

Knihovna EnergyLib

TXV 003 65.01
čtvrté vydání
červen 2012
změny vyhrazeny

Historie změn

Datum	Vydání	Popis změn
Únor 2011	1	První vydání, popis odpovídá EnergyLib_v10
Červen 2011	2	Doplněn popis bloku fbElectricityMeterPulse(), popis odpovídá EnergyLib_v11
Únor 2012	3	Přidány bloky pro měření průtoku a tepla popis odpovídá EnergyLib_v14
Červen 2012	4	Popis k verzi EnergyLib_v16

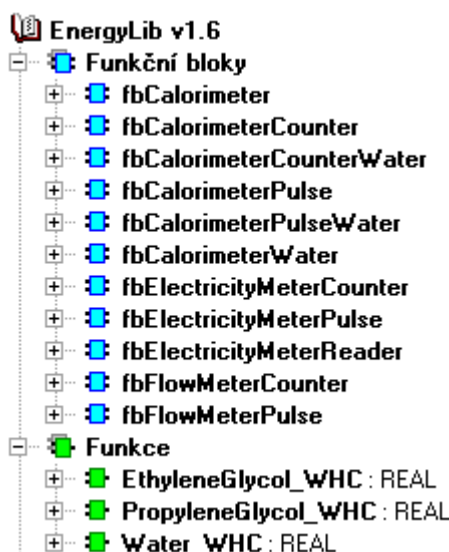
OBSAH

1 Úvod.....	3
2 Datové typy.....	3
3 Konstanty.....	4
4 Globální proměnné.....	4
5 Funkce	5
5.1 Funkce Water_WHC.....	5
5.2 Funkce EthyleneGlycol_WHC.....	6
5.3 Funkce PropyleneGlycol_WHC.....	7
6 Funkční bloky.....	8
6.1 Funkční blok fbElectricityMeterReader.....	9
6.2 Funkční blok fbElectricityMeterPulse.....	14
6.3 Funkční blok fbElectricityMeterCounter.....	16
6.4 Funkční blok fbFlowMeterPulse.....	18
6.5 Funkční blok fbFlowMeterCounter.....	20
6.6 Funkční blok fbCalorimeter.....	22
6.7 Funkční blok fbCalorimeterPulse.....	25
6.8 Funkční blok fbCalorimeterCounter.....	28
6.9 Funkční blok fbCalorimeterWater.....	31
6.10 Funkční blok fbCalorimeterPulseWater.....	33
6.11 Funkční blok fbCalorimeterCounterWater.....	36

1 ÚVOD

Knihovna EnergyLib je standardně dodávána jako součást programovacího prostředí Mosaic. Knihovna obsahuje funkční bloky umožňující načítat údaje z elektroměru připojeného přes opto hlavici (komunikace podle normy ČSN EN 62056-21) a počítat spotřebu elektrické energie na základě pulzů S0 poskytovaných elektroměrem. Dále knihovna poskytuje funkce pro výpočet průtoku a dodaného tepla na základě údaje z průtokoměru a měřených teplot.

Následující obrázek ukazuje strukturu knihovny EnergyLib v prostředí Mosaic



Pokud chceme funkce z knihovny EnergyLib použít v aplikačním programu PLC, je třeba nejprve přidat tuto knihovnu do projektu. Současně je nezbytné, aby projekt obsahoval také knihovny StdLib a ComLib, protože některé funkce z těchto knihoven jsou použity v knihovně EnergyLib. Novější verze programu Mosaic (od v2.0.25) přidávají uvedené knihovny automaticky při přidání knihovny EnergyLib (pokud tyto knihovny ještě nejsou v projektu). Knihovna EnergyLib je dodávána jako součást instalace prostředí Mosaic od verze v2.0.26.

Knihovna EnergyLib není podporovaná na systémech TC-650, u systému TC700 nelze knihovnu použít s procesorovými moduly CP-7002, CP-7003 a CP-7005.

Funkce a funkční bloky knihovny EnergyLib jsou podporovány v centrálních jednotkách řady K (TC700 CP-7000 a CP-7004, všechny varianty systému Foxtrot) od verze v4.4.

2 DATOVÉ TYPY

V knihovně EnergyLib nejsou definovány žádné datové typy.

3 KONSTANTY

V knihovně EnergyLib jsou definovány následující konstanty:

 **Globální proměnné**
 **VAR_GLOBAL CONSTANT**
 **METER_INVALID_VALUE** : REAL := -10000000000.0
 **METER_INVALID_VALUEL** : LREAL := -10000000000.0

Konstanta *METER_INVALID_VALUE* se používá jako návratová hodnota pro položky, které nejsou dostupné (např. v případě, že je elektroměr neposkytuje)

<i>Identifikátor</i>	<i>Typ</i>	<i>Hodnota</i>	<i>Význam</i>
<i>METER_INVALID_VALUE</i>	REAL	-10000000000.0	Neplatná hodnota
<i>METER_INVALID_VALUEL</i>	LREAL	-10000000000.0	Neplatná hodnota

4 GLOBÁLNÍ PROMĚNNÉ

V knihovně EnergyLib nejsou definovány žádné globální proměnné.

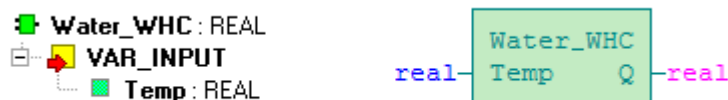
5 FUNKCE

V knihovně EnergyLib je definována následující funkce.

<i>Funkce</i>	<i>Popis</i>
<i>Water_WHC</i>	Výpočet objemové tepelné kapacity vody v závislosti na teplotě
<i>EthyleneGlycol_WHC</i>	Výpočet objemové tepelné kapacita ethylglykolové teplotnostné směsi
<i>PropyleneGlycol_WHC</i>	Výpočet objemové tepelné kapacita propylenglykolové teplotnostného směsi

5.1 Funkce Water_WHC

Knihovna : *VzorLib*



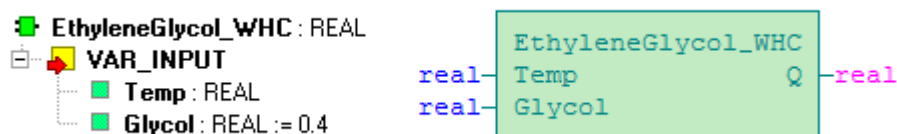
Funkce *Water_WHC* vrátí objemovou tepelnou kapacitu vody při normálním atmosférickém tlaku. Hodnoty jsou platné od 0 do 100 °C

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>temp</i>	REAL	Teplota [°C]
FlashInfo			
	<i>Návratová hodnota</i>	REAL	Objemová tepelná kapacita [kJm ⁻³ h ⁻¹]

Příklad programu s voláním funkce *Water_WHC* je součástí popisu bloku *fbCalorimeter-Counter*

5.2 Funkce EthyleneGlycol_WHC

Knihovna : *VzorLib*

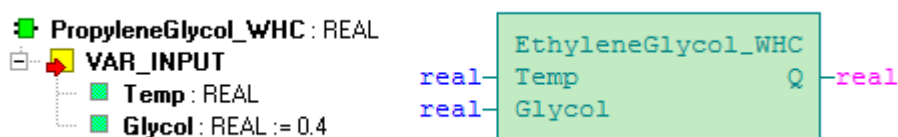
Funkce *EthyleneGlycol_WHC* vrátí objemovou tepelnou kapacitu ethylglykolové teplosné směsi při normálním atmosferickém tlaku.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>temp</i>	REAL	Teplota [°C]
	<i>Glycol</i>	REAL	Hmotnostní koncentrace glykolu v roztoku (0 - 1)
FlashInfo			
	<i>Návratová hodnota</i>	REAL	Objemová tepelná kapacita [kJm ⁻³ h ⁻¹]

Příklad programu s voláním funkce *EthyleneGlycol_WHC* je součástí popisu bloku *fbCalorimeter*

5.3 Funkce *PropyleneGlycol_WHC*

Knihovna : *VzorLib*

Funkce *PropyleneGlycol_WHC* vrácí objemovou tepelnou kapacitu ethylglykolové teplosné směsi při normálním atmosferickém tlaku.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>temp</i>	REAL	Teplota [°C]
	<i>Glycol</i>	REAL	Hmotnostní koncentrace glykolu v roztoku (0 - 1)
FlashInfo			
	Návratová hodnota	REAL	Objemová tepelná kapacita [kJm ⁻³ h ⁻¹]

Příklad programu s voláním funkce *PropyleneGlycol_WHC* je součástí popisu bloku *fb-CalorimeterPulse*

