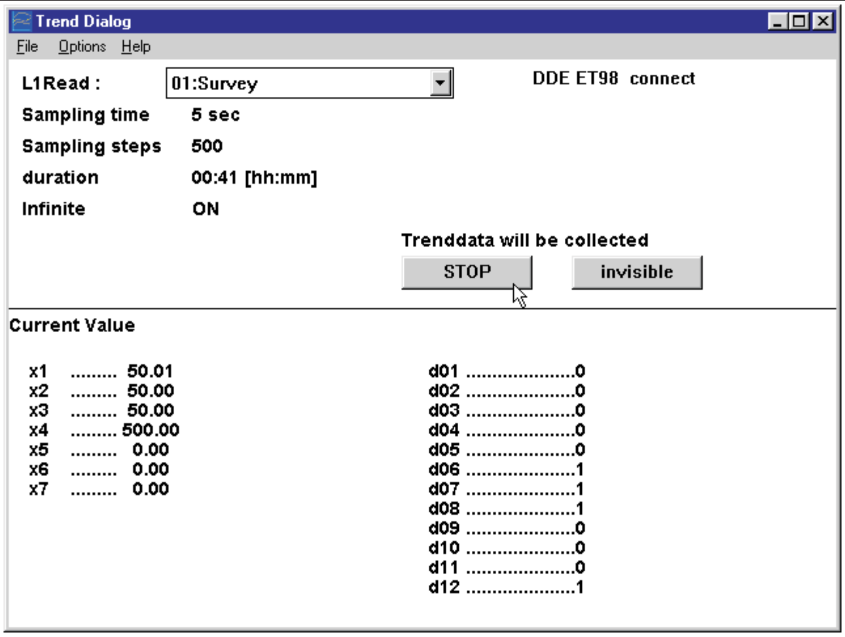
Displaying the trend curves

Clicking the “Start” button initiates the trend recording function and opens a display window. Values are displayed from left to right. Via the buttons in the dialog box, the trend display can be stopped (STOP) or moved into the bac - kground (Invisible), whereby the display window is closed. The lower part of the dialog box now shows a numeric dis - play of the actual values of the connected variables.

Note:

1. When using the simulation software SIM/KS 98-1, the “Turbo” mode should always be switched off.
2. Before starting the universal trend, the controller trend of the SIM/KS 98-1 simulation should be terminated!

Fig.: 98 Trend-dialogwindow (trend running)



L1Read : 01:Survey

DDE ET98 connect

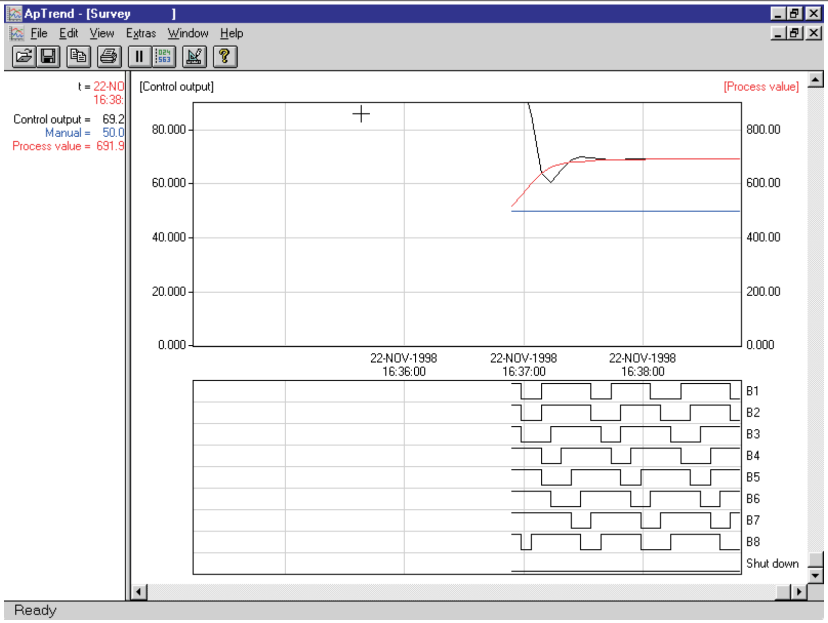
Sampling time 5 sec
Sampling steps 500
duration 00:41 [hh:mm]
Infinite ON

Trenddata will be collected

Current Value

```
x1   ......... 50.01        d01 .....................0
x2   ......... 50.00        d02 .....................0
x3   ......... 50.00        d03 .....................0
x4   .........500.00        d04 .....................0
x5   .........  0.00        d05 .....................0
x6   .........  0.00        d06 .....................1
x7   .........  0.00        d07 .....................1
                            d08 .....................1
                            d09 .....................0
                            d10 .....................0
                            d11 .....................0
                            d12 .....................1
```

Fig.: 99 Graph window



Buttons in the display window

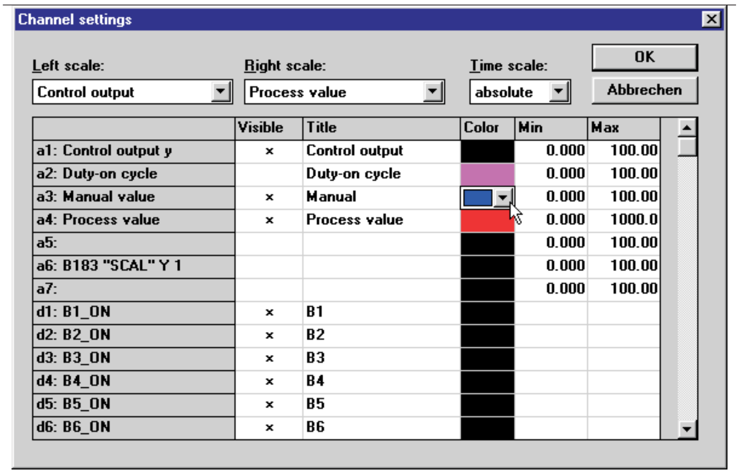
Symbol	Description	Symbol	Description
📂	Open file	⏸	Stop/continue trend recording
💾	Save file	🔢	Enable/disable cursor line
📋	Transfer selected item to the clipboard memory	📐	Parameter dialog for trend adjustments
🖨	Print file	❓	Program information

Adapting the trend curves

The menu items <Extras><Options> in the display window enable you to adapt the trend curves ("Channel settings"). While the trend recording is running, the "Channel settings" window can also be opened from the trend dialog box via <Options><Dialog>. The "Channel settings" are stored together with the Engineering in the KS 98-1 or the SIM/KS 98-1. The following settings can be adapted:

Selection of the graphic curves (x)
Description (title)
Curve colour
Value ranges (min / max)
Assignment of the left / right scales to variables
Time axis (absolute / relative)

Fig.: 100 Parameter adjustments of the trend curves



Channel settings

Left scale: Control output
Right scale: Process value
Time scale: absolute

	Visible	Title	Color	Min	Max
a1: Control output y	×	Control output	⬛	0.000	100.00
a2: Duty-on cycle		Duty-on cycle	🟪	0.000	100.00
a3: Manual value	×	Manual	🟦	0.000	100.00
a4: Process value	×	Process value	🟥	0.000	1000.0
a5:			⬛	0.000	100.00
a6: B183 "SCAL" Y 1			⬛	0.000	100.00
a7:			⬛	0.000	100.00
d1: B1_ON	×	B1	⬛		
d2: B2_ON	×	B2	⬛		
d3: B3_ON	×	B3	⬛		
d4: B4_ON	×	B4	⬛		
d5: B5_ON	×	B5	⬛		
d6: B6_ON	×	B6	⬛		

Calling the trend function without ET-KS 98

The settings in the trend dialog box are saved via the menu items <File><Save as ...> in the file name.dat. This enables the trend dialog box to be opened by calling trend_di.exe, and the required trend recording to be started without the Engineering Tool.

However, a KS 98-1 or the Simulation SIM/KS 98-1 with the corresponding Engineering must be connected.

It is also possible to open a display window directly by opening the corresponding file name.dat, if a link with trend_di.exe is provided under Windows.

Subsequent trend analysis

The contents of a display window can also be saved as a name.trd file, that can be opened later for analysis and evaluation. The cursor line, display options, and the zoom function are active.

II-5 Building an engineering

First steps with the KS98-1 engineering tool

In this chapter, we would like to help you getting started with a concise description:

1. Installing the required tool
2. Projecting/programming a small test project
3. Testing the required instrument behaviour in the simulation
4. Loading the project into KS 98-1
5. Testing the implemented functions
6. Hint to the function library and task-related structures
7. References to further detail information on tools and application

After automatic start via "Installation of PMA Tools", you will find engineering tool KS 98-1-Tools ET/KS98, Sim98-1 and instrument simulation SIM/KS 98-1 on the PMA CD.

Follow the instructions for installation and enter your licence number after being requested to do so (separate numbers for engineering tool and simulator). If you only want to test the KS 98-1 tools without obligation first, our personnel will give you temporary licence numbers for the two tools. Without the licence numbers, you will be able to test the two tools independently, however, without building up communication between instruments. In this case, transmission of a test engineering to the instrument or to the simulation is not possible. For communication with the real instrument, only the engineering tool licence is required.

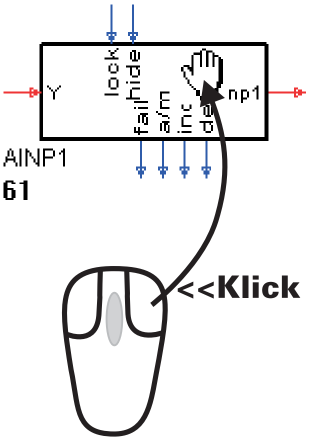
① Start the engineering tool and select the required language (German/English/French) via menu Options .

② Functions and fixed functions.
These menu items contain the function blocks for selection of the relevant function descriptions in this manual.

③ As a simple beginner's project, we want to build up a control loop. For this, we need a process value input (analog input 1) from the fixed functions and a positioning output for the controller correcting variable (analog output 1).

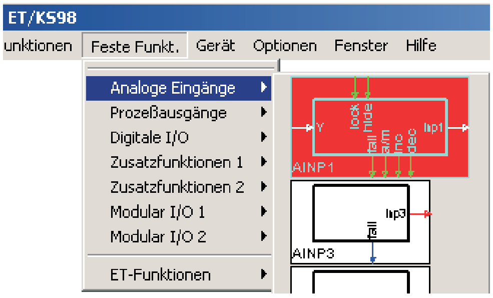
④ We select the input block from the list by clicking with the left mouse key.
Now, position the block on the left of the working surface by clicking with the right mouse key.

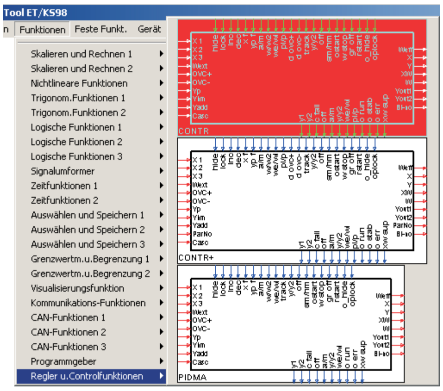
Fig. 101

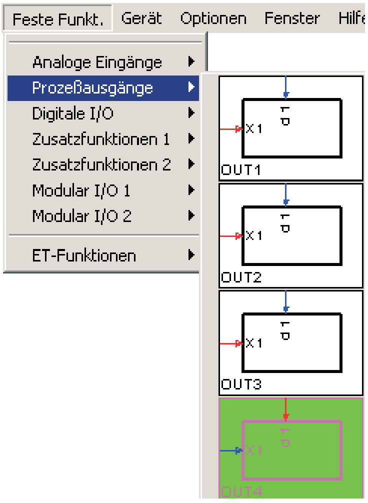


For controlling, we also need a controller (in Functions - Controllers > CONTR).

⑤ This controller should be positioned behind the input.
⑥ The same we do with the OUT4 and place him below the controller.







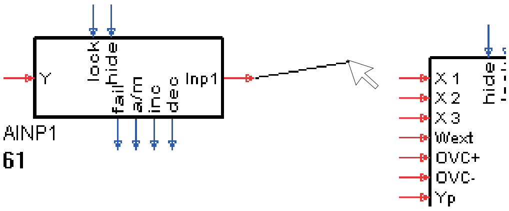
In the next step, we have to soft-wire the function blocks, in order to provide a connection from the input via the controller to the output.

⑦ For this purpose, we can select the wiring mode (via menu Edit – Wiring or by a simple double-click). The mouse pointer as an arrow is displayed.

We use it to draw a line from the tip of the AINP1 (red arrow right) output arrow to controller input arrow X1 by (means of the left key).

⑧ Accordingly, we draw a connection from controller positioning output Yout1 of the controller block to the input arrow of output block OUT4.

Fig. 102



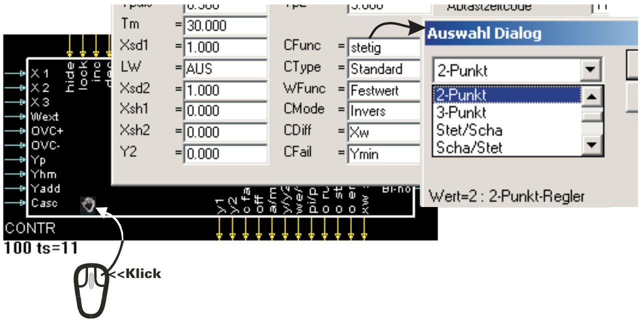
All function blocks have still to be checked for correct parameter setting. AINP1 is set to 0-20mA as default, which is OK to start with. OUT4 is relay to the first output. The controller default setting is " *continuous control*", we set it to 2-point in the field "CFUNC".

⑨ For this purpose, we return to the soft-wiring mode (hand pointer) and click with the right mouse key on the function block for which we are setting the parameters. The individual parameter dialogue of the selected function block appears.

⑩ Now the parameter is selected with a click into the text field. Its signification is displayed in concise form in the bottom field of the input mask.
When opening the input field by clicking twice, a listbox for selection is offered.

⑪ When leaving the parameter dialogue with OK, the block setting is completed.

Fig. 104



Tm = 30.000
Xsd1 = 1.000
LW = AUS
Xsd2 = 1.000
Xsh1 = 0.000
Xsh2 = 0.000
Y2 = 0.000
CFunc = stetig
CType = Standard
WFunc = Festwert
CMode = Invers
CDiff = Xw
CFail = Ymin

Auswahl Dialog
2-Punkt
2-Punkt
3-Punkt
Stet/Scha
Scha/Stet
Wert=2 : 2-Punkt-Regler

The engineering for a simple control loop is also completed. Only the function test has still to be passed. For this, either a KS 98-1 or the simulation are required.

The instrument settings are dependent on whether a simulator or a real instrument are used for testing:

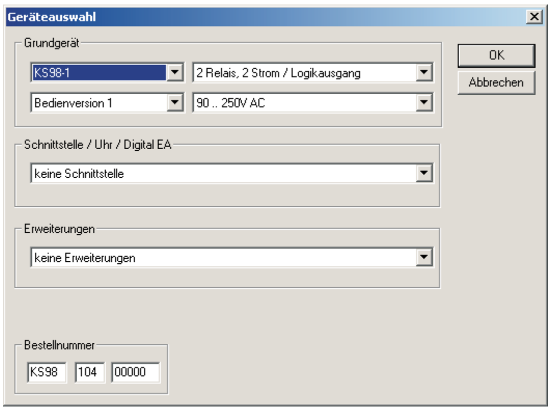
Test with device

In the ETKS98 at→ **device-device selection** the adjustment to the existing device is made.

Therefore the product-code-no. of the device (sideways on the typeplate) is written into the "order-number" field

With OK the adjustment is confirmed.

Fig. 103



Geräteauswahl

Grundgerät
KS98-1
2 Relais, 2 Strom / Logikausgang
Bedienversion 1
90 .. 250V AC

Schnittstelle / Uhr / Digital EA
keine Schnittstelle

Erweiterungen
keine Erweiterungen

Bestellnummer
KS98 104 00000

For communication with the device a serial connection (COMport from the PC) with the device via pc-adapter is set up.

Check → Menu - Options - Communication in the ETKS98, if the interface is set to Com 1.

Test with the Simulation

Start SIM-KS98-1. Select the version to be simulated at→ **Menu -Settings-Device**.

In the **ETKS98** at→ **device-device selection** fill in the product-code-no of the simulation (written in the field order-no of the simulation). Confirmed with OK.

For communication with the simulation, in the engineering-tool the interface is set to SIM/KS98-1 via the **Menu Options-Communication**.

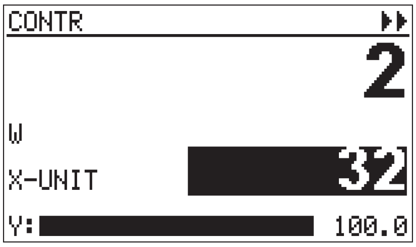
After all this preparations we can load our little test-project, via **Menu File-Project->KS98-1-Engineering**. All control requests we confirm with OK.

After Start-up, we see the adjacent controller page.

Now the setpoint is altered (See: page): Insert a setpoint and confirm it. The controller will adapt the process value (simulated by KS98-1 with its correcting variable reaction.

As a matter of fact, every controller must be matched to the process characteristics. This refers mainly to control parameters Xp1, Tn and Tv.

Fig. 106



CONTR
2
W
X-UNIT 32
Y: 100.0

For detailed information including the self-tuning function, please, see section Controller CONTR, CONTR+ and PIDMA page: 223) in this manual.

Detailed information on this subject and on the self-tuning possibility is given in this manual and in section Controller (CONTR, CONTR+ and PIDMA) of the operating instructions. The different KS 98-1 operating pages and the operating principle of these pages are described in a separate document (operating manual) apart from the function-related explanations of this manual.

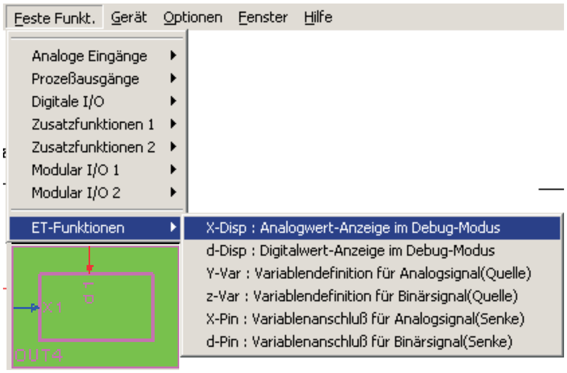
The actual internal values of the instrument/simulator can be observed in the engineering tool.

☞ This is helpful mainly when locating engineering errors.

On menu item:
Fixed functions – ET Functions – X-Disp : Display of analog value for debug mode , internal ET display items (analog/digital) can be selected.

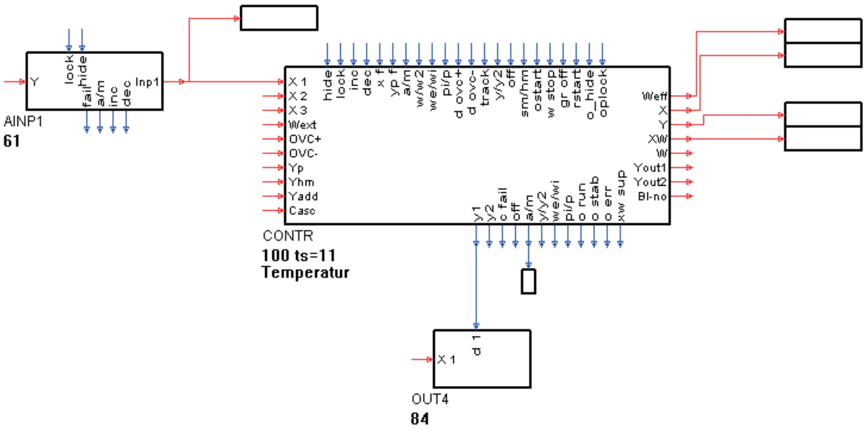
For these items, positioning and soft-wiring are done as for function blocks.
F4 (Options – Debug) can be used to activate the value display (see Fig.: Debug information).

Fig. 105


Feste Funkt.
Gerät
Optionen
Fenster
Hilfe
Analoge Eingänge
Prozeßausgänge
Digitale I/O
Zusatzfunktionen 1
Zusatzfunktionen 2
Modular I/O 1
Modular I/O 2
ET-Funktionen
X-Disp : Analogwert-Anzeige im Debug-Modus
d-Disp : Digitalwert-Anzeige im Debug-Modus
Y-Var : Variablendefinition für Analogsignal(Quelle)
z-Var : Variablendefinition für Binärsignal(Quelle)
X-Pin : Variablenanschluß für Analogsignal(Senke)
d-Pin : Variablenanschluß für Binärsignal(Senke)
X 1
d 1
OUT4

.

Fig. 107: Debug Informationen



II-6 Tips and tricks

II-6.1 Function keys

F1 **Calling up the help ...**

- General descriptions of the ET/KS 98 operating principle
- Survey and description of the library functions (with the function block selected or the parameter dialog box opened). Prerequisite: The checkbox for help must have been clicked during installation.

F2 **Calling the password dialog**

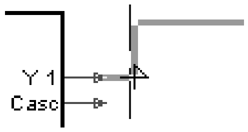
F3 **Comparison of the engineerings in KS 98-1 and ET is started.**

F4 **Debug mode is activated**

KS 98-1 or SIM/KS 98-1 must be connected!

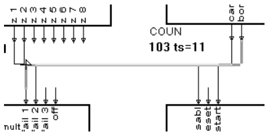
F5 **Search line overlapping**

During the wiring mode, line overlappings are searched. The first overlapping is displayed (shown marked). In the upper left corner of the screen, either Acount=0" (result negative) or Acount=1" is displayed.



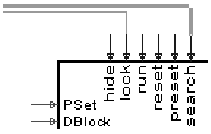
F6 **Search all line overlappings**

The overall engineering is searched for operlappings. Overlappings found are displayed shortly on the screen, but only the last overlapping found is displayed continuously.



F7 **Unify adjacent lines**

Lines which belong to one connection, with only a few parallel pixels, can be unified with F7. Thereby, a line segment must be selected. (Shifting using the mouse may be not pixel-exact, however, exact superposition is also possible by means of the arrow keys.)



F9 **Line colour / -type of logic connections**

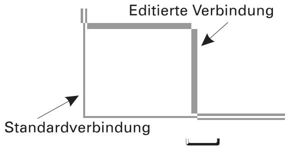
On the screen, distinction of analog and logic connections is easier in colour. In a print-out (black and white), distinction by a dashed line is better for clarity. Switch-over is always possible by pressing F9.

F10 **Cursor toggling between engineering ↔ menu bar.**

Menu operation with the arrow keys is now possible (if e.g. the mouse is not available). Function corresponds to key Alt .

F11 **Create standard connection**

Connections of two dots are drawn automatically as short as possible at right angles (standard connection) and can be edited manually. F11 transforms a selected edited connection into a standard connection.



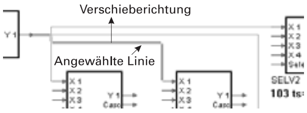
F12 **Language selection**

Language selection (German/English) for the engineering tool operator interface (menus, dialog boxes, etc.) is possible during working in the main menu. However, the language of help texts for KS 98-1 functions can be selected only during installation!

Shift + ...

... Unify line segments

When shifting a line over other lines which belong to the same signal source with the Shift key kept down, these lines are dragged and superposed automatically after releasing the shifted line (start segments cannot be shifted!). Closely adjacent lines are superposed automatically by clicking on a connection with the shift key kept down.



Ctrl + click (Wiring mode)

Multiple connections

A signal source can be connected with several inputs by marking an already existing connection and clicking on further inputs with the Strg key kept down. Thereby, the screen section can be shifted to the relevant position using the scroll keys, if the input to be connected is out of the visible area.

Ctrl + click (Survey mode)

A mouse click at any place on the screen with key ' pressed, will draw a grid that represents the print pages. In the print-out, the pages are numbered linewisely from left to right and from top to bottom and can be printed out with or without frame header.

Ctrl + Z

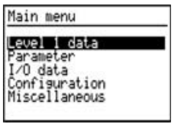
Undo:

Cancels the last action..

Ctrl + G

Producing a black and white copy

The display window of the KS 98-1 is copied into the clipboard and can be imported then to layout- or graphic-programs e.g. for documentation.



Main menu
Level 1 data
Parameter
I/O data
Configuration
Miscellaneous

Alt +Druck copies the complete simulation into the clipboard (windows function)

II-6.2 Mouse key functions

	Left mouse key	Right mouse key
Edit mode	Double-click on a free space: → Switches into the wiring mode	Click on a free space; → Inserts the previously selected function block.
	Click on a function block: → Selects (marks) the function block	Click on a function block: → Opens the parameter dialog of the block..
	Key held down: → Enables the function block to be moved.	
Wiring mode	Double-click on a free space: → Switches into the edit mode.	Click: → Switches into the survey display.
	Click on a line: →Selects (marks) the line	
	Key held down: Enables the line to be moved.	
Survey		Click: → Switches into the wiring mode. The position of the mouse pointer determines which part of the engineering is placed in the center of the wiring display.

II-6.3 Tips and tricks

- Search
  Entry of a block number (displayed in the upper left of the screen) and acknowledgement with Enter displaces the screen and displays the searched function block with marking (functions also in the survey display).

- Parameter setting
  Double click in parameter input field selects the current value for entry. Triple click on P input field opens selection dialog box (functions only once per parameter !) Setting cursor into value field with acknowledgement by pressing any key opens the selection dialog box (functions always !

- Line segments
  6 other line segments can be inserted into the last line segment (before the destination input), if the connection is selected. For this purpose, seize the last line segment before the input and draw it into the required direction using the mouse pointer in wiring mode.

- Calculation sequence
  The survey display can be used to show the calculation sequence.The sequence is indicated by consecutive marking of the function blocks.Start/stop of the marking procedure by means of the key “t”. While the procedure is running, pressing key “r” stops the marking and moves back to the previous function block. Pressing key “v” moves to the next function block.

- ↓ ↑ → ←
  Pixel-exact shifting of selected line (segments) and function blocks

- COM test
  Quick communication testing is possible by transmission of an "Empty" engineering to KS 98-1.
  Testing the communication can also be done by calling up the password dialogue with F2 without overwriting the existing engineering. The password dialogue can be ended with ESC.

- Copying parameters
  If a function block is selected during the edit mode, its parameters can be stored in the clipboard by pressing Ctrl + ...“C”. If another function block of the same type is then selected, the stored parameters can be copied into the block by pressing Ctrl + ... “V”. This is a particularly useful feature for all blocks with many parameters (e.g. CONTR; APROGD; ...). The function can also be used to transfer data into other engineerings, provided the same operating version is involved.

- Copying areas of the engineering
  If an area has been selected in the survey display by means of the capture frame, its entire contents can be stored in the clipboard by pressing Ctrl “C”. Pressing Ctrl “V” copies the stored data into the engineering, where it can be positioned with the mouse pointer. Parameters and internal connecting lines are maintained. External connections are deleted.

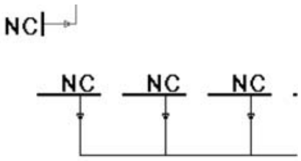
- Moving areas of the engineering
  If an area has been selected in the survey display by means of the capture frame, its entire contents can be moved with the mouse, if the Shift key is held down. Parameters and internal connecting lines are redrawn automatically.

- Abort
  Time-consuming functions such as “Compare” (F3) can be aborted with the “x” key.

- Block selection
  Insiders enter the short-form name of the required function (e.g. ADSU) followed by acknowledgement to save the detour via the menu bar. Click with the right mouse key to position the function thus selected directly (note upper/lower case!). If the required function is quite close, selecting and de-selecting it is sufficient to produce the same effect.

- Aligning blocks
  Selected function blocks of the same type can be aligned in the "survey"
  Key ■ → up!
  Key ■ → left!

- Searching for not connected "sinks" ■ ■ ■
  Placed, but not defined signal sinks make an engineering more complex. With the ■-key an internal list of all sinks not connected is set up. With repeated pressing of ■ or ■ they can be found and either be defined or deleted.

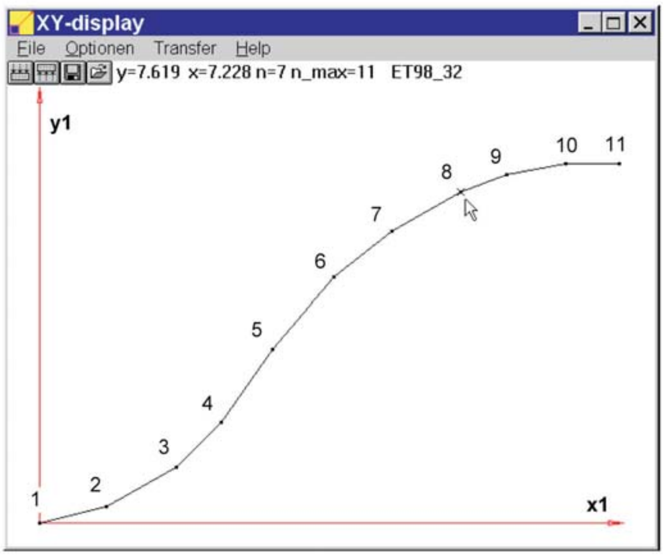


Graphic display of XY profiles

With programmers (APROG, DPROG) and characterizers (CHAR) it is very helpful to view a XY-plot and modify the adjusted parameters in a graphical way.

Button G calls the program „XY-display" and shows the actual characteristic (over the time „t" or the input „x1") of the selected function block.

- ***Characterizer CHAR***
  Es wird nur der Inhalt eines Funktionsblockes dargestellt! Grundsätzlich sind 11 Stützpunkte vorhanden. Nicht benötigte (gelöschte) Stützpunkte (→ Editierfunktionen) werden auf die Parameter des letzten Stützpunktes gesetzt.



y=7.619 x=7.228 n=7 n_max=11 ET98_32

Fig.1: Characterizer CHAR

- ***Analog programmer***
  Up to 10 data blocks (APROGD; < 100 segments) of a programmer are displayed in a window. Quantity and sequence of the blocks are defined by the wiring of the Engineering. The displayed profile starts with the 1st segment; the initial value Wp0 is not included in the display ! With changes of time values interval times Tp are calculated and transfered.

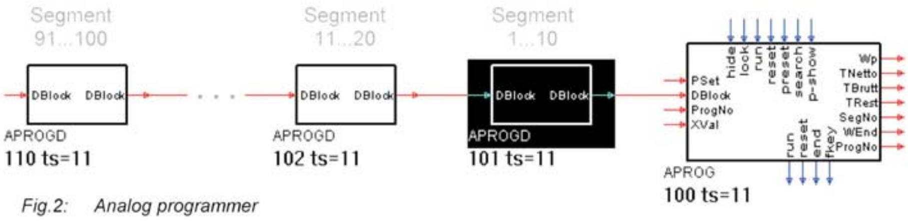
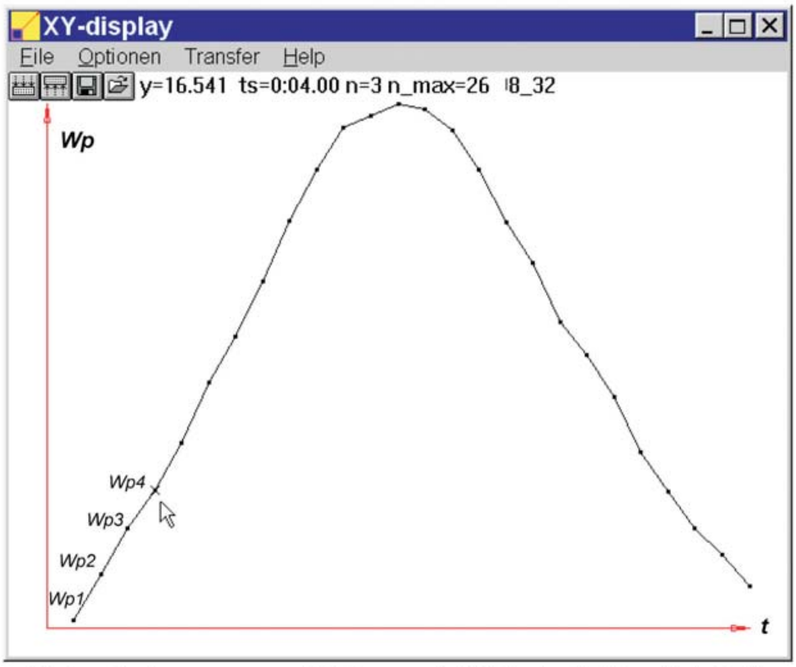


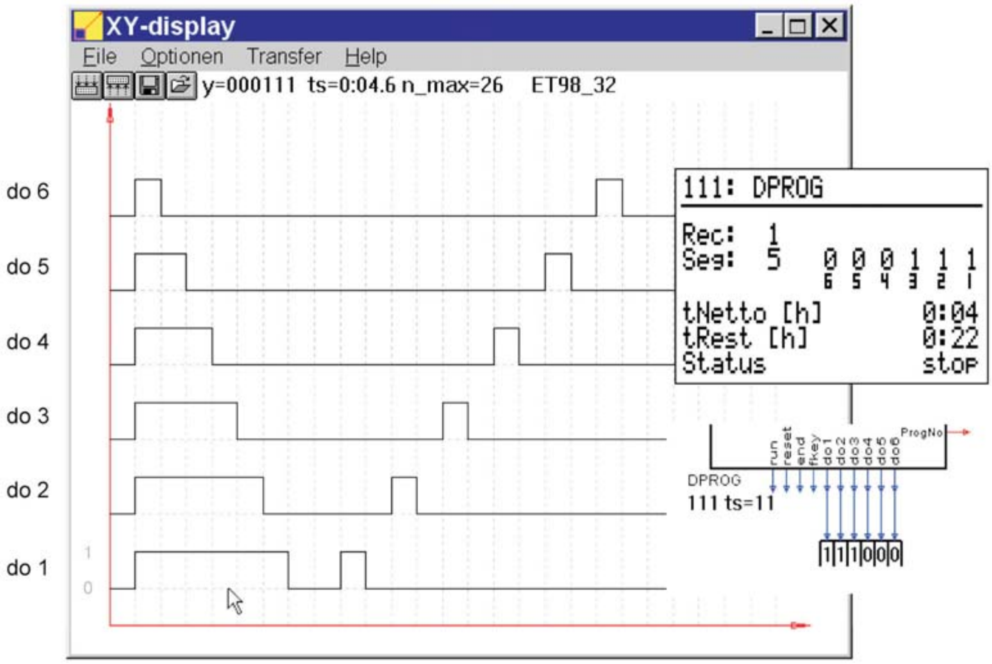
Fig.2: Analog programmer



```
y=16.541  ts=0:04.00 n=3 n_max=26  I8_32
```

Fig.3: *Analog programmer with 26 segments; initial value Wp0 is not included !*

- ***Digital programmer***
  The display of the program values starts with the 1st segment. the initial value D0 is not included ! With changes of time values specific interval times Tp are calculated and transfered.



```
y=000111  ts=0:04.6 n_max=26    ET98_32
```

```
111: DPROG

Rec:  1
Seg:  5    0 0 0 1 1 1
           6 5 4 3 2 1
tNetto [h]        0:04
tRest [h]         0:22
Status            stop
```

Fig.4: *Digital with 26 segments; initial value D0 is not included !*

Edit functions

Shift + Click Adds a „xy-point“ (analog) repectively a segment (digital)

Double click on point Selects a xy-point (analog only; marked with „x“).

Delete Deletes a selected xy-point (analog) or a segment (digital)

Left mouse key ... shifts a selected xy-point (analog) or a segment (digital).

Analog curve: With shifting a selected point in direction of the following one the rest of the curve is pushed in the same direction. Movement in direction of a foregoing point pulls back the rest of the curve.

Digital curve: Click on the actual value and shift in the requested direction produces a change of state.

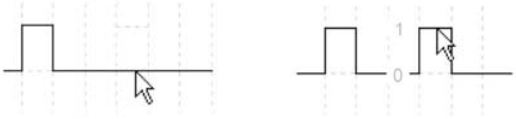
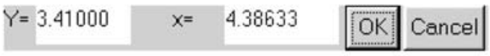


Fig.5: Change of digital values of a segment

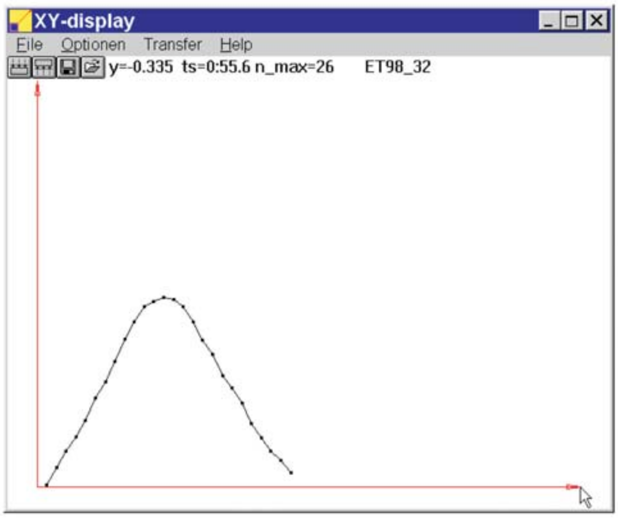
Right mouse key ... click on a selected xy-point enters a window for numerical definition of the values.



Y= 3.41000 x= 4.38633

Fig.6: Numerical input of co-ordinates

Click on co-ordinate arrows ... reduces the scale of the according axis stepwise (“Undo” with auto-scaling; see Menu functions)



y=-0.335 ts=0:55.6 n_max=26 ET98_32

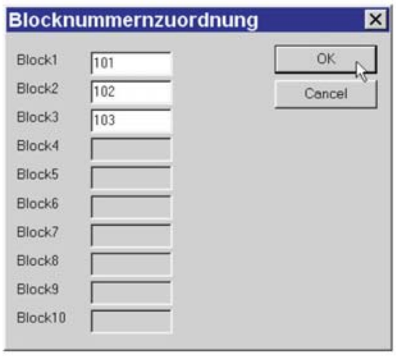
Menu functions

<File>	<OPEN>	Opens a file	
	<SAVE>	Saves a file	
	<SAVEAS>	Saves a file with file name	
	<OK>	Transfer of the parameters to the related function blocks of the Engineering. XY-display will be closed automatically.	
	<Quit>	Closes the program without changes !	
<Options>	*<Communication>*	Serial interface (SIM/COM-Port, baud rate, address). These adjustments must be re-entered when opening the program!	
	<End of range>	OK	*Confirms changes.*
		Auto	*Automatic scaling of x/ t- and y-axis within the window*
		tMax =	*Length of the x/ t- axis (unit of time is defined in APROG resp. DPROG)*
		yMax =	*Length of the y- axis*
		Anzahl =	*Number of xy-points* ▪ *Characterizer: 11 (fixed !)* ▪ *Programmer: ≤ 100 (in max. 10 data blocks)*
		deltaT=	*Grid of the x/ t-axis*
		fx =	*Factor of x/ t-values. After confirming with OK all x/ t-values will be multiplied with this factor !*
		fy =	*Factor of y-values. After confirming with OK all y-values will be multiplied with this factor !*
		x Plus minus	*Permits negative x-values (CHAR)*
		y Plus minus	*Permits negative y-values (APROGD, CHAR)*

<Blocknummern> *<Block numbers>: Displays the engaged block numbers out of the Engineering.*

The block numbers can be altered. Thus, the according parameters can be transferred to an other equivalent function block or a group of blocks (programmer). By this way it is e.g. also possible to transfer interval times Tp from an anaolog programmer into a digital programmer, but the digital values (states) will be changed as well!

Caution: A validity check will not be carried out !



Blocknummernzuordnung

Block1	101
Block2	102
Block3	103
Block4	
Block5	
Block6	
Block7	
Block8	
Block9	
Block10	

:**Transfer>** *<To KS98>* *Transfers the parameters to that KS 98 (resp. SIM/KS98), which is adressed under <Optionen><Kommunikation>.*

or

<From KS98> *Reads the parameters from that KS 98 (resp. SIM/KS98), which is adressed under <Optionen><Kommunikation>.*

or

III Function blocks:

The KS98-1 function library contains all functions which are normally used for plant operation. These include:

- Functions for calculation of mathematic formulas from simple addition to exponent function
- Logic functions and functions for realization of control sequences
- Numerous selection and storage functions are helpful for signal processing.
- Alarm and limit value functions are indispensable for plant safety.
- Interface functions facilitate communication with adjacent and supervisory systems.
- Functions for implementation of complex and flexible control and program sequences as well as profile control tasks meet the most exacting requirements.

The wiring principle of composed functions such as programmer, controller cascades and stepping control are explained in the relevant basic function descriptions in this manual.

Examples for the basic engineering as mentioned in this manual, and further application examples for various requirements are included on a CD as a collection of examples with detailed description, or available on request.

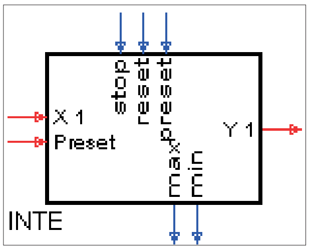
General KS 98-1 function block features

The features of multi-function unit KS 98-1 are determined by purposeful connection of standard function blocks with adjustable parameters.

In the KS98-1 engineering, a function block represents a black box with analog inputs (from left), analog outputs (to right), digital control inputs (from top) and control or status outputs (to bottom), as shown in the integrator diagram.

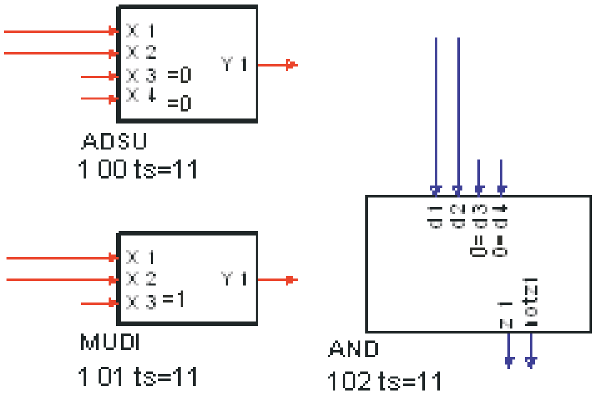
The following abbreviations are used for general inputs which mean process values and outputs which mean function results:

analog inputs:	X1, X2, ...
analog outputs:	Y1, Y2, ...
digital inputs:	d1, d2, . . .
digital outputs:	z1, z2, ...



The abbreviations for inputs and outputs with special signification are derived from their function.

Not all inputs and outputs of a function block need to be connected. The following rule is applicable: open inputs are without effect. Examples: totalizer, multiplier, AND gate. In some cases, the connection of an input has an additional effect, if e.g. priority handling is concerned (programmer control inputs).



Function blocks are numbered by the engineering tool in the order of occurrence from 100 to max. 450 as standard. Calculation of function blocks in the unit is dependent on this order. By changing the block number, the handling order is adapted. Function blocks which can be used only once or with reference to the hardware (inputs/outputs) are always within a numeric range of 0-100.

Function blocks are preset to a scanning rate (computing cycle) of 100 ms. The computing cycle can be increased to 800 in steps of 200, 400, whereby the processor load is reduced. Detailed information is given in the ET98 operating manual.

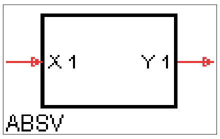
The parameters of each function block are adjustable. Apart from an individual description for documentation purposes, the majority of blocks is provided with function-specific parameters. In addition to very special ones, some parameters are quite frequent. For these general values, the same identifiers are always used:

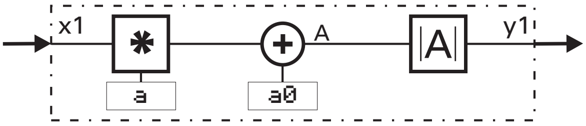
a, b, c, d	factors without special signification
a0, b0, ... x0, y0	appended 0 as a symbol for an offset x0 = offset of an input, y0 = offset of an output
T, Ti	times in seconds (delays, pulse or pause duration)
Mode	These parameters are used to select function parameter setting by means of the described parameter or by means of an analog input (dynamic parameter setting).

Digital control inputs for binary selection (e.g. SELV1 for selection of 4 analog values) are normally numbered from left to right d1, d2. Note that d2 is the less significant bit despite numbers running in opposed direction. In all cases where the bit order also expresses an order of values, please, refer to the documentation of the special block in the following chapters.

III-1 Scaling and calculating functions

III-1.1 ABSV (absolute value (No. 01))





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

The absolute value of a number is it's number without polarity sign. This is the best solution for scaling a value that can't become negative, in reference to calculating time. This function block should be used, when scaling must not use a lot of calculating time.

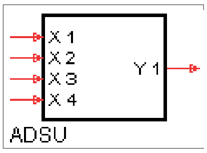
Input variable x1 is multiplied by factor a (parameter). Now, constant a0 is added. The absolute value of the resulting value is formed and output at y1.

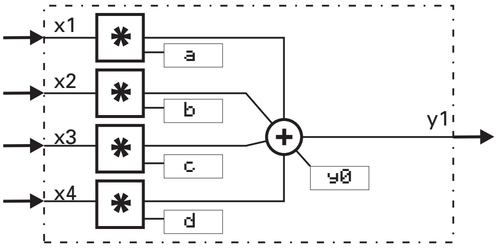
Example:

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

III-1.2 ADSU (addition/subtraction (No. 03))



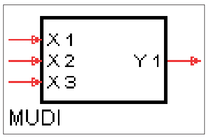


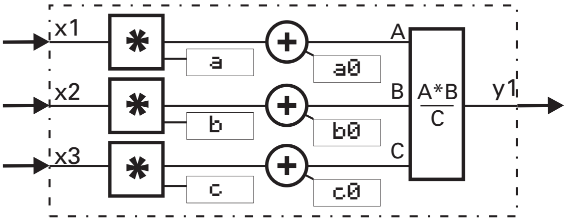
$$y_1 = a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 + y_0$$

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

III-1.3 MUDI (Multiplication / division (No. 05))





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

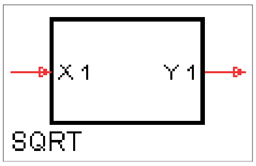
Input variables x1 ...x3 are multiplied by factors a, b, c. The relevant constants a0, b0, c0 are added. The output variable corresponds to the product.

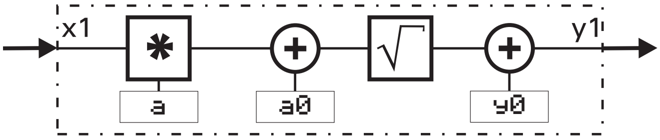
Value “1" is assigned automatically to unused inputs.

With divisions by “0" (C = c •x3 + c0 = 0) output y1 is set to 1.5 • 10^{37}.

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

III-1.4 SQRT (square root function (No. 08))





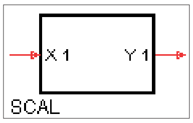
$$y_1 = \sqrt{a \cdot x_1 + a_0} \quad + y_0$$

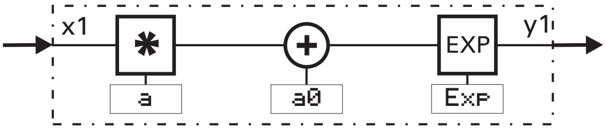
Constant a0 is added to input variable x1 multiplied by a. The result is subjected to square root extraction. Constant y0 is added to the result of square root extraction.

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

III-1.5 SCAL (scaling (No. 09))





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

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

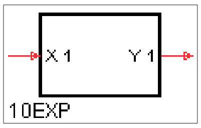
If x1 is not used, this is interpreted as x1 =0. With Exp = 0 SCAL outputs 1.

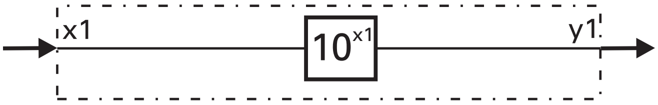
Parameter	Description	Range	Default
a	Multiplication factor	-29 999...999 999	1
a0	Offset	-29 999...999 999	0
Exp	Exponent	-7...7	1

Example: $y_1 = \sqrt[3]{x_1^2} = x_1^{\frac{2}{3}} = x_1^{0,\overline{6}}$

This function block should be used only, if the exponential function is needed. Factor a and the offset a0 are also available with functions that need less calculating time (e.g. ADSU, MUDI, ABSV).

III-1.6 10EXP (10s exponent (No. 10))





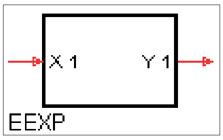
$$y_1 = 10^{x1}$$

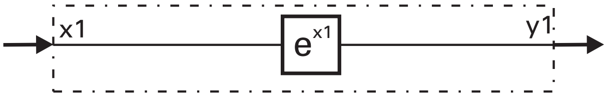
Input value x1 is calculated according to formula $y_1 = 10^{x1}$ and output at the y1 output.
An unwired x1 is interpreted as x1 = 0 (in this case y1 is 1).

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.

III-1.7 EEXP (e-function (No. 11))





$$y_1 = e^{x1}$$

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.

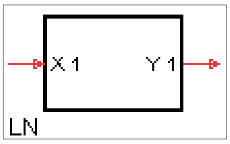
If **x1** is not wired, this is interpreted as **x1** = 0 and thus as **y1** = 1.

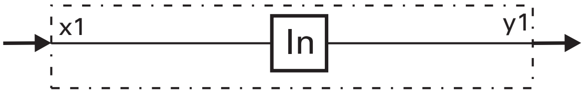
Note:
EEXP is the reversal function of function LN.

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.

III-1.8 LN (natural logarithm (No. 12))





$$y_1 = ln(x1)$$

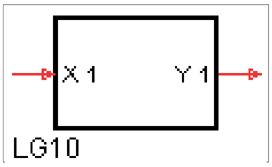
The natural logarithm of input variable **x1** is formed.
The basis of natural logarithms is constant e (2,71828182845904).
If **x1** is not wired, this is interpreted as **x1** = 1. In this case **y1** is 0.

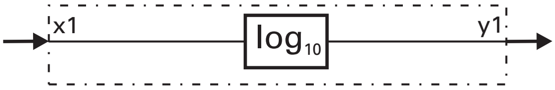
With a negative input variable **x1**, **y1** = -1,5 $_{w}$ 10^{37} is set.

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.

III-1.9 LG10 (10s logarithm (No. 13))





$$y_1 = log(x1)$$

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.

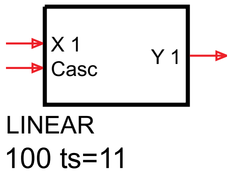
Note:
LG10 is the reversal function of function 10EXP.

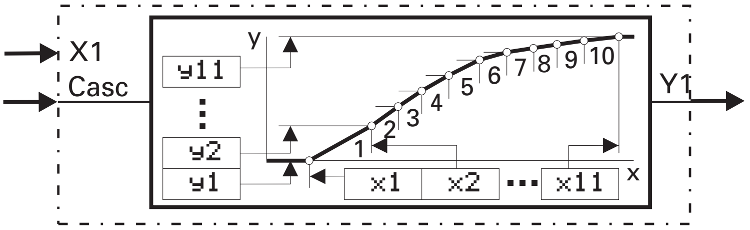
Examples:

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.

III-2 Non-linear functions

III-2.1 LINEAR (linearization function (No. 07))





Block LINEAR can be used for calculation of y = f (x).

Max. 11 adjustable segment points for simulation or linearization of non-linear functions are available. Each segment point comprises input x(1) and output y(1).

The segment points are connected automatically by straight lines. Each input value x1 has a defined output value y1. With an input value x1 smaller than parameter x(1), the output value is equal to the y(1) value. If input value x1 exceeds the highest parameter x(n) value, the output value is equal to the relevant y(n) value.

The condition for input of configuration parameters is that values are input in ascendant order (x(1) < x(2) <...< x(11)). The end of value pairs is marked by the “OFF” value in the next input value x(n+1).

This function block is cascadable. It has 2 inputs: The 1st input provides the variable which must be linearized. The 2nd input (case) is used to connect the previous linear block.

