

Knihovna RexLib

TXV 003 45.01
šesté vydání
prosinec 2010
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
Leden 2008	1	První vydání – popis PSMPC
Březen 2008	2	Doplněny bloky PIDU, PIDMA, SCU, SCUUV, MCU, PWM
Červen 2008	3	Finální úprava jmen konstant a funkčních bloků
Srpen 2009	4	Doplněné bloky PIDU_SCU, PIDU_SCUV, PIDMA_SCU, PIDMA_SCUV, fbSetPIDMA_Par, fbSetPIDMA_SCUPar, fbSetPIDMA_SCUVPar, fbSampler a příslušné typy
Listopad 2009	5	Rozšířen popis struktur, opraveny popisy bloků s PID[MA]_SCU[V]
Prosinec 2010	6	Korekce odkazů

OBSAH

1 Úvod	4
1.1 Obecná pravidla pro použití bloků.....	4
2 Datové typy	5
2.1 Typ <code>_TPID_PARAM_</code>	6
2.2 Typ <code>_TPIDU_IN_</code>	7
2.3 Typ <code>_TPIDU_SCU_IN_</code>	8
2.4 Typ <code>_TPIDMA_IN_</code>	9
2.5 Typ <code>_TPIDMA_SCU_IN_</code>	10
2.6 Typ <code>_TPSMPC_IN_</code>	12
2.7 Typ <code>_TMCU_IN_</code>	13
2.8 Typ <code>_TPWM_IN_</code>	14
2.9 Typ <code>_TSCU_IN_</code>	15
2.10 Typ <code>_TSCUV_IN_</code>	16
2.11 Typ <code>_TSR_</code>	16
3 Konstanty	17
3.1 Konstanty <code>imtype_FOPDT</code> , <code>imtype_SOPDT</code> , <code>imtype_STEP_RES</code>	17
3.2 Konstanty <code>icotype_ANALOG</code> , <code>icotype_PWM</code> , <code>icotype_SCU</code> , <code>icotype_SCUV</code> ...	17
3.3 Konstanty <code>irtype_D</code> , ... , <code>irtype_PID</code>	18
3.4 Konstanty <code>ittype_D</code> , ... , <code>ittype_PID</code>	18
3.5 Konstanta <code>Namax</code>	18
4 Funkční bloky pro regulaci	19
4.1 Funkční blok <code>fbPIDU</code>	20
4.2 Funkční blok <code>fbPIDMA</code>	24
4.3 Funkční blok <code>fbPSMPC</code>	30
4.4 Funkční blok <code>fbMCU</code>	35
4.5 Funkční blok <code>fbPWM</code>	38
4.6 Funkční blok <code>fbSCU</code>	41
4.7 Funkční blok <code>fbSCUV</code>	44
4.8 Funkční blok <code>fbPIDU_SCU</code>	48
4.9 Funkční blok <code>bPIDU_SCUV</code>	51
4.10 Funkční blok <code>fbPIDMA_SCU</code>	54
4.11 Funkční blok <code>fbPIDMA_SCUV</code>	59
4.12 Funkční blok <code>fbSetPIDMAPar</code>	64
4.13 Funkční blok <code>fbSetPIDMA_SCUPar</code>	65
4.14 Funkční blok <code>fbSetPIDMA_SCUVPar</code>	66
4.15 Funkční blok <code>fbSampler</code>	67

1 ÚVOD

Knihovna RexLib obsahuje sadu funkčních bloků určených pro regulaci soustav. Tuto knihovnu lze použít pro řídicí systémy Foxtrot a systémy Tecomat TC-700 s procesorovou jednotkou CP-7000, CP-7004.

Knihovna RexLib obsahuje následující funkční bloky:

- **fbPIDU** základní PID regulátor
- **fbPIDMA** PID regulátor s autotunerem
- **fbPSMPC** prediktivní regulátor „pulse-step“
- **fbMCU** blok pro ruční zadávání hodnot
- **fbPWM** blok pulzně šířkové modulace
- **fbSCU** krokový regulátor s polohovou zpětnou vazbou
- **fbSCUV** krokový regulátor s rychlostním vstupem
- **fbPIDU_SCU** základní PID regulátor s trojstavovým řízením se zpětnou vazbou
- **fbPIDU_SCUV** základní PID regulátor s trojstavovým řízením bez zpětné vazby
- **fbPIDMA_SCU** PID regulátor s autotunerem a trojstavovým řízením se zpětnou vazbou
- **fbPIDMA_SCUV** PID regulátor s autotunerem a trojstavovým řízením bez zpětné vazby
- **fbSampler** blok pro zachycení přechodové charakteristiky a základní identifikaci
- **fbSetPIDMAPar** blok pro přesun výsledků autotuneru do konfigurační struktury
- **fbSetPIDMA_SCUPar** blok pro přesun výsledků autotuneru do konfigurační struktury
- **fbSetPIDMA_SCUVPAr** blok pro přesun výsledků autotuneru do konfigurační struktury

Pokud chceme funkce z knihovny RexLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Knihovna je dodávána jako součást instalace prostředí Mosaic od verze 2.0.19. Knihovna RexLib vyžaduje podporu regulačních bloků ve firmware centrální jednotky.

1.1 Obecná pravidla pro použití bloků

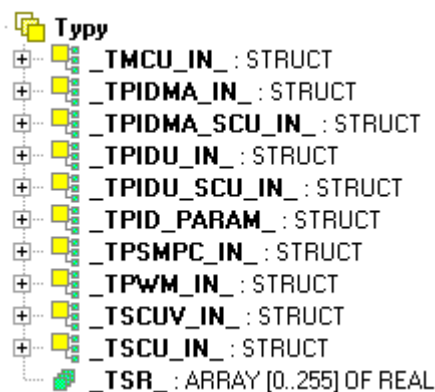
Většina bloků v této knihovně pracuje v pevném časovém rámci s periodou vzorkování T_s . Pro správnou funkci je nutné tyto bloky volat každou otočku PLC, přičemž je nutné, aby nejdelší otočka PLC byla v nejhorším případě $2 \times$ kratší než perioda T_s .

Dalším společným rysem funkčních bloků je proměnná *CHANGED* ve struktuře s parametry funkčního bloku, která se musí nastavit do hodnoty logická 1, aby blok parametry změněné během běhu PLC přijal. V hodnotě 1 musí proměnná *CHANGED* setrvat alespoň po dobu jedné periody vzorkování T_s . Jedinou výjimkou je první volání bloku po startu systému, kdy jsou parametry ze struktury přijaty bez ohledu na stav proměnné *CHANGED*.

2 DATOVÉ TYPY

V knihovně RexLib jsou definovány následující datové typy:

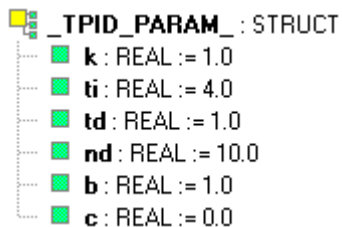
- `_TPID_PARAM_` struktura parametrů PID algoritmu
- `_TPIDU_IN_` struktura vstupních parametrů pro fbPIDU
- `_TPIDU_SCU_IN_` struktura vstupních parametrů pro fbPIDU s trojstavovým řízením
- `_TPIDMA_IN_` struktura vstupních parametrů pro fbPIDMA
- `_TPIDMA_SCU_IN_` struktura vstupních parametrů pro fbPIDMA s trojstavovým řízením
- `_TPSMPC_IN_` struktura vstupních parametrů pro fbPSMPC
- `_TMCU_IN_` struktura vstupních parametrů pro fbMCU
- `_TPWM_IN_` struktura vstupních parametrů pro fbPWM
- `_TSCU_IN_` struktura vstupních parametrů pro fbSCU
- `_TSCUV_IN_` struktura vstupních parametrů pro fbSCUV
- `_TSR_` pole pro zadání přechodové charakteristiky pro fbPSMPC



Podrobný popis je uveden v rámci popisu jednotlivých funkčních bloků.

2.1 Typ `_TPID_PARAM_`

Typ `PID_PARAM` obsahuje soubor parametrů PID algoritmu, implementovaného v blocích `fbPIDU` a `fbPIDMA`.

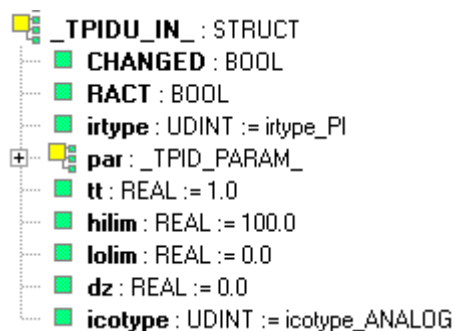


Význam jednotlivých položek struktury `_TPID_PARAM_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>k</i>	REAL	zesílení
<i>ti</i>	REAL	integrační časová konstanta v sekundách
<i>td</i>	REAL	derivační časová konstanta v sekundách
<i>nd</i>	REAL	omezení derivační složky (časová konstanta filtru v sekundách)
<i>b</i>	REAL	váhový koeficient požadované hodnoty – proporcionální složka
<i>c</i>	REAL	váhový koeficient požadované hodnoty – derivační složka

2.2 Typ `_TPIDU_IN_`

Typ `_TPIDU_IN_` definuje strukturu vstupních parametrů pro regulátor **fbPIDU**.

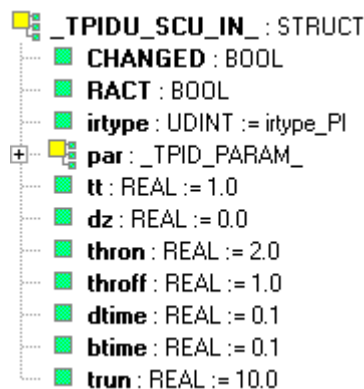


Význam jednotlivých položek struktury `_TPIDU_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>irtype</i>	UDINT	typ regulátoru (<i>irtype_D</i> , ... , <i>irtype_PID</i>)
<i>par</i>	_TPID_PARAM_	parametry PID algoritmu
<i>tt</i>	REAL	sledovací časová konstanta v sekundách pro beznárazový přechod z manuálního do automatického režimu
<i>hilim</i>	REAL	horní limit saturace
<i>lolim</i>	REAL	dolní limit saturace
<i>dz</i>	REAL	pásmo necitlivosti
<i>icotype</i>	UDINT	typ výstupu regulátoru (<i>icotype_ANALOG</i> ... <i>icotype_SCUV</i>)

2.3 Typ `_TPIDU_SCU_IN_`

Typ `_TPIDU_SCU_IN_` definuje strukturu vstupních parametrů pro bloky `fbPIDU_SCU` a `fbPIDU_SCUV`.

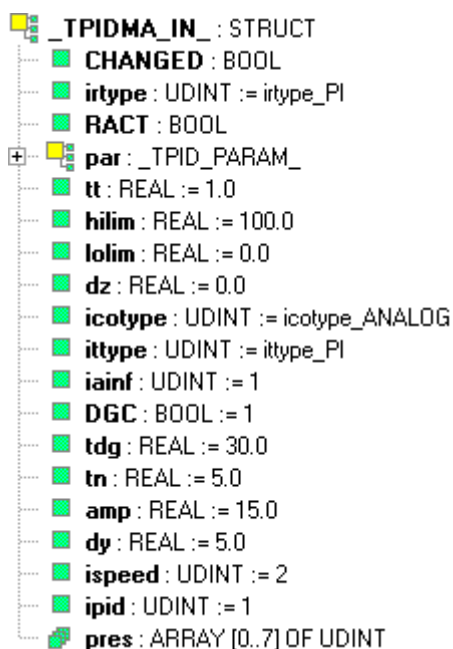


Význam jednotlivých položek struktury `_TPIDU_SCU_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>irtype</i>	UDINT	typ regulátoru (<i>irtype_D</i> , ... , <i>irtype_PID</i>)
<i>par</i>	<code>_TPID_PARAM_</code>	parametry PID algoritmu
<i>tt</i>	REAL	sledovací časová konstanta v sekundách
<i>dz</i>	REAL	pásmo necitlivosti
<i>thron</i>	REAL	práh pro zapnutí pulzu
<i>throff</i>	REAL	práh pro vypnutí pulzu
<i>dtime</i>	REAL	minimální požadované trvání pulzu v sekundách
<i>btime</i>	REAL	minimální doba prodlevy mezi dvěma následujícími pulzy opačné polarity v sekundách
<i>trun</i>	REAL	časová konstanta motoru v sekundách (určuje dobu, za kterou se motor posune o hodnotu 100)

2.4 Typ `_TPIDMA_IN_`

Typ `_TPIDMA_IN_` definuje strukturu vstupních parametrů pro regulátor **fbPIDMA**.



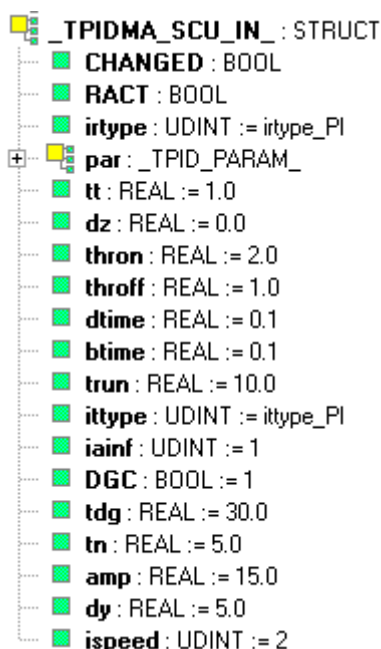
Význam jednotlivých položek struktury `_TPIDMA_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>irtype</i>	UDINT	typ regulátoru (<i>irtype_D</i> , ... , <i>irtype_PID</i>)
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>par</i>	<code>_TPID_PARAM_</code>	parametry PID algoritmu
<i>tt</i>	REAL	sledovací časová konstanta v sekundách
<i>hilim</i>	REAL	horní limit saturace
<i>lolim</i>	REAL	dolní limit saturace
<i>dz</i>	REAL	pásmo necitlivosti
<i>icotype</i>	UDINT	typ výstupu regulátoru (<i>icotype_ANALOG</i> ... <i>icotype_SCUV</i>)
<i>itype</i>	UDINT	požadovaný typ regulátoru pro návrh (<i>itype_PI</i> , <i>itype_PID</i>)
<i>iaINF</i>	UDINT	přidaná apriorní informace o procesu 1 – Statický proces 2 – Astatický proces
<i>DGC</i>	BOOL	příznak kompenzace gradientu trendu
<i>tdg</i>	REAL	doba odhadu gradientu trendu v sekundách
<i>tn</i>	REAL	doba odhadu šumu v sekundách
<i>amp</i>	REAL	amplituda pulzu

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>dy</i>	REAL	tolerance (práh) pro ukončení pulzu
<i>ispeed</i>	UDINT	požadovaná rychlost uzavřené smyčky 1 – Nízká rychlost 2 – Střední rychlost 3 – Vysoká rychlost
<i>ipid</i>	UDINT	forma zobrazení parametrů PID regulátoru 1 – Paralelní 2 – Sériová

2.5 Typ `_TPIDMA_SCU_IN_`

Typ `_TPIDMA_SCU_IN_` definuje strukturu vstupních parametrů pro bloky `fbPIDMA_SCU` a `fbPIDMA_SCUV`.



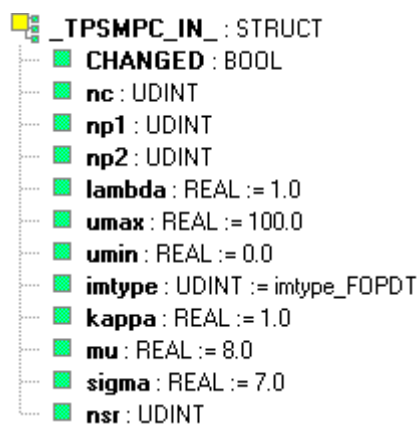
Význam jednotlivých položek struktury `_TPIDMA_SCU_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>irtype</i>	UDINT	typ regulátoru (<i>irtype_D</i> , ... , <i>irtype_PID</i>)
<i>par</i>	<code>_TPID_PARAM_</code>	parametry PID algoritmu
<i>tt</i>	REAL	sledovací časová konstanta v sekundách
<i>dz</i>	REAL	pásmo necitlivosti
<i>thron</i>	REAL	práh pro zapnutí pulzu
<i>throff</i>	REAL	práh pro vypnutí pulzu

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>dtime</i>	REAL	minimální požadované trvání pulzu v sekundách
<i>btime</i>	REAL	minimální doba prodlevy mezi dvěma následujícími pulzy opačné polariry v sekundách
<i>trun</i>	REAL	časová konstanta motoru v sekundách (určuje dobu, za kterou se motor posune o hodnotu 100)
<i>itype</i>	UDINT	požadovaný typ regulátoru pro návrh (itype_PI, itype_PID)
<i>iainf</i>	UDINT	přidaná apriorní informace o procesu 1 – Statický proces 2 – Astatický proces
<i>DGC</i>	BOOL	příznak kompenzace gradientu trendu
<i>tdg</i>	REAL	doba odhadu gradientu trendu v sekundách
<i>tn</i>	REAL	doba odhadu šumu v sekundách
<i>amp</i>	REAL	amplituda pulzu
<i>dy</i>	REAL	tolerance (práh) pro ukončení pulzu
<i>ispeed</i>	UDINT	požadovaná rychlost uzavřené smyčky 1 – Nízká rychlost 2 – Střední rychlost 3 – Vysoká rychlost

2.6 Typ `_TPSMPC_IN_`

Typ `_TPSMPC_IN_` definuje strukturu vstupních parametrů pro regulátor **fbPSMPC**.



