



PROGRAMOVATELNÉ AUTOMATY

Knihovna BuildingLib

TXV 003 47.01

Knihovna BuildingLib

2. vydání – srpen 2011

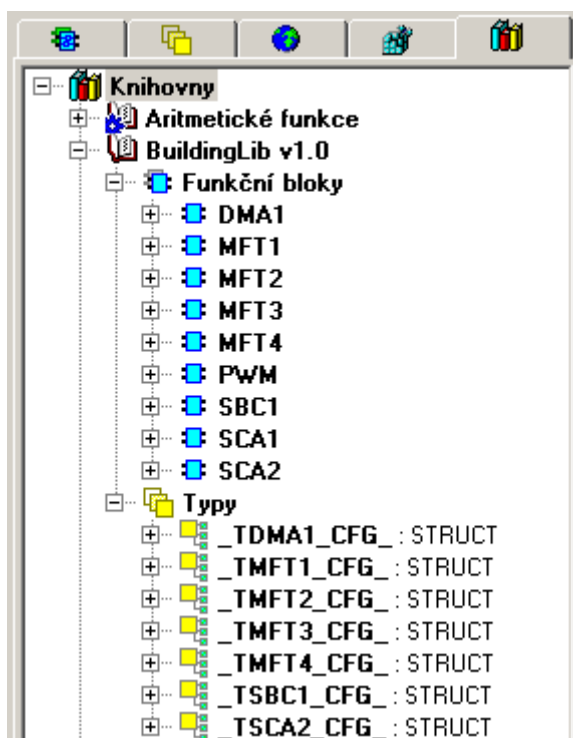
OBSAH

1.	ÚVOD.....	3
2.	JEDNOTLAČÍTKOVÝ STMÍVAČ DMA1	4
3.	SCHODIŠŤOVÝ AUTOMAT SE SIGNALIZACÍ PŘED VYPNUTÍM SCA1	7
4.	SCHODIŠŤOVÝ AUTOMAT SE STMÍVÁNÍM SCA2	9
5.	OVLADAČ ROLETOVÝCH ŽALUZÍ SBC1.....	11
6.	JEDNOÚROVŇOVÝ TERMOSTAT MFT1	12
7.	DVOJÚROVŇOVÝ TERMOSTAT MFT2.....	14
8.	DIFERENČNÍ TERMOSTAT MFT3	16
9.	TERMOSTAT S MRTVOU ZÓNOU MFT4	18
10.	PULSNĚ ŠÍŘKOVÁ MODULACE PWM.....	20

1. ÚVOD

Knihovna Building.lib je knihovnou pro programovací prostředí Mosaic. Svým obsahem je určena k podpoře programování aplikací tzv. building managementu.

Následující obrázek ukazuje strukturu knihovny BuildingLib v prostředí Mosaic.



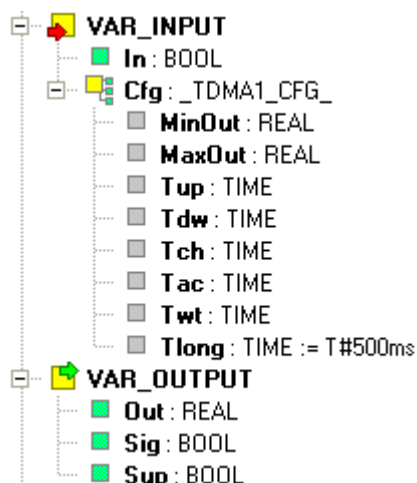
Obr. 1.1 Knihovna BuildingLib

2. JEDNOTLAČÍTKOVÝ STMÍVAČ DMA1

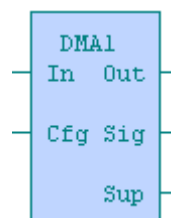
Funkční blok stmívače DMA1 slouží k plynulému ovládní jasu žárovek s možností postupného náběhu/doběhu po zapnutí/vypnutí stmívače. Stmívač je ovládán tlačítkem na vstupu *In*. Při krátkém stisku (<Tlong) výstup *Out* nabíhá na požadovanou úroveň jasu, dalším krátkým stiskem výstup *Out* vypíná. Dlouhým stiskem tlačítka (>Tlong) dochází k plynulé regulaci jasu v rozsahu hodnot *MinOut* <-> *MaxOut*. Po dosažení krajní hodnoty je regulace jasu zastavena. Po uvolnění tlačítka je intenzita jasu zapamatována a další krátké stisky tlačítka vypínají/zapínají výstup *Out* na tuto intenzitu.

Rychlost náběhu výstupu *Out* na požadovaný jas při zapnutí je dána parametrem *Tup*, rychlost sestupu při vypnutí je dána parametrem *Tdw* (pokud není náběh/doběh požadován, jsou parametry *Tup/Tdw* nastaveny na 0). Rychlost změny jasu stmívače (přeběhu) z hodnoty *MinOut* <-> *MaxOut* je dána parametrem *Tch*. Pro stmívač lze nastavit dobu svitu, po které dojde k automatickému vypnutí výstupu (parametr *Tac*), případně zpožděné zhasnutí výstupu, tzv. odchodové zpoždění (parametr *Twt*). Pokud jsou parametry *Tch/ Twt* nastaveny na 0, není funkce aktivována.

Výstup *Sig* je určen pro signalizaci chodu stmívače (např. pro ovládní signálky v ovládacím tlačítku). Výstup *Sup* je určen pro ovládní napájení předřadníků zářivek a je automaticky nastaven do log.1 v případě, že je na výstupu *Out* hodnota vyšší, než 10% hodnoty *MaxOut*.



