



---

PROGRAMOVATELNÉ AUTOMATY

# **REGULAČNÍ KNIHOVNY PRO MOSAIC**

**TXV 003 23.01**

# REGULAČNÍ KNIHOVNY PRO MOSAIC

4. vydání – listopad 2008

## OBSAH

<b>1. ÚVOD</b> .....	<b>3</b>
<b>2. KNIHOVNA REGOLIB</b> .....	<b>4</b>
2.1. EKVITERMNÍ KŘIVKY .....	5
2.1.1. Ekvitem1 - ekvitemní křivka s pevnými body venkovní teploty.....	5
2.1.2. Ekvitem2 - ekvitemní křivka s nastavitelnými body venkovní teploty .....	6
2.2. HYSTEREZNÍ REGULÁTORY .....	8
2.2.1. Hyst1 - hystereze .....	8
2.2.2. Hyst2 - hystereze MIN,MAX.....	9
2.2.3. Hyst3 - dvojitá hystereze MIN,MAX.....	10
2.2.4. Hyst31 - dvojitá hystereze MIN,MAX s řídicí proměnnou.....	11
2.3. PID REGULÁTORY .....	12
2.3.1. Obecný popis .....	12
2.3.2. PID1 - regulátor s přírůstkovým řízením.....	15
2.3.3. PID11 - regulátor s přírůstkovým řízením a užším výběrem proměnných .....	16
2.3.4. PID2 - regulátor s přímým řízením .....	18
2.3.5. PID21 - regulátor s přímým řízením a užším výběrem proměnných .....	19
2.3.6. PID3 - volně nastavitelný regulátor .....	21
2.4. KASKÁDNÍ ŘAZENÍ .....	23
2.4.1. Cascade2 - kaskáda 2 stupňů se střídáním .....	24
2.4.2. Cascade3 - kaskáda 3 stupňů se střídáním .....	25
2.4.3. Cascade4 - kaskáda 4 stupňů se střídáním .....	26
2.4.4. Cascade5 - kaskáda 5 stupňů se střídáním .....	28
2.5. PORUCHOVÉ SIGNALIZACE .....	30
2.5.1. SigErr1 - binární poruchová signalizace.....	30
2.5.2. SigErr11 - binární poruchová signalizace se signalizací čísla poruchy .....	31
2.5.3. SigErr12 - binární poruchová signalizace s obsluhou propojení .....	32
2.5.4. SigErr13 - binární poruchová signalizace s volbou nulování.....	33
2.5.5. SigErr2 - analogová poruchová signalizace .....	35
2.5.6. SigErr21 - analogová poruchová signalizace se signalizací čísla poruchy.....	36
2.5.7. SigErr22 - analogová poruchová signalizace s obsluhou propojení.....	37
2.5.8. SigErr23 - analogová poruchová signalizace s volbou nulování .....	39
2.6. HISTORIE PORUCH.....	41
2.6.1. History1 - historie poruch pro jednu poruchovou signalizací .....	41
2.6.2. History5 - historie poruch pro pět poruchových signalizací .....	42
2.6.3. History10 - historie poruch pro deset poruchových signalizací.....	43
2.7. ČASOVÉ PROGRAMY .....	44
2.7.1. TProg1 - týdenní časový program s jedním úsekem ZAP/VYP za den .....	44
2.7.2. TProg2 - týdenní časový program se dvěma úseky ZAP/VYP za den .....	45
2.7.3. TProg31 - týdenní časový program s jedním úsekem doby provozu.....	46
2.7.4. TProg41 - týdenní časový program se dvěma úseky doby provozu.....	47
<b>3. KNIHOVNA IRCLIB</b> .....	<b>49</b>
3.1. IRC – PATROVÝ MODUL.....	49

# 1. ÚVOD

Knihovny funkcí a funkčních bloků jsou nedílnou součástí instalace programovacího prostředí Mosaic. Z hlediska jejich výstavby je možné knihovny rozdělit na následující typy:

- vestavěné (built-in) knihovny
- standardně dodávané externí knihovny
- uživatelsky definované knihovny

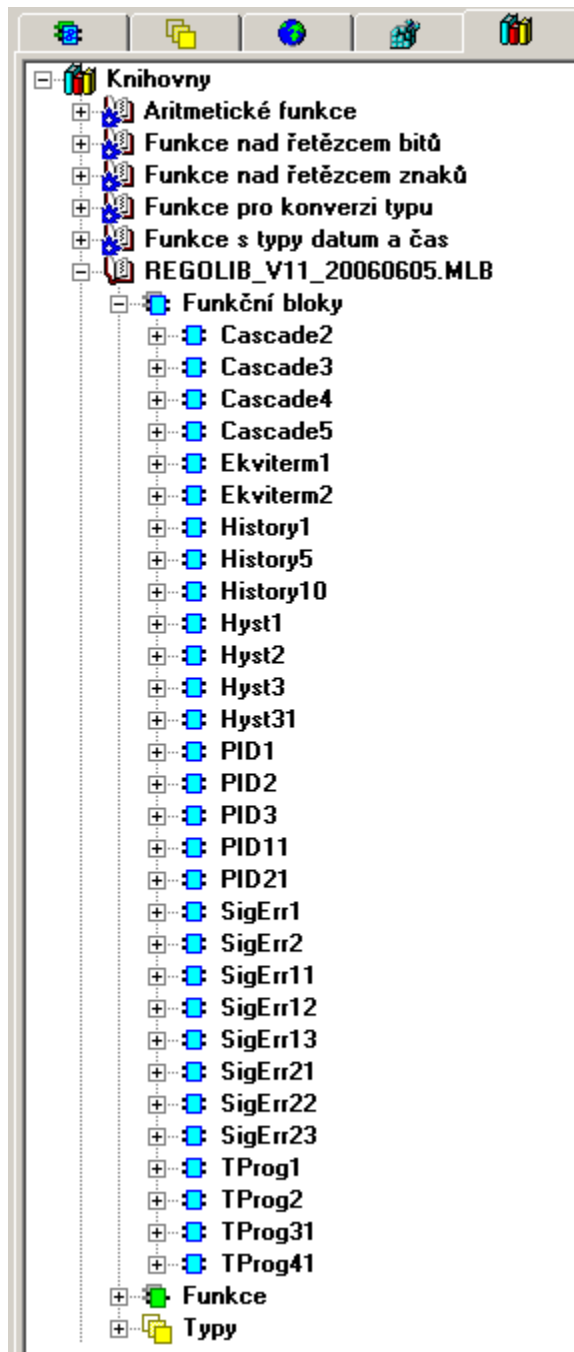
Knihovna může obsahovat deklarace funkcí, funkčních bloků, datových typů a globálních proměnných.

## 2. KNIHOVNA REGOLIB

Regulační knihovna RegoLib.mlb obsahuje základní funkční bloky, které slouží zejména při řešení regulačních úloh v PLC Tecomat. Množství funkčních bloků této knihovny však najde široké uplatnění i mimo regulační úlohy.

Základem knihovny RegoLib je sada vybraných komponent z programovacího prostředí Merkur. Jsou to především funkční bloky PID, ekvitermních a hysterezních regulací, kaskádního řazení, poruchových signalizací, historií poruch, časových programů.

Následující obrázek ukazuje strukturu knihovny RegoLib v prostředí Mosaic.



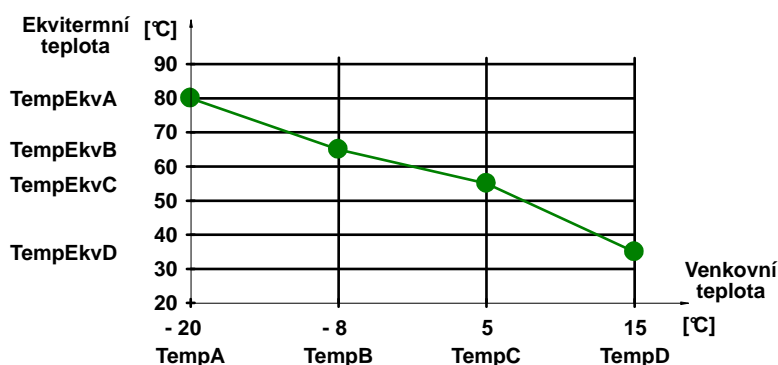
Obr. 2.1 Knihovna RegoLib

## 2.1. EKVITERMNÍ KŘIVKY

### 2.1.1. Ekviterm1 - ekvitermní křivka s pevnými body venkovní teploty

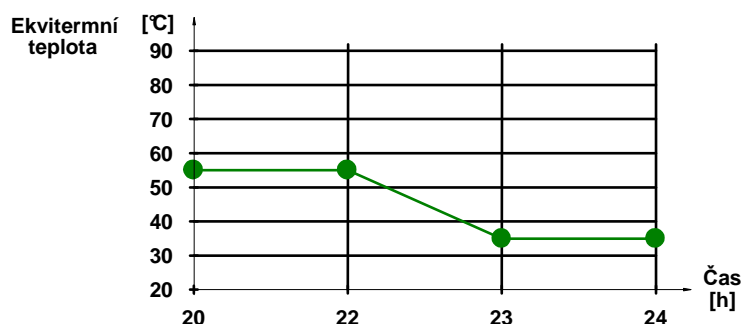
Funkční blok Ekviterm1 provádí výpočet žádané teploty *Out* na základě měřené venkovní teploty *In*, nastavených parametrů čtyřbodové ekvitermní křivky a požadavku útlumového programu.

Parametry *TempEkvA-TempEkvD* umožňují pohybovat nastavenou ekvitermní křivkou ve vertikálním směru. Teploty zlomu ekvitermní křivky jsou stanoveny pevně na *TempA* = -20°C, *TempB* = -8°C, *TempC* = +5°C, *TempD* = +15°C. Maximální vypočtená teplota dle ekvitermní křivky je rovna bodu *TempEkvA*, minimální je rovna bodu *TempEkvD*.



Obr. 2.2 Ekvitermní křivka

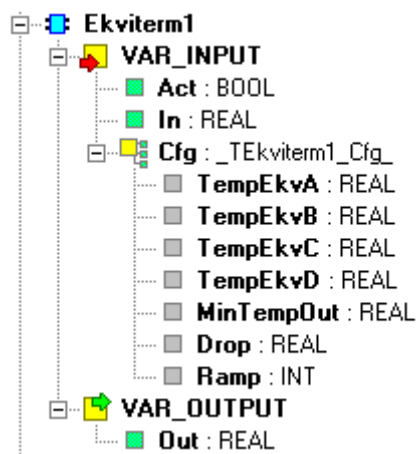
Vypočtená ekvitermní teplota může být snížena o nastavený útlum, ne však méně, než určuje parametr minimální výstupní teploty *MinTempOut*. Pokud je vstupní proměnná *Act* v log.0, je vypočtená ekvitermní teplota postupně snižována dle rampy na hodnotu ekvitermní sníženou o nastavený útlum *Drop*. Pokud se proměnná *Act* změní z log.0 do log.1, je opět výstupní teplota dle rampy zvyšována na vypočtenou ekvitermní teplotu. Doba, za kterou je tato hodnota dosažena, je dána parametrem *Ramp*. Funkci útlumového algoritmu ilustruje následující obrázek.



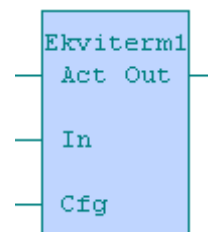
Obr. 2.3 Útlumová křivka

Graf zobrazuje průběh výstupní ekvitermní teploty *Out* při přechodu do útlumového provozu pro :

vypočtená ekvitermní teplota  $OUT = 55^{\circ}\text{C}$ ,  
 velikost útlumu  $DROP = 20^{\circ}\text{C}$ ,  $RAMP = 60 \text{ min}$   
 vypnutí provozu topení ve 22 hodin.



Obr. 2.4 Struktura FB Ekviterm1



Obr. 2.5 Vzhled FB Ekviterm1

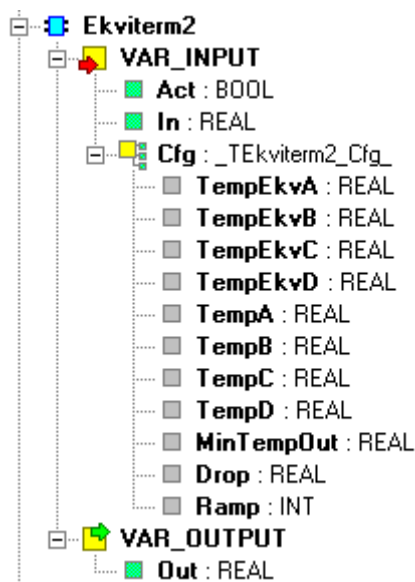
Popis proměnných :

Název	Význam	Typ	Formát
Act	provoz topení (1/0 – topit/útlum)	vstupní	bool
In	venkovní teplota [°C]		real
Cfg	konfigurační parametry bloku		_TEkviterm1_Cfg_
.TempEkvA	požadovaná ekvitermní teplota při venkovní teplotě bod A (-20) [°C]		real
.TempEkvB	požadovaná ekvitermní teplota při venkovní teplotě bod B (-8) [°C]		real
.TempEkvC	požadovaná ekvitermní teplota při venkovní teplotě bod C (+5) [°C]		real
.TempEkvD	požadovaná ekvitermní teplota při venkovní teplotě bod D (+15) [°C]		real
.MinTempOut	minimální ekvitermní teplota na výstupu [°C]		real
.Drop	útlum výstupní teploty [°C]		real
.Ramp	časové zpoždění začátku a konce útlumového provozu [min]	int	
Out	výstupní ekvitermní teplota [°C]	výstupní	real

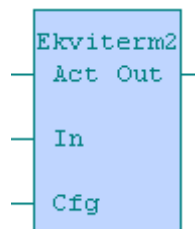
### 2.1.2. Ekviterm2 - ekvitermní křivka s nastavitelnými body venkovní teploty

Funkční blok Ekviterm2 provádí výpočet žádané teploty *Out* na základě měřené venkovní teploty *In*, nastavených parametrů čtyřbodové ekvitermní křivky a požadavku útlumového programu.

Funkcí je funkční blok Ekviterm2 shodný s Ekviterm1, rozdíl je pouze v tom, že teploty zlomu ekvitermní křivky *TempA* – *TempD* jsou volitelné. Parametry *TempEkvA*-*TempEkvD* umožňují pohybovat nastavenou ekvitermní křivkou ve vertikálním a parametry *TempA*-*TempD* v horizontálním směru.



Obr. 2.6 Struktura FB Ekvitem2



Obr. 2.7 Vzhled FB Ekvitem2

Popis proměnných :

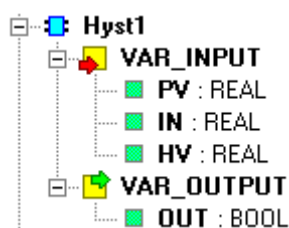
Název	Význam	Typ	Formát
Act	provoz topení (1/0 – topit/útlum)	vstupní	bool
In	venkovní teplota [°C]		real
Cfg	konfigurační parametry bloku		_TEkvitem2_Cfg_
.TempEkvA	požadovaná ekvitermní teplota při venkovní teplotě bod A [°C]		real
.TempEkvB	požadovaná ekvitermní teplota při venkovní teplotě bod B [°C]		real
.TempEkvC	požadovaná ekvitermní teplota při venkovní teplotě bod C [°C]		real
.TempEkvD	požadovaná ekvitermní teplota při venkovní teplotě bod D [°C]		real
.TempA	požadovaná venkovní teplota bod A [°C]		real
.TempB	požadovaná venkovní teplota bod B [°C]		real
.TempC	požadovaná venkovní teplota bod C [°C]		real
.TempD	požadovaná venkovní teplota bod D [°C]		real
.MinTempOut	minimální ekvitermní teplota na výstupu [°C]		real
.Drop	útlum výstupní teploty [°C]		real
.Ramp	časové zpoždění začátku a konce útlumového provozu [min]		int
Out	výstupní ekvitermní teplota [°C]		výstupní

## 2.2. HYSTEREZNÍ REGULÁTORY

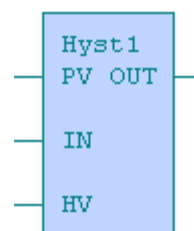
### 2.2.1. Hyst1 - hystereze

Funkční blok Hyst1 provede porovnání měřené hodnoty  $IN$  s nastavenou požadovanou hodnotou  $PV$  a uvažováním hystereze  $HV$ .

Pokud je měřená hodnota  $IN$  vyšší než žádaná hodnota  $PV+HV/2$ , je výstupní signál  $OUT$  nastaven do log. "1". Pokud je měřená hodnota  $IN$  nižší než žádaná hodnota  $PV-HV/2$ , je výstupní signál  $OUT$  nastaven do log. "0".