Význam jednotlivých položek struktury `_TPSMPC_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>nc</i>	UDINT	horizont řízení
<i>np1</i>	UDINT	začátek intervalu predikce
<i>np2</i>	UDINT	konec intervalu predikce
<i>lambda</i>	REAL	pokutový koeficient změn řízení
<i>umax</i>	REAL	horní mez výstupu regulátoru
<i>umin</i>	REAL	dolní mez výstupu regulátoru
<i>imtype</i>	UDINT	typ modelu řízené soustavy (<i>imtype_FOPDT</i> , <i>imtype_SOPDT</i> , <i>imtype_STEP_RES</i>)
<i>kappa</i>	REAL	κ – statické zesílení
<i>mu</i>	REAL	μ – celková časová konstanta v sek. – míra zpoždění soustavy
<i>sigma</i>	REAL	σ – míra délky odezvy soustavy
<i>nsr</i>	UDINT	délka diskretní přechodové charakteristiky

2.7 Typ *_TMCU_IN_*

Typ *_TMCU_IN_* definuje strukturu vstupních parametrů pro funkční blok **fbMCU**.

```

_TMCU_IN_ : STRUCT
  CHANGED : BOOL
  SATF : BOOL := true
  tt : REAL := 1.0
  tm : REAL := 1.0
  y0 : REAL
  q : REAL := 5.0
  ta : REAL := 4.0
  tf : REAL := 8.0
  hilim : REAL := 100.0
  lolim : REAL := -100.0

```

Význam jednotlivých položek struktury *_TMCU_IN_* je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>SATF</i>	BOOL	příznak saturace
<i>tt</i>	REAL	časová konstanta vysledování vstupní hodnoty v sekundách
<i>tm</i>	REAL	časová konstanta počáteční strmosti najíždění
<i>y0</i>	REAL	počáteční hodnota výstupu
<i>q</i>	REAL	faktor určující změnu strmosti najíždění
<i>ta</i>	REAL	časový interval v sekundách, po kterém dojde ke zvýšení strmosti najíždění
<i>tf</i>	REAL	čas v sekundách, po kterém je ukončeno zvyšování strmosti najíždění
<i>hilim</i>	REAL	horní mez výstupu
<i>lolim</i>	REAL	dolní mez výstupu

2.8 Typ `_TPWM_IN_`

Typ `_TPWM_IN_` definuje strukturu vstupních parametrů pro funkční blok **fbPWM**.

```

└─ _TPWM_IN_ : STRUCT
├─ CHANGED : BOOL
├─ pertm : REAL := 10.0
├─ dtime : REAL := 0.1
├─ btime : REAL := 0.1
├─ offtime : REAL := 1.0
├─ asyfac : REAL := 1.0
└─ SYNCH : BOOL

```

Význam jednotlivých položek struktury `_TPWM_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	<i>BOOL</i>	nejméně jeden parametr se změnil
<i>pertm</i>	REAL	perioda šířkové modulace
<i>dtime</i>	REAL	minimální požadované trvání pulzu v sekundách
<i>btime</i>	REAL	minimální doba prodlevy mezi dvěma následujícími pulzy opačné polarity v sekundách
<i>offtime</i>	REAL	minimální doba prodlevy mezi pulzy opačné polarity
<i>asyfac</i>	REAL	faktor asymetrie
<i>SYNCH</i>	BOOL	synchronizační příznak pro začátek periody

2.9 Typ `_TSCU_IN_`

Typ `_TSCU_IN_` definuje strukturu vstupních parametrů pro regulátor **fbSCU**.

```

_TSCU_IN_ : STRUCT
  CHANGED : BOOL
  thron : REAL := 2.0
  throff : REAL := 1.0
  dtime : REAL := 0.1
  btime : REAL := 0.1
  RACT : BOOL
  trun : REAL := 10.0

```

Význam jednotlivých položek struktury `_TSCU_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>thron</i>	REAL	práh pro zapnutí pulzu
<i>throff</i>	REAL	práh pro vypnutí pulzu
<i>dtime</i>	REAL	minimální požadované trvání pulzu v sekundách
<i>btime</i>	REAL	minimální doba prodlevy mezi dvěma následujícími pulzy opačné polarity v sekundách
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>trun</i>	REAL	časová konstanta motoru v sekundách (určuje dobu, za kterou se motor posune o hodnotu 100)

2.10 Typ `_TSCUV_IN_`

Typ `_TSCUV_IN_` definuje strukturu vstupních parametrů pro regulátor **fbSCUV**.

```

└─┬─ _TSCUV_IN_ : STRUCT
   │ CHANGED : BOOL
   │ thron : REAL := 2.0
   │ throff : REAL := 1.0
   │ dtime : REAL := 0.1
   │ btime : REAL := 0.1
   │ RACT : BOOL
   │ CWOI : BOOL
   │ trun : REAL := 10.0
   │ tt : REAL := 1.0

```

Význam jednotlivých položek struktury `_TSCUV_IN_` je následující:

<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
<i>CHANGED</i>	BOOL	nejméně jeden parametr se změnil
<i>thron</i>	REAL	práh pro zapnutí pulzu
<i>throff</i>	REAL	práh pro vypnutí pulzu
<i>dtime</i>	REAL	minimální požadované trvání pulzu v sekundách
<i>btime</i>	REAL	minimální doba prodlevy mezi dvěma následujícími pulzy opačné polarity v sekundách
<i>RACT</i>	BOOL	opačný směr působení výstupu regulátoru
<i>CWOI</i>	BOOL	regulátor bez integrační složky
<i>trun</i>	REAL	doba přeběhu motoru v sekundách (určuje dobu, za kterou se motor posune o hodnotu 100)
<i>tt</i>	REAL	časová konstanta vysledování v sekundách

2.11 Typ `_TSR_`

Typ `_TSR_` je pole hodnot typu REAL, které se používá pro předání přechodové charakteristiky funkčnímu bloku **fbPSMPC**.

```
└─┬─ _TSR_ : ARRAY [0..255] OF REAL
```


3 KONSTANTY

V knihovně RexLib jsou definovány následující konstanty:

VAR_GLOBAL CONSTANT	
icotype_ANALOG	: UDINT := 1
icotype_PWM	: UDINT := 2
icotype_SCU	: UDINT := 3
icotype_SCUV	: UDINT := 4
imtype_FOPDT	: UDINT := 1
imtype_SOPDT	: UDINT := 2
imtype_STEP_RES	: UDINT := 3
irtype_D	: UDINT := 1
irtype_I	: UDINT := 2
irtype_ID	: UDINT := 3
irtype_P	: UDINT := 4
irtype_PD	: UDINT := 5
irtype_PI	: UDINT := 6
irtype_PID	: UDINT := 7
ittype_D	: UDINT := 1
ittype_I	: UDINT := 2
ittype_ID	: UDINT := 3
ittype_P	: UDINT := 4
ittype_PD	: UDINT := 5
ittype_PI	: UDINT := 6
ittype_PID	: UDINT := 7
Namax	: UINT := 256

3.1 Konstanty *imtype_FOPDT*, *imtype_SOPDT*, *imtype_STEP_RES*

Tyto konstanty jsou typu UDINT a slouží pro výběr modelu řízené soustavy (parametr *imtype* pro funkční blok *fbPSMPC*). Význam parametrů je následující:

- *imtype_FOPDT* jako model bude použita soustava 1.řádu
- *imtype_SOPDT* jako model bude použita soustava 2.řádu
- *imtype_STEP_RES* jako model bude použita přechodová charakteristika

3.2 Konstanty *icotype_ANALOG*, *icotype_PWM*, *icotype_SCU*, *icotype_SCUV*

Tyto konstanty jsou typu UDINT a slouží pro výběr typu výstupu regulátoru (parametr *icotype* u regulátorů *fbPIDU*, *fbPSMPC*, *fbPIDMA*). Význam parametrů je následující:

- *icotype_ANALOG* analogový výstup
- *icotype_PWM* šířkově modulovaný výstup
- *icotype_SCU* výstup pro krokový regulátor s polohovou zpětnou vazbou
- *icotype_SCUV* výstup pro krokový regulátor bez polohové zpětné vazby

3.3 Konstanty *irtype_D*, ... , *irtype_PID*

Tyto konstanty jsou typu UDINT a slouží pro výběr typu regulátoru (výběr zákona řízení, parametr *irtype* u regulátorů fbPIDU a fbPIDMA). Význam parametrů je následující:

- *irtype_D* regulátor D
- *irtype_I* regulátor I
- *irtype_ID* regulátor ID
- *irtype_P* regulátor P
- *irtype_PD* regulátor PD
- *irtype_PI* regulátor PI
- *irtype_PID* regulátor PID

3.4 Konstanty *itype_D*, ... , *itype_PID*

Tyto konstanty jsou typu UDINT a slouží pro výběr požadovaného typu regulátoru pro autotuner (výběr zákona řízení, parametr *itype* u regulátoru fbPIDMA). Autotuner podporuje momentálně pouze poslední dva. Význam parametrů je následující:

- *itype_D* regulátor D
- *itype_I* regulátor I
- *itype_ID* regulátor ID
- *itype_P* regulátor P
- *itype_PD* regulátor PD
- *itype_PI* regulátor PI
- *itype_PID* regulátor PID

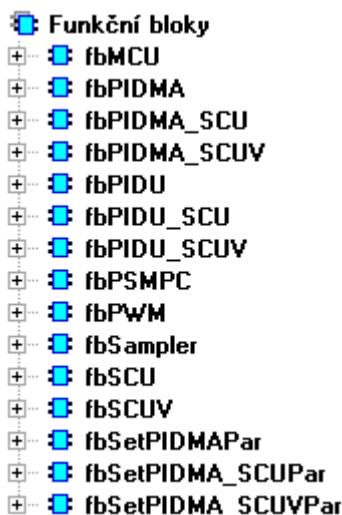
3.5 Konstanta *Namax*

Konstanta *Namax* je typu UINT a udává maximální možnou délku bufferu pro uložení přechodové charakteristiky pro regulátor fbPSMPC. Hodnota této konstanty je 256.

4 FUNKČNÍ BLOKY PRO REGULACI

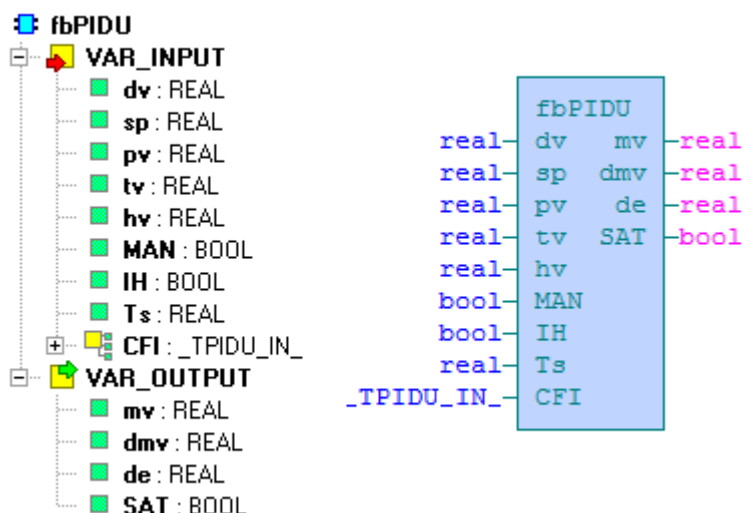
Knihovna RexLib obsahuje následující funkční bloky pro regulaci:

- **fbPIDU** základní PID regulátor
- **fbPIDMA** PID regulátor s autotunerem
- **fbPSMPC** prediktivní regulátor „pulse-step“
- **fbMCU** blok pro ruční zadávání hodnot
- **fbPWM** blok pulzně šířkové modulace
- **fbSCU** krokový regulátor s polohovou zpětnou vazbou
- **fbSCUV** krokový regulátor s rychlostním vstupem
- **fbPIDU_SCU** základní PID regulátor s trojstavovým řízením se zpětnou vazbou
- **fbPIDU_SCUV** základní PID regulátor s trojstavovým řízením se zpětnou vazbou
- **fbPIDMA_SCU** PID regulátor s autotunerem a trojstavovým řízením se zpětnou vazbou
- **fbPIDMA_SCUV** PID regulátor s autotunerem a trojstavovým řízením se zpětnou vazbou
- **fbSampler** blok pro zachycení přechodové charakteristiky a základní identifikaci
- **fbSetPIDMAPar** blok pro přesun výsledků autotuneru do konfigurační struktury
- **fbSetPIDMA_SCUPar** blok pro přesun výsledků autotuneru do konfigurační struktury
- **fbSetPIDMA_SCUVPar** blok pro přesun výsledků autotuneru do konfigurační struktury



4.1 Funkční blok fbPIDU

Základní PID regulátor



Funkční blok *fbPIDU* je základní blok pro vytvoření úplného regulátoru PID (*P, I, PI, PD, PID, PI+S*). V nejjednodušším případě může pracovat zcela samostatně a plnit standardní funkci PID regulátoru s dvěma stupni volnosti v automatickém ($MAN=0$) nebo manuálním režimu ($MAN=1$).

V automatickém režimu ($MAN=0$) realizuje blok *fbPIDU* řídicí zákon PID regulátoru se dvěma stupni volnosti ve tvaru

$$U(s) = \pm k \left\{ b \cdot W(s) - Y(s) + \frac{1}{ti \cdot s} [W(s) - Y(s)] + \frac{td \cdot s}{\frac{td}{nd} \cdot s + 1} [c \cdot W(s) - Y(s)] \right\} + Z(s)$$

kde $U(s)$ je Laplaceova transformace řídicí veličiny mv , $W(s)$ je Laplaceova transformace požadované hodnoty sp , $Y(s)$ je Laplaceova transformace regulované veličiny pv , $Z(s)$ je Laplaceova transformace dopředné vazby dv a k , ti , td , nd , b , c jsou parametry regulátoru. Znaménko pravé strany závisí na parametru *RACT*. Rozsah řídicí veličiny mv (polohového výstupu regulátoru) je omezen parametry *hilim*, *lolim*. Parametr dz udává pásmo necitlivosti v integrační složce regulátoru. Navíc integrační složka může být vypnuta a zafixována na své aktuální hodnotě vstupem *IH* ($IH=1$). Pro správnou funkci regulátoru je nutné propojit výstup regulátoru mv se vstupem tv a správně nastavit časovou konstantu vysledování tt . Doporučená hodnota je

$$tt \leq \sqrt{ti \cdot td}$$

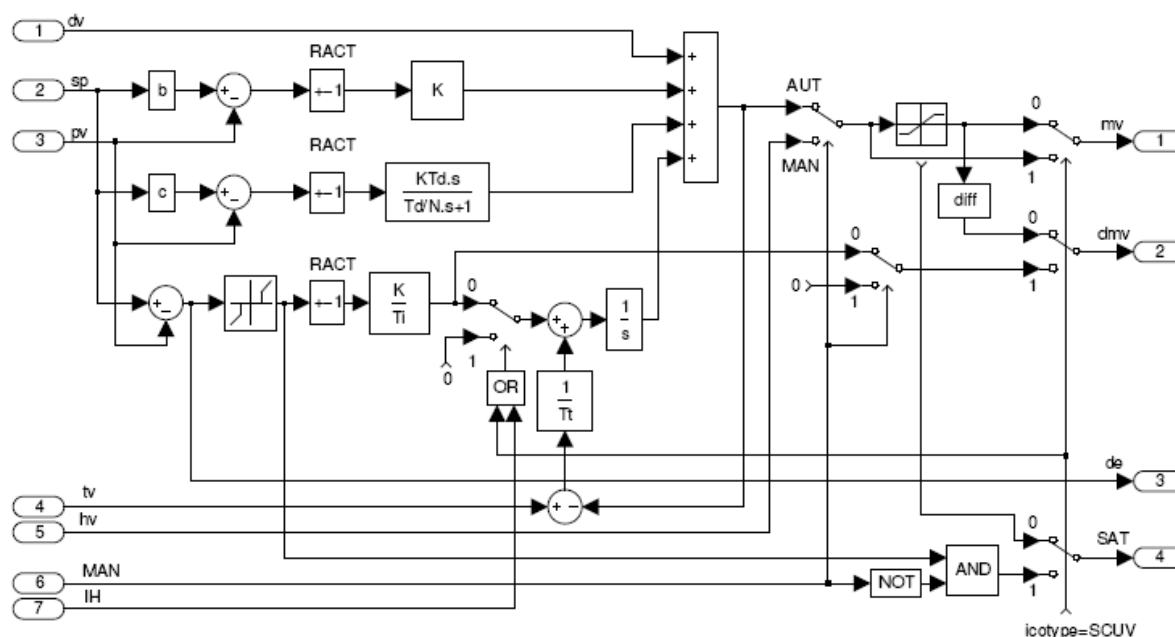
Tím bude zaručen bezrázový přechod při přepínání režimu regulátoru (manuální, automatický) a správná funkce regulátoru při saturaci výstupu mv (tzv. antiwindup). Přídavné výstupy dmv , de a SAT poskytují po řadě rychlostní výstup regulátoru (diference mv), regulační odchylku a příznak saturace výstupu regulátoru mv .

Jestliže je blok *fbPIDU* propojen s blokem *fbSCUV* (za účelem realizace krokového regulátoru bez polohové zpětné vazby), potom parametr *icotype* musí být nastaven na hodnotu 4 (*OUT_SCUV*) a význam výstupů mv , dmv a SAT je v tomto případě pozměněn: výstup mv je roven součtu P a D složky regulátoru, zatímco výstup dmv poskytuje diferenci jeho I složky a výstup SAT nese informaci pro blok *fbSCUV*, zda je regulační odchylka de v automatickém režimu menší než

pásmo necitlivosti dz . Pro případ propojení bloků $fbPIDU$ a $fbSCUV$ se navíc doporučuje volit váhový koeficient požadované hodnoty pro derivační složku c rovný nule.





Výchozí propojení $fbPIDU$ a $fbSCUV$ je realizováno v bloku $fbPIDU_SCUV$.

V manuálním režimu ($MAN=1$) je vstup hv kopírován na výstup mv . Celková regulační funkce bloku $fbPIDU$ je zřejmá z následujícího obrázku.

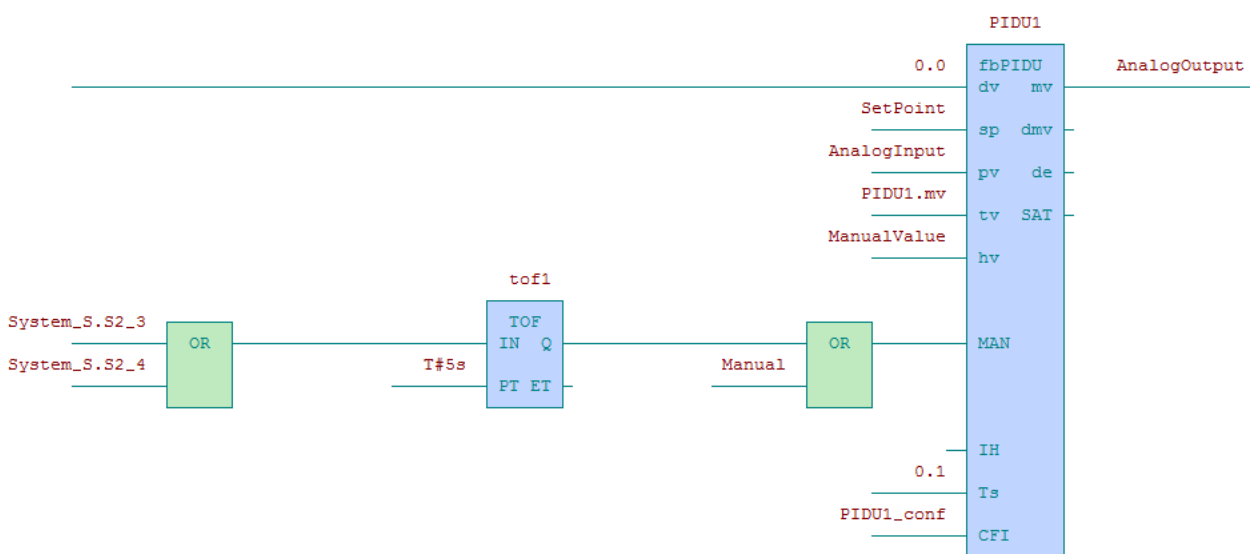


Popis proměnných :

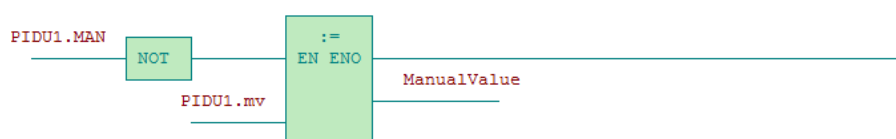
	Proměnná	Typ	Význam
VAR_INPUT			
	dv	REAL	dopředná vazba
	sp	REAL	žádaná veličina
	pv	REAL	regulovaná veličina
	tv	REAL	vstup pro vysledování integrační složky (propojit s výstupem mv)
	hv	REAL	výstup regulátoru v manuálním režimu
	MAN	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	IH	REAL	pozastavení integrace 0 = integrace je povolena, 1 = integrace je pozastavena
	Ts	REAL	vzorkovací perioda regulátoru v sekundách
	CFI	<code>_TPIDU_IN_</code>	parametry regulátoru (viz 2.2 Typ <code>_TPIDU_IN_</code>)

	Proměnná	Typ	Význam
VAR_OUTPUT			
	<i>mv</i>	REAL	řídící veličina (polohový výstup regulátoru)
	<i>dmv</i>	REAL	rychlostní výstup regulátoru
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován

Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDU* v jazyce FDB. Časovač TOF zajišťuje bezrázový start regulátoru po restartu.