Obr. 2.1 Struktura FB DMA1



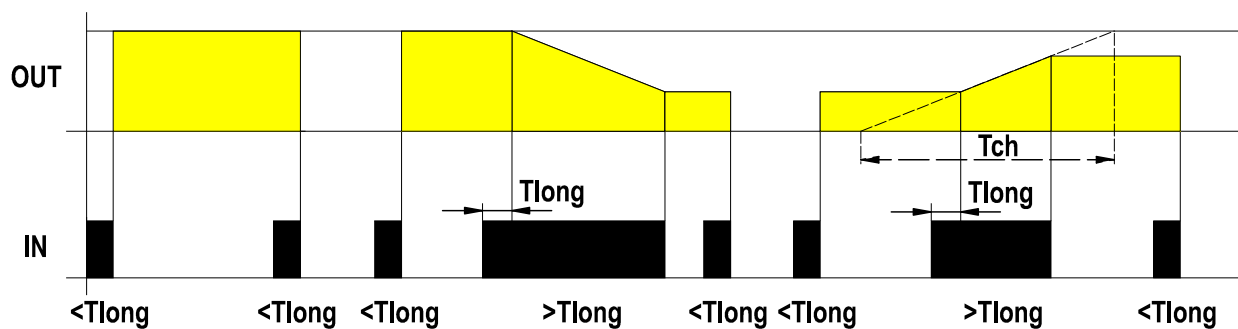
Obr. 2.2 Vzhled FB DMA1

Popis proměnných :

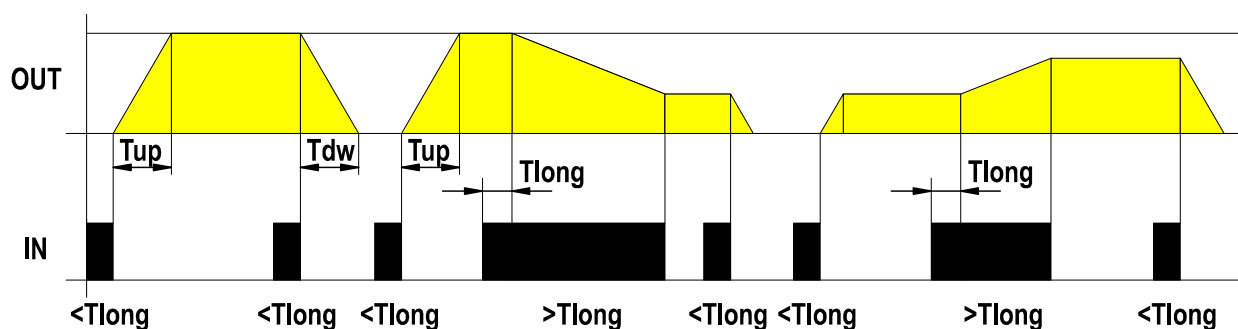
Název	Význam	Typ	Formát
In	ovládací tlačítko	Vstupní	bool
Cfg	konfigurační struktura bloku		_TDMA1_CFG_
.MinOut	minimum výstupu [%]		real
.MaxOut	maximum výstupu [%]		real
.Tup	doba náběhu při zapnutí		time
.Tdw	doba doběhu při vypnutí		time
.Tch	doba přeběhu stmívače při změně jasu		time
.Tac	doba svitu		time
.Twt	doba zpožděného vypnutí (odchodové zpožd.)		time
.Tlong ¹⁾	prodleva vyhodnocení dlouhého stisku		time (default 500ms)
Out	výstup stmívače [%]	Výstupní	real
Sig	signalizace chodu stmívače		bool
Sup	ovládní předřadníků		bool

1) Parametr je zařazen od verze 1.1 knihovny BuildingLib. Ve verzi 1.0 prodleva pevně 500ms.

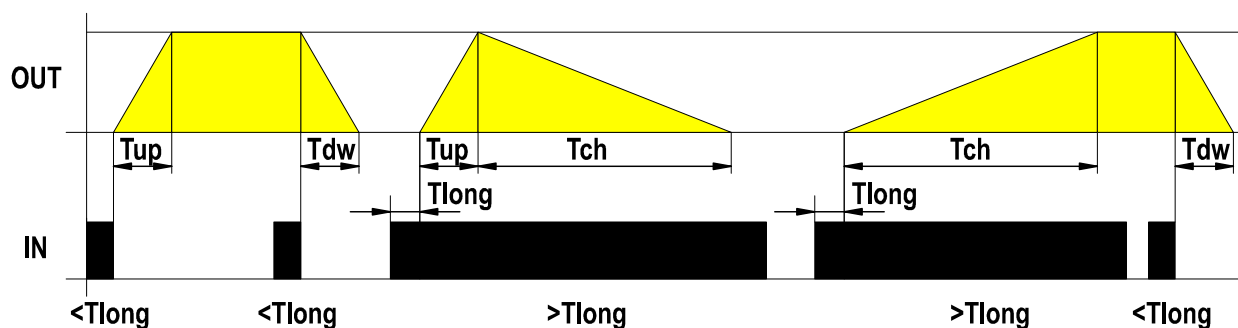
2.JEDNOTLAČÍTKOVÝ STMÍVAČ DMA1



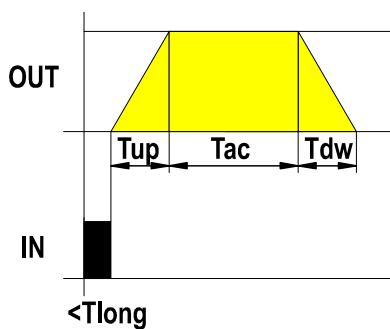
Obr. 2.3 Funkce stmívače bez aktivovaného náběhu/doběhu



