UVR 1611

Version A4.05 / A5.02 EN

Freely programmable universal controller

en

Part 2:
Description of the function modules

TECHNISCHE ALTERNATIVE

Contents

Description of the function modules **4**
Solar control 5
Solar priority 7
Start function 9
Cooling function 10
Heating circuit controller 11
Mixer control 19
Comparison 20
Charging pump 21
Heating demand 23
DHW demand 26
Boiler cascade 28
DHW circulation 31
PID control (speed control) 33
Analogue function 37
Profile function 39
Logic function 41
Time switch 43
Timer function 45
Synchronisation 48
Heat meter 49
Meter 51
Maintenance FN 52
Function control 53
Factory setting **55**

Description of the function modules

The following modules are available at present:

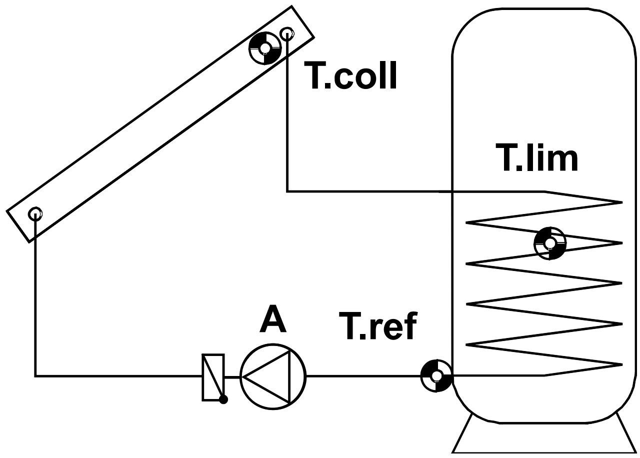
Solar control	Differential controller including various auxiliary functions
Solar priority	Priority assignment between multiple solar differential controllers
Start function	Start help for solar thermal systems
Cooling function	Cooling of an overheated solar cylinder during the night
Heating circuit controller	A mixer controller including heating circuit pump
Mixer regulation	Stabilizing the temperature variations by means of a mixer
Comparison	Comparison of two temperatures with one another (= thermostat
Charging pump	Differential and thermostat control of a charging pump
Heating demand	Burner demand by the buffer cylinder
DHW demand	DHW demand
Boiler cascade	Controls the burner demand of max. three boilers
DHW Circulation	Time and temperature control of a circulation pump
PID Control	Pump speed control
Analogue function	Searches for the lowest / highest temperature or the average
Profile function	Creates time-related (temperature) values (such as: for age-hardening of Floor pavement)
Logic function	AND-, OR-, holding function (Flip-Flop)
Time switch	Freely usable time switch clock
Timer function	Freely usable time interval function
Synchronization	Creates date-related switching signals
Heat meter	Energy evaluation
Meter	Freely usable interval or operating hours meter
Maintenance function	As a chimney sweep assistance and for exhaust measurement
Function control	Freely usable monitoring of sensors and differences
Menu Messages	System monitoring and output of error messages (The messages module is listed directly in the basic menu due to its properties.)

A maximum of 44 modules can be entered in the functions list.
When using memory-intensive functions (e.g. heating circuit controller), this number can be reduced.

Input variables which are absolutely necessary are highlighted in **bold** in the following function module description. The other input variables are optional.

Solar control

Basic diagram:



Input variables:	**Output variables:**
Enable solar circuit **Collector temperature** = T.coll **Reference temperature** = T.ref Limit temperature = T.lim	Status solar circuit Indication of the output A Status MAX limit reached = cylinder limit has been reached

Simple description of the function:
Release of the solar pump A, if the temperature in the collector T.coll is greater by a difference than the reference temperature T.ref, which is the (outlet) temperature of the cylinder. In addition, T.ref must not have reached its upper limit yet.

Special features:
- The system comes to a standstill when the collector exceeds the temperature of 140°C to prevent damage from steam. This means that the heat medium is no longer circulated, so that T.coll has a set upper limit (T.coll.MAX) including hysteresis.
- The differential temperature does not have a hysteresis that can be adjusted and is divided into a switch on and a switch of differential.
- If the cylinders have bare-tube heat exchangers, the reference temperature sensor should be screwed into the heat exchanger outlet using a t-shaped connector and an immersion sleeve (see the section on installing sensors in the instruction manual). If the surface of the collector is too large, the return flow temperature will increase too quickly, causing the limiter at T.ref to switch of the system to quickly. However, T.ref also cools down quickly in the standstill medium in the cold section of the cylinder. The pump will then start operating again, etc. To prevent this "cycling" and to prevent the cylinder from overheating if good layered cylinders are used, an **additional** optional upper limit has been defined in the module "solar control" for T.lim.
- The output variable “MAX limit reached:” shows that the upper limit has been reached (status: OFF/ON).
- If no additional limit sensor T.lim is used, it suffices to indicate *User* as the "source:" in the input variables.

Solar control

Entire menu view:

DES: SOLAR1	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
COLLECTOR TEMP.:	
T.coll.ACT: 74.3 °C	Current collector temperature
T.Ccoll.MAX: 130 °C	Pump is blocked when T.coll.MAX has been reached
Hysteresis: 10 K	Release at T.coll.MAX minus hysteresis
REFERENCE TEMP:	
T.ref.ACT: 65.7 °C	Current cylinder temperature (bottom/return)
T.ref.MAX: 70 °C	Cylinder limit
Hysteresis: 3.0 K	Release at T.ref.MAX minus hysteresis
DIFFEREN. COLL-REF:	
DIFF.ON: 7.0 K	Switch-on differential T.coll – T.ref
DIFF.OFF: 4.0 K	Switch-off differential T.coll – T.ref
LIMIT TEMP.:	
T.lim.ACT: 54.0 °C	Current temperature of the limit sensor
T.lim.MAX: 70 °C	Blocked by sensor
Hysteresis: 3.0 K	Release at T.lim.MAX minus hysteresis

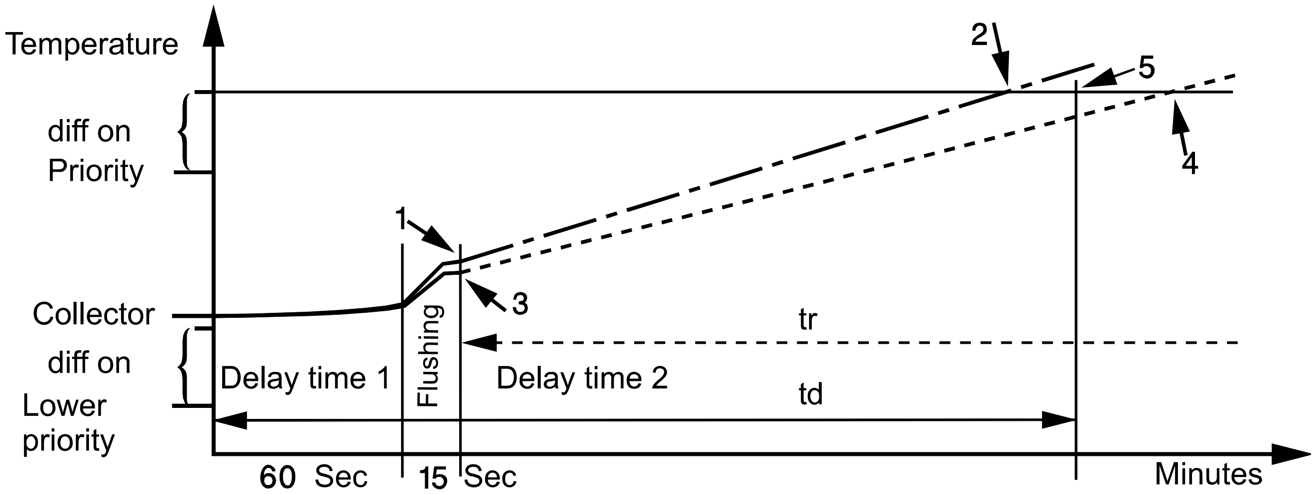
Because the module is used for all kinds of consumers, the labels "reference temperature" and "limit temperature" have been specified as generally applicable.
When using the third sensor as a limiter, the upper limit of reference sensor "T.ref.MAX" should be set high enough so that it does not affect operation.
The output variable “MAX limit reached:” has the status "ON" if the reference sensor's **or** the limit sensor's upper limit has been reached.

Solar priority

In solar heating systems used to charge more than one consumer (such as a cylinder, a buffer, and a basin), priorities must be set for the various circuits. There are basically two sets of rules used for a system of priority and lower priority assignment.

- **Absolute priority**: Only when a cylinder that has priority has reached its upper limit for temperature does the system switch to the next lower priority.
- **Relative priority**: The charge begins with the colder cylinder (because the collector will reach the differential here first), even if this cylinder does not have priority.

While the consumer that does not have priority is being charged, the device monitors the collector temperature. If the collector temperature once again reaches the switch-on differential for the consumer currently being charged while the pump is running, the priority timer is activated. If a radiation sensor is used, it must exceed at threshold value in lieu of the switch-on differential.



The priority timer switches the pump off for delay time 1 (60 sec). After flushing (1, 3), the computer then calculates the increasing collector temperature. It recognizes whether the set delay time 2 td will suffice to heat up the collector to the priority temperature (5). If not, the system waits for priority temperature to be reached before switching (case 2). If the computer determines that the increase will not suffice within the delay time 2 td (case 4), it interrupts the procedure and disables the priority timer until runtime tr has expired.
If runtime = 0, the lower priority is allowed once the maximum threshold of the current priority has been reached. In other words, the system is switched to absolute priority assignment mode.

Solar priority

Input variables:	**Output variables:**
Enable solar priority Solar radiation = Radiation sensor **Involved functions** = Entry of all solar functions in the function list	Status flushing process Indication of the output for the flushing

Special features:

- In this function blocks, the "involved functions" are not individual values, but rather entire function module input variables.
- The program automatically looks for all of the values needed in the function modules involved and also automatically blocks the modules involved that are lower in the hierarchy.

Entire menu view:

(Assumption: Six solar functions are entered in the function list)

DES: SOL. PRI.	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
SOLAR1 1	SOLAR 1 is the highest priority
SOLAR2 2	SOLAR 2 is the second priority
SOLAR3 3	SOLAR 3 is the last priority
SOLAR4 1	SOLAR 4 is the highest priority
SOLAR5 2	SOLAR 5 is the second priority
SOLAR6 3	SOLAR 6 is the last priority
SEC. TIMER:	
from prim. stage 2	SOLAR 1 and 4 are loaded as "absolute" without time element
Solar rad: 488 W/m²	Current solar radiation (does not apply without radiation sensor)
Thres val: 200 W/m²	Activation threshold for timer (does not apply without radiation sensor)
Runtime: 20 Min	Runtime for the consumer of lower priority until timer starts
delay time: 5 Min	The collector must reach the temperature of the priority cylinder within five minutes; otherwise, the cylinder of lower priority will be charged

As this example makes clear, it is also possible to assign the same priority. Generally, this approach is only useful if the system has multiple collector fields. The priorities in the example correspond to the system with two collector fields charging three consumers (such as Solar 1 = collector 1 for cylinder 1 and Solar 2 = collector 1 for cylinder 2, etc.).
SOLAR 1 and SOLAR 4 are admissible first as the lower priority time element is only active starting at priority level 2 until the consumer has reached its maximum temperature (absolute). Only then does the priority treatment shift to the other solar functions via the priority timer (relative).

Start function

Simple description of the function:

Sometimes, the collector sensor is not immersed in the heated heat medium quickly enough. In other words, the system goes into operation too late. There may not always be enough gravity force if the collector fields are completely horizontal, if the connections between the absorber strips meander, and especially if **vacuum tubes** are used with forced circulation.
This module puts the solar pump into operation in preset intervals, thus transporting the content of the collector to the sensor. To reduce energy losses, this interval operation is only launched within a certain time window and only if the solar radiation reaches a certain level (determined using **GBS** radiation sensor - special accessory) **or** with constant monitoring of the collector temperature. Without a radiation sensor, the computer first attempts to determine the current weather conditions by constantly measuring the collector temperature. By this, it finds the right time for a brief rinsing interval so that the temperature for normal operation can be maintained. Each collector field must have its own start function.

Input variables:	**Output variables:**
Enable start function Solar radiation = Radiation sensor Reference temperature = Input of the collector sensor **Involved functions** = Entry of all solar functions for the collector field in the function list	Status flushing process Indication of the output for the flushing

Entire menu view:

`DES: SOL START`	
`FUNCTION STATUS:`	
`INPUT VARIABLE:`	
`OUTPUT VARIABLE:`	
`Activation time:`	
`07:00   -  20:00 h`	Time window for permission of start function
`Runtime:      15 Sec`	Rinsing time
`Interval:      20 Min`	Maximum delay between rinses
`Activ.grad.:       20`	or radiation threshold - see description below
`Start attempts:   13`	Sum of attempted starts today
`unsuccessful:     11`	Number of unsuccessful starts
`since last run:    6`	Number of attempts since the system last operated correctly

If a radiation sensor is used, instead of "Activ.grad." the computer displays the desired radiation threshold above which the start function is to be active. In many cases, you can do without this sensor altogether. Then, a mean value is calculated from the collector temperature, with special attention paid to the lowest temperatures reached. The start function is enabled when the collector temperature is warmer than the mean value by the difference of the activation gradient. A lower activation gradient therefore leads to an earlier start attempt, a higher gradient to later attempts.
If you need more than ten tries to start the system, the activation gradient must be increased and must be reduced, if the number of attempts is below four.
If the activation gradient is set to zero, then only the activation or interval time applies without consideration of the temperature curve at the collector sensor.

Cooling function

Simple description of the function:

Solar heating systems with partial solar heating produce more warm water in the summer than can be used. This function can be used at night to remove some of the excess energy from the lower section of the cylinder back into the collector when a critical temperature has been reached in the buffer cylinder by regulating the speed of circulation. During the day, the system will then not have to be brought to a standstill as often due to excess temperature.

Input variable:	Output variable:
Enable cooling function **Reference temperature** = Measuring point that sets off the function Maximum temperature reference = Temperature that sets off the function	Correcting variable = Speed stage RPM output Indication of the RPM control-output Status switching output, Indication of the switching output

Special features:

- Generally, the maximum value for the thermostat threshold can be set. This value is defined as an input variable to make it easy to link it. You only need to enter "source" *User* as the setting value. The user can see it as a common function parameter when it appears in the menu of the function.
- In addition to the output indicated with a set speed, the function module also provides a switchable output. This output can be used to block other functions during the cooling phase.
- The maximum set value does not have a hysteresis that can be set, but rather a switch-on and a switch-off differential.

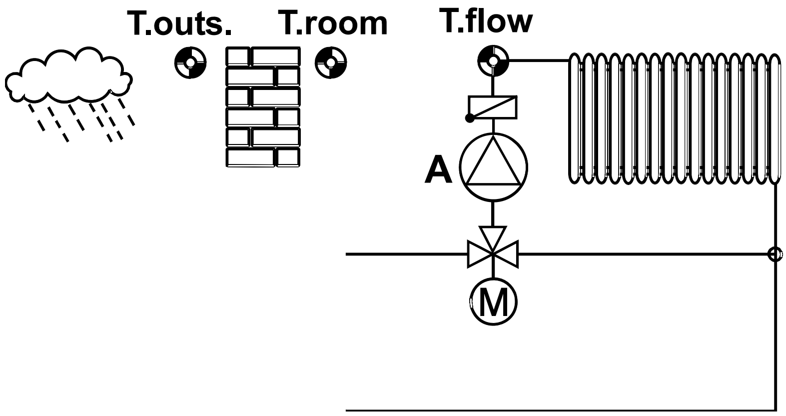
Entire menu view:

Menu	
DES: COOL FUNC	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
TIME WINDOW:	
00:00 - 06:00 h	Time window for active cooling
REFERENCE TEMP.:	
T.ref.ACT: 65.7 °C	Current cylinder temperature (bottom/return)
T.ref.MAX: 90 °C	Cylinder limit
DIFF.ON: 5.0 K	Cooling active between 12 midnight and 6 a.m. above 95°C
DIFF.OFF: 0.0 K	The cooling function is switched off once the temperature drops below 90°C
CORRECT.VAR.: 15	The pump runs at speed stage 15

Tests have shown that cooling is also sufficient at the lowest speed stages. We therefore recommend that you use a speed stage just above circulation standstill. For instance, at the stage 5 the pump only consumes 10% as much energy.