Inputs/outputs

Analog inputs	
X	Input variable which must be linearized
Casc	Cascade input

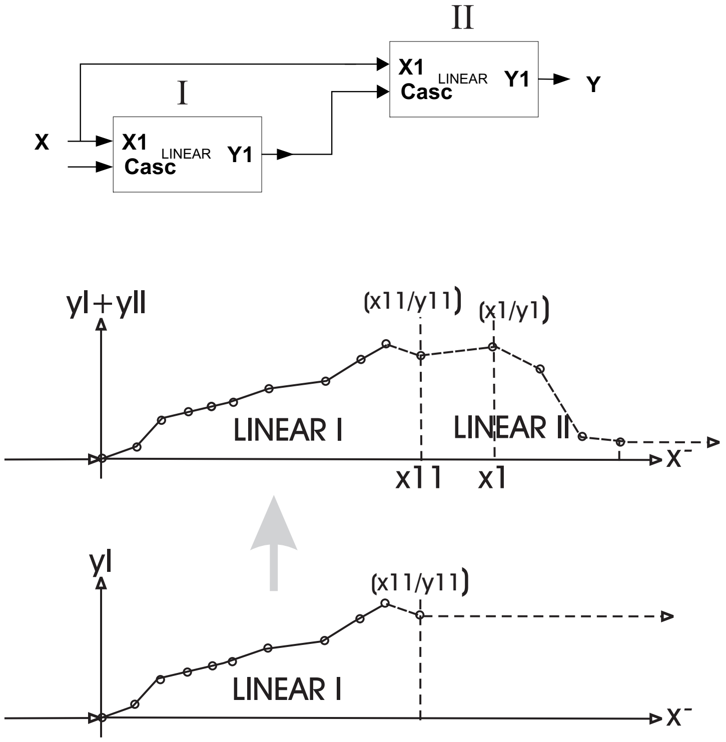
Analog outputs	
Y	Linearization result

Parameter

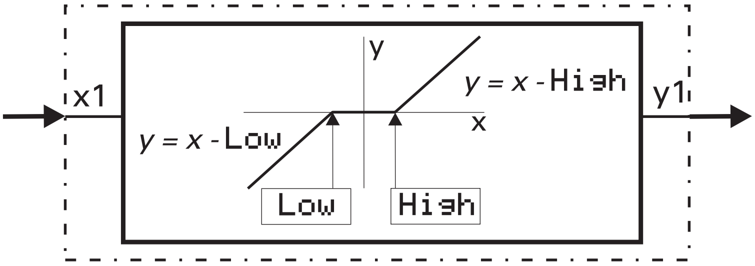
Parameter	Description	Range	Default
x(1) ... x(11)	Input value for segment point 1...11	-29999...999999, OFF x(1)<x(2)<.....x(11)	x(1) = 0, x(2) = 1, x(3) = 2, ..., x(11) = 10
y(1) ... y(11)	Output value for segment point 1...11	-29999 ... 999 999	y(1) = 0, y(2) = 1, y(3) = 2, ..., y(11) = 10

Example: linear as a cascade

Fig. 108



III-2.2 GAP (dead band (No. 20))



$y_1 = x1 - Low$ bei x1 < Low

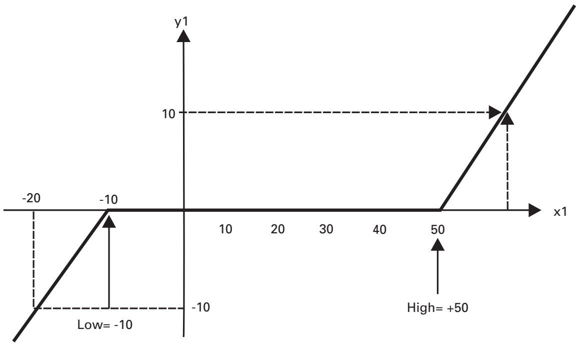
$y_1 = 0$ bei x1 = Low...High

$y_1 = x1 - High$ bei x1 > High

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.

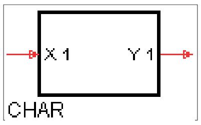
Example:
In the following example, -10 for Low and 50 for High was used.

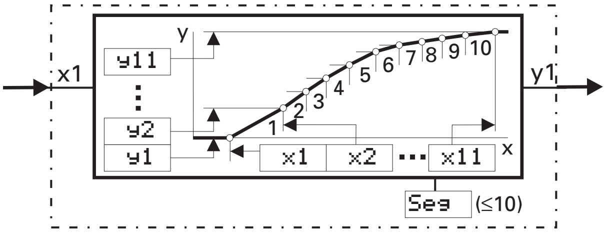
Fig. 109



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

III-2.3 CHAR (function generator (No. 21))





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)).

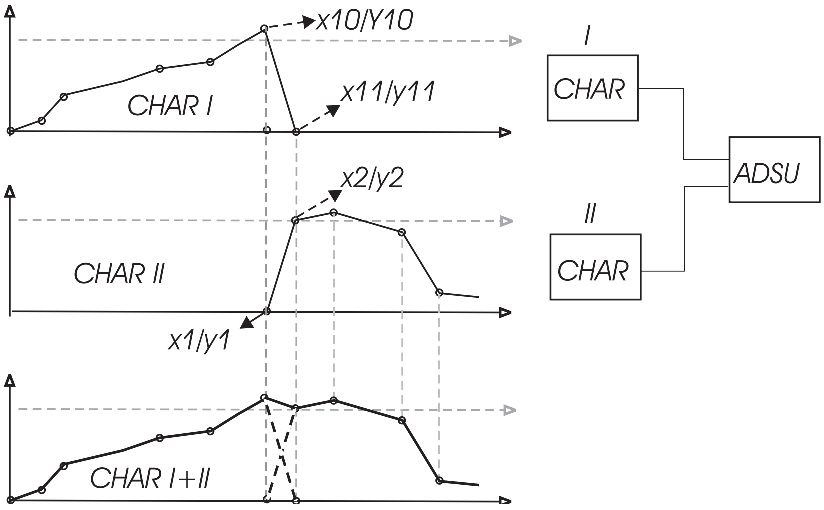
Configuration	Description	Range	Default
Seg	Number of segments	1...10	2
x(1)...(11)	Input value for curve point	-29 999...999 999	0...10*
y(1)..(11)	Output value for curve point	-29 999...999 999	0...10*

* 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:

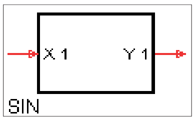
whereby x10 of CHAR I = x1 of CHAR II and x11 of CHAR I = x2 of CHAR II

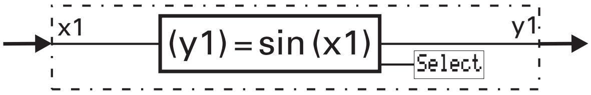
Fig. 110



III-3 Trigonometric functions

III-3.1 SIN (sinus function (No. 80))





$$y_1 = sin(x1)$$

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° / ± ó/$_2$)). Internal limiting is not provided. If input value x1 is out of the range in which the sinus function can still provide purpose-ful values, output y1 is set to 1,5 • 10^{37}.

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

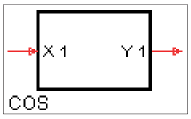
Example radian:
y1 = sin(x1) x1 = 90rad ≙ y1 = 0,89399666

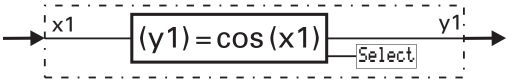
Parameter	Description	Controller display
Select	Unit: degree of angle (default)	Ang. deg
	Unit: radian	Radian

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°)!

III-3.2 COS (cosinus function (No. 81))





$$y_1 = cos(x1)$$

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 • 10^{37}.

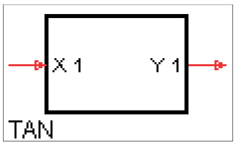
Example degree of angle:
y1 = cos(x1) x1 = 60° ≙ y1 = 0,5

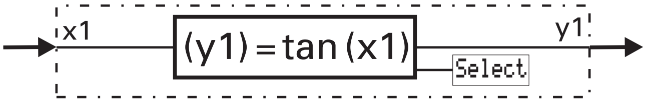
Example radian:
y1 = cos(x1) x1 = 45rad ≙ y1 = 0,525321988

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

Important: When controlling with the pocket calculator see → page 100

III-3.3 TAN (tangent function (No. 82))





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

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

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 $-{}^{\pi}/_{2}$ ... ${}^{\pi}/_{2}$). If input value x1 is out of this range, output y1 -1, • 10^{37} (x1<-90 [$-{}^{ó}/_{2}$]) or 1,5 • 10^{37} (x1>90 [${}^{\pi}/_{2}$]) is set.

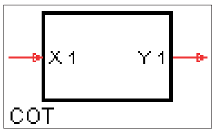
Example degree of angle:
y1 = tan(x1) x1 = 60° ≙ y1 = 1,73205

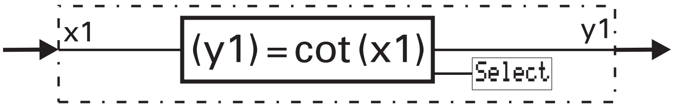
Example radian:
y1 = tan(x1) x1 = 1,53rad ≙ y1 = 24,498

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

Important: When controlling with the pocket calculator see → page 100

III-3.4 COT (cotangent function (No. 83))





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

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 • 10^{37} (x1 <0) or 1.5 • 10^{37} (x1 >180 [x1 >π]).

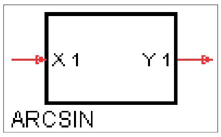
Example degree of angle:
y1 = tan(x1) x1 = 45° ≙ y1 = 1

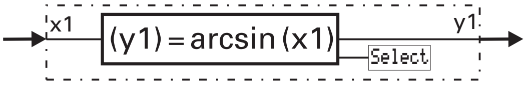
Example radian:
y1 = tan(x1) x1 = 0,1rad ≙ y1 = 9,967

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

Important: When controlling with the pocket calculator see → page 100

III-3.5 ARCSIN (arcus sinus function (No. 84))





$$y_1 = arcsin(x1)$$

$$\text{valid for } x_1: -1 \leq x_1 \leq +1$$

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.

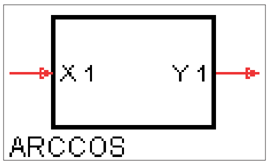
The calculation is output as degree of angle [-90° ... 90°] or as radian [-ó/2 ... ó/2]. With arguments out of the function validity range, output y1 is limited to -1,5 • 10^{37} (x1 <-1) or 1,5 • 10^{37} (x1 >1).

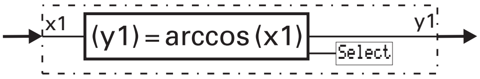
Example degree of angle:
y1 = arcsin(x1) x1 = 0,5° ≙ y1 = 30

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

III-3.6 ARCCOS (arcus cosinus function (No. 85))





$$y_1 = arccos(x1)$$

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.

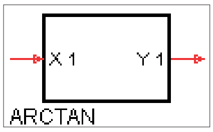
Calculation is either as degree of angle [0° ... 180°] or as radian [0...π]. With arguments out of the function validity range, output y1 is set to 1,5 • 10^{37} (x1 <-1) or -1,5 • 10^{37} (x1 >1).

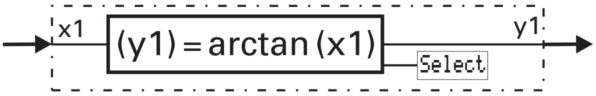
Example degree of angle:
y1 = arccos(x1) x1 = 0,5° ≙ y1 = 60

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

III-3.7 ARCTAN (arcus tangent function (No. 86))





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

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.

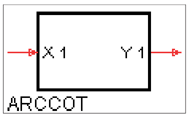
The calculation is output either as degree of angle [-90° ... 90°] or as radian [-$\Pi/_2$ ... $\Pi/_2$].

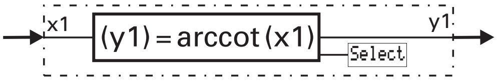
Example degree of angle:
y1 = arctan(x1) x1 = 1° ≙ y1 = 45

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

III-3.8 ARCCOT (arcus cotangent function (No. 87))





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

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].

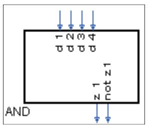
Example degree of angle:
y1 = arccot(x1) x1 = 45° ≙ y1 = 1,273

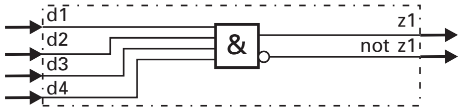
Example radian:
y1 = arccot(x1) x1 = -12rad ≙ y1 = 3,058

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

III-4 Logic functions

III-4.1 AND (AND gate (No. 60))



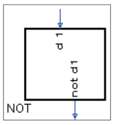


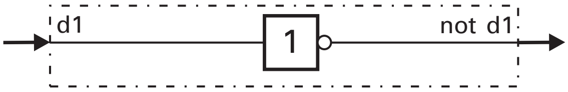
$$z_1 = d_1 \text{ AND } d_2 \text{ AND } d_3 \text{ AND } d_4$$

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	1	0

III-4.2 NOT (inverter (No. 61))





$$z_1 = \overline{d_1}$$

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

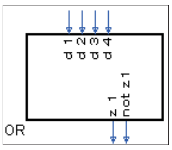
Not behaves different, dependent from

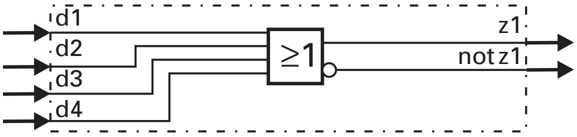
-Download e.g. POWER ON (RAM-Buffer empty)

-POWER ON (RAM-Buffer o.k.)

for example	initialization	first calculation
Download or online → offline	z1 = 0	z1 = 1
POWER ON and RAM o.k.	z1 = 1	z1 = 1

III-4.3 OR (OR gate (No. 62))



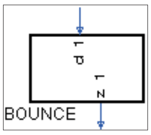


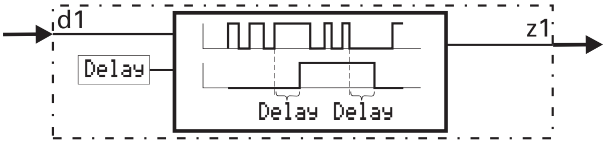
$$z_1 = d_1 \ \text{OR} \ d_2 \ \text{OR} \ d_3 \ \text{OR} \ d_4$$

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	1	0

III-4.4 BOUNCE (debouncer (No. 63))





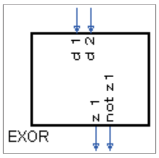
This function is used for de-bouncing a logic signal. The change of input signal **d1** is transferred to output **z1** only, when it remained constant for the time adjusted in parameter Delay. The time-out accuracy is dependent of the sampling interval assigned to the function.

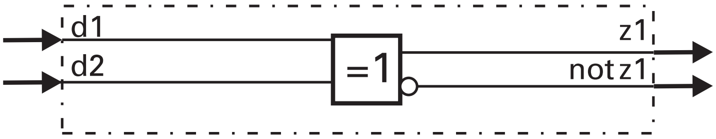
Example:

– 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

III-4.5 EXOR (exclusive OR gate (No. 64))





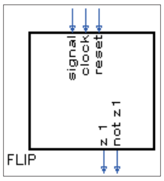
$$z_1 = d_1 \text{ EXOR } d_2$$

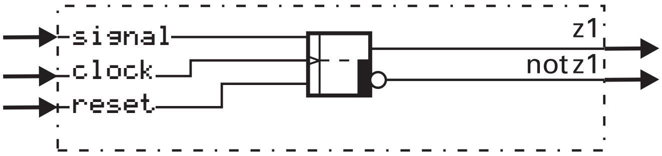
Logic inputs **d1** and **d2** are combined into **z1** according to the truth table given below. Unused inputs are interpreted as logic 0.

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

III-4.6 FLIP (D flipflop (No. 65))





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.

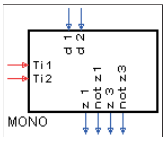
⚠ **This function has a “memory”. This means: after power-on, it continues operating with the statuses at z1 and not z1, which existed at power-off, provided that the RAM data are still unchanged.**

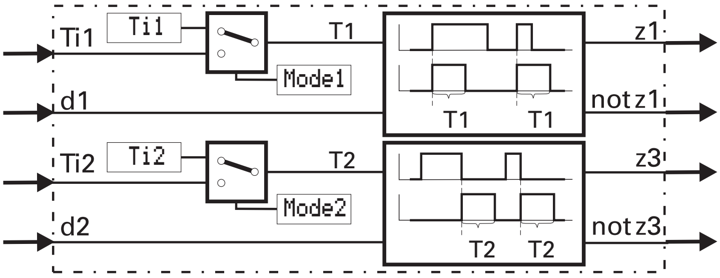
Inputs/outputs

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

III-4.7 MONO (monoflop (No. 66))





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

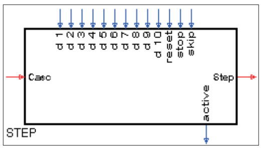
Analog inputs	
Ti1	Pulse duration Ti$_1$ [s] of the pulse generated by d1, when Mode 1 = Para.Ti1.
Ti2	Pulse duration Ti$_2$ [s] of the pulse generated by d2, when Mode 2 = Para.Ti2.

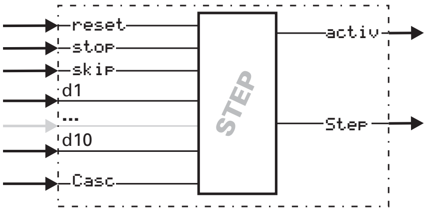
Digital outputs	
z1	Positive pulse of length Ti$_1$, when a positive flank at input d1 was detected.
not z1	Negative pulse of length Ti$_1$, when a positive flank at input d1 was detected.
z3	Positive pulse of length Ti$_2$, when a positve flank at input d2 was detected.
not z3	Negative pulse of length Ti$_2$, 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

III-4.8 STEP (step function for sequencing (No. 68))





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.

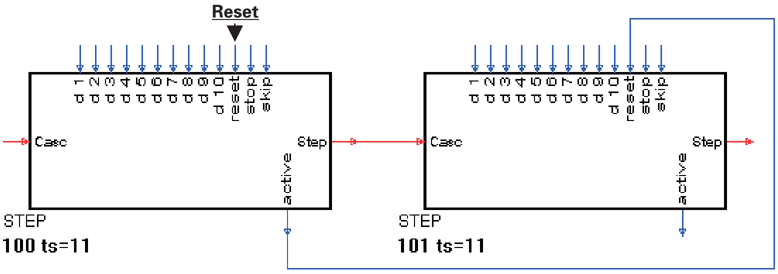
Example:
Switch-over from step 3 (Step = 3) to step 4 (Step = 4) is only after the condition at d3 was met (d3 = 1). The condition at d4 is checked only when calling up the function for the next time. Thus immediate switch-over is prevented. As long as d3 = 0, the value of output Step remains 3.

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.



For resetting the cascaded stepping control, reset wiring at the 1st function block is required.

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_1 = 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_i input into account.

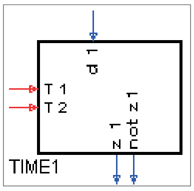
Analog input	
Casc	Used for STEP function cascading. At the first STEP function of a cascade

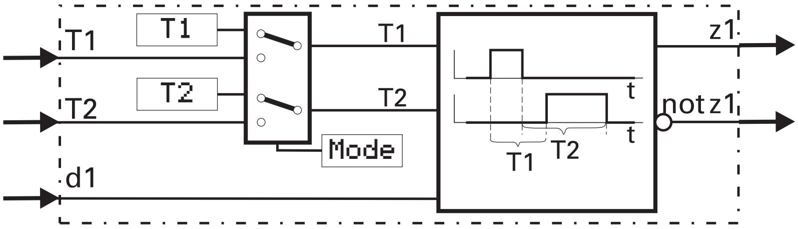
Digital output	
activ	activ =1 indicates that the STEP function is still in the active status or in reset. activ =0 indicates that the STEP function has elapsed.

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!

III-4.9 TIME1 (timer (No. 69))





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.

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	

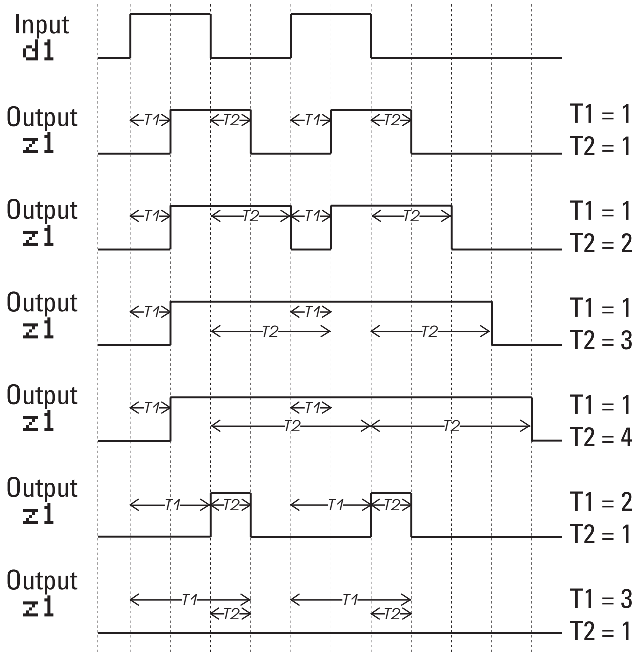
The pulse duration accuracy is dependent of the time group to which the function is assigned. It is an integer multiple of the sampling interval adjusted for this block (100, 200, 400, 800ms).

Example:

T1 = 0,7s with assignment to
- sample time 100ms means, delay time of the positive flank is 0,7s.
- sample time 200ms means, delay time of the positive flank is 0,8s.
- sample time 400ms means, delay time of the positive flank is 1,2s.
- sample time 800ms means, delay time of the positive flank is 1,6s.

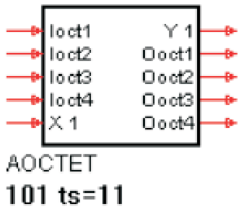
Example with different delay time T1 and T2

Fig. 111



III-5 Signal converters

III-5.1 AOCTET (data type conversion (No. 02))



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
Ioct1..4	analog output byte value 1

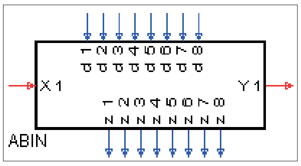
Parameters:

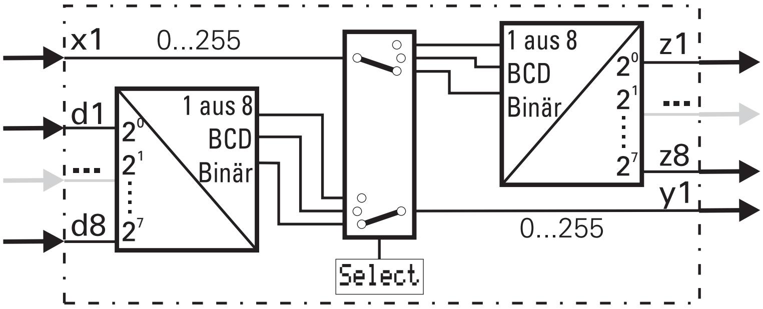
Parameter	Description	Value range	Default
Ioct	data type of analog > byte conversion	0...999 999 [s]	0
Ooct	data type of byte > analog conversion		

The following data types are available

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

III-5.2 ABIN (analog ↔ binary conversion (No. 71))





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	z1	z2	z3	z4	z5	z6	z7	z8
	2			2	2			2
x1 = 83	1	1	0	0	0	0	1	1
BCD	3				8			

Out of the range, the output allocation is:

Input	z1	z2	z3	z4	z5	z6	z7	z8
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	d1	d2	d3	d4	d5	d6	d7	d8
	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	0

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.

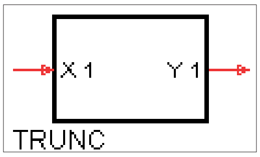
Digital outputs	
z1 ...z8	Converted binary value, BCD value or value selection.

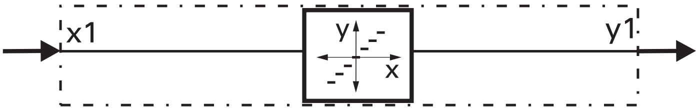
Analog output	
y1	Converted analog value.

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	

III-5.3 TRUNC (integer portion (No. 72))





$$y_1 = \text{INT}(x_1)$$

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

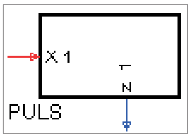
Inputs/outputs

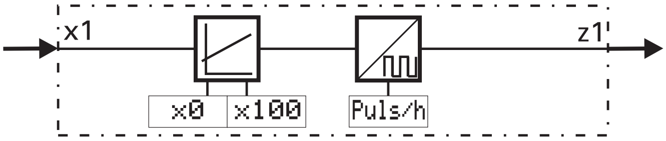
Analog input	
x1	Input variable to be handled

Analog output	
y1	Integer portion of x1

No parameters!

III-5.4 PULS (analog pulse conversion (No. 73))





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

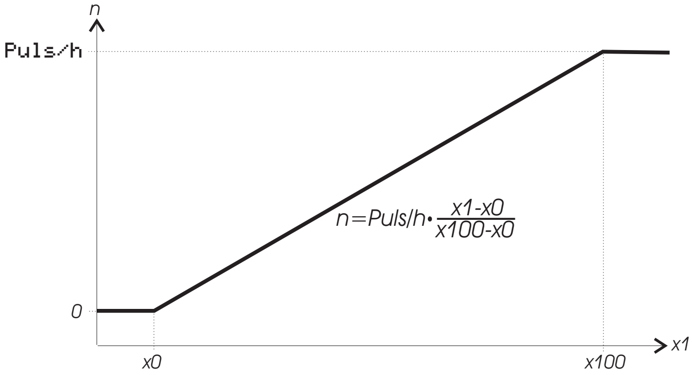
n = Number of pulses per hour
x0 = Parameter
x100 = Parameter
x1 = Analog input

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.

Fig. 1

Puls/h	=	max. Pulsenumberl/h
x0	=	0% of Pulse/h
x100	=	100% of Pulse/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 ≤ x0 no pulses are produced (area start, creeping flow suppression)

x100 = If the analog input is x1 ≤ x100 n remains = constant = Puls/h

Puls/h = Parameter. Pulse number/hour for analog input $x_1 = x_{100}$

Fig. 2

Example:

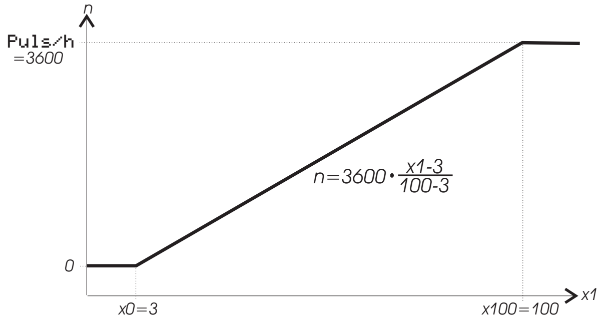
x1 = 3...100% ≙ 0...3600/h

$x_0 = 3$

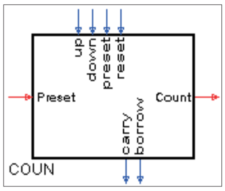
$x_{100} = 100$

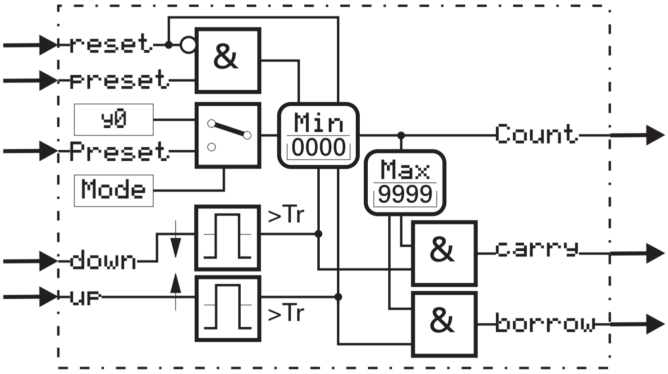
Puls/h = 3600

sampling period ≤ 400 ms



III-5.5 COUN (up/down counter (No. 74))

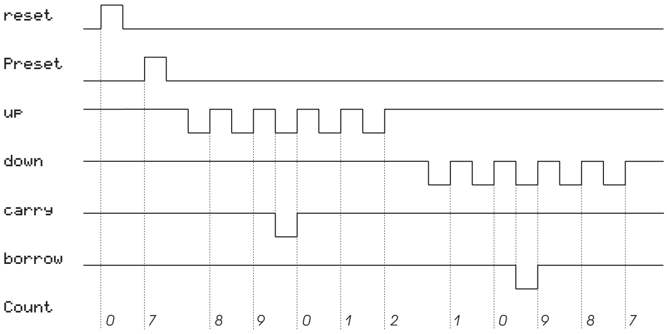




'COUN' is an up/down counter and counts the events at input up or down, which are available at the up or down input for at least the duration of the time group in which the function runs.

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

Fig. 3 Pulse diagram of the up/down counter:



"up,down, carry und borrow" sind in inaktivem Zustand 1.

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

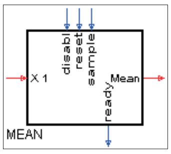
Digital outputs	
carry	Carry output (clock - up)
borrow	Borrow output (clock - down)

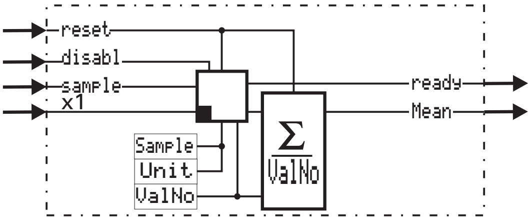
Analog output	
Count	Count output

Parameter:

Parameter	Description	Value range	Default
Mode	Source of preset-value	0: Para y0	←
		1: InpPreset	
y0	Preset-value	-29 999...999 999	0
Max	Max. limit	-29 999...999 999	1
Min	Min. limt	-29 999...999 999	0

III-5.6 MEAN (mean value formation (No. 75))





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.

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

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=	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

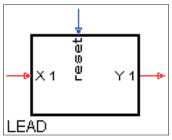
Analog output	
Mean	Calculated mean value

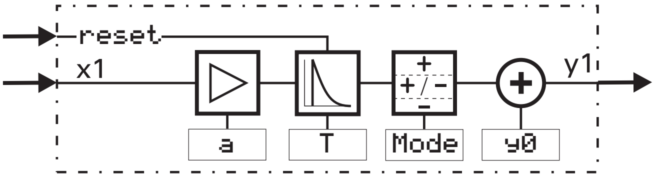
Configuration:

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

III-6 Time functions

III-6.1 LEAD (differentiator (No. 50))





The differentiator forms the difference quotient according to equation:

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

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	

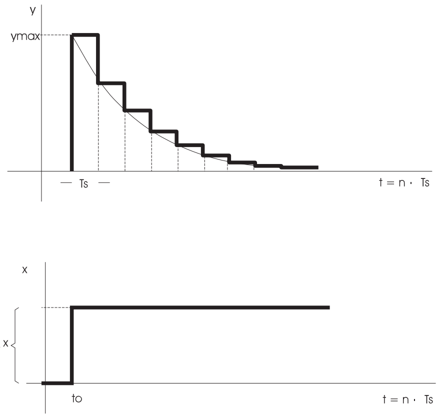
Step response:

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

Fig. 4



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

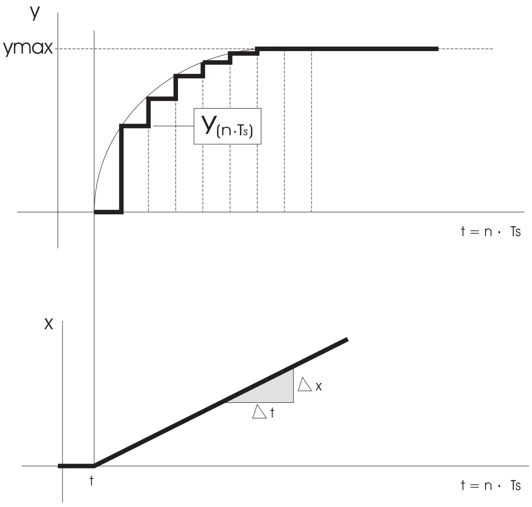
Ramp response:

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

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

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

Fig. 5

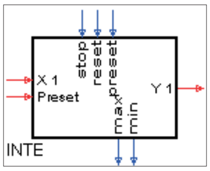


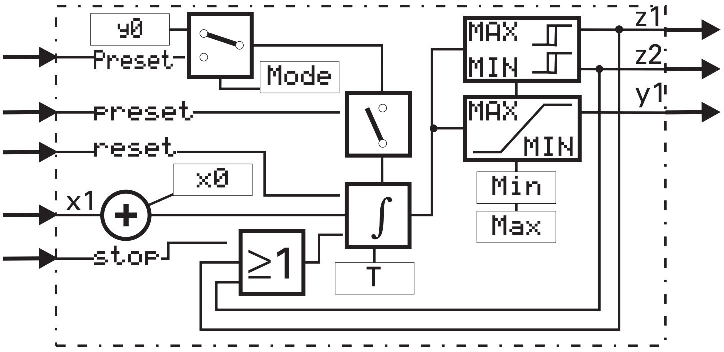
Thereby, $m = m = {}^{dx}\!/_{dt}$ is the gradient factor of the input function. Relative error F after n calculating cycles Ts referred to the final value is calculated as follows:

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

III-6.2 INTE (integrator (No. 51))





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

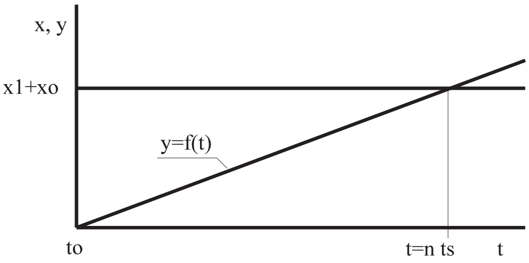
$$y1(t) = y(t0) + n \cdot \frac{ts}{T} \cdot (x1 + x0)$$

$$t = n \cdot ts$$

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

Ramp response:

Fig. 6



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.

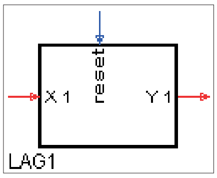
Example: Which is the value of output variable y after t=20s with a time constant of 100s, if a constant of x1 = 10 Volt is preset. Sampling interval ts is 100ms.

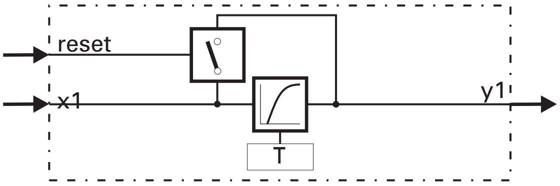
$$n = \frac{t}{t_s} \qquad n = \frac{20s}{0{,}1s} = 200s$$

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

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

III-6.3 LAG1 (filter (No. 52))





Dependent of control input reset, input variable x1 is passed on to output y1 with delay (reset= 0) or without delay (re - set = 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:

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

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

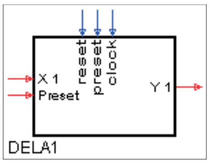
Analog output	
y1	Delayed output variable

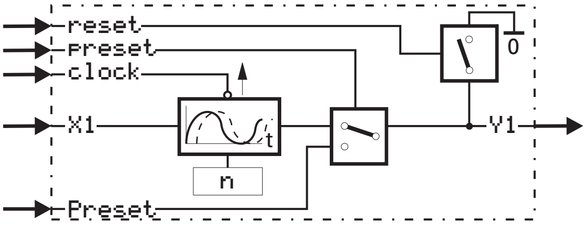
Parameter:

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

No configuration parameters!

III-6.4 DELA1 (delay time (No. 53))





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

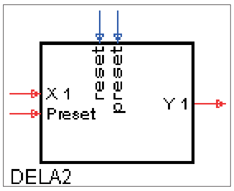
With several simultaneous control commands:
reset = 1 has priority over preset and stop
preset = 1 has priority over stop

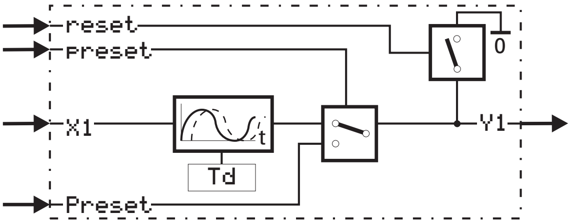
Analog output	
y1	Delayed output variable

Parameter:

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

III-6.5 DELA2 (delay time (No. 54))





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

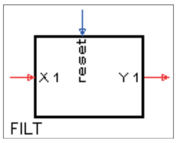
Analog input	
x1	Input variable to be delayed
Preset	Value output with delay by preset=1

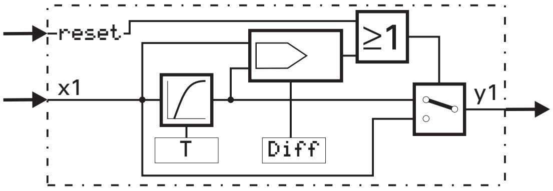
Analog output	
y1	Delayed output variable

Parameters:

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

III-6.6 FILT (filter with tolerance band (No. 55))



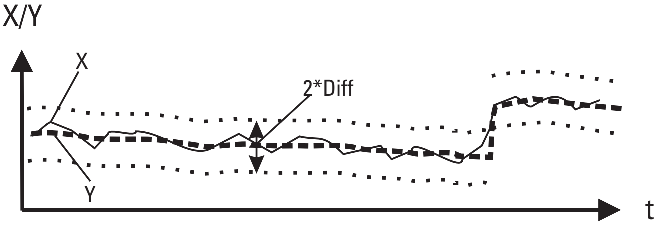


The complex transfer function of the filter within a tolerance band around the last output value ($| x1 - y1| <<= \delta$) is:

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

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.

Fig. 7



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:

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

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

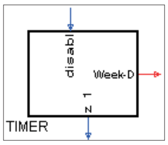
Analog input	
x1	Input variable to be delayed

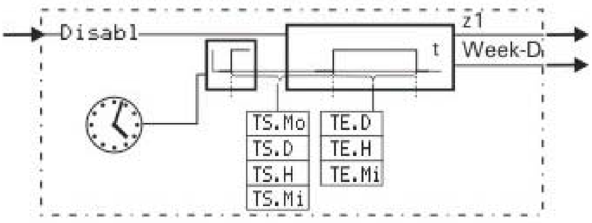
Analog output	
y1	Delayed output variable

Parameters:

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

III-6.7 TIMER (timer (No. 67))





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*:*

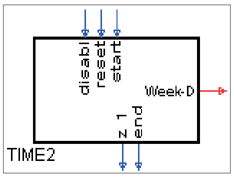
Parameter	Description	Range	Default
TS.Mo	Switch-on time month	0...12	0
TS.D	Switch-on time day	0...31	0
TS.H	Switch-on time hour	0...23	0
TS.Mi	Switch-on time minute	0...59	0
TE.D	Time duration days	0...255	0
TE.H	Time duration hours	0...23	0
TE.Mi	Time duration minutes	0...29	0

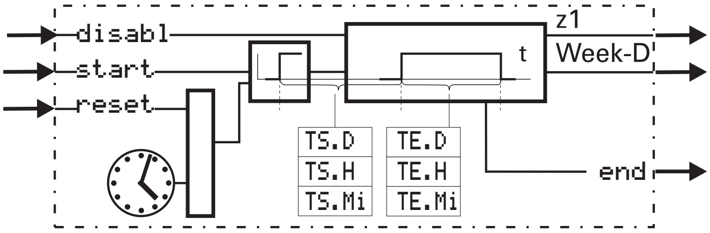
Configuration:

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!

III-6.8 TIME2 (timer (No. 70))





The function timer2 can only be used with real-time clock. 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)

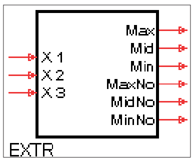
Parameters:

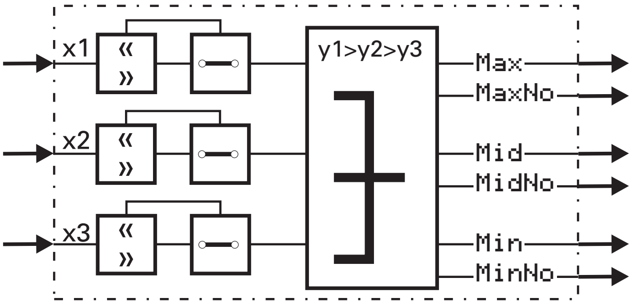
Parameter	Description	Range	Default
TS.D	Switch-on delay day	0...255	0
TS.H	Switch-on delay hour	0...23	0
TS.Mi	Switch-on delay minute	0...59	0
TE.D	Switch-on duration days	0...255	0
TE.H	Switch-on duration hours	0...23	0
TE.Mi	Switch-on duration minutes	0...29	0

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

III-7 Selecting and storage

III-7.1 EXTR (extreme value selection (No. 30))





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 not included into the extreme value selection, if:

-the input is not wired

-or the input value is higher than 1,5 . 10^{37} or smaller than -1,5 . 10^{37}.

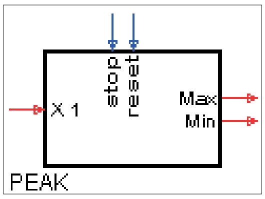
Number of failed inputs	Max	Mid	Min	MaxNo	MidNo	MinNo
0	x_{max}	x_{mid}	x_{min}	number of x_{max}	number of x_{mid}	number of x_{min}
1	x		x	number of x		number of x
2	the valid value			number of the valid value		
3	1,5 $_{1037}$	1,5 $_{1037}$	1,5 $_{1037}$	0	0	0

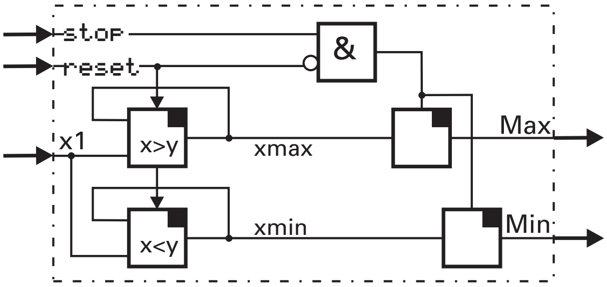
Inputs/outputs

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)

III-7.2 PEAK (peak value memory (No. 31))





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!

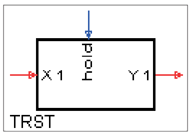
Inputs/outputs

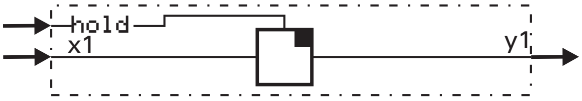
Digital inputs	
stop	With the stop input set to 1, instantaneous values Max and Min are unchanged.
reset	The reset input deletes the Min and Max values.

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

Analog outputs	
Max	Maximum value
Min	Minimum value

III-7.3 TRST (hold amplifier (No. 32))





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!

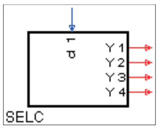
Inputs/outputs

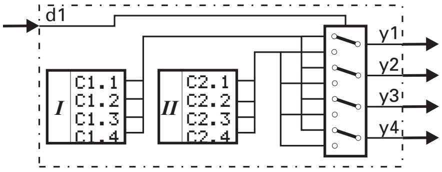
Digital input	
hold	Storage signal for the $_{x1}$ value

Analog input	
x1	Process value which can be output stored.

Analog output	
y1	Function output

III-7.4 SELC (Constant selection (No. 33))





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

Inputs/outputs

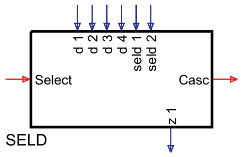
Digital input	
d1	Selecting the constant group (0 = group 1; 1= group 2)

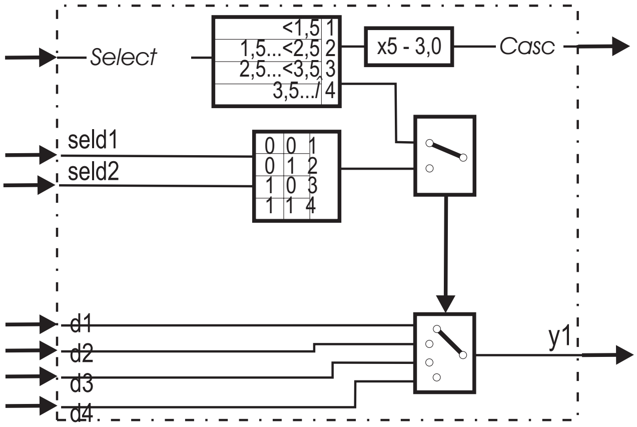
Analog outputs		
	d1 = 0 ≙ group 1	d1 =1 ≙ group 2
y1	c1.1	c2.1
y2	c1.2	c2.2
y3	c1.3	c2.3
y4	c1.4	c2.4

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

III-7.5 SELD (selection of digital variables - function no. 06)





Selection of one of the 4 digital inputs is either by an analog “Select” signal or by the two digital control signals seld1, seld2. If analog control signal Select is connected, selecting is done using this control signal. If the input is not connected, selection is by means of the two digital control inputs seld1, seld2.

This function block is cascadable. The select input can be linked with the cas output of another SELD block into a choice of 8 digital variables.

Inputs/outputs

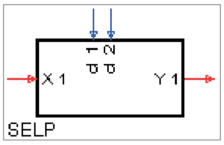
Digital inputs	
d1	Input, is output at z1, when seld1=0 and seld2=0
d2	Input, is output at z1, when seld1=0 and seld2=1
d3	Input, is output at z1, when seld1=1 and seld2=0
d4	Input, is output at z1, when seld1=1 and seld2=1
seld1	1st control signal for variable selection (least significant bit)
seld2	2nd control signal for variable selection (most significant bit)

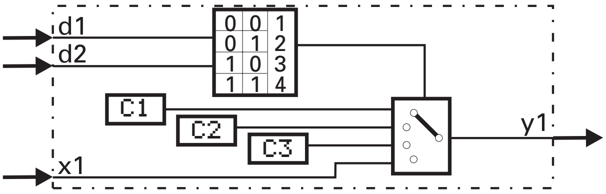
Analog inputs	
Select	Dependent on input value, the relevant variable is output at the z1 output C1.4

Digital outputs	
z1	d1, d2, d3. or d4 According to the input value of Select (or to values seld1, seld2) the relevant input variable is output.

Analog outputs	
Casc	Cascade output = Select – 3.0

III-7.6 SELP (parameter selection (No. 34))





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

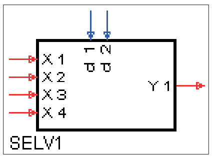
Analog input	
x1	Input is output at y1 , when d1 = 1 and d2 =1

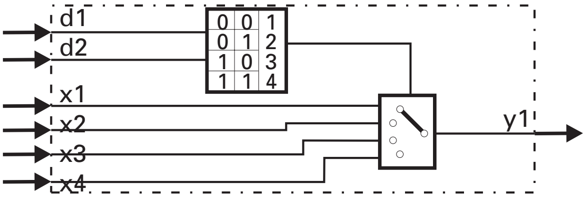
Analog outputs		
	d1	d2
y1 = C1	0	0
y1 = C2	0	1
y1 = C3	1	0
y1 = x1	1	1

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

III-7.7 SELV1 (variable selection (No. 35))





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

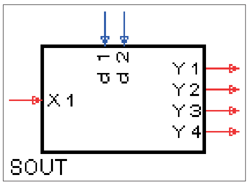
Digital inputs	
d1	1st digital input for parameter selection
d2	2nd digital input for parameter selection

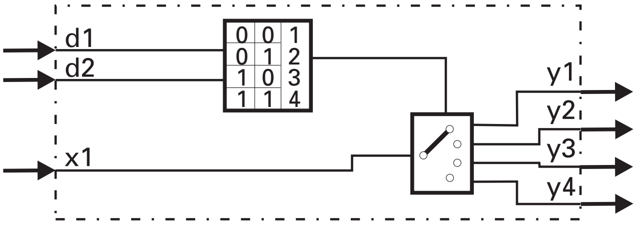
Analog inputs	
x1	Input is output at y1, when d1 = 0 and d2 =0
x2	Input is output at y1, when d1 = 0 and d2 =1
x3	Input is output at y1, when d1 = 1 and d2 =0
x4	Input is output at y1, when d1 = 1 and d2 =1

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

No parameters!

III-7.8 SOUT (Selection of output (No. 36))





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

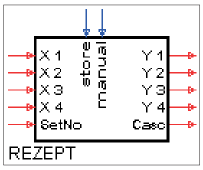
Digital inputs	
d1	1st digital input for output selection
d2	2nd digital input for output selection

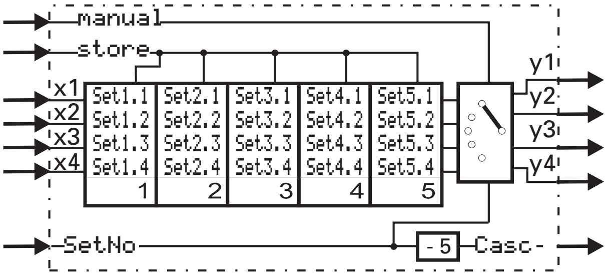
Analog input	
x1	Input is output at y1 , when d1 = 0 and d2 =0

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

No parameters!

III-7.9 REZEPT (recipe management (No. 37))




manual
store
x1
x2
x3
x4
Set1.1 Set2.1 Set3.1 Set4.1 Set5.1
Set1.2 Set2.2 Set3.2 Set4.2 Set5.2
Set1.3 Set2.3 Set3.3 Set4.3 Set5.3
Set1.4 Set2.4 Set3.4 Set4.4 Set5.4
1 2 3 4 5
y1
y2
y3
y4
SetNo
- 5
Casc

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.

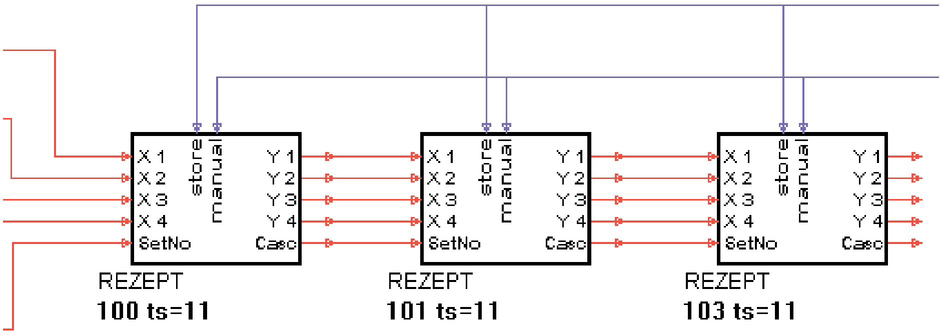
In mode STORE (store = 1), the values applied to x1... x4 are written into the memory addresses of the recipe block selected with input SetNo.

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.

Values of the used analog inputs are stored as parameter values when detecting a positive edge at the store- input. This input should be activated only with relevant changes of the input values. Too frequent storing operations may lead to the destruction of the EEPROM! (→ page 311)

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 $_{x1}$... $_{x4}$ 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.

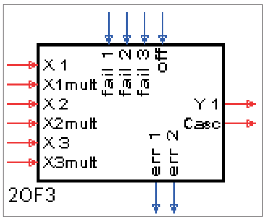
Analog outputs	
y1...y4	The values at y(i) correspond either to the recipe block selected with SetNo or to inputs x(i) in manual mode (store =1).
Casc	The value at output Casc is the value of input SetNo reduced by 5 and is used for cascading.

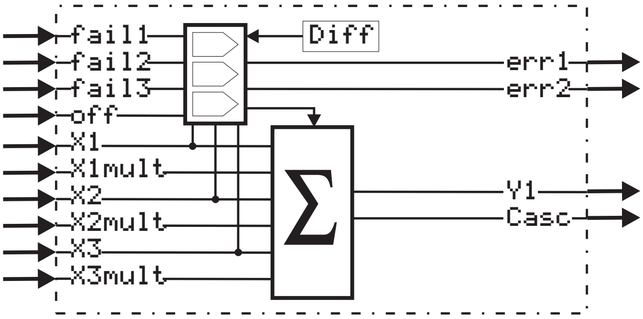
Parameters:

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

III-7.10 2OF3 (2-out-of-3 selection with mean value formation (No. 38))





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.