Následující obvod zajišťuje bezrázový přechod z automatického do manuálního režimu:



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogOutput – analogový výstup ovládající akční člen

SetPoint – proměnná s žádanou hodnotou

Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU1_Conf – struktura s parametry funkčního bloku

V jazyce ST předchozím zapojením odpovídá následující program.

```
VAR GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
ManualValue : REAL;
PIDU1_conf : _TPIDU_IN_ := (irtype:= irtype_PID,
                           par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                   nd:= 10.0, b:= 1.0, c:= 0.0),
                           tt:= 1.0, hilim:= 100.0, lolim:= 0.0,
                           dz:= 0.0, icotype:= icotype_ANALOG);
END_VAR

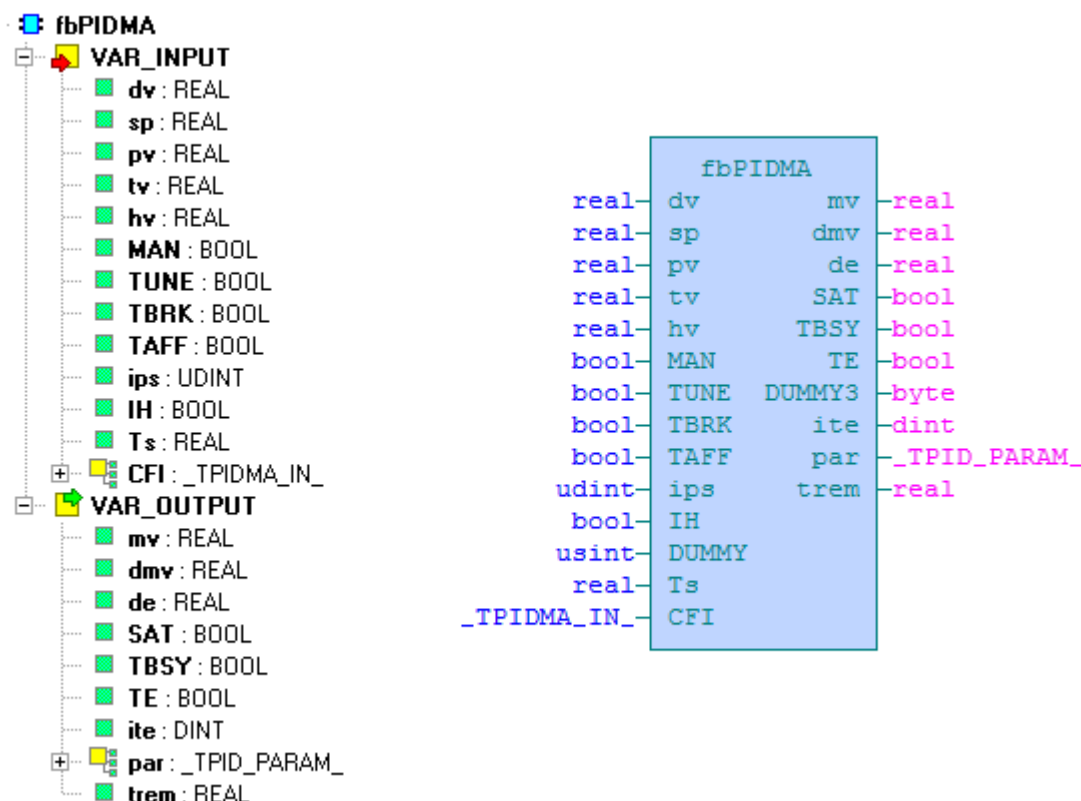
PROGRAM prgMain
VAR
  PIDU1 : fbPIDU;
  tof1  : TOF;
END_VAR

tof1(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDU1(sp := SetPoint, pv := AnalogInput, tv := PIDU1.mv,
      hv := ManualValue, MAN := tof1.Q OR Manual, Ts := 0.1,
      CFI := PIDU1_conf, mv => AnalogOutput);

IF NOT PIDU1.MAN THEN
  ManualValue := PIDU1.mv;
END_IF;
END_PROGRAM
```

4.2 Funkční blok fbPIDMA

PID regulátor s autotunerem

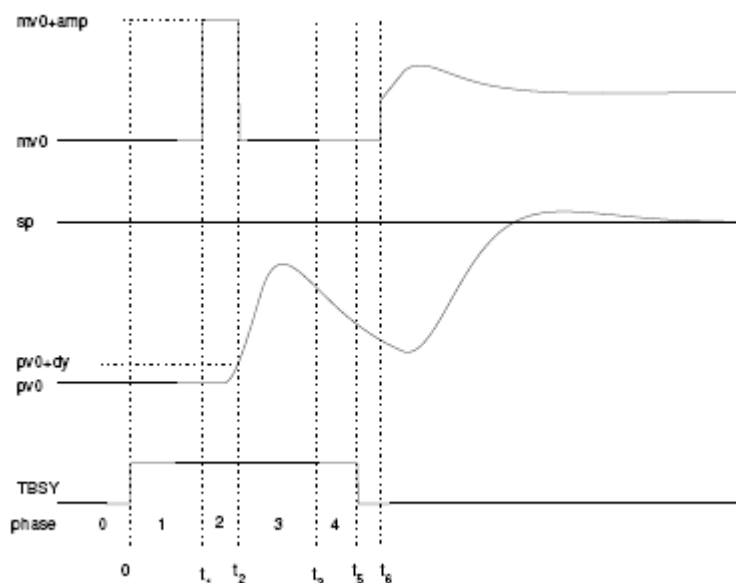


Blok *fbPIDMA* rozšiřuje řídicí funkci standardního PID regulátoru (viz 4.1 Funkční blok *fbPIDU*) o vestavěné automatické nastavování parametrů (PID autotuner). Před spuštěním autotuneru musí operátor ve vhodném pracovním bodě dosáhnout ustáleného stavu, zvolit požadovaný typ regulátoru *ittype* (*PI* nebo *PID*) a nastavit další parametry autotuneru (*iainf*, *DGC*, *tdg*, *tn*, *amp*, *dy* a *ispeed*). Identifikační experiment se startuje vstupem *TUNE* (vstupem *TBRK* jej lze předčasně ukončit). V tomto módu (*TBSY=1*) je nejprve odhadnut drift a šum regulované veličiny (ve specifikovaném čase $tdg+tn$) a poté je na vstup procesu aplikován pravoúhlý puls.

Z odezvy procesu jsou odhadnuty první tři momenty jeho impulsní odezvy. Amplituda pulsu se nastavuje parametrem *amp*. Puls je ukončen poté, co se hodnota regulované veličiny *pv* změní o více, než určuje tolerance (práh) *dy*. Pokud je nastaven příznak *DGC*, používá se při zpracování signálu speciální kompenzace trendu signálu. Odhad času zbývajících do konce procesu ladění je přiveden na výstup *trem*.

Pokud experiment skončí úspěšně ($TE=0$) a vstup $ips=0$, objeví se optimální parametry na výstupech *pk*, *pti*, *ptd*, *pnd*, *pb*, *c*. V opačném případě ($TE=1$) určuje výstup *ite* kód chyby experimentu. Další hodnoty vstupu *ips* jsou rezervovány pro speciální účely.

Funkce autotuneru je demonstrována na následujícím obrázku.

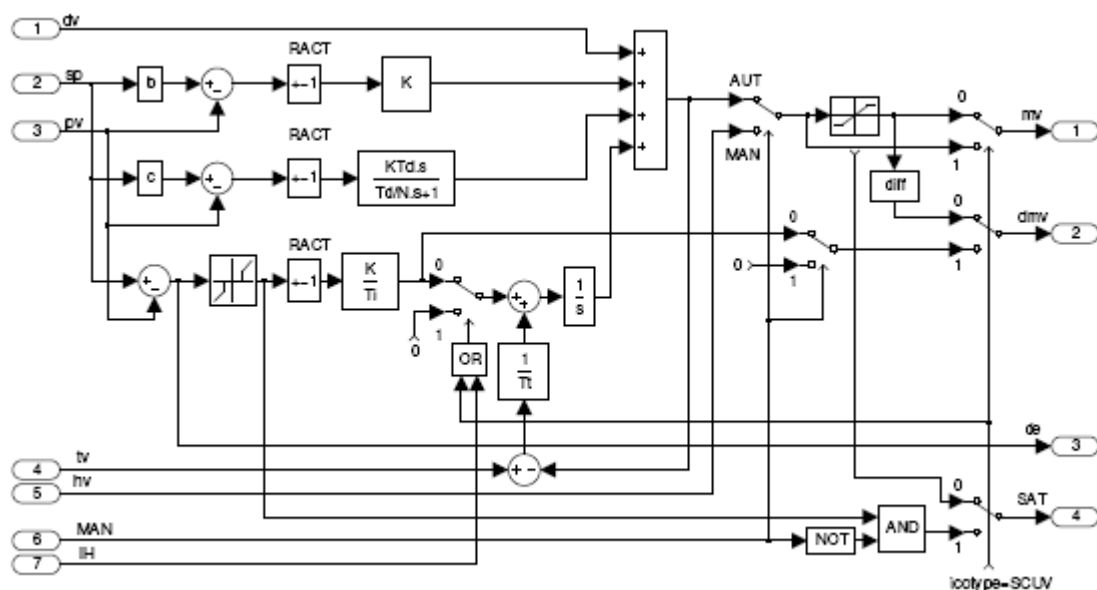


Během identifikačního experimentu výstup *ite* indikuje jednotlivé fáze činnosti autotuneru. Ve fázi odhadu strmosti odeznívání odezvy (*ite*=-4) může být proces ladění předčasně manuálně ukončen. V tomto případě jsou parametry regulátoru řádně navrženy, avšak jejich možná nepřesnost je indikována varovným kódem *ite*=100. Po ukončení experimentu (*TBSY* → 0) je funkce regulátoru závislá na nastaveném režimu (manuální, automatický). Jestliže *TAFF*=1, potom jsou navržené parametry okamžitě použity.

V automatickém režimu (*MAN*=0) realizuje funkční blok *fbPIDMA* shodný řídicí zákon PID regulátoru se dvěma stupni volnosti jako blok *fbPIDU*. Stejná pravidla platí i pro vstupy *dv*, *sp*, *pv*, *tv*, *hv*, *MAN*, *IH* a *Ts*, propojení s blokem *fbSCUV* (za účelem realizace krokového regulátoru bez polohové zpětné vazby) a kopírování hodnoty *hv* na výstup *mv* v manuálním režimu (viz 4.1 Funkční blok *fbPIDU*).











Výchozí propojení *fbPIDMA* a *fbSCUV* je realizováno v bloku *fbPIDMA_SCUV*.

Celková regulační funkce bloku *fbPIDMA* je zřejmá z následujícího obrázku.



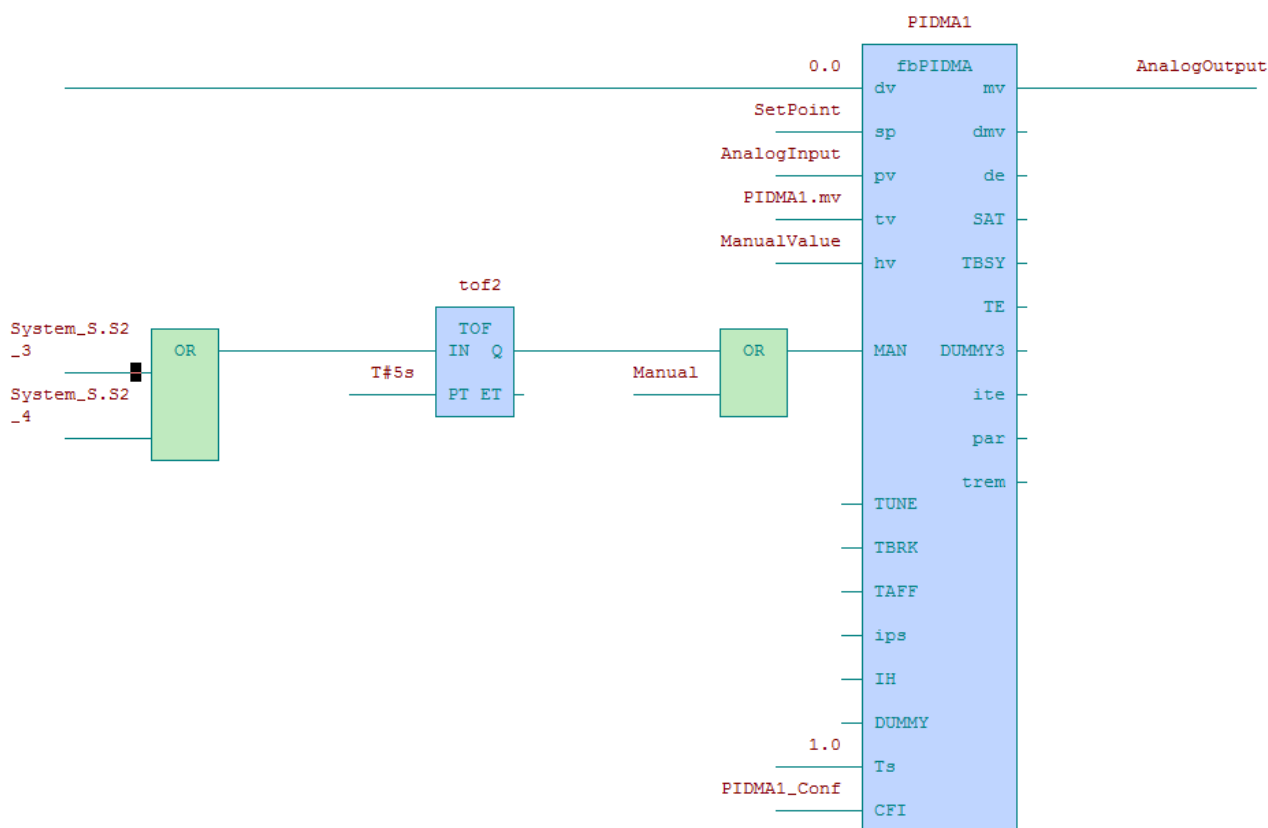
Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>dv</i>	REAL	dopředná vazba
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>tv</i>	REAL	vstup pro vysledování integrační složky
	<i>hv</i>	REAL	výstup regulátoru v manuálním režimu
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>IH</i>	REAL	pozastavení integrace 0 = integrace je povolena, 1 = integrace je pozastavena
	<i>TUNE</i>	BOOL	spuštění ladicího algoritmu (0 → 1)
	<i>TBRK</i>	BOOL	přerušování ladicího algoritmu
	<i>TAFF</i>	BOOL	příznak pro potvrzení parametrů; určuje způsob zacházení s novými parametry 0 ... parametry se do regulátoru nenastavují 1 ... parametry se nastaví do regulátoru ihned po ukončení jejich výpočtu 0 → 1 parametry se nastaví do regulátoru jednorázově při této změně

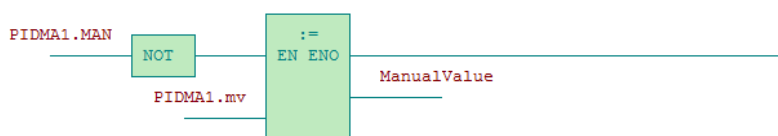
	Proměnná	Typ	Význam
	<i>ips</i>	UDINT	význam prvků výstupní struktury <i>par</i> (<i>pk</i> , <i>pti</i> , <i>ptd</i> , <i>pnd</i> , <i>pb</i> a <i>pc</i>) 0 ... navržené parametry: <i>k</i> , <i>ti</i> , <i>td</i> , <i>nd</i> , <i>b</i> a <i>c</i> PID regulátoru 1 ... momenty procesu: zesílení (<i>pk</i>), průměrná časová konstanta (<i>pti</i>), doba trvání odezvy (<i>ptd</i>) 2 ... tříparametrový model procesu: zesílení (<i>pk</i>), dopravní zpoždění (<i>pti</i>), časová konstanta (<i>ptd</i>) 3-5 ... slouží pro diagnostické účely
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	_TPIDMA_IN_	parametry regulátoru (viz 2.4 Typ _TPIDMA_IN_)
VAR_OUTPUT			
	<i>mv</i>	REAL	řídící veličina (polohový výstup regulátoru)
	<i>dmv</i>	REAL	rychlostní výstup regulátoru
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 ... regulátor pracuje v lineární oblasti 1 ... výstup regulátoru je saturován
	<i>TBSY</i>	BOOL	příznak ladicího módu (<i>TBSY=1</i>)
	<i>TE</i>	BOOL	příznak chyby ladění 0 ... bez chyby, 1 ... experiment selhal
	<i>ite</i>	DINT	specifikace chyby (po experimentu) 0 ... bez chyby 1 ... příliš malá hodnota prahu pro ukončení pulsu 2 ... příliš velká amplituda pulsu 3 ... nebylo dosaženo ustáleného stavu 4 ... příliš malá amplituda pulsu 5 ... selhání procedury hledání vrcholu 6 ... došlo k saturaci výstupu regulátoru při experimentu 7 ... pro vybraný typ regulátoru není podporováno automatické nastavování 8 ... nedodržena podmínka monotónnosti procesu 9 ... selhání extrapolace 10 ... neočekávané hodnoty momentů (fatální) 11 ... ruční přerušení experimentu uživatelem 12 ... nesprávný směr řídící veličiny (změňte parametr <i>RACT</i>) 100 ... ruční ukončení ladění (varování) 0 ... čekání na ustálený stav před začátkem experimentu -1 ... odhad driftu a šumu (parametry <i>tdg</i> a <i>tn</i>) -2 ... generování obdélníkového pulsu (puls končí při změně <i>pv</i> o hodnotu větší než <i>dy</i>) -3 ... hledání vrcholu odezvy -4 ... odhad rychlosti ustalování odezvy Poznámka: Náběžná hrana vstupu <i>TUNE</i> během fází -2, -3 -4 způsobuje ukončení dané fáze a přechod do fáze následující (nebo ukončení experimentu ve fázi -4).

	Proměnná	Typ	Význam
	<i>trem</i>	REAL	odhad času, zbývajícího do ukončení experimentu
	<i>pk</i>	REAL	navržené zesílení K regulátoru ($ips=0$)
	<i>pti</i>	REAL	navržená integrační časová konstanta T_i ($ips=0$)
	<i>ptd</i>	REAL	navržená derivační časová konstanta T_d ($ips=0$)
	<i>pnd</i>	REAL	navržený parametr N filtru derivační složky ($ips=0$)
	<i>pb</i>	REAL	navržený váhový koeficient požadované hodnoty -- proporcionální složka ($ips=0$)
	<i>pc</i>	REAL	navržený váhový koeficient požadované hodnoty -- derivační složka ($ips=0$)

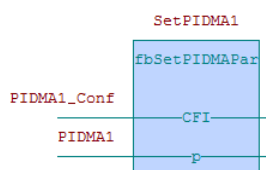
Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDMA* v jazyce FBD. Časovač TOF zajišťuje bezrázový start regulátoru po restartu.



Následující obvod zajišťuje bezrázový přechod z automatického do manuálního režimu:



Pro přenos parametrů navržených tunerem do konfigurační struktury, lze z výhodou využít blok *fbSetPIDMAPar* (viz 4.12 Funkční blok fbSetPIDMAPar) :



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogOutput – analogový výstup ovládající akční člen

SetPoint – proměnná s žádanou hodnotou

Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDMA1_Conf – struktura s parametry funkčního bloku

V jazyce ST předchozím zapojením odpovídá následující program.

```

VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
ManualValue : REAL;
PIDMA1_Conf : _TPIDMA_IN_ := (irtype:= irtype_PID,
                             par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                     nd:= 4.0, b:= 1.0, c:= 0.0), tt:= 1.0,
                             hilim:= 100.0, lolim:= 0.0,
                             dz:= 0.0, icotype:= icotype_ANALOG,
                             ittype:= ittype_PID, iainf:= 1,
                             DGC:= 1, tdg:= 10.0,
                             tn:= 5.0, amp:= 5.0, dy:= 1.0,
                             ispeed:= 2, ipid:= 1);
END_VAR

PROGRAM prgMain
VAR
  PIDMA1      : fbPIDMA;
  tof2        : TOF;
  SetPIDMA1   : fbSetPIDMAPar;
END_VAR

tof2(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDMA1(sp := SetPoint, pv := AnalogInput, tv := PIDMA1.mv,
        hv := ManualValue, MAN := tof2.Q OR Manual, Ts:=1.0,
        CFI:= PIDMA1_Conf, mv => AnalogOutput);

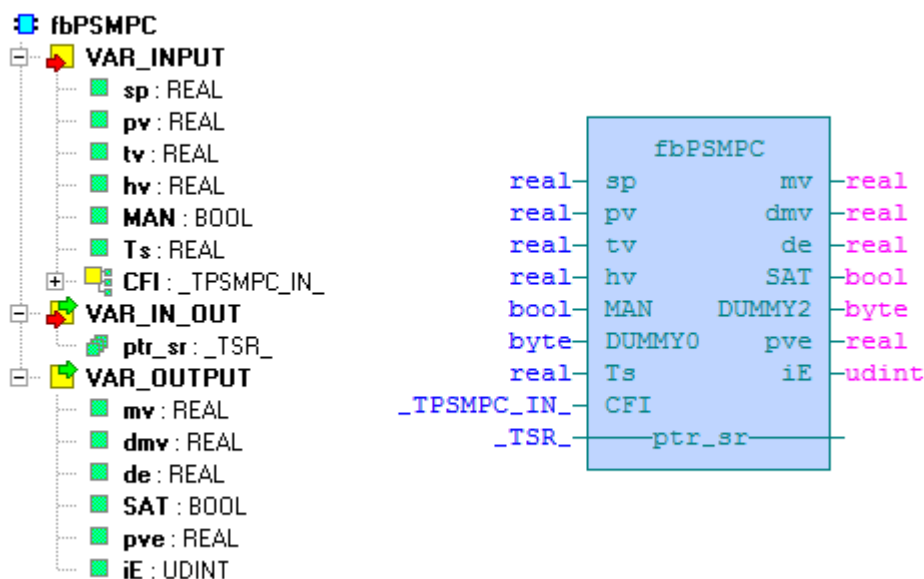
IF NOT PIDMA1.MAN THEN
  ManualValue := PIDMA1.mv;
END_IF;

SetPIDMA1(CFI := PIDMA1_Conf, p := PIDMA1);
END_PROGRAM

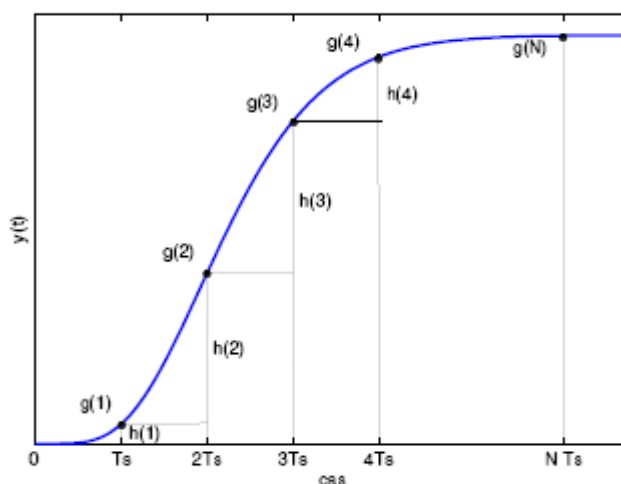
```

4.3 Funkční blok fbPSMPC

Prediktivní regulátor „pulse-step“



Funkční blok **fbPSMPC** (Pulse Step Model Predictive Control) je určen pro realizaci vysoce kvalitních regulátorů pro obtížně regulovatelné lineární časově invariantní soustavy s omezením akční veličiny (např. soustavy s dopravním zpožděním nebo s neminimální fází). Zvláště výhodný je pro případy, kdy je požadován velmi rychlý přechod z jedné hodnoty regulované veličiny na druhou bez překmitu. Regulátor *fbPSMPC* však může být obecně použit všude tam, kde je běžně nasazován standardní PID regulátor a kde žádáme vysokou kvalitu regulace.



