Library
BuildingLib

TXV 003 47.01

Library BuildingLib

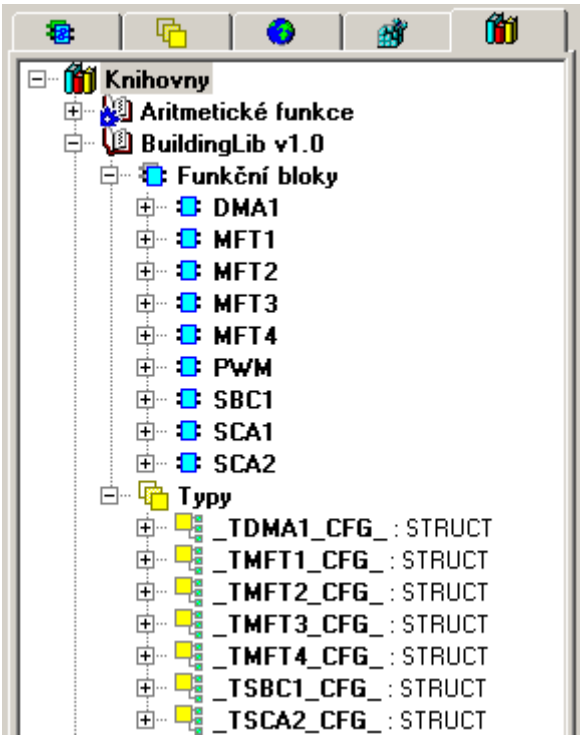
1. edition – August 2008

CONTENT

1. INTRODUCTION

Building.lib library is a library for the Mosaic programmable environment. By its content it is determined for the support of programming aplications, so called builiding management.

The following picture shows the structure of the BuildingLib library in the Mosaic environment.



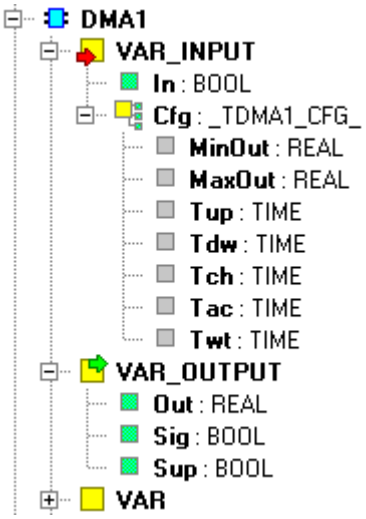
Pic. 1.1 BuildingLib library

2. ONE-BUTTON DIMMER DMA1

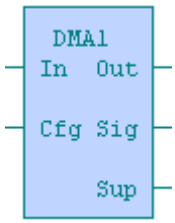
The function block of the DMA1 dimmer is used for gradual control of bulb brightness with the possibility of gradual start/finish after switch on/off of the dimmer. The dimmer is controlled by a button on the input *In*. Short press (<0.5s) cause that the output *Out* starts gradually to the required level of brightness, next short press causes that the output *Out* is switched off. Long press (>0.5s) causes the gradual regulation of brightness within the value range *MinOut* <-> *MaxOut.* After the outer values are reached, the regulation of brightness is ceased. When the button is released, the brightness intensity is remembered and other short presses switch on/off the output Out to this intensity.

The start speed of the output Out to the required brightness when switched on is set by the parameter *Tup*, the speed of dimming when switched off is set by the parameter *Tdw* (if the start/finish is not required, parameters *Tup/Tdw* are set to 0). The speed of the brightness change (overrun) from the value *MinOut* <-> *MaxOut* is set by the parameter *Tch*. The period of shine can be set after which the output is automatically switched off (parameter Tac), eventually delayed switched off of the output, so called outbound delay (parameter Twt). If parameters *Tch/ Twt* are set to 0, the function is not activated.

Output *Sig* is designated for dimmer run indication (e.g. for control of the indicator on the control button). Output *Sup* is designated for the control of lighting ballast supply and is automatically set to the log.1 in case that the value on the output Out is higher than 10% of the value *MaxOut*.



Obr. 2.1 FB DMA1structure



Obr. 2.2 FB DMA1appearance

Variables description :

Name		Signification	Type	Format
In		Control button	input	bool
Cfg		Block structure configuration		_TDMA1_CFG_
	.MinOut	Output minimum [%]		real
	.MaxOut	Output maximum [%]		real
	.Tup	Switch on start time		time
	.Tdw	Switch off finish time		time
	.Tch	Dimmer overrun time during brightness change		time
	.Tac	Shine time		time
	.Twt	Delayed switch off time (outbound delay)		time
Out		Dimmer output [%]		real