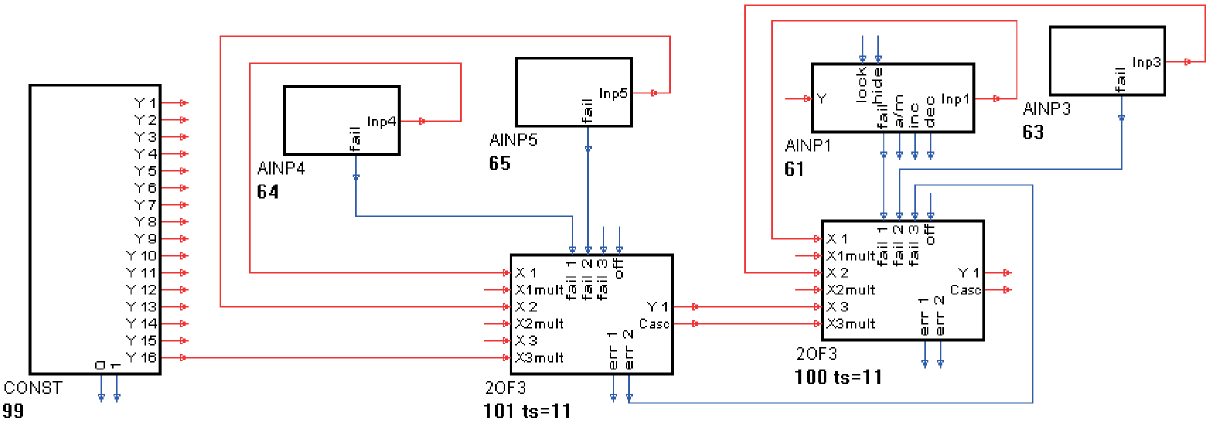
Output Casc indicates the number of values used for mean value formation. This is important with 2OF3 function cascading.

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

Fig. 8 Cascading example



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 $_{x1}$ 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.

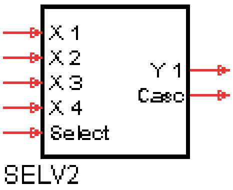
Analog outputs	
y1	arithmetic mean value or switching to x1 occurred (off = 1 or several inputs defective).
Casc	Factor: number of the values used for mean value formation. Casc = x1mult + x2mult + x3mult.

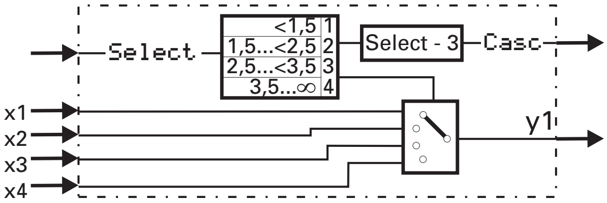
Parameter

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!

III-7.11 SELV2 (cascadable selection of variables (No. 39))

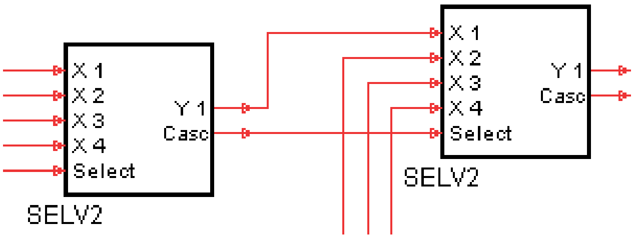




Dependent of input Select,, one of the four inputs x1...x4 is connected with output y1. Unused inputs are interpreted as 0. Output Casc = input Select -3.

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

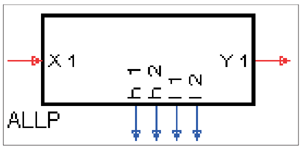
Analog inputs	
x1	Input is output at y1 with Select< 1,5.
x2	Input is output at y1 with 1,5 < Select < 2,5.
x3	Input is output at y1 with 2,5 < Select < 3,5.
x4	Input is output at y1 with Select < 3,5.
Select	Dependent of input value, the relevant variable is output at y1 .

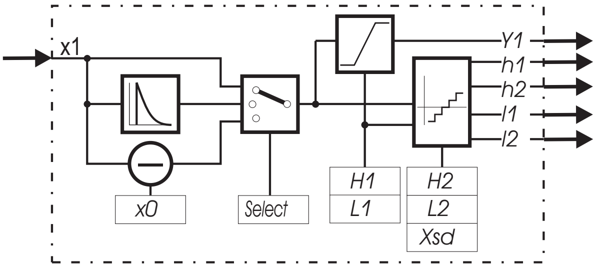
Analog outputs	
y1	According to the input value of Select, the relevant input variable is output.
Casc2	Cascade output = Select - 3

No parameters!

III-8 Limit value signalling and limiting

III-8.1 ALLP (alarm and limiting with fixed limits(No. 40))

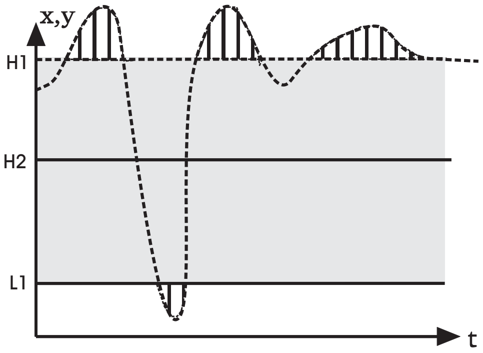




Signal limiting:

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.

Fig. 9 Limiting at H1 < L1



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.

Fig. 11 Offset of the alarm limits

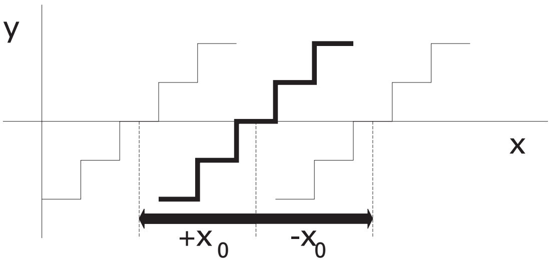
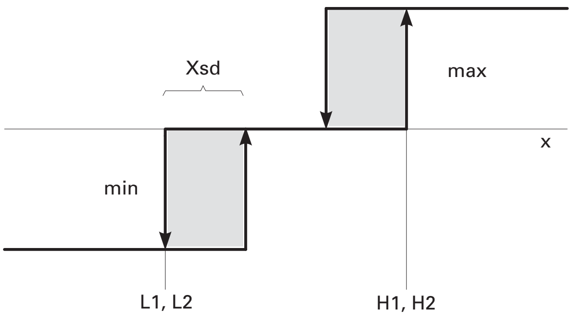


Fig. 10 Switching hysteresis and alarm limits:



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 *x1*.

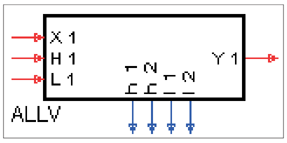
Configuration parameter:

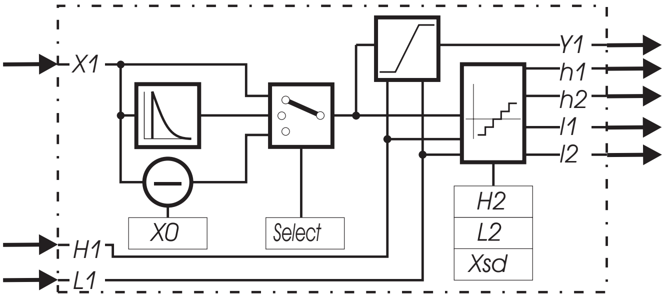
Parameter	Description		Range	Default
Select	Selection of the variable to be monitored	x1	x1	←
		D alarm	dx1/dt	
		Alarm with offset	x1-x0	

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

III-8.2 ALLV (alarm and limiting with variable limits (No. 41))



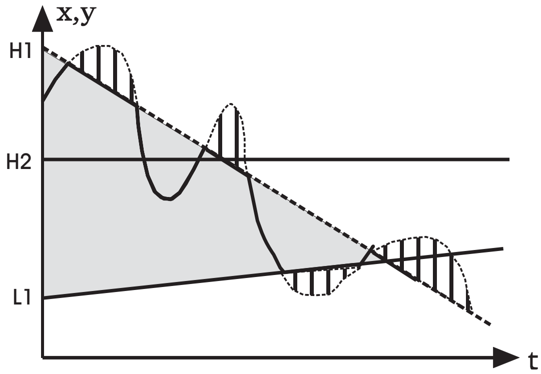


Signal limiting:

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 !

Fig. 12 Limiting at H1 < L1



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.

Fig. 14 Offset of the alarm limits

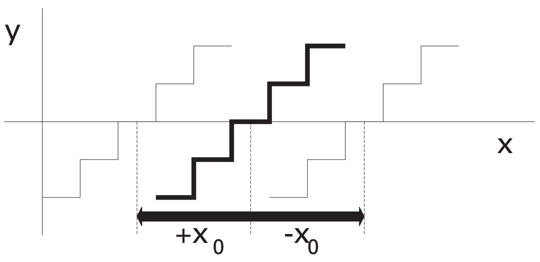
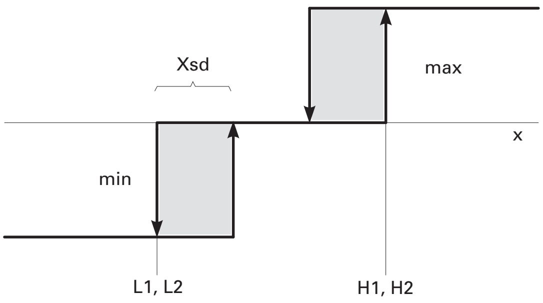


Fig. 13 Switching hysteresis and alarm limits:



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.

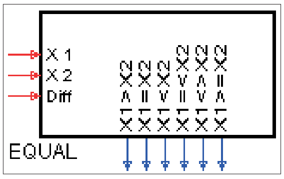
Configuration parameters:

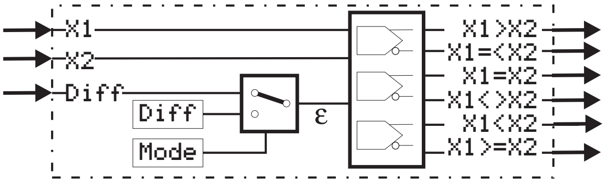
Parameter	Description		Range	Default
Select	Selection of variable to be monitored	x1	x1	←
		D alarm	dx1/dt	
		Alarm with offset	x1-x0	

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

III-8.3 EQUAL (comparison (No. 42))





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

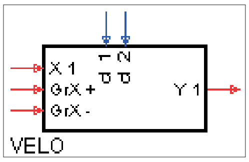
Digital outputs	
z1	z1 = 1 with x2 + Diff < x1
z2	z2 = 1 with x2 - Diff ≤ x1 ≤ x2 + Diff
z3	z3 = 1 with x2 - Diff > x1
z4	z4 = 1 with x2 + Diff ≥ x1
z5	z5 = 1 with x2 - Diff > x1 > x2 + Diff
z6	z6 = 1 with x2 - Diff ≤ x1

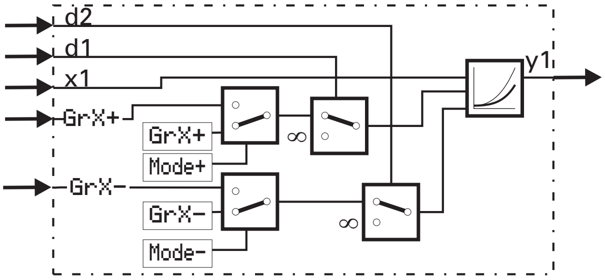
No configuration parameters!

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

III-8.4 VELO (rate-of-change limiting (No. 43))





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 [1/s] *with parameter Mode+ = Inp. GrX+*
GrX−	negative gradient [1/s] *with parameter Mode- = Inp. GrX-*

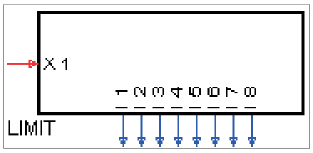
Analog output	
y1	Limited input value *x1*

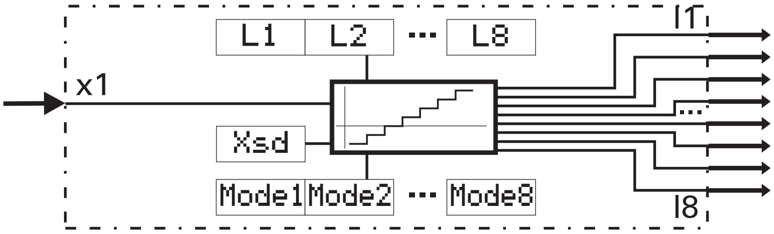
No configuration parameters!

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 [1/s] with parameter	Mode+ = Para.GrX+	0 ... 999 999	0
Grx−	negative gradient [1/s] with parameter	Mode− = Para.GrX−	-29 999 ... 0	0

III-8.5 LIMIT (multiple alarm (No. 44))



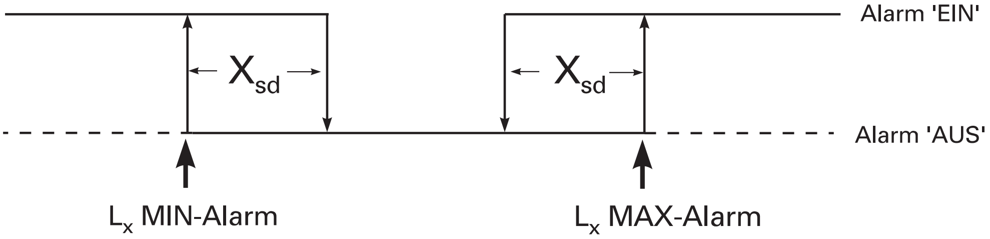


The function checks input variable x1 for 8 alarm values L1...L 8. Dependent of configuration by Mode 1 ... Mode 8, the relevant alarm value is evaluated as MAX or MIN alarm.

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).

Fig. 15



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

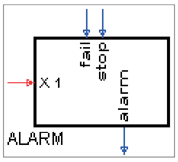
Configuration parameters:

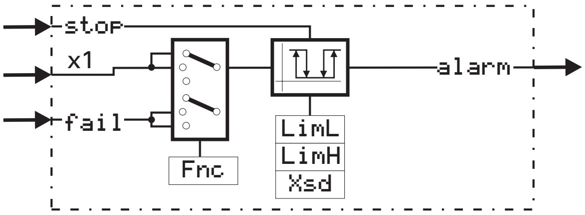
Parameter	Description		Range	Default
Mode1 ... Mode8	alarm functions of the 8 alarms	**max-alarm**	MAX-alarm	←
		min-alarm	MIN-alarm	

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

III-8.6 ALARM (alarm processing (No. 45))



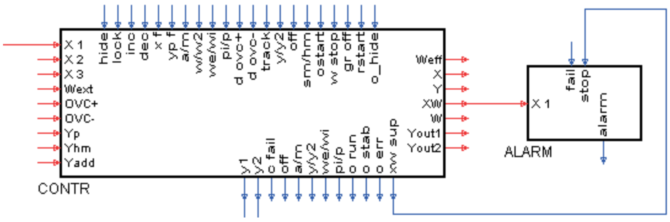


x1 is checked for a lower and an upper alarm value. Additionally, digital alarm input fail can be used. Configuration parameter Fnc can be used to select which signal shall be monitored (x1, x1 + fail or fail).

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 change.

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

Fig. 16 Alarm suppression with change



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:

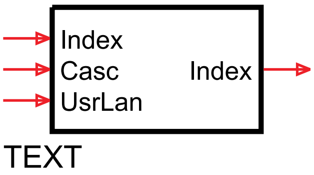
Parameter	Description		Range	Default
Fnc	Alarm function	**only x1 is monitored**	Mes.val.X1	←
		x1 and fail are monitored	X1 + fail	
		only fail is monitored	fail	

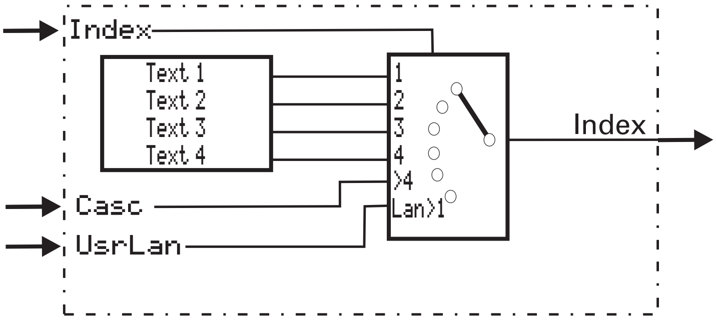
Parameters:

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

III-9 Visualization

III-9.1 TEXT (text container with language-dependent selection (No. 79))





The text block contains a list of user texts which can be displayed by various operating pages (programmer, VWERT and ALARM). These texts can be displayed and adjusted as a selection list on a VWERT page (e.g. for plain text selection of recipes).

The function block is cascadable, if more than 4 texts are availabe for selection.

Texts can be entered only via engineering tool: 4 texts of 16 characters

Inputs/outputs

Analog inputs	
Index	Input for text selection
Casc	Cascade input for further text blocks in the same language
UsrLan	Input for a text block with texts in a further language

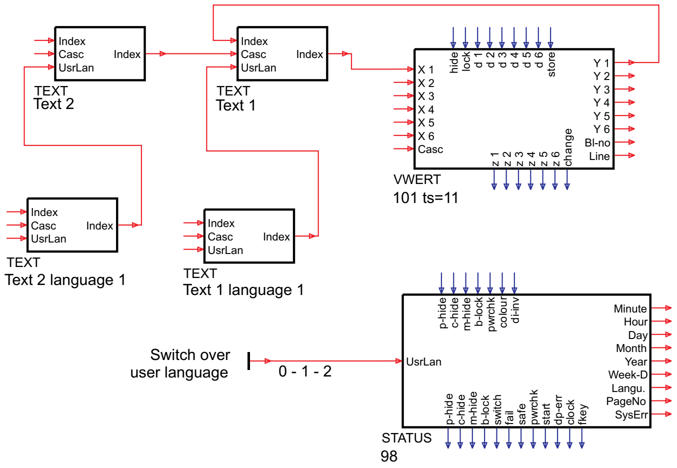
Analog outputs	
Index	Number of selected text in the text block

Output “Index” of the last text block in a text cascade must be wired to the block on the operating page of which the texts must be used, e.g. VWERT. The number of the display text is allocated to the index input of this text block.

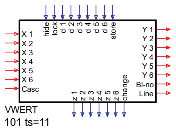
The texts can be extended into any number of texts via the cascade input (Casc). For this, wire the index output of the subordinated block (texts 5 ...8) to input “Casc” of the next text block. The index for text selection is adjustable only at the index input of the last block (see example below).

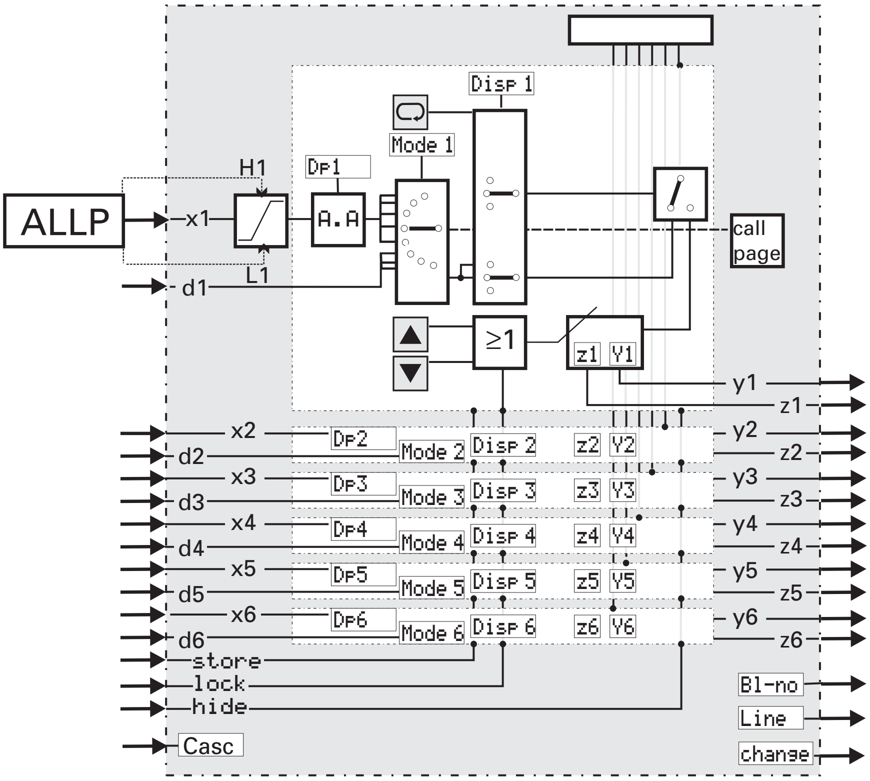
For user language selection, wire the index output of the (language) text block to language input UsrLan of the used text block. During selection of the user language, its texts will replace the texts of the first text block. User language selection is central at the status block.

Fig.17 : Wiring of cascaded text blocks. User language selection is via the status block.



III-9.2 VWERT (display / definition of process values (No. 96))





General

This function permits display or definition of 6 analog or digital process values in 6 display lines. These values can be changed also via the KS98-1 communication interface. The function block can be cascaded, whereby a scroll field with more than 6 lines can be implemented on the operating page.

- 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).
- Possible display functions are: analog, digital, text, menu, push-button, switch and radio button.
- Normally, the values applied to the inputs are displayed.
- A value adjustable at the front panel is output at the relevant function output.
- Only adjustable lines are selectable.
- The change of these values from the operating level can be switched off (lock).
- 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.

⚠ **Values of the used analog inputs are stored as parameter values when detecting a positive edge at the store- input. This input should be activated only with relevant changes of the input values. Too frequent storing operations may lead to the destruction of the EEPROM! (→ page 311)**

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
change	If a value is changed during operation, the change-output of the VWERT-block is set to 1 for one calculating cycle.

Analog inputs:	
x1 ... x6	Process values to be displayed (default = 0)
casc	By wiring the casc-input with the bl-no output of another VWERT, cascades can be set up

Analog outputs:	
y1 ... y6	Valid process values
Bl-no	Block number of this output
line	If a value is changed during operation, for one calculating cycle the line-output of the VWERT-block is set to that value that was changed (1-6) .

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	analog value display	analog	←
		digital value display	digital	
		value display in time format	time	
		selection group (radio button)	radio	
		toggle function	switch	
		push-button function (pressed =1)	push-button	
		text selection	text	
		menu function (page changing)	menu	
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. Dependent on whether a line was configured as an analog, digital, radio, switch, push-button or menu line, all 16 characters (e.g. mMode x = digital) or only the first 6 characters (e.g. mode x= analog) are shown in the device. Further detailed information on the various display types is given at the end of the section.

With digital displays (digital, switch, push-button and radio):

signal = 0: dependent of line of 'Text1 a ... Text6 a'
signal = 1: dependent of line of 'Text1 b ... Text6 b'

VWERT operating page

The VWERT operating page can be selected in the operating page menu with non-activated 'hide' input.

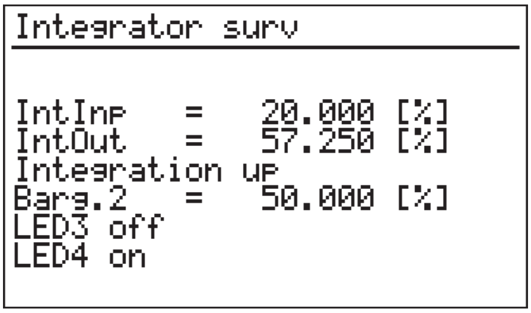
For information on the operation, see section Operating pages on page 36

☞ If aline is configuratd as display, the value can't be changed.

☞ Operation of line modes radio, switch and push-button is described in section I.9.4 "Adjusting values".

☞ When using the VWERT block in an enginering, operation and function must be described separately.

Fig. 18



```
Integrator surv

IntInp    =    20.000 [%]
IntOut    =    57.250 [%]
Integration up
Barg.2    =    50.000 [%]
LED3 off
LED4 on
```

VWERT block cascade connection
For linking several VWERT operating pages, the Bl-no output of a further VWERT must be wired with the Casc input of the calling VWERT. Thereby, the last page which must be connected can be linked also to the start page (ring structure).

Cascade connection of a VWERT block is indicated by arrows ▲▼ on the display page. A previous block (Bl-no output wiring) above the first line and a next block (Casc input wiring) below the last line are marked, otherwise, these arrows are omitted. When setting the cursor onto one of these arrows and pressing key Enter, a change to the relevant VWERT page occurs. With standard exit from the called up VWERT page a change to the selection list of the operating pages is made.

Selectable display modes in detail

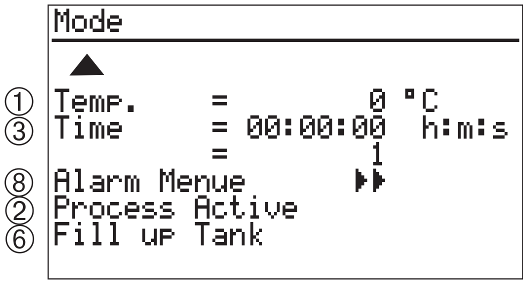
① **Data type "analog"**

The line contains 2 static texts (6 characters) and the analog value connected to X1...X6. Changing the value is done as described above, if changeability is configured.

With the corresponding input Xn wired via a type ALLP function block, its limits H1 (max. limit) and L1 (min. limit) are used as adjustment limits for this value. Unless an ALLP block is connected to the input, limits –29999 to 999999 remain valid.

Example: Value with limits:
Apart from its maximum number of digits behind the decimal point, each value can have its own adjustment limits, which are determined by parameter values L1 and H1 of a previous ALLP block. Unless the source of the display value is the VWERT, the ALLP uses these parameter values for limiting the value.

Fig. 19 VWERT page with the line modes: marking of previous page, analog, time, text selection, menu, digital, switch



```
  Mode
   ▲
① Temp.    =         0 °C
③ Time     = 00:00:00  h:m:s
           =         1
⑧ Alarm Menue       ▸▸
② Process Active
⑥ Fill up Tank
```

② Data type "digital"

Dependent digital input bit value of the relevant line, text "0" (Name_n) or text "1" (Unit_n) is displayed.
With a static input value, static text output can be generated (e.g. headline).

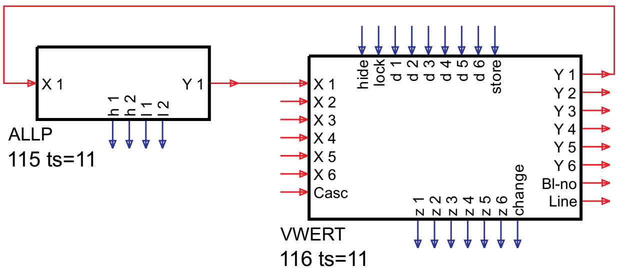
Example: value with limits:
In addition to its maximum number of digits behind the decimal point, each value can have its adjustment limits which are determined by parameter values L1 and H1 of a connected ALLP block.
Unless the display value source is the VWERT itself, the ALLP limits the value with these parameter values.

③ Data type "time" (analog output)

Data type "time" can be used to display or to adjust times in HH:MM:SS or HH:MM, whereby the last decimal digit indicates the full minutes. The digits behind the decimal point are for display of seconds.

- With the digits behind the decimal point DP set to 0, adjustment of seconds is not possible. Only hours and minutes can be adjusted. If the relevant configuration value DP is 2, adjustment of seconds is also possible.

Fig. 20



- From a time above 100 hours, seconds are not displayed any more.
- The adjustment range is within 00:00:00 - 15999:59 hours. Due to the limited resolution of a float value, adjustment ist possible only in steps of 6 seconds from value 16:40:00.

④ Data type "radio" (radio button; digital output)

Data type "radio button" can be used for changing over between combined selection fields.

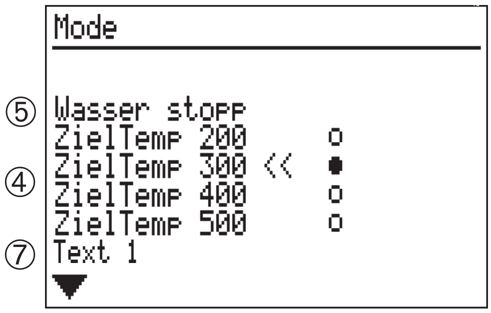
- After selecting, changing is done directly without starting with function key .
- "Radio button" arranged successively in one VWERT form a common group.
- Only one element of this group is activated.
- By actuating the function key, the radio button on which the cursor is standing is activated. All relevant other ones are de-activated.
- A new group starts, if another data type is defined between 2 radio buttons.
- Unless a radio button is activated during data transmission to VWERT, all radio buttons remain inactive. If more than 1 button is active, the 1st one of the group is activated, the other ones are inactive.

⑤ Data type "switch" (digital output)

Data type "switch" can be used to implement switch-on/switch-off (toogle) functions.

- After selection, adjusting is done directly without starting with function key .
- Pressing the function key will activate a de-activated switch and de-activate an activated switch.

Fig. 21



```
Mode

Wasser stopp
ZielTemp 200      ○
ZielTemp 300 <<   ●
ZielTemp 400      ○
ZielTemp 500      ○
Text 1
▼
```

⑥ Data type "push-button" (digital output)

Data type "push-button" can be used to realize short switch-on/switch-off (hold) functions.

- After selection, adjusting is done directly without starting with function key H.
- As long as the function key is pressed, the output is activated. When releasing the key, the output is de-activated.

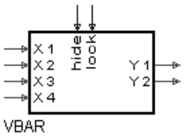
⑦ **Data type "text" (analog output, see also: Function block TEXT)**
Data type "text" can be used for display of indexed texts for analog integer signals. Moreover, an analog value can be allocated to a text when adjusting.

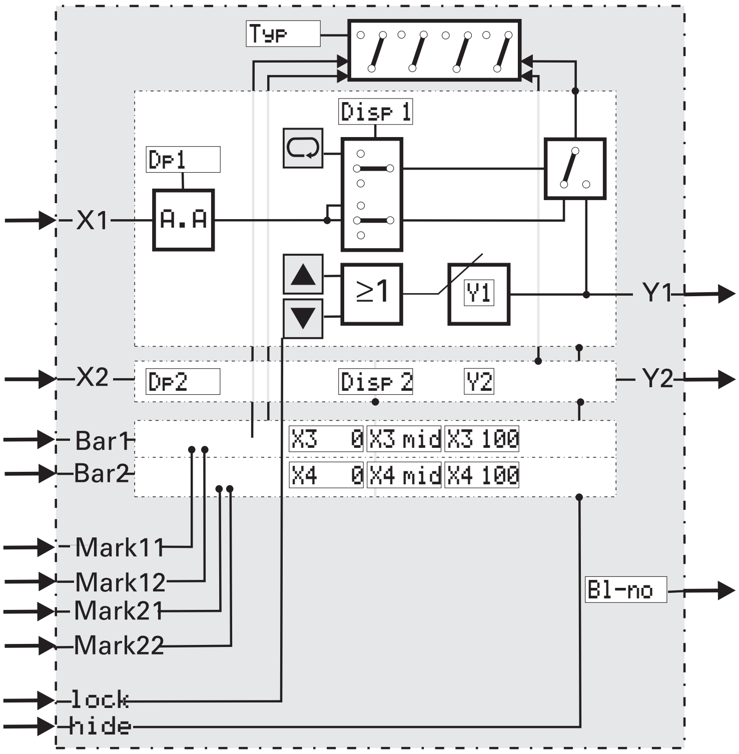
- The corresponding input must be connected with the index output of a text block.
- The number of the text to be selected (VWERT output Y1...Y6) is applied to the index input of the first (next to VWERT) text block.
- The text blocks cascaded by wiring the index output of another text block with the Casc input of the relevant text block. Text selection is always via the index input of the text block next to VWERT.
- Via the UsrLan input, text blocks of different language can be appended. Language switchover (language index) is defined by the value at the UsrLan input of status block 98. Unless an appropriate text block for the language is available (e.g. language index too high), the corresponding text is output in the last language block found.
- When selecting a text in the VWERT to be displayed, the number of selectable texts is limited by the number of connected text blocks.
- If the index for text selection comes from a different origin, no text is displayed if the index is beyond the possible text selection (0 or >max). VWERT marks the line with "——————————".
- With text selection at VWERT, the initial value (parameter Y1...Y6) should be set > 0 to avoid a start value of "——————————" .

⑧ **Data type "menu"**
Data type "menu " can be used for changing to other operating pages (menu without sub-menu items, linking not possible).

- The value applied to the corresponding input is interpreted as block number of the operating page to which changing is required.
- Changing to the specified page is done by pressing the Enter key. Unless the page is accessible, a change to the operating page selection list is made. This list contains all the blocks which are available for selection .
  If a page is not accessible, the reasons may be:
  1. Block number not defined
  2. Block number does not have an operating page
  3. Block cannot be displayed instantaneously because hide = 1 .
- When making a standard exit from the operating page, return is to the calling VWERT page.
- When using this procedure for changing to a VWERT operating page which also contains a menu type line, no further change will occur.

III-9.3 VBAR (bargraph display (No. 97))





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. Another 4 analog inputs can be used to position 2 markings at each side of the bar within the bargraph range. These markings can be e.g. alarm limits or reference values. With open marker inputs, or out-of-range marker values, display of markers is suppressed.

- Determination of horizontal or vertical bargraph is via configuration (type).
- Determination if value displays are visible or switched off
- Configuration of start values x3mid or x4mid determines, if the bargraph points only in one direction (from top or bottom) or in 2 directions from the middle.
- 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 are the initial values for Power-On.
- 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 a positive edge at the store input, the values applied to the signal inputs are stored as parameters y1 and y2, i.e. as output values.
- When an ALLP is connected at inputs x1 and x2, its limits L1 and H1 are used for parameter adjustment.

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.

⚠ **Values of the used analog inputs are stored as parameter values when detecting a positive edge at the store- input. This input should be activated only with relevant changes of the input values. Too frequent storing operations may lead to the destruction of the EEPROM! (→ page 311)**

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

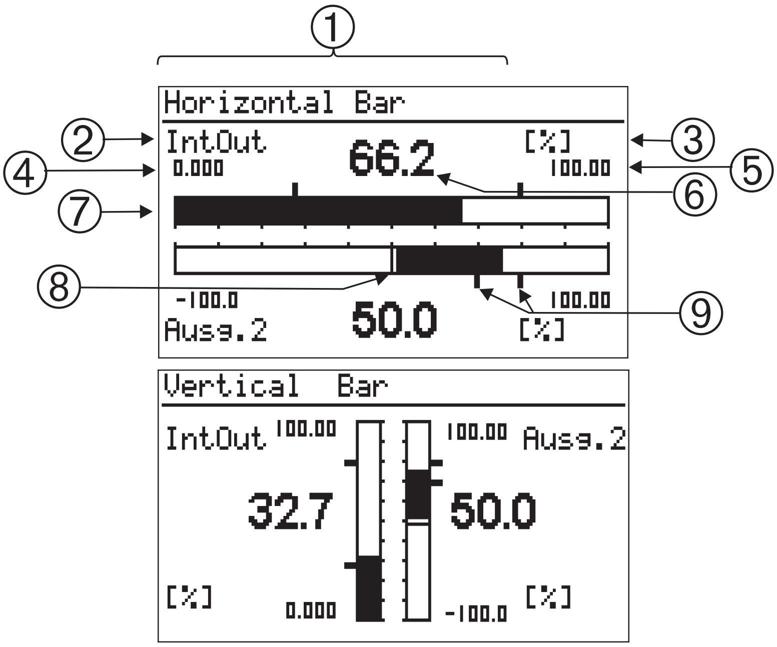
Configuration	Description		Values	Default
Disp1 Disp2	Function of numeric display 1 and 2	display x1 / x2, value adjustable	disp+adj	
		only display x1 / x2	display	←
		x1 / x2 = empty	empty	
Dp1 / Dp2	Digits behind the decimal point in numeric display 1 / 2		0 ... 3	0
Typ	Position of bargraphs	Both 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

VBAR operating page

VBAR has an operating page, which can be selected in the operating page menu with the 'hide' input not used. Changing the texts displayed in the unit is only possible in the engineering tool! Max.16 characters can be entered in each text parameter. A value configured as display cannot be changed.

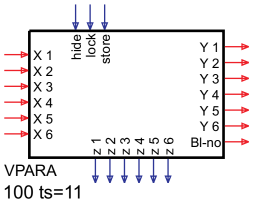
The following values or texts are displayed:

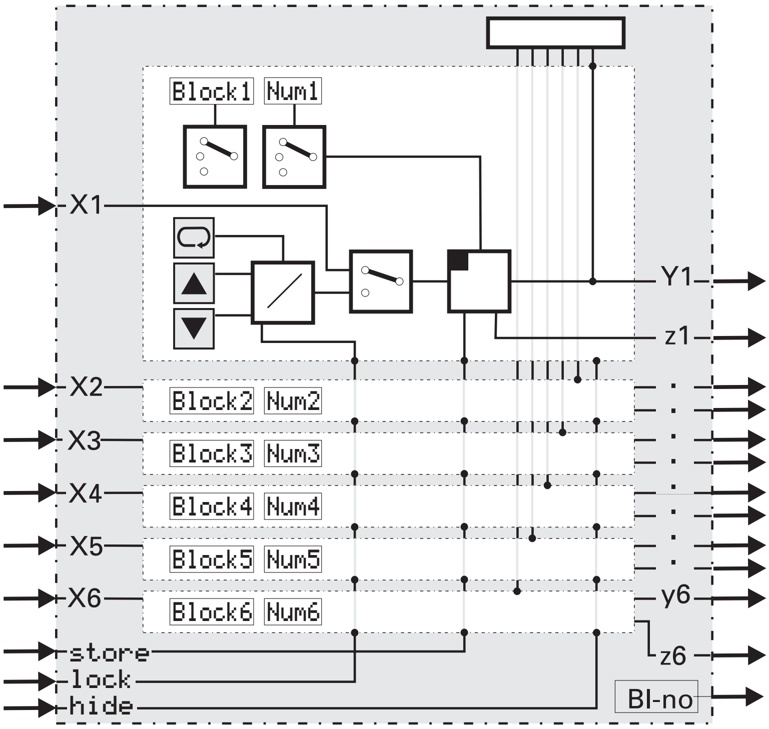
Fig. 22



① Title
② Process value name for X1 (first 6 characters of 'Name 1')
③ Unit for X1 (first 6 characters of 'Unit 1')
④ Scale start of bar for value X1
⑤ Scale end of bar for value X1
⑥ Process value display/input field
⑦ Bar for value X1
⑧ Middle of bar X2 (starting point)
⑨ Marker at right/bottom bar for X2 (the same applies accordingly to the other bar)

III-9.4 VPARA (parameter operation (No. 98))





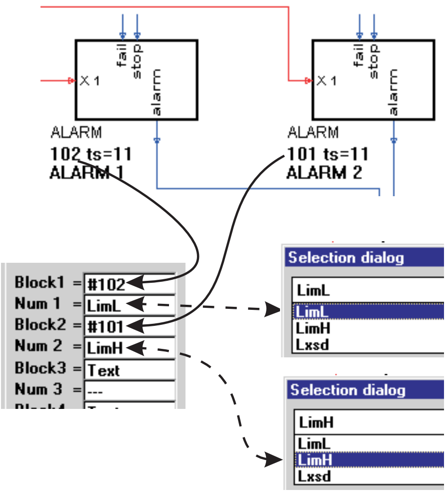
General

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.

Fig. 23



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'.
BL-no	Block number of this output

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

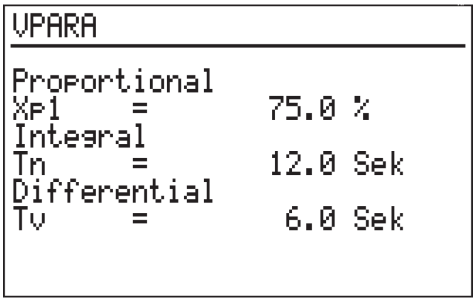
Changing the texts displayed in the unit is possible only in the engineering tool! Max. 16 characters can be entered in each text parameter. Dependent of whether a line is allocated to a block number or defined as a text line, all characters (Blockx = Text) or only the first 6 characters (Blockx = #xxx)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

VPARA operating page

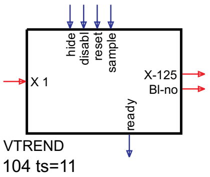
VPARA has an operating page, which can be selected in the operating menu with input 'hide' not used.

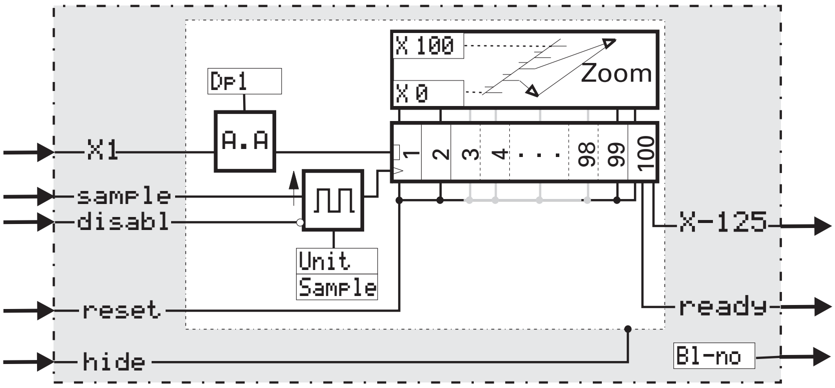
Fig. 24



```
VPARA

Proportional
Xp1    =      75.0 %
Integral
Tn     =      12.0 Sek
Differential
Tv     =       6.0 Sek
```

III-9.5 VTREND (trend display(No. 99))





General

Function VTREND collects 125 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 125 values, the value 125 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.

The properties of KS98-1 function block VTREND are:

- The resolution of the KS98-1 Y axis is 60 pixels.
- The resolution of the X axis is 125 pixels.
- With further trend blocks connected at the output of a trend block (cascade), these blocks can be viewed by shifting the time axis (scrolling the time axis).
- The Y resolution can be magnified by factor 4, and scrolling throughout the range is possible in steps of 12,5%. The zero shift remains unchanged in the background when returning to normal resolution.
- The old settings also remain unchanged when leaving the operating page and calling it up again.
- The lower scanning time limit is set to 0,01 for unit hours.
- Output Bl-no provides the operating page block number.

5 accesses via the communication interface, each providing data packages of 25 trend data from KS 98-1, are available.

When connecting 2 trend blocks to one trend output by mistake in the case of a cascade, the one with the lower number is ignored. The number of cascaded blocks is not limited.

If blocks in the chain have different scanning times or different ranges, data display is faulty. No warning is output. The trend display continues when scrolling in the time axis (paging into the past) rather than being stopped.

With voltage failure, the sampled values remain unchanged.

☞ The changing of the texts displayed, is only possible in the engineering-tool! Max. 16 digits can be entered for every textparameter.

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).
BL-no	Block number of this output

Configuration data

Configuration	Description		Values	Default
Unit	Unit of sampling interval	seconds (s) Minutes (m) Hours (h)	sec. min. 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

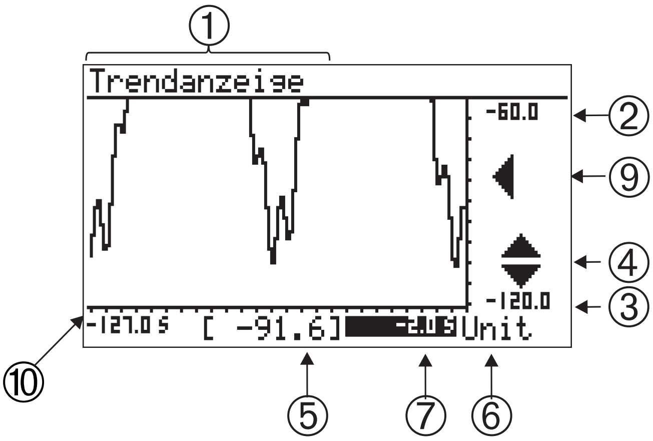
Text display and entry

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

VTREND operating page

VTREND has an operating page which can be selected in the operating page menu with input 'hide' no connected. The operating page is provided only for display of trend data. Making changes in the input fields will change only the visible trend display section, without making changes to the data.

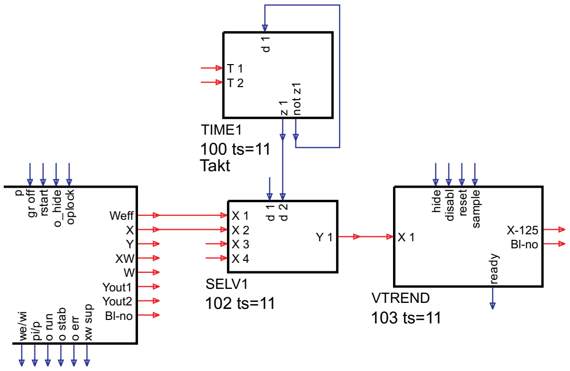
Fig. 25



① Title
② ③ scale end values
④ Zoom switchover
⑤ Value at time ⑦ / actual input value
⑥ Unit of value
⑦ Origin (start) of time axis related to the actual value
(=0) shift of time axis (scrolling into the past)
⑨ Axis shift signalling
⑪ End of time axis / earliest value in the displayed trend

Examples:

Trend recording with 2 curves



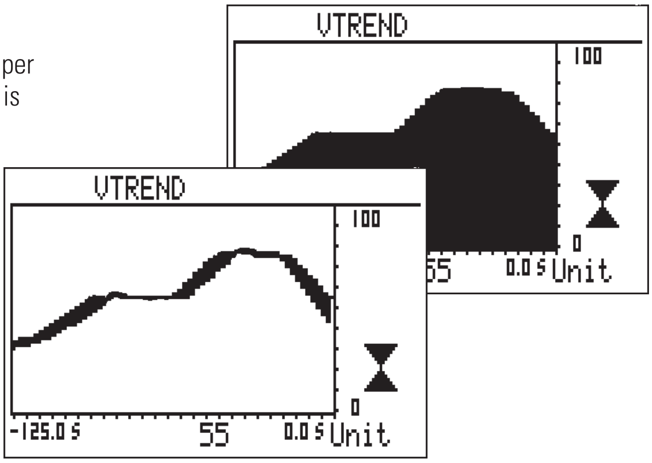
Although distinction of different curves is not possible, display of two values on a trend page may be purposeful (e.g. controller and process value, or one value and zero, in order to have a curve).

In the example, a clock triggering switch-over between the values together with the SELV1 is generated by means of a pulse.

In the example, a TIME1 can be used to generate a clock for switching over between values with SELV1.

E.g. for making a record in VTREND at intervals of a second, unit is s and sample is 1.

In order for the TIME1 to change between 0 and 1 once per second, T1 and T2 must be set to 0,9 s. One cycle (0.1s) is lost for detection of the own output change).



In the following example, a pulse is used to create a clock which generates switchover between values with SELV1.

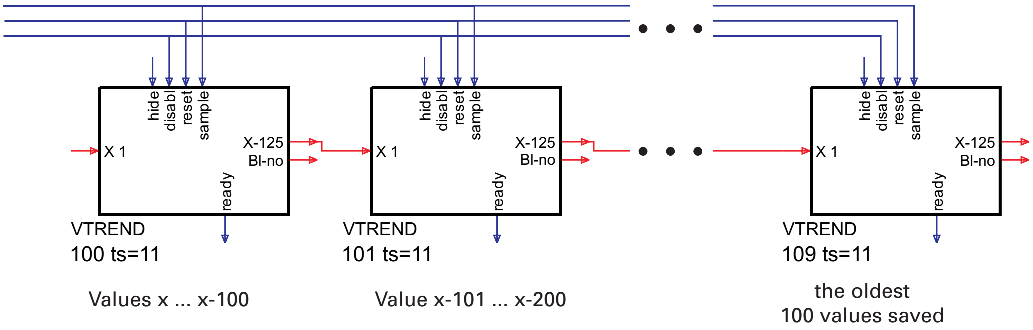
For making a record at intervals of a second
e.g. in VTREND,
Unit is set to s and
Sample is set to 1.

Settings:
Unit = s and S
ample =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

Fig. 26 Example of trend- /datarecording with n values



Trend or data recording with any number of values can be realized by cascading VTREND function blocks.

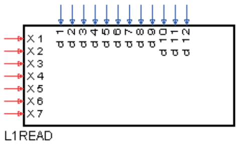
Limiting refers only to the number of available block numbers and to the calculation time. The data sequence is dependent of VTREND function block wiring. In wiring direction the block numbers must be ascending.

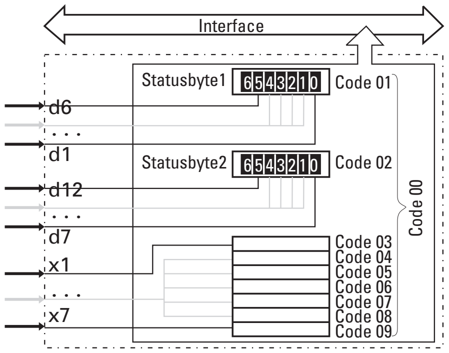
III-10 Communication

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.

III-10.1 L1READ (read level1 data(No. 100))





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

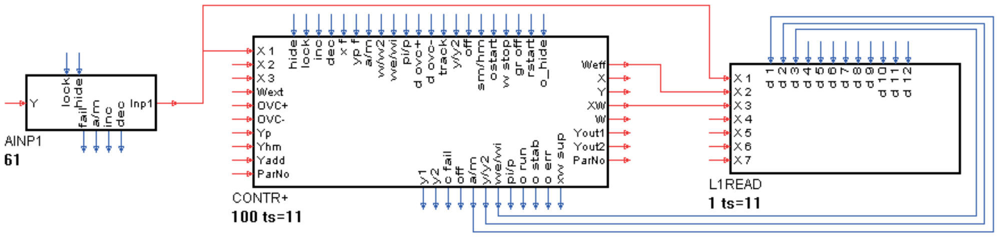
Digital inputs:	
d1 ... d6	Digital process values, which can be read via interface (status byte 1). (Default = 0)
d7 ... d12	Digital process values, which can be read via interface (status byte 2). (Default = 0)

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

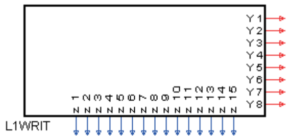
Engineering example

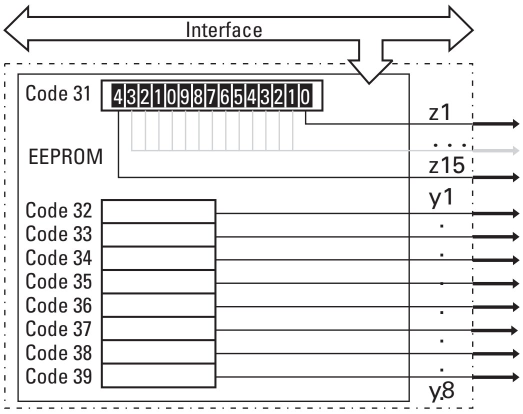
In the following example, several process data (process value, effective and control deviation) and controller statuses (automatic/manual, Wint/Wext and y/Y2) are connected with the L1READ function block. Now, these data can be read in a message via interface.

Example for L1READ engineering



III-10.2 L1WRIT (write level1 data (No. 101))





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

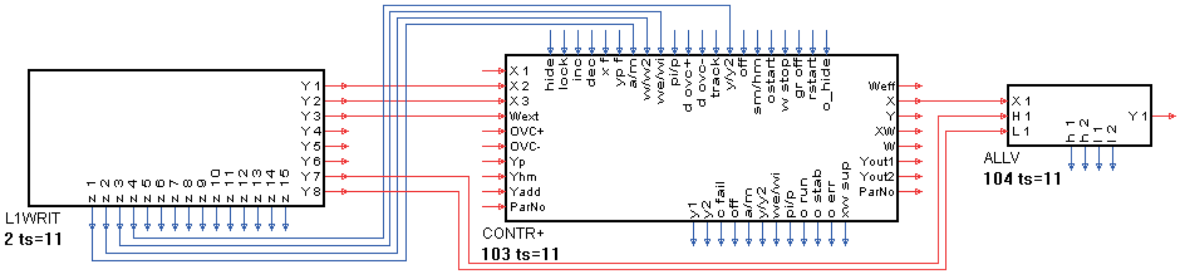
Digital outputs:	
z1 ... z12	Digital process values, which can be written via the interface (default = 0)

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 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.