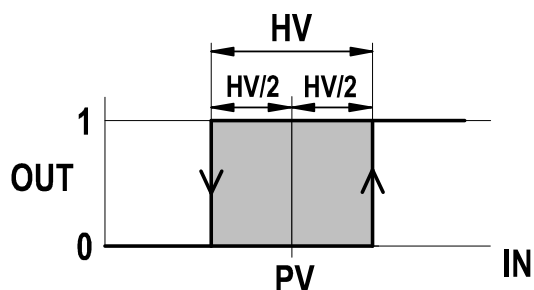
Obr. 2.8 Struktura FB Hyst1



Obr. 2.9 Vzhled FB Hyst1

Popis proměnných :

Název	Význam	Typ	Formát
PV	požadovaná hodnota	vstupní	real
IN	měřená hodnota		real
HV	hystereze		real
OUT	výstupní hodnota	výstupní	bool

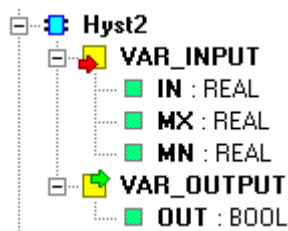


Obr. 2.10 Graf průběhů FB Hyst1

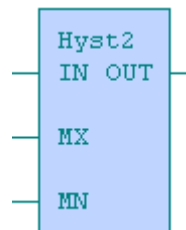


2.2.2. Hyst2 - hystereze MIN,MAX

Pokud měřená hodnota *IN* překročí požadovanou maximální hodnotu *MX*, je výstupní binární signál *OUT* nastaven do log. "1". V případě, že měřená hodnota *IN* podkročí požadovanou minimální hodnotu *MN*, je výstupní signál *OUT* nastaven do log. "0".



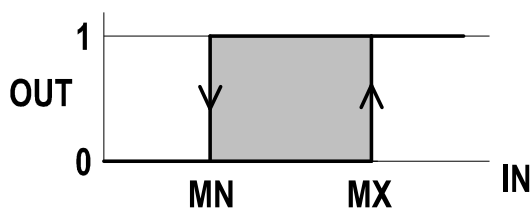
Obr. 2.11 Struktura FB Hyst2



Obr. 2.12 Vzhled FB Hyst2

Popis proměnných :

Název	Význam	Typ	Formát
IN	měřená hodnota	vstupní	real
MX	maximum		real
MN	minimum		real
OUT	výstupní hodnota	výstupní	bool

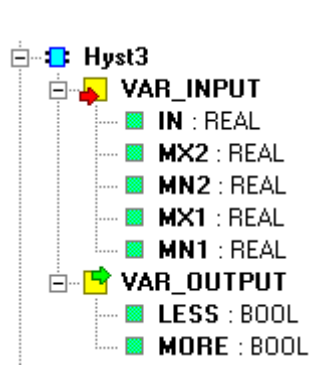


Obr. 2.13 Graf průběhů FB Hyst2

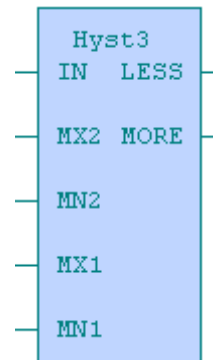
### 2.2.3. Hyst3 - dvojitá hystereze MIN,MAX

Pokud měřená hodnota *IN* překročí požadovanou maximální hodnotu *MX2*, je výstupní binární signál *LESS* nastaven do log. "1". V případě, že měřená hodnota *IN* podkročí požadovanou minimální hodnotu *MN2*, je výstupní signál *LESS* nastaven do log. "0".

Pokud měřená hodnota *IN* podkročí požadovanou minimální hodnotu *MN1*, je výstupní binární signál *MORE* nastaven do log. "1". V případě, že měřená hodnota *IN* překročí požadovanou maximální hodnotu *MX1*, je výstupní signál *MORE* nastaven do log. "0".



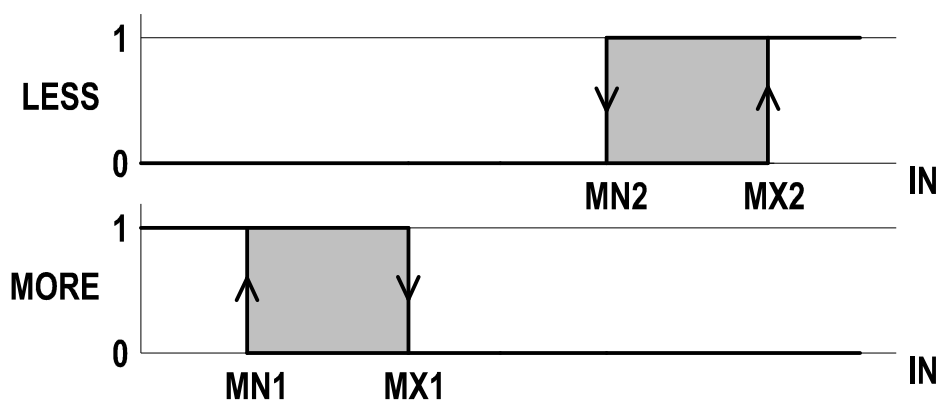
Obr. 2.14 Struktura FB Hyst3



Obr. 2.15 Vzhled FB Hyst3

Popis proměnných :

Název	Význam	Typ	Formát
IN	měřená hodnota	vstupní	real
MX2	maximum 2		real
MN2	minimum 2		real
MX1	maximum1		real
MN1	minimum 1		real
LESS	výstup „méně“	výstupní	bool
MORE	výstup „více“		bool



Obr. 2.16 Graf průběhů FB Hyst3

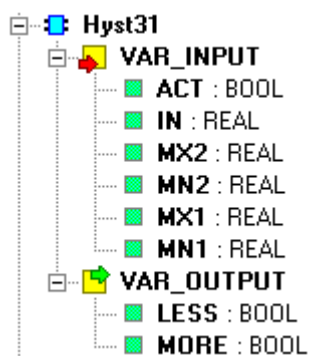
2.2.4. Hyst31 - dvojitá hystereze MIN,MAX s řídicí proměnnou

Pokud je vstupní proměnná *ACT* v log.1, funguje tento funkční blok stejně jako Hyst3, tzn :

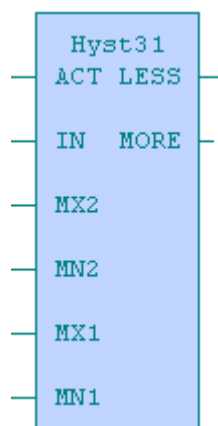
Pokud měřená hodnota *IN* překročí požadovanou maximální hodnotu *MX2*, je výstupní binární signál *LESS* nastaven do log. "1". V případě, že měřená hodnota *IN* podkročí požadovanou minimální hodnotu *MN2*, je výstupní signál *LESS* nastaven do log. "0".

Pokud měřená hodnota *IN* podkročí požadovanou minimální hodnotu *MN1*, je výstupní binární signál *MORE* nastaven do log. "1". V případě, že měřená hodnota *IN* překročí požadovanou maximální hodnotu *MX1*, je výstupní signál *MORE* nastaven do log. "0".

Pokud je vstupní proměnná *ACT* v log.0, pak jsou i výstupy *LESS* i *MORE* nastaveny do log.0.



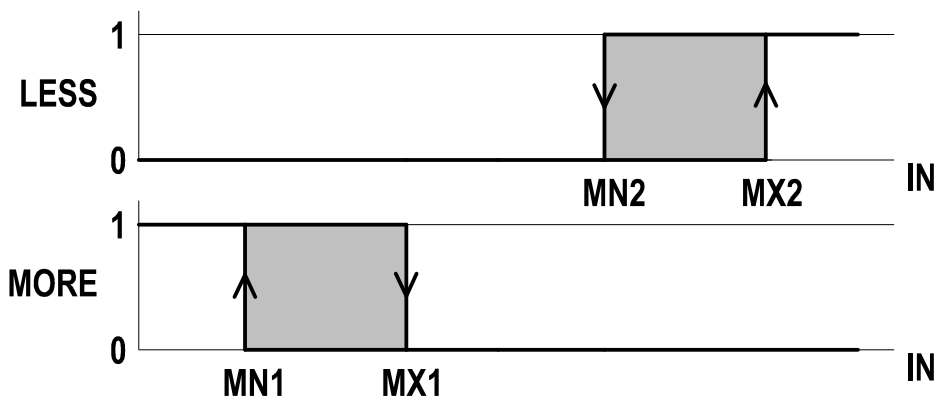
Obr. 2.17 Struktura FB Hyst31



Obr. 2.18 Vzhled FB Hyst31

Popis proměnných :

Název	Význam	Typ	Formát
ACT	řídící proměnná	vstupní	bool
IN	měřená hodnota		real
MX2	maximum 2		real
MN2	minimum 2		real
MX1	maximum1		real
MN1	minimum 1		real
LESS	výstup „méně“	výstupní	bool
MORE	výstup „více“		bool



Obr. 2.19 Graf průběhů FB Hyst31 (při ACT= 1)

## 2.3. PID REGULÁTORY

### 2.3.1. Obecný popis

PID regulátor je u systémů PLC Tecomat implementován přímo v instrukčním souboru jednotlivých PLC. Jednotlivé funkční bloky PID regulací s touto instrukcí pracují tak, aby programátorovi zjednodušily nastavení a obsluhu regulátoru. Z celé struktury regulátoru nabízejí pouze vybrané proměnné a do některých proměnných nastavují implicitní hodnoty.

Regulátor pracuje podle diskretní verze rovnice

$$u(t) = K * \left[ e(t) + \frac{1}{Ti} \int_0^t e(\tau) d\tau + Td * \frac{de(t)}{dt} \right]$$

#### Obecná struktura proměnných:

- MinY** - minimální měřená hodnota, používá se pro normalizaci odchylky
- MaxY** - maximální měřená hodnota, používá se pro normalizaci odchylky
- Input1** - měřená (regulovaná) veličina
- gW** - žádaná hodnota, ležící v intervalu měřené veličiny <MinY,MaxY>
- tiW** - časová konstanta pro filtr 1. řádu nebo lineární interpolaci žádané hodnoty v násobcích cyklu automatu
- ConW** - žádaná hodnota současná
- Dev (e)** - odchylka skutečné hodnoty od žádané [%]
- Output** - výstup žádaný algoritmem nebo manuálně. Akční zásah může ležet maximálně v rozsahu -10000 až +10000 (tj. -100,00% až +100,00%). Je tedy normován tak, že pro zesílení 1 (pásmo proporcionality 100%) a odchylku 100,00% je zásah 100,00%. Rozsah je vždy omezen v rozsahu <MinU, MaxU>.
- LastOut** - minulý akční zásah, tj. o 1 krok zpožděný [%] nebo poloha ventilu
- CurOut** - výstup skutečně žádaný v daném kroku [%] nebo přírůstek akčního zásahu.
- ConOut** - výstup regulátorem realizovaný [%] nebo skutečná hodnota realizovaná výstupní jednotkou nebo časově proporcionalním řízením on/off v absolutní hodnotě
- DefOut** - Implicitní hodnota výstupu při chybě měření
- MinU** - minimální povolený akční zásah 0-10000 [ 0-100,00 % ]. Akční zásah přímý nemůže být menší než tato hodnota.

## 2.KNIHOVNA REGOLIB

- MaxU** - maximální povolený akční zásah 0-10000 [ 0-100,00 % ]. Akční zásah přímý nemůže být větší než tato hodnota.
- dMaxU** - maximální povolený přírůstek akčního zásahu 0-10000 [ 0-100,00 %]. Nový akční zásah se nemůže v absolutní hodnotě lišit o více než dMaxU od minulé hodnoty.
- OutCycle** - délka výstupního cyklu, perioda vzorkování [setiny s] v rozsahu 1 až 60000 ( tj. 10 ms až 10 min po 10 ms ). Určuje periodu, po kterou se nemění akční zásah, respektive periodu opakovacího kmitočtu pro časově proporcionální řízení
- Pbnd** - pásmo proporcionality, nastavuje se v rozsahu 1 až 30000 ( 0,1 až 3000,0% ). Určuje zesílení vztahem
- $$K = \frac{1000}{PBnd}$$
- RelCool** - pomocné pásmo proporcionality pro zápornou odchylku, nastavuje se v rozsahu 1 až 30000 ( 0,1 až 3000,0% ). Zesílení je pak určeno vztahem
- $$K = \frac{1000}{PBnd} * \frac{1000}{RelCool}$$
- Ti** - integrační konstanta, nastavuje se v rozsahu 1 až 30000 ( 0,1 až 3000,0 s ). Pro nulovou hodnotu je integrační složka vypnutá.
- Td** - derivační konstanta, nastavuje se v rozsahu 1 až 30000 ( 0,1 až 3000,0 s ).
- Egap** - symetrické pásmo necitlivosti, nastavuje se v rozsahu 0 až 10000 ( 0 až 100,00% ). Je-li odchylka menší než EGap, zůstává akční zásah neměnný.
- Dgap** - symetrické pásmo odchylky, ve kterém působí derivační složka, rozsah je 0 až 10000 ( 0 až 100,00% ). To znamená, že derivační složka působí stále pro DGap = 10000 .
- Igap** - symetrické pásmo odchylky, ve kterém působí integrační složka, rozsah je 0 až 10000 ( 0 až 100,00% ). To znamená, že integrační složka působí stále pro IGap = 10000 .
- Control** - řídicí slovo slouží k nastavení činnosti regulátoru. Regulátor se může nacházet v režimu automatickém, ručním nebo havarijním. Může pracovat jako regulátor s přímým nebo s přírůstkovým algoritmem. Je-li jako akční orgán použitý servoventil, je možné použít pro korekci akčního zásahu naměřenou hodnotu jeho polohy, tj. jedná se o kaskádní řízení. Za předpokladu delšího výstupního cyklu je možné realizovat časově proporcionální řízení výstupu on/off. Rozlišení je dáno dobou cyklu automatu. Např. je-li doba cyklu automatu 100ms a výstupní cyklus 10s, rozlišení je 1%.

.15	.14	.13	.12	.11	.10	.9	.8	.7	.6	.5	.4	.3	.2	.1	.0
FU2	FU1	FU0	-	-	P41	RIO	RF	HR	AM	IP	BU	KC	A12	AO	RC

- RC - 1 - žádost o studený start regulátoru (komponenta sama nuluje bit)
- AO - 1 - posun nuly výstupu regulátoru pro rozsah 4-20mA
- A12 - 1 - výstup na dvanácti bitový převodník D/A

## Regulační knihovny pro Mosaic

KC	- 1 - kaskádní řízení
BU	- 0 - unifikovaný výstup 1 - binární výstup (časově proporcionální, řízení on/off)
IP	- 0 - přímé řízení 1 - přírůstkové řízení
AM	- 0 - ruční režim 1 - automatický režim
HR	- 1 - režim spolehlivějšího měření, využívá dvě měřené hodnoty
RF	- 0 - úprava žádané hodnoty filtrem 1. řádu 1 - úprava žádané hodnoty lineární interpolací
RIO	- 1 - poměrová regulace
P41	- 1 - instrukce PID se volá v procesu P41, tj. v rastru 10 ms

*Poznámka: I v tomto případě je možné použít nastavení periody v proměnné OutCycle. Praktický význam má toto jen pro řízení typu on / off. (Např. je-li OutCycle = 100, je perioda 1s a rozlišení šířky výstupního pulzu je 10ms, tj. 1%. Při použití např. výstupní jednotky 220V, je možné takto realizovat výstup typu cyklického řízení, tj. s rozlišením jedné periody fázového napětí.*

FU2-FU0	- filtrace krátkých akčních zásahů. Obecně platí, že pokud $CurOut < 32 * FU$ , zásah se neprovede a obsah CurOut je vynulován. 0 - všechny akční zásahy povoleny 1 - potlačeny akční zásahy menší než 32 (tj. 0,32%) 2 - potlačeny akční zásahy menší než 64 (tj. 0,64%) 3 - potlačeny akční zásahy menší než 96 (tj. 0,96%) 4 - potlačeny akční zásahy menší než 128 (tj. 1,28%) 5 - potlačeny akční zásahy menší než 160 (tj. 1,6%) 6 - potlačeny akční zásahy menší než 192 (tj. 1,92%) 7 - potlačeny akční zásahy menší než 224 (tj. 2,24%)
---------	--

**Status** - Slouží zejména pro přenos hodnoty bitů pro on/off řízení, tedy pokud je akční zásah řešen jako časově proporcionální řízení (šířka pulsů). Dále obsahuje chybové bity měření.