Blok *fbPSMPC* je prediktivní regulátor s explicitně zadaným intervalovým omezením akční veličiny. Pro účely predikce je použit model ve tvaru diskrétní přechodové charakteristiky $g(j)$, $j=1, \dots, N$. Na obrázku výše je naznačen způsob, jakým lze tuto posloupnost získat ze spojitě přechodové charakteristiky. Poznamenejme, že N musí být zvoleno dostatečně velké, aby přechodová charakteristika byla popsána až do ustáleného stavu ($NT_s > t_{95}$, kde T_s je perioda vzorkování regulátoru a t_{95} je doba ustálení na 95% konečné hodnoty). Pro nasnímání přechodové

charakteristiky lze s výhodou využít blok *fbSampler* (viz 4.15 Funkční blok *fbSampler*). Pro systémy s monotónní přechodovou charakteristikou je alternativně možné použít momentový množinový model a popsat systém pouze třemi charakteristickými čísly κ (*kappa*), μ (*mu*) a σ^2 (*sigma*), které je možno určit z jednoduchého pulzního experimentu (viz 4.2 Funkční blok *fbPIDMA*) nebo nasnímáním přechodové charakteristiky blokem *fbSampler* (viz 4.15 Funkční blok *fbSampler*). Řízený systém pak aproximujeme buď přenosem prvního řádu s dopravním zpožděním

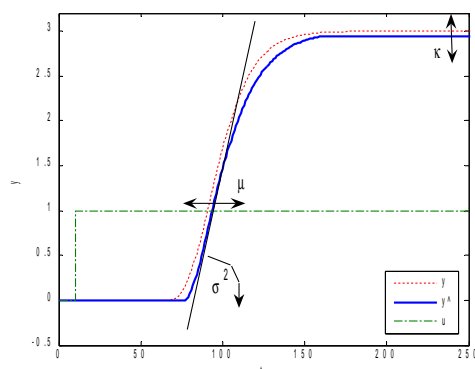
$$F_{FOPDT}(s) = \frac{K}{\tau \cdot s + 1} \cdot e^{-D \cdot s}, \kappa = K, \mu = \tau + D, \sigma^2 = \tau^2$$

nebo přenosem druhého řádu s dopravním zpožděním

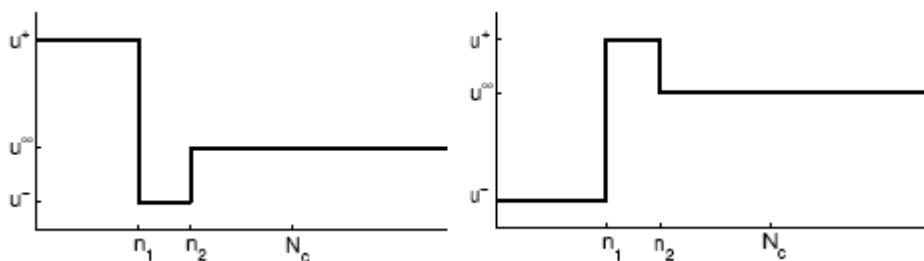
$$F_{SOPDT}(s) = \frac{K}{(\tau \cdot s + 1)^2} \cdot e^{-D \cdot s}, \kappa = K, \mu = 2 \cdot \tau + D, \sigma^2 = 2 \cdot \tau^2$$

se stejnými charakteristickými čísly. Typ aproximace se zadává parametrem *imtype*.

Následující obrázek ukazuje, že charakteristická čísla mají pro výše uvedené systémy jasný fyzikální význam, takže je možné je ručně doladit na základě prostého porovnání odezvy reálného systému a modelu na vstupní signál (např. skok). Charakteristické číslo κ je statické zesílení, číslo μ má charakter dopravního zpoždění (posouvá odezvu podél časové osy) a parametr σ^2 mění rychlost náběhu přechodové charakteristiky.



Pro zjednodušení on-line optimalizace v otevřené smyčce je množina přípustných posloupností řízení omezena pouze na posloupnosti ve tvaru "puls-skok" zobrazené na obrázku níže.



Poznamenejme, že každá takováto posloupnost je jednoznačně určena jen třemi čísly $n_1, n_2 \in (0, \dots, N_c)$ a $u^\infty \in \langle u^-, u^+ \rangle$, kde $N_c \in (0, 1, \dots)$ je horizont řízení a u^-, u^+ označují po řadě zadanou dolní a horní mez akční veličiny regulátoru. On-line optimalizace (vzhledem k n_1, n_2 a u^∞) spočívá v minimalizaci kritéria













$$I = \sum_{i=N_1}^{N_2} \hat{e}(k+i|k)^2 + \lambda \sum_{i=0}^{N_C} \Delta \hat{u}(k+i|k)^2 \rightarrow \min$$



kde $\hat{e}(k+i|k)^2$ je v kroku k predikovaná regulační odchylka na intervalu predikce $i \in (N_1, N_2)$, $\Delta \hat{u}(k+i|k)^2$ je příslušná diference řízení na intervalu $i \in \{0, N_C\}$ a λ je koeficient penalizace změn akční veličiny. Realizovaná (zpětnovazební) posloupnost řízení na posloupnost typu "puls-skok" omezená ovšem není, neboť v každém kroku je použit pouze první člen optimální posloupnosti. Parametry prediktivního regulátoru, kromě modelu řízené soustavy a omezení jejího vstupu, jsou horizont řízení N_C , horizont predikce N_1 , N_2 a koeficient λ . Pouze poslední uvedený parametr je určen pro ruční doladění kvality regulace při rutinním uvádění do provozu. V případě použití modelu soustavy ve tvaru přenosu F_{FOPDT} nebo F_{SOPDT} jsou parametry N_1 , N_2 zvoleny automaticky na základě charakteristických čísel μ a σ^2 . Regulátor potom může být efektivně laděn "ručně" pouze seřizováním charakteristických čísel procesu.

Důležité:

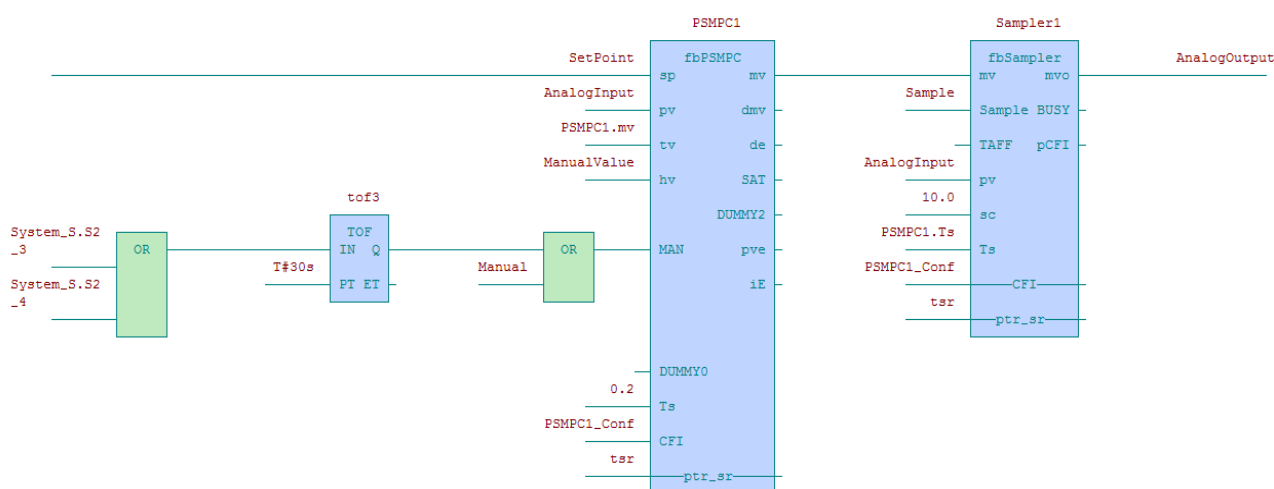
Parametr horizont řízení N_C významně ovlivňuje dobu vykonávání algoritmu řízení a jeho hodnota by typicky neměla přesáhnout 5. S náročností algoritmu je třeba počítat i při volbě vzorkovací periody, která musí být delší než čas vykonávání algoritmu. Typická doba vykonávání jedné regulační smyčky na centrále CPM-K při N_C menší než 5 se pohybuje okolo 100 ms. Doporučujeme volit periodu vzorkování minimálně dvojnásobnou.

Popis proměnných :

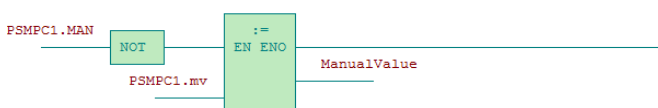
	Proměnná	Typ	Význam
VAR_INPUT			
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>tv</i>	REAL	vstup pro vysledování (realizovaná akční veličina)
	<i>hv</i>	REAL	výstup regulátoru v manuálním režimu
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<code>_TPSMPC_IN_</code>	parametry regulátoru (viz 2.6 Typ <code>_TPSMPC_IN_</code>)
VAR_OUTPUT			
	<i>mv</i>	REAL	řídící veličina (polohový výstup regulátoru)
	<i>dmv</i>	REAL	rychlostní výstup regulátoru
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován
	<i>pve</i>	REAL	predikovaná hodnota regulované veličiny na základě množinového modelu

	Proměnná	Typ	Význam
	 <i>iE</i>	UDINT	číslo chyby 0 ... bez chyby 1 ... chyba v zadání modelu FOPDT 2 ... chyba v zadání modelu SOPDT 3 ... chyba v zadání přechodové charakteristiky 4 ... vnitřní buffer v bloku fbPSMPC není dost velký pro zadaný model (je nutné zvětšit periodu vzorkování)
VAR_IN_OUT			
	 <i>ptr_sr</i>	_TSR_	pole pro předání diskretní přechodové charakteristiky

Následující příklady ukazují základní zapojení funkčního bloku *fbPSMPC* v jazyce FBD.



Následující obvod zajišťuje bezrázový přechod z automatického do manuálního režimu:



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogOutput – analogový výstup ovládající akční člen

SetPoint – proměnná s žádanou hodnotou

Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PSMPC1_Conf – struktura s parametry funkčního bloku

tsr – pole s přechodovou charakteristikou řízeného procesu

Sample – příznak snímání přechodové charakteristiky

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL
  Sample      : BOOL;
END_VAR

VAR_GLOBAL RETAIN
  Manual      : BOOL;
  SetPoint    : REAL;
  ManualValue : REAL;

  PSMPC1_Conf : _TPSMPC_IN_ := (nc := 3);
  tsr : _TSR_;
END_VAR

PROGRAM prgMain
  VAR
    tof3 : TOF;
    PSMPC1 : fbPSMPC;
    Sampler1 : fbSampler;
  END_VAR

  tof3(IN := System_S.S2_3 OR System_S.S2_4, PT := T#30s);

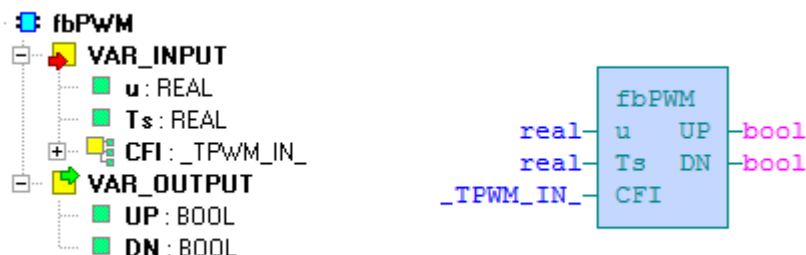
  PSMPC1(sp := SetPoint, pv := AnalogInput, tv := PSMPC1.mv,
    hv := ManualValue, MAN := tof3.Q OR Manual, Ts := 0.2,
    CFI:= PSMPC1_Conf, ptr_sr:= tsr, mv => Sampler1.mv);

  Sampler1 (Sample := Sample, pv := AnalogInput, sc := 10.0,
    Ts := PSMPC1.Ts, CFI := PSMPC1_Conf, ptr_sr := tsr,
    mvo => AnalogOutput);

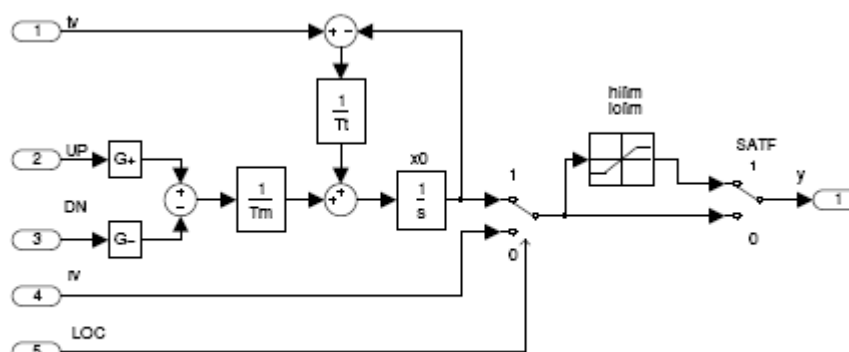
  IF NOT PSMPC1.MAN THEN
    ManualValue := PSMPC1.mv;
  END_IF;
END_PROGRAM
```

4.4 Funkční blok fbMCU

Blok pro ruční zadávání hodnot



V lokálním režimu ($LOC=1$) je funkční blok **fbMCU** určen k ručnímu zadávání výstupu y pomocí tlačítek „více“ (vstup UP) a „méně“ (vstup DN). Strmost najíždění z počáteční hodnoty y_0 na žádanou hodnotu je určena integrační konstantou tm a dobou stlačení ovládacích tlačítek. Po uplynutí každých ta sekund je strmost vždy násobena faktorem q , až do vypršení doby tf . Rozsah výstupu y může být omezen ($SATF=1$) saturačními mezemi $lolim$ a $hilim$. V případě, že žádné z tlačítek není stlačeno ($UP=0$ a $DN=0$), vysleduje výstup y vstupní hodnotu tv . Rychlost výsledování je dána integrační časovou konstantou tt . V případě $LOC=0$ je vstup rv s případnými omezeními ($SATF=1$) kopírován na výstup y . Podrobná funkce bloku je přímo patrná z obrázku, zobrazujícího vnitřní schéma bloku.

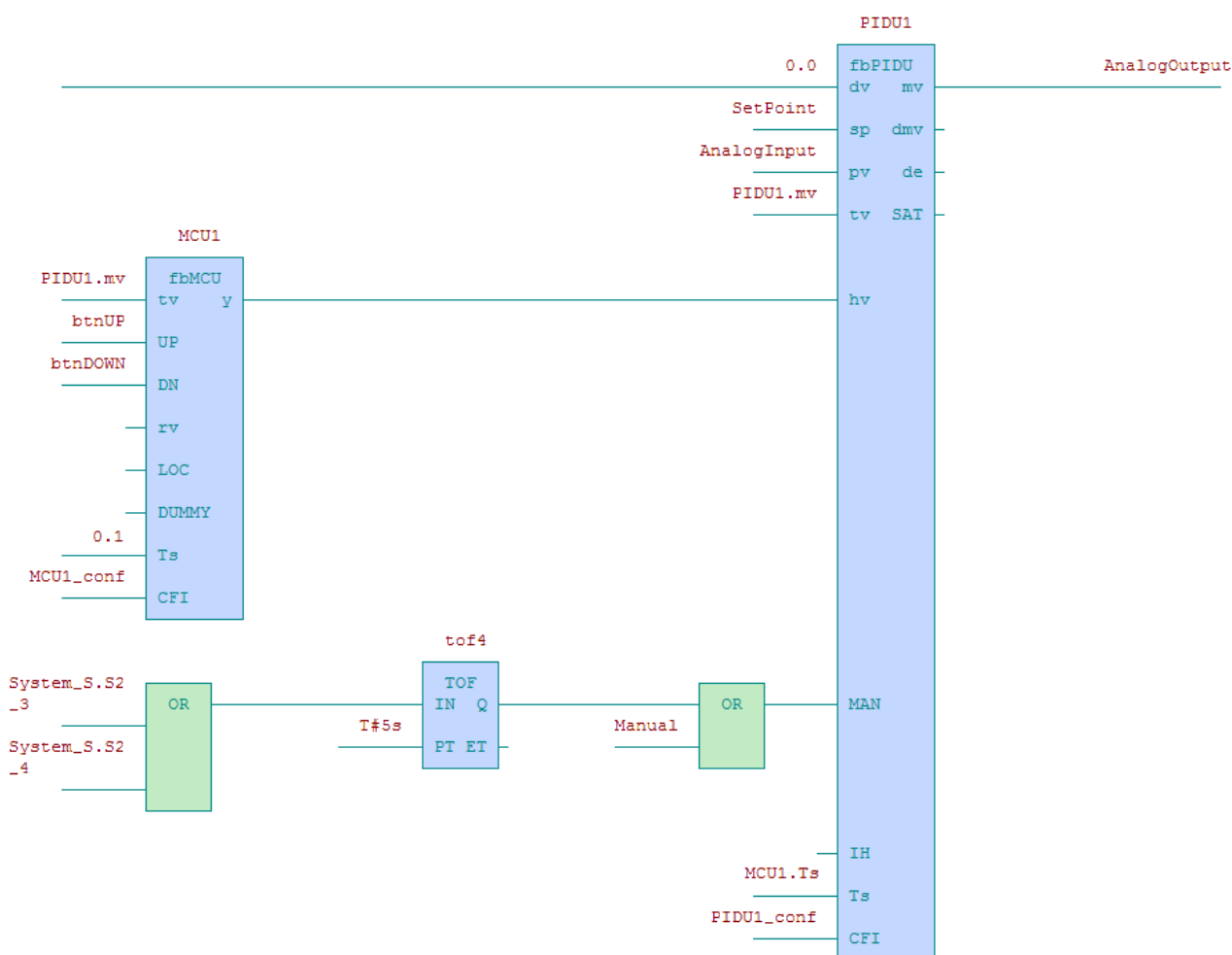


Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	tv	REAL	vstup pro výsledování integrační složky
	UP	BOOL	„více“ - najíždění nahoru s přednastavenou strmostí
	DN	BOOL	„méně“ - sjíždění dolů s přednastavenou strmostí
	rv	REAL	hodnota pro externí zadávání výstupu v režimu $LOC=0$
	LOC	BOOL	režim činnosti zadávací jednotky
	Ts	REAL	vzorkovací perioda regulátoru v sekundách
	CFI	$_TMCU_IN_$	parametry bloku (viz 2.7 Typ $_TMCU_IN_$)
VAR_OUTPUT			

	Proměnná	Typ	Význam
➡	<i>mv</i>	REAL	řídící veličina (polohový výstup regulátoru)
➡	<i>dmv</i>	REAL	rychlostní výstup regulátoru
➡	<i>de</i>	REAL	regulační odchylka
➡	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován

Následující příklad ukazuje základní zapojení funkčního bloku *fbMCU* v jazyce FBD.



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogOutput – analogový výstup ovládající akční člen

btnUP – binární vstup pro tlačítko zvyšující žádanou hodnotu

btnDOWN – binární vstup pro tlačítko snižující žádanou hodnotu

SetPoint – proměnná s žádanou hodnotou

Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU1_Conf – struktura s parametry funkčního bloku

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;

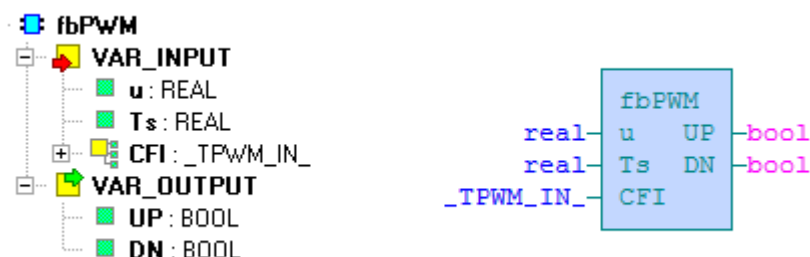
PIDU1_conf  : _TPIDU_IN_ := (irtype:= irtype_PID,
                             par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                       nd:= 10.0, b:= 1.0, c:= 0.0),
                             tt:= 1.0, hilim:= 100.0, lolim:= 0.0,
                             dz:= 0.0, icotype:= icotype_ANALOG);
MCU1_conf   : _TMCU_IN_ := (tm:= 2.0, q:= 2.0, ta:= 4.0, tf:= 12.0,
                             hilim := 100.0, lolim:= 0.0);
END_VAR

PROGRAM prgMain
VAR
  tof4 : TOF;
  PIDU1 : fbPIDU;
  MCU1 : fbMCU;
END_VAR

tof4(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
MCU1(tv := PIDU1.mv, UP := btnUP, DN := btnDOWN, rv := MCU1.rv,
     Ts:= 0.1,CFI:=MCU1_conf);
PIDU1(sp := SetPoint, pv := AnalogInput, tv := PIDU1.mv,
     hv := MCU1.y, MAN := tof4.Q OR Manual, Ts := MCU1.Ts,
     CFI := PIDU1_conf, mv => AnalogOutput);
END_PROGRAM
```