Heating circuit controller

Basic diagram:



Input variable:	**Output variable:**
Enable heating circuit controller Enable pump Enable mixer Room temperature = T.room Flow temperature = T.flow Outdoor temperature = T.outs. External Switch = Switching to frost protection mode (status: ON) / Operation according to the setting of the unit (status: OFF)	Set flow temp. = Temperature of the flow calculated by the control unit T.flow SET Effective set room temperature = Valid indoor temperature according to time program T.room EFF Status of the heating circuit pump, indication of the output Status of the mixer, indication of the output Status of the maintenance mode Status of the frost protection mode

Simple description of the function:

Mixer controller based on outdoor and indoor temperature with consideration of the heating and lowering temperature specified in the switching times. The heating pump can be enabled in various parameters.

Special features:

- The input variable "**EXTERNAL SWITCH**" can be used to switch a remote switch between frost protection mode and normal operation according to the device settings. Furthermore, the external setting of an operating mode is possible via a dimensionless number (64 to 67).
- In addition to the pump and the mixer, this function also provides a calculated flow temperature (T.flow SET) and the status of maintenance and frost protection modes, e.g. for messages.
- Another output variable is the effective room temperature (T.room EFF), which is influenced by the timer and other functions. A heating control unit **without a mixer** can thus be set up with a downstream speed control module.
- Under "**OPERATION**," special functions such as PARTY and HOLIDAY, etc. are callable.
- A **derivating time** which can be selected and is based on the outdoor temperature also affects the switching between lowering and heating modes.
- Four criteria can be selected to switch off the pump.
- If, when the function is first called, or using "CHANGE FUNCTION" **yes** is specified to the time program "with set value?", then each time window receives its own adjustable room temperature, which replaces the value "T.room STD".
- If a room sensor is indicated in the input variables but the sensor is short-circuited, the heating circuit controller will operate as though no room sensor were indicated in the parameters.
- The mixer runtime is reloaded when the mixer output is in manual mode, is triggered by a message (dominant ON or OFF), or a trigger switches from OPEN to CLOSED. If mixer enable is OFF the mixer remains stationary in the last position.

Heating circuit controller

Basic menu view:

DES: HEAT.CIR.1 FUNCTION STATUS: INPUT VARIABLE: OUTPUT VARIABLE:	
OPERAT.: RS STANDARD	The heater is controlled by a room sensor RAS and is currently running in heating mode (*STANDARD*)
ROOM TEMPERATURE:	
T.room ACT: 20.7 °C	Current room temperature
T.roomSETBACK:16 °C	Desired room temperature doing lowering time
T.room STD: 20 °C	Desired room temperature doing heating time
TIME PRG:	Submenu for heating times (see **Time programs**)
Deriv. time: 30 Min	The heating time begins 30 min earlier if the outdoor temperature is below -10°C
T.room EFF: 20°C	Current desired room temp. = 20°C (current heating operation)
FLOW TEMPERATURE:	
T.flow ACT 58.4 °C	Current flow temperature
T.flow SET: 58.2 °C	Calculated flow temperature
HEAT CURVE:	Submenu for the calculation of the flow temperature
OUTDOOR TEMPERATURE:	
T.outs.ACT: 3.6 °C	Current outdoor temperature
AVERAGE:	Settings for calculation of outdoor temperature for the calculation of flow temperature and to switch off the pump
SHUTDOWN CONDITION:	Submenu to switch off the pump and conditions for mixer
FROST PROT:	Submenu where the outdoor temperature is indicated to keep the room at a certain minimum temperature

OPERATION

***TIME*/AUTO** may also be found under "**OPERAT**:" if "unused" has been entered as an input variable for the room sensor. Furthermore, you can also switch to the following heating functions here regardless of whether a room sensor is being used:

- ***STANDBY*** Switches the control to standby (frost protection remains activated)
- ***SETBACK*** The controller is switched to manual mode - setback
- ***STANDARD*** The controller is switched to manual mode - heating (standard)
- ***HOLIDAY*** The controller takes the heating times for Saturday starting today and the times of Sunday for the last day indicated
- ***BANK HOL.*** Up to date xx, 00:00 hours, the controller will only work in lowering mode
- ***PARTY*** The heating mode remains in operation until the indicated time xx

For the modes of operation ***HOLIDAY***, ***BANK HOL.*** and ***PARTY***, the controller switches back to automatic mode after the time indicated has expired.
The room sensor is not evaluated in simulation mode; therefore there is **no** "**RS**" operating display.

Further possible displays under "OPERATION":

FROST PROT The frost protection function is activated. The activation conditions are described in the section "Frost protection".

EXT/STANDBY The input variable „External Switch“ is a digital "**ON**" signal.

MAINTENANCE The maintenance function is active (see function "Maintenance"). The flow temperature is controlled to match the setting T.flow MAX set in the menu HEAT CURVE. The function module remains active for three minutes after maintenance mode has been switched off.

FAULT If the line to the external sensor (measuring value > 100°C) is interrupted, the heating circuit is switched off. In the worst-case scenario, the system may be damaged by frost. To prevent such damage, the heating circuit is operated according to a set outdoor temperature of 0°C if the outdoor temperatures are clearly too high, and ***FAULT*** is displayed under "OPERAT:".

Status of the heating circuit pump and the mixer

relative to the operation mode and releases:

Operation mode	Enable heating circuit	Enable pump	Enable mixer	Status: pump	Status: mixer
X	OFF	x	x	OFF	OFF
Maintenance	x	x	x	ON	AUTO[1]
Standby, external standby	x	x	x	OFF	OFF
Frost protection, fault	ON	x	ON	ON	AUTO
Frost protection, fault	ON	x	OFF	ON	OFF
Time / Auto, standard, setback party, bank holiday, holiday	ON	OFF	OFF	OFF	OFF
Time / Auto, standard, setback party, bank holiday, holiday	ON	ON	OFF	AUTO	OFF
Time / Auto, standard, setback party, bank holiday, holiday	ON	OFF	ON	OFF	OFF[2]
Time / Auto, standard, setback party, bank holiday, holiday	ON	ON	ON	AUTO	AUTO
RS	ON	as with Standby, Time/Auto, Standard, Setback,			

x… Status and operation mode do not matter

[1]… In this case, AUTO means that the settings for T.flow MAX are used in the menu HEAT CURVE.

[2]… OFF does not apply if in "SHUTDOWN CONDITION" the setting "control" is set under "if heating circ. OFF => MIXER:"

EXTERNAL SWITCH

The “EXTERNAL SWITCH” input variable also accepts analogue values on external operating mode switching:

Value (dimensionless):	Operating mode:
64	Standby/frost protection
65	Time/Auto
66	Standard
67	Setback
127	Switch back to internal operation

These analogue values can originate from another function or also come from the GSM module from the Bootloader as a network input. Values **64 - 67** are dominant, i.e. it is not possible to set any other controller operating mode, as long as the value "external switch" exists.

NB: If nevertheless at attempt is made to set another operating mode, then the controller display jumps back from the operating mode specified from the "External switch" and remains in the original operating mode. However, the controller "notices" this change and adopts this operating mode after resetting using the value 127 at the "External switch". If during this time an operating mode **other** than "**RS**" is selected, then this operating mode cannot be changed on the **RS**, rather only on the controller, on the CAN monitor or via the browser. As soon as the value at the "External switch" is 127, manual alteration of the operating mode is possible at any time.

Important note: The external switch must never be linked with a temperature sensor because the controller can be damaged otherwise.

TIME PROGRAM

Parameterizing of the time program is described in the chapter "Menu functions".

The room temperature T.room STD or the set value applies in the time window. T.room SETBACK applies outside the time window. The switchover causes a corresponding parallel shift of the heating curve and therefore a change to the set flow temperature, T.flow SET.

"CHANGE FUNCTION" is used to change the number of required time programs and the number of windows per program and to specify the use of a special set value per time window:

`Scope of TIME PROG:`	
`Number of Prog.: 3`	Maximum 5 time programs can be set
`No. of windows:  3`	Maximum 3 time windows per time program can be set
`with set value ? no`	The input "*no*" means that the same set value T.room STD is used for all time windows. The input "*yes*" makes it possible to allocate a special set value instead of T.room STD.
`CHANGE?         no`	Activation of the change with “CHANGE? ***yes***“

Derivation TIME

Depending on the outdoor temperature, fixed heating times can cause the system to switch on or off too quickly. The rate time can shift the switch point relative to the outdoor temperature. This entry refers to an outdoor temperature of -10°C and is 0 at plus 20°C. For example, if the rate time is 30 minutes and the outdoor temperature is 0°C, the system will switch on 20 minutes earlier under normal operation.

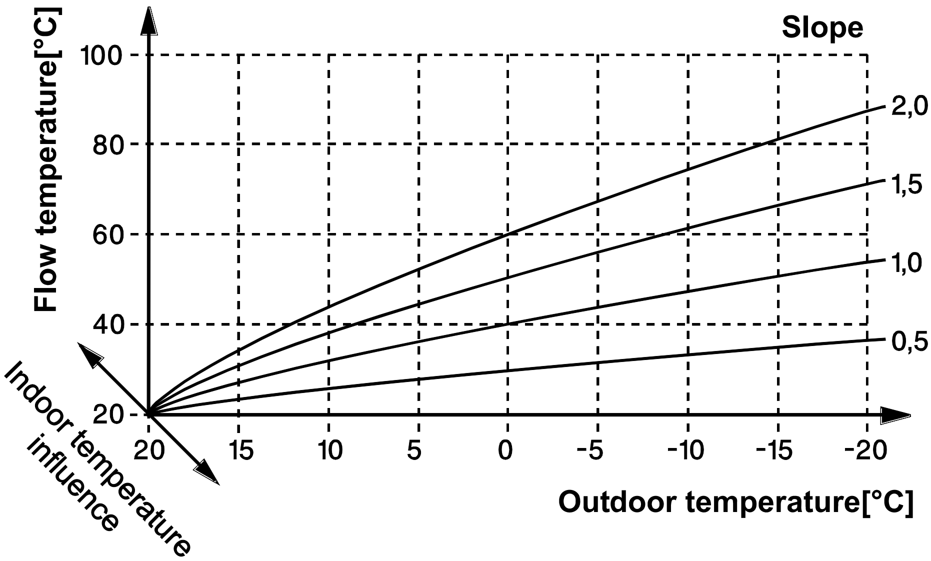
HEAT CURVE

The flow temperature is usually calculated from the outdoor temperature and the heating curve. The heating curve is calculated based on a room set temperature of +20°C and is subject to an appropriate parallel offset for other room set temperatures. An exception is the fixed value control. The flow is set in setback mode to the listed temperature of +10°C and in standard mode to that of -20°C.
The module allows you to set parameters for the heating curve in two ways:

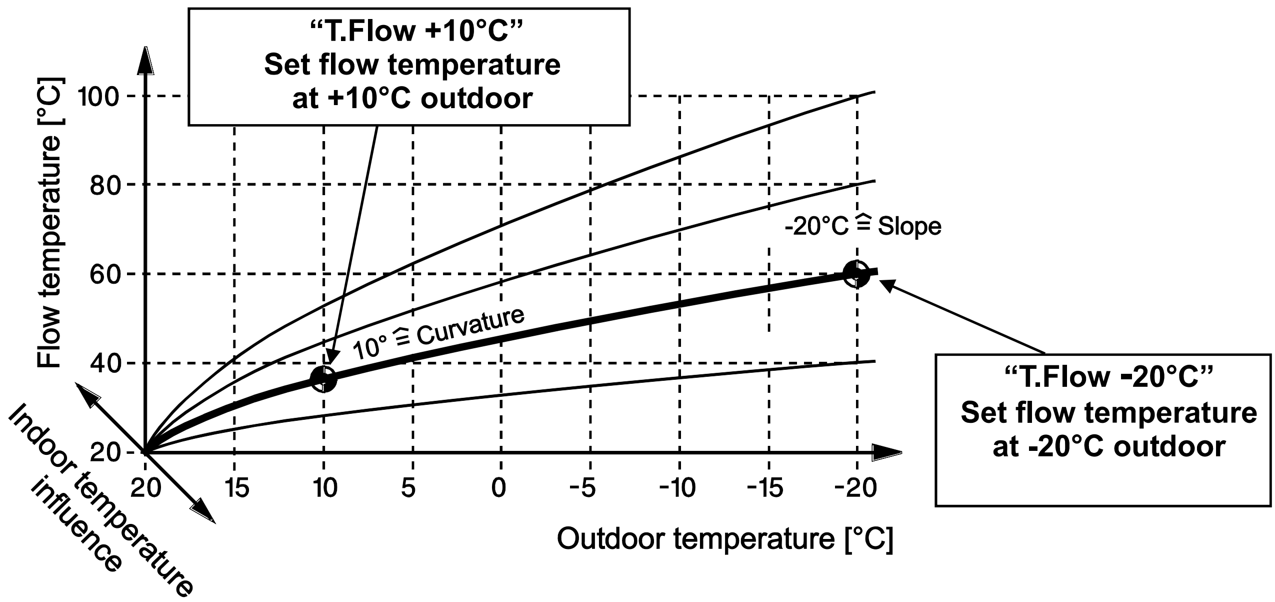
- With reference to the slope as is common in many heating control units.
- As a product of the relation between the outdoor temperature (at +10°C and -20°C) and the flow temperature. Here, another reference point is set at plus 20°C outdoor temperature = plus 20° flow temperature.

In both of these methods, the influence of the outdoor temperature on the flow temperature is **not linear**. Via "***Slope***" the curvature is set according to the standards. Via "***Temperature***" a curvature of the heating characteristic is created with the desired flow temperature at 10°C in order to take into account the different heat emissions of various heating systems.

Heat Curve "Slope"



Heat Curve "Temperature"



The following entries are found in this submenu “Heat curve”:

HTG CIRC.1 MODE: CONTROL : Outs.temp. ***or*** Fixed val.	Control based on outdoor temperature and heating curve The flow is set in setback mode to the indicated temperature at +10°C and in standard mode to the one at -20°C.
HEAT CURVE: Temp. ***or*** Slope	Heating curve via temperature points +10°C and -20°C Heating curve via input of the slope (0.05-2.50)
Room inf.: 0 %	The room temperature is taken into consideration for the calculation of the flow at xx%, setting range 0 – 90% The room influence is also active in fixed value mode.
Start excess: 0 %	The previous switch-off time of the heating circuit pump leads to a (decreasing over time) excessive increase in flow temperature. (maximum up to T.flow MAX). Setting range 0 – 20% *) for detailed explanation see below
T.flow+10°C: 35 °C T.flow-20°C: 60 °C ***or:*** Slope 0.60	Desired flow temp. at +10°C outdoor temp. (heating curve) Desired flow temp. at -20°C outdoor temp. (heating curve) Specification of the slope (for selection Heat Curve: Slope)
T.flow MAX: 65 °C T.flow MIN: 20 °C	The flow must not exceed this value The flow must not drop below this value

*) Start excess (“Increasing on Start”)

The precise formula for the start excess is:

T.flow SET Start excess = T.flow SET + T.flow SET* (Start excess / 100) * (Meter/ 30)

The meter is increased by 1 every 20 minutes with a **switched-off** heating circuit pump, with a **switched-on** heating circuit pump, it is lowered by 1 every minute down to 0.