.7	.6	.5	.4	.3	.2	.1	.0
-	EY3	EY2	EY1	DR	U-	UC	UH

.0 (UH)	- výstup pro kladný akční zásah, tj. topení
.1 (UC)	- výstup pro záporný akční zásah, tj. chlazení
.2 (U-)	- signalizace akčního zásahu 0 - kladný akční zásah 1 - záporný akční zásah
.3 (DR)	- detekce průběhu lineární interpolace žádané hodnoty 1 - interpolace je aktivní
.4 (EY1)	- detekce chyby měření y1 (Input1) 1 - y1 mimo interval <MinY, MaxY>
.5 (EY2)	- detekce chyby měření y1 (Input2) 1 - y2 mimo interval <MinY, MaxY>
.6 (EY3)	- detekce chyby měření y1 (Input3) 1 - y3 mimo interval <MinY, MaxY>

**AuxD** - pomocné proměnné regulátoru. Zápis do této zóny je zakázán!!

### 2.3.2. PID1 - regulátor s přírůstkovým řízením

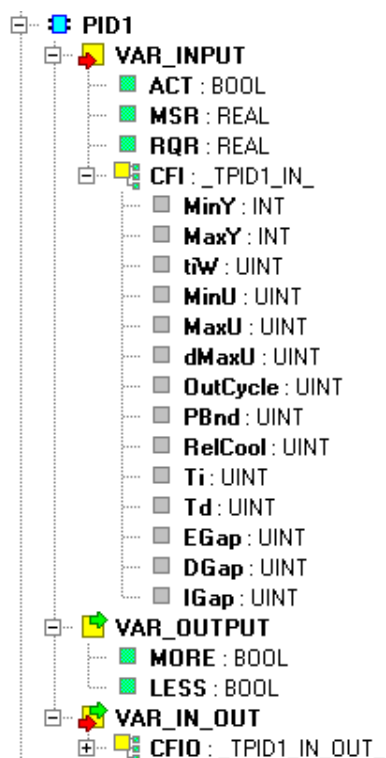
Pokud je proměnná *ACT* v log.1, jsou akceptovány ostatní vstupní proměnné funkčního bloku a je aktivován PID algoritmus regulátoru - regulace hodnoty měřené *MSR* na hodnotu žádanou *RQR*. V proměnných *MORE* a *LESS* jsou předávány povely pro akční orgán (regulační ventil).

Funkční blok nastavuje tyto proměnné obecné PID struktury :

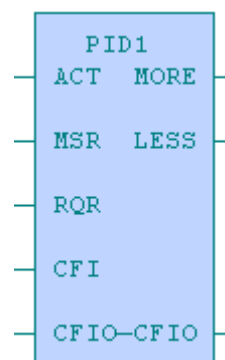
```
MSR      = Input1
RQR      = gW
MORE     = Status.0
LESS     = Status.1
CONTROL = $0071 (pouze při startu regulace)
```

Pokud nejsou do vstupní struktury funkčního bloku *CFI* přiřazeny uživatelské hodnoty, přednastaví funkční blok při studeném restartu do struktury tyto hodnoty :

```
CFI :=
  (MinY := 0, // Hodnota unifikovaného rozsahu odpovídající 0%
   MaxY := 1000, // Hodnota unifikovaného rozsahu odpovídající 100%
   tiW := 0, // Časová konstanta filtru žádané hodnoty
   MinU := 0, // Minimální povolený akční zásah
   MaxU := 10000, // Maximální povolený akční zásah
   dMaxU := 1000, // Maximální přírůstek akční veličiny za jednu periodu
   OutCycle:= 820, // Perioda vzorkování regulačního algoritmu [10ms]
   PBnd := 500, // Pásmo proporcionality
   RelCool := 1000, // Relativní pásmo proporcionality pro záporné regulační odchylky
   Ti := 740, // Integrační časová konstanta
   Td := 26, // Derivační časová konstanta
   EGap := 10, // Symetrické pásmo necitlivosti
   DGap := 10000, // Symetrické pásmo působení derivační složky
   IGap := 10000); // Symetrické pásmo působení integrační složky
```



Obr. 2.20 Struktura FB PID1



Obr. 2.21 Vzhled FB PID1

Popis proměnných :

Název	Význam	Typ	Formát
ACT	aktivace	vstupní	bool
MSR	měřená hodnota		real
RQR	žádaná hodnota		real
CFI	vstupní řídicí struktura		_TPID1_IN_
.MinY	hodnota unifikovaného rozsahu pro 0%		int
.MaxY	hodnota unifikovaného rozsahu pro 100%		int
.tiW	časová konstanta filtru žádané hodnoty		uint
.MinU	minimální povolený akční zásah		uint
.MaxU	maximální povolený akční zásah		uint
.dMAxU	maximální přírůstek akční veličiny za periodu		uint
.OutCycle	perioda vzorkování regulačního algoritmu[10ms]		uint
.PBnd	pásmo proporcionality		uint
.RelCool	relativní pásmo proporcionality pro záporné hodnoty regulační odchylky		uint
.Ti	integrační časová konstanta		uint
.Td	derivační časová konstanta		uint
.Egap	symetrické pásmo necitlivosti		uint
.DGap	symetrické pásmo působení derivační složky		uint
.IGap	symetrické pásmo působení integrační složky	uint	
MORE	kladný akční zásah	výstupní	bool
LESS	záporný akční zásah		bool
CFIO	vstupně/výstupní řídicí struktura	vstup/ výstupní	_TPID1_IN_OUT_ _TPID_Control_
.Control	řídicí slovo regulátoru (viz. kap.2.3.1.)		

### 2.3.3. PID11 - regulátor s přírůstkovým řízením a užším výběrem proměnných

Pokud je proměnná *ACT* v log.1, jsou akceptovány ostatní vstupní proměnné funkčního bloku a je aktivován PID algoritmus regulátoru - regulace hodnoty měřené *MSR* na hodnotu žádanou *RQR*. V proměnných *MORE* a *LESS* jsou předávány povely pro akční orgán (regulační ventil).

Funkční blok tedy pracuje stejně jako PID1, rozdíl je pouze v užším výběru nastavitelných proměnných. Proměnné *tiW*, *RelCool*, *DGap*, *IGap* jsou nastavovány do pevných hodnot.

Funkční blok nastavuje tyto proměnné obecné PID struktury :

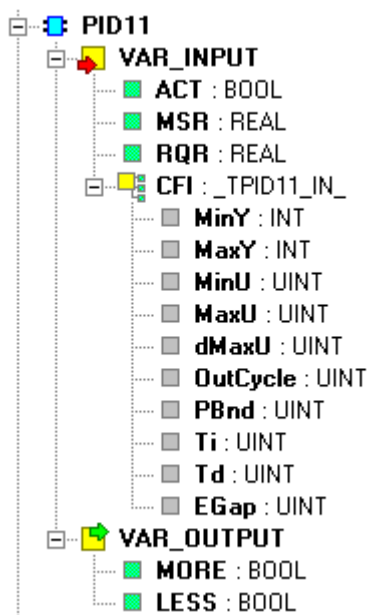
```

MSR      = Input1
RQR      = gW
MORE     = Status.0
LESS     = Status.1
CONTROL  = $0071 (pouze při startu regulace)
tiW      = 0
RelCool  = 1000
DGap     = 10000
IGap     = 10000
    
```

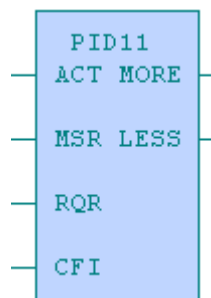
Pokud nejsou do vstupní struktury funkčního bloku *CFI* přiřazeny uživatelské hodnoty, přednastaví funkční blok při studeném restartu do struktury tyto hodnoty :



```
CFI :=
  (MinY := 0, // Hodnota unifikovaného rozsahu odpovídající 0%
   MaxY := 1000 // Hodnota unifikovaného rozsahu odpovídající 100%
   MinU := 0, // Minimální povolený akční zásah
   MaxU := 10000, // Maximální povolený akční zásah
   dMaxU := 1000, // Maximální přírůstek akční veličiny za jednu periodu
   OutCycle:= 1000, // Perioda vzorkování regulačního algoritmu [10ms]
   PBnd := 1000, // Pásmo proporcionality
   RelCool := 1000, // Relativní pásmo proporcionality pro záporné regulační odchylky
   Ti := 1000, // Integrovaná časová konstanta
   Td := 0, // Derivační časová konstanta
   EGap := 10) // Symetrické pásmo necitlivosti
```



Obr. 2.22 Struktura FB PID11



Obr. 2.23 Vzhled FB PID11

Popis proměnných :

Název	Význam	Typ	Formát
ACT	aktivace	vstupní	bool
MSR	měřená hodnota		real
RQR	žádaná hodnota		real
CFI	vstupní řídicí struktura		_TPID11_IN_
.MinY	hodnota unifikovaného rozsahu pro 0%		int
.MaxY	hodnota unifikovaného rozsahu pro 100%		int
.MinU	minimální povolený akční zásah		uint
.MaxU	maximální povolený akční zásah		uint
.dMAXU	maximální přírůstek akční veličiny za periodu		uint
.OutCycle	perioda vzorkování regulačního algoritmu [10ms]		uint
.PBnd	pásmo proporcionality		uint
.Ti	integrovaná časová konstanta		uint
.Td	derivační časová konstanta		uint
.Egap	symetrické pásmo necitlivosti	uint	
MORE	kladný akční zásah	výstupní	bool
LESS	záporný akční zásah		bool

### 2.3.4. PID2 - regulátor s přímým řízením

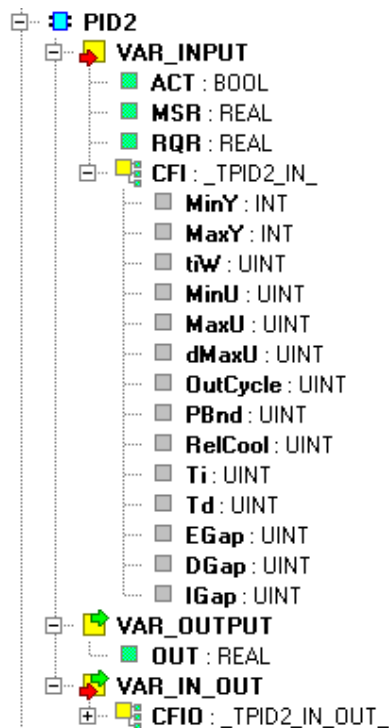
Pokud je proměnná *ACT* v log.1 jsou akceptovány ostatní vstupní proměnné a je aktivován algoritmus regulátoru - regulace hodnoty měřené *MSR* na hodnotu žádanou *RQR*. V proměnné *OUT* je předáván požadovaný akční zásah pro akční orgán (regulační ventil) v rozsahu 0-100%, odpovídající PCT formátu analogových výstupních karet.

Funkční blok nastavuje tyto proměnné obecné PID struktury :

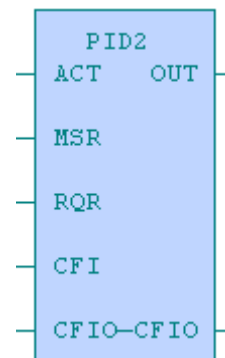
MSR = Input1  
 RQR = gW  
 OUT = Output  
 CONTROL = \$0041 (pouze při startu regulace)

Pokud nejsou do vstupní struktury funkčního bloku *CFI* přiřazeny uživatelské hodnoty, přednastaví funkční blok při studeném restartu do struktury tyto hodnoty :

CFI :=  
 (MinY := 0, // Hodnota unifikovaného rozsahu odpovídající 0%  
 MaxY := 1000, // Hodnota unifikovaného rozsahu odpovídající 100%  
 tiW := 0, // Časová konstanta filtru žádané hodnoty  
 MinU := 0, // Minimální povolený akční zásah  
 MaxU := 10000, // Maximální povolený akční zásah  
 dMaxU := 1000, // Maximální přírůstek akční veličiny za jednu periodu  
 OutCycle:= 820, // Perioda vzorkování regulačního algoritmu [10ms]  
 PBnd := 500, // Pásmo proporcionality  
 RelCool := 1000, // Relativní pásmo proporcionality pro záporné regulační odchylky  
 Ti := 740, // Integrační časová konstanta  
 Td := 26, // Derivační časová konstanta  
 EGap := 10, // Symetrické pásmo necitlivosti  
 DGap := 10000, // Symetrické pásmo působení derivační složky  
 IGap := 10000); // Symetrické pásmo působení integrační složky



Obr. 2.24 Struktura FB PID2



Obr. 2.25 Vzhled FB PID2

Popis proměnných :

Název	Význam	Typ	Formát
ACT	aktivace	vstupní	bool
MSR	měřená hodnota		real
RQR	žádaná hodnota		real
CFI	vstupní řídicí struktura		_TPID2_IN_
.MinY	hodnota unifikovaného rozsahu pro 0%		int
.MaxY	hodnota unifikovaného rozsahu pro 100%		int
.tiW	časová konstanta filtru žádané hodnoty		uint
.MinU	minimální povolený akční zásah		uint
.MaxU	maximální povolený akční zásah		uint
.dMAxU	maximální přírůstek akční veličiny za periodu		uint
.OutCycle	perioda vzorkování regulačního algoritmu [10ms]		uint
.PBnd	pásmo proporcionality		uint
.RelCool	relativní pásmo proporcionality pro záporné hodnoty regulační odchylky		uint
.Ti	integrační časová konstanta		uint
.Td	derivační časová konstanta		uint
.Egap	symetrické pásmo necitlivosti		uint
.DGap	symetrické pásmo působení derivační složky	uint	
.IGap	symetrické pásmo působení integrační složky	uint	
OUT	akční zásah	výstupní	real
CFIO	vstupně/výstupní řídicí struktura	vstup/ výstupní	_TPID2_IN_OUT_ _TPID_Control_
.Control	řídicí slovo regulátoru (viz. kap.2.3.1.)		

### 2.3.5. PID21 - regulátor s přímým řízením a užším výběrem proměnných

Pokud je proměnná *ACT* v log.1 jsou akceptovány ostatní vstupní proměnné a je aktivován algoritmus regulátoru - regulace hodnoty měřené *MSR* na hodnotu žádanou *RQR*. V proměnné *OUT* je předáván požadovaný akční zásah pro akční orgán (regulační ventil) v rozsahu 0-100%, odpovídající PCT formátu analogových výstupních karet.

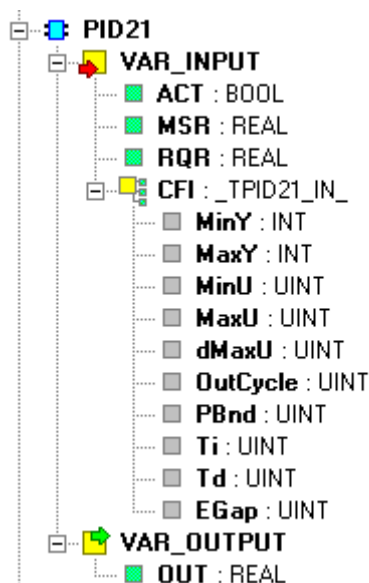
Funkční blok tedy pracuje stejně jako PID2, rozdíl je pouze v užším výběru nastavitelných proměnných. Proměnné *tiW*, *RelCool*, *DGap*, *IGap* jsou nastavovány do pevných hodnot.