4.5 Funkční blok fbPWM

Pulzně šířková modulace



Funkční blok **fbPWM** provádí pulzně šířkovou modulaci vstupního signálu z intervalu od -1 do +1. Užitím tohoto bloku je možné realizovat proporcionální akční veličinu i u akčních členů s jedním (např. topení zapnuto/vypnuto) nebo dvěma (např. topení zapnuto/vypnuto a chlazení zap./vyp.) binárními vstupy. Šířka L vstupního pulsu je určena vztahem:

$$L = pertm \cdot |u| ,$$

kde $pertm$ je perioda modulace. Je-li $u > 0$ (resp. $u < 0$), puls je generován na výstupu UP (resp. DN). Z praktických důvodů je však délka generovaného pulsu dále upravována podle zadaných parametrů bloku. Faktor asymetrie $asyfac$ definuje poměr mezi délkou negativního pulsu DN a délkou pozitivního pulsu UP . Modifikované délky se počítají podle vztahů:

$$\text{jestliže } u > 0 \text{ potom } L(UP) = \begin{cases} L & \text{pro } asyfac \leq 1.0 \\ L/asyfac & \text{pro } asyfac > 1.0 \end{cases}$$

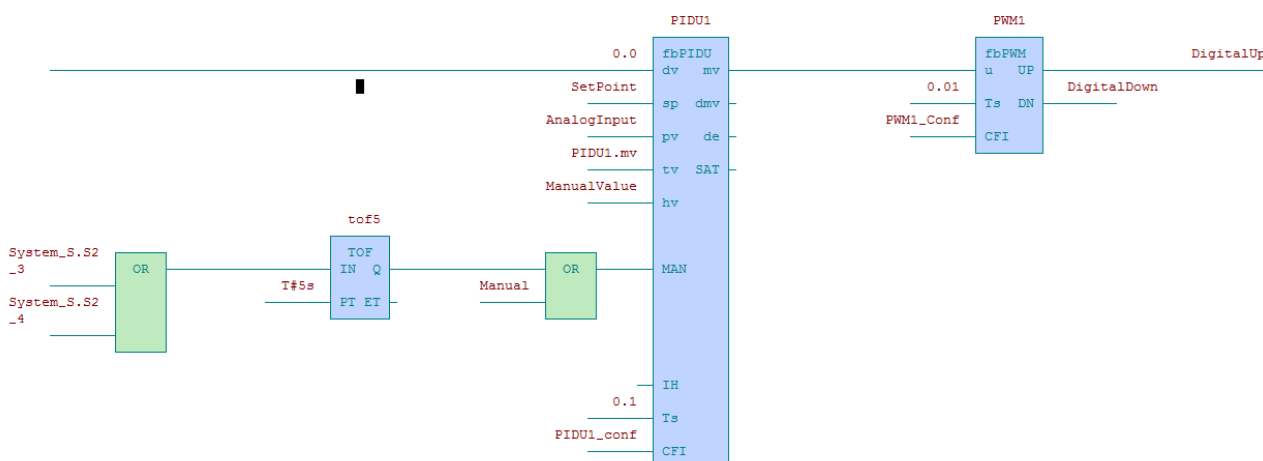
$$\text{jestliže } u < 0 \text{ potom } L(DN) = \begin{cases} L \cdot asyfac & \text{pro } asyfac \leq 1.0 \\ L & \text{pro } asyfac > 1.0 \end{cases}$$

kteří pro libovolnou hodnotu $asyfac > 0$ zajišťují, že maximální délka generovaných pulsů je rovna $pertm$. Dále, jestliže vypočtená délka pulsu je menší než $dtime$, potom je výsledná délka nastavena na nulu. Jestliže se vypočtená délka pulsu liší od $pertm$ méně než $btime$, potom je výsledná délka nastavena na $pertm$. Jestliže kladný puls UP je následovaný záporným pulsem DN nebo obráceně, potom pozdější puls je v případě potřeby posunut tak, že vzdálenost mezi těmito dvěma pulsy je alespoň $offtime$. Jestliže $SYNCH=1$, potom změna vstupu u způsobí okamžitý přepočtení délky výstupního pulsu za předpokladu, že není splněna synchronizační podmínka mezi začátkem periody modulace a okamžikem změny vstupu u .

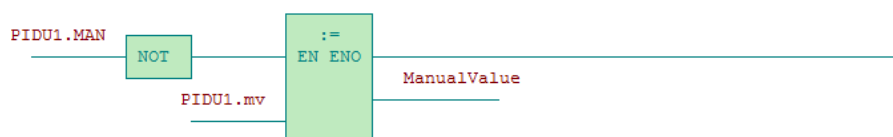
Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	u	REAL	vstupní hodnota
	T_s	REAL	vzorkovací perioda regulátoru v sekundách
	CFI	_TPWM_IN_	parametry (viz 2.8 Typ _TPWM_IN_)
VAR_OUTPUT			
	UP	BOOL	výstupní puls „nahoru“
	DN	BOOL	výstupní puls „dolů“

Následující příklad ukazuje základní zapojení funkčního bloku *fbPWM* v jazyce FBD.



Následující obvod zajišťuje bezrázový přechod z automatického do manuálního režimu:



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

SetPoint – proměnná s žádanou hodnotou

Manual – příznak ručního režimu

ManualValue – hodnota výstupu v ručním režimu

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU1_Conf – struktura s parametry funkčního bloku

PWM1_Conf – struktura s parametry šířkové modulace

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
ManualValue : REAL;
PIDU1_conf  : TPIDU_IN := (irtype:= irtype_PID,
                           par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                   nd:= 10.0, b:= 1.0, c:= 0.0),
                           tt:= 1.0, hilim:= 100.0, lolim:= 0.0,
                           dz:= 0.0, icotype:= icotype_ANALOG);

PWM1_conf   : TPWM_IN := (pertm:= 1.0);
END_VAR

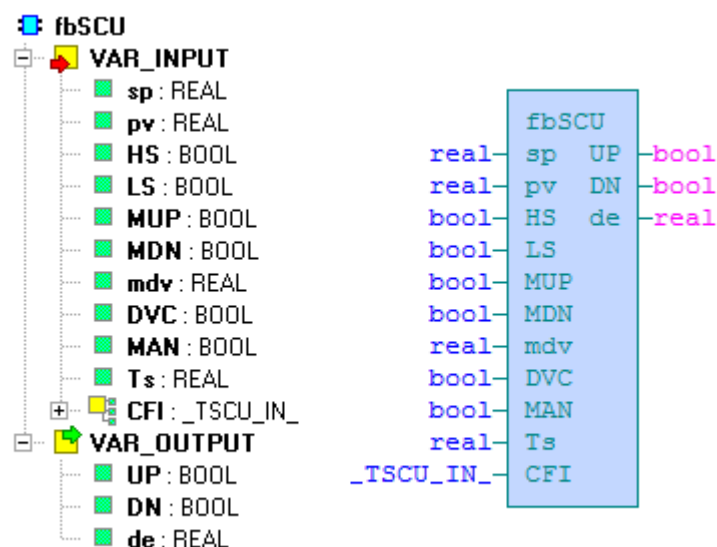
PROGRAM prgMain
VAR
  tof5      : TOF;
  PIDU1     : fbPIDU;
  PWM1     : fbPWM;
END_VAR

tof5(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDU1(sp := SetPoint, pv := AnalogInput, tv := PIDU1.mv,
      hv := ManualValue, MAN := tof5.Q OR Manual, Ts := 0.1,
      CFI := PIDU1_conf);
PWM1(u := PIDU1.mv, Ts := 0.01, CFI := PWM1_conf,
     UP => DigitalUp, DN => DigitalDown);

IF NOT PIDU1.MAN THEN
  ManualValue := PIDU1.mv;
END_IF;
END_PROGRAM
```


4.6 Funkční blok fbSCU

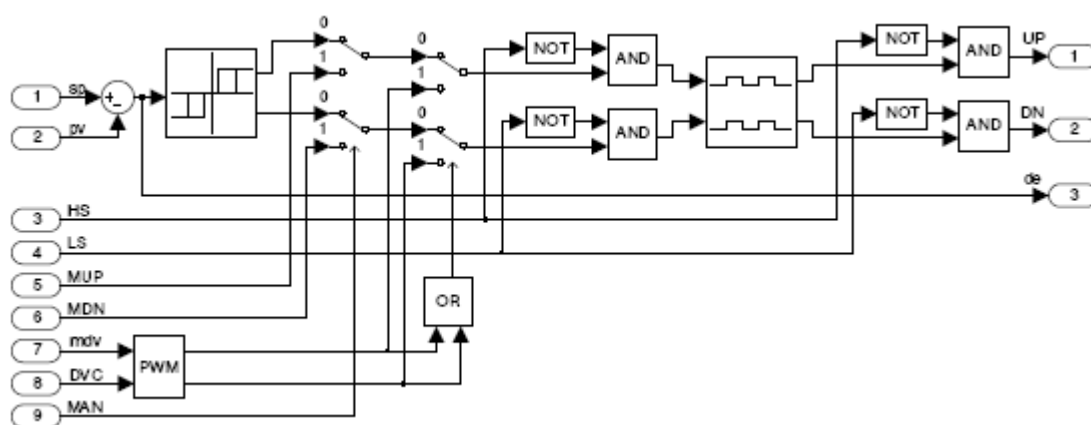
Krokový regulátor s polohovou zpětnou vazbou

















Funkční blok **fbSCU** je polohový regulátor servoventilu s třístavovým výstupem. Ve spojení s nadřazeným blokem *fbPIDU* (nebo *fbPIDMA*, atd.) je určen k realizaci třístavového krokového regulátoru s polohovou zpětnou vazbou. Toto spojení realizují bloky *fbPIDU_SCU* a *fbPIDMA_SCU*.

Blok *fbSCU* nejprve zpracovává regulační odchylku $sp - pv$ na třístavový výstup symetrickým třístavovým algoritmem (STA) s parametry (práhy) *thron* a *throff*. Přitom parametr *RACT* určuje, pro kterou polaritu odchylky jsou generovány pulsy *UP* (více) nebo *DN* (méně). Výstupy symetrického třístavového algoritmu jsou dále zpracovávány tak, že délka libovolného generovaného pulsu (*UP*, *DN*) na výstupu bloku je alespoň *dtime* a prodleva mezi dvěma po sobě jdoucími pulsy opačné polarity je alespoň *btime*. Jsou-li dostupné signály od koncových spínačů servoventilu, potom by měly být připojeny na vstupy *HS* (horní spínač) a *LS* (dolní spínač).

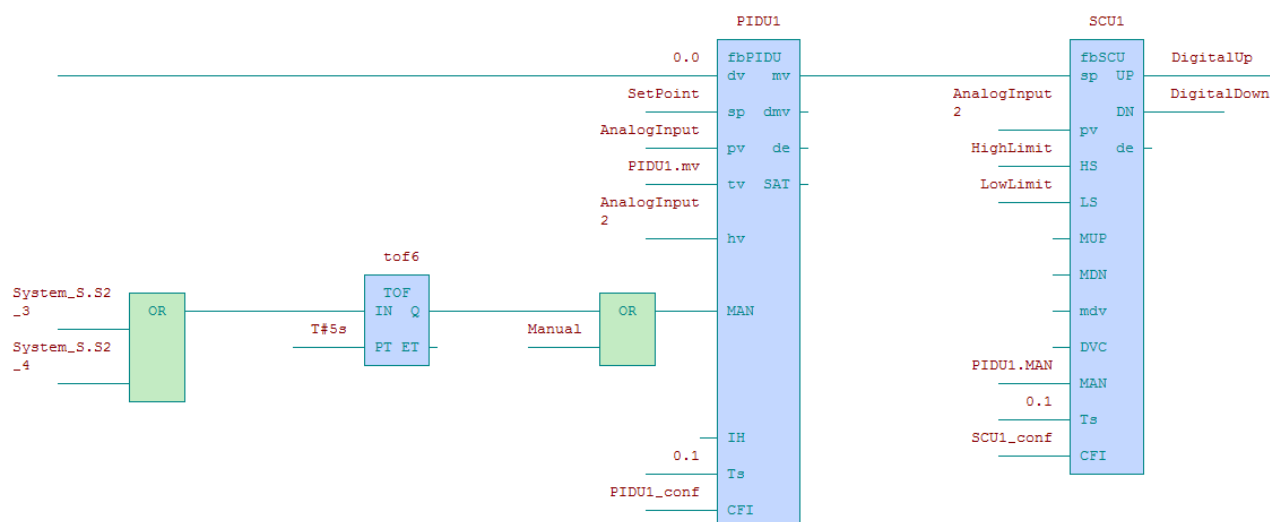
Celková funkce bloku *fbSCU* je zřejmá z následujícího obrázku.



Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>sp</i>	REAL	požadovaná hodnota (výstup primárního regulátoru)
	<i>pv</i>	REAL	řízená veličina (poloha servopohonu ventilu)
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0->1)
	<i>MAN</i>	BOOL	režim funkce regulátoru <i>fbSCU</i> 0 = automatický režim, 1 = manuální režim
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<u>_TSCU_IN_</u>	parametry regulátoru (viz 2.9 Typ <u>_TSCU_IN_</u>)
VAR_OUTPUT			
	<i>UP</i>	BOOL	výstupní signál „nahoru“
	<i>DN</i>	BOOL	výstupní signál „dolů“
	<i>de</i>	REAL	regulační odchylka

Následující příklad ukazuje základní zapojení funkčního bloku *fbSCU* s regulátorem *fbPIDU* v jazyce FBD.



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogInput2 – analogový vstup s polohou akčního členu

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

HighLimit – horní omezení akčního členu

LowLimit – dolní omezení akčního členu

SetPoint – proměnná s žádanou hodnotou

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU1_Conf – struktura s parametry funkčního bloku

SCU1_Conf – struktura s parametry trojstavového řízení

V jazyce ST tomuto zapojení odpovídá následující program.

```

VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDU1_conf  : _TPIDU_IN_ := (irtype:= irtype_PID,
                             par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                       nd:= 10.0, b:= 1.0, c:= 0.0),
                             tt:= 1.0, hilim:= 100.0, lolim:= 0.0,
                             dz:= 0.0, icotype:= icotype_SCU);
SCU1_conf   : _TSCU_IN_ := (thron:= 2.0, throff:= 0.5, dtime:= 0.1,
                             btime:= 0.2, trun:= 10.0);
END_VAR

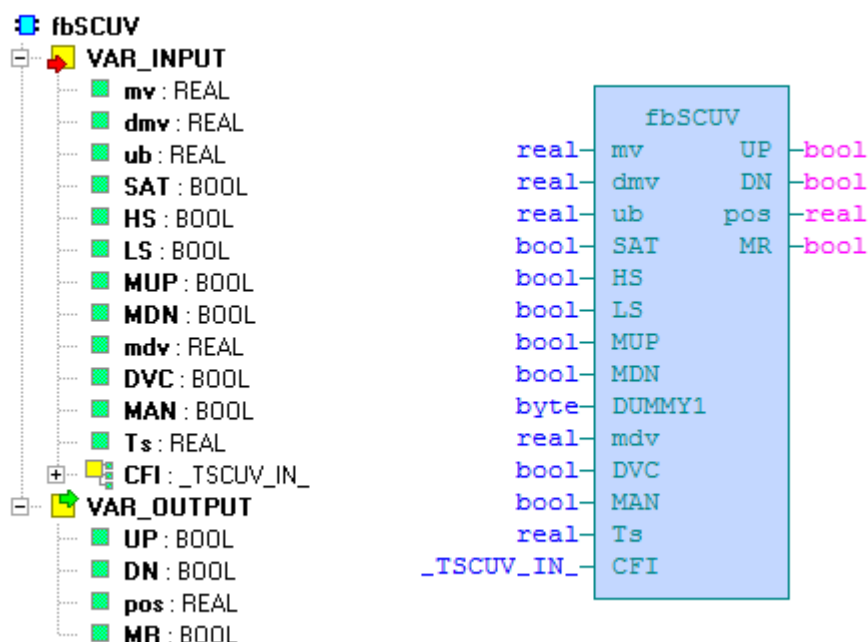
PROGRAM prgMain
VAR
  tof6 : TOF;
  PIDU1 : fbPIDU;
  SCU1 : fbSCU;
END_VAR