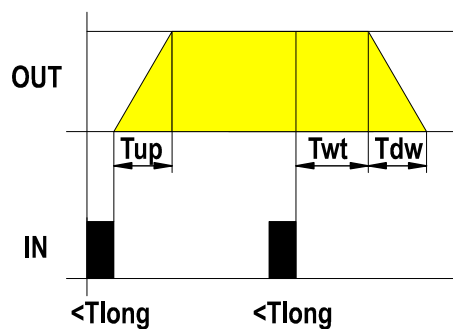
Obr. 2.4 Funkce stmívače s aktivovaným náběhem/doběhem



Obr. 2.5 Časování náběhu, doběhu a přeběhu stmívače



Obr. 2.6 Časování doby svitu



Obr. 2.7 Časování odchodového zpoždění

Příklad programu s voláním funkčního bloku **DMA1** v jazyce ST :

```
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

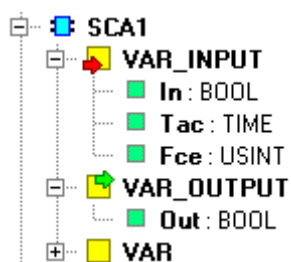
inst_DMA1 (In := input, Cfg := DMA1_Cfg, Out => output, Sig => signal,
           Sup => support);
END_PROGRAM
```

3. SCHODIŠŤOVÝ AUTOMAT SE SIGNALIZACÍ PŘED VYPNUTÍM SCA1

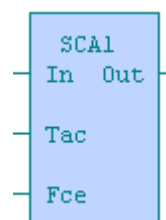
Funkční blok slouží k ovládání osvětlení schodiště (zpozděné vypnutí). Interval doby svícení je dán parametrem *Tac*. Obsahuje výstrahu před blížícím se vypnutím výstupu dvojitým prokliknutím 40s a 30s před vypnutím. Blok může pracovat v jednom ze 3 možných režimů (parametr *Fce*) :

- (0) AUTO – časování bez možnosti prodloužení času
- (1) PROG – časování s možností prodloužení času počtem stisků tlačítka
- (2) ON – výstup je trvale sepnutý (servisní režim, úklid, ...)

V režimech AUTO a PROG je možno časování předčasně ukončit dlouhým stiskem tlačítka (>2s). Tato funkce též slouží jako ochrana trvalého svícení při zablokování stisknutého tlačítka. V režimu AUTO lze dobu sepnutí výstupu prodloužit až po signalizaci blížícího se výpnutí výstupu (40s před vypnutím výstupu).



Obr. 3.1 Struktura FB SCA1



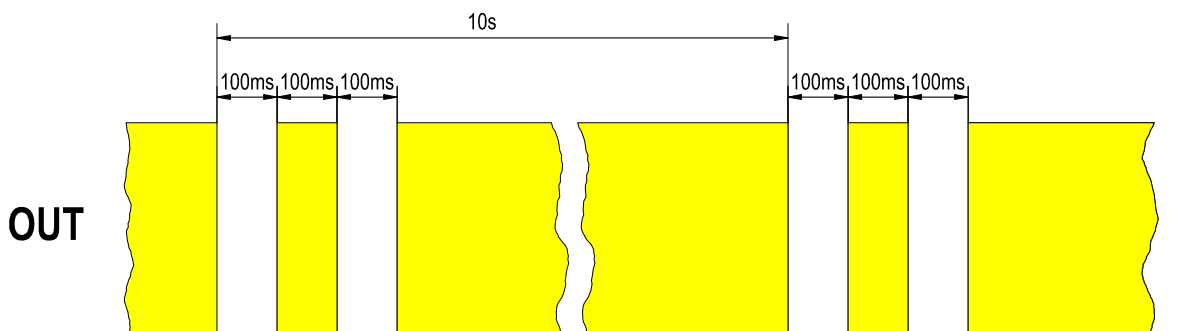
Obr. 3.2 Vzhled FB SCA1

Popis proměnných :

Název	Význam	Typ	Formát
In	ovládací tlačítko	vstupní	bool
Tac	čas svícení		time
Fce	režim schodišťového automatu		usint
Out	výstupní kontakt	výstupní	bool

Jednotlivé časové funkce bloku ilustrují následující obrázky :

FCE	Popis	Graf
0	AUTO - časování bez možnosti prodloužení času	
1	PROG - časování s možností prodloužení času počtem stisků tlačítka	
2	ON - výstup je trvale sepnutý	



Obr. 3.3 Časové poměry výstražného problíknutí

V případě nastavení doby sepnutí výstupu na dobu kratší než 30s není výstražné problíknutí aktivováno.

Příklad programu s voláním funkčního bloku **SCA1** v jazyce ST :

```

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

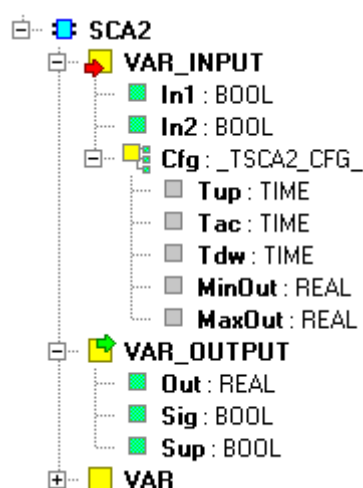
inst_SCA1(In := input, Tac := T#3m, Fce := 0, Out => output);
END_PROGRAM
    
```


4. SCHODIŠŤOVÝ AUTOMAT SE STMÍVÁNÍM SCA2

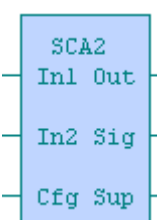
Funkční blok slouží k inteligentnímu ovládní osvětlení schodiště s funkcí postupného rozsvícení (náběhová rampa) a stmívání (doběhová rampa). Jako ovládací vstup lze použít tlačítko *In1* nebo vypínač *In2*. Doba náběhu se nastavuje parametrem *Tup*, doba svícení parametrem *Tac* a doba doběhu parametrem *Tdw*. Výstup *Out* se při činnosti schodišťového automatu může pohybovat v mezích *MinOut*, *MaxOut*.