Funkční blok nastavuje tyto proměnné obecné PID struktury :

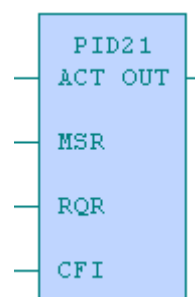
*MSR* = Input1  
*RQR* = gW  
*OUT* = Output  
*CONTROL* = \$0041 (pouze při startu regulace)  
*tiW* = 0  
*RelCool* = 1000  
*DGap* = 10000  
*IGap* = 10000

Pokud nejsou do vstupní struktury funkčního bloku *CFI* přiřazeny uživatelské hodnoty, přednastaví funkční blok při studeném restartu do struktury tyto hodnoty :

```
CFI :=
  (MinY := 0, // Hodnota unifikovaného rozsahu odpovídající 0%
   MaxY := 1000 // Hodnota unifikovaného rozsahu odpovídající 100%
   MinU := 0, // Minimální povolený akční zásah
   MaxU := 10000, // Maximální povolený akční zásah
   dMaxU := 1000, // Maximální přírůstek akční veličiny za jednu periodu
   OutCycle:= 1000, // Perioda vzorkování regulačního algoritmu [10ms]
   PBnd := 1000, // Pásmo proporcionality
   RelCool := 1000, // Relativní pásmo proporcionality pro záporné regulační odchylky
   Ti := 1000, // Integrovaná časová konstanta
   Td := 0, // Derivační časová konstanta
   EGap := 10) // Symetrické pásmo necitlivosti
```



Obr. 2.26 Struktura FB PID21



Obr. 2.27 Vzhled FB PID21

Popis proměnných :

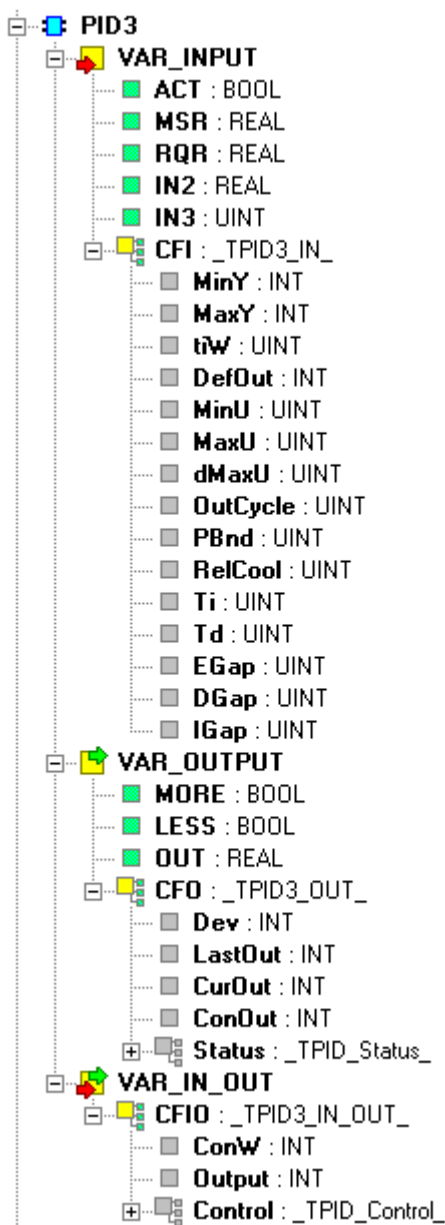
Název	Význam	Typ	Formát
ACT	aktivace	vstupní	bool
MSR	měřená hodnota		real
RQR	žádaná hodnota		real
CFI	vstupní řídicí struktura		_TPID21_IN_
.MinY	hodnota unifikovaného rozsahu pro 0%		int
.MaxY	hodnota unifikovaného rozsahu pro 100%		int
.MinU	minimální povolený akční zásah		uint
.MaxU	maximální povolený akční zásah		uint
.dMAXU	maximální přírůstek akční veličiny za periodu		uint
.OutCycle	perioda vzorkování regulačního algoritmu [10ms]		uint
.PBnd	pásmo proporcionality		uint
.Ti	integrovaná časová konstanta		uint
.Td	derivační časová konstanta		uint
.Egap	symetrické pásmo necitlivosti	uint	
OUT	akční zásah	výstupní	real

### 2.3.6. PID3 - volně nastavitelný regulátor

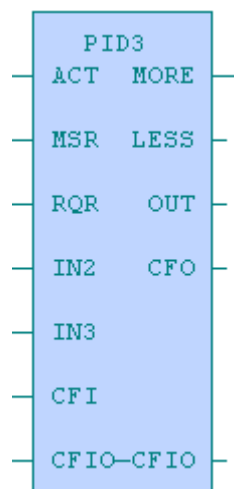
Tento funkční blok umožňuje přístup do všech proměnných obecné datové struktury PID regulace (viz. kap.2.3.1.) a umožňuje tak uživateli libovolné využití všech možností PID regulace.

Funkční blok nastavuje tyto proměnné obecné PID struktury :

- MSR = Input1
- RQR = gW
- MORE = Status.0
- LESS = Status.1
- OUT = Output



Obr. 2.28 Struktura FB PID3



Obr. 2.29 Vzhled FB PID3

## Regulační knihovny pro Mosaic

Popis proměnných :

Název	Význam	Typ	Formát
ACT	aktivace	vstupní	bool
MSR	měřená hodnota		real
RQR	žádaná hodnota		real
IN2	poměrová regulace		real
IN3	poloha servoventilu		uint
CFI	vstupní proměnné PID struktury		_TPID3_IN_
.MinY	hodnota unifikovaného rozsahu pro 0%		int
.MaxY	hodnota unifikovaného rozsahu pro 100%		int
.tiW	časová konstanta filtru žádané hodnoty		uint
.DefOut	předdefinovaný akční zásah při poruše		int
.MinU	minimální povolený akční zásah		uint
.MaxU	maximální povolený akční zásah		uint
.dMAxU	maximální přírůstek akční veličiny za periodu		uint
.OutCycle	perioda vzorkování regulačního algoritmu [10ms]		uint
.PBnd	pásmo proporcionality		uint
.RelCool	relativní pásmo proporcionality pro záporné hodnoty regulační odchylky		uint
.Ti	integrační časová konstanta		uint
.Td	derivační časová konstanta		uint
.Egap	symetrické pásmo necitlivosti		uint
.DGap	symetrické pásmo působení derivační složky		uint
.IGap	symetrické pásmo působení integrační složky	uint	
MORE	kladný akční zásah	výstupní	bool
LESS	záporný akční zásah		bool
OUT	akční zásah		real
CFO	výstupní proměnné PID struktury		_TPID3_OUT_
.Dev	regulační odchylka		int
.LastOut	minulý akční zásah		int
.CurOut	skutečně žádaný akční zásah		int
.ConOut	realizovaný akční zásah	int	
.Status	stavové slovo regulátoru (viz. kap.2.3.1.)	_TPID_Status_	
CFIO	vstupně/výstupní proměnné PID struktury	vstup/ výstupní	_TPID3_IN_OUT_
.ConW	aktuální žádaná hodnota		int
.Output	akční zásah		int
.Control	řídící slovo regulátoru (viz. kap.2.3.1.)		_TPID_Control_

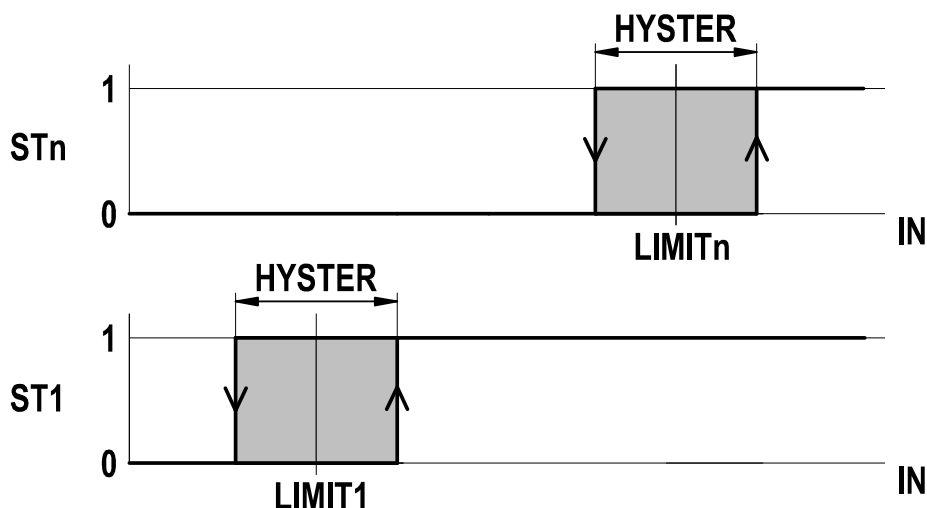
## 2.4. KASKÁDNÍ ŘAZENÍ

Funkční bloky kaskádního řazení pracují podle níže uvedeného popisu. Jednotlivé funkční bloky kaskádního řazení se liší pouze počtem spínaných kaskádních stupňů (2 až 5).

Pokud je proměnná *ACT* v log.1 jsou na základě vstupní proměnné *IN* ovládány jednotlivé stupně kaskády *STx*. Stupně jsou spínány a odepínány podle jednotlivých mezí *LIMITx* (vztažených ke vstupní proměnné *IN*) a nastavené hystereze *HYSTER*. V případě poruchy některého ze stupňů *ERRx* je tento vypnut a nahrazen jiným (dalším v kaskádě).

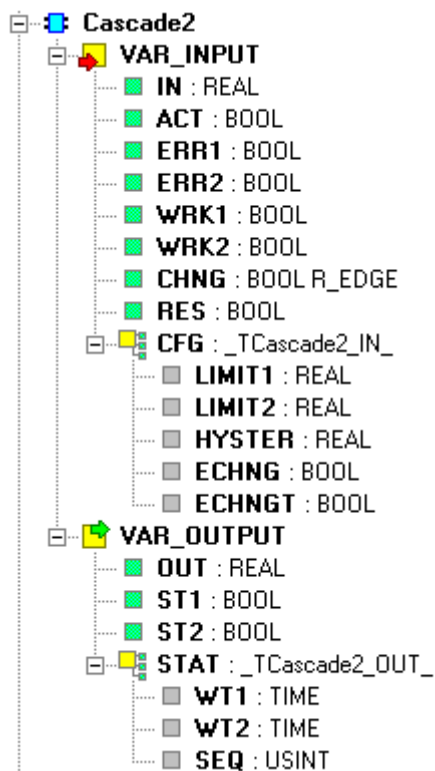
Komponenta dále obsahuje funkci pro změnu pořadí spínání stupňů. Pokud je proměnná *ECHNG* v log.1, je na náběžnou hranu binárního signálu *CHNG* změněno pořadí stupňů. Jednotlivé stupně jsou řazeny tak, aby jejich použití bylo rovnoměrné (viz. tabulka pořadí). V případě, že je navíc proměnná *ECHNGT* v log.1, jsou stupně kaskády řazeny podle naběhaných provozních hodin *WTx* tak, aby ten s nejmenším počtem hodin byl v kaskádě na prvním místě. Čítání provozních hodin stupňů kaskády je aktivováno nastavením příslušné proměnné *WRKx* do log.1 v době, kdy je daný stupeň v chodu. Signálem *RES* se provozní hodiny všech stupňů kaskády nulují.

Vstup *IN* se připojuje na spojitý výstup regulačních PID funkčních bloků a může nabývat hodnot v rozsahu od 0 do 100[%]. Hodnota vstupu *IN* je pro potřeby dalšího kaskádování kopírována na výstup *OUT*.

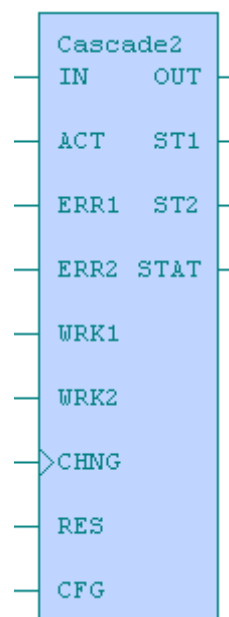


Obr. 2.30 Graf průběhů FB Cascade

### 2.4.1. Cascade2 - kaskáda 2 stupňů se střídáním



Obr. 2.31 Struktura FB Cascade2



Obr. 2.32 Vzhled FB Cascade2

Popis proměnných :

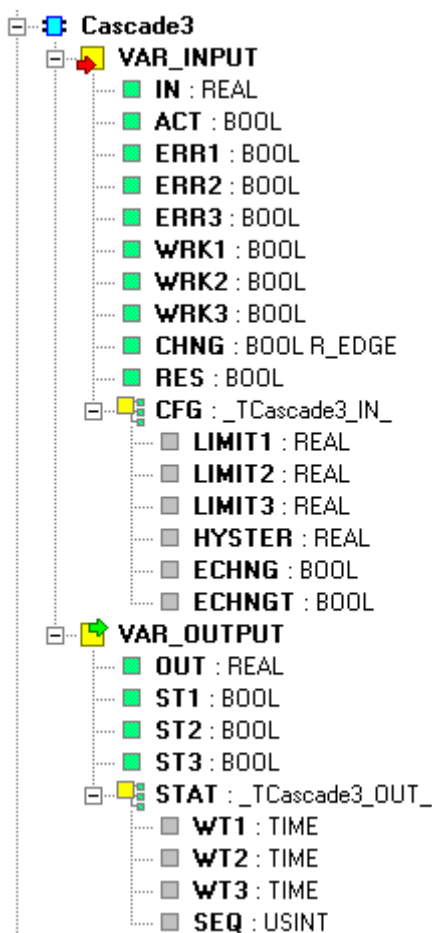
Název	Význam	Typ	Formát	
IN	vstupní požadovaný výkon	vstupní	real	
ACT	aktivace činnosti		bool	
ERRx	porucha x.stupně kaskády		bool	
WRKx	chod x.stupně kaskády		bool	
CHNG	změna pořadí stupňů kaskády		bool r_edge	
RES	reset provozních hodin		bool	
CFG	řídící struktura		_Tcascade2_IN_	
	.LIMITx		mez spínání/vypínání x. stupně kaskády	real
	.HYSTER		hystereze spínání/vypínání	real
	.ECHNG		povolení střídání stupňů kaskády	bool
	.EGNGT	povolení střídání stupňů kaskády podle provozních hodin	bool	
OUT	výstupní požadovaný výkon	výstupní	real	
STx	x. kaskádní výstup		bool	
STAT	stavová struktura		_Tcascade2_OUT_	
	.WTx		provozní hodiny x. stupně kaskády	time
	.SEQ		pořadí stupňů kaskády	usint

Tabulka pořadí :

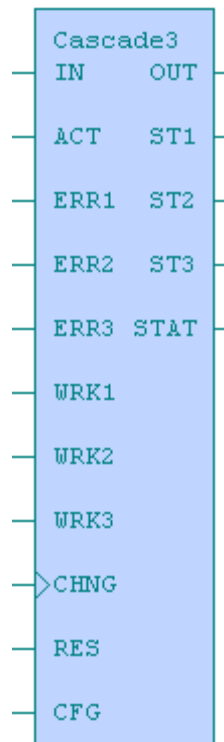
SEQ	Pořadí
0	12
1	21



2.4.2. Cascade3 - kaskáda 3 stupňů se střídáním



Obr. 2.33 Struktura FB Cascade3



Obr. 2.34 Vzhled FB Cascade3

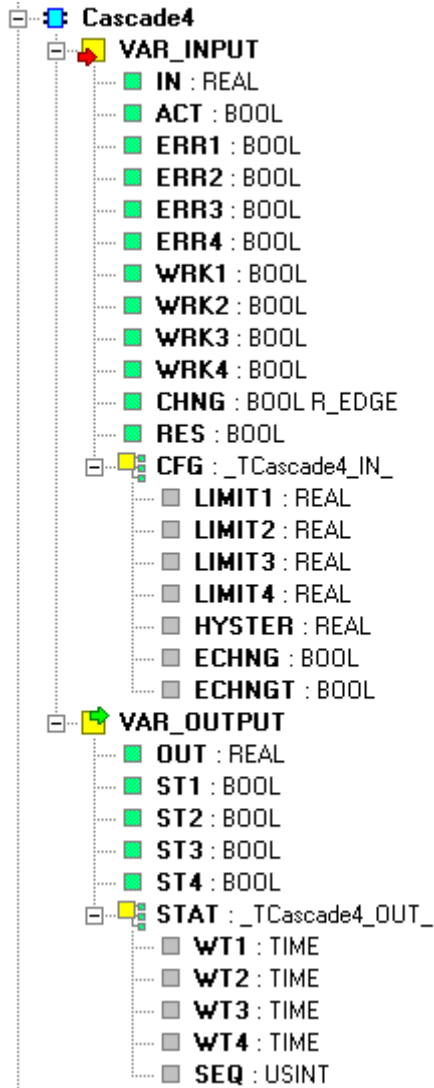
Popis proměnných :

Název	Význam	Typ	Formát	
IN	vstupní požadovaný výkon	vstupní	real	
ACT	aktivace činnosti		bool	
ERRx	porucha x.stupně kaskády		bool	
WRKx	chod x.stupně kaskády		bool	
CHNG	změna pořadí stupňů kaskády		bool r_edge	
RES	reset provozních hodin		bool	
CFG	řídící struktura		_Tcascade3_IN_	
	.LIMITx		mez spínání/vypínání x. stupně kaskády	real
	.HYSTER		hystereze spínání/vypínání	real
	.ECHNG		povolení střídání stupňů kaskády	bool
	.EGNGT	povolení střídání stupňů kaskády podle provozních hodin	bool	
OUT	výstupní požadovaný výkon	výstupní	real	
STx	x. kaskádní výstup		bool	
STAT	stavová struktura		_Tcascade3_OUT_	
	.WTx		provozní hodiny x. stupně kaskády	time
	.SEQ		pořadí stupňů kaskády	usint

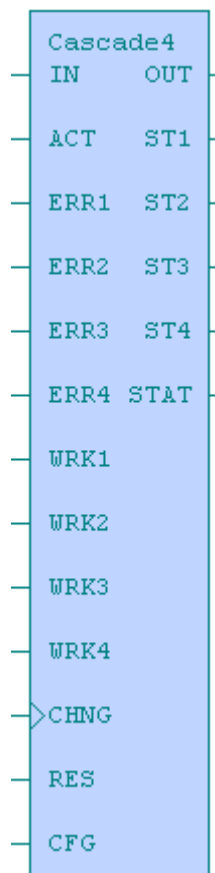
Tabulka pořadí :