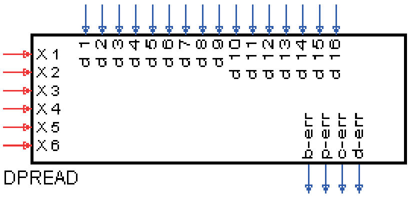
Example for L1WRIT engineering

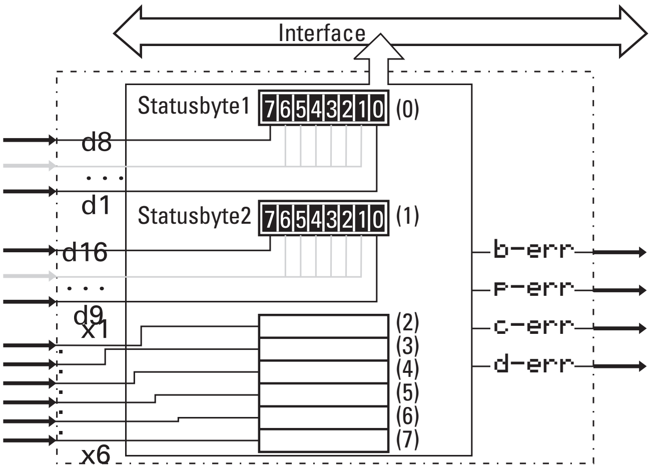


PROFIBUS

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.

III-10.3 DPREAD (read level1 data via PROFIBUS (No. 102))





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).

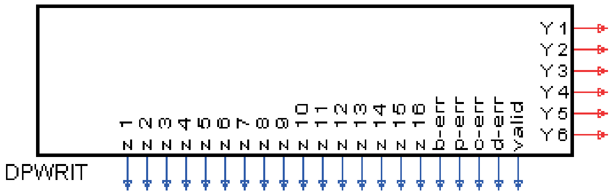
Inputs/outputs

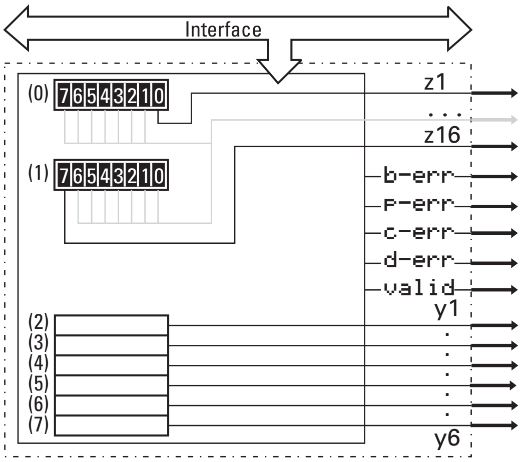
Digital inputs:	
d1 ... d8	Digital process values, which can be read via the PROFIBUS (status byte 1)
d9 ... d16	Digitale process values, which can be read via the PROFIBUS (status byte 2)

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

III-10.4 DPWRIT (write level1 data via PROFIBUS (No. 103))





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

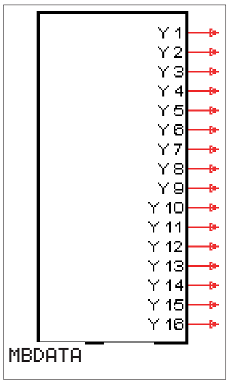
Digital outputs:	
z1 ... z16	Digital process values, which can be written via the Profibus.
b-err	PROFIBUS status: 1 = bus access not successful
p-err	PROFIBUS status: 1 = faulty parameter setting
c-err	PROFIBUS status: 1 = faulty configuration
d-err	PROFIBUS status: 1 = no data communication
valid	PROFIBUS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

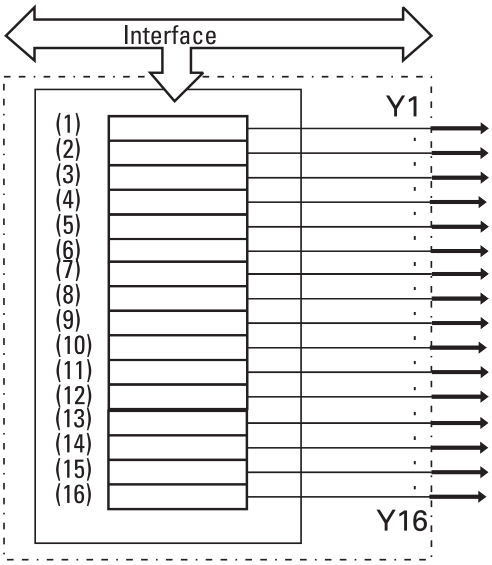
Analog outputs:	
y1 ... y6	Analog process values, which can be written via the Profibus.

MODBUS

In total, a maximum of 5 function blocks can be configured. Any combination of functions is possible. In the functions, any data may be used.

III-10.5 MBDATA (read and write parameter data via MODBUS - no. 104)



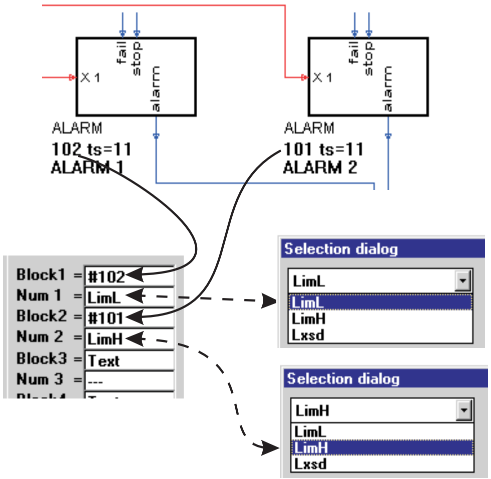


General

The new MBDATA function block behaves analogously to the already known function block VPARA and provides access via MODBUS. Thus up to 16 parameters of other function blocks available in the engineering can be read or changed via MODBUS.
Each parameter is declared to the MODBUS function including block and parameter number by means of two configuration data.

The engineering tool supports parameter setting using a special operating sequence in which the parameter numbers of the selected block are selected by means of the parameter descriptions (→ refer to the drawing opposite) .

For additional information :
see KS98-1 Modbus interface description
“sb_ks98-1_mod_e_9499-040-88711.pdf”.



Inputs/outputs

Analog outputs:	
Y1...Y16	Analog process values that can be read or written via interface (default = value of the assigned parameter or "0"). The values of the 16 parameters are output. Unused parameters provide a value of '0'.

Configuration data

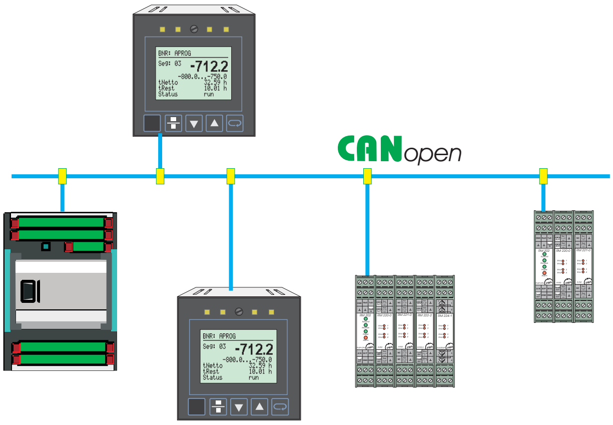
Configuration	Description	Values	Default
Block1...Block16	Block number of the parameter	— —*	— —*
Num 1...Num 16	Number of the parameter	— —*	— —*

* To avoid confusion and thus operator errors, we recommend setting the block numbers and parameters exclusively using the engineering tool where the parameters must be specified with their short-form descriptions.

III-11 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 KS98-1 (cross communication)



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 KS98-1 can be used. With the SIL switch closed, the terminating resistor is activated.

As default, the SIL switch is open (see opposite).

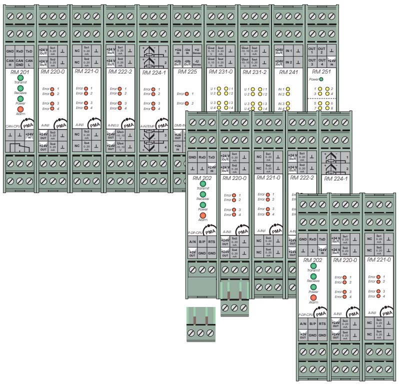
III-11.1 RM 211, RM212 and RM213 basic modules

The RM 200 system is a basic module (housing) for snap-on rail mounting with 3, 5 or 10 slots.

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).

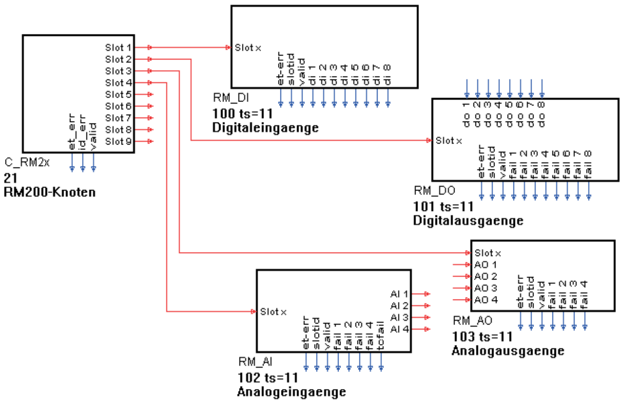
⚠ **Wiring in the engineering-tool must be according to real wiring (Position = Slot = socket).**

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

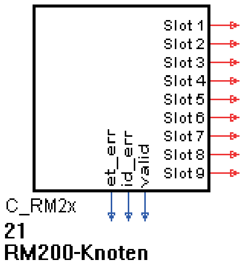


The connecting terminals can be withdrawn easily from the modules.

Fig. 27 Partial engineering for communication with an RM200 node



III-11.2 C_RM2x (CANopen fieldbuscoupler RM 201 (No. 14))



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

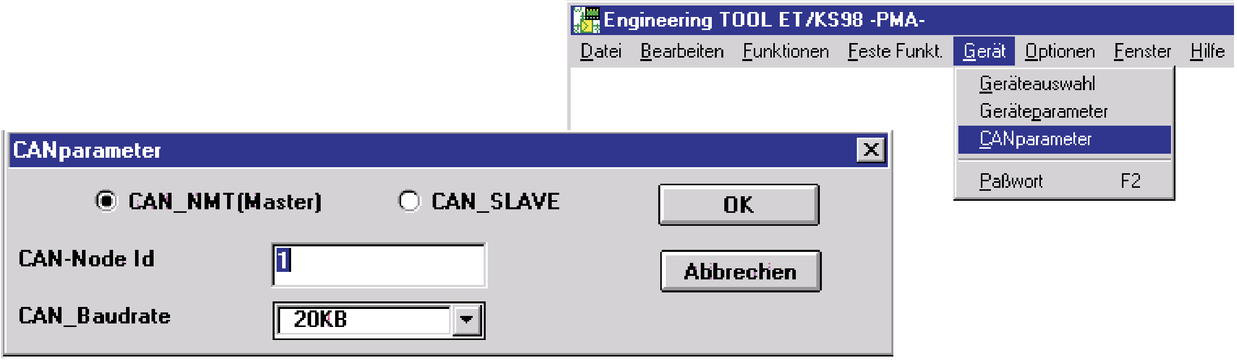
Unlike the other KS 98-1 functions, only one data function may be connected to the analog outputs.

Parameters and configuration data

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

Prerequisite for communication between KS98-1 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.

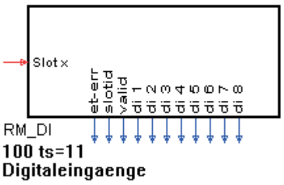


● CAN_NMT(Master) ○ CAN_SLAVE

CAN-Node Id 1

CAN_Baudrate 20KB

III-11.3 RM_DI (RM 200 - (digital input module (No. 15))



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)

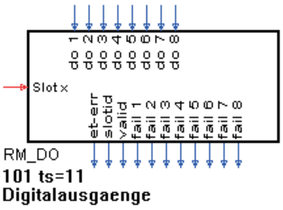
Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions at a slot)
slotid	0 = correct slot assignment	1 = faulty slot assignment (wrong RM module inserted)
valid	0 = no data	1 = data could be received
di 1 ... di 8	1st to 8th digital input signal	

* Slot = conn.-no., socket e.g. 2...10

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

III-11.4 RM_DO (RM 200 - digital output module (No. 16))



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	s for digital inputs 1 to 8

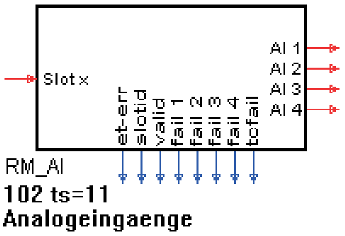
Digital outputs		
et-err	0 = no engineering error detected	1 = engineering error (several RM module functions at a slot)
slotId	0 = correct slot assignment	1 = faulty slot assignment (faulty RM module fitted)
valid	0 = no data	1 = data could be received
di 1 ... di 8	1st to 8th digital input signal	

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

Note related to hardware type RM 251
The outputs are monitored pairwisely. To avoid faulty displays, unused outputs should be short circuited.

III-11.5 RM_AI (RM200 - analog input module (No. 17))



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

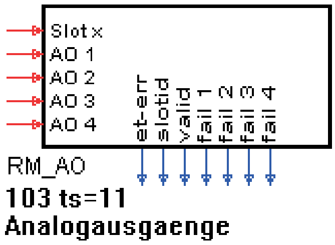
Parameter and configuration data

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

Potentiometer connection and calibration

See chapter "calibration" → page 35

III-11.6 RM_AO (RM200 - analog output module (No. 18))



Function RM_AO handles the data from connected analog output modules.

Input and outputs

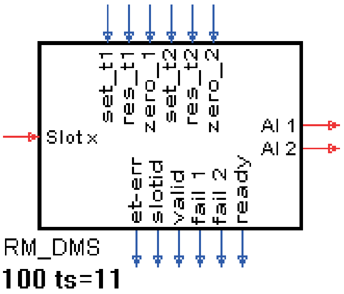
Analog inputs	
Slotx	Connection of one of the Slot outputs of the RM 200 node (C_RM2x)
AO 1...AO 4	1st to 4th analog output signal

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

III-11.7 RM_DMS (strain gauge module (No. 22))



Function RM_DMS reads data from a special strain gauge module of KS98-1 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

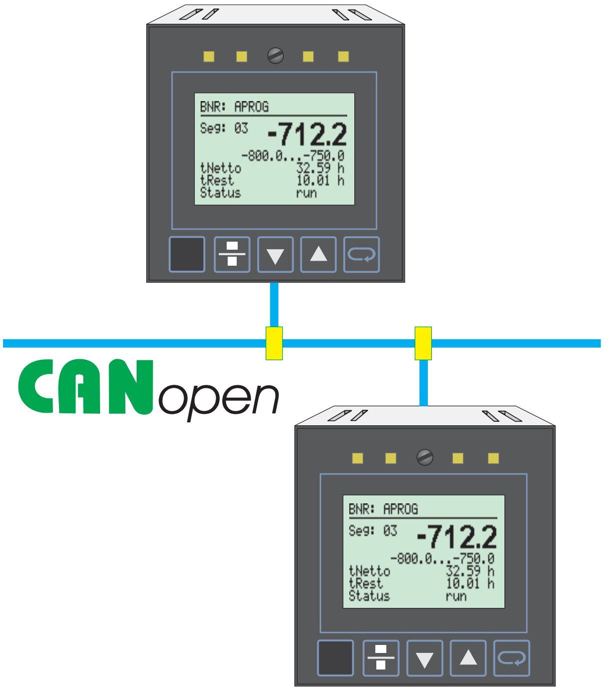
Analog inputs:	
	connection of one of the slot outputs of the RM201-1-node block

Analog outputs:	
AI 1	1st measured value of strain gauge channel 1
AI 2	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)

III-12 KS 98-1- KS 98-1 cross communication (CANopen)

Data exchange between KS 98-1 and RM200, KS800 or KS816 must be done exclusively via KS98-1.



KS 98-1 RM

To every KS 98-1, even a slave, one or more RM nodes can be associated. But he can only access his own I/O.

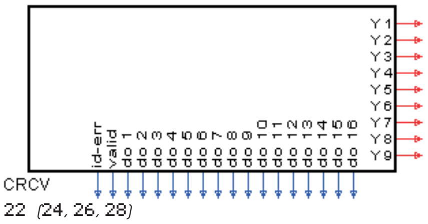
Data exchange between several KS 98-1 of a CAN network is via send modules (CSEND; block numbers 21, 23, 25, 27) and receive modules (CRCV; block numbers 22, 24, 26, 28).

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: 175

III-12.1 CRCV (receive mod. block no's 22,24,26,28 (No.56))



Function CRCV can receive data from a different KS98-1. 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 KS98-1 from CSEND no. 21
CRCV no. 24 reads the data of another KS98-1 from CSEND no. 23
CRCV no. 26 reads the data of another KS98-1 from CSEND no. 25
CRCV no. 28 reads the data of another KS98-1 from CSEND no. 27

Outputs

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

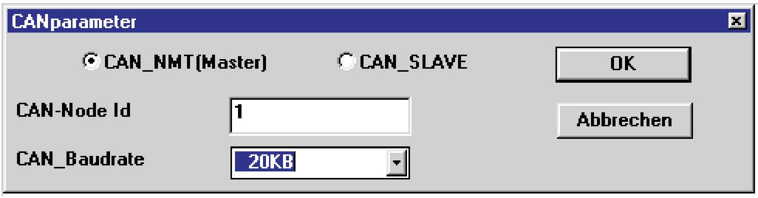
Digital outputs		
id-err	0 = correct node Id	1 = faulty node Id
valid	0 = no data	1 = data could be received

do 1
...
do 16
	Status values 1 to 16	

Parameter and configuration data

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

* 1) The node address of the sending KS98-1 is adjustable in engineering tool window "CANparameter" or via the instrument parameters on the front panel (during off-line mode).

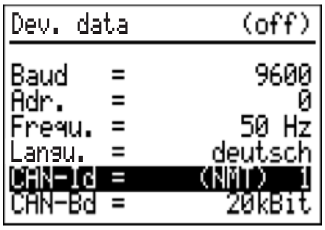


CANparameter

● CAN_NMT(Master) ○ CAN_SLAVE

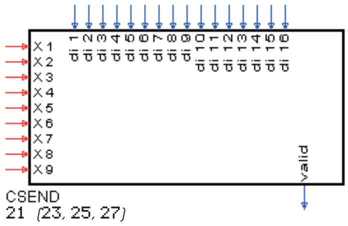
CAN-Node Id: 1

CAN_Baudrate: 20KB



```
Dev. data      (off)
Baud    =       9600
Adr.    =          0
Frequ.  =      50 Hz
Langu.  =    deutsch
CAN-Id  =    (NMT) 1
CAN-Bd  =     20kBit
```

III-12.2 CSEND (Send mod. blockno.'s 21, 23, 25, 27 - (No. 57))



Function CSEND provides data for other KS98-1 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.**

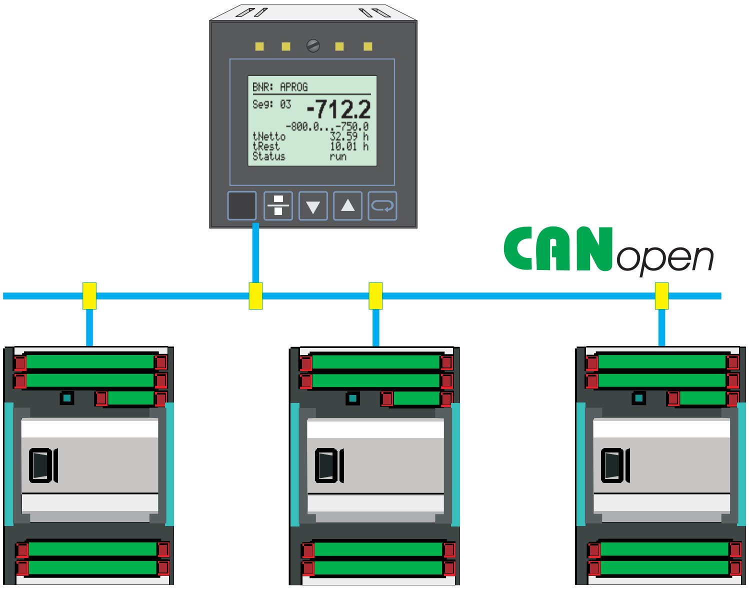
Digital output		
valid	**0 = invalid data (e.g. no KS98-1 but only KS 98-1)**	**1 = data could be received**

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.

III-13 Connection of KS 800 and KS 816



Function blocks C_KS8x and KS8x can be used for communication of multifunction unit KS98-1 and multi-channel temperature controllers KS 800 and KS 816.

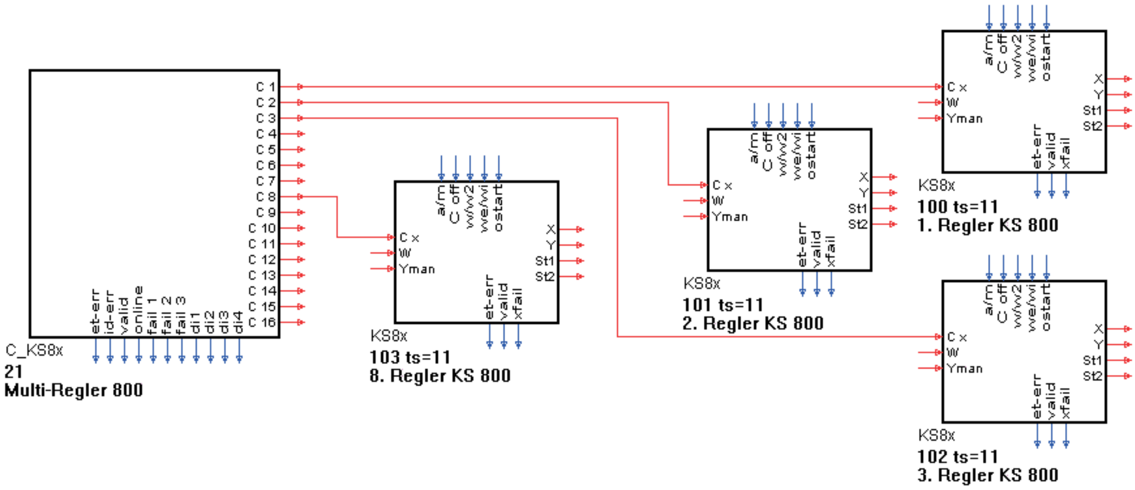
A node function C_KS8x is allocated to each KS 800 or KS816.

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

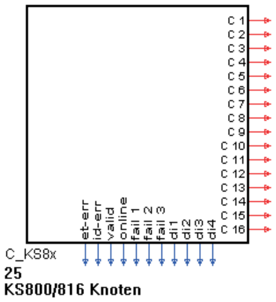
For BUS terminating resistor, see page 175

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

Fig. 28



III-13.1 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 KS 98-1 functions, only one data function can be soft-wired to each analog o utput.
Prerequisite for communication of KS98-1 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	

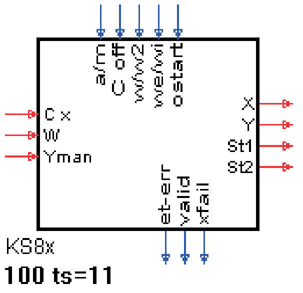
Parameter und Konfigurationsdaten

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).

III-13.2 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**
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 is active (safety)
we/wi	0 = external is active	1 = internal 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	

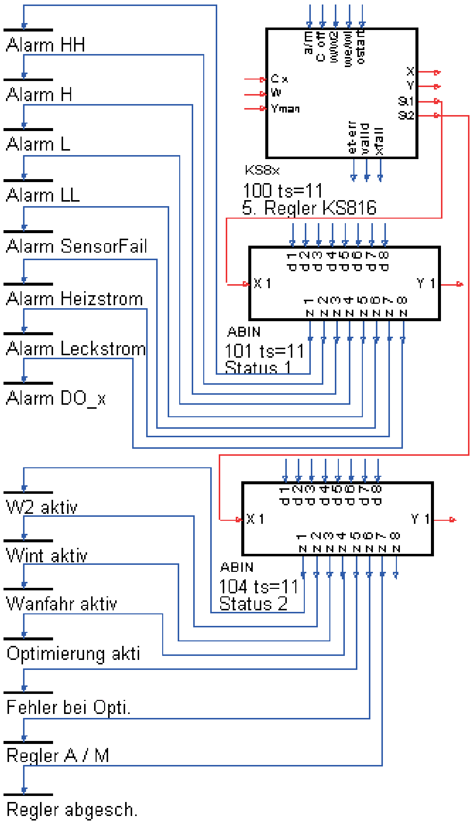
St.1 Statusbyte 1

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	- - - -

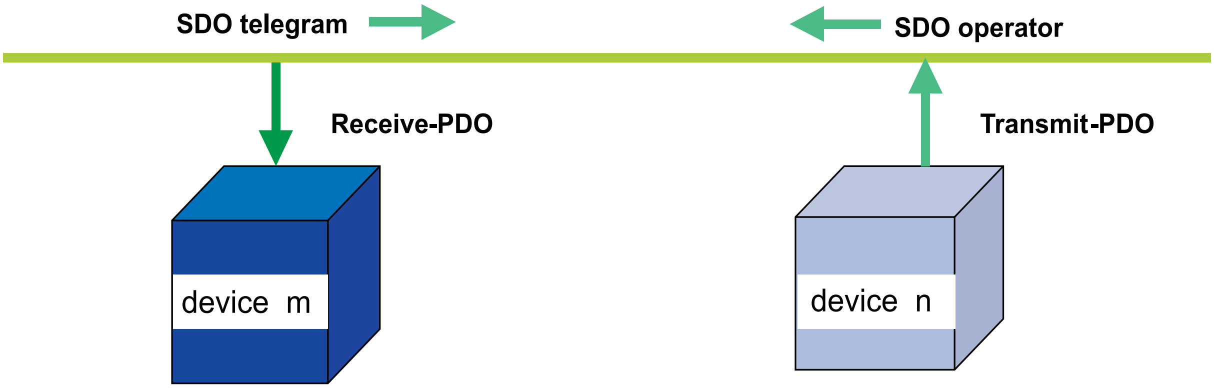
Engineering example for evaluationSt1/ St2



III-14 Description of KS 98-1 CAN bus extension

There are various modes for KS 98-1 communication via the CAN bus. The unit can be master for handling the NMT services (NMT = Network ManagemenT), or slave, can send or receive PDOs (PDO = process data object) cyclically or send SDO telegrams asynchronously (SDO = service data object). A KS98-1 can contact any bus parties simultaneously with other KS98-1, allocated remote IOs, KS800 multi-controllers and up to 40 sensors or actuators, and via asynchronous telegrams. Max. 42 CAN nodes can be addressed.

KS 98-1 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 the details 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.



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

Data-producer

Priority control via address

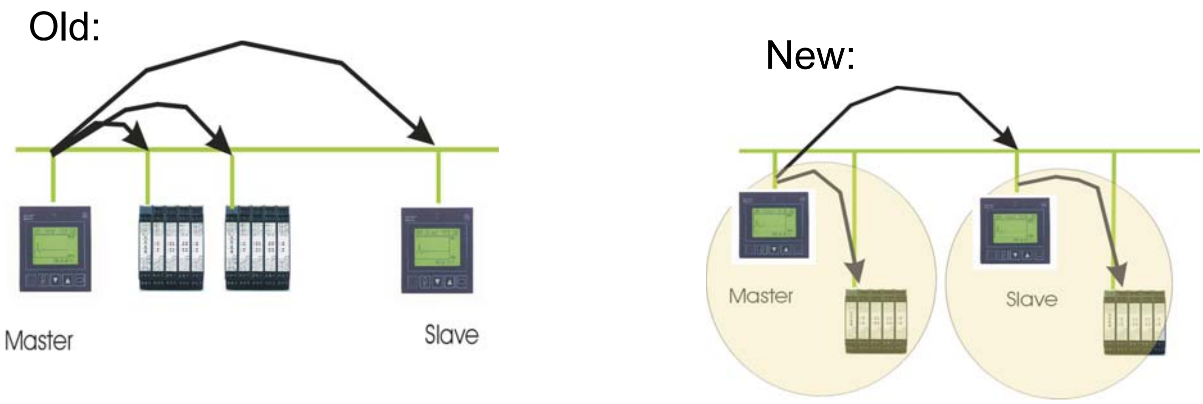
Instrument-specific data contents

Service Data Objects (SDO)

for data without real-time requirement

asynchronous, confirmed messages

Repartition over several telegrams is possible



KS98-1 CAN communication features

Every message on the bus activates the KS 98-1 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 KS 98-1 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 KS 98-1.

KS98-1 cross communication

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

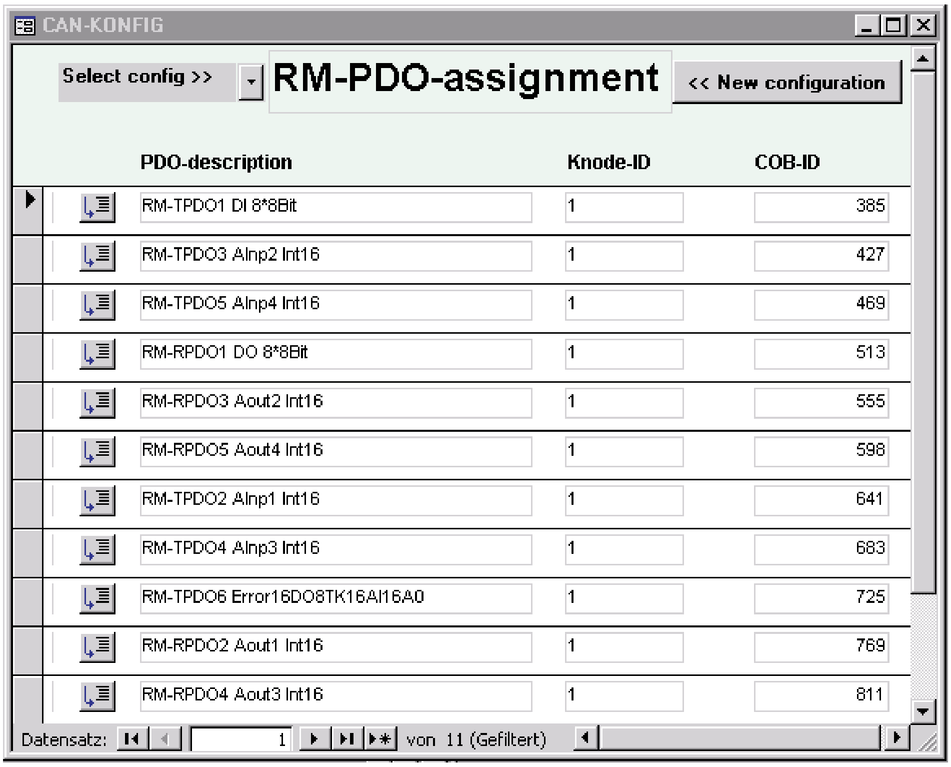
Max. 5 PDOs are sent dependent on the quantity of data connected to CSEND.
Max. 5 PDOs are received by KS 98-1.

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 (KS 98-1 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.



RM-PDO-assignment

PDO-description	Knode-ID	COB-ID
RM-TPDO1 DI 8*8Bit	1	385
RM-TPDO3 AInp2 Int16	1	427
RM-TPDO5 AInp4 Int16	1	469
RM-RPDO1 DO 8*8Bit	1	513
RM-RPDO3 Aout2 Int16	1	555
RM-RPDO5 Aout4 Int16	1	598
RM-TPDO2 AInp1 Int16	1	641
RM-TPDO4 AInp3 Int16	1	683
RM-TPDO6 Error16DO8TK16AI16A0	1	725
RM-RPDO2 Aout1 Int16	1	769
RM-RPDO4 Aout3 Int16	1	811

Datensatz: 1 von 11 (Gefiltert)

In block number range 21-40, max. 40 PDO addresses (COB-ID=Communication OBject Identifier: basic address + node address) can be addressed.

The data definition according to DS301 V4.0 complies with the Intel notation. The hearbeat protocol, which is offered by some manufacturers, is not supported.

☞ Recommendation for safe operation:

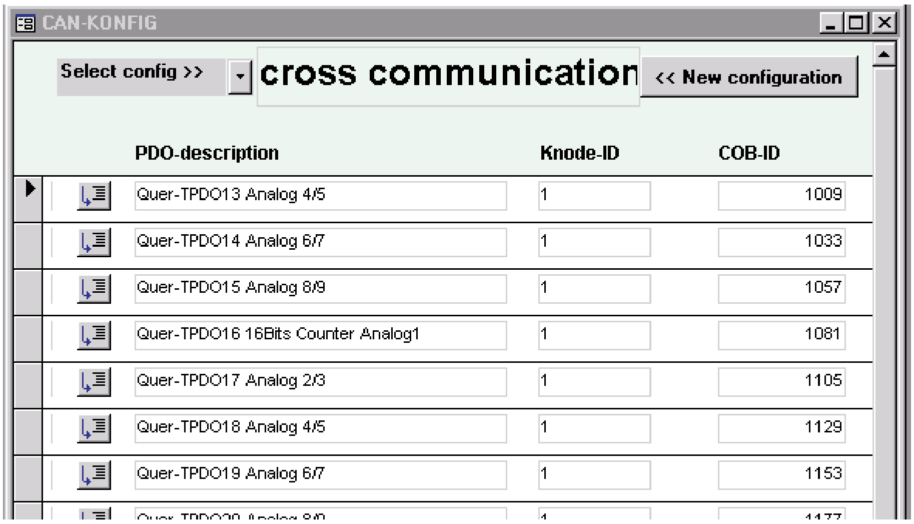
Bus load limitation

≤ 100 telegrams / 100 ms, Baudrate ³ 250 kbit/s = 250m distance

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

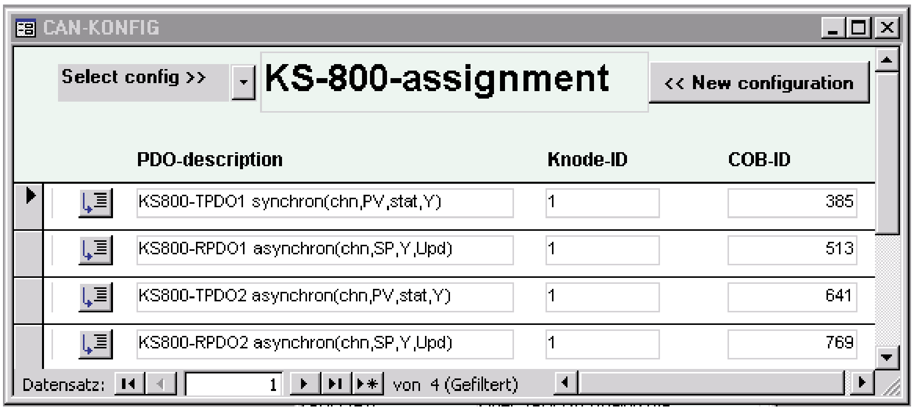
Send frequency for sensors ³ 100ms (inhibit time)

COB-ID allocation example for internal PMA CAN communication for node address 1:



cross communication

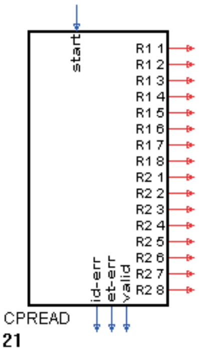
PDO-description	Knode-ID	COB-ID
Quer-TPDO13 Analog 4/5	1	1009
Quer-TPDO14 Analog 6/7	1	1033
Quer-TPDO15 Analog 8/9	1	1057
Quer-TPDO16 16Bits Counter Analog1	1	1081
Quer-TPDO17 Analog 2/3	1	1105
Quer-TPDO18 Analog 4/5	1	1129
Quer-TPDO19 Analog 6/7	1	1153
Quer-TPDO20 Analog 8/9	1	1177



KS-800-assignment

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

III-14.1 CPREAD (CAN-PDO read function (No. 88))



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.

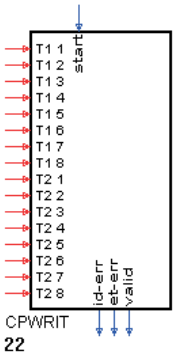
Digital inputs:	
start	The function is active with the input not connected, or if start=1 is connected.

Digital outputs:		
slotid	0	= module correct
	1	= module wrong
et-err	0	= no engineering error
	1	= no CAN-HW (KS 98-1 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:	
R11 ...R1 8	1st to 8th analog input value in byte format (8-bit) for COB-ID 1
R21...R28	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

III-14.2 CPWRIT (CAN-PDO write function (No. 89))



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.

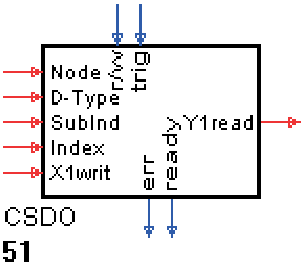
Digital inputs:	
start	The function is active, unless the input is connected, or if start=1 is connected.

Digital outputs:		
slotid	0	= module correct
	1	= module wrong
et-err	0	= no engineering error
	1	= no CAN-HW (KS 98-1 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:	
R1 1 ...R1 8	1st to 8th output value in byte format (8-bit) for COB-ID 1
R2 1...R2 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

III-14.3 CSDO (CAN-SDO function (No. 92))



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 KS 98-1 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 KS 98-1 in the background, there may be start-up collisions the consequences of which are removed only by restarting KS 98-1.

⚠ **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:	
r/w	access mode : 0 = read, 1 = write

Analog inputs:	
Node	decimal CAN-nodeaddress,1..42 (KS 98-1 is the CAN Object Identifier according to CiA DS301, node ID + 600H)
D-Type	datatype of the connected value, 0..6. Following datatypes 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
X1writ	data value (–29999 ... 999999)

Digital outputs:	
err	0 = no error
	1 = error detected.
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 (KS98-1 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)

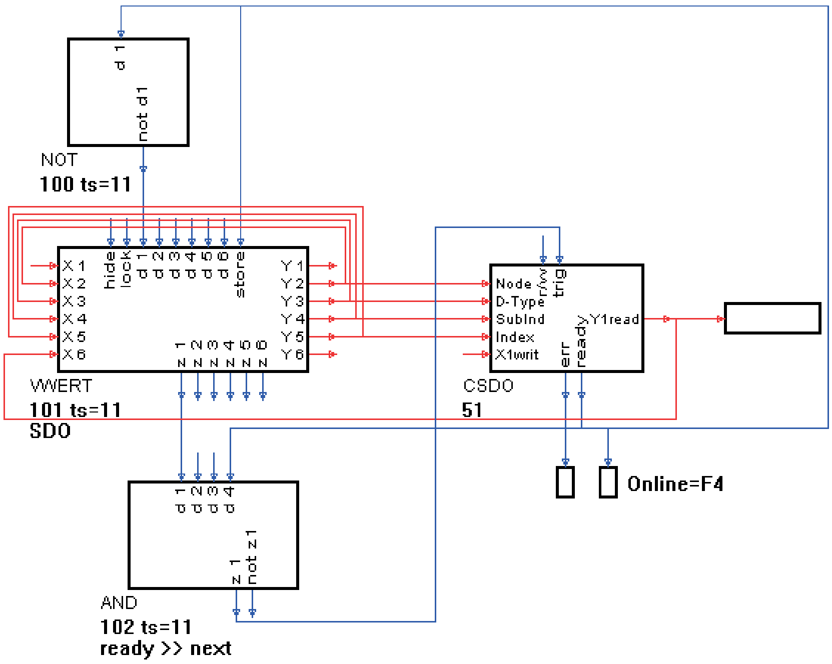
Possible errors (on output err):

- Faulty KS 98-1 hardware. KS98-1 CAN 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.

Engineering examples

SDO for data reading

Fig. 29

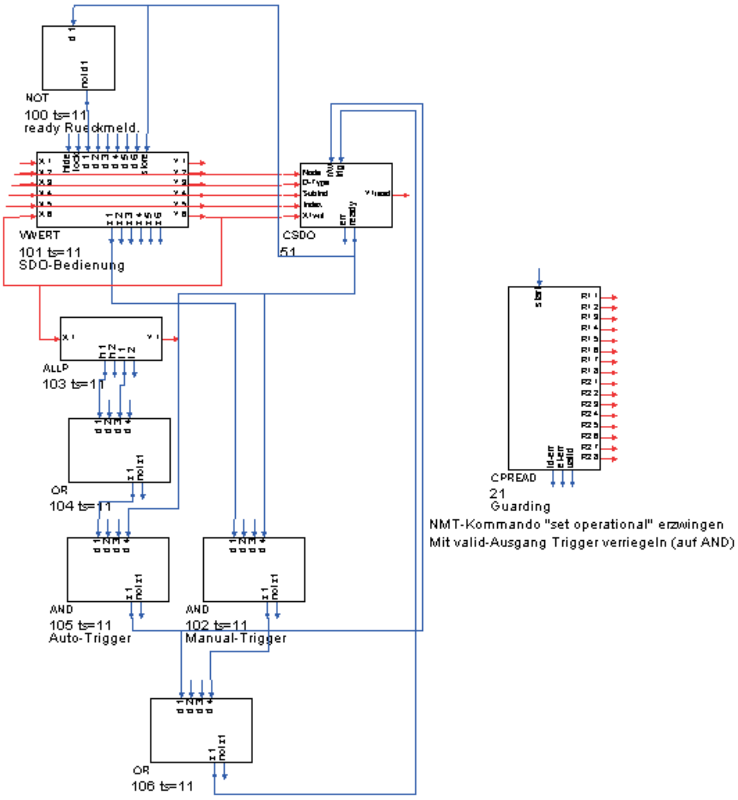


This example shows a possibility for data reading via an SDO access. Node address, data type, index and sub-index can be adjusted on an operating page. On the first line, a trigger bit which is reset by the following “ready” signal of the SDO block can be set. The engineering cannot be used to put a 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.

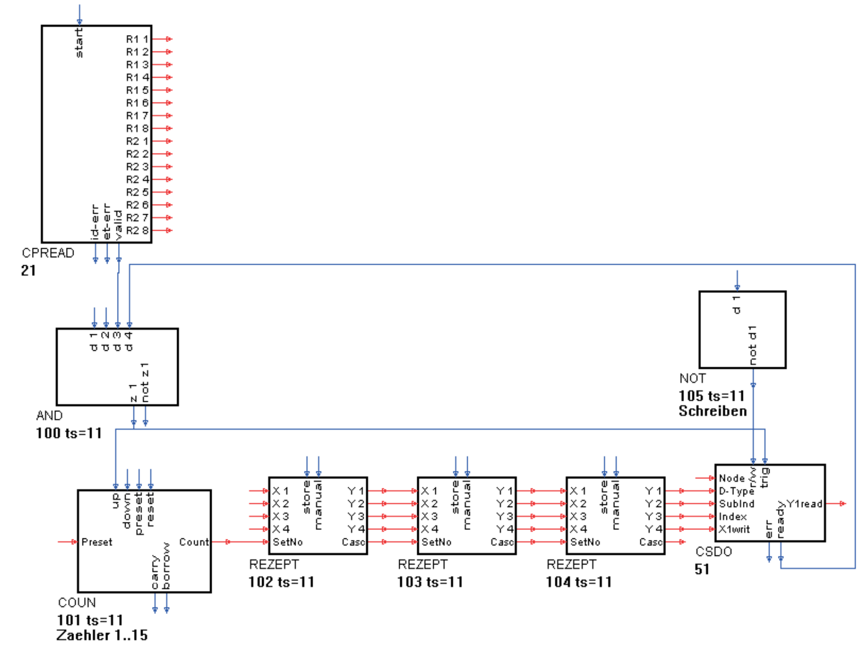
Fig. 30



Generating an SDO command sequence

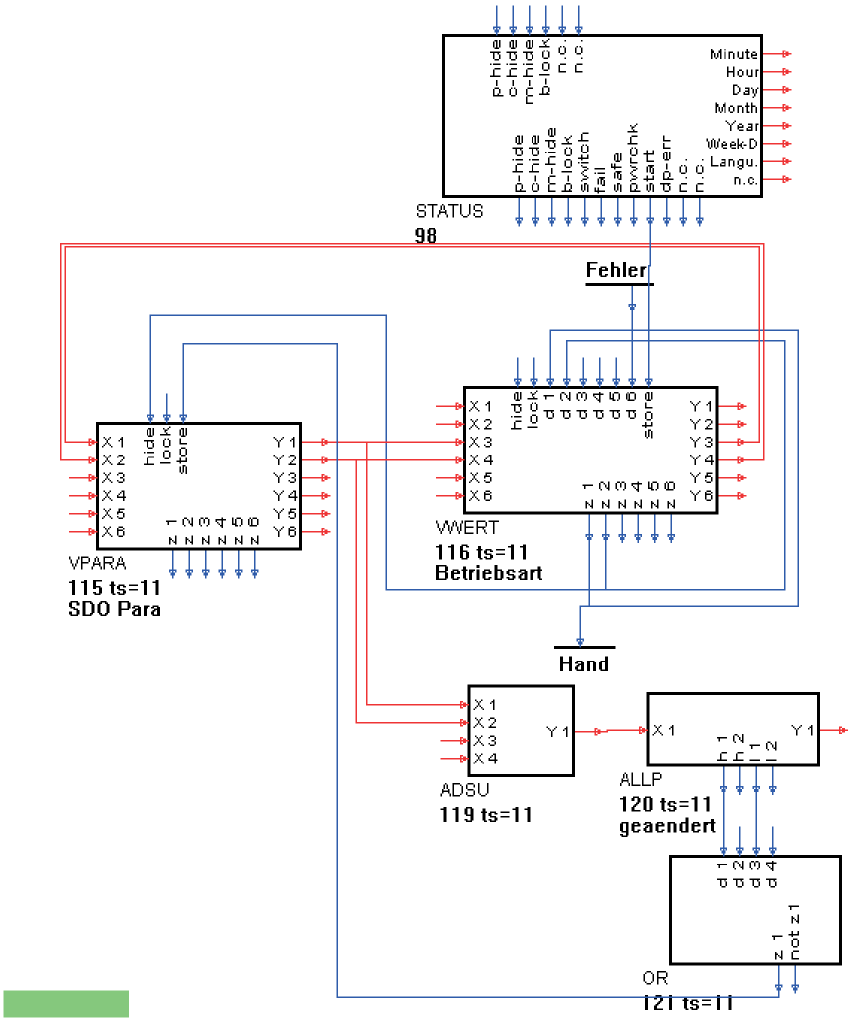
Engineering example SDO-SEQ.EDG shows the generation of an endless SDO command sequence. The values for D-type, sub-index, index and value are stored in the recipe blocks. The counter (COUN) counts from 1 to 15 continuously.

Fig. 31



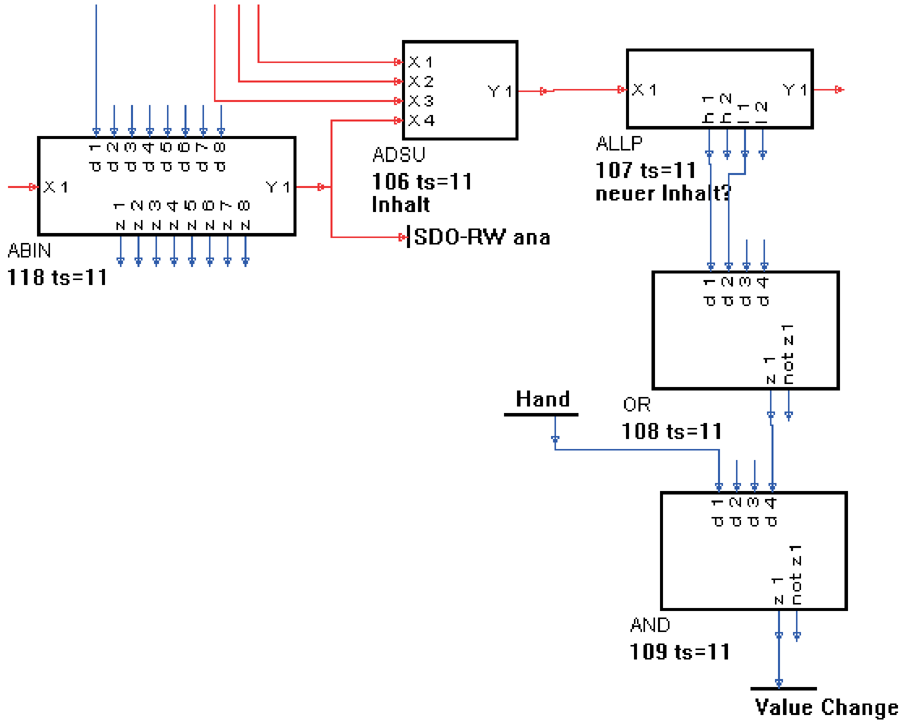
An extended engineering for advanced users SDO-SEQ2.EDG shows further functions and possibilities of KS 98-1 engineerings in conjunction with command sequences.

Fig. 32



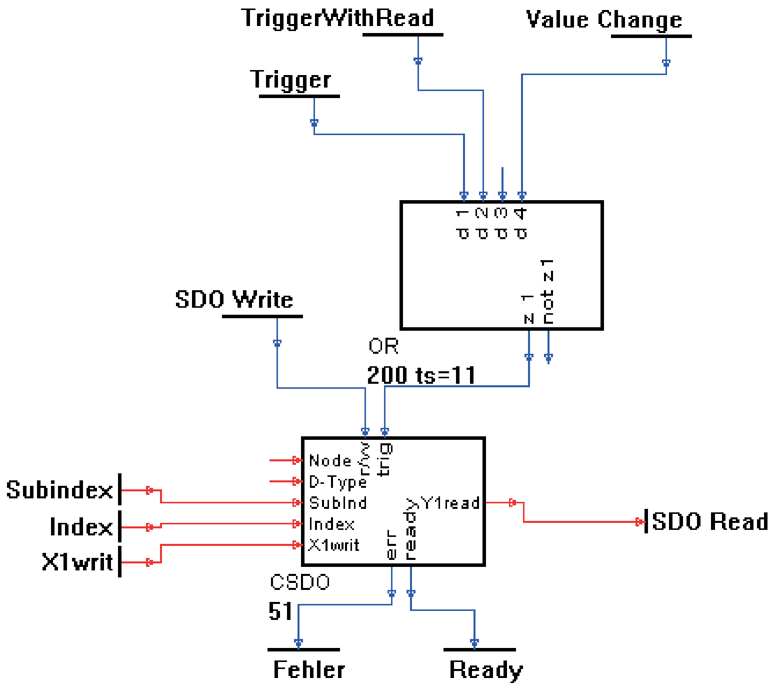
This partial engineering shows the possibility of access to SDO block parameters via an operating page.