Výstup *Sig* je určen pro signalizaci chodu stmívače. Výstup *Sup* je určen pro ovládní napájení předřadníků zářivek a je automaticky nastaven do log.1 v případě, že je na výstupu *Out* hodnota vyšší, než 10% hodnoty *MaxOut*.

Právě probíhající cyklus svícení lze prodloužit opětovným stiskem tlačítka. Ovládní pomocí vypínače je funkčně nadřazeno ovládní pomocí tlačítka a lze jej s výhodou využít pro servisní účely (úklid, kontrola a výměna osvětlovacích prvků, ...).



Obr. 4.1 Struktura FB SCA2

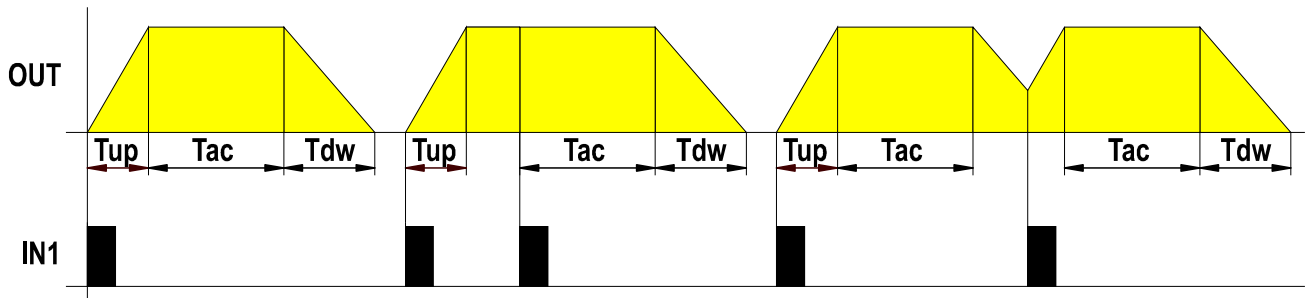


Obr. 4.2 Vzhled FB SCA2

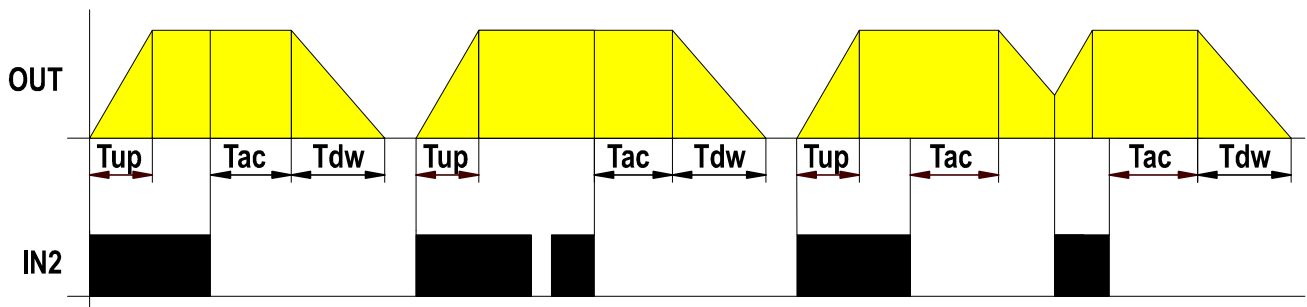
Popis proměnných :

Název	Význam	Typ	Formát
In1	ovládací tlačítko	vstupní	bool
In2	ovládací spínač		bool
Cfg	konfigurační struktura bloku		_TSCA2_CFG_
.Tup	čas náběhu		time
.Tac	čas svícení		time
.Tdw	čas doběhu		time
.MinOut	minimum výstupu		real
.MaxOut	maximum výstupu	real	
Out	výstup stmívače	výstupní	real
Sig	signalizace chodu		bool
Sup	ovládání předřadníků		bool

Činnost bloku ilustrují následující obrázky.



Obr. 4.3 Funkce bloku při ovládání tlačítkem In1



Obr. 4.4 Funkce bloku při ovládání vypínačem In2

Příklad programu s voláním funkčního bloku **SCA2** v jazyce ST :

```

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

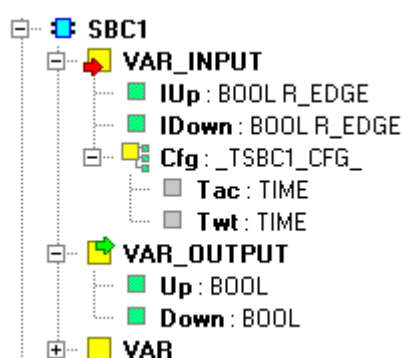
inst_SCA2 (In1 := input1, In2 := input2, Cfg := SCA2_cfg, Out => output,
          Sig => signal, Sup => support);
END_PROGRAM
    
```

5. OVLADAČ ROLETOVÝCH ŽALUZÍ SBC1

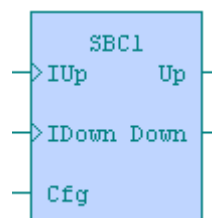
Funkční blok slouží k ovládání pohonu roletových žaluzií. Ovládání je realizováno pomocí dvou tlačítek. Blok zajišťuje časovou kontrolu doby běhu pohonu a prodlevu při reverzaci chodu pohonu (ochrana pohonu z důvodů elektrických i mechanických).