Sig	Dimmer run indication	output	bool
Sup	Lighting ballasts control		bool

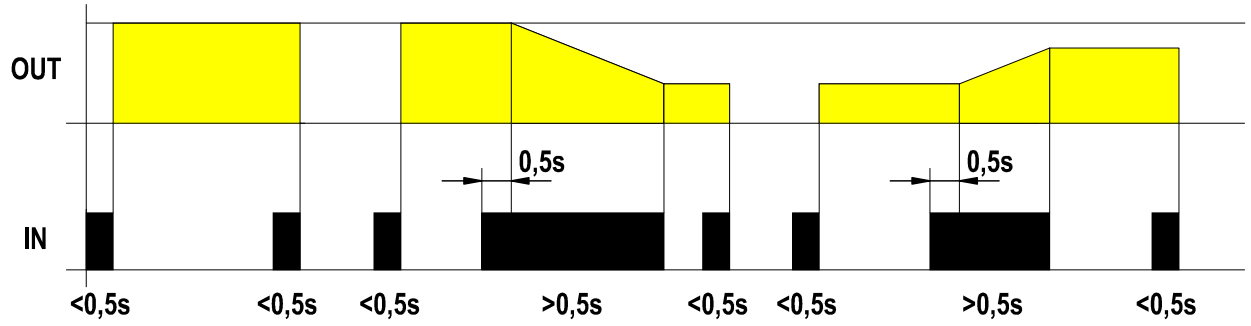


Fig. 2.3 Dimmer function when start/finish is not activated

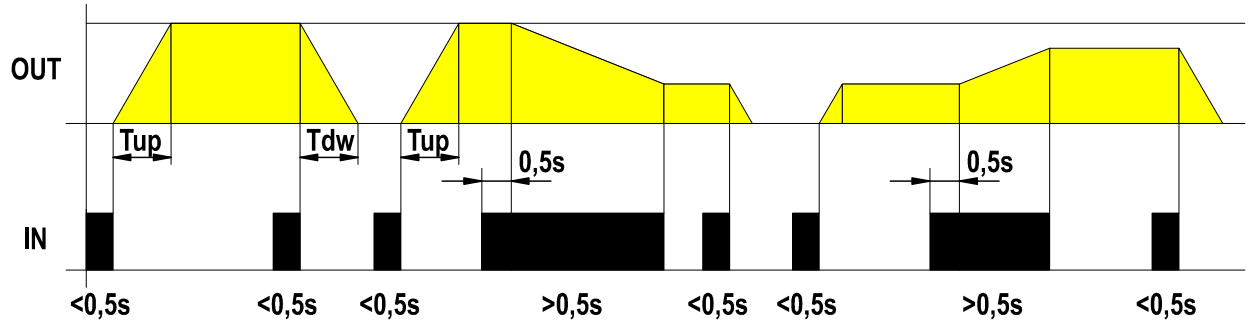


Fig. 2.4 Dimmer function when start/finish is activated

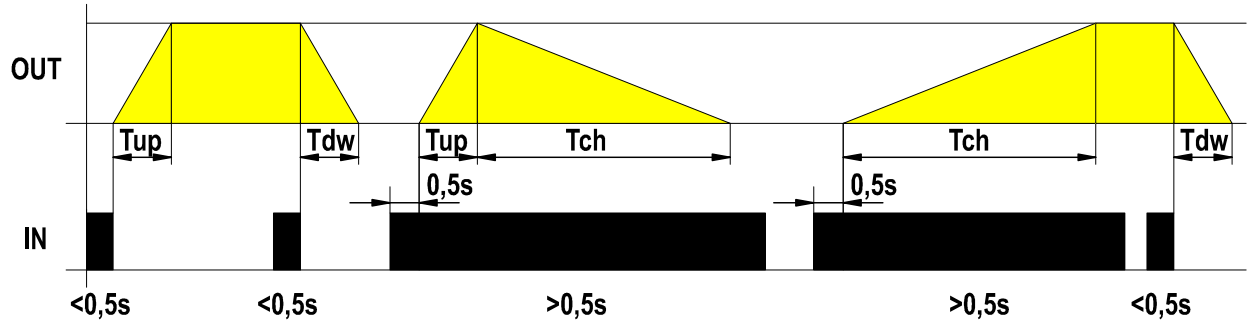


Fig. 2.5 Timing of start, finish and overrun of the dimmer

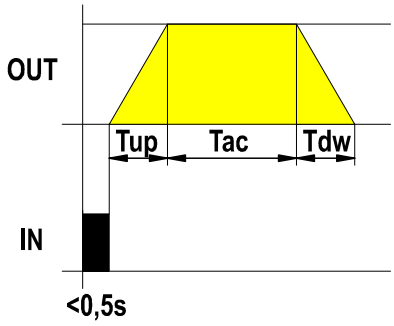


Fig. 2.6 Shine period timing

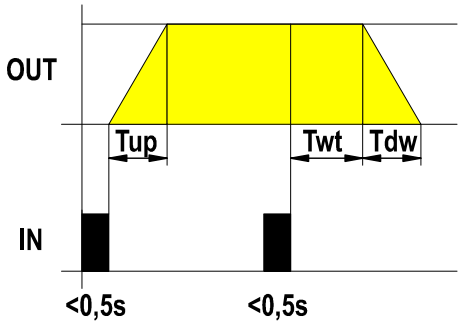


Fig. 2.7 Outbound delay timing

Example of the program with the call of the funciton block **DMA1** in ST language :

```
PROGRAM Example_DMA1
  VAR
    input : bool;
    DMA1_Cfg : _TDMA1_CFG_ := (MinOut:= 0.0,
                               MaxOut:= 100.0,
                               Tup:= T#200ms,
                               Tdw:= T#200ms,
                               Tch:= T#2s);
    inst_DMA1 : DMA1;           //instance FB DMA1
    output : real;
    signal : bool;
    support: bool;
  END_VAR

  inst_DMA1 (In := input, Cfg := DMA1_Cfg, Out => output, Sig => signal,
             Sup => support);
END_PROGRAM
```

3. LANDING AUTOMATIC MACHINE WITH THE SWITCH OFF SIGNALISATION SCA1

The funcion block is used for the control of the landing illumination (delayed switch off). Interval of luminance time is set by the parameter *Tac.* It contains a warning before forthcoming switch off proved by the double blink 40s and 30s prior to the switch off. The block can operate in one of three possible modes (parameter *Fce*) :

- (0) AUTO – timing without the possibility of time prolongation
- (1) PROG – timing with the possibility of time prolongation by the number of button presses
- (2) ON – output is clutched permanently (service mode, maintance, …)

In AUTO and PROG modes, it is possible to cease the timing by the long button press (>2s). This function is also used as a protection of the permanent allumination when the pressed button is blocked. In the AUTO mode, the clutch time of the output can be prolonged only after the signalization of the forthcoming output switch off (40s prior to the output switch off).