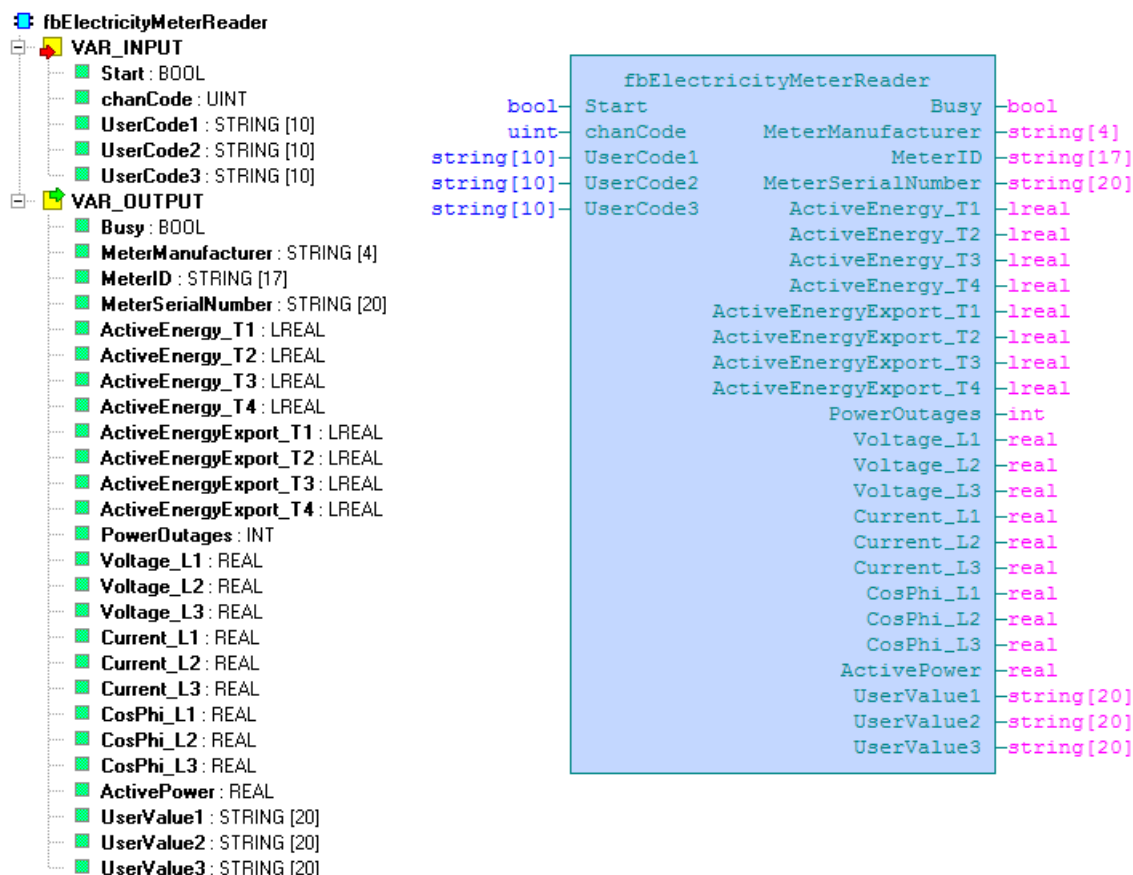
6 FUNKČNÍ BLOKY

V knihovně EnergyLib jsou definovány následující funkční bloky:

-  Funkční bloky
-  fbCalorimeter
-  fbCalorimeterCounter
-  fbCalorimeterCounterWater
-  fbCalorimeterPulse
-  fbCalorimeterPulseWater
-  fbCalorimeterWater
-  fbElectricityMeterCounter
-  fbElectricityMeterPulse
-  fbElectricityMeterReader
-  fbFlowMeterCounter
-  fbFlowMeterPulse

<i>Funkční blok</i>	<i>Popis</i>
<i>fbElectricityMeterReader</i>	Načte hodnoty z elektroměru přes optickou hlavici
<i>fbElectricityMeterPulse</i>	Na základě pulzů z elektroměru počítá okamžitou spotřebu, celkovou spotřebu a proud
<i>fbElectricityMeterCounter</i>	Na základě hodnoty čítače pulzů z elektroměru počítá okamžitou spotřebu, celkovou spotřebu a proud
<i>fbFlowMeterPulse</i>	Na základě pulzů z průtokoměru počítá okamžitý průtok a celkový proteklý objem
<i>fbFlowMeterCounter</i>	Na základě hodnoty čítače pulzů z průtokoměru počítá okamžitý průtok a celkový proteklý objem
<i>fbCalorimeter</i>	Na základě průtoku a dvou teplot počítá okamžitý výkon a celkové teplo
<i>fbCalorimeterPulse</i>	Na základě pulzů z průtokoměru a dvou teplot počítá okamžitý výkon a celkové teplo
<i>fbCalorimeterCounter</i>	Na základě hodnoty čítače pulzů z průtokoměru a dvou teplot počítá okamžitý výkon a celkové teplo
<i>fbCalorimeterWater</i>	Varianta <i>fbCalorimeter</i> pro vodu jako teplotonosné médium
<i>fbCalorimeterPulseWater</i>	Varianta <i>fbCalorimeterPulse</i> pro vodu jako teplotonosné médium
<i>fbCalorimeterCounterWater</i>	Varianta <i>fbCalorimeterCounter</i> pro vodu jako teplotonosné médium

6.1 Funkční blok *fbElectricityMeterReader*

Knihovna : *EnergyLib*

Funkční blok *fbElectricityMeterReader* slouží k načtení údajů z elektroměru, který je připojen přes optickou čtecí hlavu (např. ZPA S10 IR). Elektroměr musí podporovat komunikaci podle normy ČSN EN 62056-21.

Funkční blok *fbElectricityMeterReader* vrací následující údaje :

- identifikační údaje elektroměru (kód výrobce, kód přístroje, výrobní číslo)
- stavy počítadel odebrané energie
- stavy počítadel dodané energie
- počet výpadků napájení
- napětí, proud a $\cos \varphi$ v jednotlivých fázích
- okamžitou spotřebu

Elektroměr nemusí nutně poskytovat všechny uvedené položky (např. jednofázový elektroměr neposkytuje údaje o fázích L2 a L3, navíc dodavatel elektroměru může nastavit, které údaje bude elektroměr přes optické rozhraní poskytovat). Tyto položky pak mají na výstupu bloku hodnotu *METER_INVALID_VALUE*. Naopak elektroměr může poskytovat některé další údaje oproti uvedenému výčtu. Tyto položky lze získat zadáním OBIS kódu položky do vstupních proměnných *UserCode1* až *UserCode3*.

Čtení údajů z elektroměru je zahájeno na náběžnou hranu vstupní proměnné *Start*, která musí mít hodnotu log. 1 po celou dobu vyčítání informací z elektroměru. Vstupní proměnná *chanCode* musí obsahovat číslo sériového kanálu, přes který je připojena optická čtecí hlava (viz kon-

stanty *CH1_uni* až *CH10_uni* z knihovny ComLib). Během načítání údajů je nastavena výstupní proměnná *Busy* na hodnotu *TRUE*. To může trvat několik desítek vteřin v závislosti na počtu údajů, které elektroměr poskytuje. Po načtení všech údajů se nastaví proměnná *Busy* na hodnotu *FALSE* a do ostatních výstupních proměnných bloku se zapíše načtené údaje. Vyčtení nových údajů je nutné spustit novou náběžnou hranou vstupu *Start*.

Funkční blok *fbElectricityMeterReader* je podporován na centrálních jednotkách řady K (TC700 CP-7000, CP-7004, CP-7007 a všechny verze Foxtrot) od verze firmware v4.4.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Start</i>	BOOL	Řídicí proměnná. Náběžná hrana spouští vyčtení dat z elektroměru. Log 0 zastaví řešení funkčního bloku
	<i>chanCode</i>	UINT	Kód komunikačního kanálu, kam je připojena optická čtecí hlava <i>CH1_uni</i> sériový kanál CH1, režim uni <i>CH10_uni</i> sériový kanál CH10, režim uni
	<i>UserCode1</i>	STRING[10]	OBIS kód pole, které chce uživatel načíst
	<i>UserCode2</i>	STRING[10]	OBIS kód pole, které chce uživatel načíst
	<i>UserCode3</i>	STRING[10]	OBIS kód pole, které chce uživatel načíst
VAR_OUTPUT			
	<i>Busy</i>	BOOL	Příznak, že se právě načítají data z elektroměru
	<i>MeterManufacturer</i>	STRING[4]	Kód výrobce elektroměru
	<i>MeterID</i>	STRING[17]	Identifikační řetězec elektroměru
	<i>MeterSerialNumber</i>	STRING[20]	Výrobní číslo elektroměru
	<i>ActiveEnergy_T1</i>	LREAL	Stav počítadla odebrané energie pro tarif T1 (zpravidla vysoký tarif)
	<i>ActiveEnergy_T2</i>	LREAL	Stav počítadla odebrané energie pro tarif T2 (zpravidla nízký tarif)
	<i>ActiveEnergy_T3</i>	LREAL	Stav počítadla odebrané energie pro tarif T3
	<i>ActiveEnergy_T4</i>	LREAL	Stav počítadla odebrané energie pro tarif T4
	<i>ActiveEnergyExport_T1</i>	LREAL	Stav počítadla dodané energie pro tarif T1 (zpravidla vysoký tarif)
	<i>ActiveEnergyExport_T2</i>	LREAL	Stav počítadla dodané energie pro tarif T2

	Proměnná	Typ	Význam
			(zpravidla nízký tarif)
	<i>ActiveEnergyExport_T3</i>	LREAL	Stav počítadla dodané energie pro tarif T3
	<i>ActiveEnergyExport_T4</i>	LREAL	Stav počítadla dodané energie pro tarif T4
	<i>PowerOutages</i>	INT	Počet výpadků napájení
	<i>Voltage_L1</i>	REAL	Napětí fáze L1 ve Voltech
	<i>Voltage_L2</i>	REAL	Napětí fáze L2 ve Voltech
	<i>Voltage_L3</i>	REAL	Napětí fáze L3 ve Voltech
	<i>Current_L1</i>	REAL	Proud ve fázi L1 v Ampérech
	<i>Current_L2</i>	REAL	Proud ve fázi L2 v Ampérech
	<i>Current_L3</i>	REAL	Proud ve fázi L3 v Ampérech
	<i>CosPhi_L1</i>	REAL	Kosinus ϕ fáze L1
	<i>CosPhi_L2</i>	REAL	Kosinus ϕ fáze L2
	<i>CosPhi_L3</i>	REAL	Kosinus ϕ fáze L3
	<i>ActivePower</i>	REAL	Okamžitá spotřeba ve Watech
	<i>UserValue1</i>	STRING[20]	Hodnota OBIS položky, která je požadovaná ve vstupní proměnné UserCode1
	<i>UserValue2</i>	STRING[20]	Hodnota OBIS položky, která je požadovaná ve vstupní proměnné UserCode2
	<i>UserValue3</i>	STRING[20]	Hodnota OBIS položky, která je požadovaná ve vstupní proměnné UserCode3