Po stisku tlačítka *IUp* je aktivován výstup pro pohyb žaluzií směrem nahoru *Up*. Dalším stiskem kteréhokoliv tlačítka je výstup *Up* deaktivován. Po stisku tlačítka *IDown* je aktivován výstup pro pohyb žaluzií směrem dolů *Down*. Dalším stiskem kteréhokoliv tlačítka je výstup *Down* deaktivován.

Při aktivaci výstupů na dobu delší než *Tac* budou výstupy automaticky deaktivovány. Při reverzaci chodu pohonu je mezi aktivace jednotlivých výstupů vždy vložena časová prodleva *Twt*.



Obr. 5.1 Struktura FB SBC1



Obr. 5.2 Vzhled FB SBC1

Popis proměnných :

Název	Význam	Typ	Formát
IUp	ovládací tlačítko nahoru	vstupní	bool
IDown	ovládací tlačítko dolů		bool
Cfg	konfigurační struktura bloku		_TSBC1_CFG_
.Tac	maximální doba běhu pohonu		time
.Twt	prodleva při reverzaci chodu pohonu		time
Up	výstup pro chod pohonu nahoru	výstupní	bool
Down	výstup pro chod pohonu dolů		bool

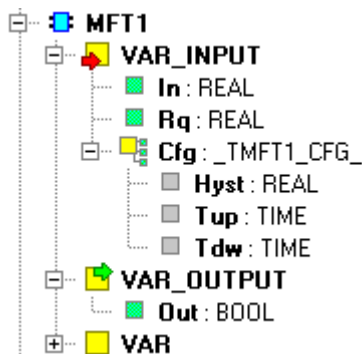
Příklad programu s voláním funkčního bloku **SBC1** v jazyce ST :

```
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

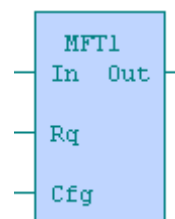
inst_SBC1(IUp := input1, IDown := input2 , Cfg := SBC1_Cfg, Up => output1,
          Down => output2);
END_PROGRAM
```

6. JEDNOÚROVŇOVÝ TERMOSTAT MFT1

Blok řeší algoritmus klasického jednoúrovňového termostatu. Výstupní kontakt *Out* je sepnut do doby dosažení měřené teploty *In* nastavené teplotě *Rq*, kdy vypne. Nastavitelná hystereze *Hyst* a časová zpoždění *Tup*/*Tdw* zabraňují častému spínání (kmitání) výstupu *Out*.



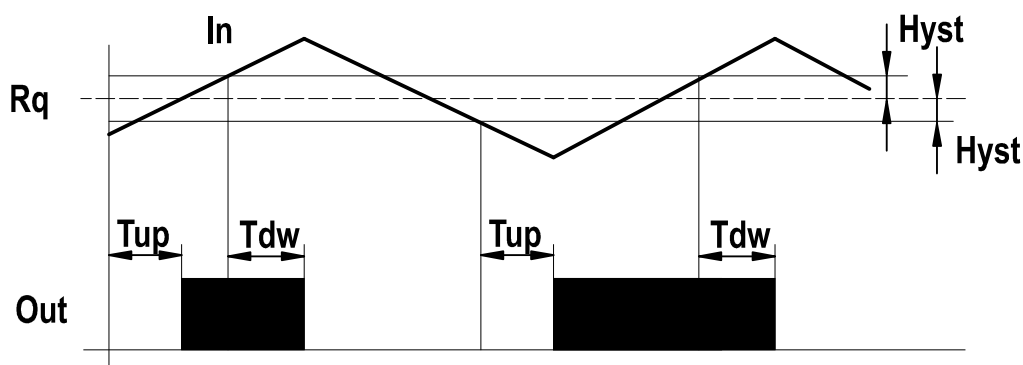
Obr. 6.1 Struktura FB MFT1



Obr. 6.2 Vzhled FB MFT1

Popis proměnných :

Název	Význam	Typ	Formát
In	měřená teplota	vstupní	real
Rq	žádaná teplota		real
Cfg	konfigurační struktura bloku		_TMFT1_CFG_
.Hyst	hystereze		real
.Tup	prodleva sepnutí		time
.Tdw	prodleva vypnutí		time
Out	výstup termostatu	výstupní	bool



Obr. 6.3 Funkce bloku MFT1

6.JEDNOÚROVŇOVÝ TERMOSTAT MFT1

Příklad programu s voláním funkčního bloku **MFT1** v jazyce ST :