Fig. 33



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

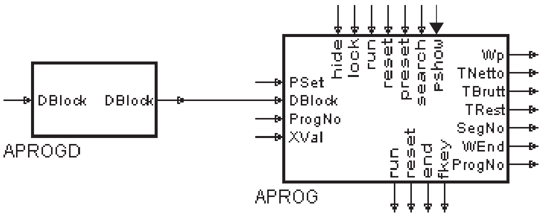
Fig. 34

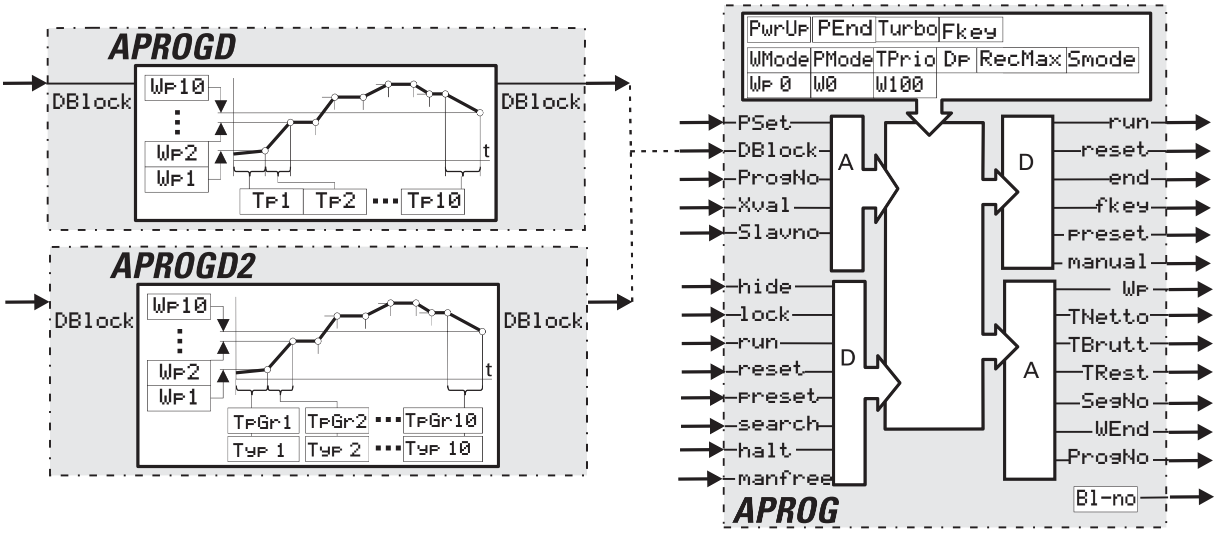


Command triggering is subject to various conditions: when reading, after changing during manual mode and cyclically in automatic mode.

III-15 Programmer

III-15.1 APROG (analog programmer (No. 24)) / APROGD (APROG data (No. 25))





General

An analog programmer comprises a programmer (APROG) and min. one data block (APROD or APROGD2), whereby output DBlock of the APROGD/APROGD2 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. APROGD and APROGD2 may not be mixed within a recipe.

Limiting is only in the number of available block numbers and in the calculation time.

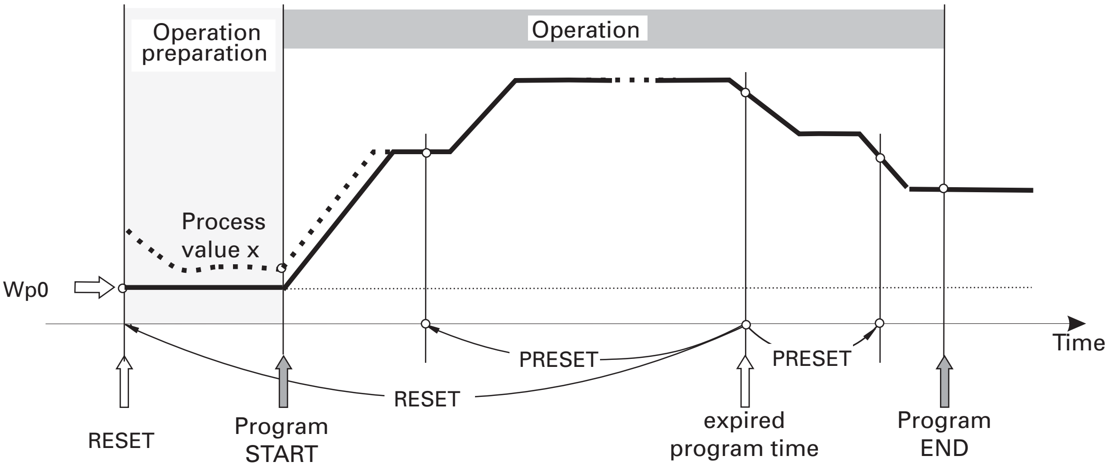
The data block (APROGD or APROGD2) has an analog output, at which the own block number is made available.

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.

Fig. 35



APROG

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)	
p-show	Program editing enabled	
halt	Program interruption (e.g. due to an exceeded bandwidth detected outside the programmer). 0 = program not halted 1 = program halted	
manfree	Manual mode disabled 0 = switchover to manual mode not permitted 1 = switchover to manual mode is permitted	

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 ☺ key / interface function `fkey' (:pressing key ☺ causes switch-over (0 or 1))
preset	This output indicates a programmer preset operation. With a single preset command, a pulse is output for the duration of one cycle (dependent on the time slot in which the programmer is classified). If the programmer is held in preset continuously, this output is always active. 0 = no preset status 1 = APROG stands in preset status
manual	This output indicates the progrmmer manual mode. 0 = APROG works in automatic mode 1 = APROG works in manual mode

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
SlavNo	SlavNo: Block number of a connected slave block (for coupling master and slave blocks (APROG or DPROG)

Analog outputs (APROG):	
Wp	Programmer
TNetto	Net program time (Σ Trun)
TBrutt	Gross program time (Σ Trun + Σ Tstop)
TRest	Programmer rest time
SegNo	Actual segment number
WEnd	End value of actual segment
ProgNo	Actual program number (recipe)
SegRest	Remaining segment time
Bl-no	Own block-number (e.g. for coupling master and slave blocks)

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	
Dp	Decimals for setpoint		0..3	3
RecMax	Max.recipees		1..99	99
Smode	Smode (search mode):	0 = search run in segment 1 = search run in program/section 2 = no search run		
Wp0	Program at reset		-29999 ... 999 999	0
W0	Untere Sollwertgrenze		-29 999 ...999 999	-29 999
W100	Obere Sollwertgrenze		-29 999 ...999 999	999 999

APROGD

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

Analog outputs (APROGD):	
DBlock	Own block number

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 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 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 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.

APROGD2

Analog inputs (APROGD2):	
DBlock	Block number of cascaded data function `APROGD´

Analog outputs (APROGD2):	
DBlock	Own block number

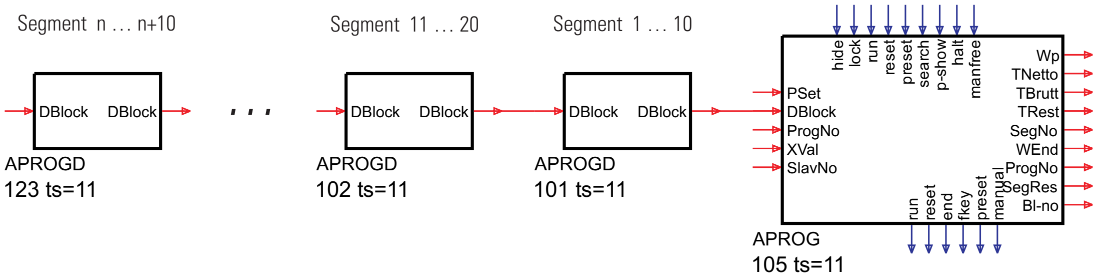
Parameter APROGD2	Description	Value Range		Default	
		ET	Unit	ET	Unit
Type 1	(Type for segment 1)	0	Time segment	←	←
		1	Gradient segment		
		2	Haltesegment		
		3	Sprungsegment		
		4	Time segment and wait at the end		
		5	Gradient segment and wait at the end		
		6	Hold segment and wait at the end		
		7	Step change segment and wait at the end		
Typ10	Type for segment 10	like type 1 0 .. 7	...		
TpGr1	Time or gradient for segment 1	0 ... 59,999	0 ... 999:59	OFF	—:—
Wp1	Final value for segment 1	-29999 ... 999999	-29999 ... 999999	0	
. . .					
TpGr10	Time or gradient for segment 10	0...59,999	0 ... 999:59	OFF	—:—
Wp10	Final value for segment 10	-29999 ... 999999	-29999 ... 999999	0	0

Configuration APROGD2	Description		Value
PwrUp	Behaviour after mains recovery	Continue program (default)	Prog.Fort
		Search run in the actual segment	Fort.Seg.
		Continue at actual time	Fort.Zeit
PEnd	Behaviour at program end PEnd: 0 = Stop 1 = Reset 2 = Reset + Stop (End-condition is reset with stop)	stop after program end (default)	Stop
		Reset after program end	Reset
Turbo	Time unit	Time = hours : minutes (default)	Std:Min
		Time = minutes : seconds	Min:Sek
FKey		FKey (⊡ key function): 0 = fkey output status switchover by ⊡ key (previous function) 1 = ⊡ key used to generate a pulse at the fkey output (pulse length = 1 cycle) 2 = ⊡ key controls the programmer (fkey output generates a pulse when actuating the key, pulse length = 1 cycle)	

Cascading

Cascading APROGD/APROD2 function blocks permits realization of a programmer with any number of segments. The segment sequence is dependent of the APROGD/APROD2 function block wiring (→ see below). The block numbers are without signification related to the order. Segment parameters from right to left in the data blocks

Fig. 36 Example of an analog programmer with n segments

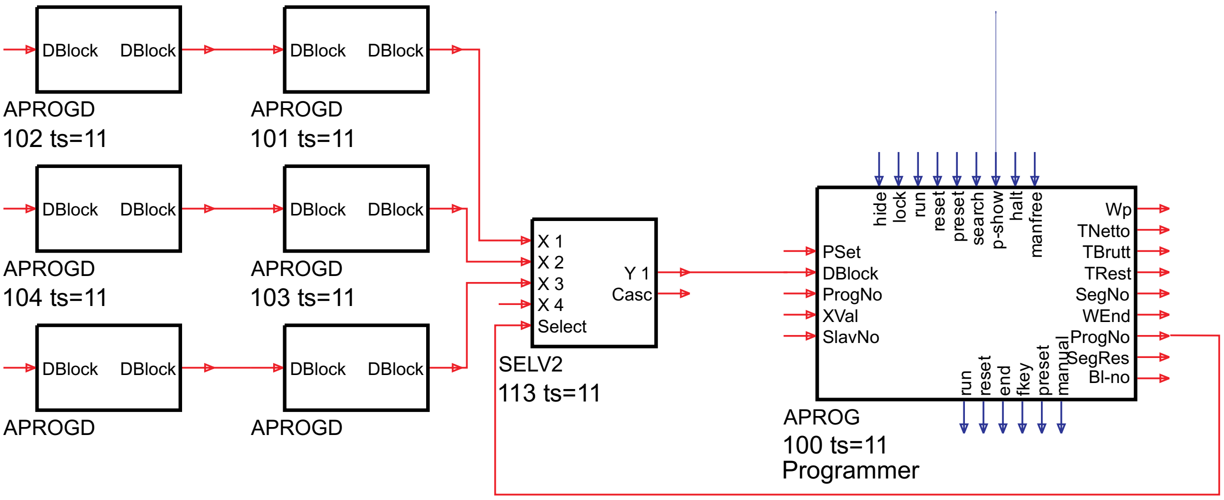


Recipes

Analog output 'ProgNo', at which the actual recipe number is output, and one or several SELV2 function blocks can be used to select a recipe the block number of which is switched to the APROG input (→ see below40). 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.

Fig. 37 Example of an analog programmer with 3 recipes à 20 segments



Max. 800 ms after switching over, the block number of the first parameter block of a new recipe must be applied to the Dblock input. In a cascade, the SELV2 blocks must be arranged in ascending order.

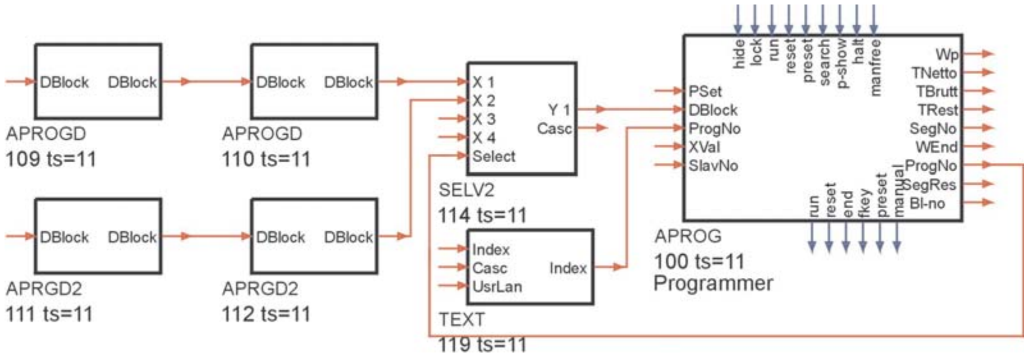
Recipe changing - program selection

Whilst a program is active, switching over to a different recipe on the programmer operating page is not possible. Recipe changing is possible only in reset status!

Recipe names

By linking TEXT blocks to the ProgNo input, display of recipe names rather than of recipe numbers is possible.

Fig. 38 Recipe names



This procedure can be used for both internal and external recipe selection. With external recipe selection, the required recipe number must be applied to the index input of the TEXT block next to the APROG block (block 102 in this example). This recipe number is fed to the ProgNo input of the programmer. With internal recipe selection (via operation or level 1 interface data), the index input of the text block need not be connected.

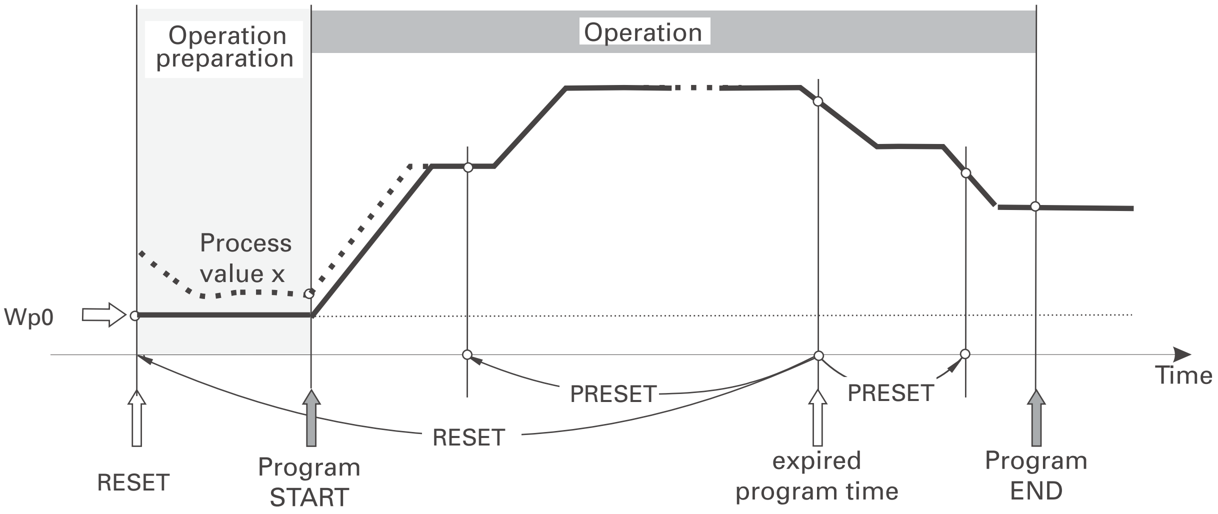
Operation preparation and end position

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 of the first segment is activated immediately.

At program end, either

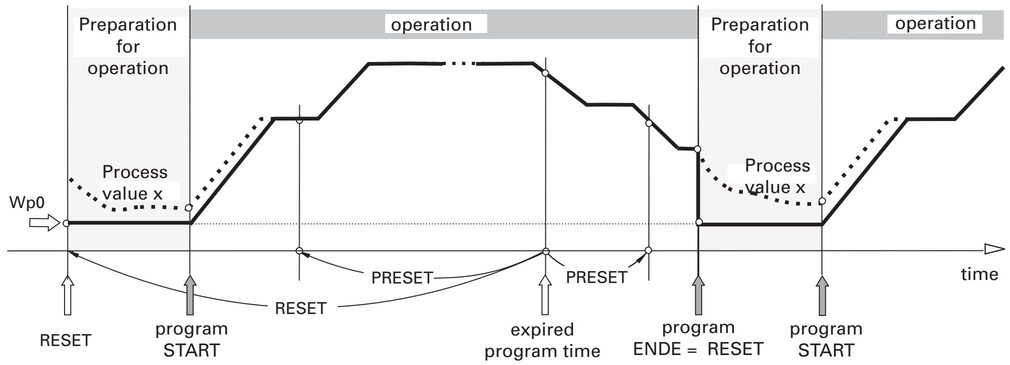
- 0= Stop: the of the last segment is maintained (→ see below),

Fig.: 39 Profile with stop at end position



- 1=reset:or the programmer goes to rest position WP0 (→ see below) and restarts automatically, if run is still valid.

Fig. 41 Profile with automatic reset at program end



- 2 = reset + stop: Programmer goes to rest condition WP0

At program end, the number of the last segment increased by 1 is output as active segment number (SegNo output of operating page and interface). This is required to bring the slave block to the end status safely with a segment preset.

Start

The programmer uses a common start Wp0 for all programs. However, using an individual start for a recipe is possible as follows:

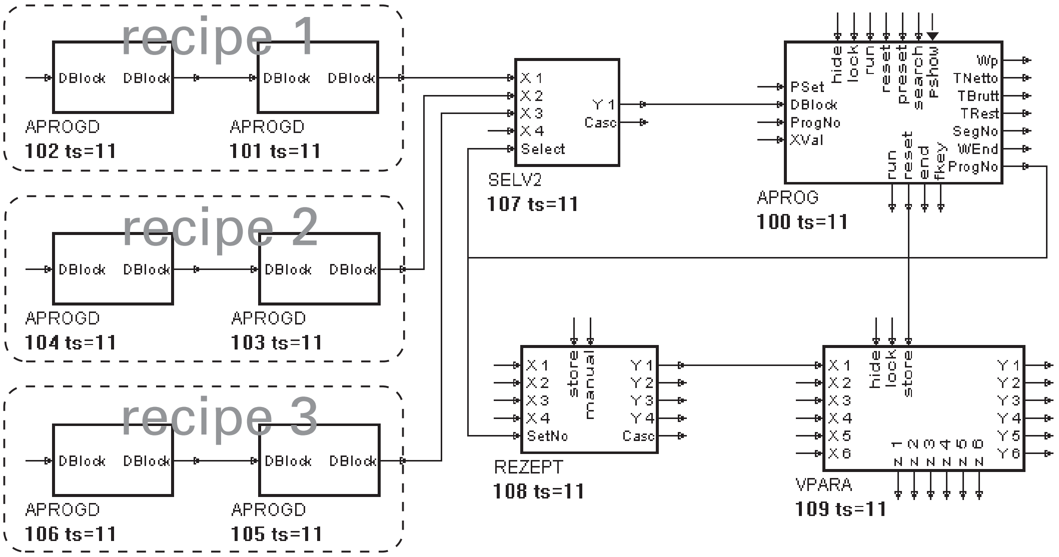
Use the of the 1st segment of each program as start value.

Set the relevant segment time (Tp1) to 0.

Set the search run parameter Smode to ‘Search run in program sections‘. Now, the search run is not limited to the 1st segment and program start at the process value in the 2nd segment is possible (see search run, page -> marking(SS)).

If each recipe shall have a separate reset (Wp0), function blocks REZEPT and VPARA can be used as shown up 40. Note the calculation order (APROG ⟶ REZEPT ⟶ VPARA).

Fig. 40 Recipees with separate starting setpionts



Halt status

Used e.g. for bandwidth monitoring

The halt status can be switched on and off only via the halt control input. Unlike the stop status, the run status remains unchanged (run output remains active) in the halt status.

Status display is “halt”.

Automatic/manual operation

The programmer can operate both in automatic or in manual mode:

automatic: The effective is determined by the programmer.

manual: The effective can be altered via programmer operating page or via interface. However, the program continues running and can be influenced via control inputs and operation/interface as during automatic mode (run/stop/reset/preset/search).

- When switching over from automatic ⇒ manual, the effective remains at the last value which was effective before switching over.
- Manual ⇒ automatic switchover: The effective changes from the manual to the actual programmer .
- Switching over can be done via the programmer operating page (“automatic” < >. “manual”) or via interface.
- The automatic/manual mode displayed only via the digital manual output.
  0 = automatic
  1 = manual
- The “manfree” control input can be used to enable switchover.
  0 = switchover to manual is disabled
  1 = switchover to manual is enabled

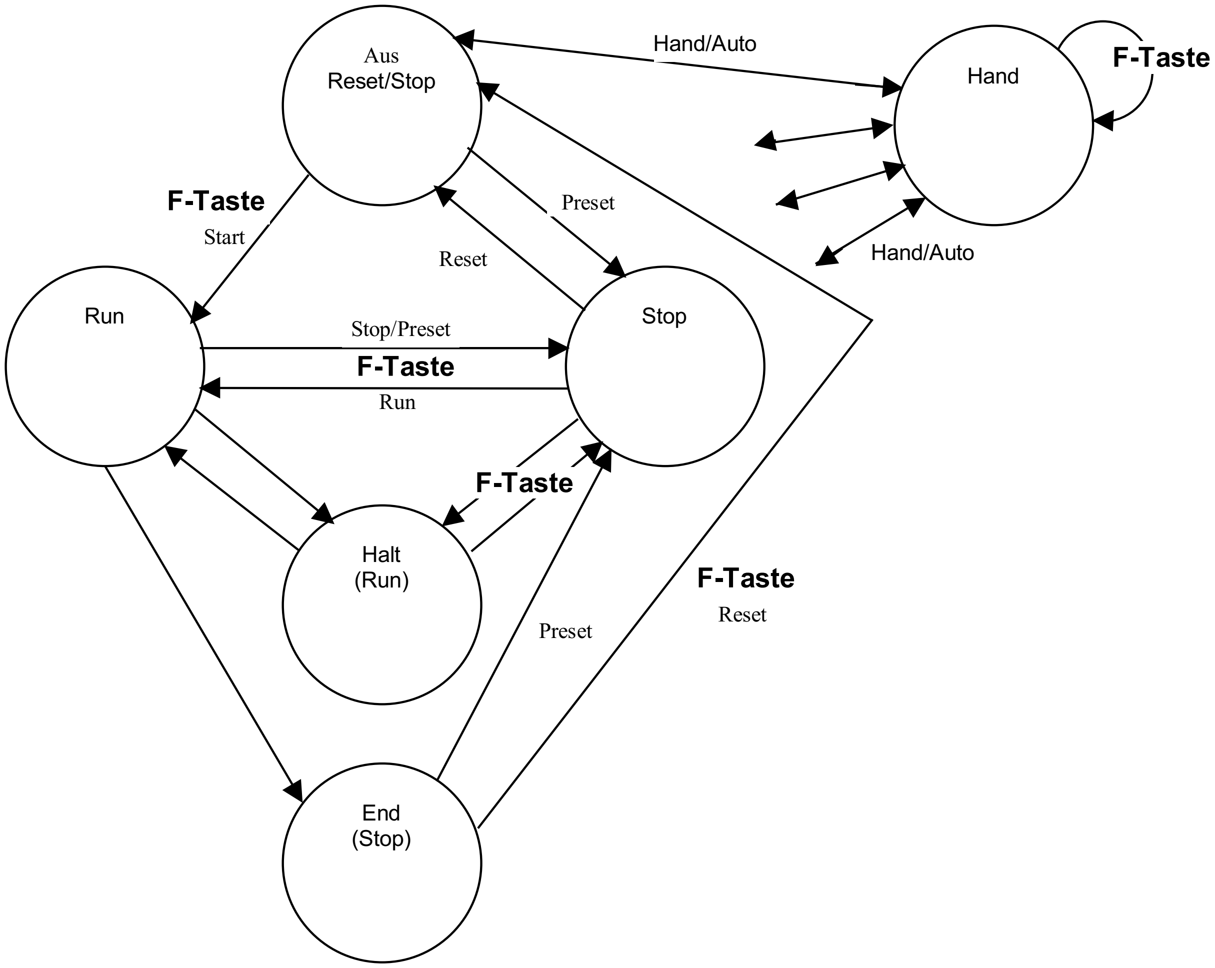
Programmer control via key

Programmer control is possible by means of digital function block inputs, status changing on the operating page, interface and also using key . For selection of the key functionality, a configuration parameter is offered:

FKey: 0 = toggle bit changes at each key pressure at output fkey
1 = F key function with pulse at output fkey
2 = F key controls the programmer (fkey output generates a pulse when a key is actuated)

Hereby, the rule that the statuses at connected control inputs have priority over the operation is applicable. The following diagram describes the status sequence dependent on the actions:

Fig. 42 Statusdiagram of the programmer and the effect of the function key



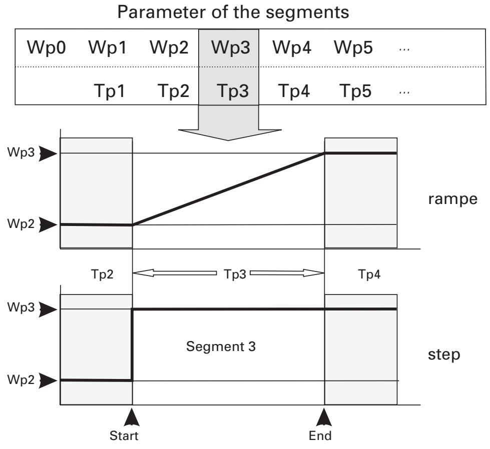
Change mode (ramp/step)

If the shall change in a step or ramp is determined by a parameter (Wmode) valid for all segments of a recipe (default: ramp).

Ramp: The 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 goes to value Wp immediately at segment start and maintains it during segment time Tp.

Fig. 43



Segment types

Different segment types for each individual segment can be determined separately in data block APRGD2.

Like the APROGD, the APRGD2 block contains the parameter for 10 segments. Apart from parameters and time, APRGD2 has the segment type as a third parameter. I.e. the following basic segment types are possible within a program:

- Time segment with target and segment time
- Gradient segment with target and gradient
- Hold segment with hold time
- Step change segment with and segment time

Two variants of all segment types are available: with and without wait status at the segment end.

Segments with wait status feature particularities which must be taken into account:

- A segment of this type does not limit the search run over several segments (see search run on page -> Page 211).
- Behaviour after a short power failure (≤ 0.5 Std.) with configuration PwrUp = 2 (continue at actual time):
  If the program time from power failure to power recovery includes min. one segment with wait status at the end, the search run in the segment in which the program would be without power failure is omitted and the program stops at the first wait status without search run.
- As with the APROGD block, the recipe end is determined by switching off a time parameter (TP=off) or by a DBLOCK input which is not connected any more.
- When connecting the APRGD2 block to the DBLOCK input of APROG, the new segment types are used automatically. The setting of parameter WMode is ignored when using the APRGD2 block.
- Mixing APROGD and APRGD2 blocks in a recipe is not permitted. However, a programmer can be operated with both data block types as long as only one parameter block type per recipe is used.

Program sequence changes

During the running program, s 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 TF 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:

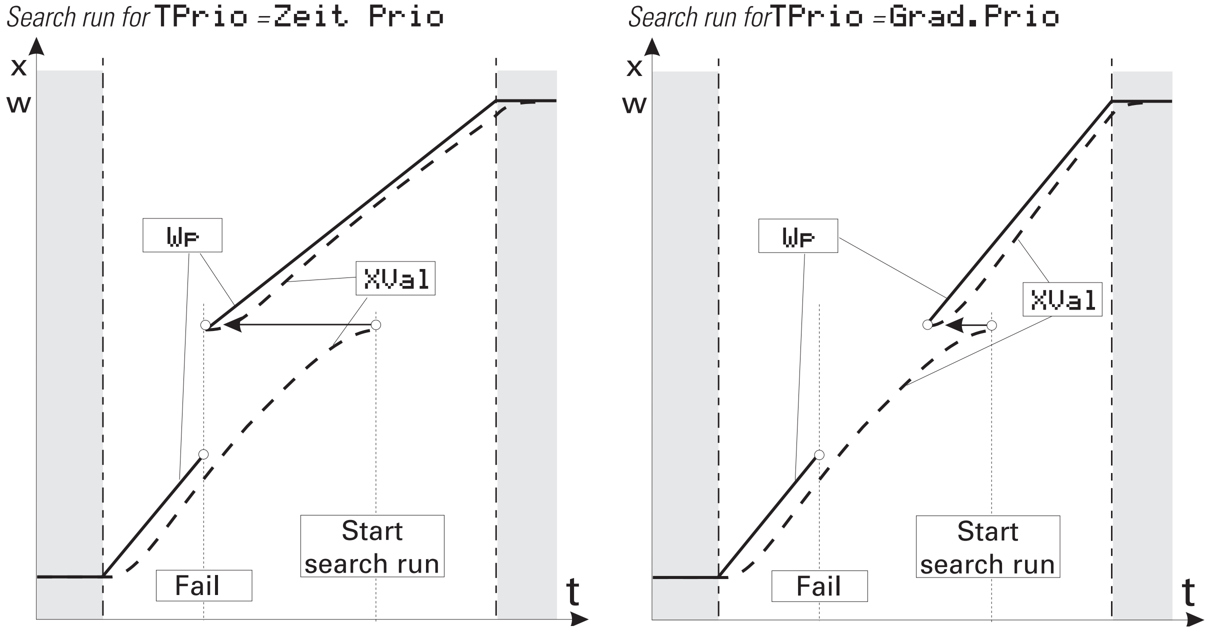
- Start via operation
- Start via interface
- Start with search = 1
- Program start after Reset
- After short power failure with PowerUp = Cont.seg. or Cont.time

Search run in program segment

When starting the search run, 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 (segment without gradient; holding time), the program is continued at the segment start.

With a step segment (including the APROGD segments with WMode = 1 (step)), the relevant target setpoint is always used from the segment start.


Search run for TPrio = Zeit Prio

Search run for TPrio = Grad.Prio

Search run in program section

Apart from the search run in the actual segment as described above, searching over several segments or search run switch-off are possible. The various search run functions can be selected via parameter SMode:

- 0 = search run in segment
- 1 = search run in program section
- 2 = no search run

Searching is limited to a section of several segments which have the same polarity sign as their gradient. Hereby, a hold segment is neutral ⇒ no change of polarity sign.

As the throughput times with a search run dependent on the number of segments may be very long, searching is limited to several time slots so that searching is done only in one segment per time slot.

A search run is done in the following cases :

- Search run at program start: search over several segments up to the next gradient change
- search run started via control input, interface or via operation: forward and backward search from the actual program point to the next gradient change
- Search run after power failure at PwrUp = 1: forward and backward search from the point of failure to the next gradient change
- Search run after power failure at PwrUp = 2: forward and backward search from the program time in which the program would be without power failure until the next gradient change
- Search run in the hold segment (Gradient =0): A search run is done only if there is min. one further segment (except hold segment) in this section. With a further hold segment directly before or behind this segment, the search run takes place only in the actual segment.

☞ Search run at TPrio = 1 (time priority): The search run is limited to the actual segment, i.e. the runs from the actual process value to the segment end value within the actual remaining segment time.

☞ Segments with wait status at the end do not limit the search range, exept it's the search run after voltage failure!

☞ A search run may lead to program termination.

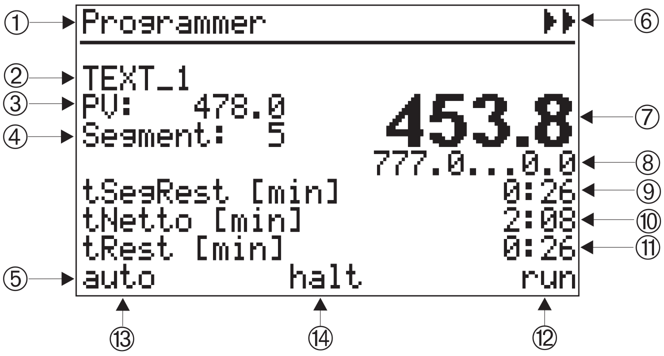
Analog programmer operating page

Analog programmer APROG has an operating page, which can be selected in the operating page menu with input 'hide' not connected.

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.

① Name of programmer block
② Recipe name:
③ **Process value**
④ **Segment number**
⑤ **Status (r/w) automatic/manual**
⑥ Block change
⑦
⑧ **Segment start and end value**
⑨ **Remaining segment time**
⑩ Program net time
⑪ Remaining program time
⑫ Status (r/w) stop, run reset, search, program, quit, error
⑬ Status (r) halt, end

Fig. 44



```
Programmer              ▸▸
TEXT_1
PV:    478.0
Segment:   5        453.8
               777.0...0.0
tSegRest [min]        0:26
tNetto [min]          2:08
tRest [min]           0:26
auto        halt       run
```

Run, reset, preset and search are concerned, see following table:

Input field		Operation	Display	FB input
Header		Selection of slave block	Display of slave data	- : -
auto/manual		Operating mode selection	auto or manual	- : -
setpoint		Automatic: programmer , Manual mode: operator setting in input field	Active	- : -
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
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	

Notes related to the properties of the operating page

The display fields shown in **bold** and **underlined** letters on the picture above mark the elements which are switched over when changing to a slave page (see sections Master/slave operation on page 215).

The remaining fields continue indicating statuses and values of the master page.

Recipe name:

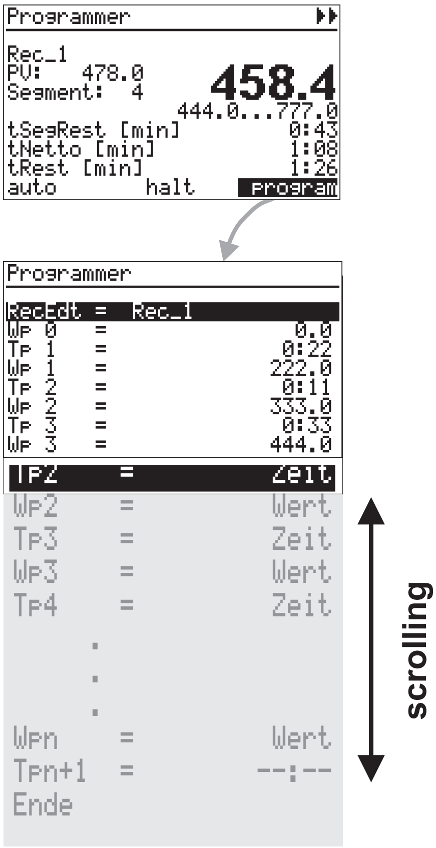
Recipes can be selected in the reset status. Unless a user text (TEXT block at ProgNo input) is provided, 'Rec n' is displayed (n stands for the current recipe number).

- The process value is visible only with the process value input connected.
- The segment number is adjustable only with preset to segment.
- The can be adjusted only during manual mode.
- Display of the remaining segment time is suppressed with preset to segment (e.g. with digital slave programmers).
- The program net time is adjustable with preset to time.
- 3 statuses are displayed (partly adjustable, dependent on operating status):
  - Status left: automatic / manual (adjustable)
  - Status middle: halt / end (unless one of the two statuses is active, this display is omitted)
  - Status right: stop / run / reset / search / program / quit / error

Direct programmer adjustment

Program s 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.

Fig. 46



```
Programmer                 ▸▸
Rec_1
PV:    478.0
Segment:  4          458.4
               444.0...777.0
tSegRest [min]         0:43
tNetto [min]           1:08
tRest [min]            1:26
auto       halt     program
```

```
Programmer
RecEdt =  Rec_1
Wp 0   =              0.0
Tp 1   =             0:22
Wp 1   =            222.0
Tp 2   =             0:11
Wp 2   =            333.0
Tp 3   =             0:33
Wp 3   =            444.0
Tp2    =             Zeit
Wp2    =             Wert
Tp3    =             Zeit
Wp3    =             Wert
Tp4    =             Zeit
         .
         .
         .
Wpn    =             Wert
Tpn+1  =            --:--
Ende
```

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, APROGD2, 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 (Fig. 45).

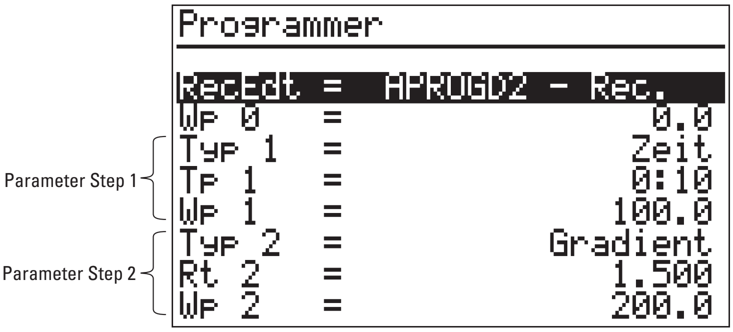
See chapter → page 38

The programmer adjustment via the main menu parameter level remains possible. In this case, however, each data block APROGD, APRGD2 or DPROGD must be selected separately. But parameters W0, W100 (adjustment limits) and Dp (decimal points) which pertain to the APROG are not effective when entering.

Recipe names used via text blocks are displayed also on the editing page. By changing the recipe name, switching over to the display of another recipe is possible. This can be done at any time and does not cause changing of the active recipe.

When using APRGD2 blocks, the following editing page is displayed.

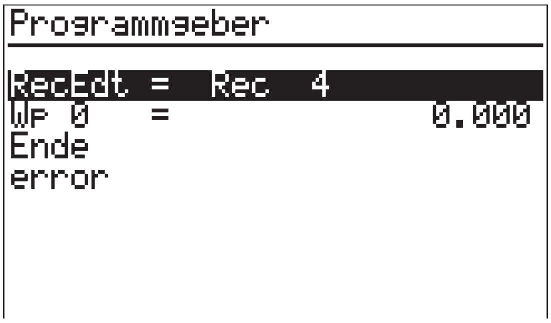
Fig. 45



```
Programmer
RecEdt =  APROGD2 - Rec.
Wp 0   =              0.0
Typ 1  =             Zeit
Tp 1   =             0:10
Wp 1   =            100.0
Typ 2  =         Gradient
Rt 2   =            1.500
Wp 2   =            200.0
```

If an X input of recipe switchover block SELV2 is not connected and the relevant recipe is selected nevertheless (should be disabled via the adjustment range of the recipe number), the following error display is output:

Fig. 48

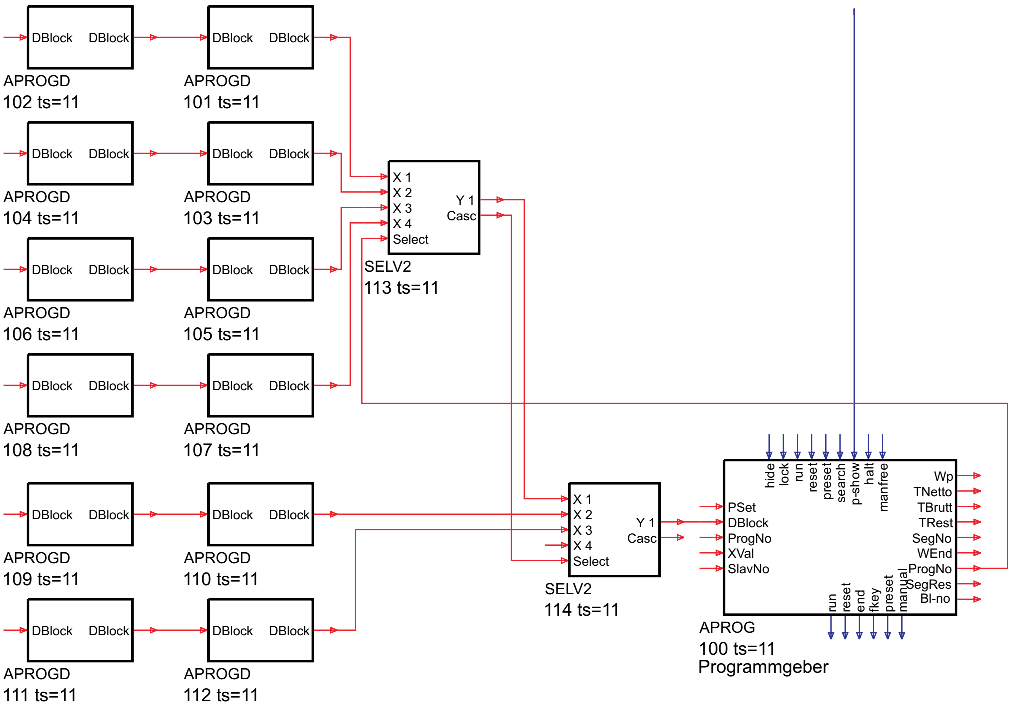


```
Programmgeber
RecEdt =  Rec   4
Wp 0   =            0.000
Ende
error
```

Access to parameters of inactive recipes

To enable the access to all recipes relevant for this programmer block from the programmer program edit page (including the inactive ones), the following wiring principle is compulsory:

Fig. 47



The SELV2 block switches the parameter block number onto the programmer Dblock input. Via the SELV2 block structure information, the programmer has access to all recipes.

Unless wiring is via SELV2, switchover to a different recipe on the parameter setting page is not possible, i.e. the recipe cannot be displayed.

☞ For active recipe switchover during reset status, another wiring type can be selected as well; however, at the latest 800 ms after switching over, the block number of the first parameter block of a new recipe must be applied safely at the Dblock input. The order of SELV2 block numbers plays an important part, in particular, if these blocks were assigned to the 800ms time slot. Unless the order is ascending, there will be an additional 800 ms delay at each cascade step.

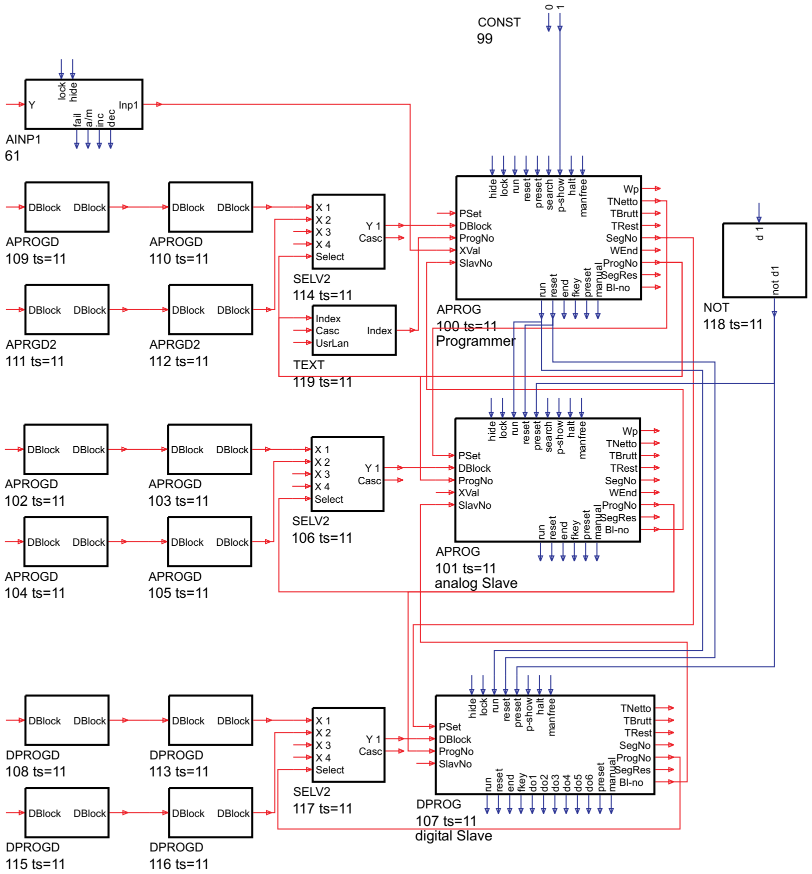
Master/slave operation

Programmers frequently consist of several coupled blocks which have a common time or segment structure (e.g. master block: oven temperature, 1st slave block atmosphere/C level, 2nd slave block 1..6 digital control signals). Such a programmer is provided with an operating page extending to all blocks in KS98-1. On the master operating page, the slave block data can be displayed via symbol ▶▶ .

Wiring

Synchronization of several programmers is by slave block preset coupling. The slave blocks are forced to the same time or to the same segment number via time or segment preset by the master.

Fig. 49 Programmer with two analog and one digital blocks



To facilitate operation of the coupled blocks, the programmer has a SlavNo input and a Bl-no output. These are used by the slave block for transfer of the block number to the following block (see Fig 49). The block the Blo-no output of which is not connected (block 100 in this example) should operate as master. Its Tnetto or SegNo output is wired to the PSet input of further blocks.

By linking the blocks (Bl-no > SlavNo), a programmer with coupled blocks is created. From the master block operating page, access to the data (incl. parameters) relevant for the slave for display or adjustment is possible easily.

Operation of a programmer with several blocks

Calling up a master block operating page via the operating page menu (page survey):
When selecting the operating page of a master programmer via the operating page menu with master/slave wiring via Bl-no > SlavNo as described above, changing between the relevant programmer blocks can be done easily via the -symbol ▶▶. The order is determined by the wiring order (in the above example: 100 ⇒ 101 ⇒ 102 ⇒ 100 ⇒ ...).

However, changing to the next programmer block is not complete. Only some of the values and texts (e.g. title) relevant for the next block are displayed. The remaining elements continue indicating only the master information (see analog programmer operating page).

In case a change to the operating page menu (page survey) is made in this condition of the operating page, the block selection remains unchanged. I.e. when calling up the master block again later, the data of the slave block indicated last are displayed.

Display information which is firmly assigned to the master block:

Recipe name (can be switched over in reset state)
Program net time (adjustable for Preset to time)
Remaining program time
Status display for halt/end
Status display for stop/run/reset/search/program/quit/error (adjustable)

Display information of the actual programmer (master or slave):

Programmer block name
Process value
Segment number (adjustable for preset to segment only with master)
Actual or actual control bits (both adjustable in manual mode)
Segment start and end value
Remaining segment time
Status display for automatic/manual (adjustable, if permitted via manfree input)

As only the master shall decide upon the change of an active recipe, the wiring structure must be of such kind that changing is effective also for all relevant slave blocks (master ProgNo output ⇒ slave ProgNo input, see above diagram). This kind of master/slave coupling permits only a central recipe change for all blocks coupled accordingly.

Calling up a slave block operating page via the operating page menu (page survey):
When calling up the operating page of a slave programmer via the operating page menu, display of symbol ▶▶ is suppressed. Easy changing to other blocks connected via the Bl-no > SlavNo coupling as described above is not possible. Moreover, no data of the connected master are displayed.

To prevent inadmissible adjustments (recipe selection, run/stop/reset) from being offered on a called up operating page,outputs ProgNo, run and reset of the master programmer should be wired to the relevant slave programmer inputs. For plant operation, slave programmer display is suppressed purposefully with hide=1 when the page survey is active (PageNo at status block = 0).

Subordinated parameter page (program editing page):
On the subordinated parameter page, recipe switchover is possible at any time. But switchover acts only on the recipe parameter display on this page rather than changing to the effective recipe.

Changing directly to the parameters of the next programmer is not possible. For this, the superordinate operating page of the corresponding programmer must be used.

Programmers without coupling:

On an operating page with a function block which is not connected with other programmer blocks via Bl-no > SlavNo coupling, display of symbol ▶▶is suppressed.

Remaining segment time

The remaining time of the actual segment is displayed on the operating page.

It is:

- readable via interface
- available as an additional analog output signal
- always 0 with reset
- suppressed with "preset to segment"

Incompatibility to earlier KS 98 functionality

Recipe switchover:
KS 98: The receipe number can be switched over at any time on the programmer operating page. However, the new selected recipe will be effective only after the next reset. On the subordinated parameter page, switching over has the same effect.

KS 98-1: On the programmer operating page, the recipe number can be switched over only during reset status. Switchover is effective immediately. On the subordinated operating page, switching over continues being possible at any time. Nevertheless, only the recipe to be displayed with its parameters is switched over. The instantaneously active recipe remains unaffected.

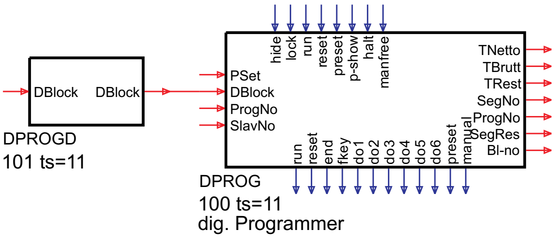
End behaviour with PEnd = ’stop‘:
KS 98: Program is at the end, status is ’run‘, reset command leads to an immediate re-start

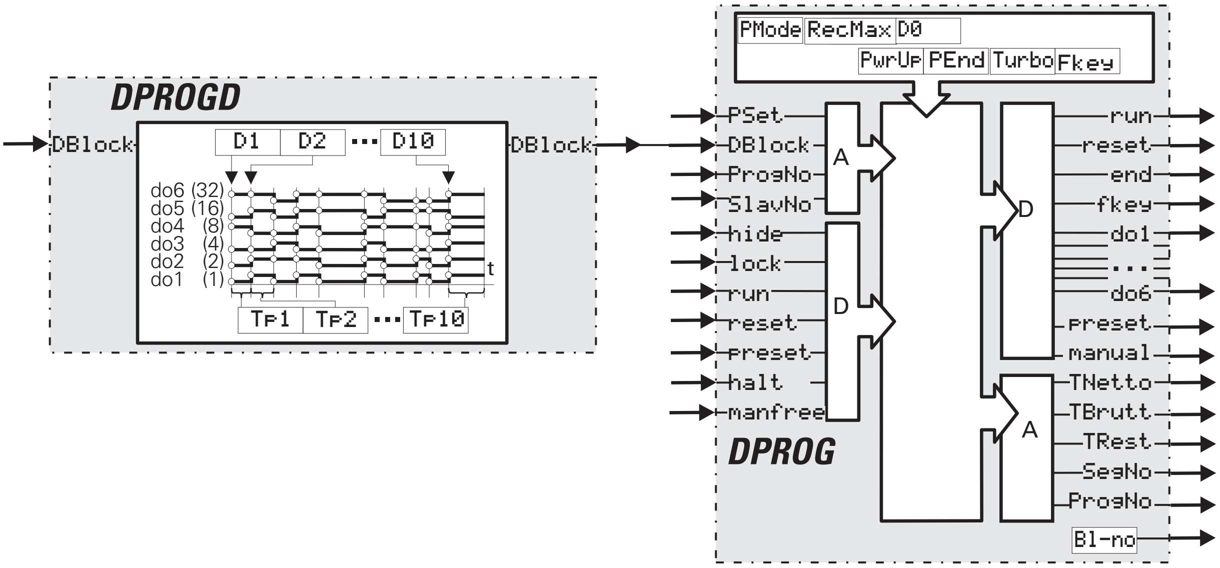
KS 98-1: Program stands at the end, status is ’stop‘, the program remains in reset status after a reset command.

Segment number at program end:
KS 98: At program end, the number of the last segment is displayed as segment number (SegNo output, operating page, interface).

KS 98-1: At program end, the number of the last segment + 1 is displayed as segment number (SegNo output, operating page, interface), in order to bring also any slave programmer to the end status.

III-15.2 DPROG (digital programmer (No. 27)) / DPROGD (DPROG data(No. 28))





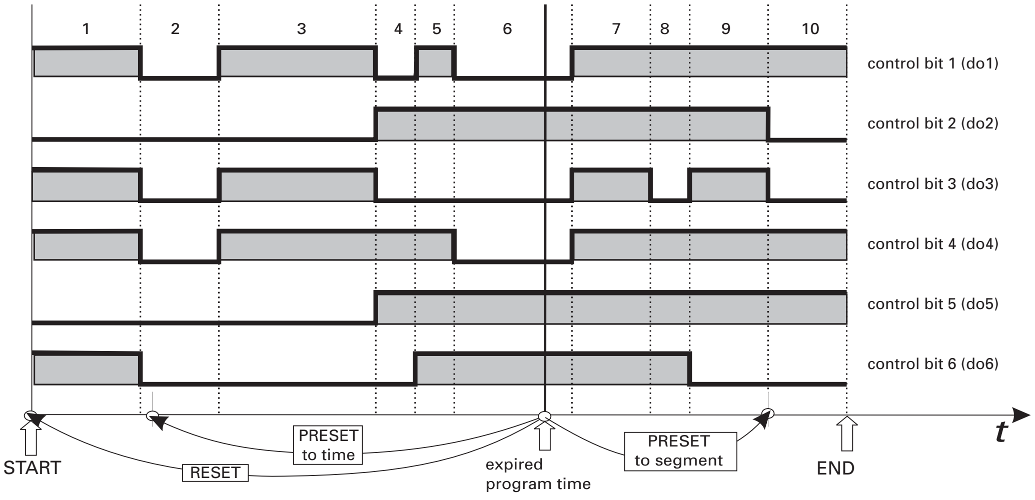
General

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.