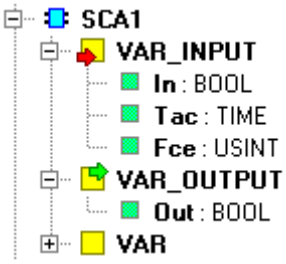


Fig. 3.1 FB SCA1 structure

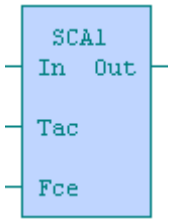
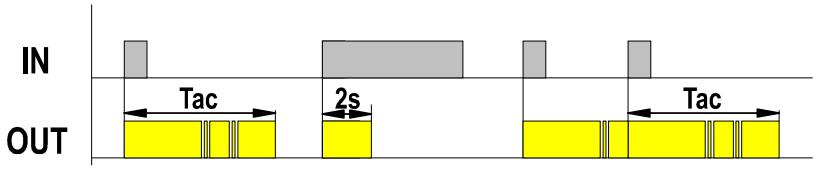
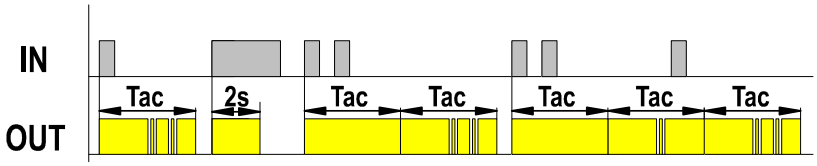
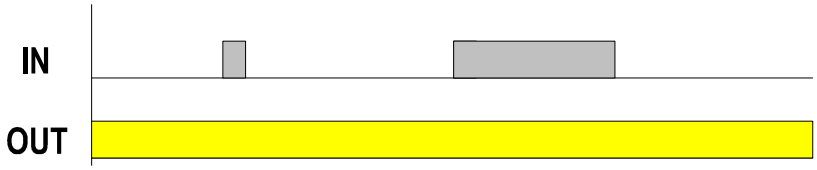


Fig. 3.2 FB SCA1 appearance

Variable description :

Name	Signification	Type	Format
In	Control button	input	bool
Tac	Luminance time		time
Fce	Landing automatic machine mode		usint
Out	Output contact	output	bool

Individual block time functions are shown in the following pictures :

FCE	Description	Graph
0	AUTO – timing without the possibility of time prolongation	
1	PROG - timing with the possibility of time prolongation by the number of button presses	
2	ON – output is clutched permanently	

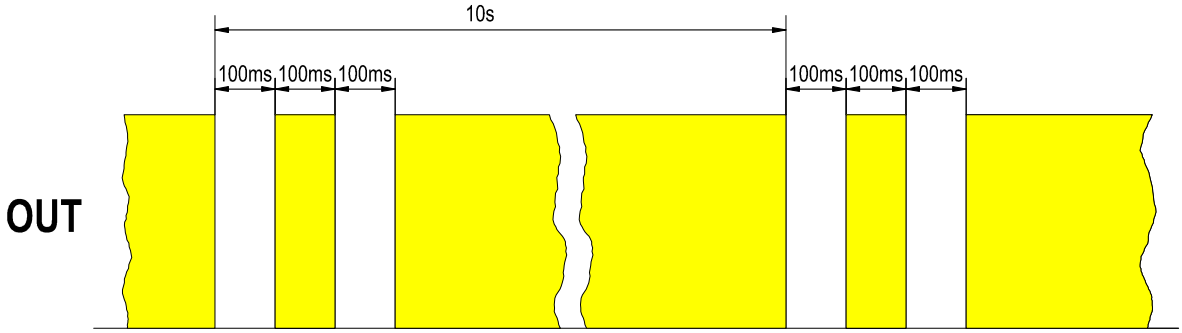


Fig. 3.3 Time proportions of the warning blink

In case that the output switch on time is set to the period shorter than 30s, the warning blink is not activated.

Example of the program with the call of the function block **SCA1** in ST language:

```
PROGRAM Example_SCA1
  VAR
    input : bool;
    inst_SCA1 : SCA1;           //instance FB SCA1
    output : bool;
  END_VAR

  inst_SCA1(In := input, Tac := T#3m, Fce := 0, Out => output);
END_PROGRAM
```

4. LANDING AUTOMATIC MACHINE WITH DIMMING SCA2

The fundcion block is used for inteligent control of the landing illumination with the function of the gradual switch on (entering ramp) and dimming (finishing ramp). The button In1 or the switch In2 can be used as a control input. The start time is set by the parameter *Tup,* the luminance period by the parameter *Tac* and finish time by the parameter *Tdw*. Output *Out* can operate within the range *MinOut, MaxOut* while the landing automatic machine is running.

Output *Sig* is designated for dimmer run indication. Output *Sup* is designated for the control of lighting ballasts supply and is automatically set to log.1 in case that on the output Out the value is higher than 10% of the *MaxOut* value.

Currently running luminance cycle can be prolonged by repeated button press. The control via the remote control is functionally superordinated and can be with the advantage used for service purposes (maintance, control and exchange of the lighting members,…).