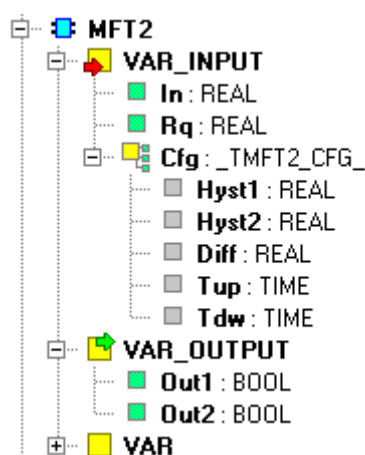
```
PROGRAM Example_MFT1
VAR
  input    : real;
  request  : real := 21.0;
  MFT1_Cfg : _TMFT1_CFG_ := (Hyst:= 0.1,
                             Tup:= T#5s,
                             Tdw:= T#5s);
  inst_MFT1 : MFT1;           //instance FB MFT1
  output    : bool;
END_VAR

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

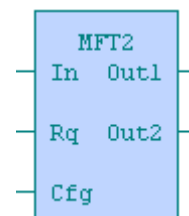
7. DVOJÚROVŇOVÝ TERMOSTAT MFT2

V pásmu nastavené difference *Diff* pracuje výstup *Out1* jako klasický termostat ke vstupu *In* a žádané teplotě *Rq*. Pokud však teplota poklesne pod nastavenou diferenci *Diff*, sepne i výstup *Out2*. Oba dva výstupy *Out* rozepnou současně při dosažení požadované hodnoty *Rq*. Při spínání a rozpínání výstupů *Out* jsou uvažovány hystereze *Hyst1/Hyst2* a časové prodlevy *Tup/Tdw*.

Typický případ tohoto termostatu je např. při řízení dvoustupňového kotle, kdy první stupeň je řízen dle nastavené teploty a pomocný druhý stupeň je zapínán, poklesne-li teplota pod nastavenou diferenci. Tímto prvním stupni vypomáhá, pokud se teplota prudce sníží.



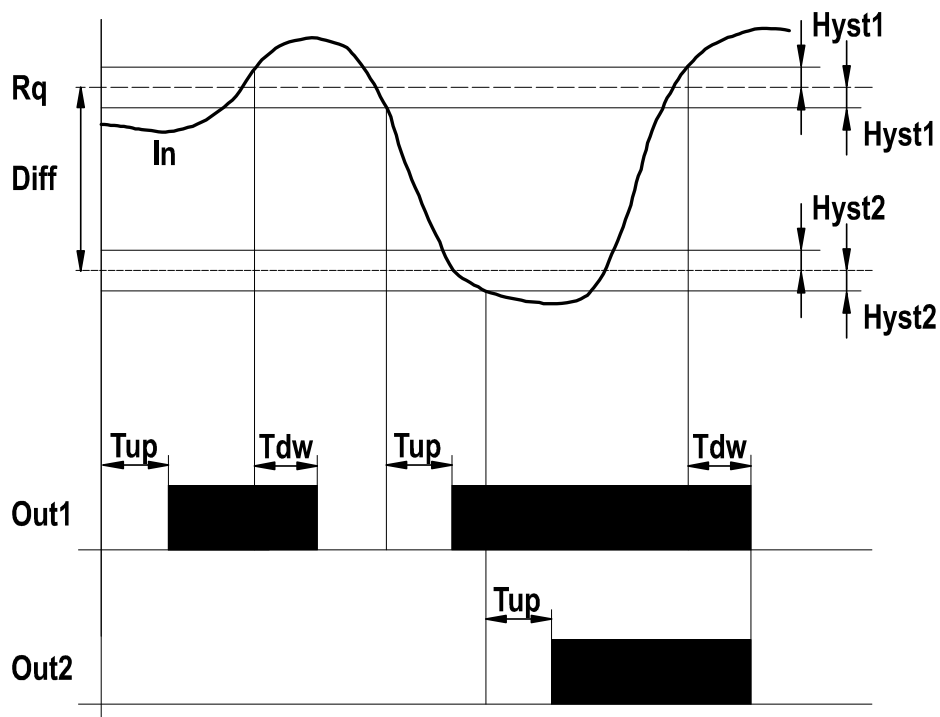
Obr. 7.1 Struktura FB MFT2



Obr. 7.2 Vzhled FB MFT2

Popis proměnných :

Název	Význam	Typ	Formát
In	měřená teplota	vstupní	real
Rq	žádaná teplota		real
Cfg	konfigurační struktura bloku		_TMFT2_CFG_
.Hyst1	hystereze k Out1		real
.Hyst2	hystereze k Out2		real
.Diff	teplotní diference		real
.Tup	prodleva sepnutí		time
.Tdw	prodleva vypnutí	time	
Out1	výstup 1. stupně	výstupní	bool
Out2	výstup 2. stupně		bool



Obr. 7.3 Funkce bloku MFT2

Příklad programu s voláním funkčního bloku **MFT2** v jazyce ST :