Fig.:50 Digital programmer definition



Inputs/outputs

Digital inputs (DPROG):		
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)	
p-show	Program editing enabled	
halt	Program interruption (e.g. due to an exceeded bandwidth detected outside the programmer). 0 = program not halted 1 = program halted	
manfree	Manual mode disabled 0 = switchover to manual mode not permitted 1 = switchover to manual mode is permitted	

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 ☝ key / interface function `fkey' (:pressing key ☝ causes switch-over (0 or 1))
do1... do6	status of control outputs in the actual segment
preset	This output indicates a programmer preset operation. With a single preset command, a pulse is output for the duration of one cycle (dependent on the time slot in which the programmer is classified). If the programmer is held in preset continuously, this output is always active. 0 = no preset status 1 = APROG stands in preset status
manual	This output indicates the progrmmer manual mode. 0 = APROG works in automatic mode 1 = APROG works in manual mode

Analog inputs (DPROG):	
PSet	Preset value for program
DBlock	Block number of 1st data function `APROGD'
ProgNo	Required program number (recipe)
SlavNo	SlavNo: Block number of a connected slave block (for coupling master and slave blocks (APROG or DPROG)

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

Analog outputs (DPROG):	
TNetto	Net program time (Σ Trun)
TBrutt	Gross program time (Σ Trun + Σ Tstop)
TRest	Programmer rest time
SegNo	Actual segment number
ProgNo	Actual program number (recipe)
SegRest	Remaining segment time
Bl-no	Own block-number (e.g. for coupling master and slave blocks)

Analog outputs (DPROGD):	
DBlock	Own block number

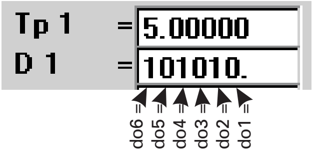
Parameter and configuration data

Parameter DPROG	Description		Range	Default
PMode	Preset Mode:	Preset to segment Preset to time	Pres.time Pres.seg.	←
RecMax	Max.recipees		1..99	99
D0	Status of control outputs 6...1 with reset		0 / 1 per output	000000

Parameter DPROGD	Description	Range ET	 Unit	Default ET	 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.



Tp 1 = 5.00000

D 1 = 101010.

Configuration DPROG	Description		Value
PwrUp	Behaviour after mains recovery	Continue program (default)	Prog.Fort
		Search run in the actual segment	Fort.Seg.
		Continue at actual time	Fort.Zeit
PEnd	Behaviour at program end PEnd: 0 = Stop 1 = Reset 2 = Reset + Stop (End-condition is reset with stop)	stop after program end (default)	Stop
		Reset after program end	Reset
Turbo	Time unit	Time = hours : minutes (default)	Std:Min
		Time = minutes : seconds	Min:Sek
FKey	FKey (key function): 0 = fkey output status switchover by key (previous function) 1 = key used to generate a pulse at the fkey output (pulse length = 1 cycle) 2 = key controls the programmer (fkey output generates a pulse when actuating the key, pulse length = 1 cycle)		

DPROG functions

All functions effective with the digital programmer are mentioned on the following list. As the procedure is almost completely identical for the analog programmer, we refer to the description in the corresponding APROG chapter.

Data blocks are cascadable I (like APROG, →page)
Program selection (like APROG, →page 221)
Progam changes during an active recipe (like APROG, →page)
Access to parameters of inactive recipes (like APROG, →page)
Programmer control via key (like APROG, →page 208)
Halt status (like APROG, →see page 208)
Automatic/manual mode with adjustability of control bits (like APROG, →page 208)
Recipe changing during reset status (like APROG →page 221)
Recipe names via TEXT block coupling (like APROG, →page)
Program end behaviour (like APROG, →page 206)
Master/slave operation (like APROG, →page 215)
Remaining segment time on the operating page and as output signal (like APROG, →page 218)
Operating page elements like for the APROG (with display of control blocks and relevant block number, →page 212)

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.

① *Program block name

② Recipe name: recipe can be switched over in reset status

③ Segment number: adjustable with segment preset

④ *automatic / manual (adjustable)

⑤ *Page changing:
permits switching to an analog or digital page connected by block coupling. This switchover type applies only to the values marked with *. The remaining display elements invariably show the master programmer values.

⑦ Control bit

⑧ *Control bits: can be altered individually in manual mode.

⑨ Remaining segment time: display is suppressed with preset to segment (e.g. with digital slave programmers). Otherwise the invariable overall segment time would be displayed.

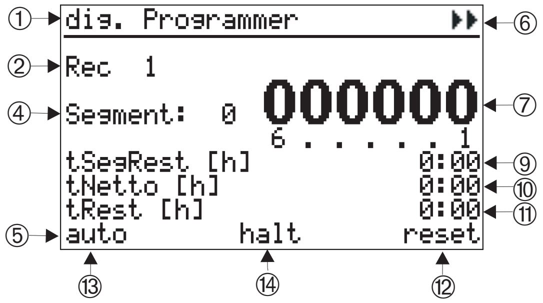
⑩ *Program net time: adjustable with preset to time

⑪ Remaining program time

⑫ Programmer status stop / run / reset / program / quit / error

⑬ halt / end (if none of the two conditions is active, this display remains empty)

Fig. 51



```
dig. Programmer              ▸▸
Rec  1
Segment:  0  000000
               6 . . . . . 1
tSegRest [h]               0:00
tNetto [h]                 0:00
tRest [h]                  0:00
auto        halt          reset
```

DPROG program edit page

Like the analog programmer, changing from the operating page to the subordinated parameter page is done by setting the status text shown at the bottom right to 'program' (possible only with p-show = 1). Display includes recipe name, the 6 start control bits and the segment parameters of the instantaneously active recipe.

Switchover to display of a different recipe is by changing the recipe name. This is possible at any time and does not cause switchover of the active recipe.

① Name of program block

② Recipe name:
Recipe changing is possible at any time .

③ Control bit status in reset mode

④ Time for segment 1

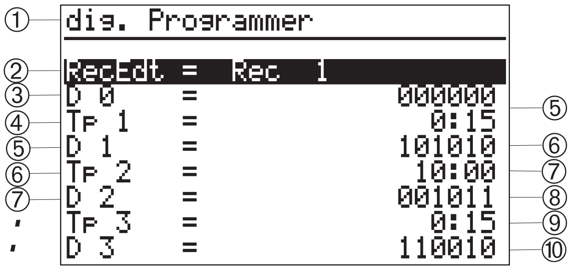
⑤ Control bit status in segment 1

⑥ Time for segment 2

⑦ Control bit status in segment 2

⑧ :

Fig. 52



```
dig. Programmer
RecEdt =  Rec  1
D 0    =        000000
Tp 1   =          0:15
D 1    =        101010
Tp 2   =         10:00
D 2    =        001011
Tp 3   =          0:15
D 3    =        110010
```

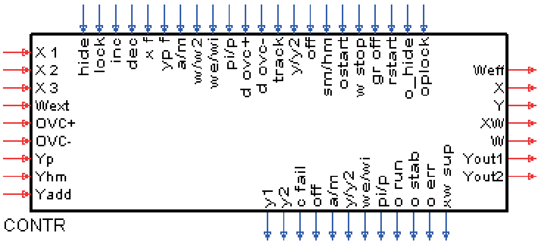
III-16 Controller

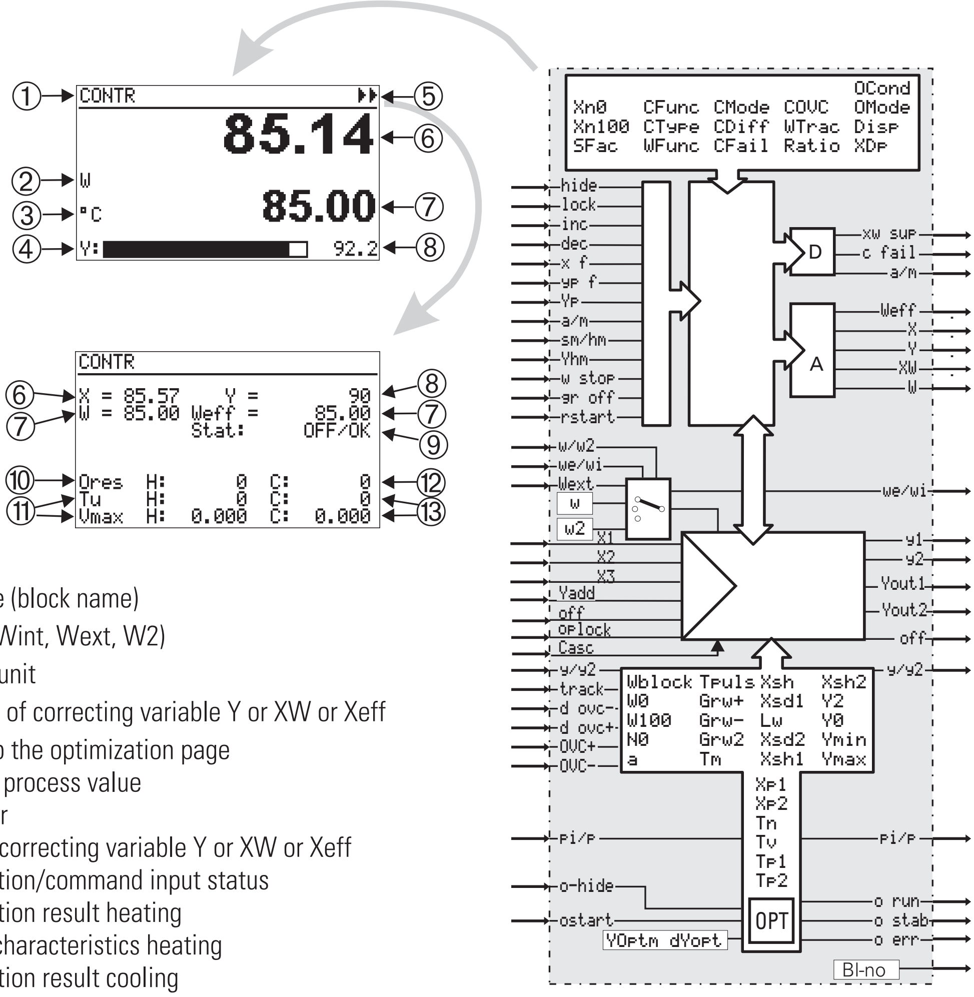
General: Function blocks CONTR and CONTR+ and PIDMA provide a complex control function. Unlike CONTR, CONTR+ includes six selectable control parameter sets. However, PIDMA provides a special control algorithm and different self-tuning.

In the following sections, the main features of the functionblocks CONTR, CONTR+ together and PIDMA separately are described. Then the common controltechnical application ranges are explained.

III-16.1 CONTR (Controllerfunction with one parameterset (No. 90))

The CONTR block provides a PID controller with numerous functions such as ramp, internal/external/W2 switchover, /process value tracking, self-tuning, override control, feed-forward control, positioning signal forcing, ratio and three-element control in 12 different controller types (continuous/2-point/3-point/3-point stepping/ ...).

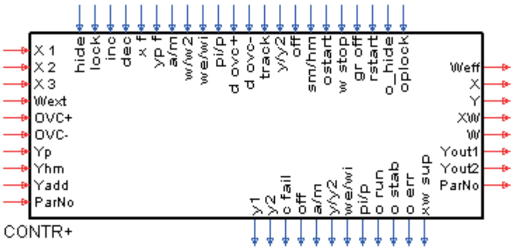


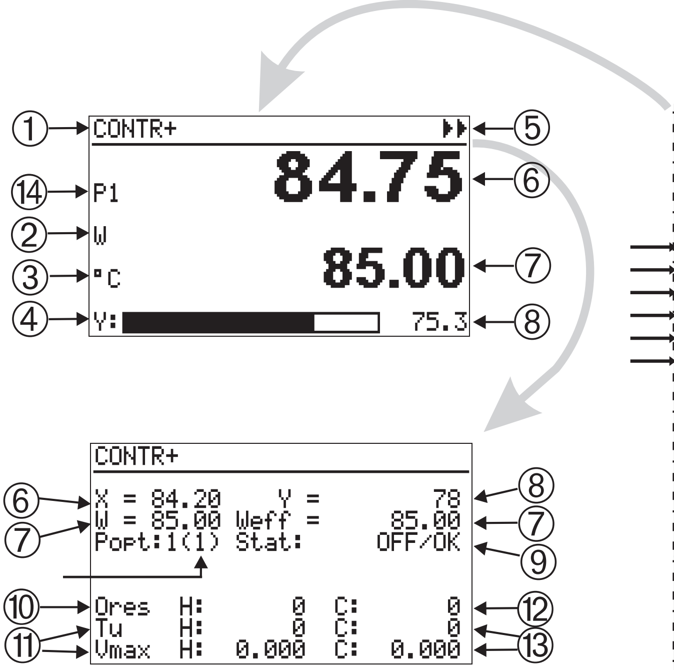


① Page title (block name)
② source (Wint, Wext, W2)
③ Physical unit
④ Bargraph of correcting variable Y or XW or Xeff
⑤ Entry into the optimization page
⑥ Effective process value
⑦ Controller
⑧ Value of correcting variable Y or XW or Xeff
⑨ Optimization/command input status
⑩ Optimization result heating
⑪ Process characteristics heating
⑫ Optimization result cooling
⑬ Process characteristics cooling

III-16.2 CONTR+ (Controllerfunction with six parametersets (No. 91))

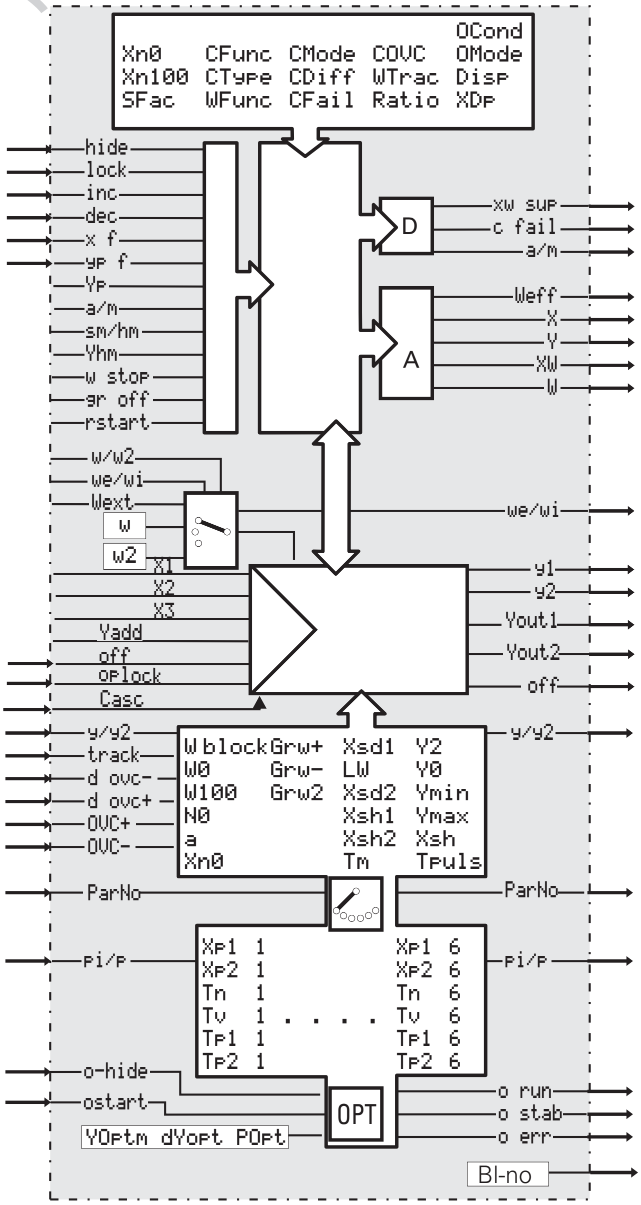
Function block CONTR+ provides the same functionality as the CONTR block. Additionally, it permits adaptation by adjustment. Six parameter sets can be activated dependent on process criteria (process value, , correcting variable, control deviation), plant or batch characteristics. The parameter sets can be determined independently by self-tuning.





```
CONTR+                 ▸▸
P1               84.75
W
°C               85.00
Y:█████████░░░     75.3
```

```
CONTR+
X = 84.20    Y =      78
W = 85.00 Weff =   85.00
Popt:1(1) Stat:   OFF/OK

Ores   H:     0  C:     0
Tu     H:     0  C:     0
Vmax   H: 0.000  C: 0.000
```



① Page title (block name)
② source (Wint, Wext, W2)
③ Physical unit
④ Bargraph of correcting variable Y or XW or Xeff
⑤ Entry into the optimization page
⑥ Effective process value
⑦ Controller
⑧ Value of correcting variable Y or XW or Xeff
⑨ Optimization/command entry status
⑩ Optimization result heating
⑪ Process characteristics heating
⑫ Optimization result cooling
⑬ Process characteristics cooling
⑭ Control parameter set selection

Inputs/outputs for CONTR and CONTR+

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 ID).
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. 1 = W2
we/wi	0 = external 1 = internal
pi/p	0 = PI action; 1 = P action 1) (r page PI/P switch-over) (not applicable to PIDMA)
d ovc+	1 = override control + with 3-point stepping controller (r page ff) (not applicable to PIDMA)
d ovc-	1 = override control - with 3-point stepping controller(r page ff) (not applicable to PIDMA)
track	0 = tracking function off; 1 = tracking function on(r page , ,)
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 r page
w stop	1 = effective freeze
gr off	1 = gradient suppression
rstart	1 = Start the ramp r the makes a step change towards the process value and goes to the adjusted process value according to GRW+ (GRW-). The rising flank (0r1) is evaluated.
o-hide	1 = self-tuning page display suppression
oplock	Blockage of key H (with oplock = 1, switchover to manual by means of key H 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
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 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
OVC+	Override control + r page
OVC-	Override control - r page
Yp	Position feedback
Yhm	Output with hard manual
Yadd	Feed-forward control
ParNo	Only with CONTR+; required parameter set
Casc	Cascade input for controller cascade

Analog outputs:	
Weff	Effective
X	Effective process value
Y	Effective output value
XW	Control deviation
W	Internal
Yout1	Output value yout1 (heating)
Yout2	Correcting variable yout2 (cooling; only with continuous controller with split-range behaviour r CFunc= splitRange)
ParNo	Only with CONTR+; effective parameter set
Bl-no	Own block number

III-16.3 Parameter and Konfiguration for CONTR, CONTR+

Parameter for CONTR and CONTR+

Parameter	Description		Description	Default	Unit
W_BWlock	switchover disable function	Switchover is disabled via front-panel operation	0: block All	←	
		Wext ↔ Wint switchover is disabled	1: block We		
		W ↔ W2 switchover is disabled	2; block W2		
		All switchover functions are enabled	3: none		
W0	Min. limit (Weff)		-29999...999999	0	0
W100	Max. limit (Weff)		-29999...999999	100	100
W2	Additional setpoint		-29999...999999	100	100
Grw+[3]	Gradient plus unit/min		0,001...999999	Off	----
Grw-[3]	Gradient minus unit/min		0,001...999999	Off	----
Grw2[3]	Setpoint gradient for W2 unit/min		0,001...999999	Off	----
N0	Zero offset for ratio control		-29999...999999	0	0
a	Factor a for 3-element control + setpoint ramps		-9,99...99,99	1	1
Xsh[2]	Trigger point separation (stepping controller)		0,2...20,0%	0,2	0,2
Tpuls	Min. positioning step time (stepping controller)		0,1...2,0[s]	0,3	0,3
Tm	Actuator response time (stepping controller)		5...999999 [s]	30	30
Xsd1	Switching difference (signaller control)		0,10...999999	1	1
LW	Distance additional contact (signaller control)		-29999...999999	Off	----
Xsd2	Switching difference additional contact (signaller control)		0,10...999999	1	1
Xsh1[1]	Trigger point separation (PD) (3-point controller)		0,0...1000,0[%]	0	0
Xsh2[1]	Trigger point separation (PD) (3-point controller)		0,0...1000,0[%]	0	0
Y2	Additional correcting variable (not for stepping controllers)		-105,0...105,0[%]	0	0
Ymin	Min. correcting variable limiting (not for stepping controller)		-105,0...105,0[%]	0	0
Ymax	Max. correcting variable limiting (not for stepping controller)		-105,0...105,0[%]	100	100
Y0	Working point of the controller (not for stepping controller)		-105,0...105,0[%]	0	0
Y0ptm[4]	Positioning value during process at rest		-105,0...105,0[%]	0	0
dYopt[4]	Self tuning step height		5...100[%]	100	100
P0pt[4]	Only for 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 split range)		0,1...999,9[%]	100	100
Tn 1...6	Integral action time (Tn = 0 → I-part is not effective)		0,0...999999[s]	10	10
Tv 1...6	Derivative time (Tv = 0 → D-part is not effective)		0,0...999999[s]	10	10
Tp1 1...6	Cycle time heating (2-and 3-point controller)		0,4...999,9[s]	5	5
Tp2 1...6	Cycle time cooling (3-point controller)		0,4...999,9[s]	5	5

[1] %-specifications are related to measuring range x_{n0} ... x_{n100}

[2] The neutral zone x_{sn} for 3-point stepping controllers depends on T_{puls}, T_m and x_{p1} (→ V. Hints for self-tuning).

[3] Gradient control → page 261

[4] Self optimization→ page 242 ff

Configuration data CONTR, CONTR+

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	Ratio	
		3-element controller → page 267	3-elem	
WFunc	function	control → page		←
		/cascade control → page	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. tracking	No tracking of Wint	Off	←
		tracking	SP-track	
		Process value tracking	PV-track	
Ratio	Ratio controller function:	(x1 + N0) / x2 → page 2667	Type 1	←
		(x1 + N0) / (x1 + x2) → page 266	Type 2	
		(x2 -x1 + N0) / x2 → page 266	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

III-16.4 Control behaviour

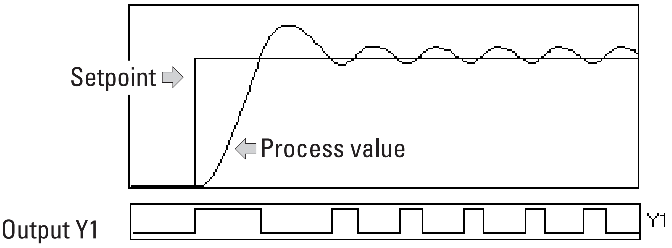
The following chapter describes the different control behaviours adjustable with the configuration parameter CFUNC and determines the parameters effective then.

All available parameters can be adjusted in the engineering tool. However it is not recognizable, which of the adjusted values are really affectingthe process.

The following compilation shall help to enlighten, which parameters are really used, dependent from the adjusted controller type. Therefore the relevant parameters for control behaviour are accented grey in the tables

Signaller, 1 output

The signaller is suitable for processes with small T_u and low v_{max} .



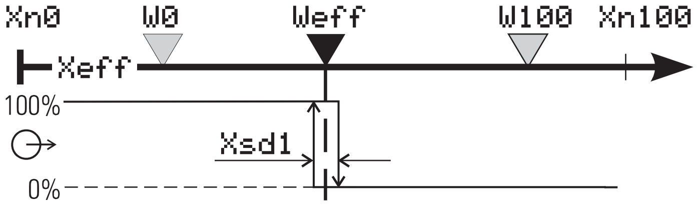
The advantage is in the low switching frequency. Switch-on is always at a fixed value below the , switch-off is always at a fixed value above the .

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 is the limit value. The trigger point is symmetrical to the ; hysteresis X_{sd1} is adjustable.

Fig.: 53
Static operating principle of the signalling function of a signaller, 1 output



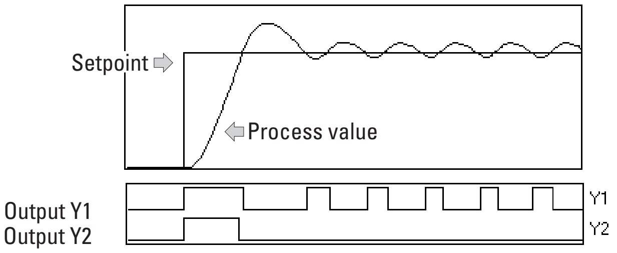
Configuration	Effective controller parameters of a signaller with one output			
CFunc = signaller, 1 output	W0	1)	Lower limit for Weff	-29 999 ...999 999
	W100	1)	Upper limit for Weff	-29 999 ...999 999
	W2	1)	Additional	-29 999 ...999 999
	Grw+	2)	gradient plus	off / 0,001 ... 999 999
	Grw-	2)	gradient minus	off / 0,001 ... 999 999
	Grw2	2)	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 characters
	W int		Internal after transmission of the engineering to KS 98-1	-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 261.

Signaller, 2 outputs

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 , whereas switch-off is always at a fixed point above the .

The control variable oscillation band is determined as a result of :

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

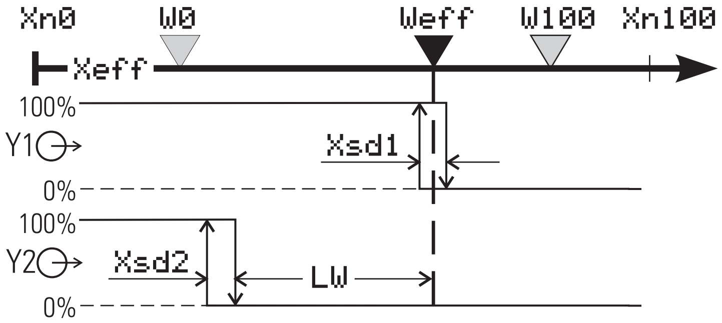
The signalling function provides alarm signalling, whereby the is the limit value. The trigger point is symmetrical to the ; hysteresis X_{sd1} is adjustable.

The signaller with two outputs has an additional “limit contact”. Its difference from the is adjustable in parameter LW (including polarity sign).

Fig.: 54
Static operating principle of the signalling function

Signaller, 2 outputs

LW is shown as a negative value in the example (e.g. -20)



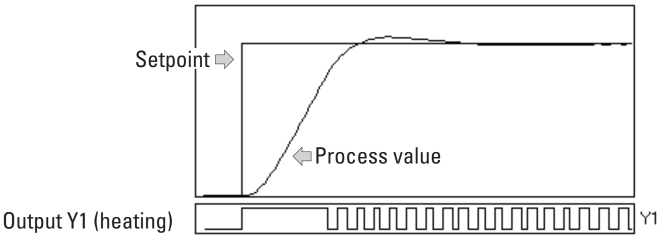
Configuration	Effective controller parameters of a signaller with two outputs			
CFunc = Signaller, 2 outputs	W0	1)	Lower limit for Weff	-29 999 ...999 999
	W100	1)	Upper limit for Weff	-29 999 ...999 999
	W2	1)	Additional	-29 999 ...999 999
	Grw+	2)	gradient plus	off / 0,001 ... 999 999
	Grw-	2)	gradient minus	off / 0,001 ... 999 999
	Grw2	2)	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 = OFF
	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 after transmission of the engineering to KS 98-1	-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 261.

Two-point controller

Switching controller with two switching statuses:



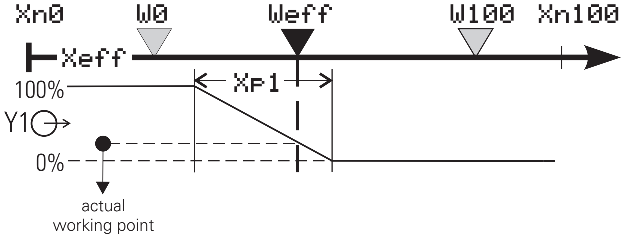
1. Heating switched on; ⟶ output Y1 = 1
2. Heating switched off; ⟶ output Y1 = 0

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.

Fig.: 55
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 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 limit for Weff	-29 999 ...999 999
	W100	1)	Upper limit for Weff	-29 999 ...999 999
	W2	1)	Additional	-29 999 ...999 999
	Grw+	2)	gradient plus	off / 0,001 ... 999 999
	Grw-	2)	gradient minus	off / 0,001 ... 999 999
	Grw2	2)	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 after transmission of the engineering to KS 98-1	-29 999 ...999 999
	A/H		Controller status after transmission of the engineering to KS 98-1	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 261.

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).

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).

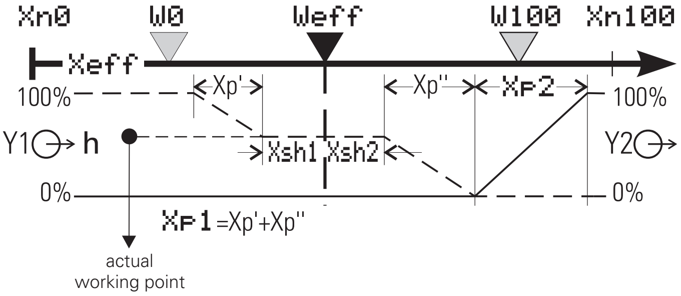
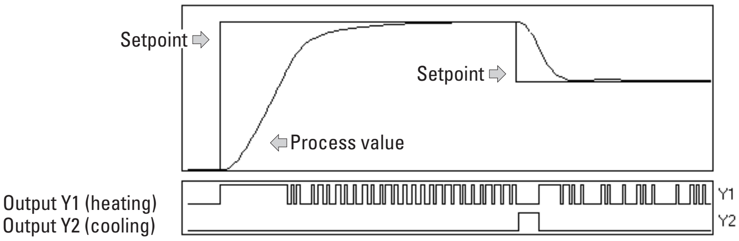


Fig.: 56
Static operating principle
of a three-point controller

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.

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 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.: 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 .

The type of positioning signals is selectable:

CFunc = 3-point	heating switching,	cooling switching
CFunc = cont/switch	heating continuous,	cooling switching
CFunc = switch/cont	heating switching,	cooling continuous

Combination "heating continuous" and "cooling continuous" is covered by "splitRange - continuous controller with split-range behaviour". → see also "continuous controller" page: 238.

Configuration	Effective controller parameters with three-point controller		
CFunc = 2-Punkt	Popt	Parameter set for self-tun ing (only with CONTR+)	1...6
	W0 1)	Lower limit for Weff	-29 999 ...999 999
	W100 1)	Upper limit for Weff	-29 999 ...999 999
	W2 1)	Additional	-29 999 ...999 999
	Grw+ 2)	gradient plus	off / 0,001 ... 999 999
	Grw- 2)	gradient minus	off / 0,001 ... 999 999
	Grw2 2)	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 cooling	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 after transmission of the engineering to KS 98-1	-29 999 ...999 999
	A/H	Status of controller after transmission of the engineering to KS 98-1	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 261.

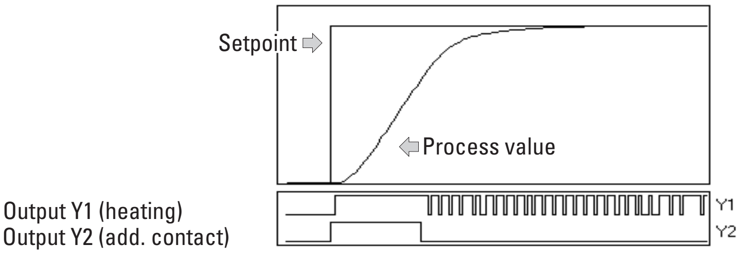
3) % specifications are related to measuring range Xn100 - Xn0. There is no relation to values W0 and W100.

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).

∆ / Y / off

The principle is identical to the control behaviour of a 2-point controller with additional contact.

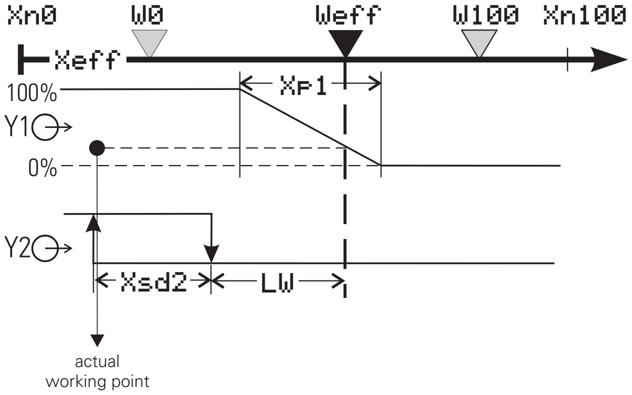


Output Y2 is used for switchover of the connected circuit between "∆" and "Y". Output Y1 switches the heating energy on and off.

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.: 57
Fig. Static operating principle of the ∆ / Y / off function



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 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.

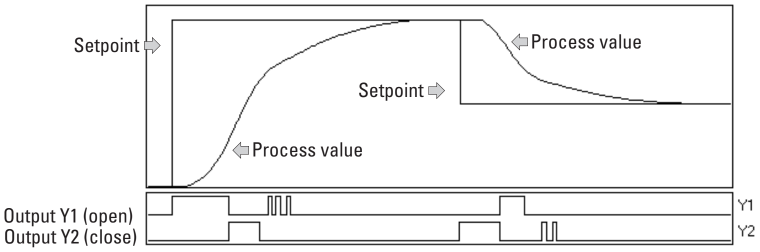
The shortest switch-on or off time is 100ms.

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 limit for Weff	-29 999 ...999 999
	W100	1)	Upper limit for Weff	-29 999 ...999 999
	W2	1)	Additional	-29 999 ...999 999
	Grw+	2)	gradient plus	off / 0,001 ... 999 999
	Grw-	2)	gradient minus	off / 0,001 ... 999 999
	Grw2	2)	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 after transmission of the engineering to KS 98-1	-29 999 ...999 999
	A/H		Controller status after transmission of the engineering to KS 98-1	0 or1

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 261.
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)

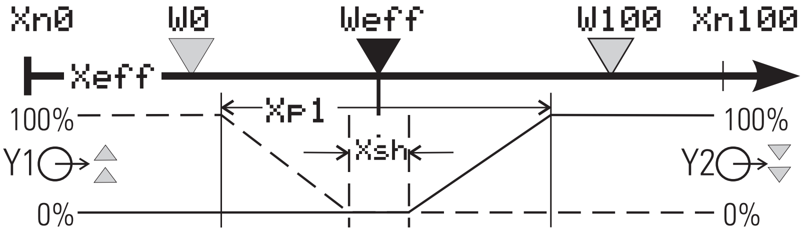


1. Open valve; → outputs Y1 = 1, Y2 = 0
2. Don’t move valve; → outputs Y1 = 0, Y2 = 0
3. Close valve; → outputs Y1 = 0, Y2 = 1

To validate the adjusted X_{p1} for the actuator response time, response time T_m must be adjusted. The smallest positioning step is 100ms.

☞ **With PMA controllers the position feedback has no influence on the PID-behaviour!**

Fig.: 58
Static operating principle of a three-point stepping controller



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.

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 the figure above.

The hysteresis shown in this diagram is practically unimportant, but can be calculated from the adjustable min. pulse length $T_{puls} \geq$ 100ms.

$$X_{sh} = \frac{Tpuls}{2} \cdot 0{,}1 \cdot \frac{Xp}{Tm}$$

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:

$$X_{sh} = 12{,}5 \cdot Xp \cdot \frac{Tpuls'}{Tm} - 0{,}75$$

Configuration	Effective controller parameters with three-point stepping controller		
CFunc = Step Step Yp	Popt	Parameter set for self-tuning (only with CONTR+)	1...6
	W0 1)	Lower limit for Weff	-29 999 ...999 999
	W100 1)	Upper limit for Weff	-29 999 ...999 999
	W2 1)	Additional	-29 999 ...999 999
	Grw+ 2)	gradient plus	off / 0,001 ... 999 999
	Grw- 2)	gradient minus	off / 0,001 ... 999 999
	Grw2 2)	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 after transmission of the engineering to KS 98-1	-29 999 ...999 999
	A/H	Controller status after transmission of the engineering to KS 98-1	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 261.

3) % specifications are related to measuring range Xn100 - Xn0 .
There is no relationship to values W0 and W100.

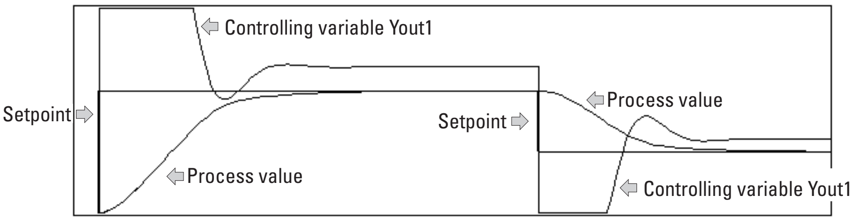
4) (1...6) refers to the six parameter sets of CONTR+ (e.g. Xp1, Xp2, Xp3...Xp6).

Continuous controller / Split range

Continuous controller

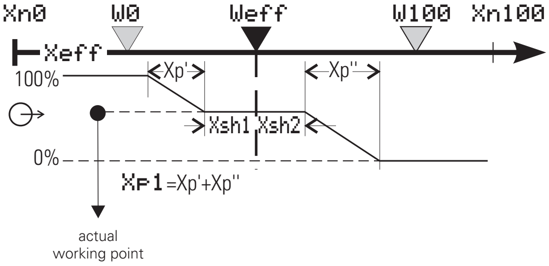
An analog value is provided as correcting variable by output Yout1, e.g. temperature control with electrical heating and thyristor power regulator.

A continuous controller in 'split-range' operation is comparable with a three-point controller. The neutral zone is also separately adjustable.



Within limits Xsh1 and Xsh2, the control deviation for calculation of the controller reaction is set to zero. A pure P controller does not change the correcting variable within these limits. A PID controller has a dynamic behaviour which has not always decayed, also when reaching "control deviation = 0". Both D and I action can still have an effect according to the characteristic determined by Tv due to a preceding disturbance or a setpoint step change. This effect can be strong enough to cause the control deviation to leave range Xsh1/Xsh2, i.e. the P action is activated again for reaching the neutral zone.

Fig.: 59
Operating principle of the proportional part of the continous controller



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 = continuous SplitRange	Popt	Parameter set for self-tuning (only with CONTR+)	1…6
	W0 1)	Lower limit for Weff	-29 999 …999 999
	W100 1)	Upper limit for Weff	-29 999 …999 999
	W2 1)	Additional	-29 999 …999 999
	Grw+ 2)	gradient plus	off / 0,001 … 999 999
	Grw- 2)	gradient minus	off / 0,001 … 999 999
	Grw2 2)	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	(-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 after transmission of the engineering to KS 98-1	-29 999 …999 999
	A/H	Controller status after transmission of the engineering to KS 98-1	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 261.

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).

III-16.5 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 Figure below).

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).

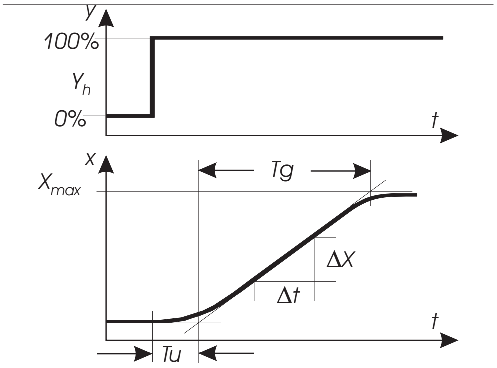
$$K = \frac{Vmax}{Xh} \cdot Tu \cdot 100\%$$

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

Fig.: 60 step response



Characteristic values of the controllers

Generally, quick line-out to the 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

Direct / inverse switchover is always possible in configuration parameter CMode (action).

III-16.6 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:

① 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.

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).

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.

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 .

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 ⑦).

⑥ 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).

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.

III-16.7 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 ④ 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 (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 242).

- 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.

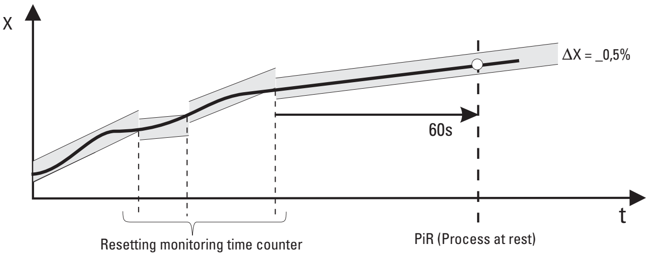
- Consider the reserve
  (→ also see reserve, page 243)

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!

Fig. 61



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.

reserve:

In order to make self-tuning possible, the separation between and process value before the output step change must be higher than 10 % of W0...W100. The reserve is provided either automatically by reducing the correcting variable during the PiRphase, or by changing the or the process value manually (manual mode).

With inverse controllers, the must exceed the process value by at least the reserve.
With direct controllers, the must be smaller than the process value by at least the reserve.
This is necessary, as the 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/manual mode" page 243).

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. 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 ⏎, 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 reserve (→ see page 243) is provided, the correcting variable is changed by output step dYOpt (increased with indirect controller, increased with direct controller).

The self-tuning procedure is realized using the varying process value.

Fig.: 62 self-tuning [grad(x)=0]

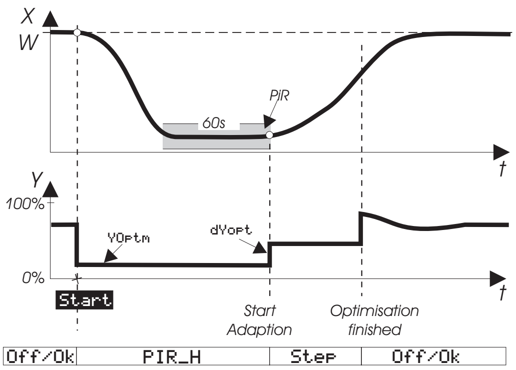
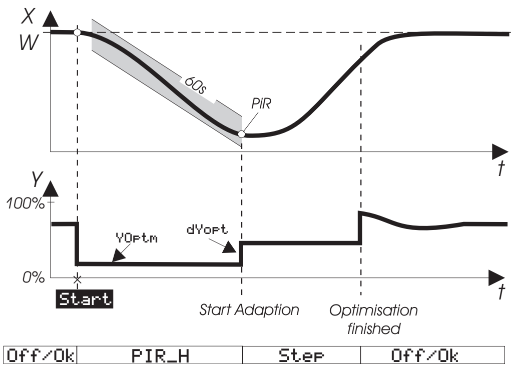


Fig.: 63 self-tuning [grad(x)<0]



After successful self-tuning, the controller goes to the automatic mode and controls the using the new parameters. Parameter Ores indicates the self-tuning result (→ see page 246) .

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.

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 can be changed at any time.

With 'Process at rest' (PiR) detection and a sufficient reserve (→ see page 243), the correcting variable is changed by the correcting variable step dYOpt. (increased with indirect controller, decreased with direct controller). *`Process at Rest'* (PiR) can be reached at starting time, i.e. the normal 60 s waiting time can be avoided.

The self-tuning procedure is realized using the varying process value.

After successful self-tuning, the controller goes to the automatic mode and controls the using the new parameters. Parameter Ores indicates the self-tuning results (→ see page 246) .

Fig.: 64 Start by rising the setpoint

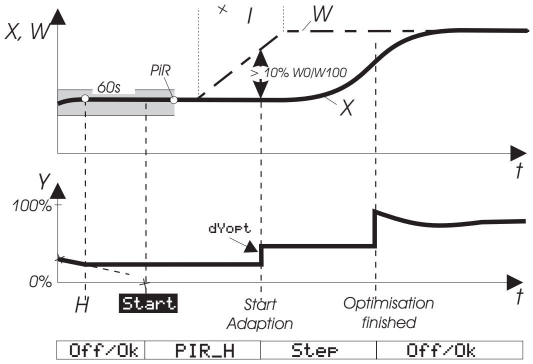
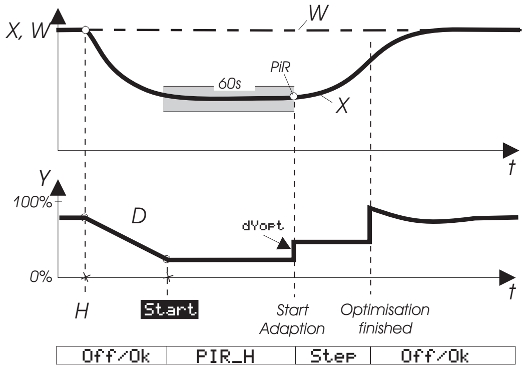


Fig.: 65 Start by lowering the setpoint



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.

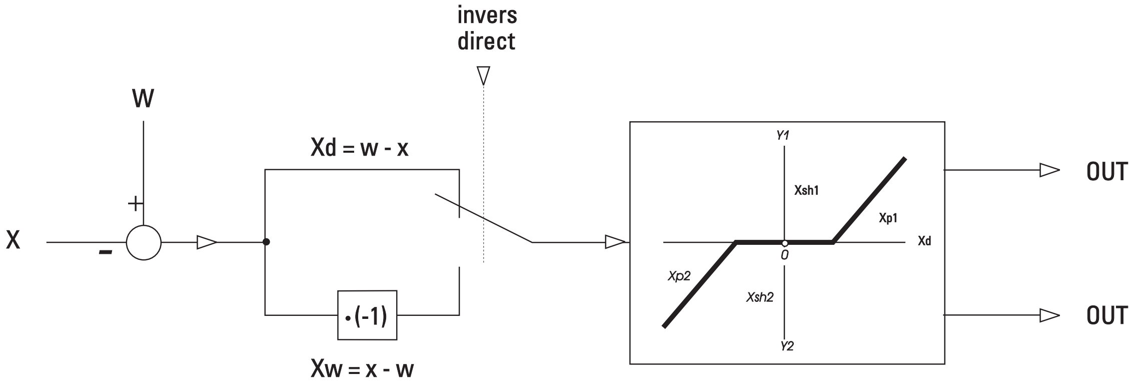
Self-tuning procedure with heating:

(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:

Fig.: 67 Principle of direct/invers switchover

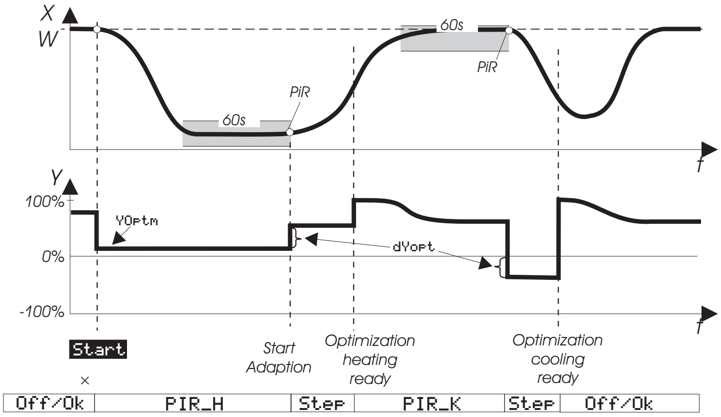


(3-point / split-range controller)

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 , 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 indicated.

Fig.: 66 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: Abgl.) is not shown in the figures.

ⓘ For maintaining a safe process condition, monitoring for an exceeded 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 star, i.e.Y2 = 0.**

Controlled adaptation

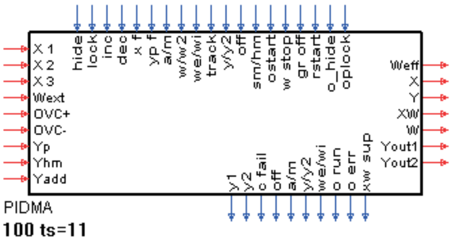
For defined applications, adaptation of the control parameter set to the current process condition is purposeful. For this, the Contr+ is provided with 6 control parameter sets, which can be selected via analog input ParNo .

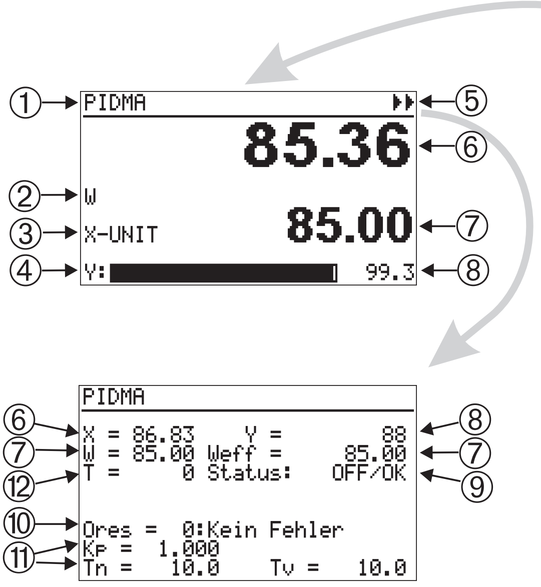
Signification of self-tuning messages ORes1/ORes2

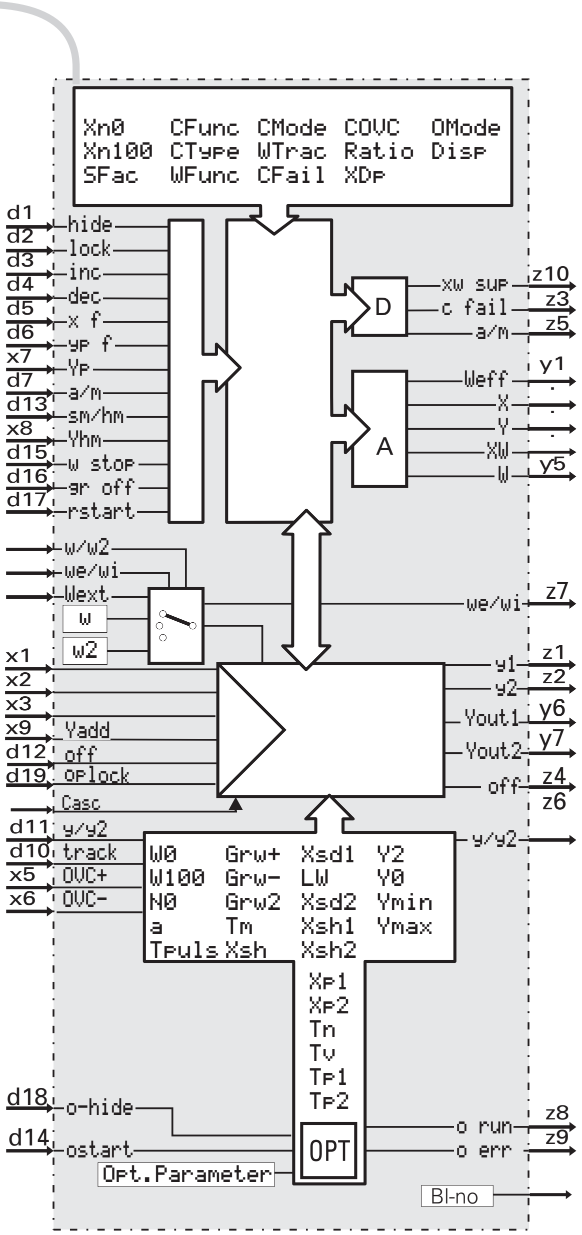
ⓘ Unless control is as required despite self-tuning, proceed additionally as described in section "Empirical self-tuning" (→ page , Hints for self-tuning, Hints for adjustment), and follow the hints given on further parameters.

III-16.8 PIDMA (Control function with particular self-tuning behaviour (No. 93)

The controller block PIDMA is particularly suitable for difficult processes (with delay time, or of higher order). The difference between PIDMA and the CONTR block is only in the PID controller kernel (self-tuning and control algorithm). The additional functions ramp, switchover, override control, feed-forward control etc., which are described in the CONTR section, are not different.







① Page title (block name)

② source (Wint, Wext, W2)

③ Physical unit

④ Bargraph of correcting variable Y or XW or Xeff

⑤ Entry into the optimization page

⑥ Effective process value

⑦ Internal setpoint

⑧ Value of correcting variable Y or XW or Xeff

⑨ Optimization/command entry status

⑩ Optimization result heating

⑪ PIDMA control parameters

⑫ Test time/remaining time

The most important differences compared to controller functions CONTR and CONTR+ are:

- Integrated, front-panel operated optimization method like PMA tune
  This function permits optimization also for difficult processes with Tg/Tu < 3 without engineering tool and laptop, which was not possible with previous PMA (and competitor) controllers.
- Parallel controller structure unlike all other PMA controllers with “serial structure”.
- Distinction of “ behaviour” and “disturbance behaviour” by adjustable factors, which can be used for individual attenuation of P (proportional action) and D (differential action) control effect on changes.
- The adjustable derivative gain VD of the D action, which is adjusted and matched to the process dynamics automatically by self-tuning. Purposeful values for VD are within 2…10, whereby all previous PMA controllers are fixed to VD=4 (empirical value for serial structure).