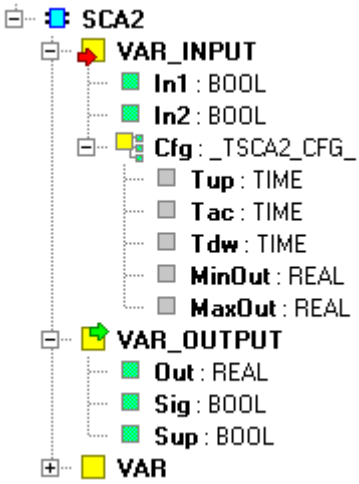


Fig. 4.1 FB SCA2 structure

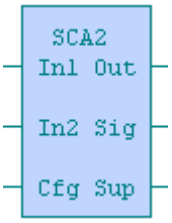


Fig. 4.2 FB SCA2 appearance

Variable description :

Name		Signification	Type	Format
In1		Control button	input	bool
In2		Control switch		bool
Cfg		Block configuration structure		_TSCA2_CFG_
	.Tup	Start time		time
	.Tac	Illuminance time		time
	.Tdw	Finish time		time
	.MinOut	Output minimum		real
	.MaxOut	Output maximum		real
Out		Dimmer output	output	real
Sig		Run indication		bool
Sup		Ballasts control		bool

Block operation illustrate following pictures.

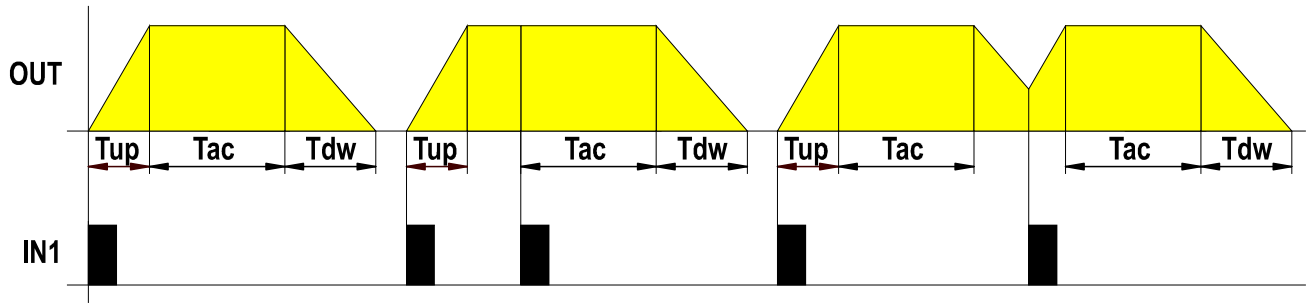


Fig. 4.3 Block function when operated by the In1 button

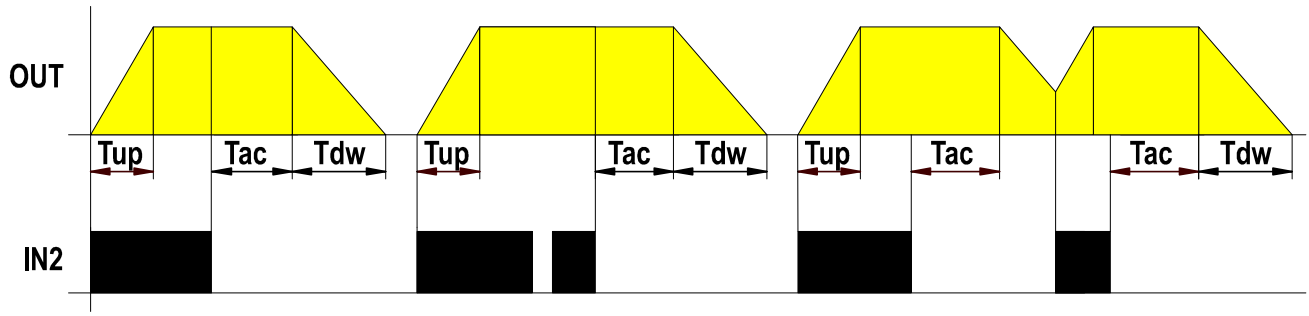


Fig. 4.4 Block function when operated by In2 button

Example of the program with the call of the function block **SCA2** in ST language :

```
PROGRAM Example_SCA2
  VAR
    input1 : bool;
    input2 : bool;
    SCA2_Cfg : _TSCA2_CFG_ := (Tup:= T#200ms,
                               Tac:= T#3m,
                               Tdw:= T#200ms,
                               MinOut:= 0.0,
                               MaxOut:= 100.0);
    inst_SCA2 : SCA2;           //instance FB SCA2
    output : real;
    signal : bool;
    support: bool;
  END_VAR

  inst_SCA2 (In1 := input1, In2 := input2, Cfg := SCA2_Cfg, Out => output,
             Sig => signal, Sup => support);
END_PROGRAM
```

5. WINDOW BLINDS CONTROLLER SBC1

The function block is used for control of the window blinds supply. The control is realized via two buttons. The block ensures the time control of the supply run period and the prolongation during the supply run reservation (supply protection due to electrical and mechanical causes).

The output for the movement of blinds up *Up* is activated after the press of the button Up. Another press of any button causes the deactivation of this output. The output for the movement of blinds down *Down* is activated after the press of the button Down. Another press of any button causes the deactivation of this output.

When outputs are activated for longer period than *Tac* , outputs are deactivated automatically. During the drive reversing of operation, there is always interposed a time prolongation Twt between individual outputs activations.