V následujícím příkladu je funkční blok *fbElectricityMeterReader* použit pro načtení údajů z elektroměru. Načtené údaje budou zveřejněny na web stránce. Proměnné *Valid_xxx* jsou použity pro řízení viditelnosti zobrazených údajů, takže viditelné budou pouze údaje, které elektroměr skutečně poskytuje.

```
PROGRAM prgMain
VAR
    start          : bool;
    EMeter         : fbElectricityMeterReader;    // ctení elmeru přes optohlavici
    valid_L1       : bool;
    valid_L2       : bool;
    valid_L3       : bool;
    valid_Export   : bool;
    valid_Power    : bool;
END_VAR

// cyklické čtení údajů z elektroměru
EMeter(Start := start, chanCode := CH1_uni);
IF NOT EMeter.Busy THEN
    start := NOT start;
END_IF;

// pokud některá položka nebyla z elektroměru načtena,
// je nastavena nahodnotu MeterInvalidValue
valid_L1 := (EMETER.Voltage_L1 <> METER_INVALID_VALUE);
valid_L2 := (EMETER.Voltage_L2 <> METER_INVALID_VALUE);
valid_L3 := (EMETER.Voltage_L3 <> METER_INVALID_VALUE);
valid_Export := (EMETER.ActiveEnergyExport_T1 <> METER_INVALID_VALUE);
valid_Power := (EMETER.ActivePower <> METER_INVALID_VALUE);

END_PROGRAM
```

Zobrazení vyčítaných dat ve web stránce může vypadat např. následovně.

Test čtení dat z elektroměru přes optohlavu

Výrobce elektroměru
Identifikační řetězec

Sériové číslo
Počet výpadků napětí

Počítadlo odebrané energie - VT kWh

Počítadlo odebrané energie - NT kWh

Počítadlo dodané energie - VT kWh

Počítadlo dodané energie - NT kWh

Aktuální odběr W

Načíst data

Síťové napětí

L1: V
L2: V
L3: V

Odebíraný proud

L1: A
L2: A
L3: A

cos phi

L1:
L2:
L3:

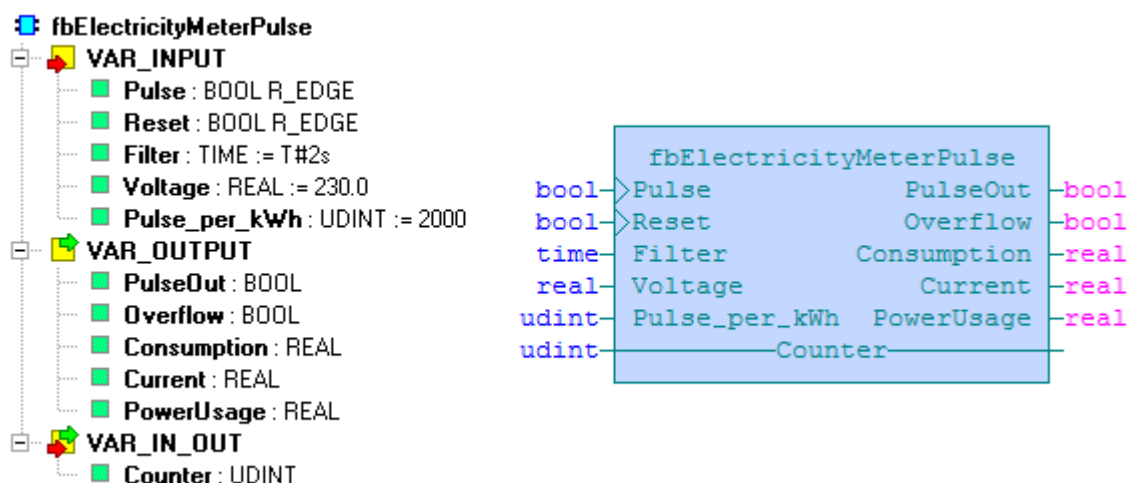
Uživatelský OBIS kód položky hodnota:

Nastavení přenosových parametrů sériového kanálu, na který je připojena optická čtecí hlava, je vidět z následujícího dialogu. Sériový kanál musí být nastaven v režimu uni, komunikační rychlost je 300 Baud, formát dat 7 bitů se sudou paritou, délka přijímací zóny 64 bytů, délka vysílací zóny 64 bytů, minimální délka klidu na lince mezi přijímanými zprávami je 0, minimální délka klidu na lince mezi vysílanými zprávami je 4.

Nastavení univerzálního režimu kanálu

Přijímací zóna Délka zóny: 64 Adresa zóny: ☐ 4 Přijímací zóna: CH1_ZoneIN	Vysílací zóna Délka zóny: 64 Adresa zóny: ☐ 4 Vysílací zóna: CH1_ZoneOUT	Komunikační rychlost: 300 Formát dat: 7b sudá parita
Počáteční znak ☐ Detekovat ☐ Vysílat Kód znaku: 0	Koncový znak ☐ Detekovat ☐ Vysílat ☐ Dva znaky Kód znaku: 0 0	Adresa stanice Adresa stanice: 0 ☐ Detekovat při příjmu ☐ Zápis při vysílání
Parita prvního bytu přijímané zprávy ☒ Stejná parita jako u ostatních ☐ Opačná parita než u ostatních	Parita prvního bytu vysílané zprávy ☒ Stejná parita jako u ostatních ☐ Opačná parita než u ostatních	Kontrolní součet ☐ Kontrola při příjmu ☐ Výpočet při vysílání Poz. prvního znaku CHS: 0
Potvrzení zprávy bez dat ☐ Detekovat ☐ Vysílat ☐ Dva znaky Kód znaku: 0 0	Délka zprávy ☐ Detekovat při příjmu ☐ Zápis při vysílání Pozice délky zprávy: 0 Maximální délka: 0	Režim řízení modemových signálů Řízení signálu RTS: podle SIGN.1 (dle programu) Řízení signálu DTR: trvale hodnota 0 ☐ Odpojení přijímače během vysílání
Min. doba klidu na lince mezi přijímanými zprávami (počet bytů): 0 Min. doba klidu na lince mezi vysílanými zprávami (počet bytů): 4		

6.2 Funkční blok *fbElectricityMeterPulse*

Knihovna : *EnergyLib*

Funkční blok *fbElectricityMeterPulse* slouží k výpočtu okamžité spotřeby, celkové spotřeby a proudu na základě pulzů poskytovaných elektroměrem. Pozor, hodnoty okamžitého spotřeby a proudu jsou pouze orientační. Pokud hodnoty aktuální spotřeby a proudu vykazují příliš prudké změny je možné upravit jejich průběh filtrem. Aktuální proud je počítán dle zadaného napětí na vstupu *Voltage* dle vzorce:

$$Current = \frac{Consumption}{Voltage}$$

Celková spotřeba je uchovávána v proměnné na vstupu *Couter* jako počet pulzů. Maximalní celková spotřeba je 4294967295 pulzů. Po dosažení této hodnoty je nastaven příznak *Overflow* a čítání se zastaví. Pro většinu aplikací by neměl limit čítače být překážkou.

Čítat celkové spotřeby je možné vynulovat náběžnou hranou na vstupu *Reset*.

Pro zachování hodnoty celkové spotřeby během výpadků napájení je nutné proměnnou na vstupu *Counter* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Pulse</i>	BOOL R_EDGE	Pulzy z elektroměru
	<i>Reset</i>	BOOL R_EDGE	Nulování čítače celkové spotřeby
	<i>Filter</i>	TIME	Časová konstanta filtru okamžité spotřeby (implicitní hodnota 2 sec)
	<i>Voltage</i>	REAL	Napětí (implicitní hodnota 230V)
	<i>Pulse_per_kWh</i>	UDINT	Počet pulzů na jednu kWh (implicitní hodnota 2000 pulzů/kWh)