Using a PIDMA control block is purposeful where conventional PMA self-tuning methods are not satisfactory. PIDMA should not be used for processes where PMA self-tuning has always been and still is unrivalled:

- Processes with a ratio Tg/Tu > 10
- (2nd order processes; with 2 [...3] energy storing elements!).

Roughly, these are applications in the plastics processing industry (extrusion, ...), where no improvements related to quick line-out without overshoot are possible (unless a “robust” controller design with stable results also with variable process dynamics and non-linearities is required)!

With classic thermal processes (ovens of all types, dryers, ...) air conditioning, level, flow, etc., however, a considerable number of difficult cases require the expenditure of many hours spent with hotline support or even at the site, in order to make a plant operate properly.

The various types of control behaviour are not explained in detail in this section, because there are no differences compared to controller blocks CONTR and CONTR+ (see from page).

Only the additional parameters explained at the beginning of chapter "PIDMA control characteristics” have to be described.

The difference between split-range and 3-point behaviour is that PIDMA does not provide parameter distinction of heating and cooling.

☞ PIDMA does not permit the adjustment of the signaller control behaviour.

Inputs/outputs for PIDMA

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. Function keys ▲▼ are not active).
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. 1 = W2
we/wi	0 = external setpoint 1 = internal setpoint
track	0 = tracking function off; 1 = tracking function on(→ page 273, 262, 263)
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
w stop	1 = effective setpoint freeze
gr off	1 = setpoint gradient suppression
rstart	1 = Start the setpoint ramp → the setpoint 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 setpoint, 1 = internal setpoint
o run	Self-tuning running
o err	Error during self-tuning
xw sup	Alarm suppression with setpoint 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
OVC+	Override control + → page 269
OVC-	Override control - → page 269
Yp	Position feedback
Yhm	Output with hard manual
Yadd	Feed-forward control
Casc	Cascade input for controller cascade
ParNo	Only with CONTR+; required parameter set

Analog outputs:	
Weff	Effective setpoint
X	Effective process value
Y	Effective output value
XW	Control deviation
W	Internal
Yout1	Output value yout1 (heating)
Yout2	Correcting variable yout2 (cooling; only with continuous controller with split-range behaviour → CFunc= splitRange)
Bl-no	Own block number
ParNo	Only with CONTR+; effective parameter set

III-16.9 Parameter and configuration for PIDMA

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
W_Block	switchover disable function	Switchover via front-panel operation disabled	0: block All	t
		Wext ←→ Wint switchover is disabled	1: block We	
		W ←→W2 switchover is disabled	2; block W2	
		All switchover functions are enabled	3: none	
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	off	----
Grw-[2]	Setpoint gradient minus unit/min	0,001...999999	off	----
Grw2[2]	Setpoint gradient for W2 unit/min	0,001...999999	off	----
N0	Zero offset with ratio control	-29999...999999	0	0
a	Factor a for 3-element control	-9,99...99,99	1	1
Xsh1[1]	Trigger point separation (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	Threshold for OPEN and CLOSE (stepping controller) currently not active	0,2...100%	0,2	0,2
throff	Threshold for OPEN and CLOSE (stepping controller) currently not active	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 261
[3] for self-tuning → page 242 ff

Configuration data 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	Ratio	
		3-element controller → page 267	3-elem	
WFunc	Setpoint function	setpointcontrol → page	setp	←
		setpoint/cascade control → page	Sp/casc	
CMode	Output action	Inverse output action	Inverse	←
		Direct output action	Direct	
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. tracking	No tracking of Wint	Off	←
		setpoint tracking	SP-track	
		Process value tracking	PV-track	
Ratio	Ratio controller function:	(x1 + N0) / x2 → page 2667	Type 1	←
		(x1 + N0) / (x1 + x2) → page 266	Type 2	
		(x2 -x1 + N0) / x2 → page 266	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	
Xn0	Span start		-29999 ... 999 999	0
Xn100	Span end		-29999 ... 999 999	100
SFac	Factor stoichiom. ratio		0,01 ... 99,99	1,00

III-16.10 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	Description	Values
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 limiting of the minimum pulse via Tpuls.

thronoff:

The initially provided parameters for motor stepping controller structure in PIDMA are ineffective in the present realization. Only parameter xsh can be used for stabilizing the positioning activities.

Xsh

Xsh can be used to influence the motor actuator switching frequency and fine setting. Xsh determines the dead band of the control deviation in the main controller. The controller I action is stopped within this zone.

Integrated position controller

With 3-point stepping controller with position feedback (step + Yp), the PIDMA function block comprises two controllers: the main controller controls the process value and provides a required actuator position to an integrated position controller. By means of position feedback, this position controller ensures that the actuator position is as required.

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.

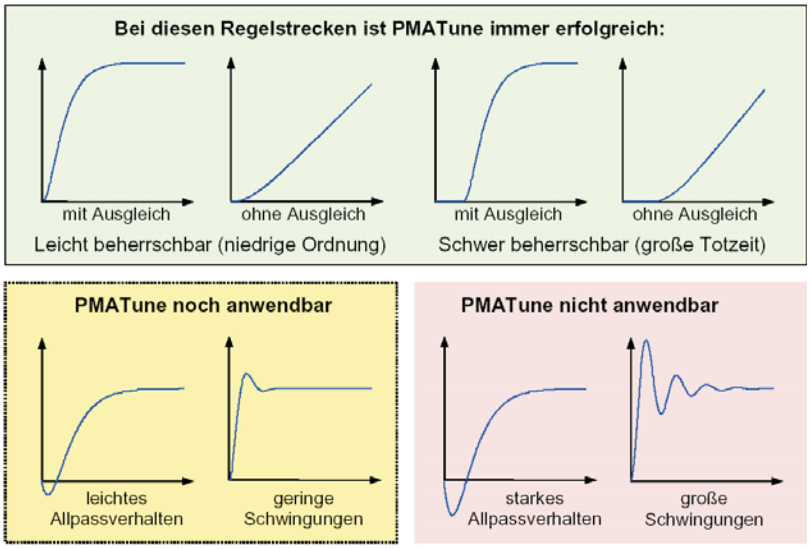
Further parameters permit independent weighting of individual controller components:

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.

Fig. 68



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' monitoring:

The PIDMA does not provide monitoring for the rest condition. The commissioning engineer is responsible for selection of a suitable start time. Optimum results are only achieved, if the process is lined out, i.e. all dynamic actions have decayed. Only in few cases in which parameter determination is imposssible due to decaying dynamics the algorithm provides a "restart" error message.

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 ⏎.

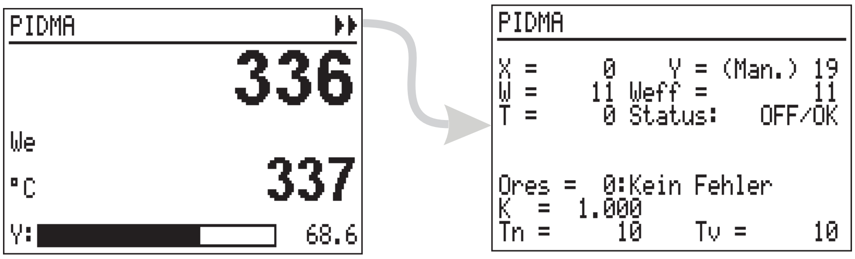
Stat: OFF/OK blinks and can be switched over to Stat: Start by pressing ▲.

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 ✋ was not disabled (1-signal on input oplook).

Fig. 69 Self-tuning page



```
PIDMA                 ▸▸
                     336
We
°C                   337
Y:                  68.6
```

```
PIDMA

X =      0    Y = (Man.) 19
W =     11 Weff =        11
T =      0 Status:   OFF/OK


Ores =  0:Kein Fehler
K  =  1.000
Tn =      10    Tv =     10
```

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 ⏎, Stat: line blinks. Press ▲ until Stat: Stop blinks. Press ⏎, the self-tuning attempt is stopped and the controller continues operating in automatic mode.

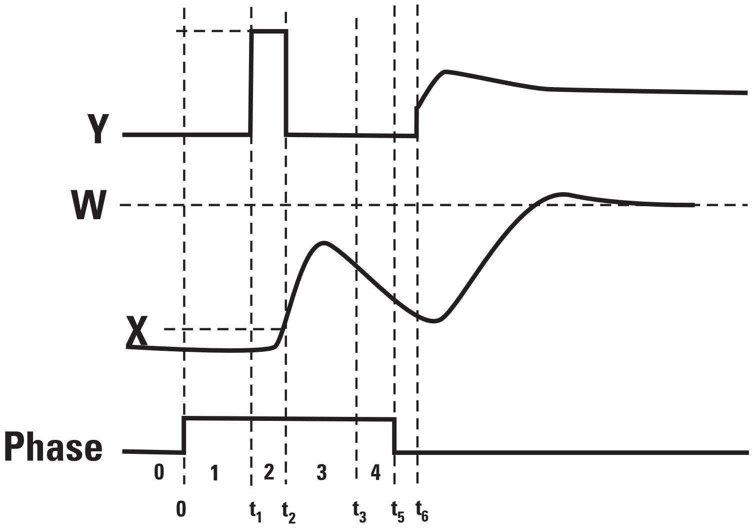
Start in manual mode or in automatic mode :

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 45) .

Fig. 70 PIDMA-optimization progress



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, 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

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.

III-16.11 Controller applications:

The following chapter describes the common characteristics of controller block connection, which are independent of the CONTR and PIDMA controller kernel, such as and correcting variable switchover and limitingas well as process value pre-processing.

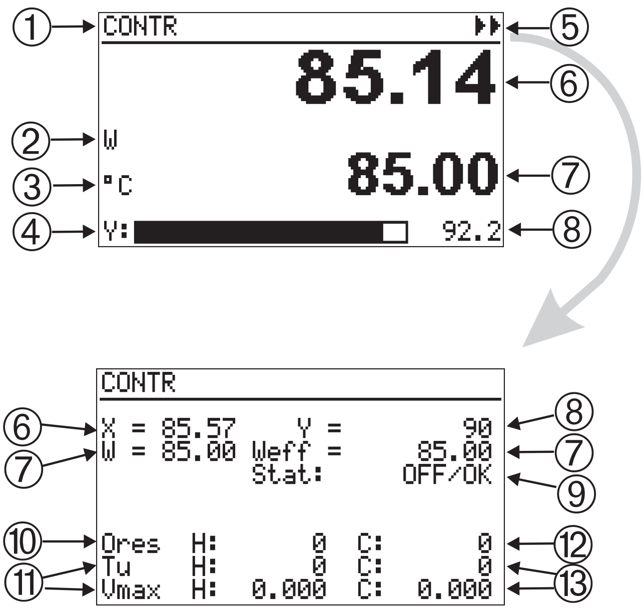
Controller front-panel operation

Controls on the controller page

Multilingual operation is not provided for the controller operating pages. Texts such as title and unit should be language-independent, if necessary.

① Page title (block name)
② source (Wint, Wext, W2)
③ Physical unit
④ Bargraph of correcting variable Y or XW or Xeff
⑤ Entry into the self-tuning page
⑥ Effective process value
⑦ Controller setpoint
⑧ Value of correcting variable Y or XW or Xeff
⑨ Self-tuning/command entry status
⑩ Self-tuning result heating
⑪ Process characteristics heating
⑫ Self-tuning result cooling
⑬ Process characteristics cooling

Fig. 71



☞ See also:
Chapter “Operation” on page
Chapter “Operating pages” on page
Chapter “Controller” on page

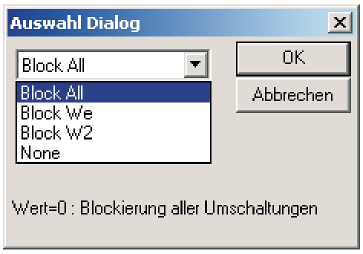
Switchover disabling

In many applications, switchover via the front panel should be not possible.

Unwanted, accidental process interventions should be prevented by all means. For these cases, switching over via front panel operation may be disabled.

This is done by parameter W Block, which is intended for blocking individual or all switchover operations purposefully.

In default setting, all switchover operations are blocked and the switchover field for front-panel operation cannot be selected.



Wert=0 : Blockierung aller Umschaltungen

☞ Switchover to Wext is blocked by configuration Wfunc = .

☞ When W < > W2 switchover is blocked and Wext < > Wint switchover is not possible (control), the field is skipped when selecting.

Further status displays on the operating page

During optimization or with cascade control, further display elements may appear on the operating page.

Statuses during optimization

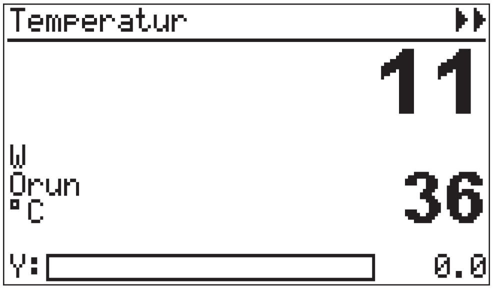
The optimization statuses are displayed with priority in the display field for manual operation.

Optimization running:	display:	Orun
Faulty optimization:	display:	Oerr

When the optimization is finished with an error, the unit waits for acknowledgement by the operator.

By pressing key twice or by input of the stop command on the optimization page, the controller returns to the initial status.

Fig.:72 Reglerseite bei gestarteter Optimierung.



Cascade control operation

The cascade is one of most frequent controller structures with coupled control loops.

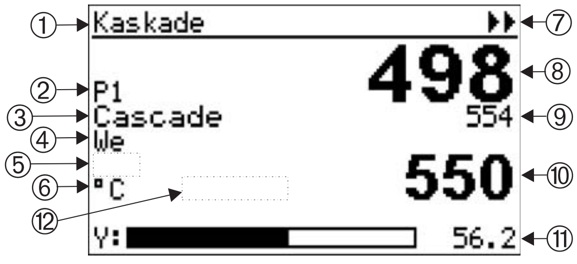
To facilitate construction and handling of these cascades, measures for connection and operation were taken at and in the controller blocks.

- A cascade consists of min. two controllers: a master controller the process value of which provides the main correcting variable and a slave controller on the process value of which the main variable is dependent.
- For building a cascade, the correcting variable output (Yout1) of the master may be wired to the input ((Wext) of the slave controller via a scaling function (SCAL).
- The slave controller is informed on the cascade by connecting the master block number output to the slave cascade input.

☞ The special operating functions of a controller cascade for master and slave are grouped on the common operating page of the slave controller.

① Title of the operating page
② Parameter set selection, if available
③ Cascade mode toggle field (open/close)
④ Master source (W_{int}, W_{ext}, W_2)
⑤ Display field for manual mode (otherwise empty)
⑥ Physical unit (master or slave)
⑦ Entry into self-tuning
⑧ Master
⑨ Slave process value
⑩ (from master during automatic mode, from slave with open cascade)
⑪ Bargraph and display (Y from slave or X/XW from master)
⑫ Display of slave selection with open cascade (otherwise empty)

Fig. 73 Operating page of a cascade controller in automatic mode

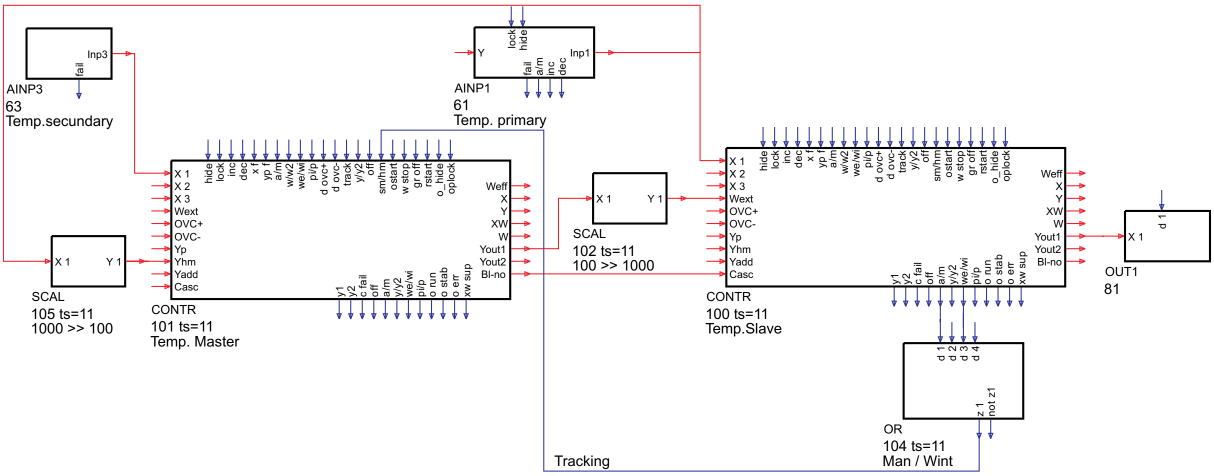


Cascade operation is possible in the following operating statuses

(see also section operating pages on page 36):

- During automatic mode, the master variables (and process value) are the relevant variables in the process. The master controller is directly adjustable. The process value of the slave controller is only displayed. "**Cascade**" is displayed.
- Like every slave controller, the slave can be switched over to its internalsetpoint or to W2 via control inputs. In this case, display is "**Casc-open**" as when opening via the operating field. Now the slave controller becomes the process-relevant variable and can be adjusted via the field (display "Slave" left in front of the setpoint). The master circuit process value is provided by the slave circuit rather than being controlled. Switching over between master or slave setpoint operation is possible at any time.
- During manual mode, the process is influenced directly by the slave controller correcting variable. The slave correcting variable is adjustable during manual mode. "**Man**" is displayed.
  During manual mode, or when the slave works with the internal setpoint or W2, the cascade is opened.
  The slave does not react on the master correcting variable any more. Suitable measures for master correcting variable tracking by the slave process value should be provided in the engineering, in order to ensure bumpless switchover to automatic mode (see example Fig.74)

Fig. 74 Engineering view of a cascade structure



Cascade optimization

switchover disabling on the slave by means of parameter W Block prevents cascade opening via front panel operation! This parameter can be used to influence the W/We/W_2 selection via the front panel.

Closing of the cascade will cause automatic slave switchover to the external We.

For identification of the data source, text “Slave” is displayed right beside the unit field when the cascade is open. A longer unit text can be overwritten partly.

Now only the first 4 characters of the unit are visible.

During cascade operation, the master information is displayed in the fields for , source, physical unit and X/XW bargraph. With open cascade (display “Slave”), the slave information is displayed instead.

Optimizing the cascade

In a cascade, the slave controller and, subsequently, the master controller must be optimized. Self-tuning entry ▶▶ on the cascade operating page always refers to the slave!
For master optimization, the master must be selected purposefully via the operating menu!

Manual operation

Changing between automatic and manual mode is by pressing key ✋. The manual mode influences only the slave controller. The master is concerned only indirectly.

The bargraph display switches over to slave variable Y. Adjustment of the correcting variable is via the value beside the bargraph.

The setpoint switchover and adjustment operations influence the master controller !

The following rules apply to the bargraph display :

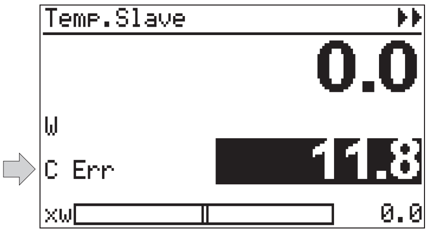
- When X or XW display is selected for the master bargraph display, the display value from the master controller is taken over.
- If Y display is selected, the slave bargraph value is used.

Faulty wiring of a controller cascade

If an invalid cascade circuit was built up in the engineering, e.g. with the cascade input not connected to output Bl-no of a master controller, the control function is not operable.

Error signalling is in the display field for the cascade:
Display: C Err

Fig.: 75 Faulty cascade wiring

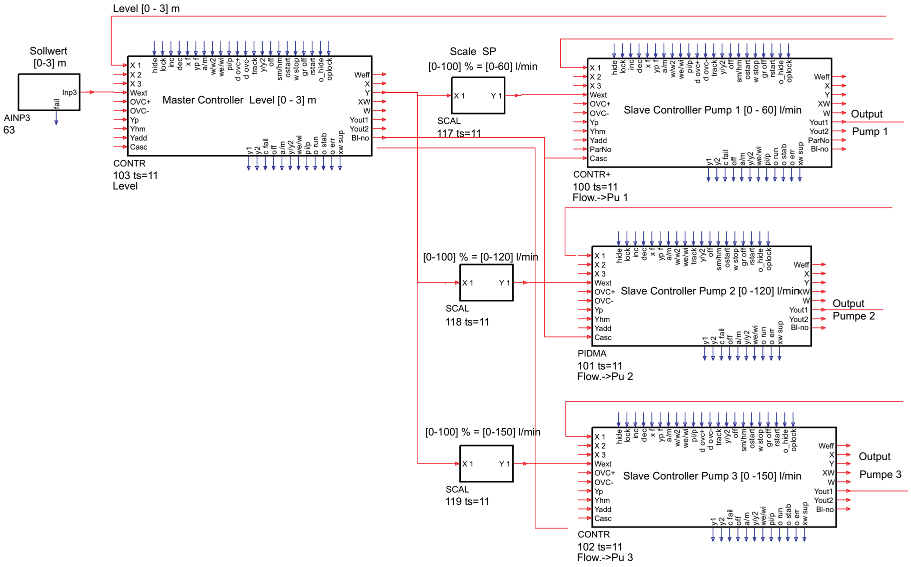


```
Temp.Slave          ▸▸
                   0.0
W
C Err             11.8
xw                 0.0
```

Multiple cascade

A cascade control loop can be built up from a master controller with one or several slave controllers (see Fig. 76 : Level control example with three subordinated flow controllers). Cascade operation is from the slave controller page. Display of the master operating page should be suppressed (hide=1).

Fig. 76 : Example of a flow-control


Level [0 - 3] m
Sollwert [0-3] m
Inp3
fail
AINP3 63
Master Controller Level [0 - 3] m
CONTR 103 ts=11 Level
Scale SP [0-100] % = [0-60] l/min
SCAL 117 ts=11
Slave Controlller Pump 1 [0 - 60] l/min
CONTR+ 100 ts=11 Flow.->Pu 1
Output Pump 1
[0-100] % = [0-120] l/min
SCAL 118 ts=11
Slave Controller Pump 2 [0 -120] l/min
PIDMA 101 ts=11 Flow.->Pu 2
Output Pumpe 2
[0-100] % = [0-150] l/min
SCAL 119 ts=11
Slave Controller Pump 3 [0 -150] l/min
CONTR 102 ts=11 Flow.->Pu 3
Output Pumpe 3
X 1
X 2
X 3
Wext
OVC+
OVC-
Yp
Yhm
Yadd
ParNo
Casc
Weff
X
Y
XW
W
Yout1
Yout2
ParNo
Bl-no
Y 1

Operator interface activation for cascade control is automatic for controllers the Casc input of which is connected with the Bl-no output of another controller.

In the above example, 3 flow controllers operate as slave controllers for level control. All three slave controllers offer the operator interface for level control with a separate operating page. Master tracking during manual mode of the slave as specified for simple cascade control in the example cannot be used without detailed considerations, because

1. two further cascade branches are still intact, when one controller is in manual mode
2. it is not clear, which process value should be used for tracking, when all controllers are in manual mode

III-16.12 Setpoint functions

Terminology

w	internal setpoint
we	external setpoint
w2	second (internal) setpoint
Weff	effective setpoint
xw	control deviation (x-w → process value -)

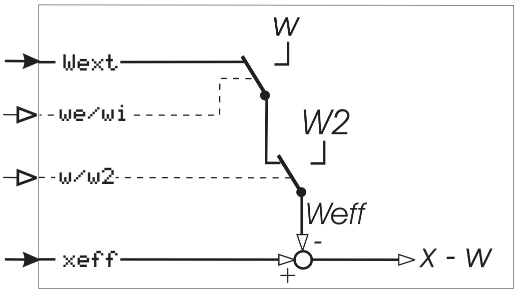
General

Several possible setpoints are available. For the priorities, see the drawing shown opposite. "Safety setpoint " w2 is given priority over the other setpoints. Switchover between setpoints is possible via interface or via the digital inputs of the conroller block.

If gradient control was activated, a change will be made effective continuously instead of being made effective by a step → gradient control page 261.

By activating digital input w stop , the instantaneously effective setpoint is maintained. In this case, neither a setpoint change nor switchover to another setpoint becomes effective.

Fig. 77



Setpoint / setpoint / cascade

Configuration word WFunc can be used to select, if switch-over to the external setpoint (setpoint / cascade) is also possible in addition to the internal setpoint.

Setpoint

(WFunc = Setpoint) Setpoint control means that the setpoint is firmly predefined by the internal setpoint W.

Setpoint/cascade

(WFunc = Fest/Folg) Setpoint /cascade control permits switchover between internal setpoint W and external setpoint We . Switchover is via digital input we/wi or via the interface.

Unless this input is connected, or if a 0 signal is applied, the external setpoint is used as effective setpoint. Unless digital input we/wi as well as analog input wext are connected, the controller invariably uses the internal setpoint.

W2 - safety setpoint

The second setpoint W2 can be activated at any time and has highest priority. The change-over between internal setpoint 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 setpoint is to be active, a logic 0 must be given on we/wi.

In the past W2 was designated as "safety setpoint". 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 setpoint Wext

Switching between the internal setpoint (wi) and the external setpoint (we) is possible only if the parameter WFunc is adjusted to W/Casc.

The change-over can be triggered via interface or the digital control input we/wi .

In order to make the internal setpoint effective, on we/wi must be attached a logic 1. If the external setpoint 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 setpoint is effective.

The internal setpoint W is evaluated with priority. If in a place (interface or the digital control input we/wi) is switched to internal setpoint it is not possible to switch over to the external setpoint Wext from another source.

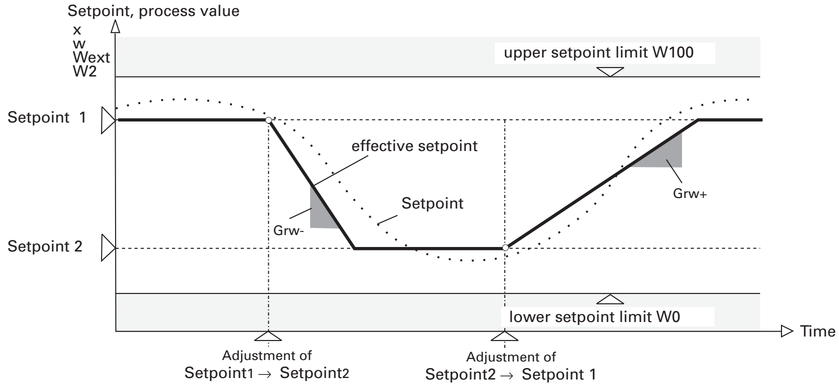
Gradient control - setpoint changes with gradients

Normally, setpoint 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 setpoint changes are made bumplessly. With digital input 'gr_off' not set, effective setpoint Weff runs linearly towards the changed setpoint (target value), whereby the slope is determined by gradients Grw+ and Grw- adjustable at parameter setting level (→ see Fig.: 78). For the second 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.78:

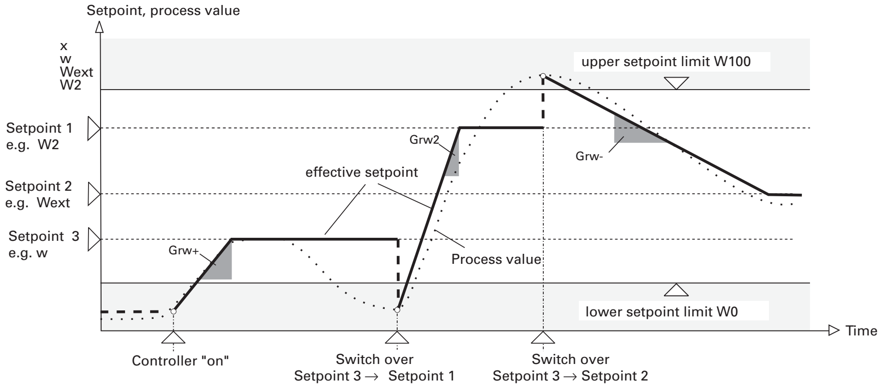


Switch-over with gradients (W→W2, W→Wext, controller"on")

The new setpoint 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 range W0/W100 at switch-over time (e.g. when starting).

Fig.79:

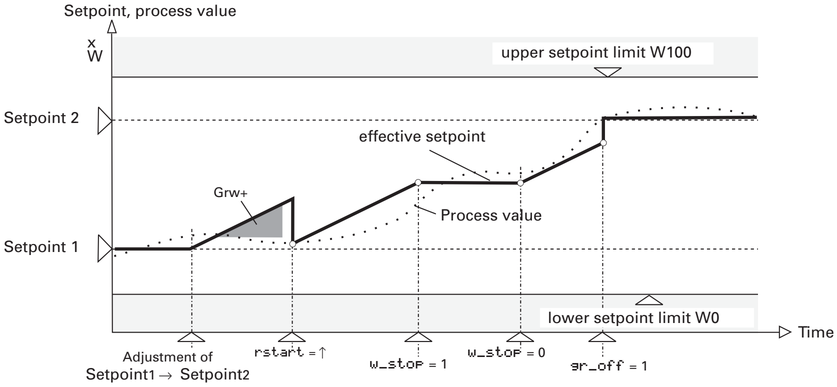


Controlling the setpoint

The digital input rstart reacts to a positive signal slope and sets the effective setpoint to the process value. The new goal 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 Weff , i.e., the effective setpoint is held to the current value, even if the effective setpoint straight approaches a new goal or a new goal is selected.

Fig. 81:



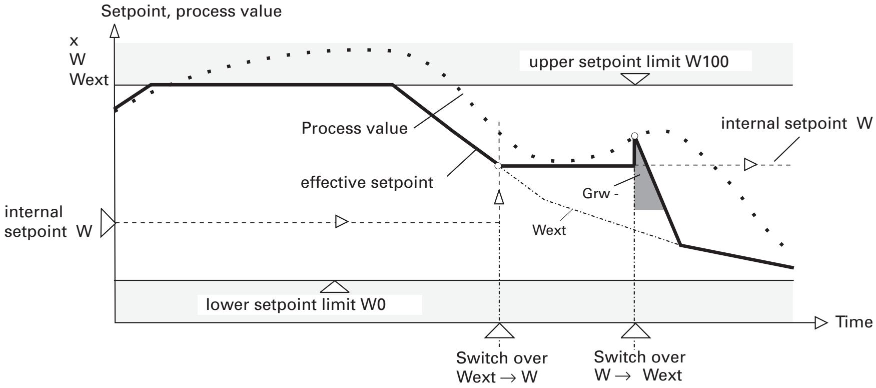
Setpoint-Tracking

During the change-over of Wext → W it can come to unwanted jumps. To aviod these jumps setpoint-tracking can be applied. Setpoint-tracking takes care that in switch-over of Wext → W, the past Wext is taken over as int. W.

The digital input "track" enables Tracking. When switching back (W → Wext) Wext is started with the attitude of Grw+/-. With the configurationword is determined if the controller shall follow process value or setpoint-tracking Wtrac. Tracking can be activated via interface or by operating the switchover Wext → W . Tracking is evaluated with priority.

If a source (interface or digital input) is activating setpoint-tracking, switching by operation switching W_{ext} → W from another source is not possible!

Fig. 80:



Process-value tracking

It can occur that the is far distant from the momentary process-value (e.g. when starting a plant).

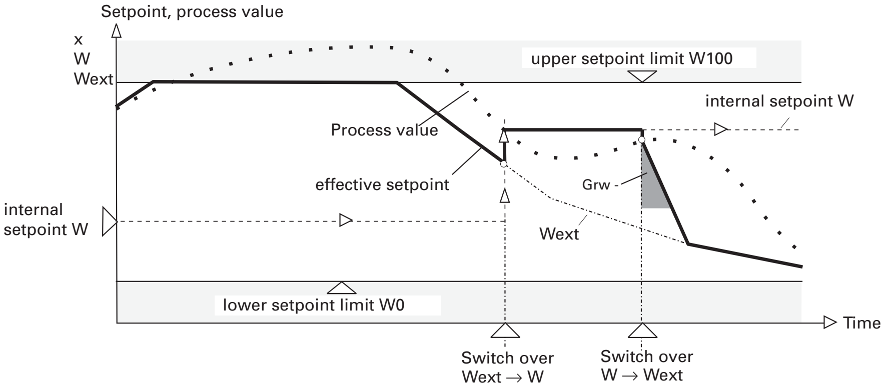
In order to prevent the jump developing here, the function process-value tracking can be used.

Process-value tracking causes a take-over of the process-value on the internal setpoint , 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 setpoint-tracking is determined with the configurationword Wtrac. The digital input "track" enables Tracking. Tracking can be activated via interface or by operating the switchover Wext → W . Tracking is evaluated with priority.

If a signal source switches to tracking, switching from another source is not possible!

Fig. 82:



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_{old} -> W_{new}	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

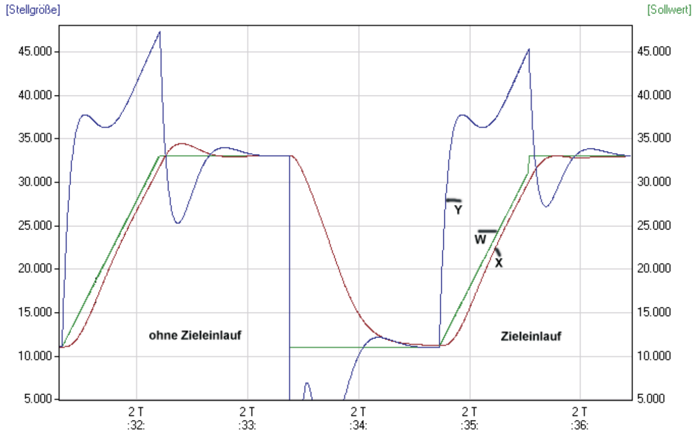
When using a setpoint ramp, a process value overshoot at the ramp end may occur. Due to the difference between setpoint and process value in the course of the ramp, an integral action is built up and must be removed after the end of the ramp. The longer the ramp, the higher the integral action. And the 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

Fig. 84



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.

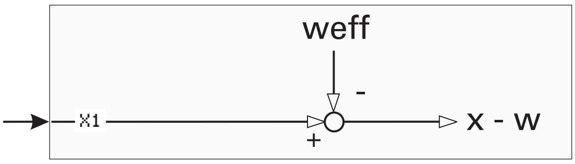
Target line-out is activated only, if the external ramp setpoint changes continuously. The function can be used both with differentiation of control deviation (XW) and differentiation of process value (X). With 3-element control, target line-out is omitted. The signification of parameter "a" is different 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.

III-16.13 Process value calculation

Standard controller

The process variable measured via analog input X1 is used as process value by the controller.

Fig. 83



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 setpoint.

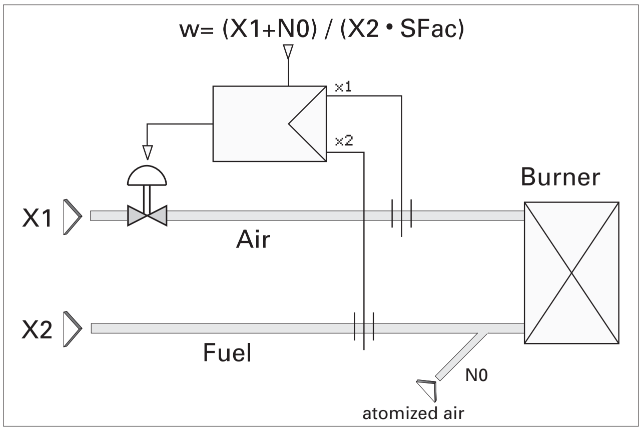
If the transmitters used by the controller are designed with a stoichiometric ratio, λ = 1 is met exactly with restless combustion.

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:

Fig. 85



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.

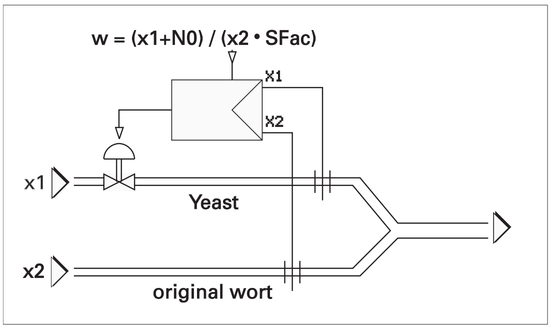
Weff effective as relative ratio is multiplied by stoichiometric factor SFac (e.g. SFac = 10), i.e. a „stoichiometric" flow ratio can be used for calculation of the control deviation. The instantaneous (controlled) process value is calculated from the physical ratio, multiplied by 1/SFac and displayed 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 $\quad W = \frac{X1 + N0}{X2 \cdot SFact}$

Fig. 86

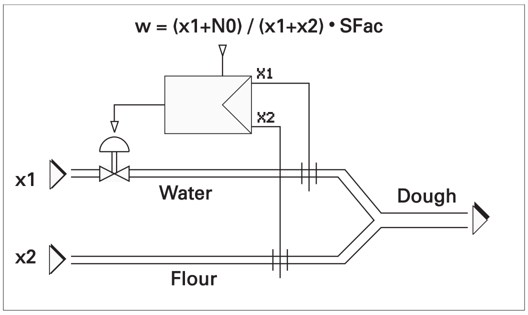


The first case is obvious. Almost everybody knows what happens during brewing.
Yeast (x1) must be batched in relation to the original wort (x2). The is adjusted in '%yeast', e.g. W= 3%. The ratio inputs are scaled in equal units. The control deviation is multiplied by 'SFac = 0,01' and calculated according 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.

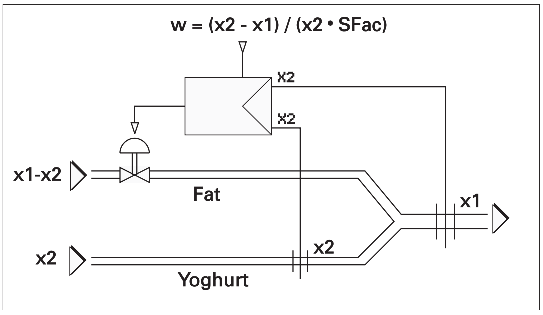
Fig. 87



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.

Fig. 88

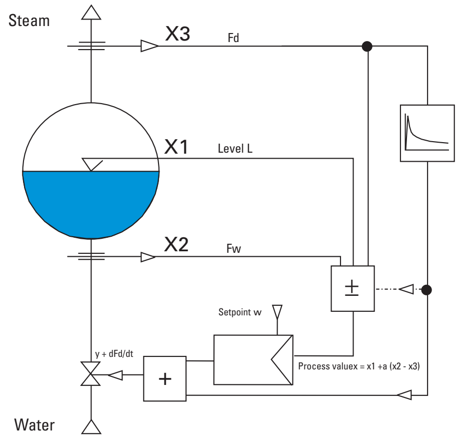


Three-element control

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.

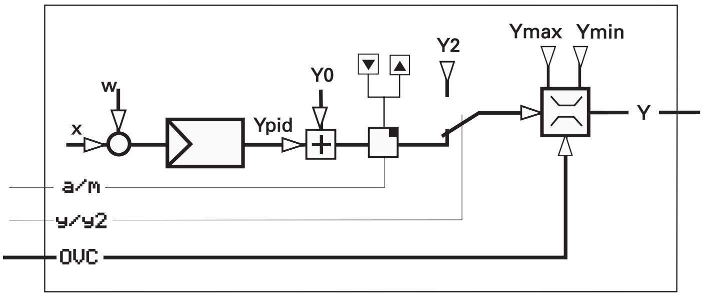
Fig. 89



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.: 90 Steps of the output signal processing



Second correcting value

Similar to 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.

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. 91 Fixed correcting variable limits

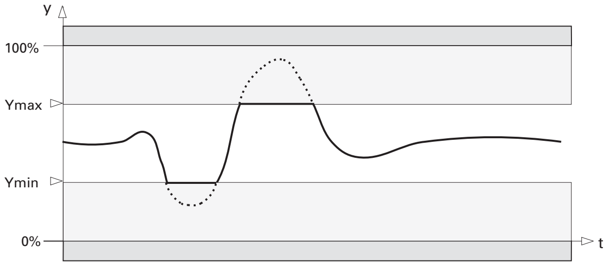


Fig.:92 Maximum value limiting

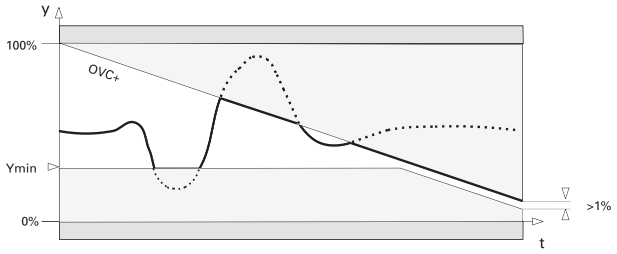


Fig.:93 Minimum value limiting

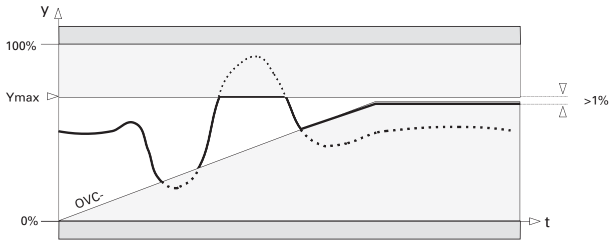
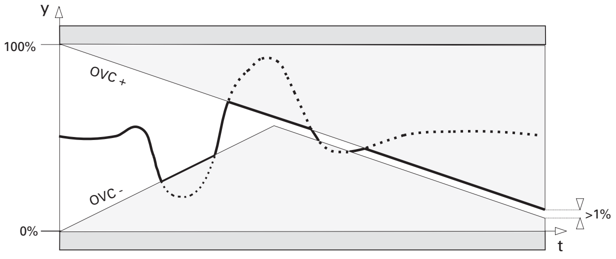


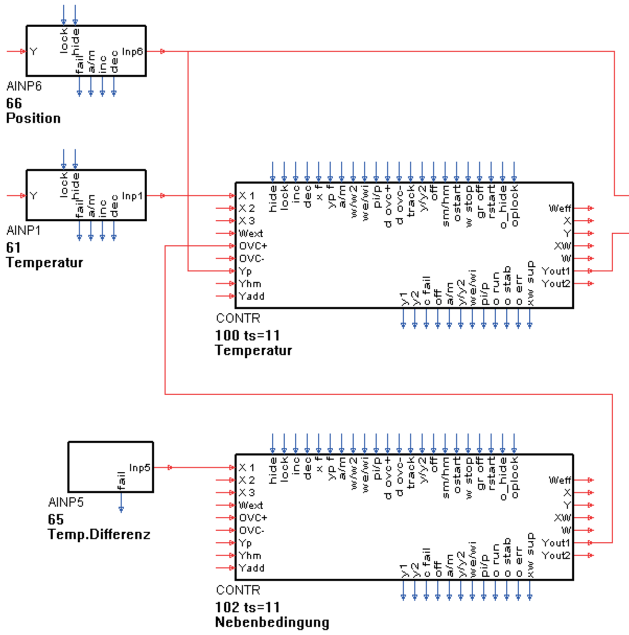
Fig.:94 Mini- and Maximum value limiting



Limiting control

Limiting with continuous output. Limiting control with three-point stepping output can be realized by using a continuous controller with the OVC function. A position controller (three-point stepping) provides override control.

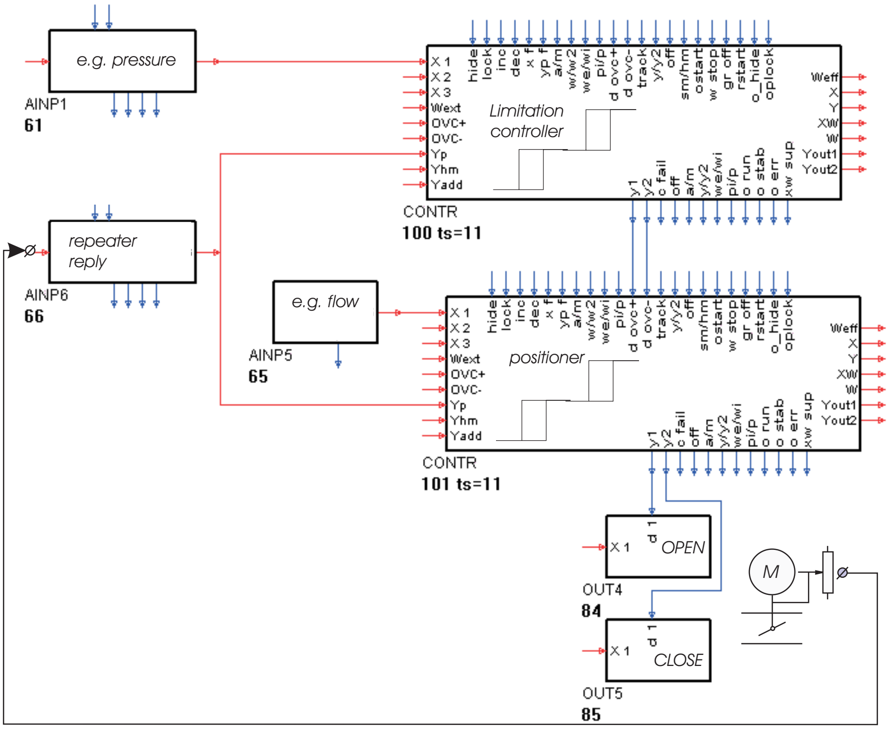
Fig.: 96 Limit control with continuous controller



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.: 95.

Fig.:95 Limit control with motor stepping controller



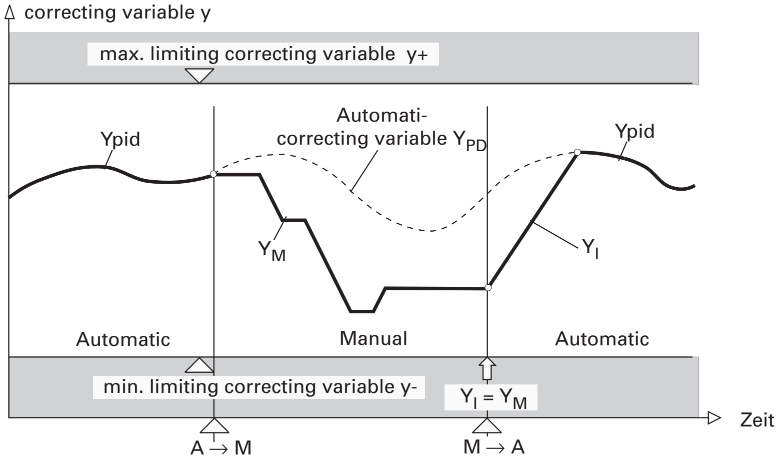
Which one of the two controllers influences the process is decided in the slave controller logic. The first “close” pulse coming from the limiting controller switches over to override control. The limited controller automatically retrieves the positioning authority, when it first tries to close the actuator further.

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 ($Y_I = Y_M + Y_{PD}$).

Fig.:97


correcting variable y
max. limiting correcting variable y+
Ypid
Automati-
correcting variable Y_{PD}
Ypid
Y_M
Y_I
Automatic
Manual
Automatic
min. limiting correcting variable y-
$Y_I = Y_M$
Zeit
A → M
M → A

Now, only the integrator, which adapts the correcting variable slowly to the stationary value according to the actual control deviation, is active.

III-16.14 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 = . 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 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 setpoint changes must be avoided. setpoint changes are bumpless in both directions, since the effective setpoint always runs towards the changed (destination) by means of gradients Grw+ or Grw-. For the second 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*.

ℹ 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 to a slave controller, which varies the output value.

✓ *Override control (OVC) → page 269*

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 setpoint 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*

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 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 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.

✓ *switch-over*

In principle, the following s are possible: internal wi, second internal w2 and external we. With program control, external we must be selected. The analog 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 (preferable) or process value is made. The evaluated and filtered output value is added to the 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 . To prevent permanent control deviations, the limit contact must be further away from the than the permanent control deviations.

✓ *Tracking*

During switch-over from external or program setpoint to internal setpoint, setpoint 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 setpoint. Setpoint-tracking: During switch-over, the external or program setpoint used so far is taken over as internal setpoint.

✓ *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*

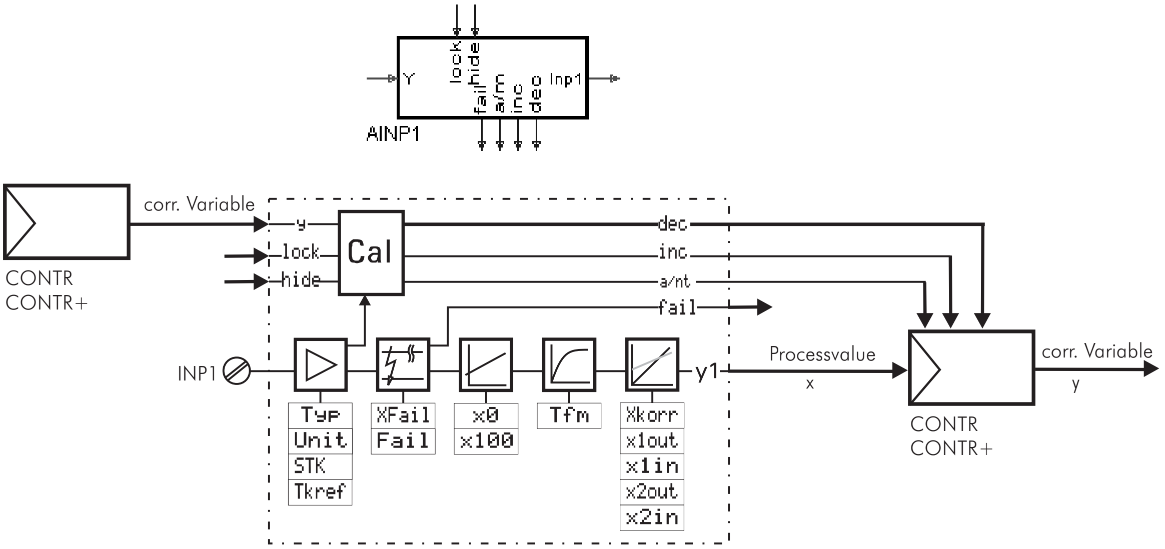
Dynamic changes of process value or setpoint affect control differently. x-differentiation: Process value changes (disturbances) are used dynamically to permit better control results. xw-differentiation: Changes of process value (disturbances) and setpoint are used dynamically to permit a better control 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“).

III-17 Inputs

III-17.1 AINP1 (analog input 1 (No. 08))



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

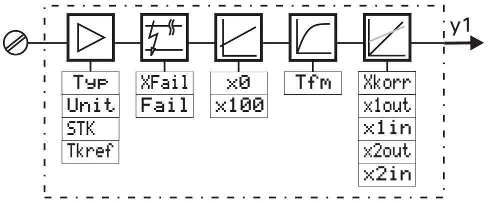
Parameter and configuration data

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	°C	←
	Unit = °F thermocouple and Pt100 setting	°F	
STK	Internal temperature compens. only effective with	int.CJC	←
	External temperature compens. thermocouple	ext.CJC	
x0	Physical value at 0% only effective with standard	-29999 ... 999 999	0
x100	Physical value at 100% signals (0/4..20mA or 0/2..10V)	-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 (control)
-minimization of linearity errors in a limited operating range (variable)
-correspondence with other measuring facilities (recorders, indicators, PLCs, ...)
-compensation of sample differences of sensors, transmitters, etc.

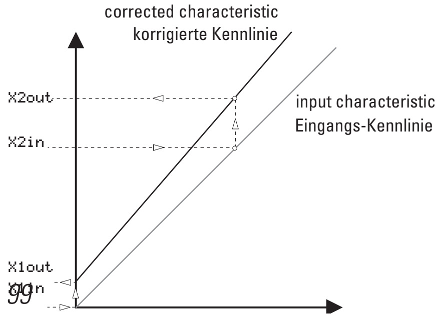
Measured value correction is designed for zero offset, gain matching and for both. It corresponds to scaling mx+b, with the difference that the KS 98-1 firmware calculates gain m and zero offset b from the value pairs for process value (x1in; x2in) and (x1out; x2out) of two reference points.