SEQ	Pořadí
0	123
1	231
2	312

### 2.4.3. Cascade4 - kaskáda 4 stupňů se střídáním



Obr. 2.35 Struktura FB Cascade4



Obr. 2.36 Vzhled FB Cascade4

## 2.KNIHOVNA REGOLIB

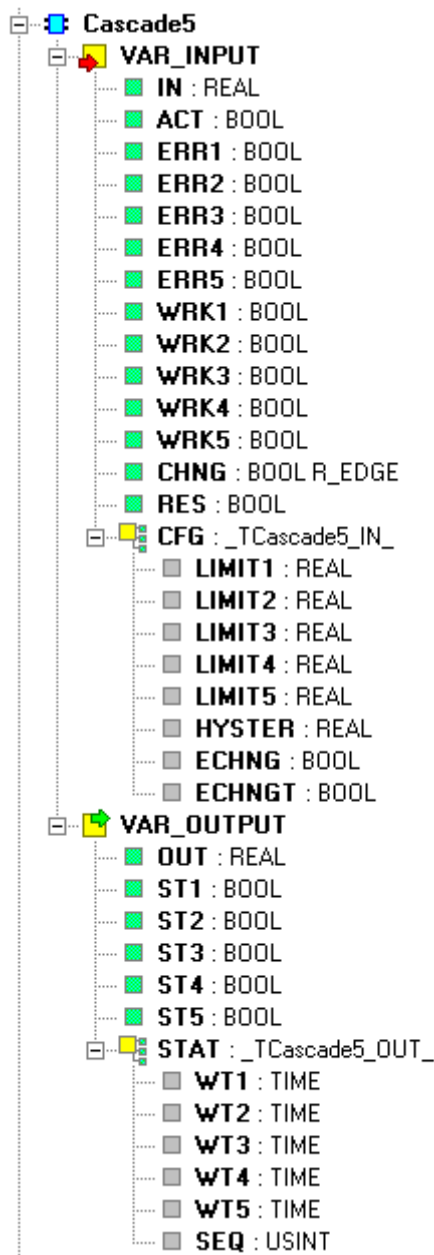
Popis proměnných :

Název	Význam	Typ	Formát	
IN	vstupní požadovaný výkon	vstupní	real	
ACT	aktivace činnosti		bool	
ERRx	porucha x.stupně kaskády		bool	
WRKx	chod x.stupně kaskády		bool	
CHNG	změna pořadí stupňů kaskády		bool r_edge	
RES	reset provozních hodin		bool	
CFG	řídící struktura		_Tcascade4_IN_	
	.LIMITx		mez spínání/vypínání x. stupně kaskády	real
	.HYSTER		hystereze spínání/vypínání	real
	.ECHNG		povolení střídání stupňů kaskády	bool
	.EGNGT	povolení střídání stupňů kaskády podle provozních hodin	bool	
OUT	výstupní požadovaný výkon	výstupní	real	
STx	x. kaskádní výstup		bool	
STAT	stavová struktura		_Tcascade4_OUT_	
	.WTx		provozní hodiny x. stupně kaskády	time
	.SEQ		pořadí stupňů kaskády	usint

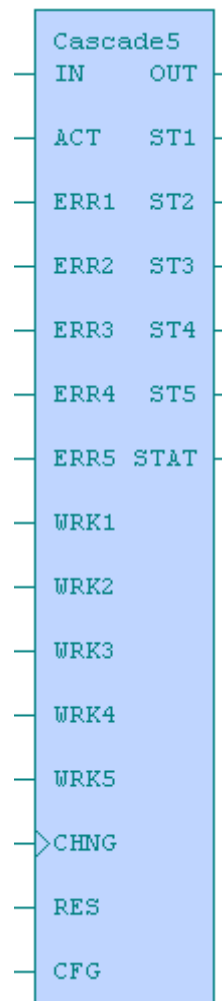
Tabulka pořadí :

SEQ	Pořadí
0	1234
1	2341
2	3412
3	4123

2.4.4. Cascade5 - kaskáda 5 stupňů se střídáním



Obr. 2.37 Struktura FB Cascade5



Obr. 2.38 Vzhled FB Cascade5

## 2.KNIHOVNA REGOLIB

Popis proměnných :

Název	Význam	Typ	Formát
IN	vstupní požadovaný výkon	vstupní	real
ACT	aktivace činnosti		bool
ERRx	porucha x.stupně kaskády		bool
WRKx	chod x.stupně kaskády		bool
CHNG	změna pořadí stupňů kaskády		bool r_edge
RES	reset provozních hodin		bool
CFG	řídící struktura		_Tcascade5_IN_
.LIMITx	mez spínání/vypínání x. stupně kaskády		real
.HYSTER	hystereze spínání/vypínání		real
.ECHNG	povolení střídání stupňů kaskády		bool
.EGNGT	povolení střídání stupňů kaskády podle provozních hodin	bool	
OUT	výstupní požadovaný výkon	výstupní	real
STx	x. kaskádní výstup		bool
STAT	stavová struktura		_Tcascade5_OUT_
.WTx	provozní hodiny x. stupně kaskády		time
.SEQ	pořadí stupňů kaskády		usint

Tabulka pořadí :

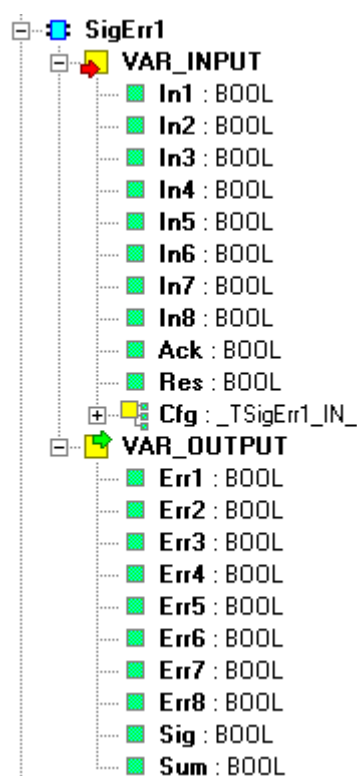
SEQ	Pořadí
0	12345
1	23451
2	34512
3	45123
4	51234

## 2.5. PORUCHOVÉ SIGNALIZACE

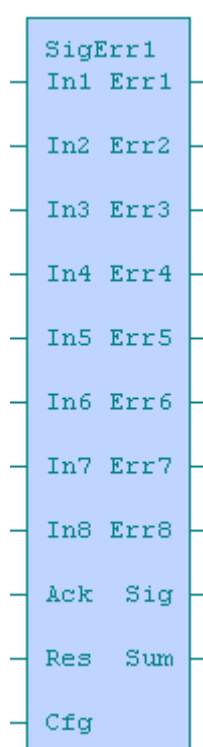
### 2.5.1. SigErr1 - binární poruchová signalizace

Funkční blok provádí vyhodnocení výskytu poruchy s nastaveným časovým zpožděním *PRESETTIME<sub>x</sub>* pro 8 binárních vstupů. Pokud je vstupní signál *IN<sub>x</sub>* aktivní déle než je nastavená předvolba, je výstupní signál *ERR<sub>x</sub>* nastaven do log. 1. Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné *SUM* a signalizaci nově vyhodnocené poruchy *SIG*.

Výskyt poruch je možné potvrdit (kvitovat) signálem *ACK* a neaktivní poruchy zrušit signálem *RES*. Každá nově vyhodnocená porucha rozblíká výstup pro optickou signalizaci *SIG* v intervalu 1 sec. Je-li po odkvitování (na vstupu *ACK* log.1) proměnná *SUM* v log.1, je optická signalizace *SIG* v log.1. V opačném případě je v log.0.



Obr. 2.39 Struktura FB SigErr1



Obr. 2.40 Vzhled FB SigErr1

Popis proměnných :

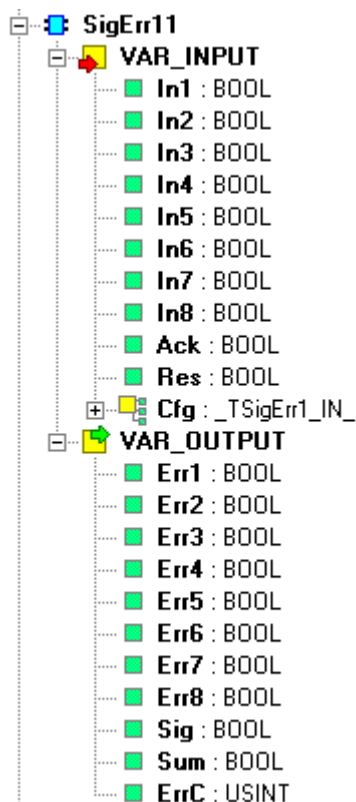
Název	Význam	Typ	Formát
InX	vstupní signál poruchy X	vstupní	bool
Ack	kvitace poruch		bool
Res	nulování poruch		bool
Cfg	konfigurační struktura bloku		<i>_TSigErr1_IN_</i>
<i>.PresetTimeX</i>	zpoždění vyhodnocení poruchy X		time
ErrX	výstupní signál poruchy X	výstupní	bool
Sig	signálka		bool
Sum	sdružená porucha		bool

2.5.2. SigErr11 - binární poruchová signalizace se signalizací čísla poruchy

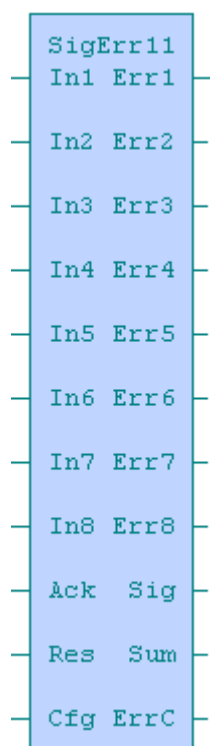
Funkční blok provádí vyhodnocení výskytu poruchy s nastaveným časovým zpožděním *PRESETTIME<sub>x</sub>* pro 8 binárních vstupů. Pokud je vstupní signál *IN<sub>x</sub>* aktivní déle než je nastavená předvoba, je výstupní signál *ERR<sub>x</sub>* nastaven do log. 1. Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné *SUM* a signalizaci nově vyhodnocené poruchy *SIG*.

Výskyt poruch je možné potvrdit (kvitovat) signálem *ACK* a neaktivní poruchy zrušit signálem *RES*. Každá nově vyhodnocená porucha rozblíká výstup pro optickou signalizaci *SIG* v intervalu 1 sec. Je-li po odkvitování (na vstupu *ACK* log.1) proměnná *SUM* v log.1, je optická signalizace *SIG* v log.1. V opačném případě je v log.0.

Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy *ERRC*, která je určena pro připojení na funkční blok historie poruch (History1,History5,History10).



Obr. 2.41 Struktura FB SigErr11



Obr. 2.42 Vzhled FB SigErr11

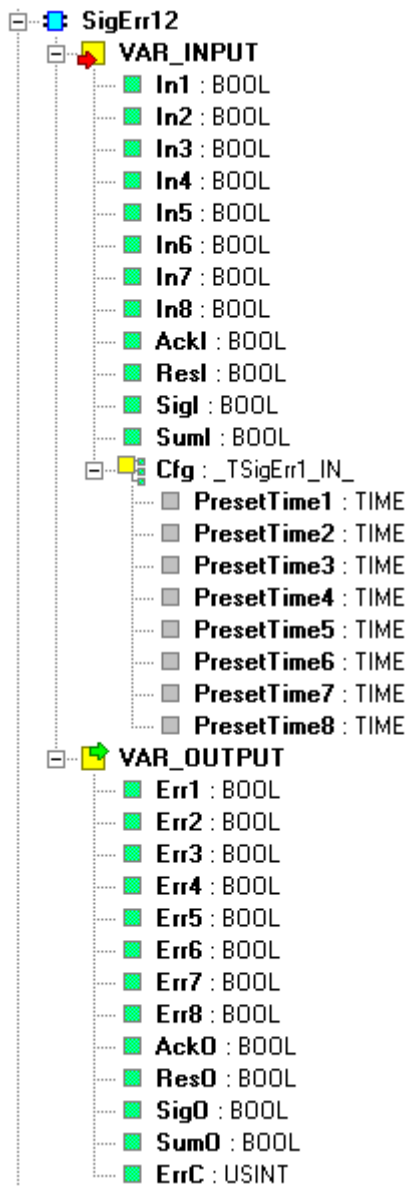
Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní signál poruchy X	vstupní	bool
Ack	kvitace poruch		bool
Res	nulování poruch		bool
Cfg	konfigurační struktura bloku		<i>_TSigErr1_IN_</i>
<i>.PresetTimeX</i>	zpoždění vyhodnocení poruchy X		time
ErrX	výstupní signál poruchy X	výstupní	bool
Sig	signálka		bool
Sum	sdužená porucha		bool
ErrC	číslo poruchy		usint

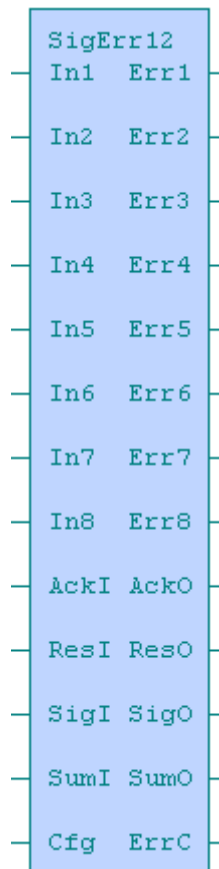
### 2.5.3. SigErr12 - binární poruchová signalizace s obsluhou propojení

Funkční blok provádí vyhodnocení výskytu poruchy s nastaveným časovým zpožděním *PRESETTIME<sub>x</sub>* pro 8 binárních vstupů. Pokud je vstupní signál *IN<sub>x</sub>* aktivní déle než je nastavená předvoba, je výstupní signál *ERR<sub>x</sub>* nastaven do log. 1. Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné *SUMO* a signalizaci nově vyhodnocené poruchy *SIGO*.

Výskyt poruch je možné potvrdit (kvitovat) signálem *ACKI* a zrušit signálem *RESI*. Proměnné *ACKO*, *RESO*, *SIGO*, *SUMO* slouží ke kaskádování více komponent. Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy *ERRC*, která je určena pro připojení na funkční blok historie poruch (History1,History5,History10). Každá nově vyhodnocená porucha rozblíká výstup pro optickou signalizaci *SIGO* v intervalu 1sec. Je-li po odkvitování (na vstupu *ACKI* log.1) proměnná *SUMO* v log.1, je optická signalizace *SIGO* v log.1. V opačném případě je v log.0.



Obr. 2.43 Struktura FB SigErr12



Obr. 2.44 Vzhled FB SigErr12



Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní signál poruchy X	vstupní	bool
AckI	kvitace poruch		bool
ResI	nulování poruch		bool
SigI	signálka		bool
SumI	sdířená porucha		bool
Cfg	konfigurační struktura bloku		_TSigErr1_IN_
.PresetTimeX	zpoždění vyhodnocení poruchy X		time
ErrX	výstupní signál poruchy X	výstupní	bool
AckO	kvitace poruch		bool
ResO	nulování poruch		bool
SigO	signálka		bool
SumO	sdířená porucha		bool
ErrC	číslo poruchy		usint

#### 2.5.4. SigErr13 - binární poruchová signalizace s volbou nulování

Funkční blok provádí vyhodnocení výskytu poruchy s nastaveným časovým zpožděním *PRESETTIME<sub>x</sub>* pro 8 binárních vstupů. Pokud je vstupní signál *IN<sub>x</sub>* aktivní déle než je nastavená předvoba, je výstupní signál *ERR<sub>x</sub>* nastaven do log. 1. Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné *SUMO*, signalizaci nově vyhodnocené poruchy optickou signalizací *SIGO* a akustickou signalizací *AKUO*.

Výskyt poruch je možné potvrdit (kvitovat) signálem *ACKI* a zrušit signálem *RESI*. Proměnné *CASI*, *CASO* slouží ke kaskádování více komponent. Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy *ERRC*, která je určena pro připojení na funkční blok historie poruch (*History1*, *History5*, *History10*). Kód *ERRC* je oproti ostatním komponentám *SigErr* nastaven pouze na jeden cyklus při výskytu nové poruchy.

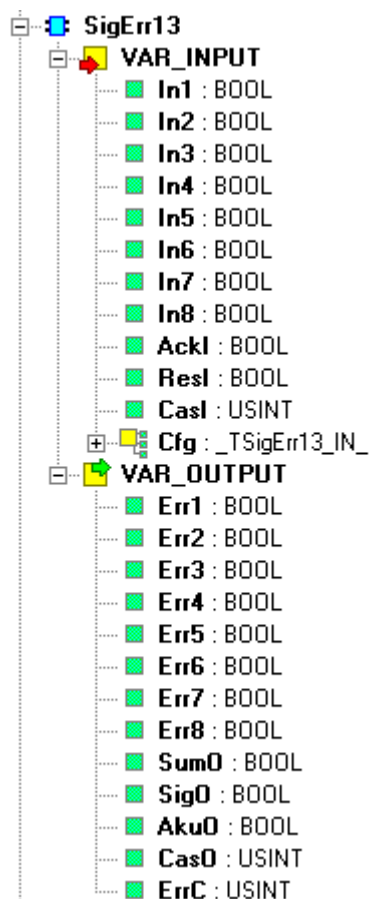
U každé poruchy lze v proměnné *CONTROL<sub>x</sub>* nastavit způsob nulování příslušné poruchy a proměnná *CONTROLS* určuje způsob optické a akustické signalizace.