The maximum meter value is 255. It is therefore reached after 85 hours switch-off time (= 255/3 hours or approx. 3.5 days). The maximum run-down time is 4.25 hours (= 255 minutes). The set excessive increase in % is effective after a switch-off time of 10 hours (= 30 x 20 minutes).

Example: T.flow SET=40°C, Start excess = 10%, Switch-off time 8 hours

The excess temperature starts at +3.2 K (above the set value) and falls uniformly to zero within 24 minutes.

Protection of heat-sensitive systems parts:

Heat-sensitive systems parts (e.g. plastic ducts) must be equipped with additional protecting devices (e.g. thermal temperature limit for floor heating) which prevent overheating in the event of failure of the controller or of another system component.

AVERAGE of outdoor temperature

Sometimes, fluctuations in outdoor temperature may even not be desirable in calculation of flow temperature or when determining the disconnection of the heating pump. Therefore, a separate calculation of the mean is available for the outdoor temperature to calculate the heating curve and the disconnection of the pump. The following entries are found in this submenu:

HTG CIRC.1	
for flow control: Aver time: 10 min	The average outdoor temperature is calculated for 10 minutes for the flow
T.outs.av.c: 13.6 °C	The current outdoor temperature average is 13.6°C
for shutdown: Aver time: 30 min	The average outdoor temperature is calculated for 30 minutes for switching off
T.ou.av.off: 13.8 °C	The current outdoor temperature average is 13.8°C

SHUTDOWN CONDITIONS and mixer behaviour

The controller allows the following switch-off conditions for the heating circuit pump:

if T.room ACT > SET ? no Hysteresis: 1.0 K	when the desired room temperature has been reached
If T.flow SET < MIN ? yes Hysteresis: 2.0 K	when the flow temperature falls below the lower limit T.FlowMIN
If T.outs. Av.off> MAX ? no T.outs.MAX: 20 °C Hysteresis: 2.0 K	when the average outdoor temperature T.Out.MAX exceeds a set value in the heating or lowering mode
If setback mode and T.outside ACT > MIN. ? no T.outs.MIN: 5 °C Hysteresis: 2.0 K	when the outdoor temperature T.Out exceeds a set value in the lowering mode
If T.flow ACT > MAX ? no	when the flow temperature is larger than T.FlowMAX (settings in the heat curve) plus a fixed hysteresis of 3K, reactivation if T.FlowACT. < T.FlowMAX
If heating circ. OFF MIXER: close	**Mixer behaviour:** In addition, in this menu you can determine how the mixer is to respond once the pump has been switched off (*close, open, unchanged, (continue to) control*). If mixer enable is "OFF" the mixer remains stationary in the last position it took up (Status Mixer: OFF).

The hystereses of the switch-off conditions generally have an **increasing** effect.

As both the outdoor temperature and the room temperature are taken into consideration when the set flow temperature is calculated (provided a sensor is used), the best method is to switch off under the limit T.flow MIN.

FROST PROTECTION

This function is only active in the standby mode or when the input variable "EXTERNAL SWITCH" is active, and even then only if the module has been partially blocked by the input variable ENABLE PUMP or a switch-off condition would block the heating circuit pump**. However, if the function is blocked via Enable heating circuit, there is no frost protection mode.**

If frost protection is activated, the set flow temperature is maintained at least at T.flow MIN (setting in the sub-menu heating curve), until the temperature, which triggered the frost protection function, increases by 2 K above the frost protection limit. This submenu has the following entries:

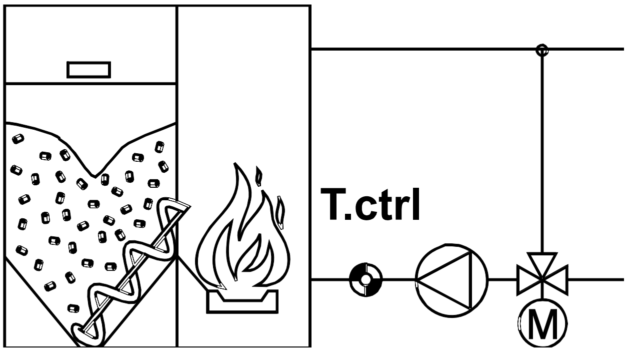
```
Activation if
T.outs.av.c  <   5°C
T.room FROST:    5°C
```

Frost protection function is active below +5°C (outdoor) and keeps the room at a temperature of 5°C

Heating circuit condition	**Frost protection function**
Operating mode STANDBY Set to RAS/RASPT	<u>without activated frost protection function:</u> T.flow SET set to +5°C, mode display: STANDBY <u>Activation of the frost protection function:</u> If T.outs.ACT < T.outs.av.c, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)
Operating mode STANDBY Controller setting	<u>without activated frost protection function:</u> T.flow SET set to +5°C, mode display: STANDBY <u>Activation of the frost protection function:</u> If T.outs.ACT < T.outs.av.c **or** (if RAS available) T.room ACT < T.room FROST, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)
Switching via digital "ON" at the "external switch" to EXT/STANDBY	<u>without activated frost protection function:</u> T.flow SET set to +5°C, mode display: EXT/STANDBY <u>Activation of the frost protection function:</u> If T.outs.ACT < T.outs.av.c **or** (if RAS available) T.room ACT < T.room FROST, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)
Switching via analogue 64 at "external switch" to STANDBY	<u>without activated frost protection function:</u> T.flow SET set to +5°C, mode display: STANDBY <u>Activation of the frost protection function:</u> If T.outs.ACT < T.outs.av.c **or** (if RAS available) T.room ACT < T.room FROST, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)
Enable pump off	<u>without activated frost protection function:</u> T.flow SET according to the setting heat curve, Mode display: selected operating mode <u>Activation of the frost protection function:</u> If RAS available: If T.room ACT < T.room FROST the pump is switched on, **independent of the outside temperature**, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)
Switching off of the pump via a switch-off condition	<u>without activated frost protection function:</u> T.flow SET set to +5°C, mode display: STANDBY <u>Activation of the frost protection function:</u> If RAS available: If T.room ACT < T.Room FROST the pump is switched on, **independent of the outside temperature**, then T.flow SET ≥ T.flow MIN (mode display: FROST PROT.)

Mixer control

Basic diagram:



Input variables:	Output variables:
Enable Mixer **Control Temperature** = T.ctrl - indication of a sensor Set Value = Control to this value (+diff)	Control Temperature = T.ctrlEFF, set temperature calculated by the controller from the control temperature and the difference Status Mixer = M, indication of the outputs

Simple description of the function:

This function allows a mixer to be constantly controlled for a set value.

Special features:

- Generally, the set value will be an adjustable one. This value has been defined as an input variable to make it easy to link it. It will then appear in the menu for the function that the user can see as a common function parameter if *User* is indicated as the "source."
- Another differential can also be used to set the set value.
- In addition to the mixer output, the function also sets the overall set value as the effective control temperature (T.ctrlEFF) for the output variable.
- As the module can only be switched via its enable, the mixer setting is preset to "enable OFF."
- In addition to *standard*, the mixer mode can also be set to *inverse* (such as a cooling function for wall heaters, etc.). Under *inverse*, the mixer opens as the temperature increases.
- The mixer running time (20 minutes) is reloaded, if the mixer output is in manual mode, triggered by a message (dominant ON or OFF), or the triggering direction changes from OPEN to CLOSED or vice versa or the enable is switched from OFF to ON.

Entire menu view:

DES: MIX CTRL INPUT VARIABLE: OUTPUT VARIABLE:	
MODE: std	The mixer closes as the temperature increases
CONTROL TEMP.:	
T.ctrl ACT: 30.4 °C	Current control temperature
T.ctrl SET: 30 °C	Preset control temperature
Differen.: 0.0 K	Additional control differential for the set value
if ENABLE = Off MIXER: unchanged	Mixer behaviour with enable = off: *close, open, unchanged*

Comparison

(Thermostat/differential function)

Simple description of the function:

Two values, Va and Vb + differential, are compared to produce the two output variables Va > Vb and Va < Vb.

Input variables:	Output variables:
Enable comparison	Status Va > Vb + diff = Value a is greater than value b
Comparative Value a = First comparison value	Status Va < Vb + diff = Value a is lower as value b, indication of the output
Comparative Value b = Second comparison value	

Special features:

- Only one sensor input / one output variable for another function is admissible for the value a. Value b can also be a flexible (temperature) value. If so, *User* has to be indicated as the "*source*." The user can see the value b as a common function parameter when it appears in the menu of the function.
- Generally this function applies to a thermostat. Use the "**function variable**" (FUNC.VAR.) to make comparisons of any figures. Possible selections: temperature, dimensionless, flow rate, output, heat amKWh, heat amMWh, number of impulses, time, solar radiation, relative humidity, wind speed and pressure.
- The comparison difference consists in the differential between the switch-on and switch-off.
- Both Va > Vb and Va < Vb can be used as output variables. When comparing a temperature sensor with a threshold value (value b is entered as "user" under the input variables), this basically functions as a mechanical thermostat with mutual contact (Va > Vb = make contact and Va < Vb = break contact).
- If sensors have been assigned to both values, the result is a simple differential function.
- If enable is set to "OFF", then **both** output variables are set to "OFF".

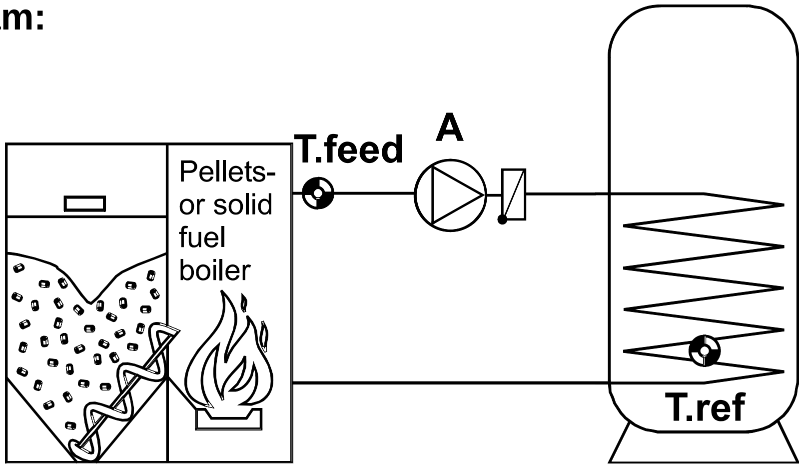
Entire menu view:

`DES:   COMP.1` `FUNCTION STATUS:` `INPUT VARIABLE:` `OUTPUT VARIABLE:`	
`FUNC.VAR: Temp.`	Comparison of two temperatures
`VALUEa:      39.1 °C` `VALUEb:      44.3 °C`	
`DIFF.ON:      5.0  K` `DIFF.OFF:     2.0  K`	Switch on if value a exceeds 49.3°C (44.3 + 5.0) Switch on if value a falls below 46.3°C (44.3 + 2.0)

WARNING: The output status of the second output variable is the inverse of the first output variable Va > Vb + diff. The designation Va < Vb + diff for the second output variable is thus inaccurate. This representation was selected because the display cannot indicate the inverse symbol.
If to sensors are compared, the connection of the warmer sensor (generator) to Va is recommended. If the link between values a and b are adversely linked in the input variables, switching will then be based on a negative differential.

Charging pump

Basic diagram:



Input variables:	**Output variables:**
Enable charging pump **Feed temperature** = T.feed **Reference temperature** = T.ref Minimum temp. Feed = Min. threshold at T.feed Maximum temp. ref. = Max. threshold at T.ref	Status of the charging pump = A Indication of the output A

Simple description of the function:

Release of charging pump A if the temperature on the boiler (feed temperature T.feed) is higher than the minimum temperature and one difference higher than the reference temperature T.ref.. In addition, T.ref must not have reached its upper limit yet.

Special features:

- In most applications, the minimum threshold for T.feed and the maximum threshold for T.ref can be set. These two thresholds are defined as input variables to make it easy to link them.
- The example used is a link with the burner demand for DHW preparation. The function *DHW demand* provides the set temperature for the cylinder as an output variable. The set temperature can therefore be used simultaneously as the maximum temperature for the charging pump function.
- If the two input variables can be set, all you need to enter is *User* as the "source." The user can see it as a common function parameter when it appears in the menu of the function.
- Neither thermostat threshold has hysteresis, but rather a switch-on/off differential to the adjustable threshold value.
- **Example**: Minimum threshold = 60°C
  DIFF.ON = 5.0 K
  DIFF.OFF = 1.0 K

  In other words, if the temperature T.feed exceeds 65°C (= 60°C + 5°K), the output is activated, whereas it is switched off if the temperature drops below 61°C (= 60°C + 1°k).

Charging pump

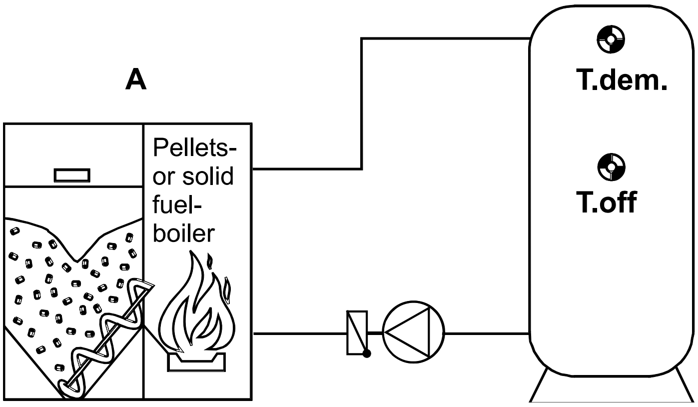
Entire menu view:

DES: LD PUMP 1	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
FEED TEMPERATURE:	
T.feed ACT: 74.3 °C	Current temperature of the "energy generator"
T.feed MIN: 60 °C	Basic switch-on threshold at sensor T.feed
DIFF.ON: 5.0 K	Switch-on differential to T.feed MIN (here, 65°C)
DIFF.OFF: 0.0 K	Switch-off differential to T.feed MIN (here, 60°C)
REFERENCE TEMP.:	
T.ref.ACT: 65.7 °C	Current cylinder temperature
T.ref.MAX: 90 °C	Cylinder limit
DIFF.ON: 1.0 K	Switch-on differential to T.ref.MAX (here, 91°C)
DIFF.OFF: 5.0 K	Switch-off differential to T.ref.MAX (here, 95°C)
DIFFERENCE FEED-REF:	
DIFF.ON: 6.0 K	Switch-on differential FEED - REF
DIFF.OFF: 3.0 K	Switch-off differential FEED - REF

While the minimal feeder temperature has to have constantly DIFF.ON > DIFF.OFF, DIFF.ON must always be < DIFF.OFF for the maximal reference temperature.

Heating demand

Basic diagram:



Input variables:	**Output variables:**
Enable heating demand **Demand temperature** = T.dem. Shutdown Temperature = T.off Set value demand = Min. threshold at T.dem Set value shutdown = Max. threshold at T.off	Status demand Indication of the output A (= Enable burner)

Simple description of the function:

Release of burner A if temperature in the buffer cylinder at the top (demand temperature T.dem.) falls below the "set value demand" (corresponds to a minimum threshold) and switch-off if the temperature in the bottom of the cylinder (shutdown temperature T.off) exceeds the "set value shutdown" (corresponds to a maximum threshold).