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.

Fig. 98

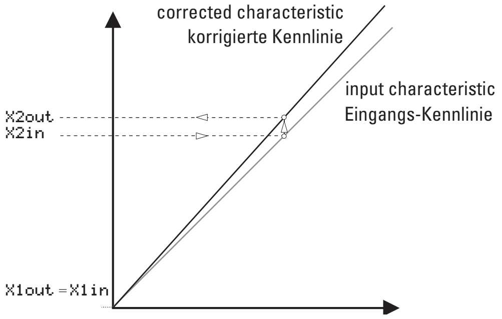


Example 2:
Gain change (rotation around the coordinate origin)

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.

Fig. 99



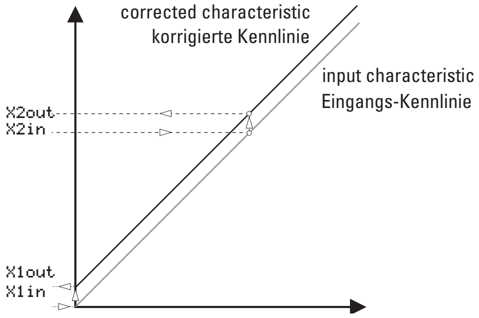
Example 3:
zero and gain matching

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.

Fig. 100



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 (<< process value) or downscale (>> 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 10Ω 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 (<< process value)
Downscale (>> 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_L 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.
See also Page 35 .
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.

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.

Calibration is as described in section 10.4. The user-specific calibration values are stored in configurations X0, X100.

X0, X100 application principle

Configurations X0, X100 are used in different ways dependent on the input type:

- Current input: X0, X100 are scaling values of the signal source (e.g. temp. transmitter): 0 mA = X0, 20 mA = X100.
- Potentiometric transducer input: X0, X100 are the user-specific calibration values. Within range X0, X100, display of 0...100% input signal is required.
- Temperature difference input: X0, X100 contain the lead resistance values after calibration by the user: X0 is the lead resistance of connected sensor 1. X100 is the lead resistance of connected sensor 2.

X0 and X100 are parameters of function block AINP1, i.e. part of the engineering.
This means that the user calibration remains unchanged after replacing the unit, if it was reloaded into the engineering after calibration.

$$\%\text{Wert} = \frac{X - AIOL}{AI1 - AIOL}$$

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 behav - iour 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)

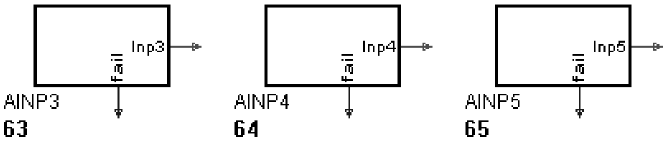
0/2...10 V voltage signal input

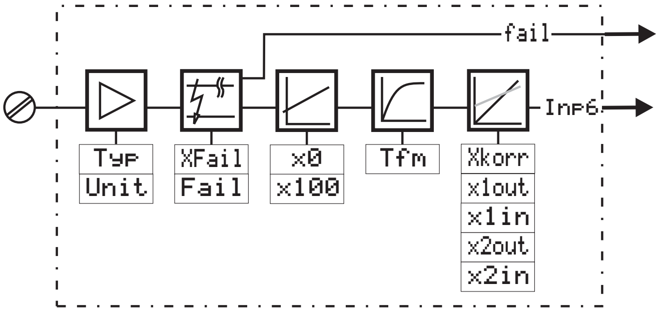
The input resistance is ≥ 100 kΩ

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)

III-17.2 AINP3...AINP5 (analog inputs 3...5 (No. 112...114))

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.

For the general functions (scaling, error control, filter...) see AINP1 ⟶ Page 275

Inputs / outputs

Digital outputs:	
fail	Signals an input error (short-circuit, wrong polarity, ..)

Analog outputs:	
Inp1	Signal input

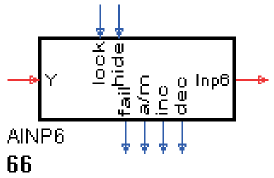
Parameter and configuration data

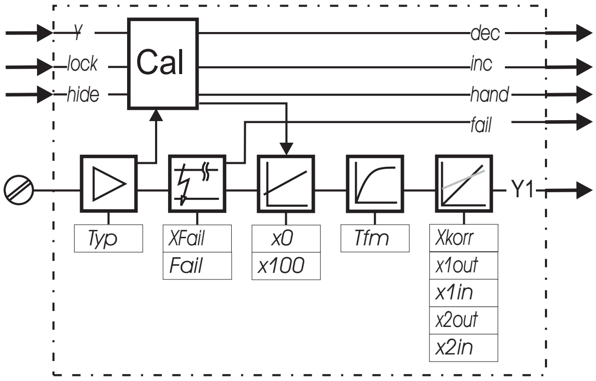
Parameter	Description	Values	Default
x1in	Measured value correction P1, input	-29999 ... 999 999	0
x1out	Measured value correction P1, output	-29999 ... 999 999	0
x2in	Measured value correction P2, input	-29999 ... 999 999	100
x2out	Measured value correction P2, output	-29999 ... 999 999	100

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

III-17.3 AINP6 (analog input 6 (No. 115))

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	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	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 (control)
- minimization of linearity errors within a limited operating range (variable)
- 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 KS 98-1 firmware calculates gain m and zero offset b from the defined value pairs for process value (x1in; x2in) and (x1out; x2out) of two reference points.

☞ For a comparative measurement with a calibrated measuring device the standard values for für x1in, x1 out (0) and x2 in, x2 out (100) must be entered first.

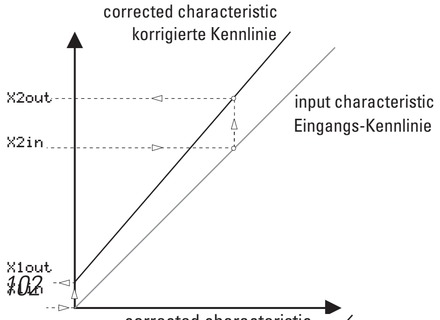
Example 1:
Zero offset

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.

Fig. 101



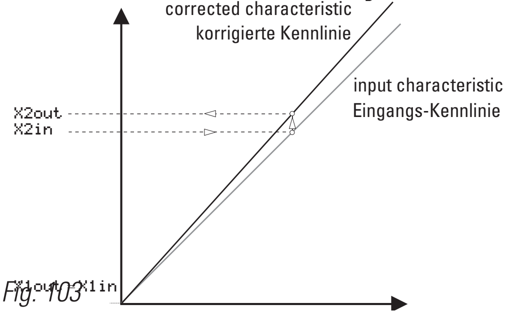
Example 2:
Gain change (rotation around the coordinate origin)

x1in = 0 x1out = 0

x2in = 300 x2out = 300 + 1,5

The corrected values diverge despite equality with the input values at x1in and x1out.

Fig. 102



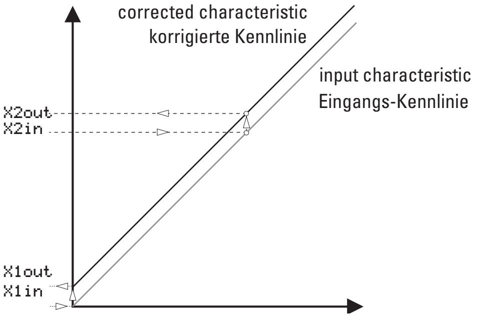
Example 3:
Zero and gain matching

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.

Fig. 103



Sensor types

The input sensor type can be defined as potentiometric transducer or as standard current signal.

Potentiometric transducer

The admissible overall resistance is ≤ 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%. 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}.

Input 6 to which a potentiometric transducer is connected features a particularity: only the wiper resistance rather than the overall resistance is measured. For this reason, the internal parameter setting and calculation of the input 6 transducer is not identical with the one of input 1.

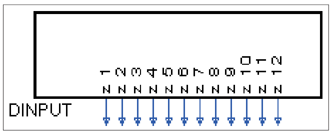
These calibration values belong to the engineering, i.e. when replacing the instrument, the user calibration remains unchanged. As a prerequisite, it has to be re-loaded into the engineering after calibration.

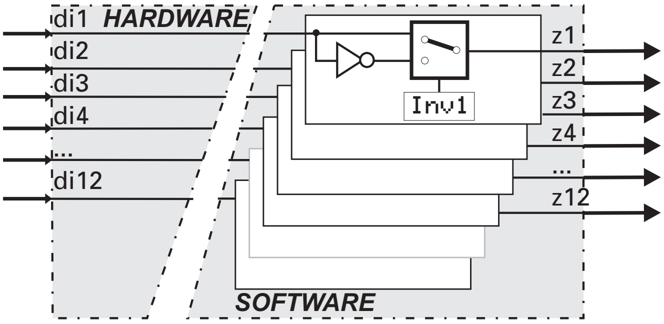
Standard 0/4...20 mA current input

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)

III-17.4 DINPUT (digital inputs (No. 121))





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 KS 98-1 hardware option.

Outputs

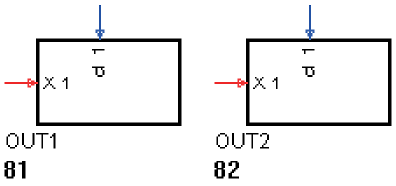
Digital outputs:	
z1...z2	Signal at digital input di1 or di2 (Digital inputs di1 and di2 are available in each unit also without options).
z3...z7	Signal at digital input di1 or di2 (Digital inputs di3 to di7 are only provided with option B).
z8...z12	Signal at digital input di1 or di2 (Digital inputs di8 to di12 are only provided with option C).

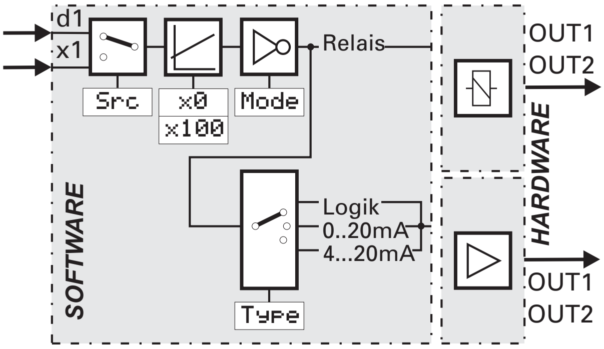
Parameter and configuration data

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	

III-18 Outputs

III-18.1 OUT1 and OUT2 (process outputs 1 and 2 (No. 116, 117))





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

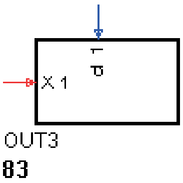
Digital input:	
d1	Input signal with digital signal conversion

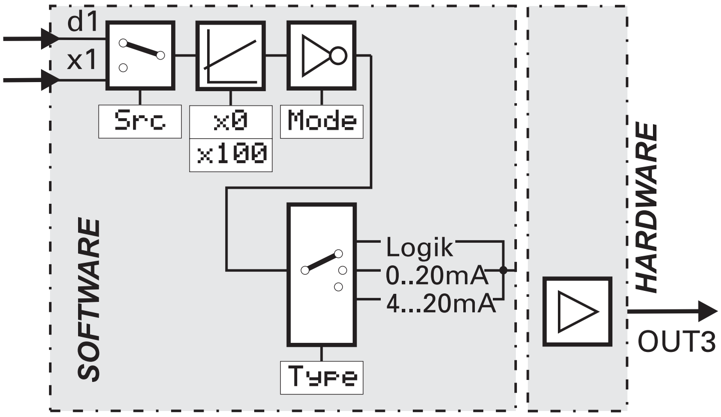
Analog input:	
x1	Input signal with analog signal conversion

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

III-18.2 OUT3 (process output 3 (No. 118))





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 output.

Analog input x1 used as signal source is taken to the continuous output linearly between x0 and x100 according to configuration.

Inputs / outputs

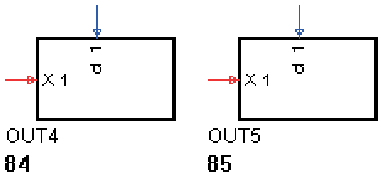
Digital input:	
d1	Input signal with digital signal conversion

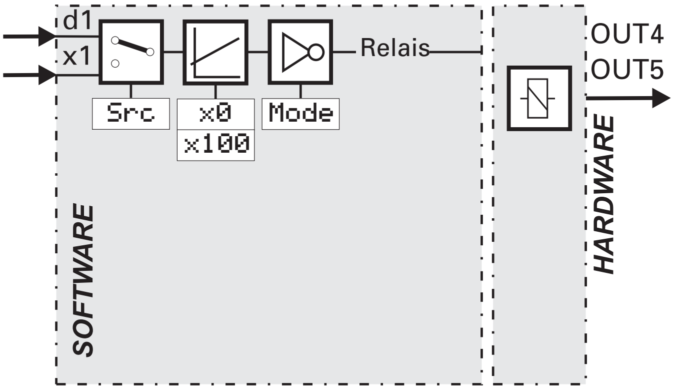
Analog input:	
x1	Input signal with analog signal conversion

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

III-18.3 OUT4 and OUT5 (process outputs 4 and 5 (No. 119...120))





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

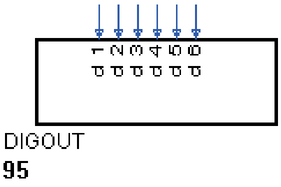
Digital input:	
d1	Standard signal with digital signal conversion

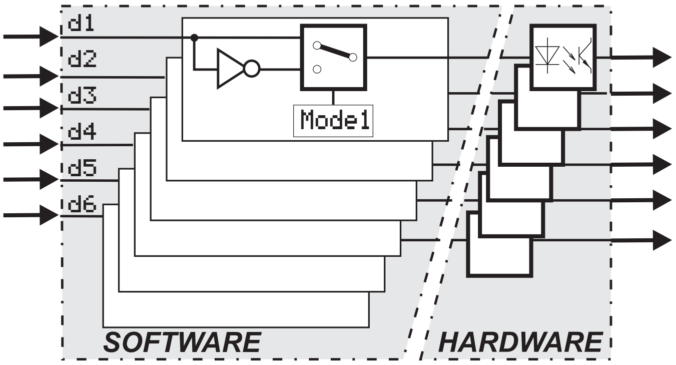
Analog input:	
x1	Input signal with analog signal conversion

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

III-18.4 DIGOUT (digital outputs (No. 122))





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 KS 98-1 hardware option.

Inputs

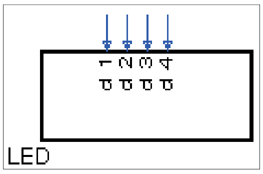
Digital inputs:	
d1 ...d4	Signal sources for control of digital outputs do1 to do4. (Digital outputs do1 to do4 are provided only in units with hardware option B).
d1 ...d4	Signal sources for control of digital outputs do5 and do6. (Digital outputs do5 and do6 are provided only in units with hardware option C).

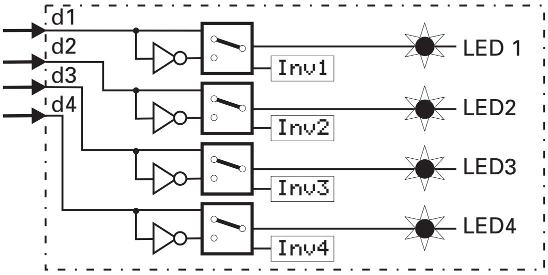
Parameter and configuration data

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	

III-19 Additional functions

III-19.1 LED (LED display (No. 123))





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:

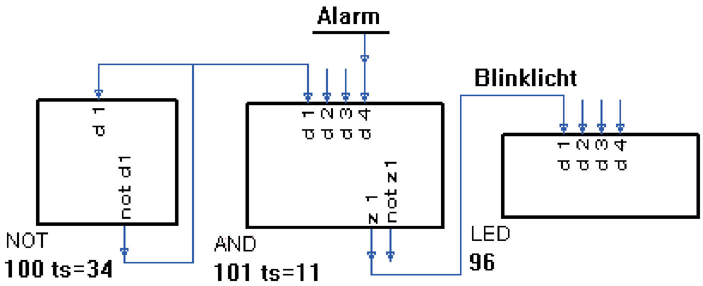
Input	Description
d1	LED 1
d2	LED 2
d3	LED 3
d4	LED 4

Parameters:

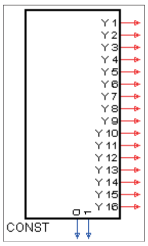
Parameter	Description	Range	Default
Inv 1	Inv1= 0 ≙ d1=1 LED1 is lit Inv1 = 1 ≙ d1= 0 LED1 is lit	0...1	0
Inv 2	Inv2= 0 ≙ d2=1 LED2 is lit Inv2 = 1 ≙ d2= 0 LED2 is lit	0...1	0
Inv 3	Inv3= 0 ≙ d3=1 LED3 is lit Inv3 = 1 ≙ d3= 0 LED3 is lit	0...1	0
Inv 4	Inv4= 0 ≙ d4=1 LED4 is lit Inv4 = 1 ≙ d4= 0 LED4 is lit	0...1	0

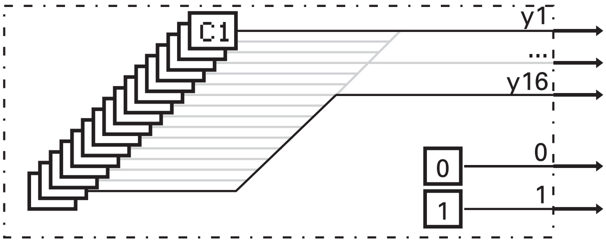
Example:

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.



III-19.2 CONST (constant function (No. 126))





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:

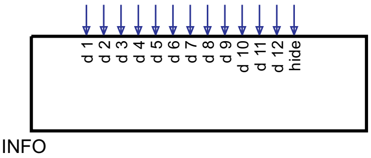
Digital outputs	
0	Logic 0 is always output at this output.
1	Logic 1 is always output at this output.

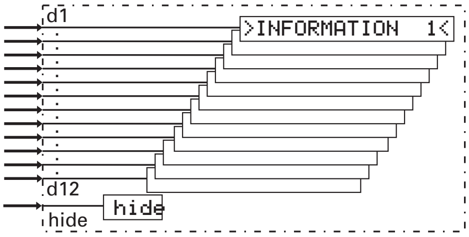
Analog outputs	
y1	Constant C1 is output.
y2	Constant C2 is output.
y3	Constant C3 is output.
y4	Constant C4 is output.
y5	Constant C5 is output.
y6	Constant C6 is output.
y7	Constant C7 is output.
y8	Constant C8 is output.
y9	Constant C9 is output.
y10	Constant C10 is output.
y11	Constant C11 is output.
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

III-19.3 INFO (information function (No. 124))





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.
The user texts are displayed on the operating pages and the operating page list.
Display of all INFO texts can be suppressed by setting the Hide signal.

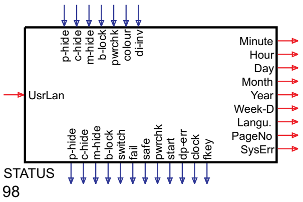
Inputs:

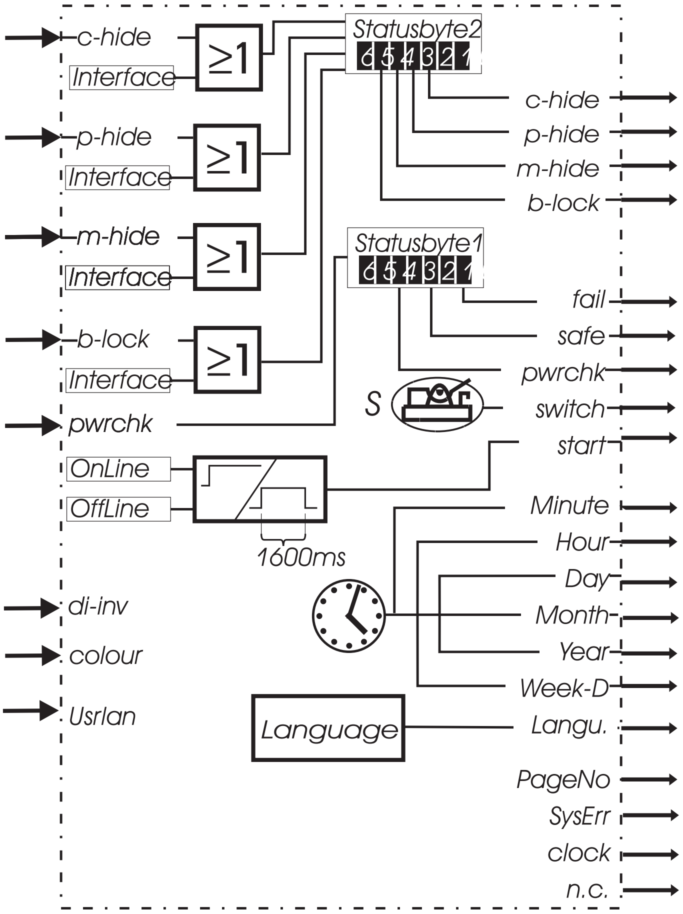
Digital inputs	
d1	=1 → the information configured in Text 1 is displayed.
d2	=1 → the information configured in Text 2 is displayed.
d3	=1 → the information configured in Text 3 is displayed.
d4	=1 → the information configured in Text 4 is displayed.
d5	=1 → the information configured in Text 5 is displayed.
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.
hide	=1 → all INFO texts are hidden, i.e. not displayed.

Parameters:

Parameter	Description	Range	Default
Text1 ... Text12	User text with max. 16 characters each	alphanumeric characters	>INFORMATION 1 < ... >INFORMATION 1 <

III-19.4 STATUS (status function (No. 125))





The function provides information from the KS 98-1 instrument status byte at its digital outputs. The block is fixed to 98 and updated per time slot.

digital Inputs	
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.
pwrchk	= 1 → Monitoring for temporary power failure is activated. See output pwrchk.
colour	green = 0, red = 1. → The display background colour is changed.
di-inv	Display is inverted (background / text&graphics).

Analog inputs	
UsrLan	Change to user language. Switch-over between text blocks connected via the language input. Three languages are selectable with 0 .. 2.

Output	Description
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
PageNo	Output of the number of the function block the operating page of which is displayed instantaneously. "0" means that no operating page is displayed.
SysErr	Start-up problem. Unlessthis output is "0", a start-up error occurred (corresponds to KS 98-1 start-up error display). Bit assignment: Bit 1 = reset command, bit 2 = quartz, bit 4 = halt, bit 5 = SW watchdog (endless loop); the other bits are not used.

Digital outputs	
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
switch	S.I.L. switch open = 0 closed = 1. This information permits blocking via the hardware.
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 reset (0) by KS 98-1 after power-on. It can be set to active (1) by a signal at the digital input to permit detection of a temporary power failure.
start	With change from offline to online, start is 1 during 800 ms. During this time, all time groups were calculated at least once.
dperr	Collected error messages Profibus
clock	1 = real-time clock provided, 0 = no real-time clock option.
fkey	Status of the function key

[1]) These outputs are = 0 if real-time clock option B with RS 422 is missing.

Powerup KS98/98-1

Start-up behaviour after power recovery.

The procedure is as follows:

① All blocks are initialized. Unless a special start-up behaviour is configured (see programmer), the two possibilities are:
a. The memory contents are still unchanged. The block outputs keep the value before power failure.
b. After prolonged power failure, the memory contents are destroyed. The function blocks are initiated regardless of the function inputs.

② All input functions are calculated once.

③ The start bit of the status block is set to 1.

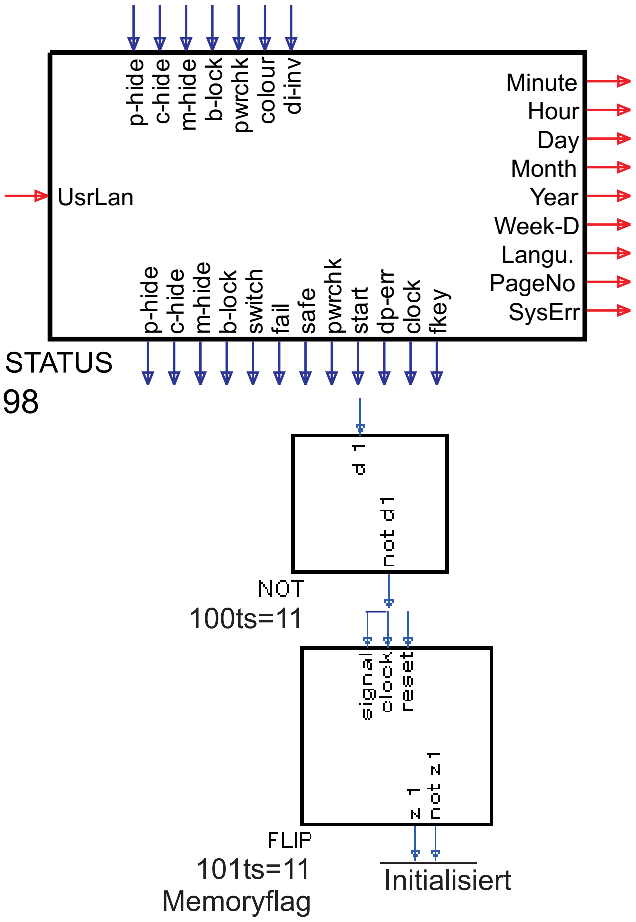
④ All blocks are calculated at intervals of 1.6 seconds (16 cycles of the 100ms time slot) in the order of block numbers.

⑤ The start bit of the status block is set to 0.

In the event of problems due to the start-up sequence, 2 items can be of importance:

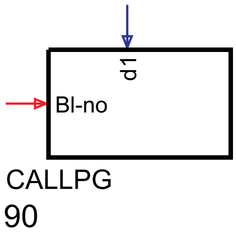
① After power failure, KS 98 /98-1 continues running during fractions of seconds and may still detect already de-activated process signals.

② If the behaviour after power recovery is different depending on whether the function blocks are still in the previous condition or whether they were initialized, the following engineering can provide the initialization information within the first 1.6 seconds after starting up.



Another possibility for power failure detection is to set an internal flag in the status block via digital input “pwrchk” at the end of the initialization phase. This Flag is available at digital output “pwrchk". After power failure, this flag and thus digital output “pwrchk” is always initialized with 0.

III-19.5 CALLPG (Function for calling up an operating page (no. 127))



With function block CALLPG, which can be used only once, event-triggered call-up of a particular operating page is possible, unless an operation is being carried out on this page (waiting time 5 s). The required operating page is determined by the number of its function block. The block number is applied to input BI-no of CALLPG.

Switch-over is with the positive edge of the logic signal at digital input d1 of CALLPG. It permits e.g. changing to a particular operating page in case of an exceeded limit value.

Exceptions: Switching over is omitted with:

active operation by the operator. Page changing is delayed and occurs only 5 seconds after the last key pressure.

a wrong page number, or if the page is blocked at activation time.

Unless the page which should be activated is available the page survey is displayed.

When leaving the operating page called up via CALLPG, the previously active operating page is displayed again.

Following functions blocks have an operating page APROG, DPROG, CONTR, CONTR+, PIDMA, VWERT, VBAR, VTREND, VPARA, ALARM

☞ With activation by CALLPG from an already selected page, this page is not called up again. I. e. if a sub-page was selected, the multifunction unit remains on this page.

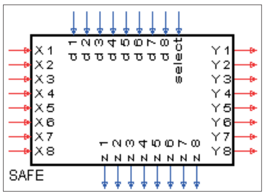
☞ With multiple page changing by activation of CALLPG, the sequence is not buffered. After leaving the page(s) activated by CALLPG the initial menu page is displayed again.

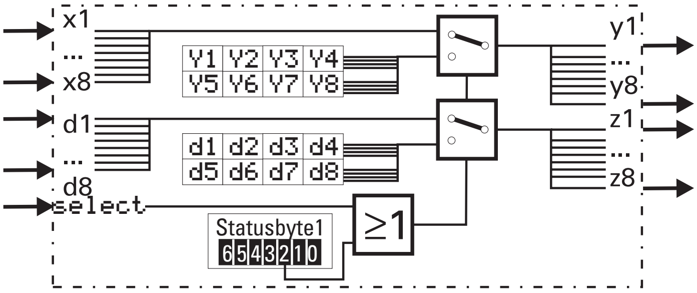
☞ In case of CALLPG whilst the multifunction unit is not at operating level (main menu: Parameter, .., Miscellaneous), CALLPG call remains active in the background. Changing to the operating page activated by CALLPG is done when selecting the operating level for the next time.

Digital input	
d1	Positive edge causes change to the operating page set at BI-no

Analog input	
Bl-no	Number of the operating page which should be displayed

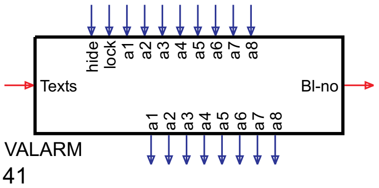
III-19.6 SAFE (safety function (No. 94))





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.

III-19.7 VALARM (display of all alarms on alarm operating pages (function no. 109)



General

Function block VALARM handles up to 8 alarms. Alarms are displayed and can be acknowledged, if acknowledgement via parameter setting is required. The alarm conditions are determined by digital inputs a1 ... a8 (0 alarm condition off, 1 alarm condition on).

Inputs/outputs

Digital inputs:	
hide	Display of this alarm operating page is suppressed
lock	Operation of this operating page is disabled, i.e. alarm acknowledgement is not possible
a1...a8	Alarm inputs alarms 1 ... 8

Analog output	
BL-no	Number of this block

Digital outputs	
a1...a8	=1 means that alarm 1 must be acknowledged

Parameter und Konfigurationsdaten

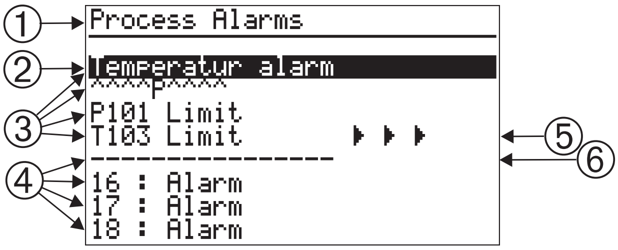
Parameter	
a1 ... a8	=1 means that alarm 1 must be acknowledged

Parameter	Description	Values	Default
Typ_a1 ... Typ_a8	Alarm function	Quit	←
		noQuit	

VALARM operating page

- ① Title
- ② Active alarm selected for acknowledging
- ③ Active alarms with texts from the text-function
- ④ Active alarms with standardtexts
- ⑤ Alarm no more active, but not acknowledged
- ⑥ Alarm no more active (if the display is rebuilded with -key , this alarm is no more displayed)

Fig. 104



```
Process Alarms
Temperatur alarm
^^^^P^^^^^
P101 Limit
T103 Limit      ▸ ▸ ▸
----------------
16 : Alarm
17 : Alarm
18 : Alarm
```

Several alarm blocks can be positioned. Block numbers 41-46 are available for this purpose. When using several alarm blocks, display of all alarm blocks except one should be suppressed, because all alarms including those of the blocks which are not selected are listed on each VALARM operating page.

The operating page title indicates the instantaneously selected block. With alarm blocks, user language selection related to the title is possible by enabling another block with a language-dependent title dependent on selected user language.

In order to avoid disturbance of the user-specific menu structure, the alarm page is displayed at the end of the operating page list independent of its block number.

Lines on the alarm page contain entries corresponding to the following classification.

• No alarm :	not existing or marked as deleted "———-" refreshed at next page actualization
• Alarm active :	Line blinks on the operating page
• Alarm active and acknowleged:	Normal display on the operating page
• Alarm not active , no ackn.:	Normal display with ">>>" at the end of the display line

The alarms are displayed with the definable name in the order of occurrence. The displayed name is taken from two text blocks which should be connected with the ALARM block. Without connected text blocks, the alarm number is displayed. The alarm numbers are calculated from block number -40 and the number of the digital input. The block numbers are 41-46, i.e. the 3rd alarm in block 41 (1st block) becomes 13.

To prevent alarm position changing, "————-" is displayed for disappearing alarms. New pending alarms are displayed only when rebuilding up the page. Build-up can be started also by pressing key .

☞ The combination of digital input and output signals indicates the four statuses of an alarm: active + not acknowledged (di=1, do=1), active and acknowledged (di=1, do=0), not active any more and not acknowledged (di=0, do=1), and not active or not active any more and acknowledged (di=0, do=0).

III-20 Modular I/O - extension-modules

The modular C-Card offers the possibility, to adjust the kind and type of process in- and outputs to the plant. Despite of following limitation, up to four modules can be fitted as desired.

Performance limits

Due to the maximum permissible self-heating, the number of analog output modules is limited. The total of performance factors (P-factor, → Technical data must not exceed 100%! Exceeded performance limits are displayed in the engineering tool.

Performance factor of the modules

R_INP	TC_INP	U_INP	F_INP	DIDO	U_OUT	I_OUT
= 5%	= 5%	= 8%	= 8%	= 15%	= 25%	= 70%

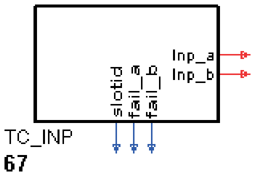
This means:

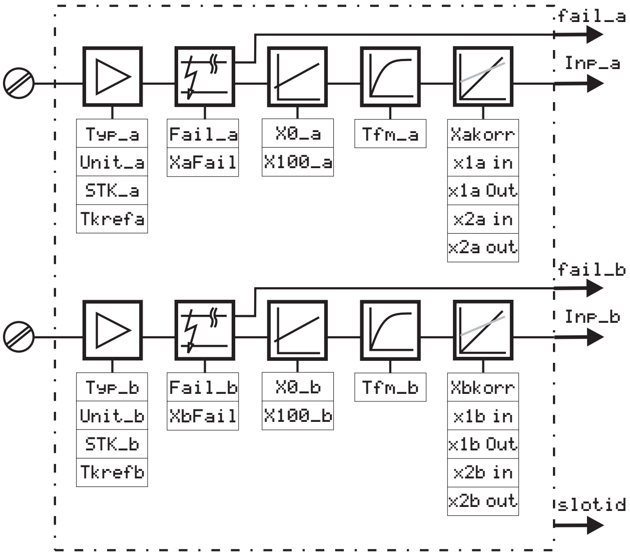
Max. one I_OUT-module (any socket)!
Max. one U_OUT-module, if there is already a current output module (but in different, galvanically isolated module groups)!

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.
Galvanic isolation: Slots 1-2 are galvanically isolated from 3-4.

III-20.1 TC_INP (analog input card TC, mV, mA)





Analog input, plugs into modular options card C. The function TC_INP is used for configuration and parameter setting of the analog inputs TC_INP. Calculation of the inputs is fixed to once per time slot.

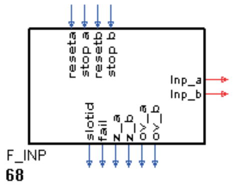
Digital outputs:	
slotid	0 = correct module fitted
	1 = wrong module fitted
fail_a	0 = no measurement error at channel a detected
	1 = measurement error at channel a detected; e.g. sensor break
fail_b	0 = no measurement error at channel b detected
	1 = measurement error at channel b detected; e.g. sensor break

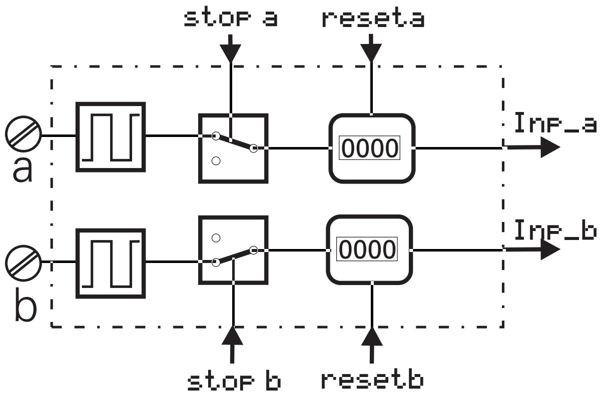
Analog outputs:	
Inp_a	measurement value channel a
Inp_b	measurement value channel b

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

III-20.2 F_Inp (frequency/counter input)





The frequency/counter input plugs into the modular options card C.The F_INP is used for configuration and parameter setting of input F_INP. Input calculation is fixed to once per time slot.

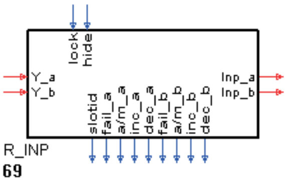
Digital inputs:	
reset a	1 = the value for Inp_a is reset to 0.
stop a	1 = the instantaneous value for Inp_a remains unchanged.
reset b	1 = the value for Inp_b is reset to 0.
stop b	1 = the instantaneous value for Inp_b remains unchanged.

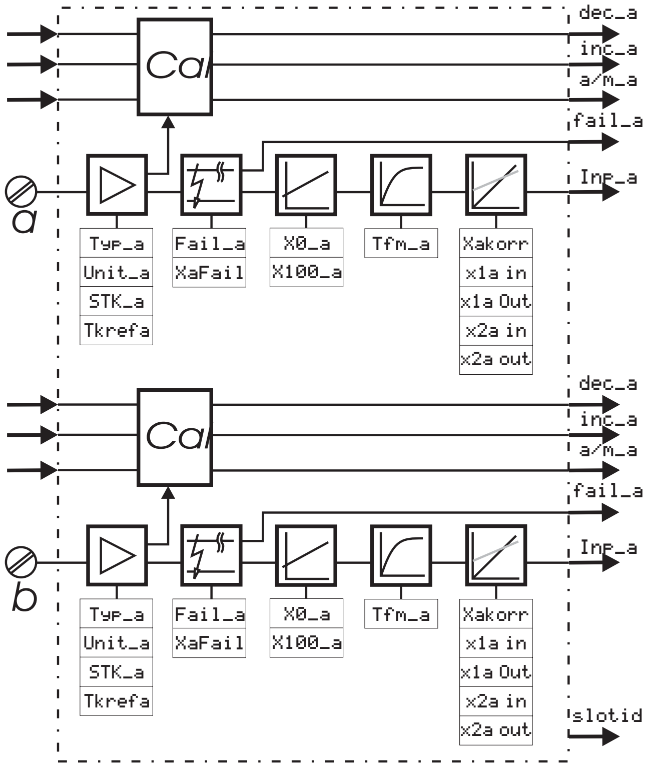
Digital outputs:	
slotid	0 = correct module fitted
	1 = wrong module fitted
fail	1 = inserted module is detected, but no communication to the module.
z_a	signal status of HW input a
z_b	signal status of HW input b
ov_a	1 = frequency at HW input a exceeds the maximum permissible 20kHz
ov_b	1 = frequency 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→ con-trol 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→ fre-quency measuring	5	
Func_b	DigInput→ con-trol input	0	1
	Count_1 → up counter	1	
	Frequenz → frequency measuring	5	
Time	for frequency measuring in seconds	0,1...20	10

III-20.3 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. The R_INP is used for configuration and parameter setting of analog inputs R_INP. Input calculation is fixed to once per time slot.

Digital inputs:	
lock	calibration disabled
hide	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

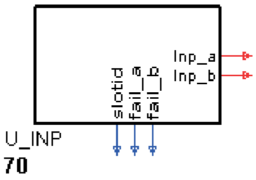
Analog inputs:	
Y_a(b)	position feedback

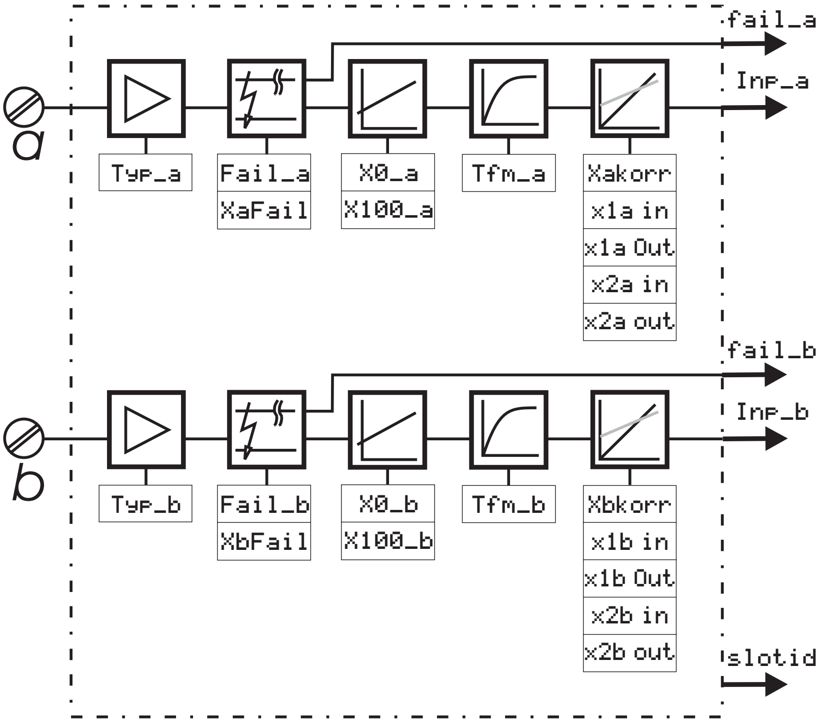
Analog outputs:	
Inp_a	measured value channel a
Inp_b	measured value channel b

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

III-20.4 U_INP (analog input card -50...1500mV, 0...10V)





Analog input, plugs into modular options card. The U_INP is used for configuration and parameter setting of the analog input U_INP. Input calculation is fixed to once per time slot.

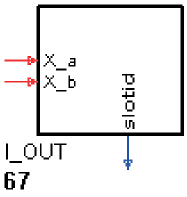
Digital outputs:	
slotid	0 = correct module fitted
	1 = wrong module fitted
fail_a	0 = no measurement error at channel a detected
	1 = measurement error at channel a detected; e.g. sensor break
fail_b	0 = no measurement error at channel b detected
	1 = measurement error at channel b detected; e.g. sensor break

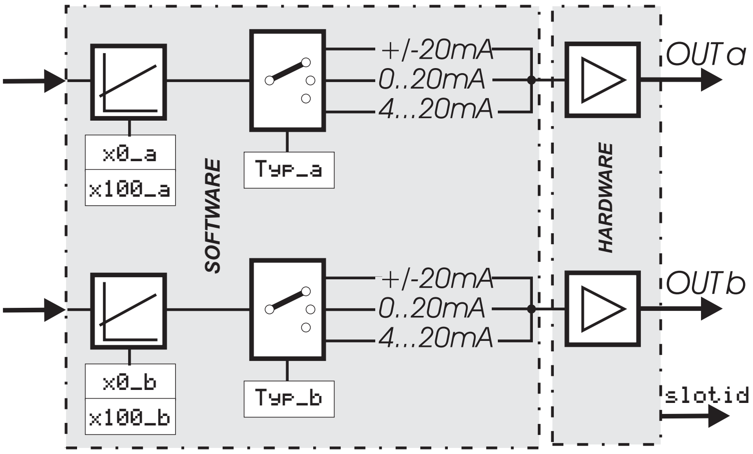
Analog outputs:	
Inp_a	measured value channel a
Inp_b	measured value channel b

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	0	0
	Voltage -50...1500mV	1	
Fail_a	Switched off	0	1
	Upscale, Inp_a = x100_a	1	
	Downscale, Inp_a = x0_a	2	
	Substitute value, Inp_a = XaFail	3	
Xakorr	Measured value correction Inp_a switched off	0	0
	Measured value correction Inp_a effective	1	
Typ_b	Voltage 0...10V	0	0
	Voltage -50...1500mV	1	
Fail_b	Switched off	0	1
	Upscale, Inp_b = x100_b	1	
	Downscale, Inp_b= x0_b	2	
	Substitute value, Inp_b = XbFail	3	
Xbkorr	Measured value correction Inp_b switched off	0	0
	Measured value correction Inp_b effective	1	
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

III-20.5 I_OUT (analog output card 0/4...20mA, +/-20mA)





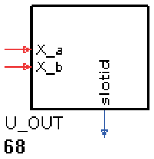
Analog output, plugs into modular options card C. The I_OUT is used for configuration and parameter setting of analog output I_OUT. Output calculation is fixed to once per time slot.

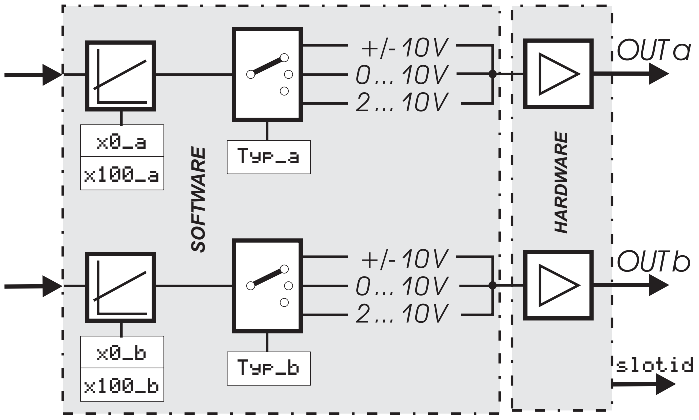
Digital output:	
slotid	0 = correct module fitted
	1 = wrong module fitted

Analog inputs:	
X_a	output value for channel a
X_b	output value for channel b

Configuration	Description	Values	Default
Typ_a	0...20mA	0	0
	4...20mA	1	
	+/-20mA	2	
x0_a	Physical value Inp_a at 0%	Real	0
x100_a	Physical value Inp_a at 100%	Real	100
Typ_b	0...20mA	0	0
	4...20mA	1	
	+/-20mA	2	
x0_b	Physical value Inp_b at 0%	Real	0
x100_b	Physical value Inp_b at 100%	Real	100

III-20.6 U_OUT (analog output card 0/2...10V, +/-10V)





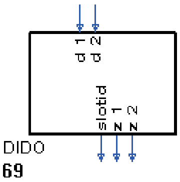
Analog output, plugs into modular options card C. The U_OUT is used for configuration and parameter setting of analog output U_OUT. Output calculation is fixed to once per time slot.

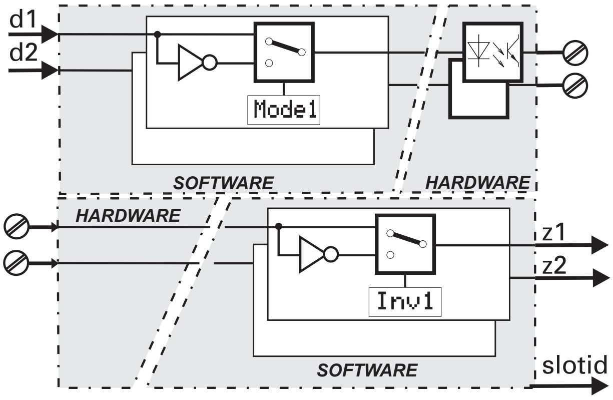
Digital output:	
slotid	0 = correct module fitted
	1 = wrong module fitted

Analog inputs:	
X_a	output value for channel a
X_b	output value for channel b

Configuration	Description	Values	Default
Typ_a	0...10V	0	0
	2...10V	1	
	+/-10V	2	
x0_a	Physical value Inp_a at 0%	Real	0
x100_a	Physical value Inp_a at 100%	Real	100
Typ_b	0...10V	0	0
	2...10V	1	
	+/-10V	2	
x0_b	Physical value Inp_b at 0%	Real	0
x100_b	Physical value Inp_b at 100%	Real	100

III-20.7 DIDO (digital input/output card)





Digital input/output card, plugs into modular options card C. The DIDO is used 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	

III-21 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 **Help / Statistics**.

III-21.1 Memory requirement and calculation time

Function	Time %	Memory%
Scaling and calculating functions		
ABSV	0,4	0,2
ADSU	0,9	0,3
MUDI	0,9	0,3
SQRT	1,3	0,2
SCAL	3,2	0,2
10EXP	3,0	0,2
EEXP	1,6	0,2
LN	1,6	0,2
LG10	1,6	0,2
Non-linear functions		
LINEAR	0,5	0,5
GAP	0,3	0,2
CHAR	0,9	0,5
Trigonometric functions		
SIN	1,4	0,2
COS	2,0	0,2
TAN	1,4	0,2
COT	2,9	0,2
ARSIN	2,4	0,2
ARCCOS	2,4	0,2
ARCTAN	1,8	0,2
ARCCOT	1,9	0,2
Logic functions		
AND	0,2	0,2
NOT	0,2	0,2
OR	0,2	0,2
EXOR	0,2	0,2
BOUNCE	0,3	0,2
FLIP	0,2	0,2
MONO	1,0	0,3
STEP	0,8	0,3
TIME1	1,2	0,3
Signal converters		
AOCTET	0,5	0,5
ABIN	1,5	0,3
TRUNC	0,3	0,2
PULS	0,9	0,2
COUN	0,4	0,3
MEAN	0,9	0,9
Time functions		
LEAD	0,7	0,3
INTE	0,6	0,3
LAG1	0,5	0,2
DELA1	0,9	1,9
DELA2	0,9	1,9
FILT	0,6	0,2

Function	Time %	Memory%
TIMER	0,5	0,2
TIME2	0,5	0,2
Selecting and storage		
EXTR	0,5	0,2
PEAK	0,3	0,2
TRST	0,3	0,2
SELC	0,3	0,3
SELD	0,2	0,2
SELP	0,3	0,3
SELV1	0,3	0,2
SOUT	0,3	0,2
REZEPT	0,7	0,5
2OF3	1,4	0,3
SELV2	0,4	0,2
Limit value signalling / limiting		
ALLP	0,8	0,3
ALLV	0,8	0,3
EQUAL	0,6	0,2
VELO	0,5	0,3
LIMIT	1,4	0,4
ALARM	0,4	0,3
Visualization		
TEXT	3,2	3,2
VWERT	0,8	1,7
VBAR	0,4	0,7
VPARA	2,5	1,1
VTREND	0,8	1,2
Communication		
L1READ	0,3	0,4
L1WRIT	0,3	0,4
DPREAD	0,5	0,4
DPWRIT	0,5	0,2
KS98-1 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_DMS	0,5	0,5
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
CPREAD	0,5	0,5
CPWRIT	0,5	0,5
CSDO	0,5	0,5

Function	Time %	Memory%
Programmer		
APROG	7,5	3,2
APROGD	0,3	0,5
APROGD2	0,3	0,5
DPROG	3,0	3,0
DPROGD	0,3	0,5
Controller		
CONTR	7,0	3,1
CONTR+	7,2	3,5
PIDMA	11,5	0,5
Inputs		
AINP1	0,5	0,5
AINP3	0,4	0,3
AINP4	0,4	0,3
AINP5	0,4	0,3
AINP6	0,5	0,5
DINPUT	0,3	0,3
Outputs		
OUT1	0,9	0,3
OUT2	0,9	0,3
OUT3	0,9	0,3
OUT4	0,9	0,2
OUT5	0,9	0,2
DIGOUT	0,2	0,3
Additional functions		
LED	0,2	0,2
CONST	0,2	0,4
INFO	0,2	0,9
STATUS	0,4	0,3
CALLPG	0,2	0,5
SAFE	0,3	0,5
VALARM	0,6	0,5
Modular Option C		
TC_Inp	0,5	0,5
F_Inp	0,9	0,2
R_Inp	0,9	0,7
U_Inp	0,9	0,4
I_Out	0,5	0,2
U_Out	0,5	0,2
DIDO	0,5	0,2

III-21.2 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

III-21.3 EEPROM data

Data are stored in non-volatile EEPROM. The manufacturers specify approx. 100 000 permissible write 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 cycles is almost precluded. With digital interface or automatic parameter changes, however, taking the maximum number of write cycles into account is indispensable, and measures against excessively frequent parameter writing must be take

III-22 Examples

During installation of the engineering tools, several examples were included. These are in path:
C:\Pmatools\Et98\prj\example

III-22.1 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, (ALQITSAV.EDG)

which are not lost also after prolonged power failure

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).

III-22.2 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 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)

III-23 Index

P.O.B. 310 229, D-34058 Kassel, Germany
Printed in Germany 9499 040 82711 (08/2009)
www.pma-online.de
Subject to alterations without notice

A4