```

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

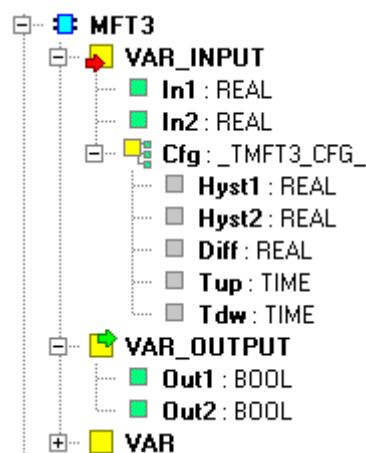
inst_MFT2(In := input, Rq := request, Cfg := MFT2_Cfg, Out1 => output1,
           Out2 => output2);
END_PROGRAM

```

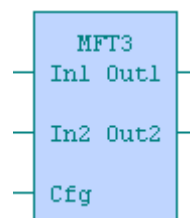
8. DIFERENČNÍ TERMOSTAT MFT3

V diferenčním termostatu dochází ke spínání vždy odpovídajícího výstupu *Outx* ke vstupu *Inx*, jehož teplota je při překročení difference *Diff* nižší. K rozepnutí výstupu dochází po zvýšení vstupní teploty o odpovídající hysterezi *Hystx*.

Diferenční termostat se používá pro udržování dvou stejných teplot např. v topných systémech (kotel - zásobník vody), solárních systémech (kolektor - zásobník), ohřevu vody (ohřivač vody – rozvod vody) apod.



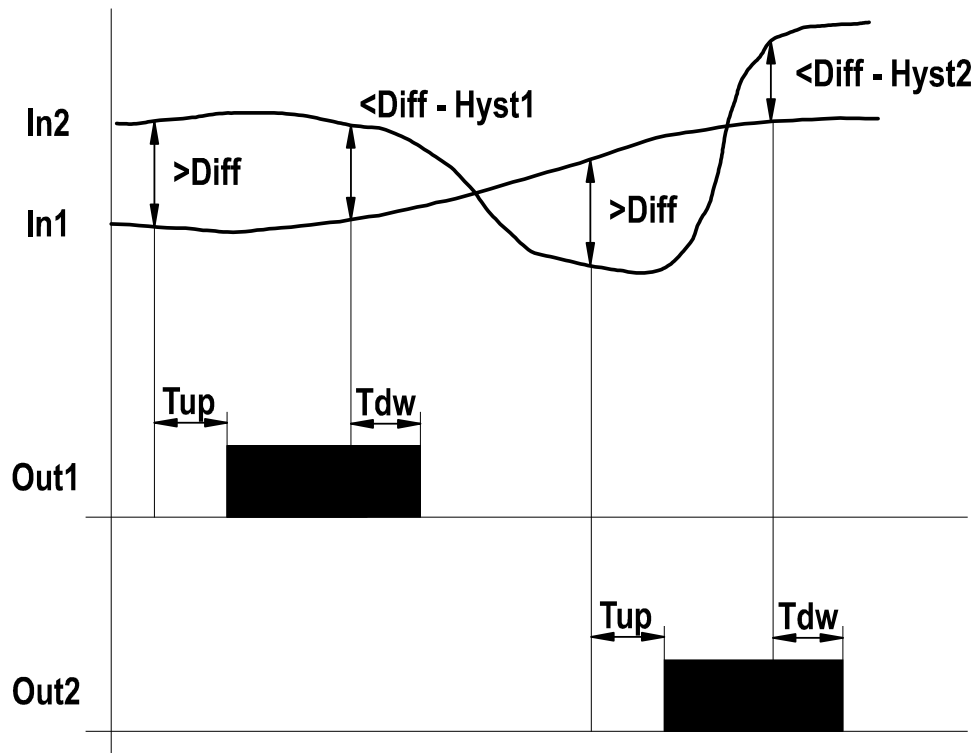
Obr. 8.1 Struktura FB MFT3



Obr. 8.2 Vzhled FB MFT3

Popis proměnných :

Název	Význam	Typ	Formát
In1	měřená teplota 1	vstupní	real
In2	měřená teplota 2		real
Cfg	konfigurační struktura bloku		_TMFT3_CFG_
.Hyst1	hystereze příslušející k teplotě 1		real
.Hyst2	hystereze příslušející k teplotě 2		real
.Diff	teplotní difference		real
.Tup	prodleva sepnutí		time
.Tdw	prodleva vypnutí		time
Out1	výstup příslušející k teplotě 1	výstupní	bool
Out2	výstup příslušející k teplotě 2		bool



Obr. 8.3 Funkce bloku MFT3

Příklad programu s voláním funkčního bloku **MFT3** v jazyce ST :