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_OUTPUT			
	<i>PulseOut</i>	BOOL	Kopie vstupních pulzů
	<i>Overflow</i>	BOOL	Příznak naplnění čítače – čítání je zastaveno, je nutné vynulovat čítač celkové spotřeby (<i>Reset</i>)
	<i>Consumption</i>	REAL	Okamžitá spotřeba [kW]
	<i>Current</i>	REAL	Proud [A]
	<i>PowerUsage</i>	REAL	Celková spotřeba [kWh]
VAR_IN_OUT			
	<i>Counter</i>	UDINT	Počítadlo pulzů celkové spotřeby (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbElectricityMeterPulse* použit pro výpočet aktuální a celkové spotřeby.

```

VAR_GLOBAL RETAIN
  ECounter : UDINT;           // čítač pulzů pro výpočet celkové spotřeby
END_VAR

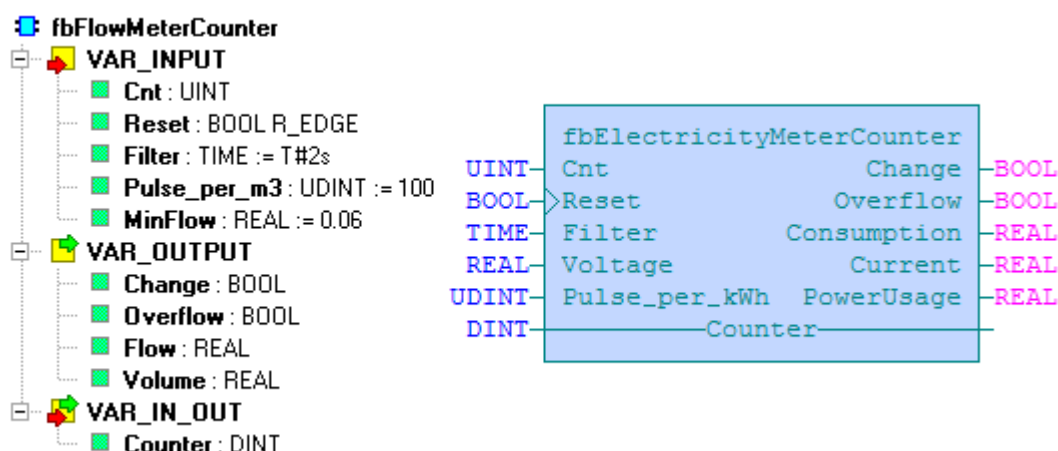
PROGRAM prgExampleMeterPulse
  VAR
    PMeter      : fbElectricityMeterPulse; // FB pro zpracování pulzů z elměru
    aktSpotreba : REAL;
    celkSpotreba : REAL;
    proud       : REAL;
  END_VAR

  PMeter( Pulse      := r0_p3_DI.DI0,
          Filter      := T#2s,
          Voltage     := 230.0,
          Pulse_per_kWh := 2000,
          Counter     := ECounter,
          Consumption => aktSpotreba,
          PowerUsage  => celkSpotreba,
          Current     => proud);

END_PROGRAM

```

6.3 Funkční blok *fbElectricityMeterCounter*

Knihovna : *EnergyLib*

Funkční blok *fbElectricityMeterCounter* je variantou bloku *fbElectricityMeterPulse*, pro případ, že vstupem nejsou pulsy, ale hodnota čítače. Výpočet okamžité spotřeby a proudu je u obou bloků shodný.

Blok je primárně navržen pro spolupráci s modulem C-AM-0600I. Z tohoto důvodu je vstup pro hodnotu čítače *Cnt* typu UINT. Blok lze použít i s 32bitovými čítači s použitím konverzní funkce UDINT_TO_UINT.

Celková spotřeba je uchovávána v proměnné na vstupu *Couter* jako počet pulzů. Maximalní celková spotřeba je 2147483647 pulzů. Po dosažení této hodnoty je nastaven příznak *Overflow* a čítání se zastaví. Pro většinu aplikací by neměl limit čítače být překážkou.

Čítat celkové spotřeby je možné vynulovat náběžnou hranou na vstupu *Reset*.

Pro zachování hodnoty celkové spotřeby během výpadků napájení je nutné proměnnou na vstupu *Counter* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Cnt</i>	UINT	Čítač pulzů z elektroměru
	<i>Reset</i>	BOOL R_EDGE	Nulování čítače celkové spotřeby
	<i>Filter</i>	TIME	Časová konstanta filtru okamžité spotřeby (implicitní hodnota 2 sec)
	<i>Voltage</i>	REAL	Napětí (implicitní hodnota 230V)
	<i>Pulse_per_kWh</i>	UDINT	Počet pulzů na jednu kWh (implicitní hodnota 2000 pulzů/kWh)

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_OUTPUT			
	<i>Change</i>	BOOL	Čítač pulzů změnil hodnotu
	<i>Overflow</i>	BOOL	Příznak naplnění čítače – čítání je zastaveno, je nutné vynulovat čítač celkové spotřeby (<i>Reset</i>)
	<i>Consumption</i>	REAL	Okamžitá spotřeba [kW]
	<i>Current</i>	REAL	Proud [A]
	<i>PowerUsage</i>	REAL	Celková spotřeba [kWh]
VAR_IN_OUT			
	<i>Counter</i>	DINT	Počítadlo pulzů celkové spotřeby (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbElectricityMeterPulse* použit pro výpočet aktuální a celkové spotřeby.

```

VAR_GLOBAL RETAIN
  ECounter2 : DINT;           // čítač pulzů pro výpočet celkové spotřeby
END_VAR

PROGRAM prgExampleMeterCounter
  VAR
    PMeter : fbElectricityMeterCounter; //FB pro zpracování pulzů z elměru
    aktSpotreba : REAL;
    celkSpotreba : REAL;
    proud : REAL;
  END_VAR

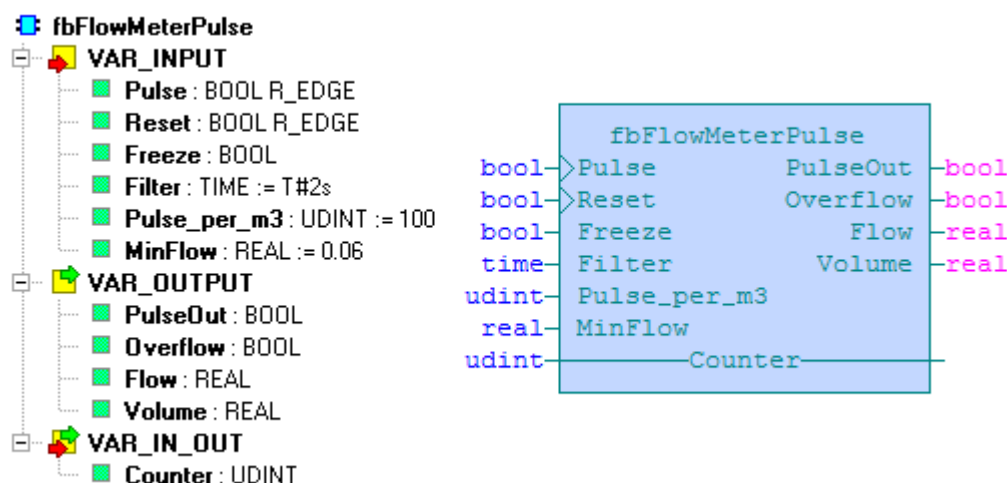
  PMeter( Cnt      := C_AM_0600I_IN.AI1,
          Filter    := T#2s,
          Voltage   := 230.0,
          Pulse_per_kWh := 2000,
          Counter   := ECounter2,
          Consumption => aktSpotreba,
          PowerUsage  => celkSpotreba,
          Current    => proud);

END_PROGRAM

```

6.4 Funkční blok fbFlowMeterPulse

Knihovna : EnergyLib



Funkční blok *fbFlowMeterPulse* slouží k výpočtu okamžitého průtoku a celkového proteklého množství (objemu) na základě pulzů poskytovaných průtokoměrem. Pozor, hodnota okamžitého průtoku je pouze orientační. Pokud hodnota okamžitého průtoku vykazuje příliš prudké změny je možné upravit její průběh filtrem.

Pokud je k dispozici informace o běhu čerpadla v uzavřeném okruhu, lze její negaci využít k vnucení nulového průtoku v čase, kdy čerpadlo neběží a tím zpřesnit měření. K tomuto účelu slouží vstup *Freeze*.

Celkové proteklé množství je uchováváno v proměnné na vstupu *Counter* jako počet pulzů. Maximalní proteklé množství je 4294967295 pulzů. Po dosažení této hodnoty je nastaven příznak *Overflow* a čítání se zastaví. Pro většinu aplikací by neměl limit čítače být překážkou.

Čítat celkové spotřeby je možné vynulovat náběžnou hranou na vstupu *Reset*.

Pro zachování hodnoty celkového proteklého množství během výpadků napájení je nutné proměnnou na vstupu *Counter* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Pulse</i>	BOOL R_EDGE	Pulzy z průtokoměru
	<i>Reset</i>	BOOL R_EDGE	Nulování čítače celkového objemu
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Filter</i>	TIME	Časová konstanta filtru okamžité spotřeby (implicitní hodnota 2 sec)
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l) (implicitní hodnota 100 pulzů/m³)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m³/h]