Special features:

- Generally, the values for demands and switch-off as thermostat threshold can be set. Here, both thresholds are defined as input variables. If the values can be set, you only have to enter *User* as the "source" in order to have it here for the user as a function parameter in the menu of the function.
- The system is switched on / off via separate threshold values and sensors, so that the two thresholds do not have any hysteresis. Instead, both thresholds have a differential to the value that can be cumulated.
  Switch-on threshold = Set value demand +DIFF.ON at sensor T.dem.
  Switch-off threshold = Set value shutdown +DIFF.OFF at sensor T.off
- The method of burner demand via a sensor and switch-off via another one is called "holding circuit" If a switch function has separate switch on/off thresholds **for just one sensor**, the input variable "SHUTDOWN TEMPERATURE" has to be set to *User / unused*. If the boiler sensor is entered instead of the cylinder sensor, the result is a sliding boiler operation. The "DEMAND TEMPERATURE" then has a differential for switch on/**off** in addition to the threshold value.
  Switch-on threshold = Set value demand + DIFF.ON
  Switch-off threshold = set value demand + DIFF.OFF
- A minimum temperature can be set via "**Low end temperature**" T.dem.MIN.
  Switch-on threshold = T.dem.MIN +DIFF.ON at sensor T.dem.
  Switch-off threshold = T.dem.MIN +DIFF.OFF at sensor T.off
  The base temperature is only effective, if the set value demand > 5°C. A value > 30°C is only useful if the function is used for sliding boiler operation. In this case, the switch off and on thresholds relate to the sensor T.dem..

Heating demand

Eco mode:

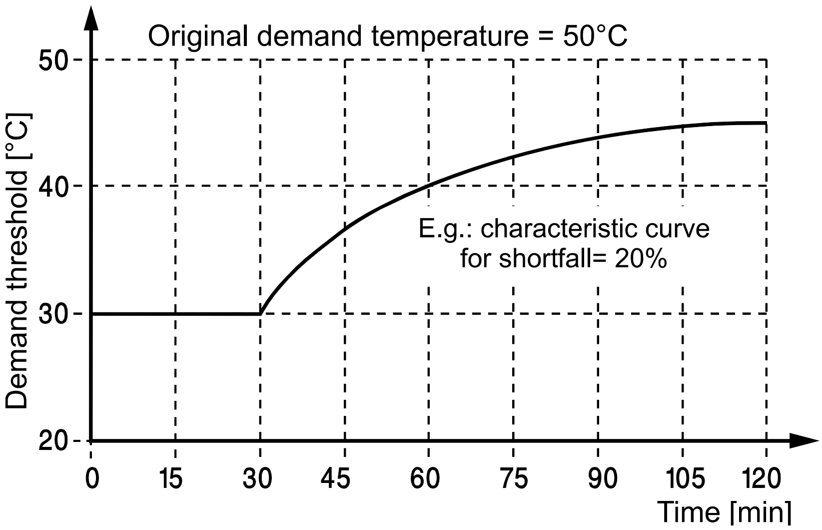
Eco operation is related by "shortfall" to a particular time span. The shortfall stage always applies for 60 minutes. For a demand temperature of 50°C shortfall of 20% has the following effect: demand after 30 minutes below 30°C or below 40°C (= 20%) after an hour or below 45°C after 2 hours. The threshold value remains the same below 30 minutes.

Formula: dT * dt = Shortfall * set value demand temperature = constant

Example:

Demand temp. = 50°C
Shortfall = 20%

=> 20% of 50°C = 10K
dt= 30min => dT= 20K
dt= 60min => dT= 10K
dt= 120min => dT= 5K
dt= 240 min => dT= 2.5K
dt= 480 min => dT= 1.25K
dt= 1440 min => dT= 0.42K



In other words, there is a demand if the demand temperature (current temperature) is 20°K below the set value for 30 minutes or 0.42K below the set value for 1440 minutes (= one day).
If the value falls below the double shortfall * set value demand temperature (corresponds to the value at 30 minutes), the characteristic curve is limited. If the difference between the set value demand and the current value of the demand temperature is greater than the double shortfall * set value demand temperature, the burner is started immediately (such as when the heating circuit switches from setback to standard mode or when a shutdown condition is no longer fulfilled and the heating circuits go into operation again).

In practice, neither the demand temperature nor the set value are constant. The difference between the two values over time will normally become greater, thus making the product of dt*dT greater, which is added to the register of sums and compared to the characteristic curve. This does not happen if the heating circuits switch, such as from standard mode into setback mode, or if the heating circuit pump shuts down due to a shutdown condition, etc. However, in such cases the energy is saved that the burner would have consumed if immediately demanded as soon as the set value has been underrun. In certain intervals, the program adds up the difference between the set value of the demand and the current value of the demand temperature. If this sum is greater than the product of shortfall * set value demand temperature for one hour with consideration of the immediate switch-on of the burner when the double shortfall is underrun, the burner is started.

Entire menu view:

DES: HEAT.REQ.	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
DEMAND TEMP.:	
T.dem.ACT: 64.3 °C	Current temperature of the sensor T.dem.
T.dem.SET: 60 °C	(Switch-on) threshold value at sensor T.dem.
DIFF.ON: 1.0 K	Switch-on difference to T.dem. (here, 61°C)
SHUTDOWN TEMP.:	
T.off ACT: 44.3 °C	Current temperature of the sensor T.off
T.off SET: 60 °C	(Switch-off) threshold value at sensor T.off
DIFF.OFF: 9.0 K	Switch-off difference to T.off (here, 69°C)
Low end temp.:	Burner demand, if T.dem falls below this value (only effective, if
T.dem.MIN: 20 °C	T.den.SET > +5°C)
Minimum runtime	
Burner: 90 Sec	
ECO MODE:	
Shortfall: 0 %	No eco mode

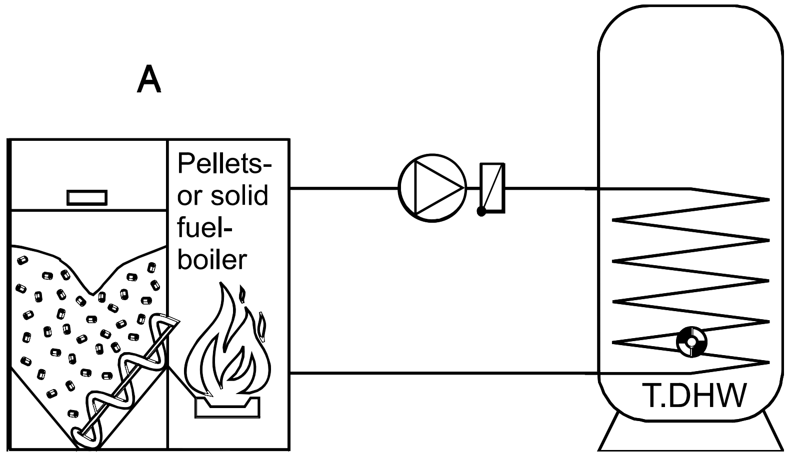
Most common example: Burner demand if the buffer cylinder is colder than the calculated heating circuit flow with the input variables:

- ENABLE / User / ON = The function is enabled
- DEMAND TEMPERATURE: = Source: / input / sensor buffer top
- SHUTDOWN TEMP.: = Source: / User / unused = only one sensor is being used
- SET VALUE DEMAND: = Source: / HEATING CIRCUIT / T.flow SET = is thermostat value

The calculated flow temperature of the function *heating circuit 1* is given as the set value (as the thermostat threshold). The controller compares this value to the "demand" temperature *buffer top* along with the switch-on/off difference. Thus, the burner is demanded when the cylinder is colder than the calculated set flow temperature + DIFF.ON and is switched off if the cylinder is hotter than the set flow temperature + DIFF.OFF.
If the boiler sensor is entered instead of the cylinder sensor, the result is a sliding boiler operation, for which a low end temperature can also be set.

DHW demand

Basic diagram:



Input variables:	Output variables:
Enable DHW demand **DHW temperature** = T.DHW Set temperature = Desired DHW temperature External Switch = Switching between "stand-ard operation" according to the time program (Status: OFF) and demand only of T.DHW MIN (Status: ON)	Effective set temperature = Time-related DHW set value T.DHW EFF Set Temperature = Desired cylinder temperature T.DHW SET Status demand, indication of the output A Burner output = Assignment only makes sense to analogue outputs A15 or A16

Simple description of the function:

Release of burner A if temperature in the cylinder (DHW temperature T.DHW) falls below the set temperature set via the time window.

Special features:

- The set temperature is also defined as an input variable in this function block. If it is only to be used as a simple set value, you need only enter *User* as the "source." Now, it will appear as a common function parameter in the function menu.
- The set temperature indicates the "desired temperature" within definable time windows. In order to ensure a minimum cylinder temperature even outside the time windows, T.DHW MIN (minimum DHW temperature) can be used to generate a demand outside these specified times.
- The effective set temperature T.DHW EFF currently set by the time window is available as an output variable. If the cylinder exceeds this temperature, 5°C is output. The burner can then be demanded via another module (such as heating demand) by comparing T.DHW EFF with the buffer temperature.
- The *set value* T.DHW SET as another output variable is the temperature that the user determines. Therefore, the setting of the desired cylinder temperature can be transferred to other function modules.
- The input variable "**EXTERNAL SWITCH**" can be used via a remote wage to switch between standard operation in accordance with the time program and a demand based only on T.DHW MIN (for instance vacation).

- Neither thermostat threshold has hysteresis, but rather a common switch-on/off difference to the adjustable threshold value.
  Example: T.DHW SET = 50°C
  DIFF. ON = 1.0 K
  DIFF.OFF = 8.0 K
  In other words, if the temperature falls below T.DHW 51°C (= 50°C + 1°K), the output is activated, while it is switched off if the temperature exceeds 58°C (= 50°C + 8°K).
- The function block provides the burner output as an output variable. It can be assigned to a speed output or to the analogue output. For instance, the burner performance can be controlled (assuming you have the appropriate burner technology) via hardware output 15 or 16 (analogue output 0 - 10V). This is recommended when the burner's performance compared to the heat exchanger's performance sets off the excess temperature protection in the boiler.
- It is also possible to charge the cylinder outside the programmed time window to the set temperature once by pressing the key.

Entire menu view:

DES: WW-REQ FUNCTION STATUS: INPUT VARIABLE: OUTPUT VARIABLE:	
SINGLE CHARGING: START	Charge the cylinder outside of the main time by pressing the key
DHW TEMP.:	
T.DHW ACT: 54.3°C	Current temperature of the DHW cylinder
T.DHW SET: 50 °C	Set temperature of the DHW cylinder
TIME PRG.:	Opens the time menu (see **Time programs**)
T.DHW MIN: 40 °C	Minimum temperature of the DHW cylinder
DIFF.ON: 0.0 K	Switch-on difference to T.DHW SET and T.DHW MIN
DIFF.OFF: 4.0 K	Switch-off difference to T.DHW SET and T.DHW MIN
Burner outp.: 100 %	Specification for burner performance

Code for Technicians:

In order to enable all of the setting parameters, open the function "User" in the device's basic menu and then select "Technician". Enter the product of 2^6 as the code!

Boiler cascade

Simple description of the function:

The coordination of up to three burner demands based on runtime and delay time by comparing the current demand temperature with a common flow temperature.
The indication of the functions involved (demand modules) automatically gives the module permission to control the burner via its internal signals "burner demand" and "set temperature." The highest set temperature is compared to the common flow temperature and issues a burner demand if need be. After the set delay time, the next burner stage is enabled if the conditions have been fulfilled, etc.

Input variables:	Output variables:
Enable boiler cascade (starting with first boiler stage)	Set Value = Highest demand temperature
Enable from second, third boiler stage	Status burner demands for boiler A, B, C Indication of the outputs
Flow temperature = Common flow	Status of boilers (1, 2, 3) = Status of burners demanded
involved functions = Indication of the demand modules involved	

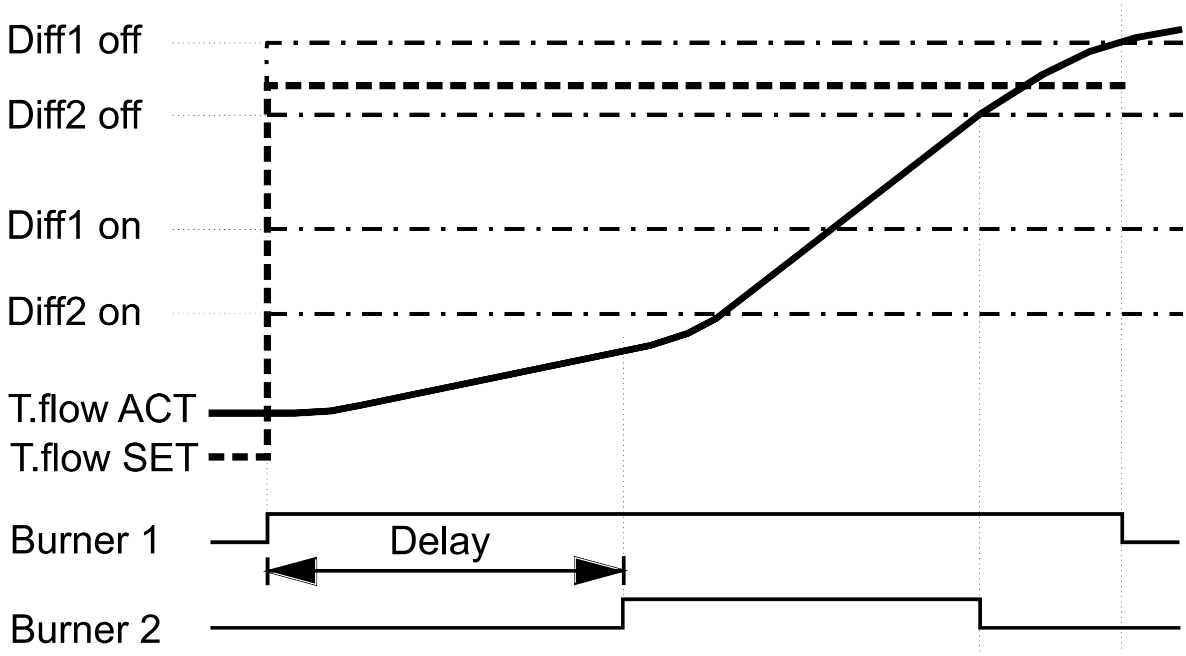
Special features:

- Few input variables as the module automatically communicates with them internally by means of the indication of the functions involved.
- Recording of the burner runtimes. You can therefore automatically switch which boiler is dominant by putting a limit on runtime.
- In addition to the necessary burner demands, the highest demand temperature (set flow value) and the switched stages are available as output variables.

Warning:

Sometimes it makes sense to link one of the output variables directly to the analogue output to create a 0 to 10 V or PWM signal. Linking this function is only allowed using analogue output A15, not with output A16.

The following runtime chart assumes that two boilers are to be controlled:



If there is demand (flow set temperature T.flow SET suddenly increases) and the flow temperature is less than the switch on temperature of the controlling boiler (=T.flow SET + DIFF1 ON), the first demand is generated. If the flow temperature remains under the switch-on temperature for the second boiler (T.flow SET + DIFF2 ON), a second demand is generated. The boilers are switched off in the same sequence as the flow temperatures exceed the switch-off temperatures (T.flow SDET + DIFF OFF).

The set flow temperature **T.flow SET** is linked with the following values of the involved functions and is determined from the highest of these temperatures:

1. From the function module **Heating demand**:
   Switch off temperature T.off SET+ DIFF.OFF,
   or demand temperature T.dem.SET+ DIFF.OFF, if no self-provided sensor is used for switching off,
   or base temperature T.dem.MIN + DIFF.OFF.
   The demand itself is generated, when the temperature falls below the demand temperature T.dem.SET + DIFF.ON or below the base temperature T.dem.MIN + DIFF.ON. An approximate burner minimum running time is not considered.
2. From the function module **DHW demand**:
   DHW set temperature T.DHW SET + DIFF.OFF
   or minimum temperature T.DHW MIN + DIFF.OFF (outside the time window)
   The demand itself is generated when the temperature falls below the hot water set temperature T.DHW SET + DIFF.ON or the minimum temperature T.DHW MIN + DIFF.ON.