```

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

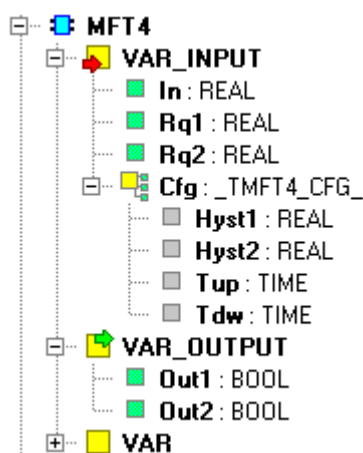
inst_MFT3(In1 := input1, In2 := input2, Cfg := MFT3_Cfg, Out1 => output1,
          Out2 => output2);
END_PROGRAM

```

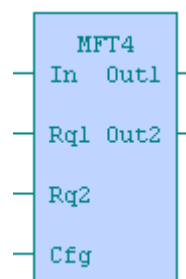
9. TERMOSTAT S MRTVOU ZÓNOU MFT4

Pokud je měřená teplota In nižší než žádaná teplota $Rq1$, spíná výstupní kontakt topení $Out1$, při teplotě vyšší než $Rq1$ vypíná. Pokud je měřená teplota In vyšší než žádaná teplota $Rq2$, spíná výstupní kontakt chlazení $Out2$, při teplotě nižší než $Rq2$ vypíná. Při spínání a rozpínání výstupů Out jsou uvažovány hysterese $Hyst1/Hyst2$ a časové prodlevy Tup/Tdw .

Tuto funkci lze využít např. pro automatické ohřívání a chlazení přiváděného vzduchu u ventilačních systémů tak, aby teplota přiváděného vzduchu byla vždy v mezích teplot $Rq1$ a $Rq2$.



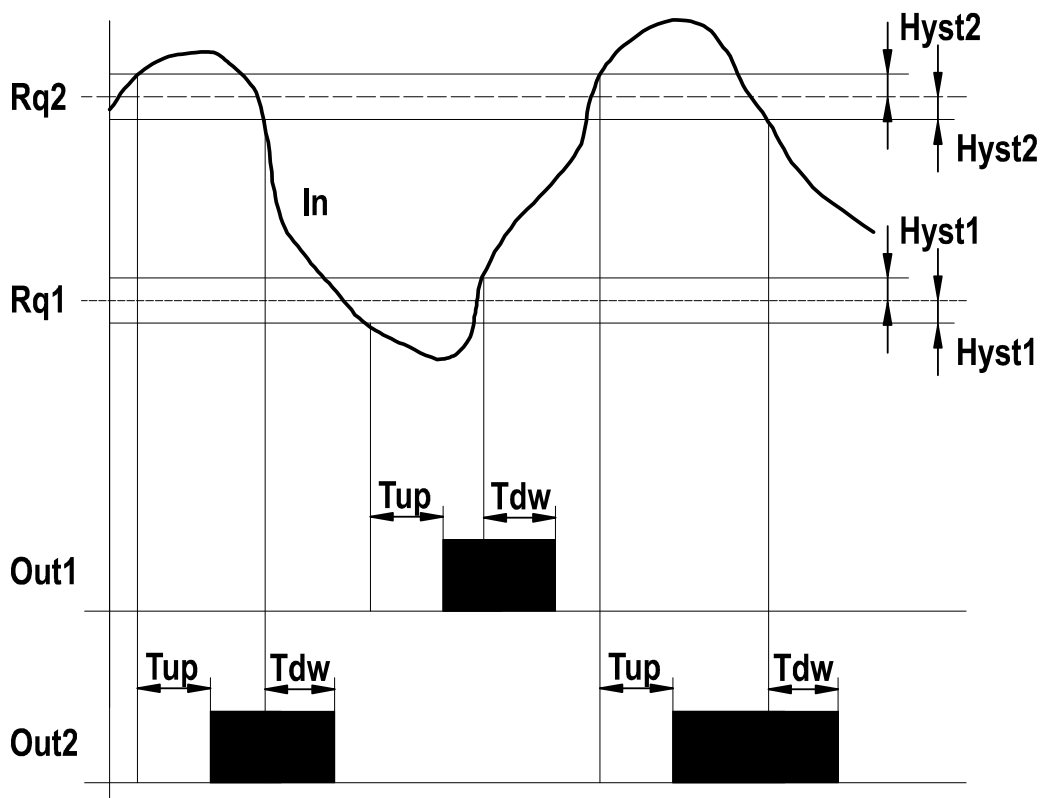
Obr. 9.1 Struktura FB MFT4



Obr. 9.2 Vzhled FB MFT4

Popis proměnných :

Název	Význam	Typ	Formát
In	měřená teplota	vstupní	real
Rq1	žádaná teplota 1		real
Rq2	žádaná teplota 2		real
Cfg	konfigurační struktura bloku		_TMFT4_CFG_
.Hyst1	hysterese příslušející k teplotě Rq1		real
.Hyst2	hysterese příslušející k teplotě Rq2		real
.Tup	prodleva sepnutí		time
.Tdw	prodleva vypnutí	time	
Out1	výstup topení	výstupní	bool
Out2	výstup chlazení		bool



Obr. 9.3 Funkce bloku MFT4

Příklad programu s voláním funkčního bloku **MFT4** v jazyce ST :

```

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);

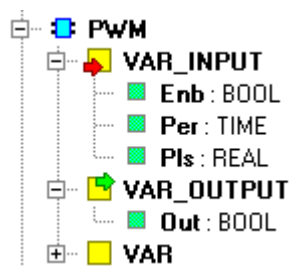
  inst_MFT4  : MFT4;           //instance FB MFT4
  output1    : bool;
  output2    : bool;
END_VAR

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

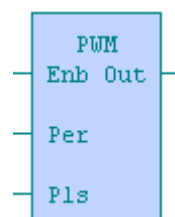
```

10. PULSNĚ ŠÍŘKOVÁ MODULACE PWM

Blok slouží k realizaci funkce pulsně šířkové modulace. Perioda modulace je dána hodnotou proměnné *Per*, šířka modulovaného pulsu je dána hodnotou proměnné *Pul*. Hodnota *Pul* se zadává v rozsahu 0-100%. Aktivace pulsně šířkové modulace na výstupu *Out* je podmíněna nastavením povolovací proměnné *Enb*.



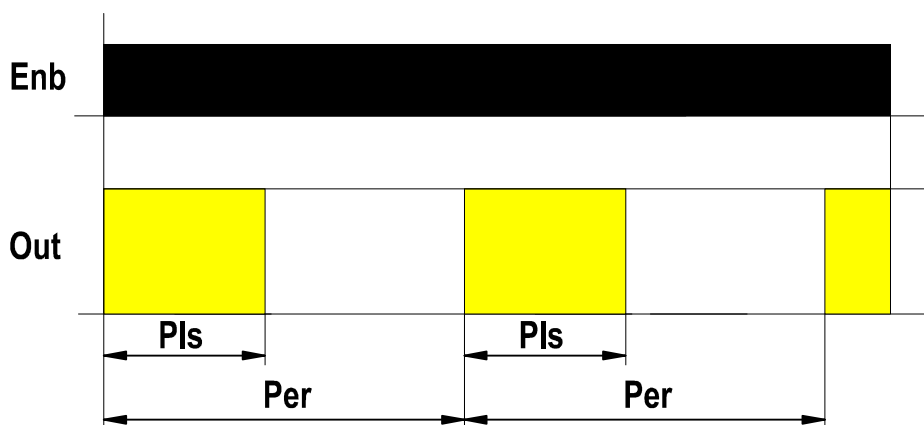
Obr. 10.1 Struktura FB PWM



Obr. 10.2 Vzhled FB PWM

Popis proměnných :

Název	Význam	Typ	Formát
Enb	povolení modulace	vstupní	bool
Per	perioda modulace		time
Pls	šířka pulsu [0-100%]		real
Out	výstup modulace	výstupní	bool



Obr. 10.3 Funkce bloku PWM

10.PULSNĚ ŠÍŘKOVÁ MODULACE PWM

Příklad programu s voláním funkčního bloku **PWM** v jazyce ST :

```
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
```

Poznámky:



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

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu
www.tecomat.cz