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_OUTPUT			
	<i>PulseOut</i>	BOOL	Kopie vstupních pulzů
	<i>Overflow</i>	BOOL	Příznak naplnění čítače – čítání je zastaveno, je nutné vynulovat čítač celkové objemu (<i>Reset</i>)
	<i>Flow</i>	REAL	Okamžitý průtok (přibližná hodnota) [m³/h]
	<i>PowerUsage</i>	REAL	Celkový objem [m³]
VAR_IN_OUT			
	<i>Counter</i>	UDINT	Počítadlo pulzů celkového průtoku (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbFlowMeterPulse* použit pro výpočet aktuálního a celkové spotřeby.

```

VAR_GLOBAL RETAIN
  FCounter : UDINT;           // čítač pulzů pro výpočet celkové spotřeby
END_VAR

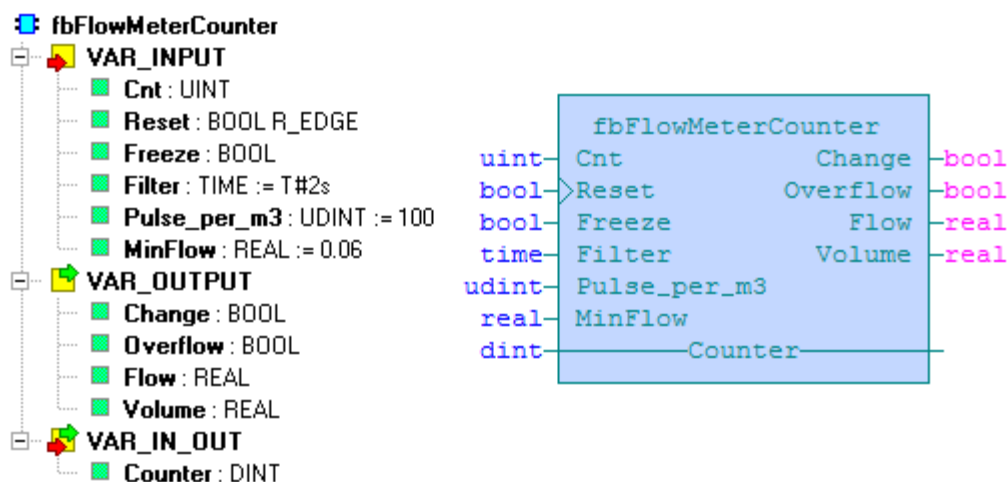
PROGRAM prgExampleFlowMeterPulse
  VAR
    FMeter      : fbFlowMeterPulse;    // FB pro zpracování pulzů
    aktPrutok   : REAL;
    celkObjem   : REAL;
  END_VAR

  FMeter( Pulse      := r0_p3_DI.DI0,
           Filter     := T#2s,
           Pulse_per_m3 := 100,
           MinFlow    := 0.06,
           Counter    := FCounter,
           Flow       => aktPrutok,
           Volume     => celkObjem);

END_PROGRAM

```

6.5 Funkční blok *fbFlowMeterCounter*

Knihovna : *EnergyLib*

Funkční blok *fbFlowMeterCounter* je variantou bloku *fbFlowMeterPulse* pro případ, že vstupem nejsou pulsy, ale hodnota čítače. Výpočet okamžitého průtoku je u obou bloků shodný.

Blok je primárně navržen pro spolupráci s modulem C-AM-0600I. Z tohoto důvodu je vstup pro hodnotu čítače *Cnt* typu UINT. Blok lze použít i s 32bitovými čítači s použitím konverzní funkce UDINT_TO_UINT.

Pokud je k dispozici informace o běhu čerpadla v uzavřeném okruhu, lze její negaci využít k vnucení nulového průtoku v čase, kdy čerpadlo neběží a tím zpřesnit měření. K tomuto účelu slouží vstup *Freeze*.

Celkové proteklé množství je uchováváno v proměnné na vstupu *Couter* jako počet pulzů. Maximalní proteklé množství je 2147483647 pulzů. Po dosažení této hodnoty je nastaven příznak *Overflow* a čítání se zastaví. Pro většinu aplikací by neměl limit čítače být překážkou.

Čítat celkové spotřeby je možné vynulovat náběžnou hranou na vstupu *Reset*.

Pro zachování hodnoty celkového proteklého množství během výpadků napájení je nutné proměnnou na vstupu *Counter* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>Cnt</i>	UINT	Čítač pulzů z průtokoměru
	<i>Reset</i>	BOOL R_EDGE	Nulování čítače celkového objemu
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Filter</i>	TIME	Časová konstanta filtru okamžité spotřeby (implicitní hodnota 2 sec)
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l) (implicitní hodnota 100 pulzů/m ³)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m ³ /h]

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_OUTPUT			
	<i>Change</i>	BOOL	Čítač pulzů změnil hodnotu
	<i>Overflow</i>	BOOL	Příznak naplnění čítače – čítání je zastaveno, je nutné vynulovat čítač celkové objemu (<i>Reset</i>)
	<i>Flow</i>	REAL	Okamžitý průtok (přibližná hodnota) [m³/h]
	<i>PowerUsage</i>	REAL	Celkový objem [m³]
VAR_IN_OUT			
	<i>Counter</i>	DINT	Počítadlo pulzů celkového průtoku (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbFlowMeterPulse* použit pro výpočet aktuálního a celkové spotřeby.

```

VAR_GLOBAL RETAIN
  FCounter2 : DINT;           // čítač pulzů pro výpočet celkové spotřeby
END_VAR

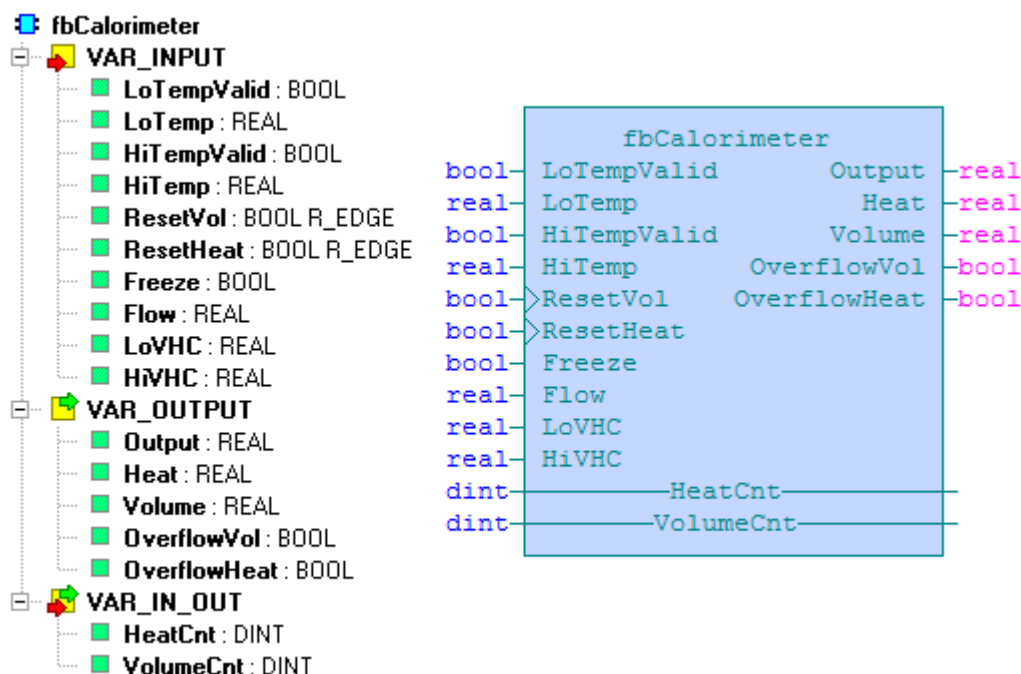
PROGRAM prgExampleFlowMeterCounter
  VAR
    FMeter      : fbFlowMeterCounter;    // FB pro zpracování pulzů
    aktPrutok   : REAL;
    celkObjem   : REAL;
  END_VAR

  FMeter( Cnt      := UDINT_TO_UINT(r0_p3_CNT_IN1.VALA),
          Filter    := T#2s,
          Pulse_per_m3 := 100,
          MinFlow   := 3.6,
          Counter   := FCounter2,
          Flow      => aktPrutok,
          Volume    => celkObjem);

END_PROGRAM

```

6.6 Funkční blok *fbCalorimeter*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeter* slouží k výpočtu spotřebovaného nebo dodaného tepla, aktuálního výkonu a celkového proteklého objemu teplotnosného média.

Na vstupech *HiTemp* a *LoTemp* se očekává teplota vstupujícího a výstupujícího média. Aby byl přírůstek tepla kladný musí platit $HiTemp > LoTemp$. Vstupy *LoTempValid* a *HiTempValid* podmiňují platnost měřených teplot pro měření tepla a výkonu. Pokud nejsou oba vstupy *LoTempValid* a *HiTempValid* v logické 1, je počítáno pouze proteklé množství.

Aktuální průtok se předává na vstupu *Flow* v m^3 za hodinu ($0,06 m^3/h = 1 l/m$).

Pokud je k dispozici informace o běhu čerpadla v uzavřeném okruhu, lze její negaci využít k vnucení nulového průtoku v čase, kdy čerpadlo neběží a tím zpřesnit měření. K tomuto účelu slouží vstup *Freeze*.

Vstupy *HiVHC* a *LoVHC* slouží pro zadání objemové tepelné kapacity teplotnosného média v $kJ/m^3 K$. Tyto hodnoty jsou závislé na teplotě, a proto je nutné předávat hodnotu odpovídající teplotě *HiTemp* (*HiVHC*) a *LoTemp* (*LoVHC*) zvlášť. Konstantní tabulkovou hodnotu lze použít pouze u orientačních měření. Pokud je teplotnosným médiem voda lze použít blok *fbCalorimeterWater*, který již výpočet objemové tepelné kapacity ve vztahu k měřeným teplotám obsahuje.

Celkový proteklý objem je uchováván v proměnné na vstupu *VolumeCnt* jako počet proteklých litrů. Maximální proteklé množství je 2147483647 litrů. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač proteklého objemu je možné vynulovat náběžnou hranou na vstupu *ResetVol*.

Teplo je čítáno do proměnné na vstupu *HeatCnt* jako počet dodaných/spotřebovaných watt hodin. Maximální celkové teplo je 2,147483647 GWh. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač tepla je možné vynulovat náběžnou hranou na vstupu *ResetHeat*.