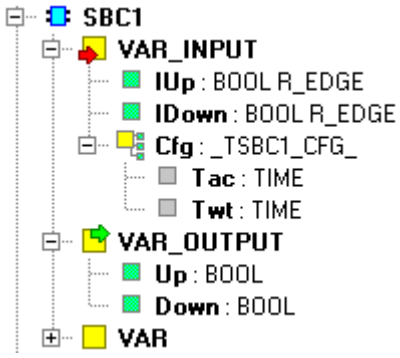


Fig. 5.1 FB SBC1 structure

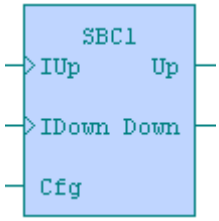


Fig. 5.2 FB SBC1 appearance

Variable description :

Name	Signification	Type	Format
IUp	Control button up	input	bool
IDown	Controll button down		bool
Cfg	Block configuration structure		_TSBC1_CFG_
.Tac	Max time of drive run		time
.Twt	Drive reversing of operation prolongation		time
Up	Output for drive run up	output	bool
Down	Output for drive run down		bool

Example of the program with the call of the function block **SBC1** in ST language :

```
PROGRAM Example_SBC1
  VAR
    input1 : bool;
    input2 : bool;
    SBC1_Cfg : _TSBC1_CFG_ := (Tac:= T#15s, Twt:= T#500ms);
    inst_SBC1 : SBC1;           //instance FB SBC1
    output1 : bool;
    output2 : bool;
  END_VAR

  inst_SBC1(IUp := input1, IDown := input2 , Cfg := SBC1_Cfg, Up => output1,
            Down => output2);
END_PROGRAM
```

6. SINGLE-LEVEL THERMOSTAT MFT1

The block deal with the algorithm of the classic single-level thermostat. Output contact *Out* is closed until the measured temperature In reached the set temperature Rq, then it is switched off. Adjustable hysteresis Hyst and time prolongations *Tup/Tdw* prevent from often switching of the output *Out*.

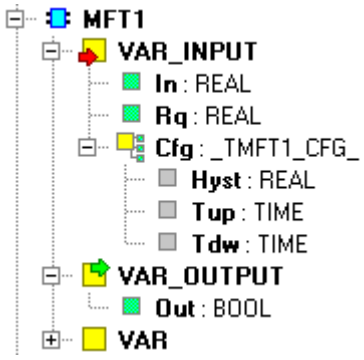


Fig. 6.1 FB MFT1 structure

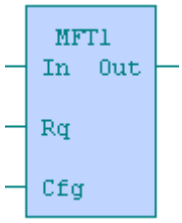


Fig. 6.2 FB MFT1 appearance

Variable description:

Name		Signification	Type	Format
In		Temperature measured	input	real
Rq		Temperature required		real
Cfg		Block configuration structure		_TMFT1_CFG_
	.Hyst	hysteresis		real
	.Tup	Switch on delay		time
	.Tdw	Switch off delay		time
Out		Thermostat output	output	bool

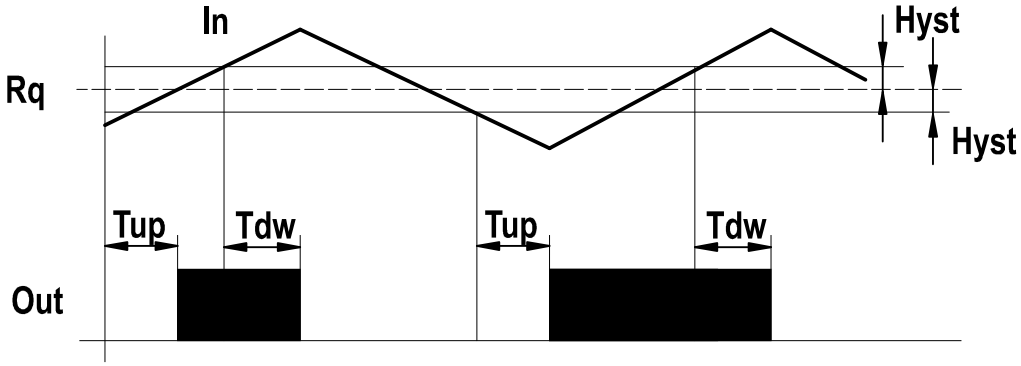


Fig. 6.3 MFT1 block function

Example of the program with the call of the function block **MFT1** in ST language :

```
PROGRAM Example_MFT1
  VAR
    input   : real;
    request : real := 21.0;
    MFT1_Cfg : _TMFT1_CFG_ := (Hyst:= 0.1,
                               Tup:= T#5s,
                               Tdw:= T#5s);
    inst_MFT1 : MFT1;           //instance FB MFT1
    output : bool;
  END_VAR

  inst_MFT1(In := input, Rq := request, Cfg := MFT1_Cfg, Out => output);
END_PROGRAM
```

7. TWO-LEVELLED THERMOSTAT MFT2

Within the range of set difference *Diff* operates the output *Out1* as a classic thermostat to the input *In* and required temperature *Rq*. However, if the temperature drops bellow the set difference *Diff*, the output *Out2* is switched on, too. Both outputs *Out* switch simultaneously when the required tepmerature *Rq* is reached. Hysteresis *Hyst1/Hyst2* and time delays *Tup/Tdw* are contemplated when outputs *Out* are closed and opened.

Characteristic example of this thermostat is e.g. the control of the two-levelled boiler, when the first level is controlled according to the set temperature and auxiliary second level is switched when the temperature drops bellow the set difference. This helps to the first level when the temperature drops significantly.