If no demand arises out of the involved functions or the release is set to "OFF", then T.flow SET is +5°C.

Overall menu view (for two boilers as shown in the chart):

DES: BOIL.CASC	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
SERVICE MENU:	
T.flow ACT: 34.6 °C	Current flow temperature
T.flow SET: 55 °C	Desired set flow temperature from the demand
BOILER 1:	
DIFF.ON: -8.0 K	Switch-on difference to T.flow SET (here, 47°C)
DIFF.OFF: 2.0 K	Switch-off difference to T.flow SET (here, 57°C)
Time delay: 0 sec	Switch-on delay for the first boiler (usually zero)
BOILER 2:	
DIFF.ON: -13 K	Switch-on difference to T.flow SET (here, 42°C)
DIFF.OFF: -1.5 K	Switch-off difference to T.flow SET (here, 53.5°C)
Time delay: 15 min	Switch-on delay for the second boiler 15 minutes

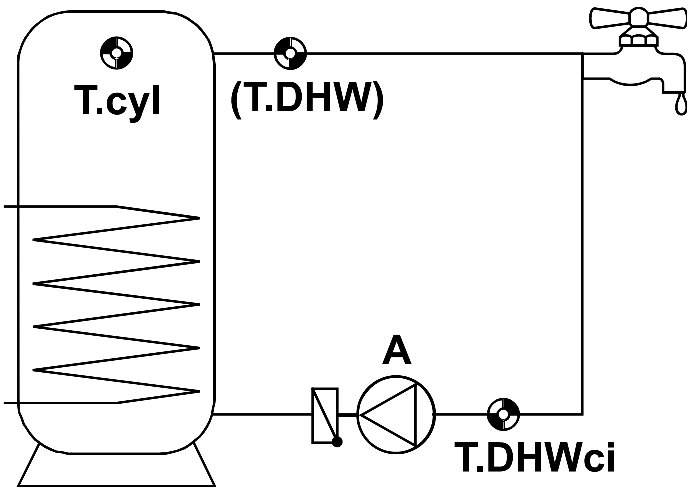
Boiler cascade

The variables in the service menu (according to example):

BOIL.CASC	
Boiler sequence:	
Boiler A: 1	Boiler A is top priority (= dominant boiler)
Boiler B: 2	Boiler B is second priority
Boiler A:	
automatic boiler change: yes	Change dominant boiler if A - B = 200 hours
Operating time 284 hrs	Total boiler runtime A = 284 hours
RESET METER: no	"yes" to reset the meter
Boiler B:	
automatic boiler change: yes	Change dominant boiler if B - A = 200 hours
Operating time 91 hrs	Total boiler runtime B = 91 hours
RESET METER: no	"yes" to reset the meter
Difference hours run for boiler change: 200 hrs	Once a difference of 200 operating hours between A and B is reached, the control boiler is changed over, provided an automatic boiler changeover is required (setting: yes).

DHW circulation

Basic diagram:



Input variables:	Output variables:
Enable DHW circulation **Return Temperature** = T.DHWci DHW temperature = T.DHW Set DHW circulation return temperature = maximum temperature allowed at T.DHWci Cylinder Temperature = T.cyl cylinder sensor for mixture protection	Effective set temperature T.DHWci.eff (takes into account also the mixture protection) Status DHW circulation, indication of the output

Simple description of the function:

Release of circulation pump A via time window as long as return sensor T.DHWci has not reached its upper limit (set temperature). In a simple application, the DHW sensor T.DHW does not have a function and is thus not used.

Special features:

- **Mixture protection 1**: Below a minimum cylinder temperature (T.cyl.MIN), the circulation function is blocked so that the layered remaining energy in the cylinder is not lost through pump operation.
- **Mixture protection 2**: To prevent a mixture above this threshold, the temperature difference is used between the cylinder temperature and the return temperature (DIFF.MIXT). If the cylinder temperature minus "DIFF.MIXT" is less than the set return temperature T.DHWciSET, this value is considered the limit temperature. Mixture protection is deactivated without a cylinder sensor ("source" user).
- If hygienic warm water is to be provided instead of a DHW cylinder, the pulse mode can be used as an alternative control method using a DHW sensor **T.DHW**. This approach requires a properly dimensioned plate heat exchanger including an **ultrafast** temperature sensor (MSP... = special accessory) at the warm water outlet. **T.DHW** also controls the water heating and circulation.

  If a faucet is opened, the temperature at **T.DHW** changes. If the temperature rises or falls by a set value within 1 second at **T.DHW**, the controller switches on the circulation pump. The pump is then switched off either after a set runtime or once the preset set value on **T.DHWciSET** has been exceeded. This ensures that DHW is available quickly at the faucet even when it is closed.
- In the **Time/pulse** mode, the time mode is active within the time window; the pulse mode, outside.

DHW circulation

Entire menu view:

DES: DHW CIRC.	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
OPERAT: Time	Switching to "Pulse" or "Time/pulse" mode (see special features)
DHW CIRC. RTN:	
T.DHWciACT: 34.7 °C	Current temperature of the return
T.DHWciSET: 50 °C	Set (maximum) temperature of the return
TIME PRG:	Open the switch time menu
DIFF.ON: 0.0 K	Switch-on difference to T.DHWciSET (here, 50°C)
DIFF.OFF: 5.0 K	Switch-off difference to T.DHWciSET (here, 55°C)
DHW TEMP.:	
T.DHW ACT: 53.2 °C	Current DHW temperature

Other menu lines appear when a sensor has been indicated for the cylinder temperature:

MIXT PROTECTION:	
T.cyl.ACT: 58.2 °C	Current temperature of the cylinder
T.cyl.MIN: 30 °C	No circulation is allowed below this cylinder temperature (Hysteresis = 3K)
DIFF.MIXT: 8.0 K	If the cylinder temperature minus DIFF.MIXT falls below T.DHWciSET, the new value is “T.DHWciSET” (= Effective circulation return temperature)

For the option *Pulse* in lieu of *Time* the following menu options are displayed in place of the time program:

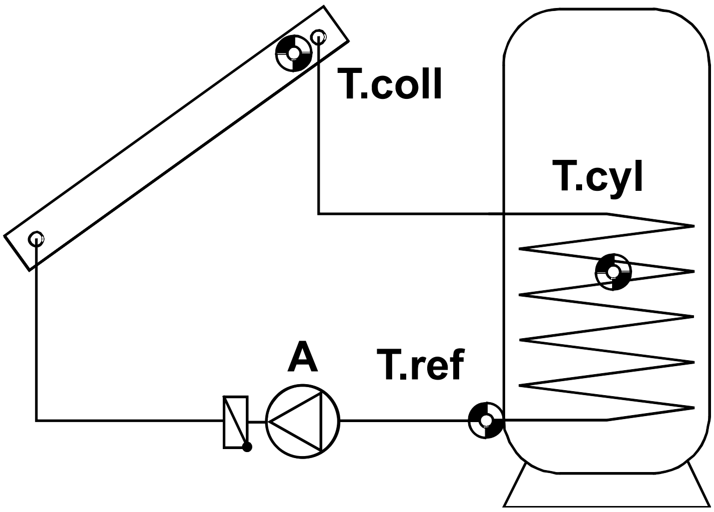
Ddiff_on: 2.0 K	Temperature change of 2K / seconds starts the pump
Runtime: 90 Sec	Maximum runtime per interval
Pause time: 10 Min	Minimum interval time (= minimum time between two pump runs)

If the operating mode *Time/pulse* is selected rather than *Time*, then within the time window, time mode is active, while outside it pulse mode is active.

In combination with hygienic service water preparation, the pulse mode runs very reliably with an ultrafast Sensor (special accessory). If standard sensors are used, the temperature change is recognized much more slowly. If there are problems, a flow switch for the circulation function can also be used instead of temperature measurement.
The suddenly increasing digital signal of the flow switch at the function input "DHW temperature T.DHW" causes immediate switching on of the circulation pump.

PID control (speed control)

The PID control can be used to change the delivered quantity – i.e. the volume flow – of usual commercial circulating pumps. That allows the system to maintain temperatures (differences). It can be used not only for the speed control but also for the control of the burner performance and others.
The following simple solar diagram illustrates the possibilities of this process:



Absolute value control = Stabilizing a sensor

T.coll can be stabilized very well at one temperature (such as 60°C) via the speed control. If solar radiation decreases, **T.coll** becomes cooler. The control unit reduces the speed and hence the flow rate. That leads to a long dwell time for the heat transfer medium in the collector, which in turn increases **T.coll**.
As an alternative, a constant return (**T.ref**.) can make sense in various systems (such as cylinder load). That requires an inverse control characteristic. If **T.ref** increases, the heat exchanger adds too little energy to the cylinder. Thus the flow rate is reduced. Greater dwell time in the exchanger cools the heat transfer medium more, thus reducing **T.ref**.
Stabilizing **T.cyl** would not make sense because the varying flow rate would not affect **T.cyl** directly and thus not lead to a functioning regulating circuit.

Differential control = Keeping the temperature between two sensors stable.

Keeping the temperature difference constant between, for example, **T.coll** and **T.ref** leads to a "sliding" operation of the collector. If **T.coll** drops as a result of reduced irradiation, the difference between **T.coll** and **T.ref** also falls. The control unit reduces the speed leading to a greater dwell time of the medium in the collector and hence to a greater difference **T.coll** – **T.ref**.

Event control = If a set temperature event occurs, the speed control starts, thus keeping a sensor constant.

If, for instance, **T.cyl** reaches 60°C (activation threshold), the collector is to be kept stable at a certain temperature. The respective sensor is stabilized just as in the absolute value control.

Note: If the absolute value control (stabilization of a sensor) and the differential control (stabilizing the difference between the two sensors) are both active at the same time, the slower speed of the two methods "wins". The event control "overwrites" the speed results from other regulation methods. This means that a defined event can block the absolute value or differential control.

PID control

Waveform

Two waveforms are available for motor control (in the menu “Outputs”).

Wave packet - only for circulating pumps with standard motor dimensions. Here, individual half-waves are blended in to the pump motor. The pump is run via pulses; the “smooth running” of the motor is only due to the moment of inertia.
Benefit: Great dynamics of 1:10 well suited for usual commercial pumps without internal electronics and a motor length of around 8 cm.
Drawback: Linearity depends on the pressure loss; there is some noise, not suitable for pumps with evidently deviating motor diameters and / or length from 8 cm.

Phase angle - for pumps and ventilation motors. The pump is switched to the mains within each half-wave at a certain point (phase).
Benefit: Suitable for almost all types of motors
Drawback: Low dynamics of 01:03 for pumps. A filter has to be inserted upstream from the unit with at least 1.8mH and 68nF to fulfill the CE standards for interference suppression (except A1 which on the other hand is able to carry a current of only up to 0.7A)
NOTICE: The menu allows a choice between wave packet and phase angle however in the standard version the output of waveform "phase angle" is not possible. Special versions on request.
Speed control via a phase angle control is not possible at outputs 2, 6 and 7.

Stability problems

The **proportional part P** represents the reinforcement of the deviation between the set and the actual value. The speed is changed in increments every **X***0.1 K of deviation from the set value. A large number makes the system more stable and leads to a greater deviation from the regulation.

The **integral part I** periodically adjusts the speed relative to the deviation remaining from the proportional part. For **1** K of deviation from the set value, the speed is changed one increment every **X** seconds. A larger number makes the system stable, but the adjustment to the set value is slower.

The **differential part D** leads to a short-term overreaction the faster a deviation occurs between the set and the actual value in order to provide the fastest compensation possible. If the set value deviates at a speed of **X***0.1 K per second, the speed is changed by one increment. Large numbers provide a more stable system, but it then takes longer to reach the desired value.

In some cases an adjustment of the PID values is necessary. The pump should be running in automatic mode if the system is ready for operation and has the appropriate temperatures. With I and D are set to zero, the proportional part P is reduced from 10 each 30 seconds until the system is instable and the pump speed changes its rhythm. It can be read in the menu above the PID parts. The proportion at which instability begins is noted as P_{crit}, with the duration of the oscillation (= time between the two highest speeds) noted as t_{crit}. The correct parameters can be determined with the following formulae.

$$P = 1{,}6 \times P_{crit} \qquad I = \frac{t_{crit} \times P}{20} \qquad D = \frac{P \times 8}{t_{crit}}$$

A typical result of **hygienic service water preparation** with an ultrafast sensor is PRO= 8, INT=9, and DIF=3. For reasons not entirely understood, the settings PRO=3, INT=1, and DIF=4 have proved to be effective. Probably, the control unit is so instable here that it oscillates very quickly, appearing balanced due to the systems and the fluids inertia.

Pump standstill

The wave packet method (standard) allows the volume flow to be changed by a factor of 10 in 30 increments. If the flow rate is too low, return flaps may cause a standstill as well as a low output of the pump at low speeds of the control unit. This may even be desirable, which is why the lower limit of 0 is admissible. A reasonable speed limit is found in a simple test. Select manual mode in the menu “Outputs“ and set a speed. The rotor can be observed with its hood removed. Now, the revolutions are reduced until the rotor comes to a standstill. Three stages above this limit will provide safe pump operation. The indication of the lower speed stage occurs in the respective function *Speed control*.

Input variables:	**Output variables:**
Enable PID control	Correcting variable = Calculated speed stage
Temperature ***absolute value control*** = Sensor which should be kept stable at the set temperature. Set value absolute value control = Desired control temperature	Definition of the speed control control output
Temperature (+) ***differential control*** = Base sensor (the warmer sensor, e.g. collector) of the differential control Temperature (-) differential control = Reference sensor (the colder sensor, e.g. cylinder) of the differential control	
Activation temperature ***event control*** = Sensor where an event is expected. Activation threshold = Temperature event at the above sensor Control temperature event control = Sensor which is kept stable after the occurrence of an event. Set value event control = Desired set regulation temperature for the event control	

Simple description of the function:

With the indication of temperature sensors the volume flow in the hydraulic system is controlled via the variability of the pump speed to keep the respective sensor constant at a set temperature.

Special features:

- The **correcting variable** is available as output variable also for other functions for further use. In addition, it also can be linked to an analogue output (A15, A16) instead of the pump outputs.
- It is possible to set all control processes separately to ***standard*** control operation (speed increases along with the temperature), to ***inverse*** operation (speed decreases as temperatures increases) or to ***OFF*** (control process is not active).
- If the absolute value control (stabilization of a sensor) and the differential control (stabilizing the difference between the two sensors) are both active at the same time, the “slower speed wins out”.
- If 2 PID controllers act simultaneously on an output, then the faster speed "wins".

PID control

- Event control “overwrites” the results of speed control from other control processes. Hence, a set event can block the absolute value control or differential control. **Example**: Keeping the collector temperature at 60°C with the absolute value control is blocked when the cylinder has already reached 50°C at the top = the fast provision of DHW is complete and is now to be continued with full volume flow (and hence a lower temperature but slightly better efficiency). To do so, a value that automatically requires full speed has to be entered as the new set temperature in the event control (e.g. coll = 10°C).
- If absolute value as well as differential control are switched off (output: maximum correcting variable), then there will be a switch from maximum correcting variable to the value that complies with the event control when the event control is activated.