Pro zachování hodnoty celkového proteklého množství a celkového tepla během výpadků napájení je nutné proměnné na vstupu *HeatCnt* a *VolumeCnt* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Flow</i>	REAL	Objemový průtok [m ³ /h]
	<i>LowVHC</i>	REAL	Objemová tepelná kapacita pro nižší teplotu [kJm ⁻³ h ⁻¹]
	<i>HiVHC</i>	REAL	Objemová tepelná kapacita pro vyšší teplotu [kJm ⁻³ h ⁻¹]
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m ³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeter* použit pro výpočet dodaného tepla a aktuálního výkonu pro ethylglykolovou směs jako teponosné médium. Pro určení objemové tepelné kapacity je použita funkce *EthyleneGlycol_WHC*.

```

VAR_GLOBAL RETAIN
  HeatCounter : DINT;
  VolCounter  : DINT;
END_VAR

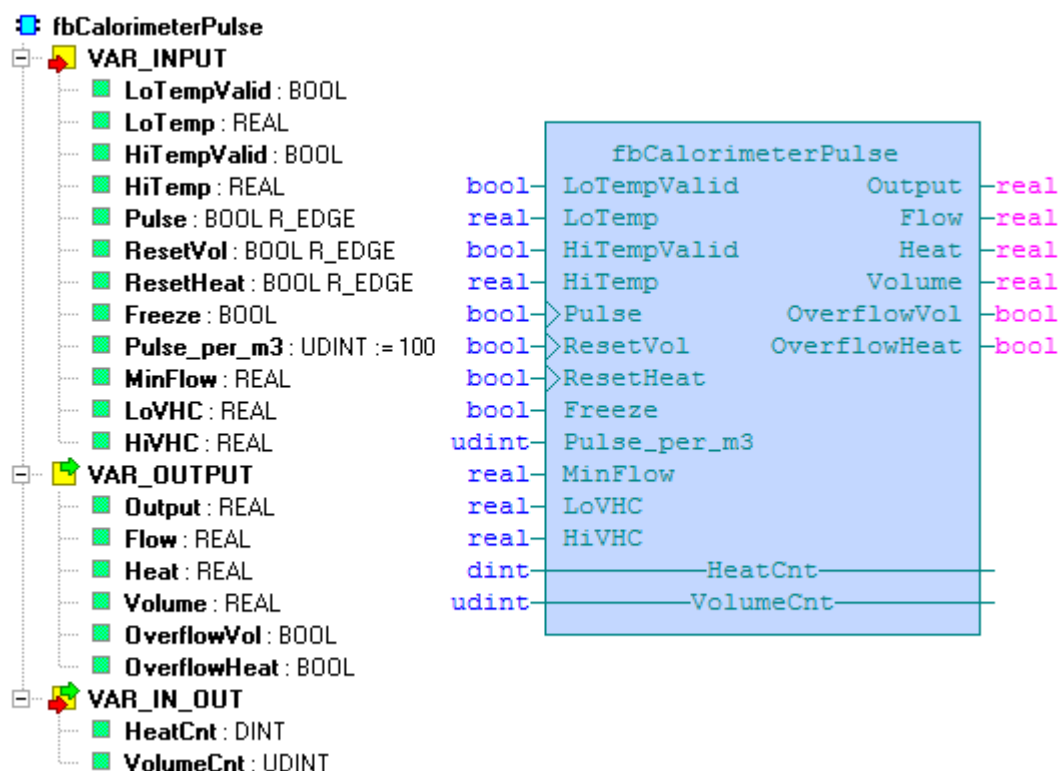
VAR_GLOBAL CONSTANT
  AnalogValid : TAIStat := ( UNF := false, UNR := false,
                              OVR := false, OVF := false,
                              FLS := false);
END_VAR

PROGRAM prgExampleCalorimeter
  VAR
    Calorimeter : fbCalorimeter;
    aktVykonn   : REAL;
    celkTeplo   : REAL;
  END_VAR

  Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
              LoTemp      := r0_p3_AI0.ENG,
              HiTempValid := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                              NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
              HiTemp      := MI_CIB1_IN.ID1_IN.AI1,
              Flow        := MI_CIB1_IN.ID1_IN.AV23.FLOW / 0.06,
              LoVHC       := EthyleneGlycol_WHC(Temp := r0_p3_AI0.ENG,
                                                  Glycol := 0.5),
              HiVHC       := EthyleneGlycol_WHC(Temp := MI_CIB1_IN.ID1_IN.AI1,
                                                  Glycol := 0.5),
              HeatCnt     := HeatCounter,
              VolumeCnt   := VolCounter,
              Output => aktVykonn,
              Heat => celkTeplo);
END_PROGRAM

```

6.7 Funkční blok *fbCalorimeterPulse*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeterPulse* slouží k výpočtu spotřebovaného nebo dodaného tepla, aktuálního výkonu a celkového proteklého objemu teplotnosného média. Jedná se o variantu bloku *fbCalorimeter* s tím rozdílem, že průtok je aproximován z pulzů průtokoměru.

Pokud je k dispozici informace o běhu čerpadla v uzavřeném okruhu, lze její negaci využít k vnucení nulového průtoku v čase, kdy čerpadlo neběží a tím zpřesnit měření. K tomuto účelu slouží vstup *Freeze*.

Na vstupech *HiTemp* a *LoTemp* se očekává teplota vstupujícího a výstupujícího média. Aby byl přírůstek tepla kladný musí platit $HiTemp > LoTemp$. Vstupy *LoTempValid* a *HiTempValid* podmiňují platnost měřených teplot pro měření tepla a výkonu. Pokud nejsou oba vstupy *LoTempValid* a *HiTempValid* v logické 1, je počítáno pouze proteklé množství.

Pulzy průtokoměru se předávají přes vstup *Pulse*.

Vstupy *HiVHC* a *LoVHC* slouží pro zadání objemové tepelné kapacity teplotnosného média v $\text{kJ/m}^3 \text{K}$. Tyto hodnoty jsou závislé na teplotě, a proto je nutné předávat hodnotu odpovídající teplotě *HiTemp* (*HiVHC*) a *LoTemp* (*LoVHC*) zvlášť. Konstantní tabulkovou hodnotu lze použít pouze u orientačních měření. Pokud je teplotnosným médiem voda lze použít blok *fbCalorimeterPulseWater*, který již výpočet objemové tepelné kapacity ve vztahu k měřeným teplotám obsahuje.

Hodnoty aktuálního výkonu *Output* a průtoku *Flow* jsou pouze orientační. Stejně je zatížen chybou aproximace průtoku z pulzů i výpočet celkového tepla *Heat*, jedná se tedy také o orientační hodnotu.

Celkový proteklý objem je uchováváno v proměnné na vstupu *VolumeCnt* jako počet pulzů. Maximalní proteklé množství je 4294967295 pulzů. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač proteklého objemu je možné vynulovat náběžnou hranou na vstupu *ResetVol*.

Teplo je čítáno do proměnné na vstupu *HeatCnt* jako počet dodaných/spotřebovaných watt hodin. Maximální celkové teplo je 2,147483647 GWh. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač tepla je možné vynulovat náběžnou hranou na vstupu *ResetHeat*.

Pro zachování hodnoty celkového proteklého množství a celkového tepla během výpadků napájení je nutné proměnné na vstupu *HeatCnt* a *VolumeCnt* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>Pulse</i>	BOOL R_EDGE	Pulz z průtokoměru
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m ³ /h]
	<i>LowVHC</i>	REAL	Objemová tepelná kapacita pro nižší teplotu [kJm ⁻³ h ⁻¹]
	<i>HiVHC</i>	REAL	Objemová tepelná kapacita pro vyšší teplotu [kJm ⁻³ h ⁻¹]
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m ³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeterPulse* použit pro výpočet dodaného tepla a aktuálního výkonu pro propylenglykolovou směs jako teplotnosné médium. Pro určení objemové tepelné kapacity je použita funkce *PropyleneGlycol_WHC*.

```

VAR_GLOBAL RETAIN
    HeatCounter : DINT;
    VolCounter2 : UDINT;
END_VAR

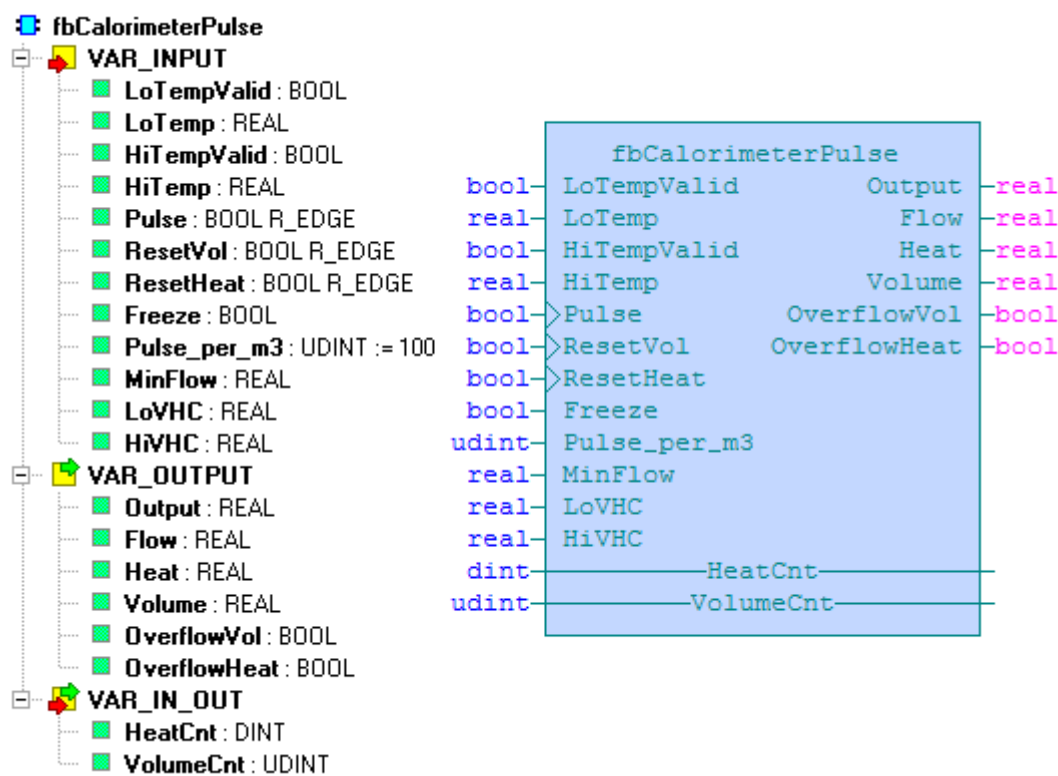
VAR_GLOBAL CONSTANT
    AnalogValid : TAISTat := ( UNF := false, UNR := false,
                                OVR := false, OVF := false,
                                FLS := false );
END_VAR

PROGRAM prgExampleCalorimeterPulse
    VAR
        Calorimeter : fbCalorimeterPulse;
        aktVykony   : REAL;
        celkTeplo   : REAL;
    END_VAR

    Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
                LoTemp      := r0_p3_AI0.ENG,
                HiTempValid := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                                NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
                HiTemp      := MI_CIB1_IN.ID1_IN.AI1,
                Pulse       := r0_p3_DI.DI0,
                LoVHC       := PropyleneGlycol_WHC(Temp := r0_p3_AI0.ENG,
                                                    Glycol := 0.5),
                HiVHC       := PropyleneGlycol_WHC(Temp := MI_CIB1_IN.ID1_IN.AI1,
                                                    Glycol := 0.5),
                Pulse_per_m3 := 1000,
                HeatCnt      := HeatCounter,
                VolumeCnt    := VolCounter2,
                Output => aktVykony,
                Heat => celkTeplo);
END_PROGRAM