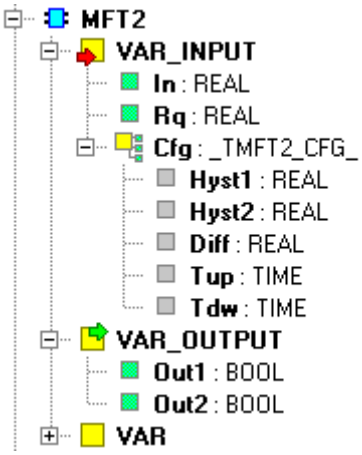


Fig. 7.1 FB MFT2 structure

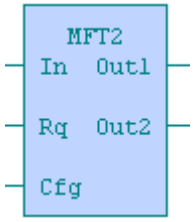


Fig. 7.2 FB MFT2 appearance

Variable description :

Name		Signification	Type	Format
In		Temperature measured	input	real
Rq		Temperature required		real
Cfg		Block configuration structure		_TMFT2_CFG_
	.Hyst1	Hysteresis to Out1		real
	.Hyst2	Hysteresis to Out2		real
	.Diff	Temperature difference		real
	.Tup	Switch on delay		time
	.Tdw	Switch off delay		time
Out1		1st grade output	output	bool
Out2		2nd grade output		bool

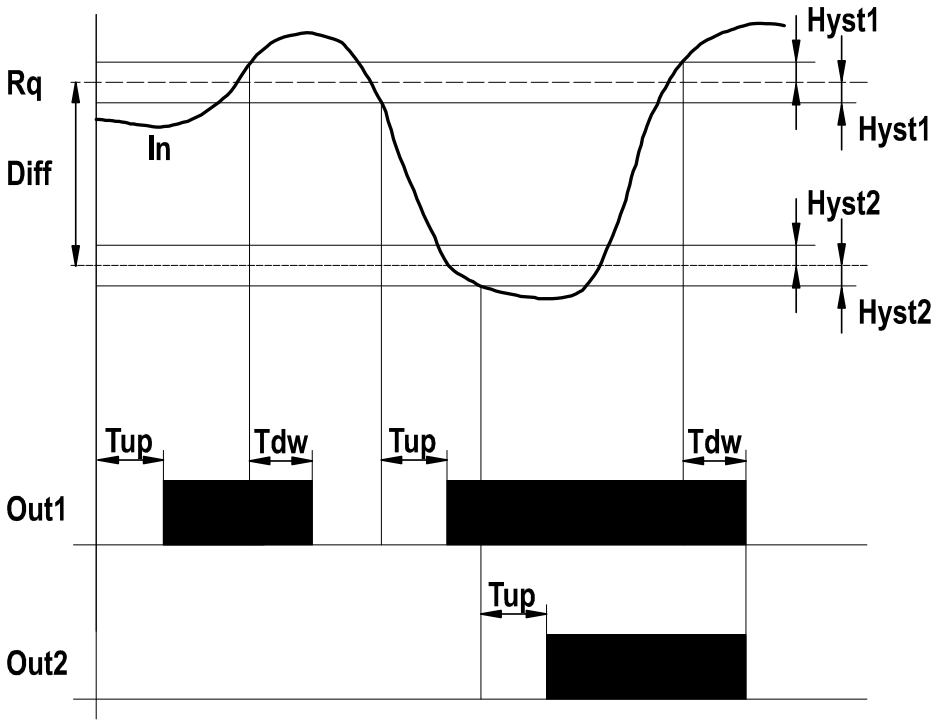


Fig. 7.3 MFT2 block function

Example of the program with the call of the function block **MFT2** in ST language:

```
PROGRAM Example_MFT2
  VAR
    input   : real;
    request : real := 21.0;
    MFT2_Cfg : _TMFT2_CFG_ := (Hyst1:= 0.1,
                               Hyst2:= 0.1,
                               Diff:= 1.0,
                               Tup:= T#20s,
                               Tdw:= T#20s);
    inst_MFT2 : MFT2;           //instance FB MFT2
    output1 : bool;
    output2 : bool;
  END_VAR

  inst_MFT2(In := input, Rq := request, Cfg := MFT2_Cfg, Out1 => output1,
            Out2 => output2);
END_PROGRAM
```

8. DIFFERENTIAL THERMOSTAT MFT3

The output *Outx* is always switched to the correspondent input *Inx,* the temperature of which is during the difference excession lower, within the differential thermostat. After the temperature of the correspondent hysteresis *Hystx* rise, the output is opened.

Differential thermostat is used for maintance of two similar temperatures, e.g. within heating systems (boiler – water tank), solar systems (collector – reservoir), water heating (water heater – water distribution) etc.

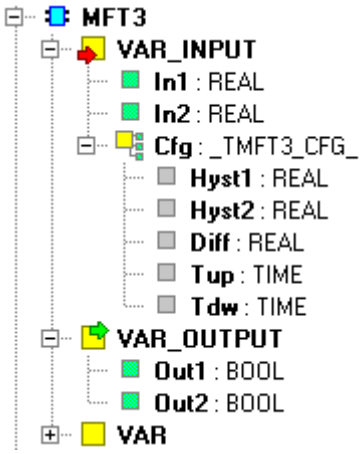


Fig. 8.1 FB MFT3 structure

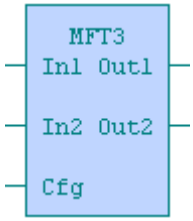


Fig. 8.2 FB MFT3 appearance

Variable description :

Name		Signification	Type	Format
In1		Measured temperature1		real
In2		Measured temperature 2		real
Cfg		Block configuration structure		_TMFT3_CFG_
	.Hyst1	Hysteresis respective to the temperature 1	input	real
	.Hyst2	Hysteresis respective to the temperature 2		real
	.Diff	Temperature difference		real
	.Tup	Switch on delay		time
	.Tdw	Switch off delay		time
Out1		Output respective to the temperature 1	output	bool
Out2		Output respective to the temperature 2		bool