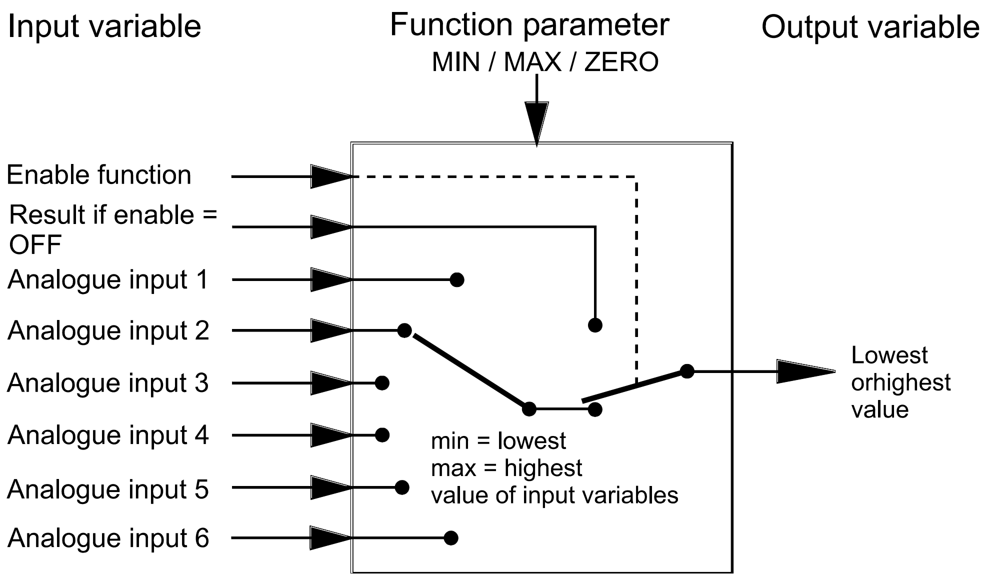
Entire menu view:

DES: PID CTRL	
FUNCTION STATUS:	
INPUT VARIABLE:	
OUTPUT VARIABLE:	
ABSOL.VAL. CTRL:	
MODE: std	Speed increases as temperature increases
T.abs.ACT: 50.3 °C	The sensor is currently measuring 50.3°C
T.abs.SET: 50 °C	The sensor is kept at 50°C
DIFFERENTIAL CONTROL:	
MODE: std	The speed increases with the difference T.diff+ to T.diff-
T.diff+.ACT: 50.3 °C	The sensor at the source is currently measuring 50.3°C
T.diff-.ACT: 42.7 °C	The reference sensor is currently measuring 42.7°C
SET DIFF 8.0 K	The desired difference (T.diff+ to T.diff-) should be 8 K
EVENT CONTROL:	
MODE: off	No event control is allowed. If *std,* then:
COND.: ACT < THRES	Condition
T.activACT: 48.1 °C	The sensor that activates the function is measuring 48.1°C
T.activTHR: 60 °C	Event control should launch at 60°C on (act-) sensor
	(fixed activation threshold, no hysteresis)
T.ctrl ACT 50.3 °C	The sensor control starting with the event is displaying 50.3°C
T.ctrl SET 90 °C	Starting with this event, the sensor is controlled to 90°C
CORRECTVAR.:	
maximum: 30	Highest admissible speed stage is the stage 30 (full run)
minimum: 8	Lowest admissible speed stage is stage 8 (0 also admissible)
current: 14	Stage 14 is currently output
CONTROL PARAMETER:	
P: 10 I: 0 D: 0	PID shares for stable operation

The control parameters P=8, I=5, and D=2 generally ensure stable operation. If the speed periodically changes (generally for 20-30 seconds), I and D should be set to zero for simple systems. Drawback: A low, constant temperature is incorrectly controlled for, and the system is a bit slower.
If a speed control is used in service water systems, the PID settings must be determined in a test (see "stability problems") to ensure optimal performance.

Analogue function

Basic diagram:



Input variables:	Output variables:
Enable analogue function Result (Enable = Off) Analogue input variables 1 - 6	Result Indication of the speed control output

Simple description of the function:
According to the basic diagram it looks for the highest (lowest) value of the analogue inputs. Besides the heating circuit and charging pump module this module represents an extremely versatile and important link to the burner demand. In addition, it offers also simple calculating operations.

Special features:
- When an entry is made in the function list, the number of analogue inputs can also be entered. Thus, not all six inputs have to be assigned.
- The function produces the following result as an output variable via a control command from the inputs:
  - ***MIN***: Output of the minimum value of the input variables.
  - ***MAX***: Output of the maximum value of the input variables.
  - ***AVERAGE***: The output variable is the **mathematical average** of all input variables. In this way, the average of multiple measurement values can be calculated.
  - ***FILTER***: The output variable is the **temporal average** of the first input variable. All other inputs are ignored. The average time can also be set.
  - ***SUM***: The output variable is created based on the following formula from the sum of input variables E(1-6): sum = E1 - E2 + E3 - E4 + E5 - E6. For example: if the input variable E2 is set to *User* and zero is entered in the parameters for E2, E1 + E3 are simply added.
  - ***ZERO***: Output of the number zero as an output variable.
- If the module is blocked (enable = off), a value is output that either the user determines via "RESULT (ENABLE: = Off)" or that is created from an own input variable. It is therefore possible to switch between analogue values via the enable.
- The specification of *User* and analogue input creates an adjustable numeric value in the menu of the function.
- In the input variables, an offset that is added to the value of the variables can also be set.
- **Digital** conditions can also be processed at the inputs: if the condition is ***OFF***, 0 is used to calculate the value; if the condition is ***ON***, the set value “Offset” of the respective input variables is used for the calculation.

Analogue function

Example of use:

The greatest temperature that the system currently demands is calculated from the three functions "heating circuit 1." "heating circuit 2" (output variable = set flow temperature), and DHW demand (output variable = effective set temperature) so that the burner demand is later correctly compared to the buffer cylinder temperature. In addition, the customer also wishes to have a constant buffer standby temperature. When this function was called, the number of input variables was already set at four. The following parameters are now set in the submenu *INPUT VARIABLES*:

INPUT VARIABLE 1: Source: HTG CIRC.1 1: Set flow temp. Offset: 0.0 K	Input variable 1 is the set flow temperature of the function HTG CIRC. 1
INPUT VARIABLE 2: Source: HTG CIRC.2 1: Set flow temp. Offset: 0.0 K	Input variable 2 is the set flow temperature of the function HTG CIRC. 2
INPUT VARIABLE 3: Source: DHW_DEM. 1: eff.set temp. Offset: 0.0 K	Input variable 3 is the effective temperature of the function DHW_DEM.
INPUT VARIABLE 4: Source: User	The user sets the socket temperature in the menu

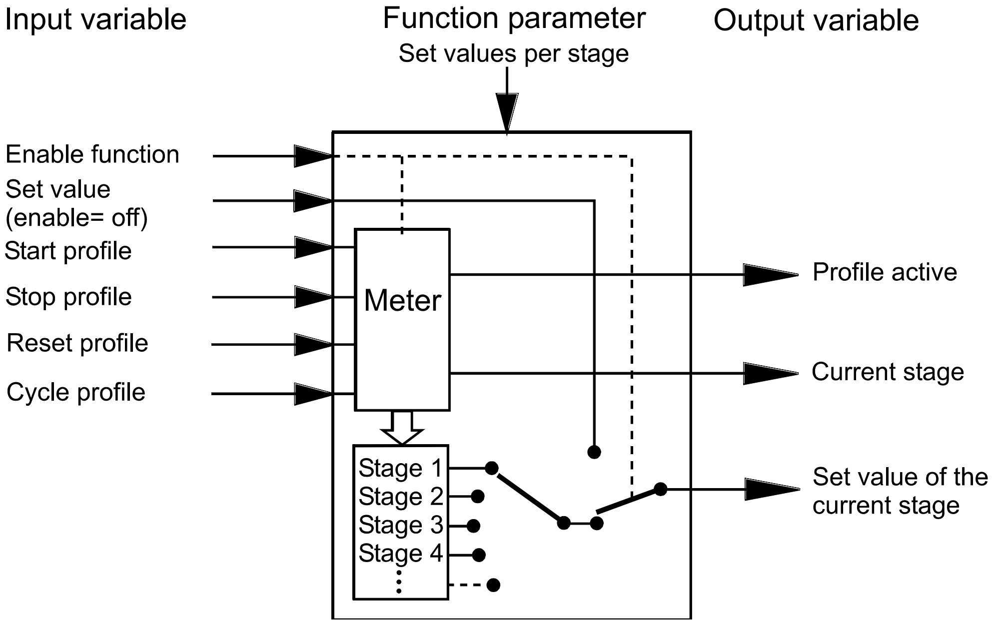
Entire menu view:

DES: MAX(an) INPUT VARIABLE: OUTPUT VARIABLE:	
FUNC.VAR: Temp.	All inputs are temperatures
FUNCTION: MAX	Output of the highest temperature of the inputs
VAR. 1: 53.6 °C	Set flow temperature of the function HTG CIRC.1
VAR. 2: 66.4 °C	Set flow temperature of the function HTG CIRC.2
VAR. 3: 5.0 °C	Effective temperature of the function DHW_DEM
VAR. 4: 40.0 °C	Low end temperature set by the user
if ENABLE = Off 0 °C	If the analogue mode has not been released, the module outputs 0°C
RESULT: 66,4 °C	Result of the analogue function

The function therefore provides 66 .4°C as the greater value for the output variable. As an input variable, this temperature now allows a comparison with the temperature at the top of the buffer cylinder in the function *heating demand*. If the buffer is colder than 66 .4°C (+ diff), the burner is required.

Profile function

Basic diagram:



Input variables:	**Output variables:**
Enable profile Set value if (Enable = Off) Start profile = Starting the time controlled process Stop profile = Stopping the time controlled process Reset profile = Reset to stage 0 (profile disabled) Cycle profile = More forward by 1 step (from step 1)	Status profile active = Output ON when set value is not OFF Definition of the output Set value = Value of the current stage Current stage

Simple description of the function:

This function generates a time-controlled output of **up to 64 values**. In each stage, the system switches from one value to the next in a set table and outputs the value as a "set value." In this way, a profile can be set up, e.g. a temperature profile that is suitable for a floor screed drying out program.

Special features:

- The input variables Start, Stop, Reset or Cycle profile must be digital commands (ON/OFF) (e.g. digital input, switching output or another function, etc.)
- Each of the input variables can be manually operated directly from the function by entering *User*. However, the command "STOP PROFILE" behaves differently than a linked input variable in manual mode. In the link, the meter is only stopped as long as the stop signal is active, thereafter the meter continues running. In the manual mode, "STOP PROFILE" also generates a reset. The meter then starts from the beginning again when restarted.
- A cyclical process is possible: the first value is called again after the last one.
- If the module is blocked (enable = off), a value is output that is either determined via "Set value if (enable = Off" or that another module creates as an input variable. It is therefore possible to use the enable to switch between the profile and an external analogue value.
- The table entry OFF means: During this step, the profile is not active. A value is output, that can either be specified via “Set value if (enable = Off)” or which originates from another module as the input variable.
- The following functional variables can be set as the set value: temperature, dimensionless, output, heat amMWh, heat amkWh, number of impulses, time and solar radiation.

Profile function

Although the profile stage is registered every 6 hours in the internal storage, it gets lost while loading new function data (load factory settings, load backup copy, data transfer from the C.M.I.).
If an internal cycle > 23.5 hours is set (such as heat drying of a pavement floor), profile stage 1 is saved in the internal memory immediately after the profile function starts. This ensures that the heat drying program continues to run when the controller is reconnected after a blackout just after the heat drying of the floor pavement has been launched.

Example:

A temperature profile for heat drying of floor pavement is to be created. It is assumed that all input variables are set to *User* so that the function can be manually changed at any time.

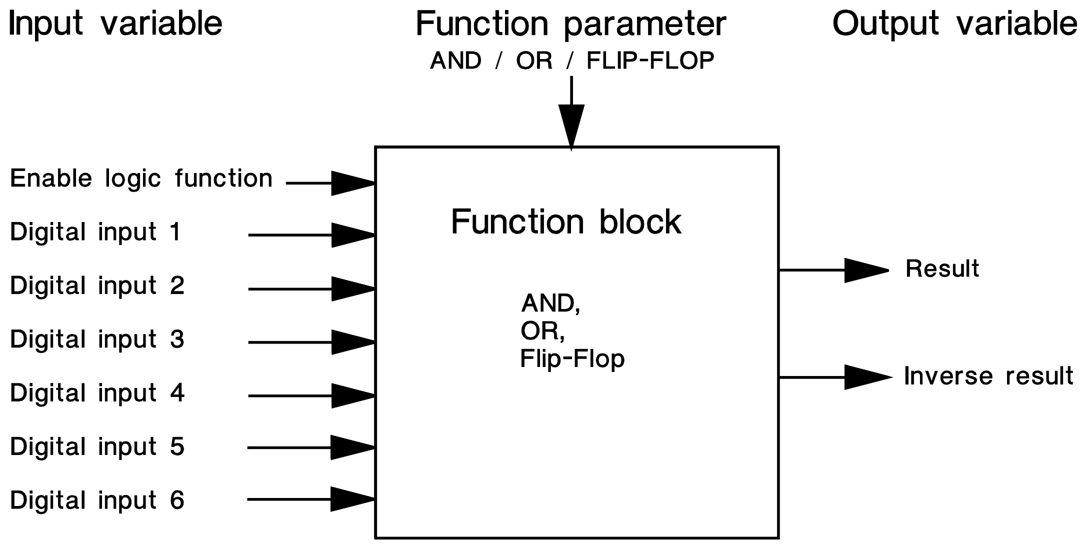
Entire menu view:

DES: PROFILE INPUT VARIABLE: OUTPUT VARIABLE:	
FUNC.VAR: Temp.	The values are interpreted as temperature
cyclical: no	No repeat after the profile has ended
Int. Sync: 24.0 Hrs	Switch to the next value every 24 hours (Adjustment range 1 sec. to 48 hours)
START PROFILE	Press the scroll wheel to manually start the function The following will appear when started: STOP PROFILE (Display only if the input variable "Start Profile“ is set to *User*)
CURRENT STAGE: 3	
SET VALUE: 26.0 °C	The set value of stage 3 is 26°C
Stage 1: 20 °C Stage 2: 23 °C Stage 3: 26 °C Stage 4: 30 °C Stage 5: 35 °C	
Stage 6: OFF	No profile is active on the sixth day, output of the set value, if Enable = OFF
Stage 7: 30 °C Stage 8: 26 °C Stage 9: 22 °C	
if ENABLE = off 0.0 °C	If the profile mode has not been released, the module outputs 0°C

If the output variable "PROFILE ACTIVE" is now assigned to the heating circuit pump and the function module "MIXER CONTROL" handles the set value, a pavement floor heat drying program has been created for nine days. Here, make sure that a heating circuit control function does not simultaneously control the output. To be on the safe side, set the enable for the heating circuit control function to *User OFF* during the process.

Logic function

Basic diagram:



Input variables:	**Output variables:**
Enable logic function Digital input variables 1 - 6	Status result, indication of the output Status inverse result, indication of the output

Simple description of the function:

AND- function:	Output = ON only if all inputs are ON.
OR- function:	Output = ON if at least one input is ON.
FLIP FLOP- function:	Output = Saves the status of the inputs

Special features:

- After having registered the function in the function list the indication of the number of digital inputs is possible. Thus, not all six inputs have to be assigned.
- The **FLIP-FLOP** function (also known as holding circuit) works according to the following formula:
  - Output = constantly ON if at least one of the inputs I1, I3, I5 were set to ON (set holding circuit), even if the input decreases again afterwards (set pulse).
  - Output = constantly OFF if at least one of the inputs I2, E4, I6 were set to ON (delete holding circuit). The “Delete” command dominates. Hence, no setting is possible when a delete input is ON (reset pulse).
- The function “OFF” is also available. In this way, the function is inactivated by the easiest means. The status *OFF* applies to the direct output and the status *ON* to the inverse output.
- In addition to the direct output, the inverse output variable is also available.
- If the module is blocked via enable, both the direct and the inverse output are ***OFF***.