```

6.8 Funkční blok *fbCalorimeterCounter*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeterCounter* je variantou bloku *fbCalorimeterPulse* pro případ, že vstupem nejsou pulsy, ale hodnota čítače. Funkce obou bloků je po všech ostatních stránkách shodná.

Blok je primárně navržen pro spolupráci s modulem C-AM-0600I. Z tohoto důvodu je vstup pro hodnotu čítače *Cnt* typu UINT. Blok lze použít i s 32bitovými čítači s použitím konverzní funkce UDINT_TO_UINT.

Pokud je k dispozici informace o běhu čerpadla v uzavřeném okruhu, lze její negaci využít k vnucení nulového průtoku v čase, kdy čerpadlo neběží a tím zpřesnit měření. K tomuto účelu slouží vstup *Freeze*.

Celkový proteklý objem je uchováváno v proměnné na vstupu *VolumeCnt* jako počet pulzů. Maximální proteklé množství je 2147483647 pulzů. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač proteklého objemu je možné vynulovat náběžnou hranou na vstupu *ResetVol*.

Teplo je čítáno do proměnné na vstupu *HeatCnt* jako počet dodaných/spotřebovaných watt hodin. Maximální celkové teplo je 2,147483647 GWh. Po dosažení této hodnoty dojde k zastavení čítání na této hodnotě a nastavení příznaku *OverflowVol*.

Čítač tepla je možné vynulovat náběžnou hranou na vstupu *ResetHeat*.

Pro zachování hodnoty celkového proteklého množství a celkového tepla během výpadků napájení je nutné proměnné na vstupu *HeatCnt* a *VolumeCnt* definovat jako VAR_GLOBAL RETAIN.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>Cnt</i>	UINT	Čítač pulzů z průtokoměru
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m³/h]
	<i>LowVHC</i>	REAL	Objemová tepelná kapacita pro nižší teplotu [kJm⁻³h⁻¹]
	<i>HiVHC</i>	REAL	Objemová tepelná kapacita pro vyšší teplotu [kJm⁻³h⁻¹]
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeterCounter* použit pro výpočet dodaného tepla a aktuálního výkonu pro vodu jako teplotnosné médium. Pro určení objemové tepelné kapacity je použita funkce *Water_WHC*.

```

VAR_GLOBAL RETAIN
  HeatCounter : DINT;
  VolCounter  : DINT;
END_VAR

VAR_GLOBAL CONSTANT
  AnalogValid : TAIStat := ( UNF := false, UNR := false,
                              OVR := false, OVF := false,
                              FLS := false);
END_VAR

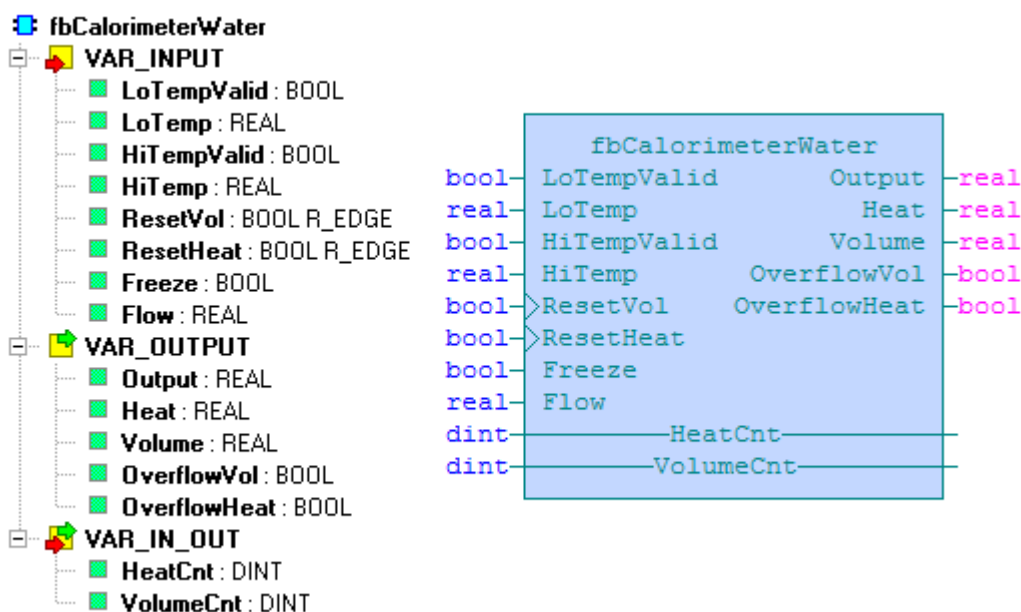
PROGRAM prgExampleCalorimeterCounter
  VAR
    Calorimeter : fbCalorimeterCounter;
    aktVykonn   : REAL;
    celkTeplo   : REAL;
  END_VAR

  Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
              LoTemp      := r0_p3_AI0.ENG,
              HiTempValid := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                              NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
              HiTemp      := MI_CIB1_IN.ID1_IN.AI1,
              Cnt         := MI_CIB1_IN.ID1_IN.AI5,
              LoVHC       := Water_WHC(Temp := r0_p3_AI0.ENG),
              HiVHC       := Water_WHC(Temp := MI_CIB1_IN.ID1_IN.AI1),
              Pulse_per_m3 := 1000,
              HeatCnt      := HeatCounter,
              VolumeCnt    := VolCounter,
              Output => aktVykonn,
              Heat => celkTeplo);

END_PROGRAM

```

6.9 Funkční blok *fbCalorimeterWater*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeterWater* je verzí bloku *fbCalorimeter* pro případ, kdy je teplosným médiem voda. Blok má integrovanou funkci *Water_WHC*. Tím odpadá nutnost vstupů pro objemovou tepelnou kapacitu. Funkčnost bloku je jinak identická s *fbCalorimeter*.

Popis proměnných :

	Proměnná	Typ	Význam
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Flow</i>	REAL	Objemový průtok [m ³ /h]