- CONTROL<sub>x</sub>* = 0 - není-li vstupní signál *IN<sub>x</sub>* aktivní je výstupní proměnná *ERR<sub>x</sub>* snulována (samovratná porucha)
- CONTROL<sub>x</sub>* = 1 - výstupní proměnná *ERR<sub>x</sub>* je snulována signálem *RESI* bez ohledu na hodnotu vstupního signálu
- CONTROL<sub>x</sub>* = 2 - není-li vstupní signál *IN<sub>x</sub>* aktivní je výstupní proměnná *ERR<sub>x</sub>* snulována signálem *RESI*

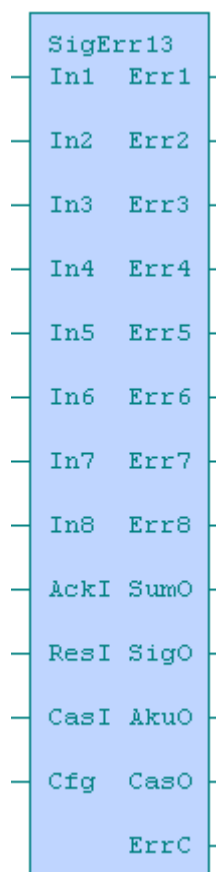
Každá nově vyhodnocená porucha rozbliká výstup pro optickou signalizaci *SIGO* v intervalu 1 sec. a nastaví do log.1 výstup pro akustickou signalizaci *AKUO*. Odkvitováním (na vstup *ACKI* log.1) je akustická signalizace snulována. Pokud je proměnná *SUMO* v log.1, je i optická signalizace *SIGO* po odkvitování v log.1, v opačném případě je v log.0.

Proměnná *CONTROLS* určuje způsob signalizace, jsou-li vyhodnocené poruchy snulovány bez předchozího odkvitování (pro samovratné poruchy).

- CONTROLS* = 0 - optická i akustická signalizace je v log.0
- CONTROLS* = 1 - akustická signalizace je v log.0, optická signalizace bliká v intervalu 1sec.
- CONTROLS* = 2 - akustická signalizace je v log.1, optická signalizace bliká v intervalu 1sec.



Obr. 2.45 Struktura FB SigErr13



Obr. 2.46 Vzhled FB SigErr13

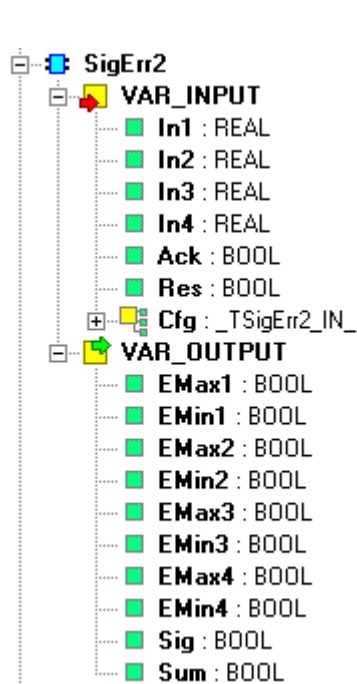
Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní signál poruchy X	vstupní	bool
AckI	kvitace poruch		bool
ResI	nulování poruch		bool
CasI	kaskádní vstup		usint
Cfg	konfigurační struktura bloku		_TSigErr1_IN_
.PresetTimeX	zpoždění vyhodnocení poruchy X		time
.controlX	řídící slovo pro způsob nulování poruchy X		usint
.controls	řídící slovo pro způsob signalizace neodkvitovaných poruch		usint
ErrX	výstupní signál poruchy X	výstupní	bool
SumO	sdužená porucha		bool
SigO	signálka		bool
AkuO	houkačka		bool
CasO	kaskádní výstup		usint
ErrC	číslo poruchy		usint

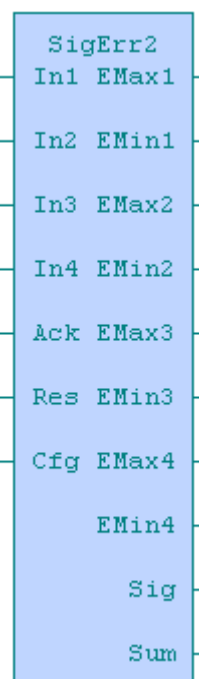
## 2.5.5. SigErr2 - analogová poruchová signalizace

Funkční blok provádí kontrolu čtyř vstupních analogových hodnot. Pokud je měřená hodnota  $IN_x$  vyšší než nastavené maximum  $PRESETMAX_x$ , po dobu delší než je nastavená předvolba  $PRESETTIME_x$ , je výstupní signál  $EMAX_x$  nastaven do log.1. V případě, že je měřená hodnota  $IN_x$  nižší než nastavené minimum  $PRESETMIN_x$ , po dobu delší než nastavená předvolba  $PRESETTIME_x$ , je do log. 1 nastaven signál  $EMIN_x$ .

Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné  $SUM$  a signalizaci nově vyhodnocené poruchy  $SIG$ . Výskyt poruch je možné potvrdit (kvitovat) signálem  $ACK$  a neaktivní poruchy zrušit signálem  $RES$ . Každá nově vyhodnocená porucha rozblíká výstup pro optickou signalizaci  $SIG$  v intervalu 1 sec. Je-li po odkvitování (na vstupu  $ACK$  log.1) proměnná  $SUM$  v log.1, je optická signalizace  $SIG$  v log.1. V opačném případě je v log.0.



Obr. 2.47 Struktura FB SigErr2



Obr. 2.48 Vzhled FB SigErr2

Popis proměnných :

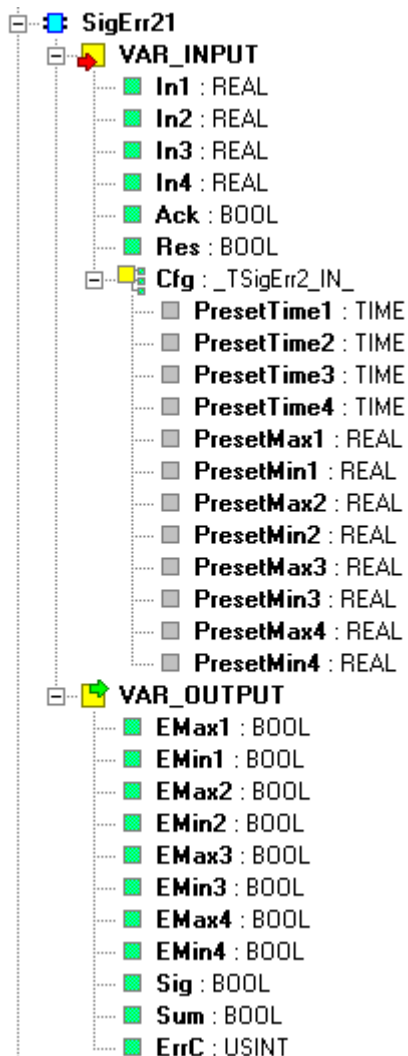
Název	Význam	Typ	Formát
$IN_x$	vstupní analogová hodnota X	vstupní	real
Ack	kvitace poruch		bool
Res	nulování poruch		bool
Cfg	konfigurační struktura bloku		<code>_TSigErr2_IN_</code>
<code>.PresetTimeX</code>	zpoždění vyhodnocení poruchy X		time
<code>.PresetMaxX</code>	mez pro vyhodnocení poruchy maxima X		real
<code>.PresetMinX</code>	mez pro vyhodnocení poruchy minima X		real
$EMAX_x$	překročení maxima na vstupu X	výstupní	bool
$EMIN_x$	podtečení minima na vstupu X		bool
Sig	signálka		bool
Sum	sdužená porucha		bool

### 2.5.6. SigErr21 - analogová poruchová signalizace se signalizací čísla poruchy

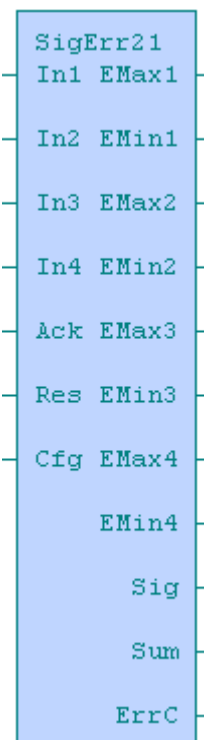
Funkční blok provádí kontrolu čtyř vstupních analogových hodnot. Pokud je měřená hodnota  $INx$  vyšší než nastavené maximum  $PRESETMAXx$ , po dobu delší než je nastavená předvolba  $PRESETTIMEx$ , je výstupní signál  $EMAXx$  nastaven do log.1. V případě, že je měřená hodnota  $INx$  nižší než nastavené minimum  $PRESETMINx$ , po dobu delší než nastavená předvolba  $PRESETTIMEx$ , je do log. 1 nastaven signál  $EMINx$ .

Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné  $SUM$  a signalizaci nově vyhodnocené poruchy optickou signalizací  $SIG$ . Výskyt poruch je možné potvrdit (kvitovat) signálem  $ACK$  a neaktivní poruchy zrušit signálem  $RES$ . Každá nově vyhodnocená porucha rozblíká výstup pro optickou signalizaci  $SIG$  v intervalu 1 sec. Je-li po odkvitování (na vstupu  $ACK$  log.1) proměnná  $SUM$  v log.1, je optická signalizace  $SIG$  v log.1. V opačném případě je v log.0.

Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy  $ERRC$ , která je určena pro připojení na funkční blok historie poruch (History1,History5,History10).



Obr. 2.49 Struktura FB SigErr21



Obr. 2.50 Vzhled FB SigErr21

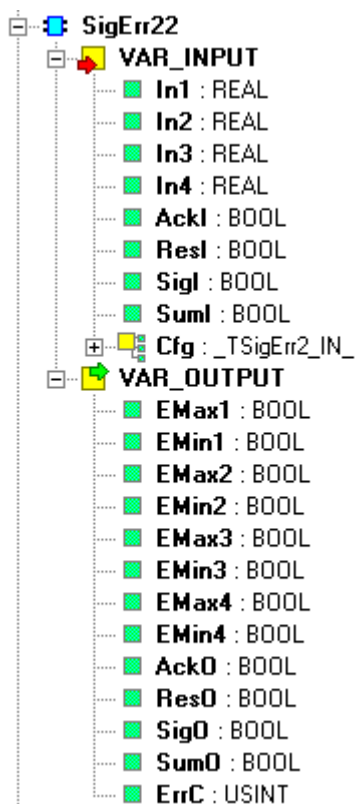
Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní analogová hodnota X	vstupní	real
Ack	kvitace poruch		bool
Res	nulování poruch		bool
Cfg	konfigurační struktura bloku		_TSigErr2_IN_
.PresetTimeX	zpoždění vyhodnocení poruchy X		time
.PresetMaxX	mez pro vyhodnocení poruchy maxima X		real
.PresetMinX	mez pro vyhodnocení poruchy minima X		real
EMaxX	překročení maxima na vstupu X	výstupní	bool
EMinX	podtečení minima na vstupu X		bool
Sig	signálka		bool
Sum	sdružená porucha		bool
ErrC	číslo poruchy		usint

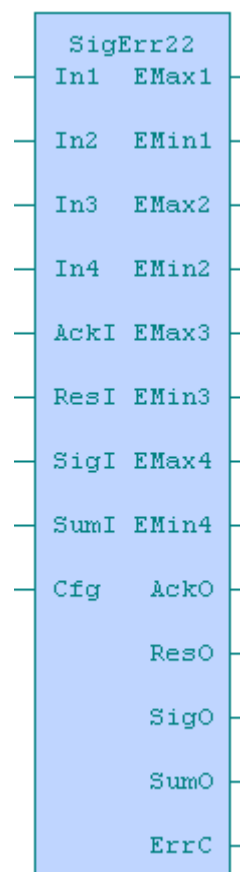
### 2.5.7. SigErr22 - analogová poruchová signalizace s obsluhou propojení

Funkční blok provádí kontrolu čtyř vstupních analogových hodnot. Pokud je měřená hodnota *INx* vyšší než nastavené maximum *PRESETMAXx*, po dobu delší než je nastavená předvolba *PRESETTIMEx*, je výstupní signál *EMAXx* nastaven do log.1. V případě, že je měřená hodnota *INx* nižší než nastavené minimum *PRESETMINx*, po dobu delší než nastavená předvolba *PRESETTIMEx*, je do log. 1 nastaven signál *EMINx*.

Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné *SUMO* a signalizaci nově vyhodnocené poruchy optickou signalizací *SIGO*. Výskyt poruch je možné potvrdit (kvitovat) signálem *ACKI* a zrušit signálem *RESI*. Proměnné *ACKO*, *RESO*, *SIGO*, *SUMO* slouží ke kaskádování více komponent. Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy *ERRC*, která je určena pro připojení na funkční blok historie poruch (*History1*,*History5*,*History10*). Každá nově vyhodnocená porucha rozbliká výstup pro optickou signalizaci *SIGO* v intervalu 1 sec. Je-li po odkvitování (na vstupu *ACKI* log.1) proměnná *SUMO* v log.1, je optická signalizace *SIGO* v log.1. V opačném případě je v log.0.



Obr. 2.51 Struktura FB SigErr22



Obr. 2.52 Vzhled FB SigErr22

Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní analogová hodnota X	vstupní	real
AckI	kvitace poruch		bool
ResI	nulování poruch		bool
SigI	signálka		bool
SumI	sdužená porucha		bool
Cfg	konfigurační struktura bloku		_TSigErr2_IN_
.PresetTimeX	zpoždění vyhodnocení poruchy X		time
.PresetMaxX	mez pro vyhodnocení poruchy maxima X		real
.PresetMinX	mez pro vyhodnocení poruchy minima X		real
EMaxX	překročení maxima na vstupu X	výstupní	bool
EMinX	podtečení minima na vstupu X		bool
AckO	kvitace poruch		bool
ResO	nulování poruch		bool
SigO	signálka		bool
SumO	sdužená porucha		bool
ErrC	číslo poruchy		usint

### 2.5.8. SigErr23 - analogová poruchová signalizace s volbou nulování

Funkční blok provádí kontrolu čtyř vstupních analogových hodnot. Pokud je měřená hodnota  $INx$  vyšší než nastavené maximum  $PRESETMAXx$ , po dobu delší než je nastavená předvolba  $PRESETTIMEx$ , je výstupní signál  $EMAXx$  nastaven do log.1. V případě, že je měřená hodnota  $INx$  nižší než nastavené minimum  $PRESETMINx$ , po dobu delší než nastavená předvolba  $PRESETTIMEx$ , je do log. 1 nastaven signál  $EMINx$ .

Dále funkční blok provádí logický součet všech vyhodnocených poruch do proměnné  $SUMO$ , signalizaci nově vyhodnocené poruchy optickou signalizací  $SIGO$  a akustickou signalizací  $AKUO$ . Výskyt poruch je možné potvrdit (kvitovat) signálem  $ACKI$  a zrušit signálem  $RESI$ . Proměnné  $CASI$ ,  $CASO$  slouží ke kaskádování více komponent. Funkční blok dále obsahuje výstupní proměnnou s číslem poslední aktivní poruchy  $ERRC$ , která je určena pro připojení na funkční blok historie poruch (History1,History5,History10). Kód  $ERRC$  je oproti ostatním komponentám SigErr nastaven pouze na jeden cyklus při výskytu nové poruchy.

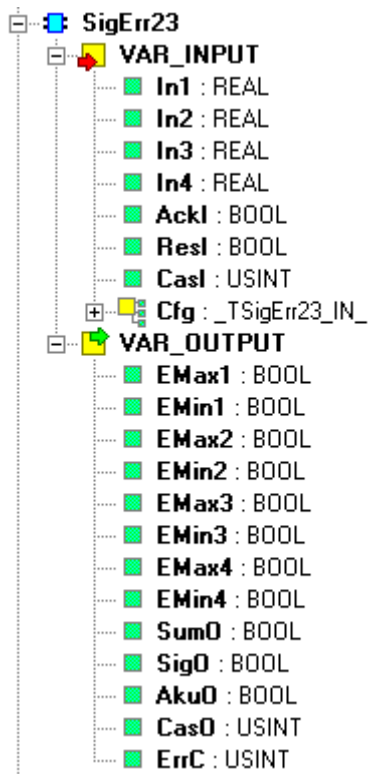
U každého vstupu lze v proměnné  $CONTROLx$  nastavit způsob nulování příslušné poruchy a proměnná  $CONTROLS$  určuje způsob optické a akustické signalizace.