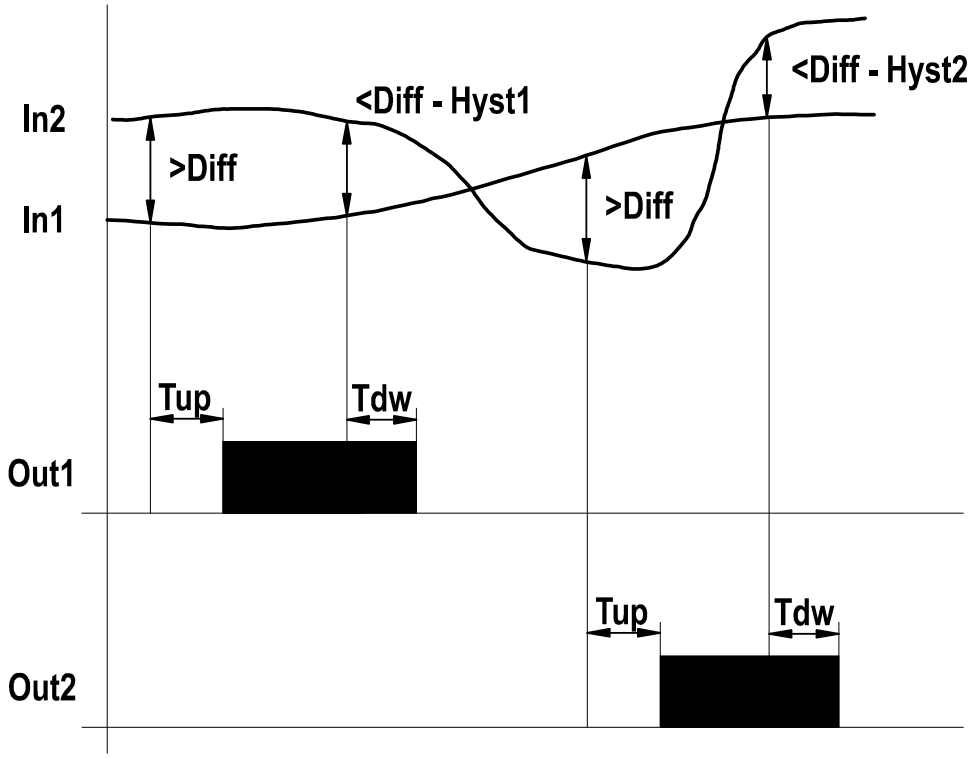


Fig. 8.3 MFT3 block function

Example of the program with the call of the function block **MFT3** in ST language :

```
PROGRAM Example_MFT3
  VAR
    input1   : real;
    input2   : real;
    MFT3_Cfg : _TMFT3_CFG_ := (Hyst1:= 0.1,
                               Hyst2:= 0.1,
                               Diff:= 1.0,
                               Tup:= T#20s,
                               Tdw:= T#20s);
    inst_MFT3 : MFT3;           //instance FB MFT3
    output1 : bool;
    output2 : bool;
  END_VAR

  inst_MFT3(In1 := input1, In2 := input2, Cfg := MFT3_Cfg, Out1 => output1,
            Out2 => output2);
END_PROGRAM
```

9. THERMOSTAT WITH THE DEAD ZONE MFT4

If the measured temperature *In1* is lower than the required temperature *Rq1*, the output contact switches the heating *Out1* on, when the temperature is higher than *Rq1*, it switches off. If the measured temperature *In1* is higher than the required temperature *Rq2*, the output contact switches the cooling *Out2* on, when the temperature is lower than *Rq2*, it switches off. During the switch on/off of outputs *Out*, hysteresis *Hyst1/Hyst2* are contemplated as well as time delays *Tup/Tdw*.

This function can be used e.g. for automatic warming and cooling of the air within the air-conditioning systems so, that the temperature of incoming air was always within the range of temperatures *Rq1* and *Rq2*.

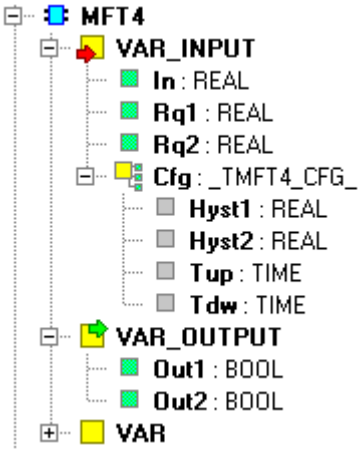


Fig. 9.1 FB MFT4 structure

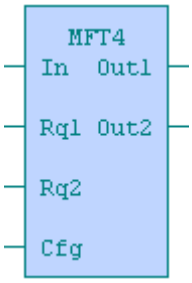


Fig. 9.2 FB MFT4 appearence

Variable description :

Name		Signification	Type	Format
In		Temperature measured	input	real
Rq1		Required temperature 1		real
Rq2		Required temperature 2		real
Cfg		Block configuration structure		_TMFT4_CFG_
	.Hyst1	Hysteresis respective to the temperature Rq1		real
	.Hyst2	Hysteresis respective to the temperature Rq2		real
	.Tup	Switch on delay		time
	.Tdw	Switch off delay		time
Out1		Heating output	output	bool
Out2		Cooling output		bool