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeterWater* použit pro výpočet dodaného tepla a aktuálního výkonu pro vodu jako teplotnosné médium.

```

VAR_GLOBAL RETAIN
    HeatCounter : DINT;
    VolCounter  : DINT;
END_VAR

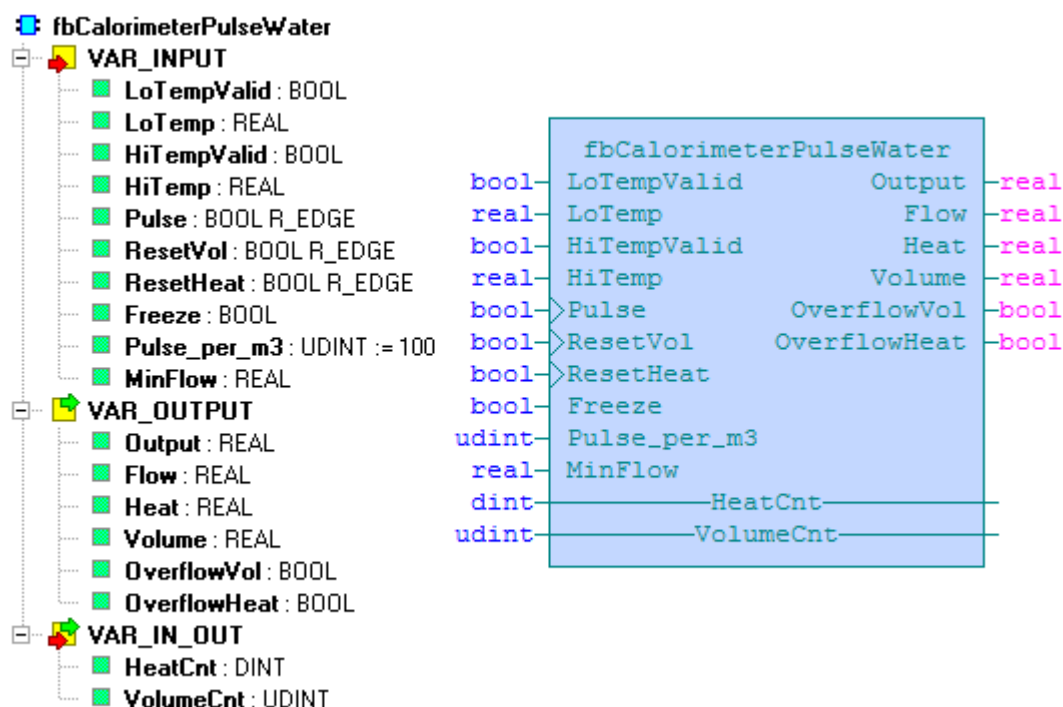
VAR_GLOBAL CONSTANT
    AnalogValid : TAIStat := ( UNF := false, UNR := false,
                                OVR := false, OVF := false,
                                FLS := false );
END_VAR

PROGRAM prgExampleCalorimeterWater
    VAR
        Calorimeter : fbCalorimeterWater;
        aktVykony    : REAL;
        celkTeplo    : REAL;
    END_VAR

    Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
                LoTemp      := r0_p3_AI0.ENG,
                HiTempValid := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                                NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
                HiTemp      := MI_CIB1_IN.ID1_IN.AI1,
                Flow        := MI_CIB1_IN.ID1_IN.AV23.FLOW / 0.06,
                HeatCnt     := HeatCounter,
                VolumeCnt   := VolCounter,
                Output      => aktVykony,
                Heat        => celkTeplo);
END_PROGRAM

```

6.10 Funkční blok *fbCalorimeterPulseWater*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeterPulseWater* je verzí bloku *fbCalorimeterPulse* pro případ, kdy je teplotním médiem voda. Blok má integrovanou funkci *Water_WHC*. Tím odpadá nutnost vstupů pro objemovou tepelnou kapacitu. Funkčnost bloku je jinak identická s *fbCalorimeterPulse*.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>Pulse</i>	BOOL R_EDGE	Pulz z průtokoměru
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m ³ /h]
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m ³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeterPulseWater* použit pro výpočet dodaného tepla a aktuálního výkonu pro vodu jako teplotonosné médium.

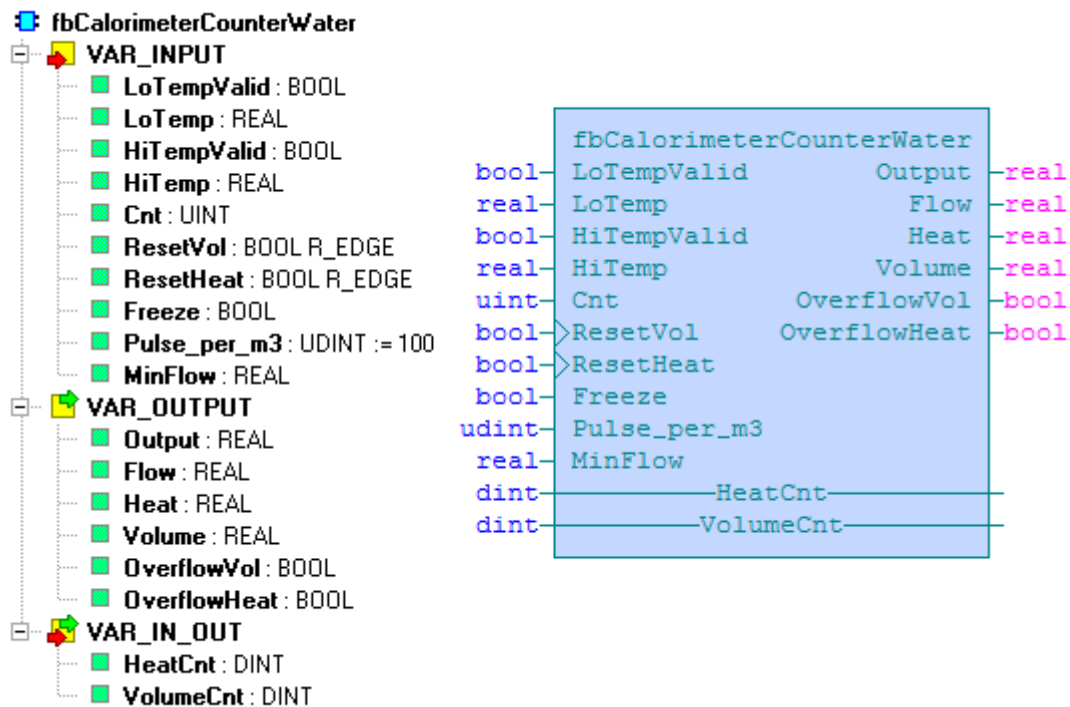
```
VAR_GLOBAL RETAIN
  HeatCounter : DINT;
  VolCounter2 : UDINT;
END_VAR

VAR_GLOBAL CONSTANT
  AnalogValid : TAISTat := ( UNF := false, UNR := false,
                             OVR := false, OVF := false,
                             FLS := false);
END_VAR

PROGRAM prgExampleCalorimeterPulseWater
  VAR
    Calorimeter : fbCalorimeterPulseWater;
    aktVykony    : REAL;
    celkTeplo    : REAL;
  END_VAR

  Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
              LoTemp       := r0_p3_AI0.ENG,
              HiTempValid  := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                             NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
              HiTemp       := MI_CIB1_IN.ID1_IN.AI1,
              Pulse        := r0_p3_DI.DI0,
              Pulse_per_m3 := 1000,
              HeatCnt       := HeatCounter,
              VolumeCnt     := VolCounter2,
              Output => aktVykony,
              Heat => celkTeplo);
END_PROGRAM
```

6.11 Funkční blok *fbCalorimeterCounterWater*

Knihovna : *EnergyLib*

Funkční blok *fbCalorimeterCounterWater* je verzí bloku *fbCalorimeterCounter* pro případ, kdy je teplotním médiem voda. Blok má integrovanou funkci *Water_WHC*. Tím odpadá nutnost vstupů pro objemovou tepelnou kapacitu. Funkčnost bloku je jinak identická s *fbCalorimeterCounter*.

Popis proměnných :

	<i>Proměnná</i>	<i>Typ</i>	<i>Význam</i>
VAR_INPUT			
	<i>LoTempValid</i>	BOOL	Nižší teplota je platná
	<i>LoTemp</i>	REAL	Nižší teplota
	<i>HiTempValid</i>	BOOL	Vyšší teplota je platná
	<i>HiTemp</i>	REAL	Vyšší teplota
	<i>Cnt</i>	UINT	Čítač pulzů z průtokoměru
	<i>ResetVol</i>	BOOL R_EDGE	Nulování počítadla proteklého objemu
	<i>ResetHeat</i>	BOOL R_EDGE	Nulování počítadla tepla
	<i>Freeze</i>	BOOL	Blokování čítání od vypnutého čerpadla
	<i>Pulse_per_m3</i>	UDINT	Počet pulzů na jeden kubický metr (1000 l)
	<i>MinFlow</i>	REAL	Minimální měřitelný průtok [m³/h]
VAR_OUTPUT			
	<i>Output</i>	REAL	Tepelný výkon [kW]
	<i>Heat</i>	REAL	Celkové teplo [kWh]
	<i>Volume</i>	REAL	Celkový objem [m³]
	<i>OverflowVol</i>	BOOL	Přetečení proteklého objemu
	<i>OverflowHeat</i>	BOOL	Přetečení čítače tepla
VAR_IN_OUT			
	<i>HeatCnt</i>	DINT	Počítadlo teplo [Wh] (musí být RETAIN!)
	<i>VolumeCnt</i>	DINT	Počítadlo proteklého množství [l] (musí být RETAIN!)

V následujícím příkladu je funkční blok *fbCalorimeterCounterWater* použit pro výpočet dodaného tepla a aktuálního výkonu pro vodu jako teplotonosné médium.

```

VAR_GLOBAL RETAIN
  HeatCounter : DINT;
  VolCounter  : DINT;
END_VAR

VAR_GLOBAL CONSTANT
  AnalogValid : TAISTat := ( UNF := false, UNR := false,
                             OVR := false, OVF := false,
                             FLS := false);
END_VAR

PROGRAM prgExampleCalorimeterCounterWater
  VAR
    Calorimeter : fbCalorimeterCounterWater;
    aktVykony   : REAL;
    celkTeplo   : REAL;
  END_VAR

  Calorimeter(LoTempValid := r0_p3_AI0.STAT = AnalogValid,
              LoTemp      := r0_p3_AI0.ENG,
              HiTempValid := MI_CIB1_IN.ID1_IN.STAT.VLD1 AND
                           NOT MI_CIB1_IN.ID1_IN.STAT.OUF1,
              HiTemp      := MI_CIB1_IN.ID1_IN.AI1,
              Cnt         := MI_CIB1_IN.ID1_IN.AI5,
              Pulse_per_m3 := 1000,
              HeatCnt     := HeatCounter,
              VolumeCnt   := VolCounter,
              Output      => aktVykony,
              Heat        => celkTeplo);
END_PROGRAM

```




teco

Objednávky a informace:

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

TXV 003 65.01

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