Logic function

Example:

The heating circuit is to be released based on the two thermostat functions "comparison_1" and "comparison_2" when one of the two (OR-function) is triggered. When this function was called, the number of input variables was already set at two. The following parameters are now set in the sub-menu *INPUT VARIABLE*:

Display	Explanation
`INPUT VARIABLE  1:`	
`Source: COMP.1`	Input variable 1 is the output of the thermostat function COMP.1
`1 : Va > Vb + diff`	
`Mode:   normal`	Adoption of normal start condition of the module
`Status:   ON`	We see current status ON
`INPUT VARIABLE  2:`	
`Source: COMP.2`	Input variable 2 is the output of the thermostat function COMP.2
`1 : Va > Vb + diff`	
`Mode:   normal`	Adoption of normal start condition of the module
`Status:    OFF`	We see current status OFF

Thus, the function forms the command ON as the output variable. In the function *heating circuit controlr*, it now allows the release of the pump as an input variable if either the "boiler thermostat" **or** the "buffer thermostat" has exceeded the required temperature.

Value table based on two inputs + enable:

AND

Enable:	Input 1:	Input 2:	Result:	Inv. result	Commentary:
ON	OFF	OFF	OFF	ON	
ON	ON	OFF	OFF	ON	
ON	OFF	ON	OFF	ON	
ON	ON	ON	ON	OFF	
OFF	X	X	OFF	OFF	Both outputs OFF

OR

Enable:	Input 1:	Input 2:	Result:	Inv. result	Commentary:
ON	OFF	OFF	OFF	ON	
ON	ON	OFF	ON	OFF	
ON	OFF	ON	ON	OFF	
ON	ON	ON	ON	OFF	
OFF	X	X	OFF	OFF	Both outputs OFF

FLIP FLOP

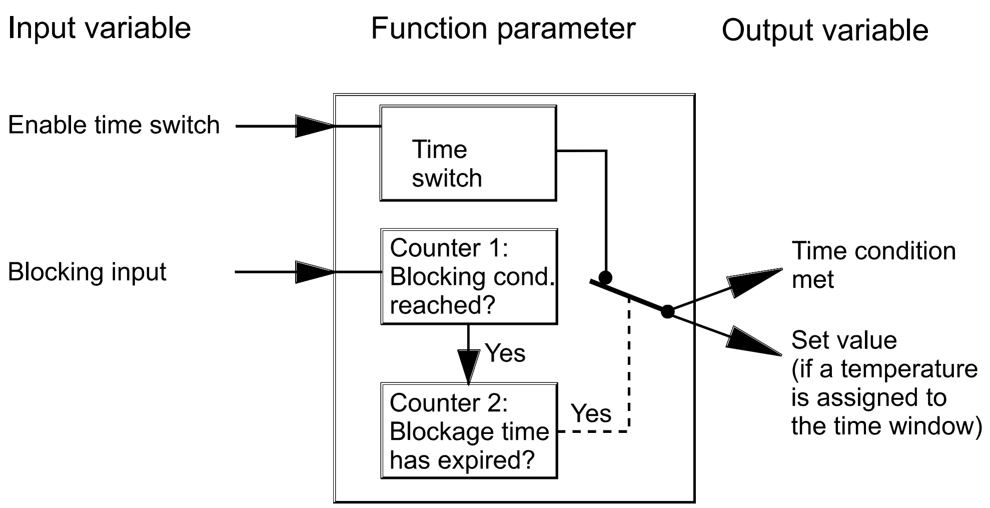
Enable:	Input 1:	Input 2:	Result:	Inv. result	Commentary:
ON	OFF	OFF	OFF	ON	Previous status
ON	ON	OFF	ON	OFF	I1 is saved!
ON	OFF	OFF	ON	OFF	Previous status
ON	OFF	ON	OFF	ON	I2 deletes output
ON	ON	ON	OFF	ON	I2 dominant
OFF	X	X	OFF	OFF	Both outputs OFF

OFF

Enable:	Input 1:	Input 2:	Result:	Inv. result	
ON	X	X	OFF	ON	
OFF	X	X	OFF	OFF	Both outputs OFF

Time switch

Basic diagram:



Input variables:	**Output variables:**
Enable time switch	Set value (if a temperature is assigned to the time window) Status time condition met
Blocking input	Indication of the output

Simple description of the function:
There are a maximum of 5 time programs each with 3 time windows available per module.
As a freely usable time switch clock this function can be employed in various ways. For instance, this function could be used to provide a time control for filter pumps in swimming pools or for fan motors in hot-air heating systems. The function block has the same operational structure as all other time switch functions, such as in the heating circuit control function.
If the function “time switch” is put before another function (such as “charging pump”) as an INPUT VARIABLE/ENABLE, the function receives additional time conditions. As with all other function modules, the time switch can be entered several times in the function list, i.e. multiple time switches can be available.

Special features:
- When the function is set up, the question "with set value?" *yes/no* appears next to the questions about scope (time programs, window). If you enter *no*, you'll have a normal digital time switch. If yes is entered, the user can assign a temperature to each time window that will later be available as an output variable according to the time windows. A set value can be entered simultaneously if the time program is not completed.
- If *User* is entered as the "source" for the input variable BLOCKING INPUT, a simple time switch function is the result.
- If another function is assigned as the "source" to the input variable BLOCKING INPUT, the time switch can be blocked for a certain period if certain events occur.

Time switch

Example:
Time switch with two time programs, each with three time windows

Entire menu view:

Display	Explanation
DES: TIME INPUT VARIABLE: OUTPUT VARIABLE:	
Mo Tu We Th Fr Sa Su 06.00 - 07.30 h 12.00 - 21.00 h 00.00 - 00.00 h	The first time program is active on all workdays Switched on on workdays at 06:00 AM and off at 07:30 AM etc. Time window not used
Mo Tu We Th Fr Sa Su 05.00 - 07.00 h 12.00 - 22.00 h 00.00 - 00.00 h	The second time window is active on the weekend Switched on at 05:00 AM and off at 07:00 AM etc. Time window not used

If using a set value, the following lines appear according to the time matrix:

Display	Explanation
Set val if TP not cmpl.: 5° C	Entry of a set value outside the time window, during blockage or with enable = OFF

If another function uses the blockage input, the following will appear:

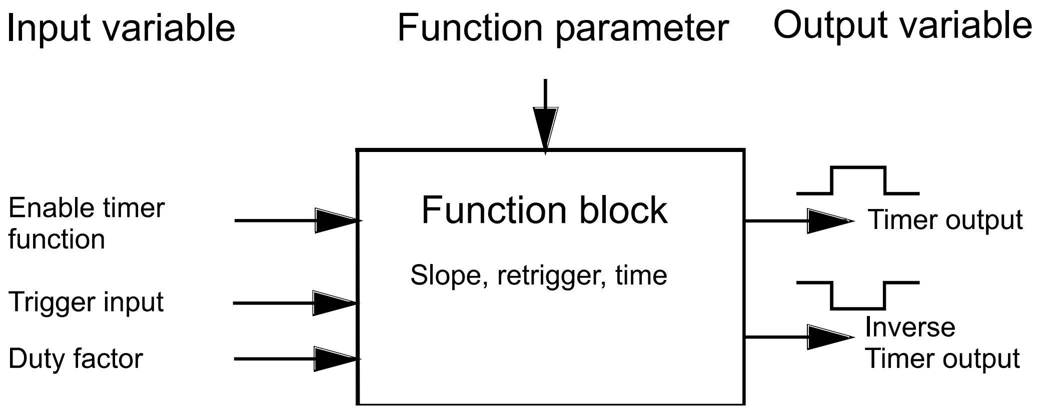
Display	Explanation
Min.time block.cond: 0 Days 5.0 min	The condition has to be met for at least five minutes
Blocking Time: 0 Days 10.0 hr	Then, the time switch is blocked for 10 hours

Legionella protection is another **example**. Here, the time switch function is used to heat up the cylinder to 60°C at night to prevent legionella from forming. If this temperature is reached anyway during the day (such as by the solar system), there is no need of additional heating, and the function is blocked:
A comparison function (thermostat) at the blocking input lets the first meter run ("Min.time block.cond") as long as the cylinder is hotter than 60°C. If the set meter time has been reached (five minutes), a second time meter blocks the time switch until it has expired (10 hours). The cylinder is therefore not heated up once again at night using oil, gas, or electricity if the protective temperature was already reached during the day.

While the time switch is already blocked once the first meter time ("Min.time block.cond ") has been reached, the second meter (blocking time of time switch) only begins to run when the blockage input returns to the status "OFF."

Timer function

Basic diagram:



Input variables:	**Output variables:**
Enable timer Trigger input = Input signal for starting the timer Duty factor = Relation between input and output signal	Status Timer Output, indication of the output Status Inverse Timer Output, indication of the output

Simple description of the function:

Independent time elements can switch time sequences between functions. An input status triggers a time laps of the timer function (= impulse time), which runs irrespective of the time of day. The impulse time can be set up to 90 seconds in increments of seconds in addition to various stages of up to 48 hours.

Special features:

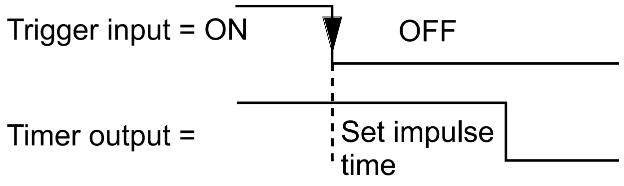
- The set impulse time can be varied from 0-100 percent via the input "DUTY FACTOR." The impulse time can thus be varied via signals and computer values. To make it an adjustable value in the menu, *User* has to be entered as "source".
- Use the command MODE to choose between six basic functions.
- If enable = OFF, then both output variables are set to OFF.

Entire menu view:

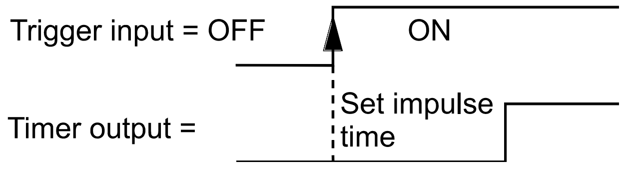
DES: TIMER FUNCTION STATUS: INPUT VARIABLE: OUTPUT VARIABLE:	
MODE: Delay	Input affects output with a certain delay
TRIGGER: retrigger: yes	A second trigger while the timer is running restarts the timer
IMP. TIME: 8 Sec DUTY FAC.: 100 %	Timer runtime 100% of 8 seconds = 8 seconds!
MAN.: TIMER START	The timer can be launched using the scroll wheel and stopped before it has expired.

Timer function

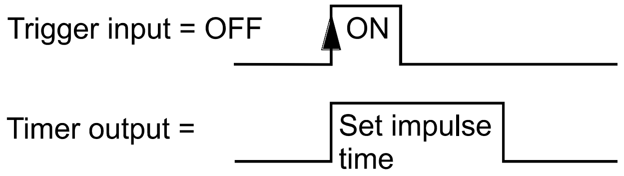
Run-on time: The ON signal at the trigger input immediately switches the output on. If the input drops (OFF), the output remains ON for the duration of the timer period.



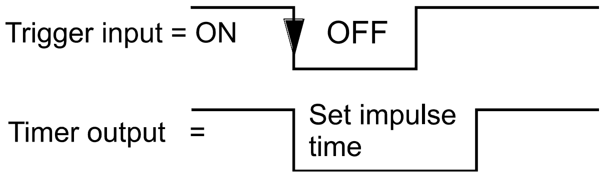
Delay: The ON signal at the trigger input is only passed on when the timer period at the output has elapsed. An OFF signal at the trigger input causes immediate switching off of the output.



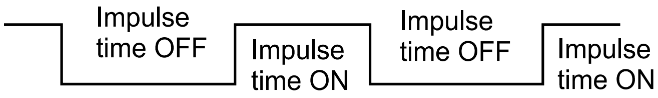
Min. runtime: The ON signal at the trigger input immediately switches the output on. If the input drops during the timer window (OFF), the output remains switched on nonetheless until the timer window has expired. If the trigger input has the status ON after the impulse time has elapsed, then the output stays switched on.



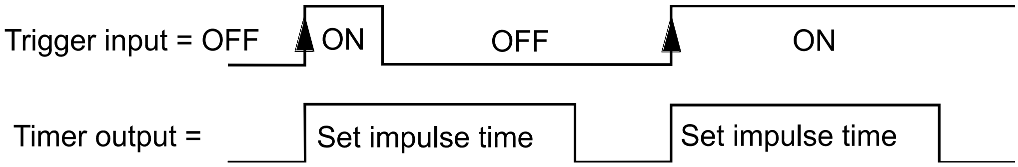
Blockage: The ON signal at the trigger input only switches the output on, after the timer window has expired since the last ON signal.



Astable: By indicating a switch-on and off time, a pulse generator is created without a trigger input. If the pulse-duty factor is also used for controls, it changes the switch-on time. The setting switch-off time = 0 is a special case: The switch-on time then corresponds to the entire period, and the pulse-duty factor to the relationship between the switch-on and the switch-off time
Example: A duty factor of 30% results in 30% ON and 70% OFF for the entered switch-on time.

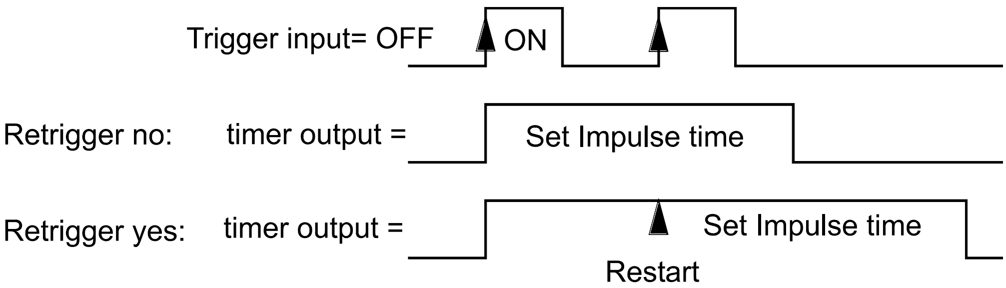


Impulse: If the selected trigger slope occurs, the output for the timer time switches on. A trigger input status change during the impulse time does not change the output status.



The trigger slope is positive if the input status changes from "OFF" to "ON" or from "switch open" to "switch closed" (= closing). A change from closed to open (= opening) is a negative trigger slope. TRIGGER SLOPE = *pos/neg* starts the timer anytime there is a status change at the input.

The properties of **retriggering** based on the example of a positive trigger slope:



Synchronisation

Simple description of the function:
This module provides an output variable relative to the date and time based on the date and time information of the device. In this way, periodic signals are available that directly relate to the time, day of the week, date, or season and allow for date or time-specific releases to control other function modules.

Input variables:	Output variables:
Enable synchronisation	Status time condition met, indication of the output Status summer time ON/OFF Status controller start = Starting up of the control unit

Special features:
- The function allows up to five date or time windows. The number has to be set after calling the module.
- Via the command “MODE”: periodic time windows in intervals of hours up to one year can be programmed.
- The setting "cyclical/once" determines whether the window whose parameters are set is to occur once or repeatedly (cyclical).
- The output "Contr. Start" only generates a 30 second pulse when the device is switched on or reset.

Example:
Assuming that a damp basement room is to be periodically heated, a time laps is prepared for other modules that then handle the heating. This procedure is to take place four times a year in the summer when there is enough solar energy in the buffer cylinder.