tof6(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDU1(sp := SetPoint, pv := AnalogInput, tv := PIDU1.mv,
      hv := AnalogInput2, MAN := tof6.Q OR Manual,
      Ts := 0.1, CFI := PIDU1_conf);
SCU1(sp := PIDU1.mv, pv := AnalogInput2,
     HS := HighLimit, LS := LowLimit,
     MAN := PIDU1.MAN, Ts := 0.1, CFI := SCU1_conf,
     UP => DigitalUp, DN => DigitalDown);
END_PROGRAM

```

4.7 Funkční blok fbSCUV

Krokový regulátor s rychlostním vstupem



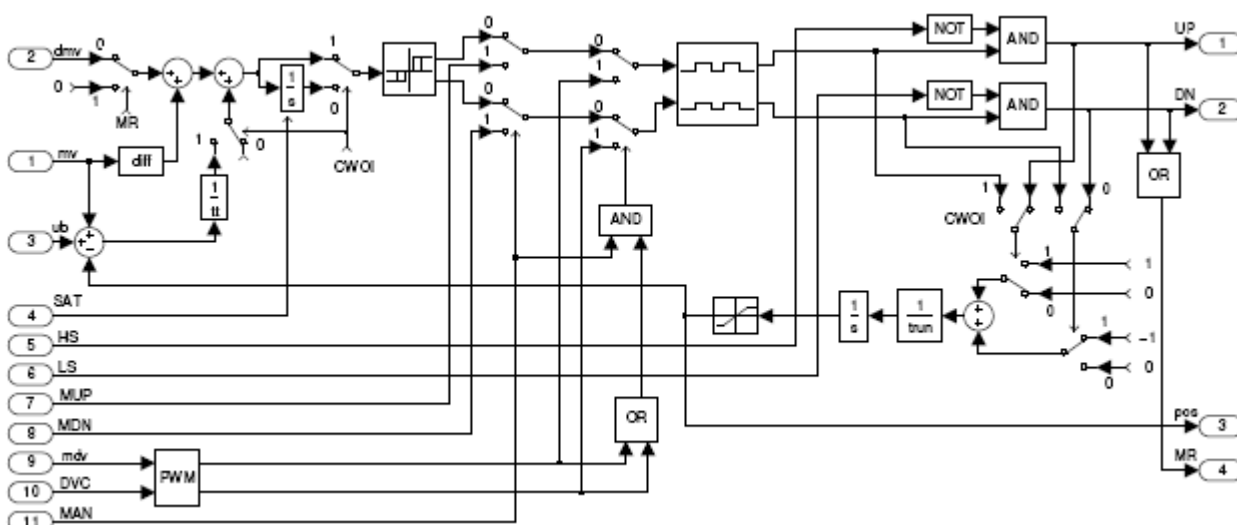
Funkční blok **fbSCUV** nahrazuje polohový regulátor *fbSCU* v úplné regulační smyčce s třístavovým krokovým regulátorem, jestliže polohový signál servoventilu není dostupný anebo dostatečně spolehlivý. Nadřazený regulátor *fbPIDU* (nebo *fbPIDMA*) je propojen s funkčním blokem *fbSCUV* signály *mv*, *dmv* a *SAT* (výstupy bloku *fbPIDU* a vstupy bloku *fbSCUV*). Toto spojení realizují bloky *fbPIDU_SCUV* a *fbPIDMA_SCUV*.

Jestliže je nadřazený regulátor typu *PI* nebo *PID* ($CWOI=0$), potom jsou všechny tři vstupy *mv*, *dmv* a *SAT* bloku *fbSCUV* zpracovávány speciálním integračním algoritmem a symetrickým třístavovým algoritmem s parametry (prahy) *thron* a *throff*. Vzniklé pulsy (více, méně) jsou dále upravovány tak, že délka libovolného generovaného pulsu (*UP*, *DN*) na výstupu bloku je alespoň *dtime* a prodleva mezi dvěma po sobě jdoucími pulsy opačné polarity je alespoň *btime*. Parametr *RACT* určuje směr otáčení motoru. Poznamenejme, že nadřazený regulátor *fbPIDU* musí mít nastavení *icotype=4*. Blok *fbSCUV* rekonstruuje rychlostní výstup nadřazeného regulátoru ze vstupů *mv* a *dmv*. Navíc, jestliže regulační odchylka nadřazeného regulátoru je menší než pásmo necitlivosti ($SAT=1$), potom je výstup vnitřního integrátoru bloku *fbSCUV* nulován. Takto je dosaženo klidu servoventilu při dostatečně malé regulační odchylce nadřazeného regulátoru ($|de| < dz$ – viz kapitola 4.1 Funkční blok *fbPIDU*).

Poloha servoventilu *pos* je odhadována dalším vnitřním integrátorem s časovou konstantou *trun*. Jsou-li dostupné signály od koncových spínačů servoventilu, potom by měly být připojeny na vstupy *HS* (horní spínač) a *LS* (dolní spínač).





Jestliže je nadřazený regulátor typu *P* nebo *PD* ($CWOI=1$), potom může být regulační odchylka nadřazeného regulátoru manuálně odstraněna vhodným nastavením vstupu *ub*. V tomto případě je řídicí algoritmus bloku *fbSCUV* lehce modifikován. Je užita rekonstruovaná hodnota polohy servoventilu *pos* a parametry *thron*, *throff* a *tt* musí být pečlivě nastaveny pro potlačení střídání pulsů více a méně v ustáleném stavu.

Celková regulační funkce bloku *fbSCUV* je zřejmá z následujícího obrázku.

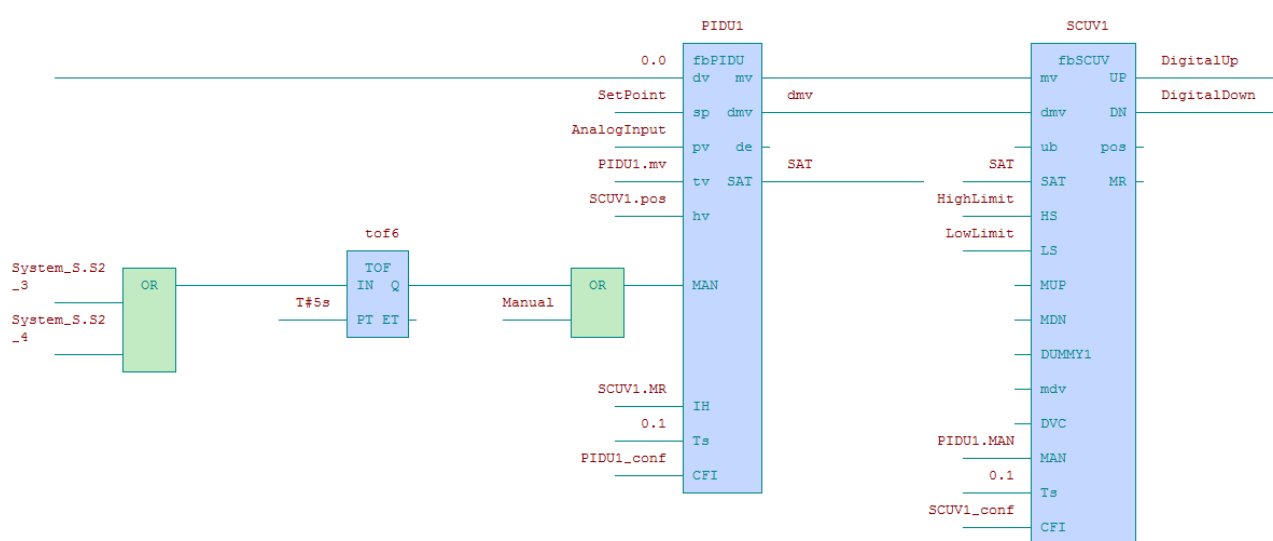


Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>mv</i>	REAL	polohový výstup primárního regulátoru
	<i>dmv</i>	REAL	rychlostní výstup primárního regulátoru
	<i>ub</i>	REAL	posunutí (jen v případě, že primární regulátor je typu <i>P</i> nebo <i>PD</i>)
	<i>SAT</i>	BOOL	nulování interního integrátoru (propojen s výstupem <i>SAT</i> primárního regulátoru)
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0->1)
	<i>MAN</i>	BOOL	režim funkce regulátoru <i>fbSCU</i> 0 = automatický režim, 1 = manuální režim
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<code>_TSCUV_IN_</code>	parametry regulátoru (viz 2.10 Typ <code>_TSCUV_IN_</code>)

	Proměnná	Typ	Význam
VAR_OUTPUT			
	UP	BOOL	výstupní signál „nahoru“
	DN	BOOL	výstupní signál „dolů“
	pos	REAL	simulovaná poloha motoru
	MR	BOOL	požadavek na běh motoru 0 = motor nemá běžet ($UP=0$ AND $DN=0$) 1 = motor se má pohybovat ($UP=1$ OR $DN=1$)

Následující příklad ukazuje základní zapojení funkčního bloku *fbSCUV* s regulátorem *fbPIDU* v jazyce FBD.



Význam jednotlivých proměnných:

- AnalogInput* – analogový vstup s regulovanou veličinou
- dmv* – pomocná proměnná pro propojení funkčních bloků
- SAT* – pomocná proměnná pro propojení funkčních bloků
- DigitalUp* – binární výstup pro kladný akční zásah
- DigitalDown* – binární výstup pro záporný akční zásah
- HighLimit* – horní omezení akčního členu
- LowLimit* – dolní omezení akčního členu
- SetPoint* – proměnná s žádanou hodnotou
- System_S.S2_3* – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib
- System_S.S2_4* – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib
- PIDU1_Conf* – struktura s parametry funkčního bloku
- SCUV1_Conf* – struktura s parametry trojstavového řízení

V jazyce ST tomuto zapojení odpovídá následující program.

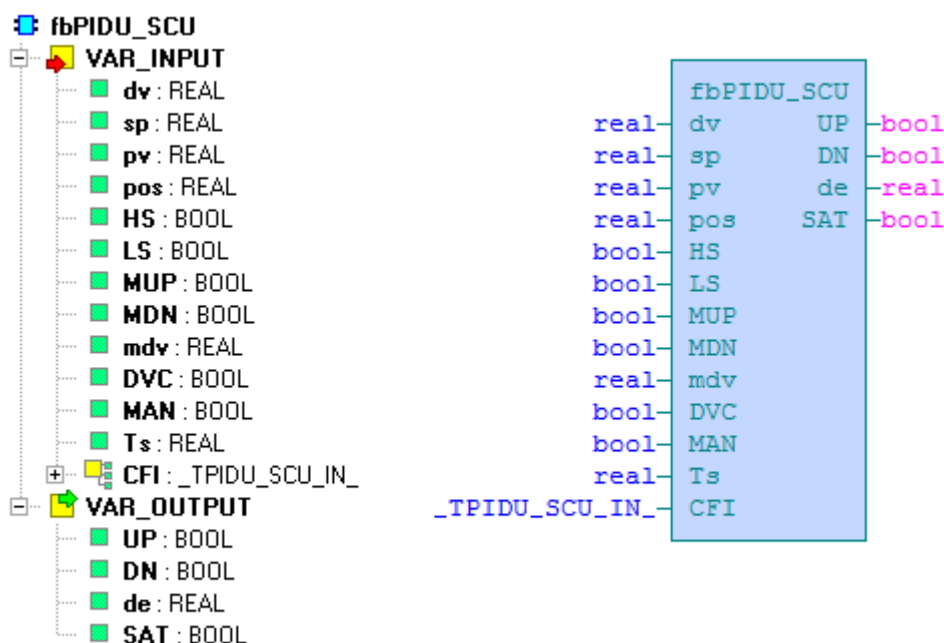
```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDU1_conf  : _TPIDU_IN_ := (irtype:= irtype_PID,
                             par := (k:= 0.75, ti:= 8.0, td:= 1.0,
                                       nd:= 10.0, b:= 1.0, c:= 0.0),
                             tt:= 1.0, hilim:= 100.0, lolim:= 0.0,
                             dz:= 0.0, icotype:= icotype_SCUV);
SCUV1_conf  : _TSCUV_IN_ := (thron := 2.0, throff := 1.0, dtime:= 0.1,
                             btime:= 0.2, trun:= 10.0);
END_VAR

PROGRAM prgMain
VAR
  tof6 : TOF;
  PIDU1 : fbPIDU;
  SCUV1 : fbSCUV;
END_VAR

tof6(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDU1(sp := SetPoint, pv := AnalogInput, tv := PIDU1.mv,
      IH := SCUV1.MR, hv := SCUV1.pos,
      MAN := tof6.Q OR Manual, Ts := 0.1, CFI := PIDU1_conf);
SCUV1(mv := PIDU1.mv, dmv := PIDU1.dmv,
      HS := HighLimit, LS := LowLimit, SAT := PIDU1.SAT,
      MAN := PIDU1.MAN, Ts := 0.1, CFI := SCUV1_conf,
      UP => DigitalUp, DN => DigitalDown);
END_PROGRAM
```

4.8 Funkční blok *fbPIDU_SCU*








PID regulátor s trojstavovým řízením se zpětnou vazbou



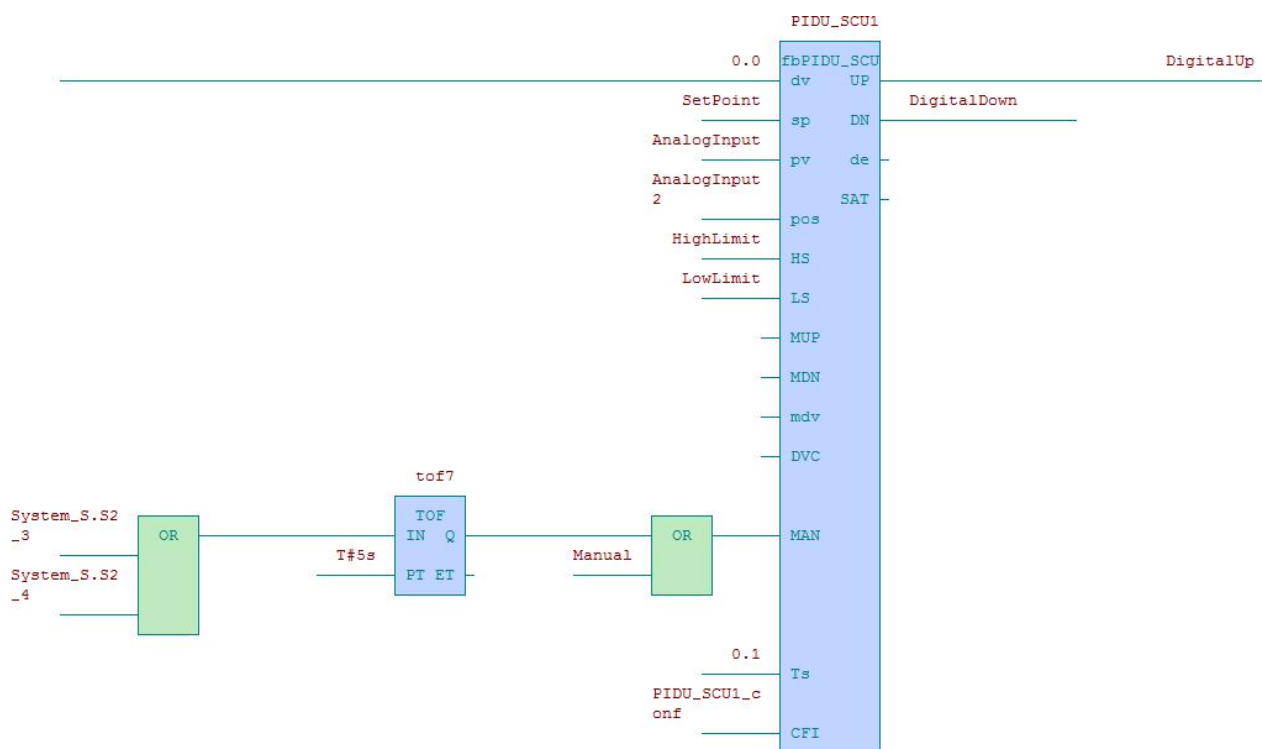
Funkční blok *fbPIDU_SCU* je propojením bloků *fbPIDU* a *fbSCU*, které realizují PID regulaci s trojstavovým výstupem na servoventil s polohovou zpětnou vazbou. Použitím bloku se minimalizuje počet nutných parametrů a možné chyby při spojení bloků.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>dv</i>	REAL	dopředná vazba
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>pos</i>	REAL	poloha servopohonu
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0→1)

	Proměnná	Typ	Význam
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<i>_TPIDU_SCU_IN_</i>	parametry regulátoru (viz 2.3 Typ <i>_TPIDU_SCU_IN_</i>)
VAR_OUTPUT			
	<i>UP</i>	BOOL	výstupní signál „nahoru“
	<i>DN</i>	BOOL	výstupní signál „dolů“
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován

Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDU_SCU* v jazyce FBD.



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogInput2 – analogový vstup s polohou akčního členu

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

HighLimit, *LowLimit* – horní, dolní omezení akčního členu

SetPoint – proměnná s žádanou hodnotou

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU_SCU1_Conf – struktura s parametry funkčního bloku

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDU_SCU1_conf : _TPIDU_SCU_IN_ := (irtype:= irtype_PID,
                                     par := (k:= 0.75, ti:= 8.0,
                                             td:= 1.0, nd:= 10.0, b:= 1.0,
                                             c:= 0.0), tt:= 1.0, dz:= 0.5,
                                     thron:= 2.0, throff:= 0.5,
                                     dtime:= 0.1, btime:= 0.2,
                                     trun:= 10.0);

END_VAR

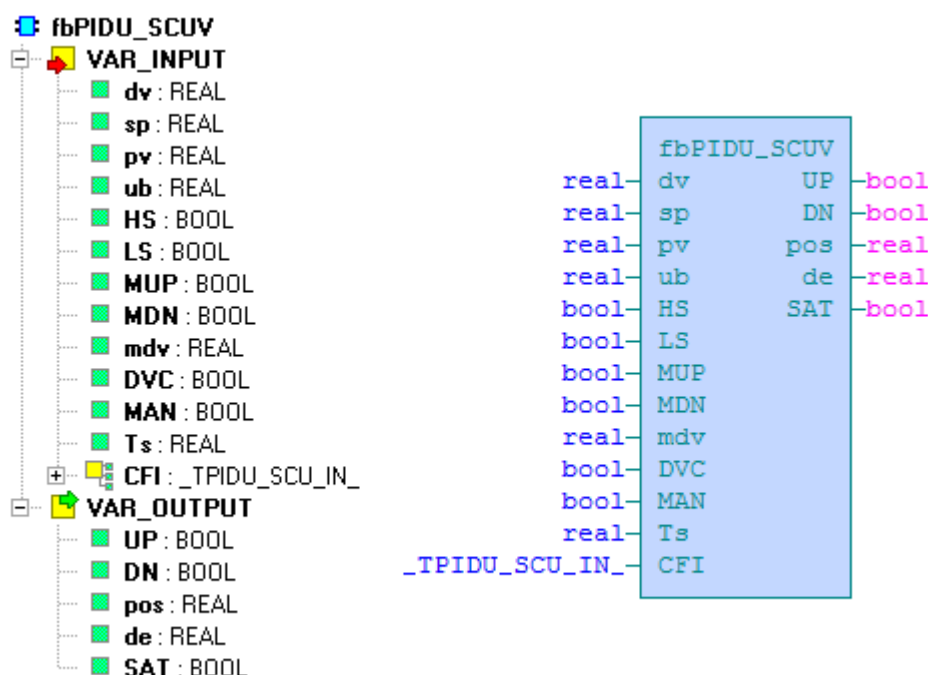
PROGRAM prgMain
  VAR
    tof7 : TOF;
    PIDU_SCU1 : fbPIDU_SCU;
  END_VAR

  tof7(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
  PIDU_SCU1(sp := SetPoint, pv := AnalogInput, pos := AnalogInput2,
            MAN := tof7.Q OR Manual,
            HS := HighLimit, LS := LowLimit,
            Ts := 0.1, CFI := PIDU_SCU1_conf,
            UP => DigitalUp, DN => DigitalDown);

END_PROGRAM
```

4.9 Funkční blok fbPIDU_SCUV

PID regulátor s trojstavovým řízením bez zpětné vazby



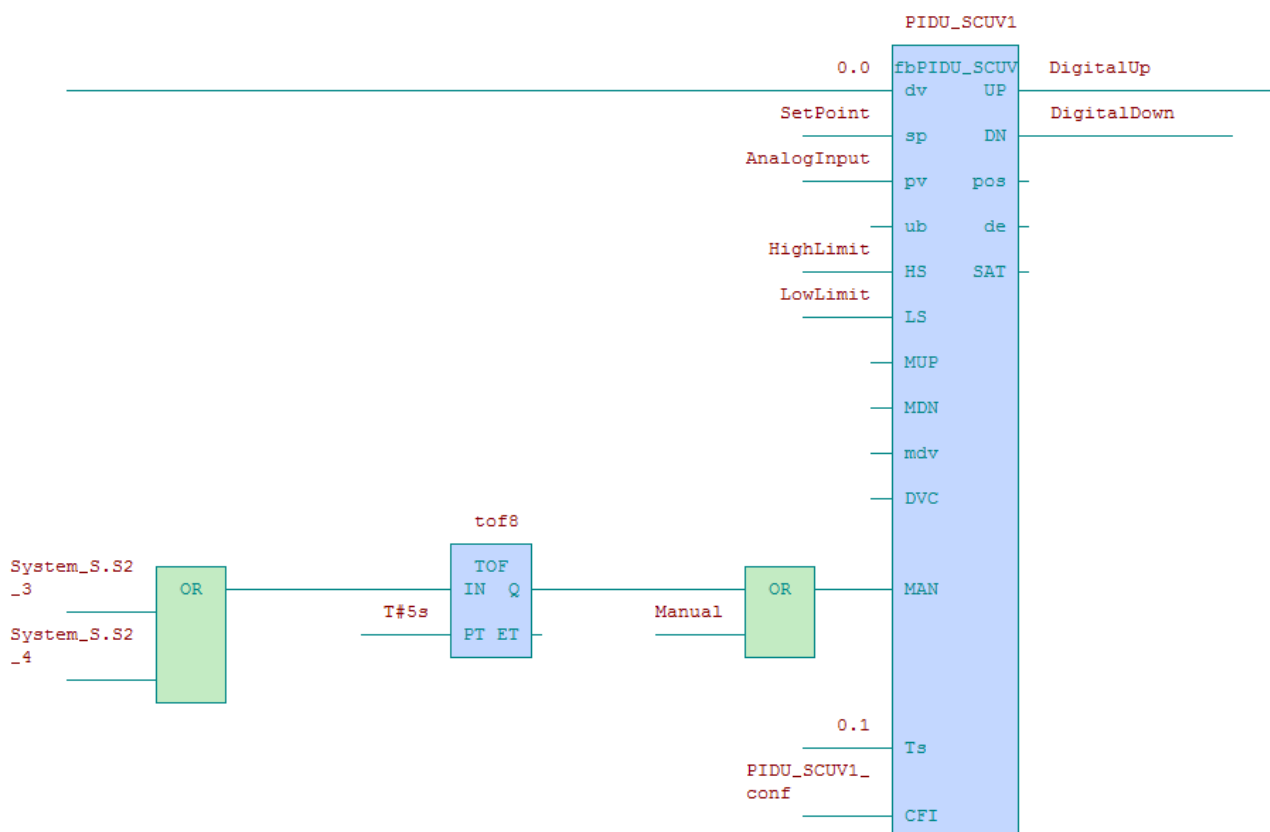
Funkční blok *fbPIDU_SCUV* realizuje propojení bloků *fbPIDU* a *fbSCUV*, které realizují PID regulaci s trojstavovým výstupem na servoventil bez polohové zpětné vazby. Použitím bloku se minimalizuje počet nutných parametrů a možné chyby při spojení bloků.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>dv</i>	REAL	dopředná vazba
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>ub</i>	REAL	posunutí (jen v případě, že primární regulátor je typu <i>P</i> nebo <i>PD</i>)
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)

	Proměnná	Typ	Význam
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0→1)
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	_TPIDU_SCU_IN_	parametry regulátoru (viz 2.3 Typ _TPIDU_SCU_IN_)
VAR_OUTPUT			
	<i>UP</i>	BOOL	výstupní signál „nahoru“
	<i>DN</i>	BOOL	výstupní signál „dolů“
	<i>pos</i>	REAL	simulovaná poloha motoru
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován

Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDU_SCUV* v jazyce FBD.