- $CONTROLx = 0$  - není-li vstupní signál  $INx$  aktivní je výstupní proměnná  $ERRORx$  snulována (samovratná porucha)
- $CONTROLx = 1$  - výstupní proměnná  $ERRORx$  je snulována signálem  $RESI$  bez ohledu na hodnotu vstupního signálu
- $CONTROLx = 2$  - není-li vstupní signál  $INx$  aktivní je výstupní proměnná  $ERRORx$  snulována signálem  $RESIN$

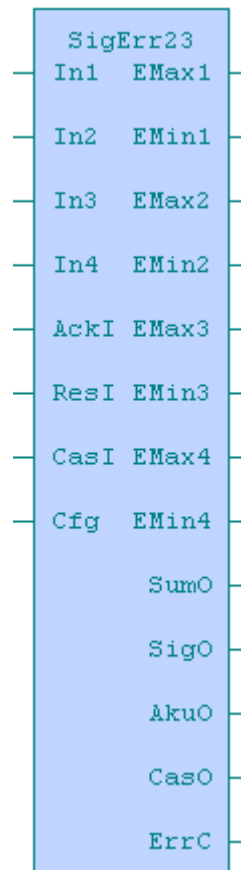
Každá nově vyhodnocená porucha rozbliká výstup pro optickou signalizaci  $SIGO$  v intervalu 1 sec. a nastaví do log.1 výstup pro akustickou signalizaci  $AKUO$ . Odkvitováním (na vstup  $ACKI$  log.1) je akustická signalizace snulována. Pokud je proměnná  $SUMO$  v log.1, je i optická signalizace  $SIGO$  po odkvitování v log.1, v opačném případě je v log.0.

Proměnná  $CONTROLS$  určuje způsob signalizace, jsou-li vyhodnocené poruchy snulovány bez předchozího odkvitování (pro samovratne poruchy).

- $CONTROLS = 0$  - optická i akustická signalizace je v log.0
- $CONTROLS = 1$  - akustická signalizace je v log.0, optická signalizace bliká v intervalu 1sec.
- $CONTROLS = 2$  - akustická signalizace je v log.1, optická signalizace bliká v intervalu 1sec.



Obr. 2.53 Struktura FB SigErr23



Obr. 2.54 Vzhled FB SigErr23

Popis proměnných :

Název	Význam	Typ	Formát
InX	vstupní analogová hodnota X	vstupní	real
AckI	kvitace poruch		bool
ResI	nulování poruch		bool
CasI	kaskádní vstup		usint
Cfg	konfigurační struktura bloku		<u>_TSigErr23_IN_</u>
.PresetTimeX	zpoždění vyhodnocení poruchy X		time
.PresetMaxX	mez pro vyhodnocení poruchy maxima X		real
.PresetMinX	mez pro vyhodnocení poruchy minima X		real
.ControlX	řídící slovo pro způsob nulování poruchy X		usint
.Controls	řídící slovo pro způsob signalizace neodkvitovaných poruch		usint
EMaxX	překročení maxima na vstupu X	výstupní	bool
EMinX	podtečení minima na vstupu X		bool
SumO	sdužená porucha		bool
SigO	signálka		bool
AkuO	houkačka		bool
CasO	kaskádní výstup		usint
ErrC	číslo poruchy		usint



## 2.6. HISTORIE PORUCH

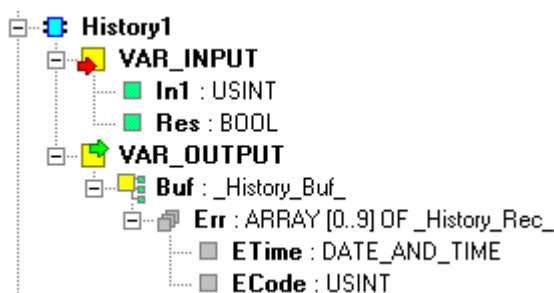
Funkční bloky historie poruch pracují podle níže uvedeného popisu. Jednotlivé funkční bloky historie poruch se liší pouze počtem poruchových signalizací, které lze k bloku historie připojit (1, 5 nebo 10).

Funkční blok provádí ukládání výskytu poruch do zásobníku typu FIFO. Na vstup *INx* funkčního bloku historie je přiveden výstup *ERRC* z poruchových signalizací (SigErr11, SigErr12, SigErr13 nebo SigErr21, SigErr22, SigErr23). Do vnitřní databáze 10-ti poruch *BUF* je uložena vždy nová příchozí porucha s datem a časem vzniku. Poslední - desátá porucha je s posunem zásobníku ztracena. Časově je tedy porucha s č.[0] nejnovější a porucha s č.[9] nejstarší. Signálem *RES* je možné databázi uložených poruch vymazat.

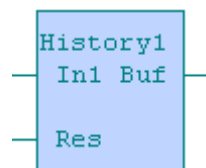
U poruch z komponent přivedených na druhý vstup a výše, je k číslu poruchy přičten offset 8 a dále násobky osmi (platí pouze pro History5 a History10).

(Př.: číslo páté poruchy z komponenty SigPor12 přivedené na vstup IN3 komponenty History5 bude 21).

### 2.6.1. History1 - historie poruch pro jednu poruchovou signalizací



Obr. 2.55 Struktura FB History1

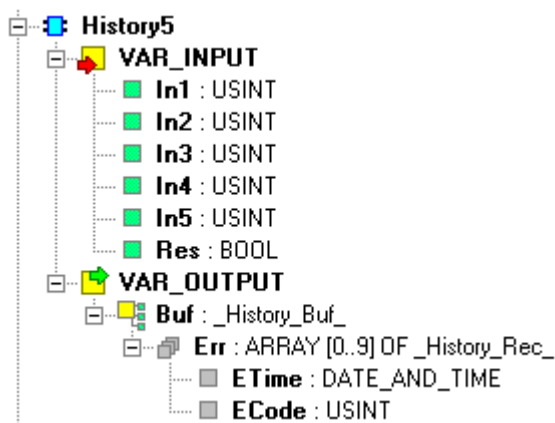


Obr. 2.56 Vzhled FB History1

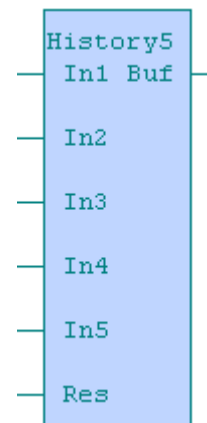
Popis proměnných :

Název	Význam	Typ	Formát
In1	číslo aktuální poruchy z bloku poruchové signalizace	vstupní	usint
Res	nulování zásobníku uložených poruch		bool
Buf	výstupní struktura	výstupní	_History_Buf_
.Err[X]	chybový buffer		array [0..9] of _History_Rec_
.ETime	datum a čas vzniku poruchy		date_and_time
.ECode	kód poruchy		usint

2.6.2. History5 - historie poruch pro pět poruchových signalizací



Obr. 2.57 Struktura FB History5

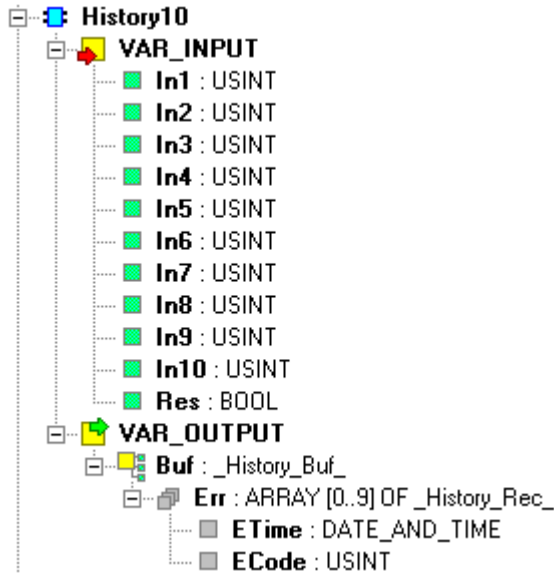


Obr. 2.58 Vzhled FB History5

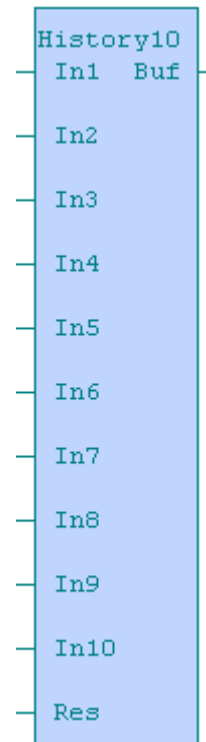
Popis proměnných :

Název	Význam	Typ	Formát
In1	číslo aktuální poruchy z poruchové signalizace 1	vstupní	usint
In2	číslo aktuální poruchy z poruchové signalizace 2		usint
In3	číslo aktuální poruchy z poruchové signalizace 3		usint
In4	číslo aktuální poruchy z poruchové signalizace 4		usint
In5	číslo aktuální poruchy z poruchové signalizace 5		usint
Res	nulování zásobníku uložených poruch		bool
Buf	výstupní struktura	výstupní	_History_Buf_
.Err[X]	chybový buffer		array [0..9] of _History_Rec_
.ETime	datum a čas vzniku poruchy		date_and_time
.ECode	kód poruchy		usint

2.6.3. History10 - historie poruch pro deset poruchových signalizací



Obr. 2.59 Struktura FB History10



Obr. 2.60 Vzhled FB History10

Popis proměnných :

Název	Význam	Typ	Formát
In1	číslo aktuální poruchy z poruchové signalizace 1	vstupní	usint
In2	číslo aktuální poruchy z poruchové signalizace 2		usint
In3	číslo aktuální poruchy z poruchové signalizace 3		usint
In4	číslo aktuální poruchy z poruchové signalizace 4		usint
In5	číslo aktuální poruchy z poruchové signalizace 5		usint
In6	číslo aktuální poruchy z poruchové signalizace 6		usint
In7	číslo aktuální poruchy z poruchové signalizace 7		usint
In8	číslo aktuální poruchy z poruchové signalizace 8		usint
In9	číslo aktuální poruchy z poruchové signalizace 9		usint
In10	číslo aktuální poruchy z poruchové signalizace 10		usint
Res	nulování zásobníku uložených poruch		bool
Buf	výstupní struktura	výstupní	_History_Buf_
.Err[X]	chybový buffer		array [0..9] of _History_Rec_
.ETime	datum a čas vzniku poruchy		date_and_time
.ECode	kód poruchy		usint

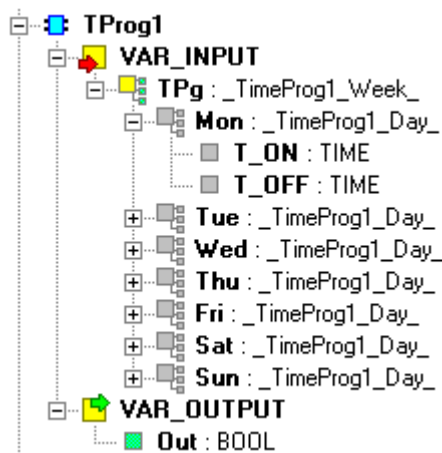
## 2.7. ČASOVÉ PROGRAMY

### 2.7.1. TProg1 - týdenní časový program s jedním úsekem ZAP/VYP za den

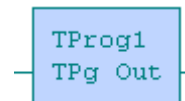
Funkční blok na základě nastaveného týdenního programu a systémového času PLC nastavuje výstupní signál provozu *OUT*.

Proměnná *OUT* je v log.1 pokud je aktuální systémový čas mezi parametry *T\_ON* a *T\_OFF* pro daný den v týdnu, jinak je v log.0. Proměnné lze nastavit v rozsahu od 00:00:00 do 24:00:00, přičemž hodnota 00:00:00 je uvažována jako počátek a hodnota 24:00:00 jako konec daného dne.

Pro každý den v týdnu lze nastavit jeden časový úsek provozu.



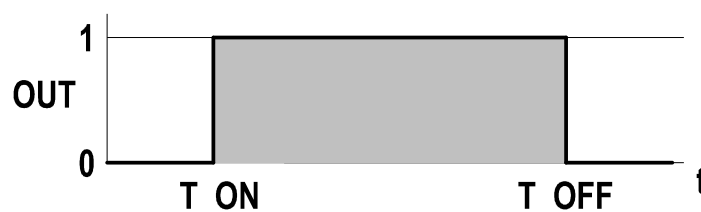
Obr. 2.61 Struktura FB TProg1



Obr. 2.62 Vzhled FB TProg1

Popis proměnných :

Název	Význam	Typ	Formát
TPg	týdenní časový program	vstupní	_TimeProg1_Week_
.Mon	časový program pro pondělí		_TimeProg1_Day_
.T_ON	začátek provozu		time
.T_OFF	konec provozu		time
.	.		.
.	.		.
.	.		.
.Sun	časový program pro neděli		_TimeProg1_Day_
.T_ON	začátek provozu		time
.T_OFF	konec provozu		time
Out	provoz	výstupní	bool



Obr. 2.63 Graf průběhů FB TProg1

**Příklady:**

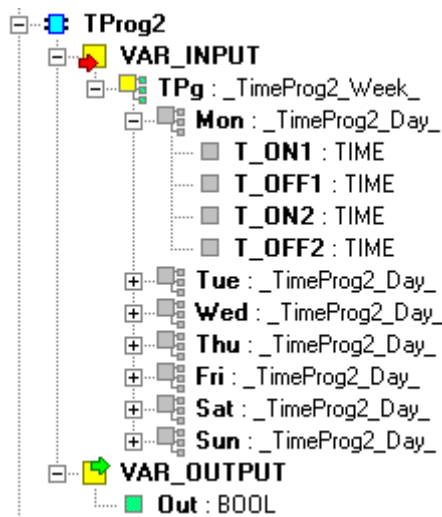
Mon.T\_ON := T#00h00m, Mon.T\_OFF := T#24h00m ... v pondělí je nepřetržitý provoz  
 Mon.T\_ON := T#06h30m, Mon.T\_OFF := T#20h15m ... v pondělí je provoz od 06:30 do 20:15  
 Mon.T\_ON := T#00h00m, Mon.T\_OFF := T#00h00m ... v pondělí je provoz vypnut

**2.7.2. TProg2 - týdenní časový program se dvěma úseky ZAP/VYP za den**

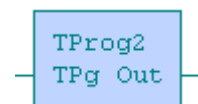
Funkční blok na základě nastaveného týdenního programu a systémového času PLC nastavuje výstupní signál provozu *OUT*.

Proměnná *OUT* je v log.1 pokud je aktuální systémový čas mezi parametry *T\_ON1* a *T\_OFF1*, nebo *T\_ON2* a *T\_OFF2* pro daný den v týdnu, jinak je v log.0. Proměnné lze nastavit v rozsahu od 00:00:00 do 24:00:00, přičemž hodnota 00:00:00 je uvažována jako počátek a hodnota 24:00:00 jako konec daného dne.

Pro každý den v týdnu lze nastavit dva časové úseky provozu.



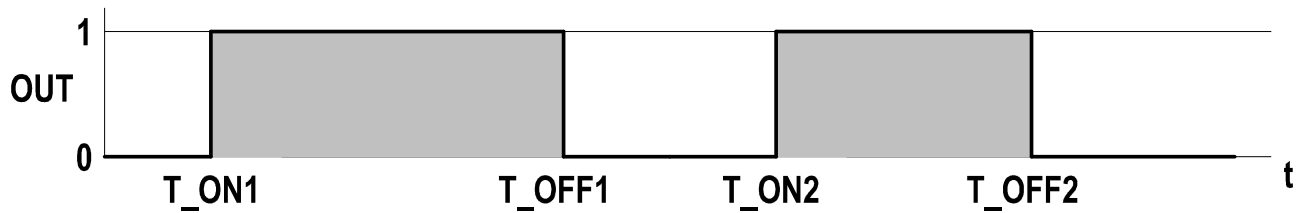
Obr. 2.64 Struktura FB TProg2



Obr. 2.65 Vzhled FB TProg2

Popis proměnných :

Název	Význam	Typ	Formát
TPg	týdenní časový program	vstupní	_TimeProg2_Week_
.Mon	časový program pro pondělí		_TimeProg2_Day_
.T_ON1	začátek provozu 1		time
.T_OFF1	konec provozu 1		time
.T_ON2	začátek provozu 2		time
.T_OFF2	konec provozu 2		time
.	.		.
.	.		.
.	.		.
.Sun	časový program pro neděli		_TimeProg2_Day_
.T_ON1	začátek provozu 1		time
.T_OFF1	konec provozu 1		time
.T_ON2	začátek provozu 2		time
.T_OFF2	konec provozu 2		time
Out	provoz	výstupní	bool



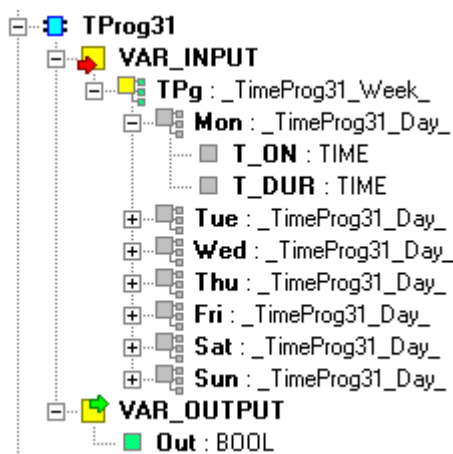
Obr. 2.66 Graf průběhů FB TProg2

### 2.7.3. TProg31 - týdenní časový program s jedním úsekem doby provozu

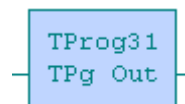
Funkční blok na základě nastaveného týdenního programu a systémového času PLC nastavuje výstupní signál *OUT*.