Entire menu view:

```
DES:   SYNC.
INPUT VARIABLE:
OUTPUT VARIABLE:

MODE:      Year
       cyclical

Day Mon     Day Mon
15.  06. -  17. 06.

05.  07.  -  07. 07.
25.  07.  -  27. 07.
10.  08.  -  12. 08.
```

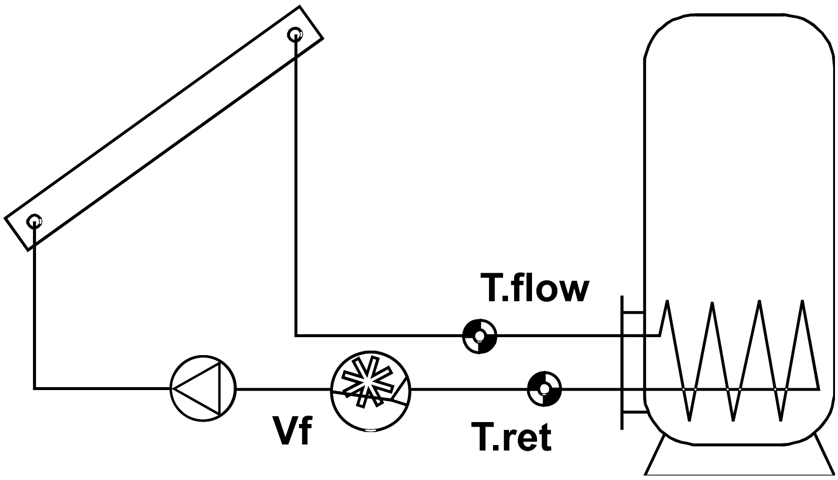
Procedure within one calendar year repeating every year

Output variable ON from the 15th 00:00 hours to the 17th of June 00:00 hours, etc.

N.B.: In modes "Year" and "Month" the time window begins and ends respectively with 00:00 hours for the specified days.
In modes “Hour” and “Day” the time window begins and ends respectively with the start of the specified minute.

Heat meter

Basic diagram:



Input variables:	**Output variables:**
Enable heat meter **Flow temperature** = T.flow **Return temperature** = T.ret Volume flow = Vf Meter reset	Current output Kilowatt hours Megawatt hours

Simple description of the function:

Calculation of the heat output and quantity via the temperature difference and volume flow with consideration of the share of frost protection in the heat medium.

Use as an electrical energy meter:

1. The sources of the input variables flow temperature and return temperature are set to *User / unused*.
2. The impulses of the electrical meter are captured as input 15 or 16 (Setting: Type: Impulse, Measurement variable: Flow rate). In this case the quotient setting does not correspond to litre/impulse, rather Wh/Impulse. This input must be defined as the input variable "*Flow rate*".
3. If the setting range (Wh/Impulse) of the input is insufficient, then this can be increased in the function menu by a factor (between 1 and 100).

For each impulse, the heat meter is increased by the Quotient * Factor (Wh).

Special features:

- When the temperature difference is calculated, the tolerance of the sensors and the measuring device may lead to large errors (if the difference is 10°K, the error is approximately 30%). To compensate for these errors, the device has a patented **calibration method** that can be found in the service menu.
- The collector sensor can also be used as the flow sensor. However, it always has to be installed on the flow outlet of the collector bar using an immersion sleeve. The heat measured then also contains the losses from the solar flow line.
- Meter reset function in the input variables and in the service menu.
- The output variables (output, MWh, kWh) can be taken up by other modules and input variables.
- A fixed value can be set as the flow rate instead of the transducer if *User* is in the input variable "volume flow."

WARNING: Although the meter of the function module Heat meter is recorded **every 6 hours** in internal memory, it is lost when loading new function data (load factory settings, load backup copy, data transfer from the C.M.I.)! Therefore for this reason, should a power failure occur, up to 6 hours meter data can be lost.

Heat meter

Calibration mode

By simultaneously measuring the same temperature with both sensors, the deviation of the sensors from each other can be calculated and included as a correction factor when calculating the heat amount in the future.
The calibration only has an influence on the sensor values in the function "Heat meter" and is not considered in other functions.
During calibration, it is important that both temperatures (flow and return) measure the same temperatures. Here, both of the sensor tips are tied together using tape or a wire. In addition, the two sensors should already be equipped with the line expansions. When using the collector sensor, the required line length has to be estimated approximately and integrated. The sensors must be connected to both parameterized inputs of the flow and return lines and are **both** submerged in a hot water bath (i.e. both measure the same temperature).

Overall submenu view - SERVICE MENU:

RESET	Reset the heat meter
METER: no	
HEAT AMOUNT:	
123.4 kWh	Total heat amount in kWh
CALIBRATION	
START: no	Start command for calibration
Status: UNCALIBRATED	The heat meter has not been calibrated
DIFFEREN. 0.56 K	Display of the difference measured during the calibration process

Calibration process:

1. Submerge the sensors in the water bath.
2. Start the calibration process by selecting "START *yes*"
3. After successful calibration, the status "CALIBRATED" appears. The measured difference value is displayed.

If the calibration was carried out incorrectly or erroneously, the result can only be corrected by carrying out a new calibration.

Entire menu view:

DES: HQC.1	
INPUT VARIABLE:	
SERVICE MENU:	
Status: CALIBRATED	
FROST PROT: 45 %	Indication of frost protection share in %
T.flow: 62.4 °C	The flow temperature is 62.4°C
T.rtn: 53.1 °C	The return temperature is 53.1°C
DIFF: 9.3 K	The difference between the flow and the return is 9.3°K
FLOW RT: 372 l/h	The current flow rate is 372 l/h
OUTPUT: 3.82 kW	The current output is 3.82 kW
HEAT AMOUNT:	
19 834.6 kWh	The current heat amount is 19,834.6 kWh

Meter

Simple description of the function:
As a operating hours meter or pulse meter (such as for the burner demand) this function represents another service function.

Input variables:	Output variables:
Enable meter (Max. 6 digital) Input variables Meter reset	Meter reading

Special features:
- When the meter function is entered in the function list, the number of "input variables" must be given. This can be changed later via "CHANGE FUNCTION." Both sensor inputs and other functions or output can be considered functions involved.
- In MODE *HRS RUN MTR (*hours run meter), the meter runs if **at least** one function involved is switched on. Only whole minutes are counted.
- In MODE *IMPULSE CTR,* as long as the status of one variable is “ON” with multiple input variables, the impulses of the other input variables are ignored. In addition, a divider can be indicated. If this divider is set to, say, 2, only every second pulse at the input variables raises the meter by one. The meter can read impulses up to a **maximum frequency of 1 Hz** (=1 pulse per second). The **minimum pulse duration** across inputs 1 to 14 is 500 ms, across inputs 15 and 16, 50 ms.
- The meter reading can be reset using input variables or the service menu.
- The output variable "Meter reading", which is not visible, can be taken over by other modules as an input variable.

Entire menu view:

```
DES: METER
INPUT VARIABLE:
SERVICE MENU:

MODE: HRS RUN MTR

Operating time:
    324 hr   18 min

Day meter prev day:
      4 hr   37 min
```

WARNING: Although the meter reading of the function module “Meter” is recorded **every 6 hours** in internal memory, it is lost when loading new function data (load factory settings, load backup copy, data transfer from the C.M.I.)! Therefore for this reason, should a power failure occur, up to 6 hours meter data can be lost.

Maintenance FN

This function is intended as a service function for the chimney sweeper and/or as a simple burner switch for the exhaust measurement. The preset output (generally 100%) is switched on for the set time after the burner has been started. In addition, the heating circuits set in the input variables are activated **with the maximum admissible flow temperature T.flow MAX**. The value of the output variable T.flow SET of this heating circuit is displayed as 5°C during the active maintenance function.

These targets could also be reached via the manual mode (switch corresponding outputs to MANUAL/ON). Assuming that the user does not have a manual for the controller or that the person does not have enough time to read the entire manual, this function should make things easier.
Use the input variable "**EXTERNAL SWITCH**" to activate the maintenance function via a specially installed switch or via a switch output of another function without any inputs at the controller being activated. An "External switch" must be set to "ON" for the duration of the maintenance function (**no runtime limit**). The function must be deactivated again via this switch.

Input variables:	**Output variables:**
External switch **involved functions** = Indication of the heating circuits	Status burner demand, indication of the output Burner output, indication of the speed control output

Entire menu view:

`FUNCTION START`	Press the scroll wheel to activate the burner and the heating circuits => number `FUNCTION STOP`
`--------------------`	
`DES:   FL G INSP`	
`Status:     OFF`	The function is deactivated (stopped)
`Runtime:      0 Min`	Remaining burner runtime
`INPUT VARIABLE:`	
`OUTPUT VARIABLE:`	
`Ttl runtime:   20 Min`	Total burner runtime after function start
`Burner outp.:    100%`	Desired burner output during service time

The function block provides the burner output as an output variable. It can be assigned to a speed output or an analogue output. For instance, the burner performance can be controlled (assuming you have the appropriate burner technology) via analogue output 15 or 16 (analogue output 0 - 10V).
The output of the burner performance is dominant in the maintenance function. In other words, no other analogue signal is allowed during maintenance (such as from DHW demand). However, digital signals can overwrite the analogue value at any time.
Once the burner demand has been switched off (function stopped), the heating circuits involved still remain active for three minutes to take the residual heat from the boiler. If the mixer behaviour is detected as "close" in the heating circuit, then the mixer is set to "closed" for 20 minutes (= maximum remaining runtime) and the heating circuit pump is switched off. Only then does the heating circuit return to the set operating mode.

Function control

In the solar and heating section, a number of functions perform important tasks that could lead to wrong reactions if there's a malfunction. For instance, if a defective cylinder sensor in the solar system detects temperatures that are too low, the solar system will run under false premises and take heat out of the cylinder. The module FUNCTION CONTROL can be used to monitor various operating modes and will issue an error message if there's a malfunction or block the inoperable function via its enable.

Input variables:	Output variables:
Control value a Control value b Enable differential control	Status error value, indication of the output Status error difference, definition of the output

Simple description of the function:
This function allows for two sensors (control value a, b) to be monitored in order to detect a short-circuit, an interruption and the maximum admissible temperature difference. Likewise, it is possible to monitor a sensor or temperature via a defined threshold value.

Special features:
- If there is interruption or short-circuit that affects the basic function of the module, an error message is issued only after 30 seconds.
- In addition, a temperature threshold or difference can be monitored using “ENABLE DIFF CTRL” If this control has been enabled, then the following applies:
- If sensors are assigned to both control values, the monitoring of the differences is active.
- If control value b is set to *User*, it is an adjustable temperature threshold that applies for the control value a as a limit value to be monitored.
- If the monitoring of the difference is not enabled, the message DIFFEREN. OK appears in the error display nevertheless. It generally suffices to monitor the circulation of just one circuit in solar warm water systems with multiple consumers (via enable). If another circuit is running, the message of monitoring should still be displayed.
- If only one sensor is monitored (control value b = *User*) or if the difference is monitored, a malfunction is only messaged after an error time that can be set. This helps to prevent unjustified error messages caused by temperature peaks when the system is starting.
- The parameters are found in their own parameter menu to provide an overview of the error evaluation at all times.
- Via the command “Save error: yes", the display **ERROR** remains until it is manually deleted even after the error has disappeared.

Warning:
Sometimes it makes sense to link one of the output variables directly to the control output to create a 0 to 10 V or PWM signal. Linking this function is only allowed using control output A15, not with output A16.

Function control

Entire menu view:

(No error)

```
DES: FUNC CTRL
INPUT VARIABLE:
OUTPUT VARIABLE:
PARAMETER:

T.collector     OK
57.4  °C
T.DHW1          OK
48.9  °C
DIFFEREN.       OK
 8.5  K

Save error:    yes

Delete error disp.?
```

(With error)

```
BEZ.: KONTR.SOL1
INPUT VARIABLE:
OUTPUT VARIABLE:
PARAMETER:

T.Collector     ERROR
9999  °C     lead brk
T.DHW1          OK
48.9  °C
DIFFEREN.       ERROR
9999  K       too high

Save error:    yes

Delete error disp.?
```

The parameter menu contains the following if monitoring a difference:

```
Error if over
at least      30 min     Error minimum time setting
CVa – CVb  >   50  K     Difference threshold setting
```

Or if value a is being monitored, for example:

```
Error if over
at least      30 Min     Error minimum time setting
CVa        >    30°C     Error threshold setting
```

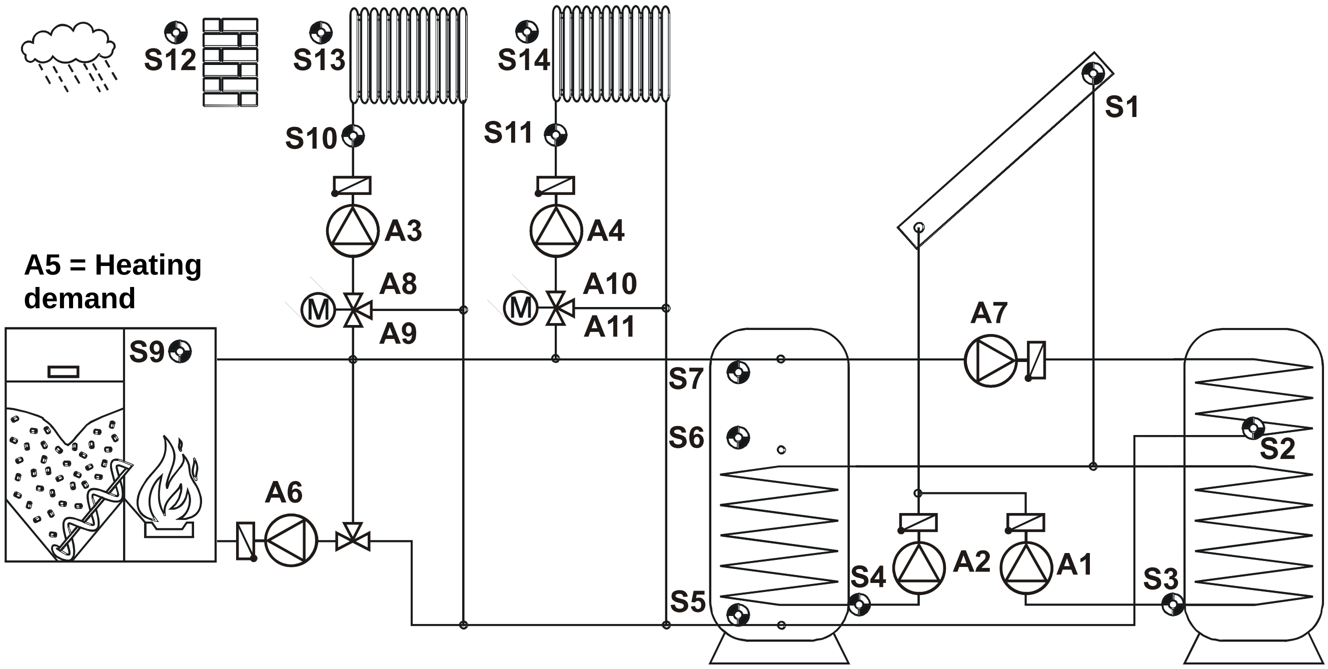
Troubleshooting:

"Save error: yes" **ERROR** remains displayed even after the remedy has been provided until the user presses the scroll wheel to confirm "Delete error display?" If the error continues after deletion, the message will be reissued after a certain delay.
"Save error: no" **ERROR** is automatically deleted once the error has disappeared.
If an output has been assigned in the output variables, it responds the same as the display.
The status lines of the function control should also be entered in the function overview using the user interface editor. In this way, users will always have the information they need in their menu.

Factory setting

TA_FACTORY SETTING - The function data with this specification have been transferred to the control unit. **The TA factory setting can be loaded by simultaneously pressing the two input keys and scroll wheel when starting up the controller.**

The factory settings are based on the following hydraulic diagram for solar warm water system with a buffer and DHW cylinder, a boiler fired with wood pellets or oil/gas, and two heating circuits:



A detailed description of the programming can be found on our homepage www.ta.co.at.

Technische Alternative RT GmbH

A-3872 Amaliendorf Langestraße 124
Tel ++43 (0)2862 53635 Fax ++43 (0)2862 53635 7
E-Mail: mail@ta.co.at --- www.ta.co.at ---