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

HighLimit, *LowLimit* – horní, dolní omezení akčního členu

SetPoint – proměnná s žádanou hodnotou

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDU_SCUV1_Conf – struktura s parametry funkčního bloku

V jazyce ST tomuto zapojení odpovídá následující program.

```

VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDU_SCUV1_conf : _TPIDU_SCU_IN_ := (irtype:= irtype_PID,
                                     par := (k:= 0.75, ti:= 8.0,
                                             td:= 1.0, nd:= 10.0, b:= 1.0,
                                             c:= 0.0), tt:= 1.0, dz:= 0.5,
                                             thron:= 2.0, throff:= 0.5,
                                             dtime:= 0.1, btime:= 0.2,
                                             trun:= 10.0);
END_VAR

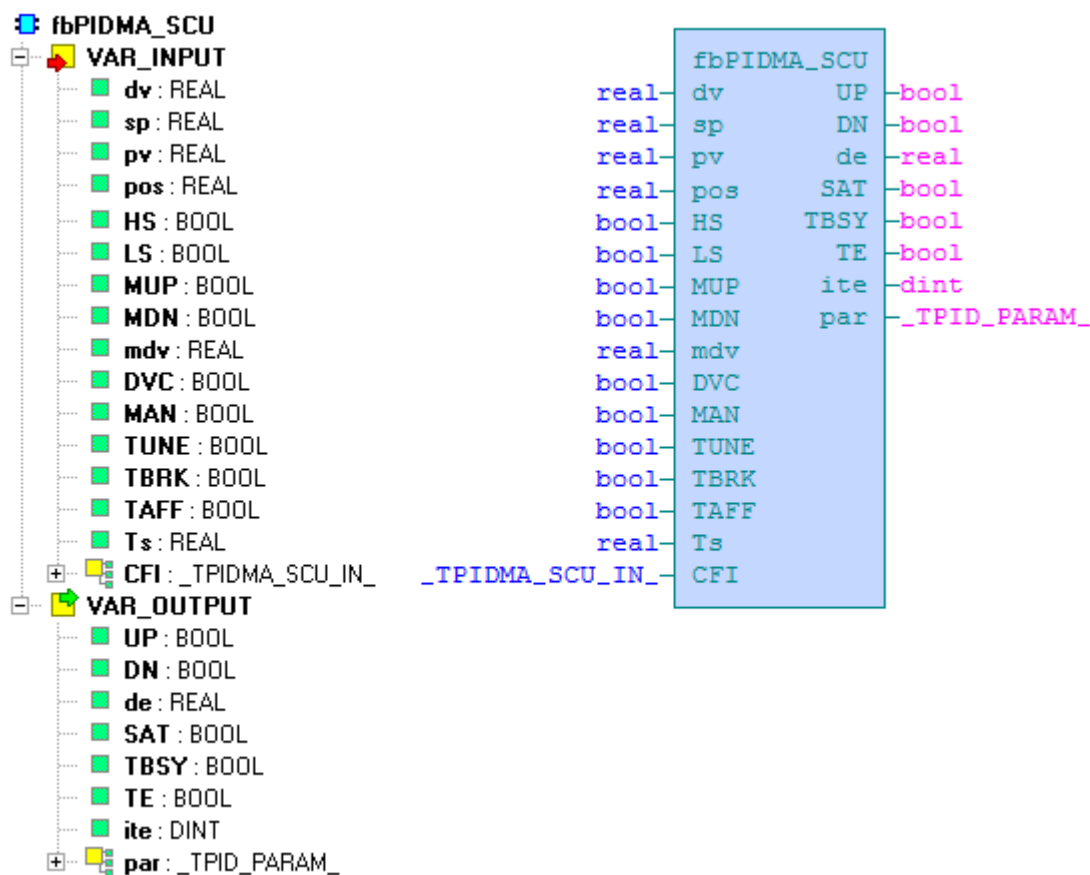
PROGRAM prgMain
VAR
  tof8 : TOF;
  PIDU_SCUV1 : fbPIDU_SCUV;
END_VAR

tof8(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDU_SCUV1(sp := SetPoint, pv := AnalogInput,
           MAN := tof8.Q OR Manual,
           HS := HighLimit, LS := LowLimit,
           Ts := 0.1, CFI := PIDU_SCUV1_conf,
           UP => DigitalUp, DN => DigitalDown);
END_PROGRAM

```






















4.10 Funkční blok *fbPIDMA_SCU*



PID regulátor s autotunerem a trojstavovým řízením se zpětnou vazbou



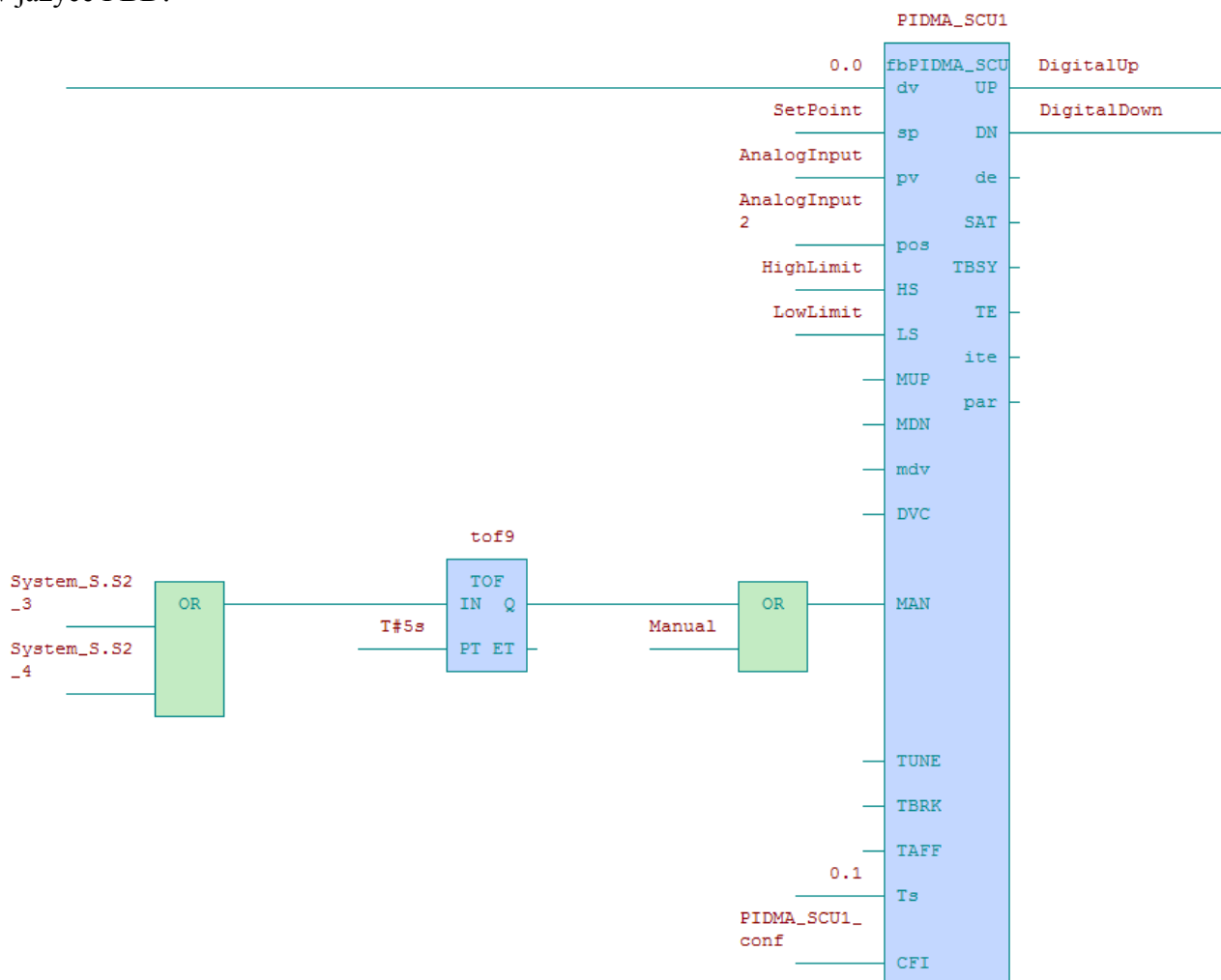
Funkční blok *fbPIDMA_SCU* realizuje propojení bloků *fbPIDMA* a *fbSCU*, které realizují PID regulaci s trojstavovým výstupem na servoventil s polohovou zpětnou vazbou s autotunerem. Použitím bloku se minimalizuje počet nutných parametrů a možné chyby při spojení bloků.

Popis proměnných :

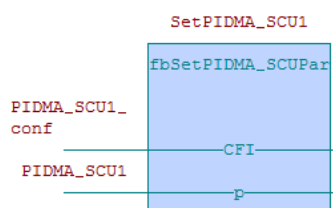
	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>dv</i>	REAL	dopředná vazba
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>pos</i>	REAL	poloha servopohonu
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0→1)
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>TUNE</i>	BOOL	spuštění ladicího algoritmu (0 → 1)
	<i>TBRK</i>	BOOL	přerušování ladicího algoritmu
	<i>TAFF</i>	BOOL	příznak pro potvrzení parametrů; určuje způsob zacházení s novými parametry 0 ... parametry se do regulátoru nenastavují 1 ... parametry se nastaví do regulátoru ihned po ukončení jejich výpočtu 0 → 1 parametry se nastaví do regulátoru jednorázově při této změně
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<code>_TPIDMA_SCU_IN_</code>	parametry regulátoru (viz 2.3 Typ <code>_TPIDU_SCU_IN_</code>)
VAR_OUTPUT			
	<i>UP</i>	BOOL	výstupní signál „nahoru“
	<i>DN</i>	BOOL	výstupní signál „dolů“
	<i>de</i>	REAL	regulační odchylka
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován
	<i>TBSY</i>	BOOL	příznak ladicího módu (<i>TBSY=1</i>)

	Proměnná	Typ	Význam
	<i>TE</i>	BOOL	příznak chyby ladění 0 ... bez chyby, 1 ... experiment selhal
	<i>ite</i>	DINT	specifikace chyby (po experimentu) 0 ... bez chyby 1 ... příliš malá hodnota prahu pro ukončení pulsu 2 ... příliš velká amplituda pulsu 3 ... nebylo dosaženo ustáleného stavu 4 ... příliš malá amplituda pulsu 5 ... selhání procedury hledání vrcholu 6 ... došlo k saturaci výstupu regulátoru při experimentu 7 ... pro vybraný typ regulátoru není podporováno automatické nastavování 8 ... nedodržena podmínka monotónnosti procesu 9 ... selhání extrapolace 10 ... neočekávané hodnoty momentů (fatální) 11 ... ruční přerušení experimentu uživatelem 12 ... nesprávný směr řídicí veličiny (změňte parametr <i>RACT</i>) 100 ... ruční ukončení ladění (varování) 0 ... čekání na ustálený stav před začátkem experimentu -1 ... odhad driftu a šumu (parametry <i>tdg</i> a <i>tn</i>) -2 ... generování obdélníkového pulsu (puls končí při změně <i>pv</i> o hodnotu větší než <i>dy</i>) -3 ... hledání vrcholu odezvy -4 ... odhad rychlosti ustalování odezvy Poznámka: Náběžná hrana vstupu <i>TUNE</i> během fází -2, -3 -4 způsobuje ukončení dané fáze a přechod do fáze následující (nebo ukončení experimentu ve fázi -4).

Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDMA_SCU* v jazyce FBD.



Pro přenos parametrů navržených tunerem do konfigurační struktury, lze z výhodou využít blok *fbSetPIDMA_SCUPar* (viz 4.13 Funkční blok *fbSetPIDMA_SCUPar*) :



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

AnalogInput2 – analogový vstup s polohou akčního členu

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

HighLimit, *LowLimit* – horní, dolní omezení akčního členu

SetPoint – proměnná s žádanou hodnotou

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDMA_SCU1_Conf – struktura s parametry funkčního bloku

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDMA_SCU1_conf : _TPIDMA_SCU_IN_ := (irtype:= irtype_PID,
                                       par := (k:= 0.75, ti:= 8.0,
                                               td:= 1.0, nd:= 10.0, b:= 1.0,
                                               c:= 0.0), tt:= 1.0, dz:= 0.5,
                                       ittype:= ittype_PID, iainf:= 1,
                                       DGC:= 1, tdg:= 10.0,
                                       tn:= 5.0, amp:= 15.0, dy:= 10.0,
                                       ispeed:= 2, thron:= 2.0,
                                       throff:= 0.5, dtime:= 0.1,
                                       btime:= 0.2, trun:= 10.0);

END_VAR

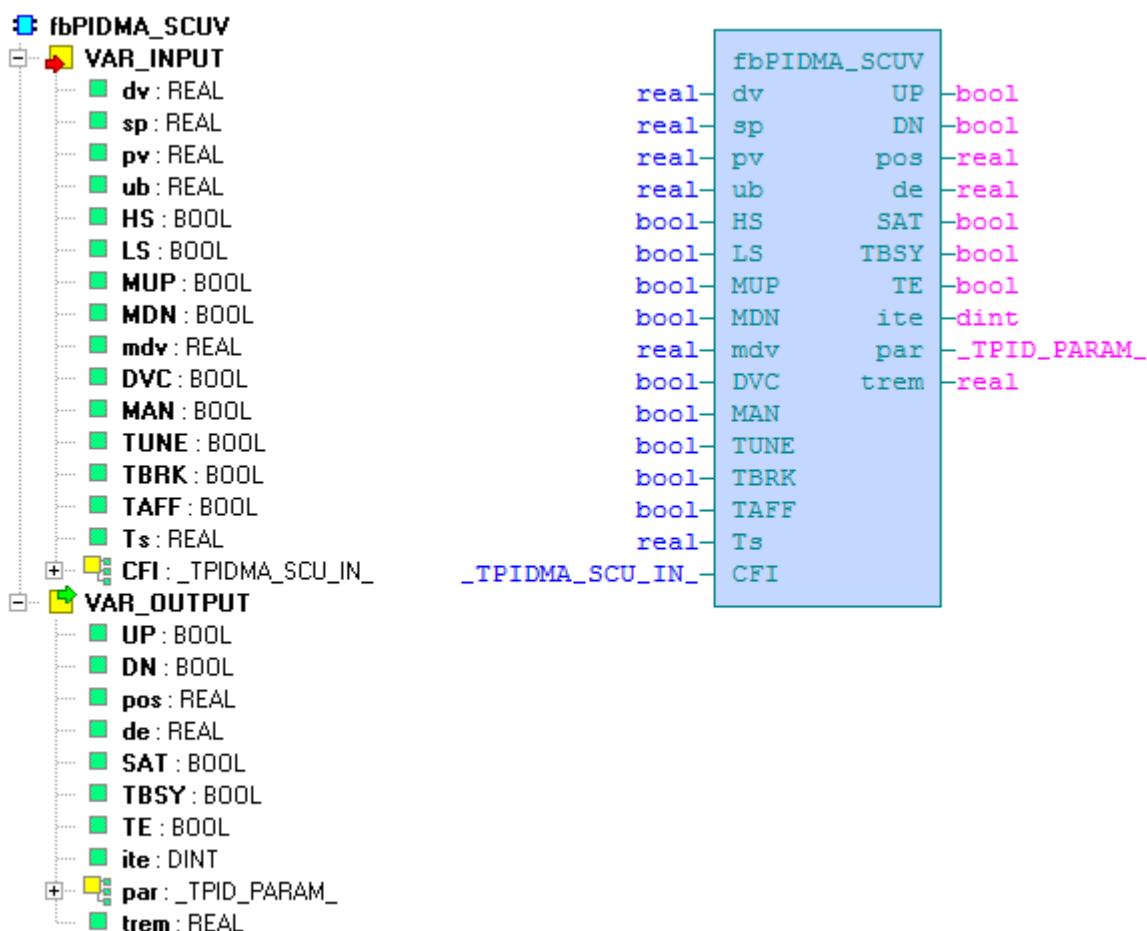
PROGRAM prgMain
VAR
  tof9 : TOF;
  PIDMA_SCU1 : fbPIDMA_SCU;
  SetPIDMA_SCU1 : fbSetPIDMA_SCUPar;
END_VAR

tof9(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDMA_SCU1(sp := SetPoint, pv := AnalogInput, pos := AnalogInput2,
           MAN := tof9.Q OR Manual,
           HS := HighLimit, LS := LowLimit,
           Ts := 0.1, CFI := PIDMA_SCU1_conf,
           UP => DigitalUp, DN => DigitalDown);
SetPIDMA_SCU1(CFI := PIDMA_SCU1_conf, p := PIDMA_SCU1);