Proměnná *OUT* je v log.1 pokud je aktuální systémový čas větší nebo roven parametru *T\_ON* a menší než parametr *T\_ON* + parametr *T\_DUR*, pro daný den v týdnu. Jinak je výstup *OUT* v log.0. Proměnné *T\_ON* a *T\_DUR* lze nastavit v rozsahu od 00:00:00 do 24:00:00, přičemž hodnota 00:00:00 je uvažována jako počátek a hodnota 24:00:00 jako konec daného dne. Pro daný den je možno v součtu hodnot *T\_ON* a *T\_DUR* dosáhnout maximálně hodnoty 24:00:00.

Pro každý den v týdnu lze nastavit jeden časový úsek provozu.



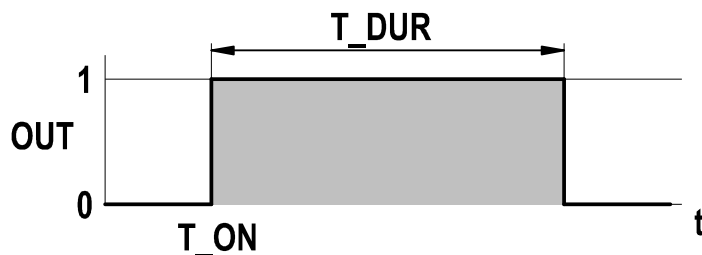
Obr. 2.67 Struktura FB TProg31



Obr. 2.68 Vzhled FB TProg31

Popis proměnných :

Název	Význam	Typ	Formát
TPg	týdenní časový program	vstupní	_TimeProg31_Week_
.Mon	časový program pro pondělí		_TimeProg31_Day_
.T_ON	začátek provozu		time
.T_DUR	doba provozu		time
.	.		.
.	.		.
.	.		.
.Sun	časový program pro neděli		_TimeProg31_Day_
.T_ON	začátek provozu		time
.T_DUR	doba provozu		time
Out	provoz	výstupní	bool



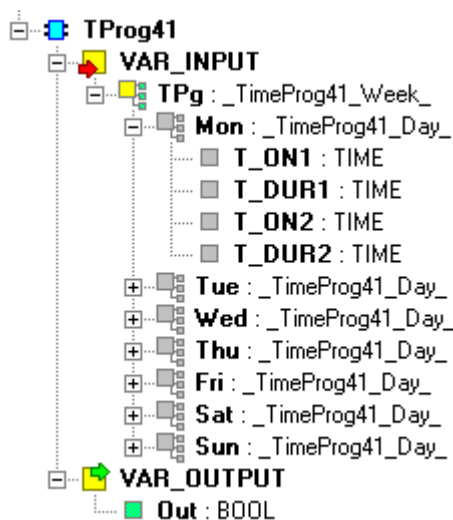
Obr. 2.69 Graf průběhů FB Tprog31

#### 2.7.4. TProg41 - týdenní časový program se dvěma úseky doby provozu

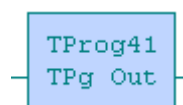
Funkční blok na základě nastaveného týdenního programu a systémového času PLC nastavuje výstupní signál *OUT*.

Proměnná *OUT* je v log.1 pokud je aktuální systémový čas větší nebo roven parametru *T\_ON* a menší než parametr *T\_ON* + parametr *T\_DUR*, pro daný den v týdnu. Jinak je výstup *OUT* v log.0. Proměnné *T\_ON* a *T\_DUR* lze nastavit v rozsahu od 00:00:00 do 24:00:00, přičemž hodnota 00:00:00 je uvažována jako počátek a hodnota 24:00:00 jako konec daného dne. Pro daný den je možno v součtu hodnot *T\_ON* a *T\_DUR* dosáhnout maximálně hodnoty 24:00:00.

Pro každý den v týdnu lze nastavit dva časové úseky provozu.



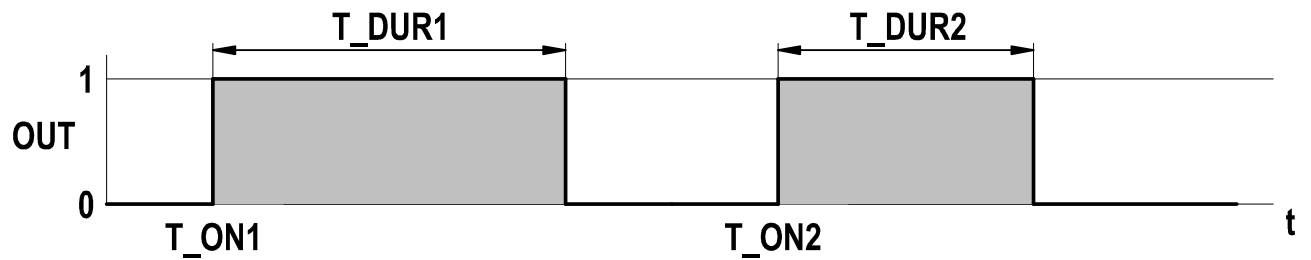
Obr. 2.70 Struktura FB TProg41



Obr. 2.71 Vzhled FB TProg41

Popis proměnných :

Název	Význam	Typ	Formát
TPg	týdenní časový program	vstupní	_TimeProg41_Week_
.Mon	časový program pro pondělí		_TimeProg41_Day_
.T_ON1	začátek provozu 1		time
.T_DUR1	doba provozu 1		time
.T_ON2	začátek provozu 2		time
.T_DUR2	doba provozu 2		time
.	.		.
.	.		.
.	.		.
.Sun	časový program pro neděli		_TimeProg41_Day_
.T_ON1	začátek provozu 1		time
.T_DUR1	doba provozu 1		time
.T_ON2	začátek provozu 2		time
.T_DUR2	doba provozu 2		time
Out	provoz	výstupní	bool



Obr. 2.72 Graf průběhů FB Tprog41



## 3. KNIHOVNA IRCLIB

Knihovna IRCLib.mlb obsahuje pouze funkční blok IRC (Intelligent Room Control).

### 3.1. IRC – PATROVÝ MODUL

Funkční blok slouží pro obsluhu až 16 pokojových modulů řady Tecoreg IRC (Tecoreg TR100) připojených na sériový kanál PLC. Komunikace s pokojovými moduly probíhá komunikačním protokolem Tnet, který je realizován ve funkčním bloku. Vlastní data získaná z pokojových IRC modulů jsou uložena v zápisníku PLC v globální veřejné datové struktuře `_IRC_PS` (tato struktura se automaticky vytvoří přidáním knihovny IRCLib.mlb do projektu).

Pro tuto strukturu je ve firemním vizualizačním nástroji Reliance implementována přímá podpora (IRC komponenta) a není tudíž nutná konkrétní znalost celé této IRC struktury.

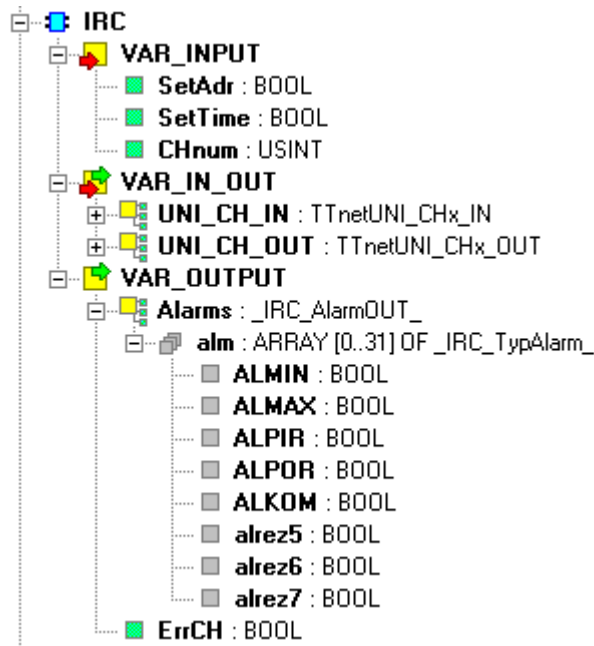
Současně s datovou strukturou `_IRC_PS` je založena i řídicí struktura `_IRC_Flags`, která obsahuje příznaky pro provedení adresace a nastavení času do pokojových modulů IRC. Funkce adresace a nastavení času jsou též dostupné přímo pomocí vstupů funkčního bloku `SetAdr` a `SetTime`.

Přiřazení funkčního bloku k sériovému kanálu je provedeno pomocí vstupní proměnné `Chnum`, která obsahuje číslo sériového kanálu nakonfigurovaného pro protokol Tnet. Vlastní komunikační zóny tohoto sériového kanálu jsou do bloku přiřazeny pomocí vstup/výstupních proměnných `UNI_CH_IN` a `UNI_CH_OUT`. Tyto proměnné je nutné propojit na systémové proměnné `UNI_CHx_IN` a `UNI_CHx_OUT` (kde `x` představuje číslo vybraného sériového kanálu).

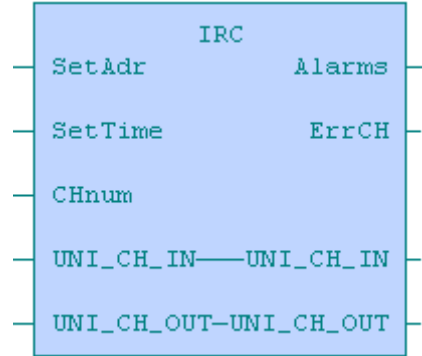
Pro použití funkčního bloku IRC je nutné mít vyhrazen jeden sériový komunikační kanál PLC. Tento kanál musí být osazen rozhraním RS-485 a nastaven do režimu „uni“. Komunikační parametry tohoto kanálu musí být nastaveny v souladu s obr.2.35. Při jiném nastavení parametrů nebude komunikace protokolem Tnet funkční !!! Současně bude nastaven výstupní signál bloku `ErrCH`. Na výstupu bloku jsou též přímo přístupná alarmová hlášení jednotlivých pokojových IRC modulů (alarmy jsou obsaženy i ve veřejné datové struktuře `_IRC_PS`).

#### Upozornění

Funkční blok IRC může být v každém projektu použit maximálně 1x !!! Vícenásobné použití tohoto bloku v jednom projektu vede ke kolizi dat ve veřejné datové struktuře `_IRC_PS`.



Obr. 3.73 Struktura FB IRC



Obr. 3.74 Vzhled FB IRC

Popis proměnných :

Název	Význam	Typ	Formát
SetAdr	adresace sítě Tnet	vstupní	bool
SetTime	nastavení času v síti Tnet		bool
CHnum	číslo sériového kanálu (nakonfigurovaného pro Tnet)		usint
UNI_CH_IN UNI_CH_OUT	vstupní komunikační zóna Tnet výstupní komunikační zóna Tnet	vstupně/ výstupní	TTnetUNI_CHx_IN TTnetUNI_CHx_OUT
Alarms	výstupní struktura	výstupní	_IRC_AlarmOUT_
.alm[X]	alarmová hlášení pokojových modulů		array [0..31] of _IRC_TypeAlarm_
.ALMIN	minimální teplota prostoru, (<+8 °C)		bool
.ALMAX	maximální teplota prostoru, (>+39,5 °C)		bool
.ALPIR	vniknutí do místnosti (je-li aktivováno)		bool
.ALPOR	porucha pokojového modulu		bool
.ALKOM	porucha komunikace s pokojovým modulem		bool
ErrCH	porucha parametrů sériového kanálu		bool

**Nastavení univerzálního režimu kanálu**

Příjemci zóna Délka zóny: 48 Adresa zóny: <input type="checkbox"/> 4 Příjemci zóna: CH2_ZoneIN	Vysílací zóna Délka zóny: 48 Adresa zóny: <input type="checkbox"/> 4 Vysílací zóna: CH2_ZoneOUT	Komunikační rychlost: 19 200 Formát dat: 8b   sudá parita
Počáteční znak <input type="checkbox"/> Detekovat <input type="checkbox"/> Vysílat Kód znaku: 0	Koncový znak <input type="checkbox"/> Detekovat <input type="checkbox"/> Vysílat <input type="checkbox"/> Dva znaky Kód znaku: 0   0	Adresa stanice Adresa stanice: 0 <input type="checkbox"/> Detekovat při příjmu <input type="checkbox"/> Zápis při vysílání
Parita prvního bytu přijímané zprávy <input checked="" type="radio"/> Stejná parita jako u ostatních <input type="radio"/> Opačná parita než u ostatních	Parita prvního bytu vysílané zprávy <input checked="" type="radio"/> Stejná parita jako u ostatních <input type="radio"/> Opačná parita než u ostatních	Kontrolní součet <input type="checkbox"/> Kontrola při příjmu <input type="checkbox"/> Výpočet při vysílání Poz. prvního znaku CHS: 0
Potvrzení zprávy bez dat <input type="checkbox"/> Detekovat <input type="checkbox"/> Vysílat <input type="checkbox"/> Dva znaky Kód znaku: 0   0	Délka zprávy <input type="checkbox"/> Detekovat při příjmu <input type="checkbox"/> Zápis při vysílání Pozice délky zprávy: 0 Maximální délka: 48	Režim řízení modemových signálů Řízení signálu RTS: automatická hodnota Řízení signálu DTR: trvale hodnota 0 <input type="checkbox"/> Odpojení přijímače během vysílání
Min. doba klidu na lince mezi přijímanými zprávami (počet bytů): 5 Min. doba klidu na lince mezi vysílanými zprávami (počet bytů): 5		

OK Zrušit nápověda

Obr. 3.75 Nastavení sériového kanálu v režimu „uni“ pro protokol Tnet (zde pro kanál CH2)

Struktury `_IRC_Flags` a `_IRC_PS` jsou mapovány v zápisníku PLC a jsou určeny pro předávání dat z/do nadřazeného systému (vizualizačního SW Reliance).

### Struktura `_IRC_Flags`

Struktura `_IRC_Flags` obsazuje v zápisníku PLC 1 byte a obsahuje pouze příznakový registr s těmito proměnnými typu bool :

```

STRUCT
  SetAdr1   : bool;  (* adresace site Tnet *)
  dummy1   : bool;
  SetTime1  : bool;  (* nastaveni casu site Tnet *)
  dummy3   : bool;
  dummy4   : bool;
  dummy5   : bool;
  dummy6   : bool;
  dummy7   : bool;
END_STRUCT;

```

Proměnná *SetAdr1* slouží k aktivaci adresačního režimu pokojových modulů, proměnná *SetTime1* slouží k nastavení systemového času pokojových modulů (podle systemového času PLC).

### Struktura *\_IRC\_PS*

Struktura *\_IRC\_PS* obsazuje v zápisníku PLC 7200 bytů a je rozdělena do čtyř datových zón. Jsou to provozní data, řídicí data, konfigurační data a alarmy pokojových modulů (alarmy jsou dostupné též jako výstupní proměnné funkčního bloku IRC).

```
STRUCT
  ProvTR : ARRAY [0..31] OF _IRC_ProvData_;(* provozni data pokoj. modulu *)
  ContTR : ARRAY [0..31] OF _IRC_ContData_;(* ridici data pokoj. modulu *)
  KfgTR  : ARRAY [0..31] OF _IRC_KfgData_; (* konfiguracioni data pok.modulu*)
  AlmTR  : ARRAY [0..31] OF _IRC_TypAlarm_;(* alarmy pokojovych modulu *)
END_STRUCT;
```

Celá struktura *\_IRC\_PS* je navržena pro 32 pokojových IRC modulů, funkční blok IRC však obsluhuje pouze 16 modulů, na adresách 0-15. Konkrétní význam jednotlivých položek celé struktury je popsán v dokumentaci TXV 138 04, Technické vybavení komunikačního modulu TR101, kapitola Veřejná datová struktura. Oproti TR101 není však tato struktura do zápisníku mapována pevně (v TR101 je to registr R100).

### Příklad použití v ST

Pokud je funkční blok IRC použit při programování v jazyce ST, je nutno při volání bloku potlačit typovou kontrolu vstup/výstupních proměnných *UNI\_CH\_IN* a *UNI\_CH\_OUT* pomocí direktivy *void()*. Zdrojový text volání FB IRC pak bude vypadat např. takto :

```
iIRC(CHnum:=2, UNI_CH_IN:=void(UNI_CH2_IN), UNI_CH_OUT:=void(UNI_CH2_OUT));
```

### Vazba IRC na Relianci

Ve firemním vizualizačním systému Reliance je pro IRC systém připravena přímá podpora v podobě IRC komponenty. Ta umožňuje komfortní parametrizaci i vlastní provoz IRC systému. Tato komponenta pracuje nad veřejnou datovou strukturou *\_IRC\_PS* a pro správnou činnost tedy vyžaduje informaci o umístění této datové struktury do zápisníku PLC. V prostředí Mosaic je tato informace dostupná v menu *Nástroje->Mapa uživatelských registrů*, proměnná *\_IRC\_PS*.

Poznámky:

Poznámky:



teco

Objednávky a informace:

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

TXV 003 23.01

Výrobce si vyhrazuje právo na změny dokumentace. Poslední aktuální vydání je k dispozici na internetu  
[www.tecomat.cz](http://www.tecomat.cz)