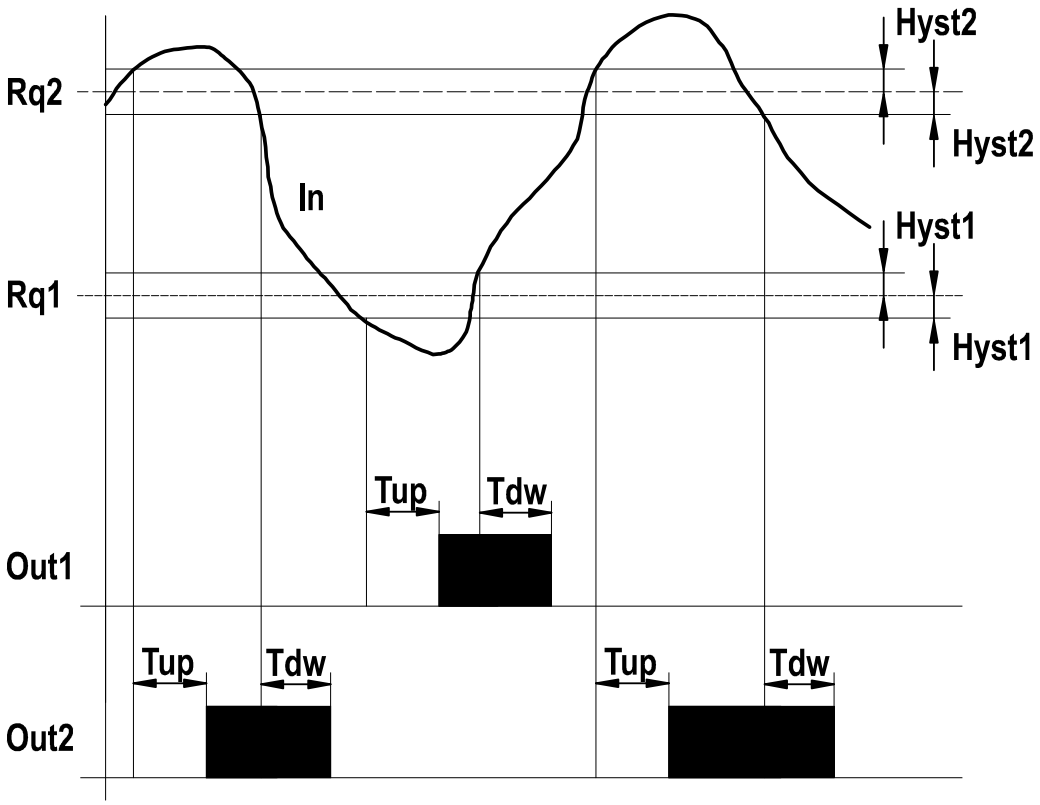


Fig. 9.3 MFT4 block function

Example of the program with the call of the function block **MFT4** in ST language:

```
PROGRAM Example_MFT4
  VAR
    input    : real;
    request1 : real := 20.0;
    request2 : real := 21.5;
    MFT4_Cfg : _TMFT4_CFG_ := (Hyst1:= 0.1,
                               Hyst2:= 0.1,
                               Tup:= T#20s,
                               Tdw:= T#20s);
    inst_MFT4 : MFT4;           //instance FB MFT4
    output1 : bool;
    output2 : bool;
  END_VAR

  inst_MFT4(In := input, Rq1 := request1, Rq2 := request2, Cfg := MFT4_Cfg,
            Out1 => output1, Out2 => output2);
END_PROGRAM
```

10. PULSE MODULATION DURATION PWM

The block is used for realization of the function of pulse modulation duration. The period of modulation is set by the value of the variable *Per*, the width of modulated pulse is set by the value of the variable *Pul*. The value *Pul* is entered within the range 0-100%. The actuation of the pulse modulation duration on the output *Out* is conditioned by the enabling variable *Enb* setup.

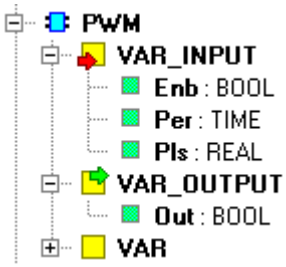


Fig. 10.1 FB PWM structure

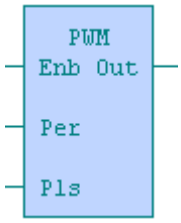


Fig . 10.2 FB PWM appearance

Variable description :

Name	Signification	Type	Format
Enb	Enabling of modulation	input	bool
Per	Period of modulation		time
Pls	Pulse width [0-100%]		real
Out	Modulation output	output	bool

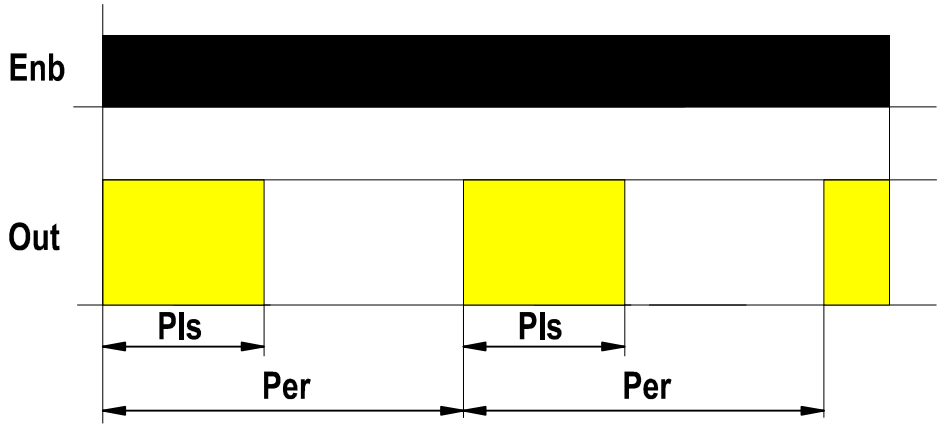


Fig. 10.3 PWM block function

Example of the program with the call of the function block **PWM** in ST language:

```
PROGRAM Example_PWM
  VAR
    enable : bool := true;
    period : time := T#5s;
    puls   : real := 20.0;
    inst_PWM : PWM;           //instance FB PWM
    output : bool;
  END_VAR

  inst_PWM(Enb := enable, Per := period, Pls := puls, Out => output);
END_PROGRAM
```

Notes :

teco

Objednávky a informace:
Teco a. s. Havlíčkova 260, 280 58 Kolín 4, tel. 321 737 611, fax 321 737 633

TXV 003 47.01