END_PROGRAM
```





















4.11 Funkční blok fbPIDMA_SCUV





PID regulátor s autotunerem a trojstavovým řízením bez zpětné vazby



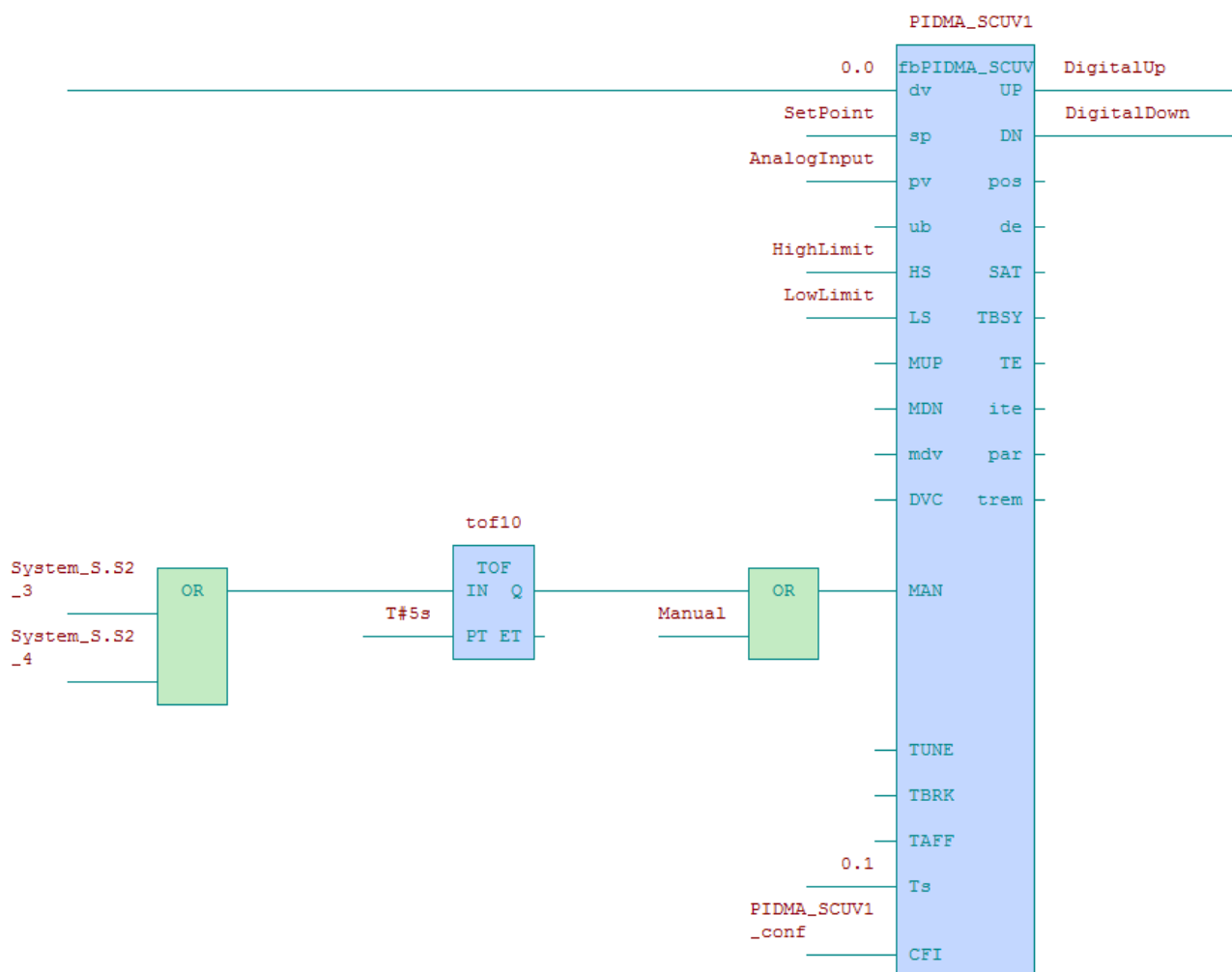
Funkční blok `fbPIDMA_SCUV` realizuje propojení bloků `fbPIDMA` a `fbSCUV`, které realizují PID regulaci s trojstavovým výstupem na servoventil bez polohové zpětné vazby s autotunerem. Použitím bloku se minimalizuje počet nutných parametrů a možné chyby při spojení bloků.

Popis proměnných :

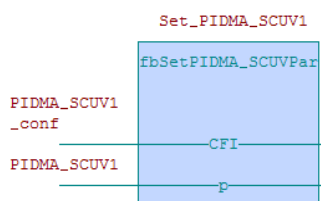
	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>dv</i>	REAL	dopředná vazba
	<i>sp</i>	REAL	žádaná veličina
	<i>pv</i>	REAL	regulovaná veličina
	<i>ub</i>	REAL	posunutí (jen v případě, že primární regulátor je typu <i>P</i> nebo <i>PD</i>)
	<i>HS</i>	BOOL	horní koncový spínač (příznak, že poloha ventilu je na horní mezi)
	<i>LS</i>	BOOL	spodní koncový spínač (příznak, že poloha ventilu je na spodní mezi)
	<i>MUP</i>	BOOL	ruční signál „nahoru“ (<i>UP</i>), kopírovaný na výstup v ručním režimu
	<i>MDN</i>	BOOL	ruční signál „dolů“ (<i>DN</i>), kopírovaný na výstup v ručním režimu
	<i>mdv</i>	REAL	ruční diferenční hodnota (požadovaný přírůstek/úbytek polohy, mající vyšší prioritu než přímé signály <i>MUP/MDN</i>)
	<i>DVC</i>	BOOL	povel pro přijetí diferenční hodnoty (0→1)
	<i>MAN</i>	BOOL	režim činnosti regulátoru 0 = automatický režim, 1 = manuální režim
	<i>TUNE</i>	BOOL	spuštění ladicího algoritmu (0 → 1)
	<i>TBRK</i>	BOOL	přerušování ladicího algoritmu
	<i>TAFF</i>	BOOL	příznak pro potvrzení parametrů; určuje způsob zacházení s novými parametry 0 ... parametry se do regulátoru nenastavují 1 ... parametry se nastaví do regulátoru ihned po ukončení jejich výpočtu 0 → 1 parametry se nastaví do regulátoru jednorázově při této změně
	<i>Ts</i>	REAL	vzorkovací perioda regulátoru v sekundách
	<i>CFI</i>	<code>_TPIDMA_SCU_IN_</code>	parametry regulátoru (viz 2.3 Typ <code>_TPIDMA_SCU_IN_</code>)
VAR_OUTPUT			
	<i>UP</i>	BOOL	výstupní signál „nahoru“
	<i>DN</i>	BOOL	výstupní signál „dolů“
	<i>pos</i>	REAL	simulovaná poloha motoru
	<i>de</i>	REAL	regulační odchylka

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
	<i>SAT</i>	BOOL	příznak saturace 0 = regulátor pracuje v lineární oblasti 1 = výstup regulátoru je saturován
	<i>TBSY</i>	BOOL	příznak ladicího módu ($TBSY=1$)
	<i>TE</i>	BOOL	příznak chyby ladění 0 ... bez chyby, 1 ... experiment selhal
	<i>ite</i>	DINT	specifikace chyby (po experimentu) 0 ... bez chyby 1 ... příliš malá hodnota prahu pro ukončení pulsu 2 ... příliš velká amplituda pulsu 3 ... nebylo dosaženo ustáleného stavu 4 ... příliš malá amplituda pulsu 5 ... selhání procedury hledání vrcholu 6 ... došlo k saturaci výstupu regulátoru při experimentu 7 ... pro vybraný typ regulátoru není podporováno automatické nastavování 8 ... nedodržena podmínka monotónnosti procesu 9 ... selhání extrapolace 10 ... neočekávané hodnoty momentů (fatální) 11 ... ruční přerušení experimentu uživatelem 12 ... nesprávný směr řídicí veličiny (změňte parametr <i>RACT</i>) 100 ... ruční ukončení ladění (varování) 0 ... čekání na ustálený stav před začátkem experimentu -1 ... odhad driftu a šumu (parametry <i>tdg</i> a <i>tn</i>) -2 ... generování obdélníkového pulsu (puls končí při změně <i>pv</i> o hodnotu větší než <i>dy</i>) -3 ... hledání vrcholu odezvy -4 ... odhad rychlosti ustalování odezvy Poznámka: Náběžná hrana vstupu <i>TUNE</i> během fázi -2, -3 -4 způsobuje ukončení dané fáze a přechod do fáze následující (nebo ukončení experimentu ve fázi -4).

Následující příklad ukazuje základní zapojení funkčního bloku *fbPIDMA_SCUV* v jazyce FBD.



Pro přenos parametrů navržených tunerem do konfigurační struktury, lze z výhodou využít blok *fbSetPIDMA_SCUVPar* (viz 4.14 Funkční blok *fbSetPIDMA_SCUVPar*):



Význam jednotlivých proměnných:

AnalogInput – analogový vstup s regulovanou veličinou

DigitalUp – binární výstup pro kladný akční zásah

DigitalDown – binární výstup pro záporný akční zásah

HighLimit, *LowLimit* – horní, dolní omezení akčního členu

SetPoint – proměnná s žádanou hodnotou

System_S.S2_3 – příznak prvního průchodu cyklu po teplém restartu z knihovny Syslib

System_S.S2_4 – příznak prvního průchodu cyklu po studeném restartu z knihovny Syslib

PIDMA_SCUV1_Conf – struktura s parametry funkčního bloku

V jazyce ST tomuto zapojení odpovídá následující program.

```
VAR_GLOBAL RETAIN
Manual      : BOOL;
SetPoint    : REAL;
PIDMA_SCUV1_conf : _TPIDMA_SCU_IN := (irtype:= irtype_PID,
                                       par := (k:= 0.75, ti:= 8.0,
                                               td:= 1.0, nd:= 10.0, b:= 1.0,
                                               c:= 0.0), tt:= 1.0, dz:= 0.5,
                                       ittype:= ittype_PID, iainf:= 1,
                                       DGC:= 1, tdg:= 10.0,
                                       tn:= 5.0, amp:= 10.0, dy:= 10.0,
                                       ispeed:= 2, thron:= 2.0,
                                       throff:= 0.5, dtime:= 0.1,
                                       btime:= 0.2, trun:= 10.0);

END_VAR

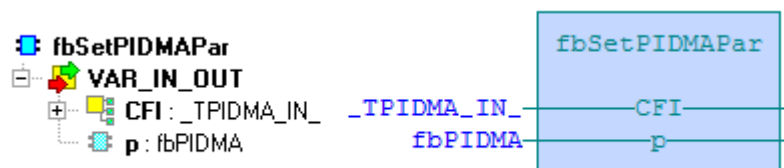
PROGRAM prgMain
VAR
  tof10 : TOF;
  PIDMA_SCUV1 : fbPIDMA_SCUV;
  Set_PIDMA_SCUV1 : fbSetPIDMA_SCUVPar;
END_VAR

tof10(IN := System_S.S2_3 OR System_S.S2_4, PT := T#5s);
PIDMA_SCUV1(sp := SetPoint, pv := AnalogInput,
            MAN := tof10.Q OR Manual,
            HS := HighLimit, LS := LowLimit,
            Ts := 0.1, CFI := PIDMA_SCUV1_conf,
            UP => DigitalUp, DN => DigitalDown);
Set_PIDMA_SCUV1 : fbSetPIDMA_SCUVPar;

END_PROGRAM
```

4.12 Funkční blok fbSetPIDMAPar

Blok pro přesun výsledků autotuneru do konfigurační struktury



Blok *fbSetPIDMAPar* slouží k předání parametrů z výstupu autotuneru bloku *fbPIDMA* (vstup *p*) do struktury parametrů (vstup *CFI*), která je obecně uložena v remanentní zóně.

Blok se řídí vstupem *TAFF* bloku *fbPIDMA*. Pokud je *TAFF* nastavené na hodnotu 0 parametry se nekopírují, pokud má hodnotu 1 parametry se kopírují automaticky po skončení výpočtu autotuneru, při přechodu z 0 → 1 se parametry jednorázově překopírují.

Překopírování parametrů je podmíněno nulovostí výstupu signalizující chybu *TE* a vstupu *isp*, který ovlivňuje formát a význam dat z autotuneru.

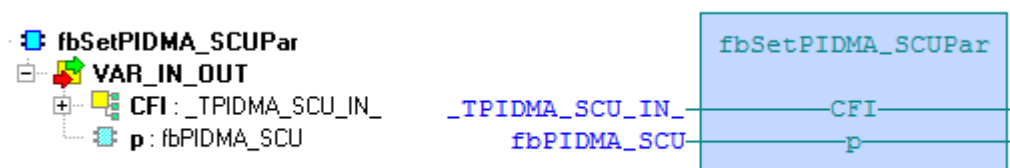
Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_IN_OUT			
	<i>CFI</i>	<code>_TPIDMA_IN_</code>	struktura s parametry regulátoru
	<i>p</i>	<code>fbPIDMA</code>	funkční blok regulátoru

Příklad použití viz kapitola 4.2 Funkční blok *fbPIDMA*.

4.13 Funkční blok *fbSetPIDMA_SCUPar*

Blok pro přesun výsledků autotuneru do konfigurační struktury



Blok *fbSetPIDMA_SCUPar* slouží k předání parametrů z výstupu autotuneru bloku *fbPIDMA_SCU* (vstup *p*) do struktury parametrů (vstup *CFI*), která je obecně uložena v remanentní zóně.

Blok se řídí vstupem *TAFF* bloku *fbPIDMA_SCU*. Pokud je *TAFF* nastavené na hodnotu 0 parametry se nekopírují, pokud má hodnotu 1 parametry se kopírují automaticky po skončení výpočtu autotuneru, při přechodu z 0 → 1 se parametry jednorázově překopírují.

Překopírování parametrů je podmíněno nulovostí výstupu signalizující chybu *TE* a vstupu *isp*, který ovlivňuje formát a význam dat z autotuneru.

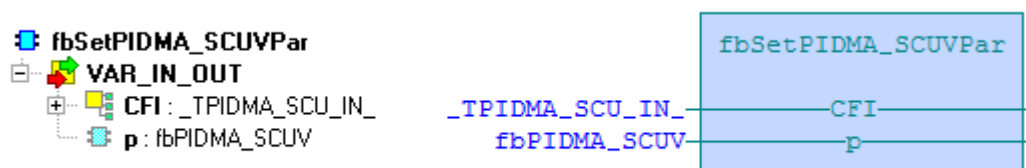
Popis proměnných:

	Proměnná	Typ	Význam
VAR_IN_OUT			
	<i>CFI</i>	_TPIDMA_SCU_IN_	struktura s parametry regulátoru
	<i>p</i>	fbPIDMA_SCU	funkční blok regulátoru

Příklad použití viz kapitola 4.10 Funkční blok *fbPIDMA_SCU*.

4.14 Funkční blok fbSetPIDMA_SCUVPAr

Blok pro přesun výsledků autotuneru do konfigurační struktury



Blok *fbSetPIDMA_SCUVPAr* slouží k předání parametrů z výstupu autotuneru bloku *fbPIDMA_SCUV* (vstup *p*) do struktury parametrů (vstup *CFI*), která je obecně uložena v remanentní zóně.

Blok se řídí vstupem *TAFF* bloku *fbPIDMA_SCUV*. Pokud je *TAFF* nastavené na hodnotu 0 parametry se nekopírují, pokud má hodnotu 1 parametry se kopírují automaticky po skončení výpočtu autotuneru, při přechodu z 0 → 1 se parametry jednorázově překopírují.

Překopírování parametrů je podmíněno nulovostí výstupu signalizující chybu *TE* a vstupu *isp*, který ovlivňuje formát a význam dat z autotuneru.

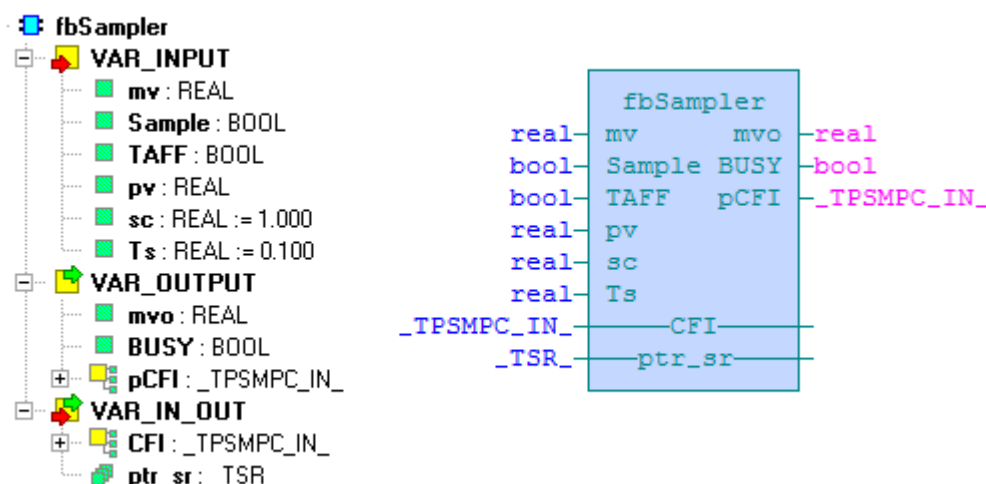
Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_IN_OUT			
	<i>CFI</i>	_TPIDMA_SCUV_IN_	struktura s parametry regulátoru
	<i>p</i>	fbPIDMA_SCUV	funkční blok regulátoru

Příklad použití viz kapitola 4.11 Funkční blok fbPIDMA_SCUV.

4.15 Funkční blok fbSampler

Blok pro zachycení přechodové charakteristiky a základní identifikaci



Funkční blok *fbSampler* slouží k získání a analýze přechodové charakteristiky (odezvy regulovaného procesu na skokovou změnu). Blok je primárně navržen k použití spolu s regulátorem realizovaným blokem *fbPSMPC*.

Blok se připojuje za regulátor vstupem *mv*, do které vstupuje výstupu regulátoru. Tento vstup je po dobu, kdy není aktivní snímání kopírován na výstup *mvo*. Snímání přechodové charakteristiky se aktivuje nastavením vstupu *Sample* do logické 1. Při aktivním snímání se na výstupu *mvo* realizuje skoková změna proti poslední hodnotě *mv* o hodnotu přivedenou na vstup *sc*. Po nastavení *Sample* zpět do hodnoty 0 nebo po naplnění pole na vstupu *ptr_sr* se snímání zastaví. Aktivní snímání signalizuje výstup *BUSY*.

Snímaná (regulovaná) veličina se přivádí na vstup *pv*. Přechodová charakteristika je normalizována hodnotou *sc* tak, aby odpovídala jednotkovému skoku. Vstup *Ts* udává periodu vzorkování, ta musí být stejná jako u regulátoru, pro který je určena přechodová charakteristika.

Získané parametry *np1* (číslo vzorku s první signifikantní změnou snímané veličiny), *np2* (číslo vzorku téměř ustáleného stavu), *kappa* (κ – zesílení soustavy), *mu* (μ – celková časová konstanta – míra zpoždění soustavy), *sigma* (σ – míra délky odezvy soustavy) a *nsr* (počet nastímaných vzorků) se po skončení snímání objeví na výstupu *pCFI*.

Pokud je vstup *TAFF* v logické jedničce, jsou přechodová charakteristika a parametry automaticky zkopírovány do proměnných na vstupu *ptr_sr* a *CFI* po skončení snímání. V případě změny *TAFF* z 0 → 1 jsou zkopírovány jednorázově. V obou případech je také nastaven příznak *CHANGED*, aby byly nové parametry regulátorem akceptovány.












Pro nasnímaní přechodové charakteristiky je důležité dodržet následující postup.

- 1) uvést soustavu do ustáleného stavu ve vhodném pracovním bodě (nejlépe v ručním režimu regulátoru)
- 2) nastavit hodnotu *sc* tak, aby byla odezva procesu dostatečně velká (v poměru k okolnímu šumu a dalším vnějším vlivům)
- 3) nastavit hodnotu *Ts* tak, aby se přechodová odezva vešla do 255 vzorků
- 4) nastavit *Sample* do logické 1 a vyčkat do nového ustáleného stavu

Získané parametry soustavy se dají využít kromě přímého propojení s blokem *fbPSMPC* i k odhadnutí parametrů PID regulátoru, například dle následujících vztahů:

$$k = \frac{1,2 \cdot \kappa \cdot (\mu - \sigma)}{\tau}, \quad ti = 2,0 \cdot (\mu - \sigma), \quad td = 1,0 \cdot (\mu - \sigma)$$

Popis proměnných:

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>mv</i>	REAL	výstup regulátoru
	<i>Sample</i>	BOOL	žádaná veličina
	<i>TAFF</i>	BOOL	příznak pro potvrzení parametrů 0 ... parametry se do proměnných na vstupu <i>ptr_sr</i> a <i>CFI</i> nekopírují 1 ... parametry se do proměnných na vstupu <i>ptr_sr</i> a <i>CFI</i> kopírují 0 → 1 parametry se nastaví do proměnných na vstupu <i>ptr_sr</i> a <i>CFI</i> kopírují jednorázově při této změně
	<i>pv</i>	REAL	snímaná (regulovaná) veličina
	<i>sc</i>	REAL	velikost skokové změny
	<i>Ts</i>	REAL	perioda vzorkování
VAR_OUTPUT			
	<i>mvo</i>	REAL	kopie výstupu regulátoru, na které se realizuje skoková změna
	<i>BUSY</i>	BOOL	příznak aktivního snímání přechodové charakteristiky
	<i>pCFI</i>	_TPSMPC_IN_	navržené parametry regulátoru (viz 4.3 Funkční blok fbPSMPC)
VAR_IN_OUT			
	<i>CFI</i>	_TPSMPC_IN_	parametry regulátoru (viz 4.3 Funkční blok fbPSMPC)
	<i>ptr_sr</i>	_TSR_	přechodová charakteristika regulovaného procesu

Příklad použití viz kapitola 4.3 Funkční blok fbPSMPC.



teco

Objednávky a informace:

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

TXV 003 45.01

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