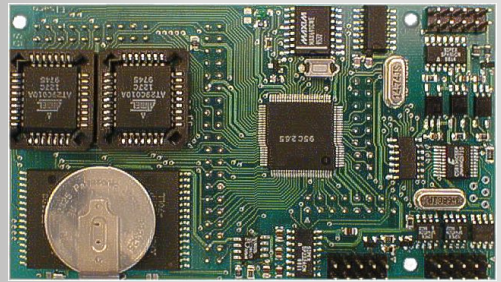


**PROMOS™**  
**line 2**

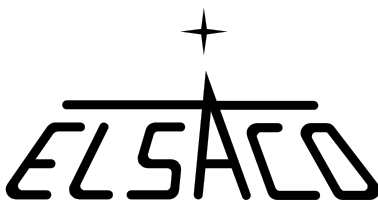


# **Knihovna PL2**

## **grafického vývojového prostředí**

# **F R E D**

**pro vytváření aplikačních programů  
do mikropočítačové stavebnice  
PROMOS line 2**



Jaselská 177, 280 02 KOLÍN 3  
tel./fax: 321 727 753

verze 3.021

© 2007 sdružení ELSACO

4.4.2007

Účelová publikace ELSACO - předběžná

**ELSACO, Jaselská 177, 280 02 Kolín 3**

Tel./fax/modem: 321 727 753 / 321 727 759

**Pobočka v HK:** 605 272 863

Internet : [www.elsaco.cz](http://www.elsaco.cz)

**Připomínky :** [vacek@elsaco.cz](mailto:vacek@elsaco.cz)

BIOS centrál **1.32m**

firmware verze **3.021**

**POZOR - pouze pro Flash 512 kB!**

# O b s a h

---

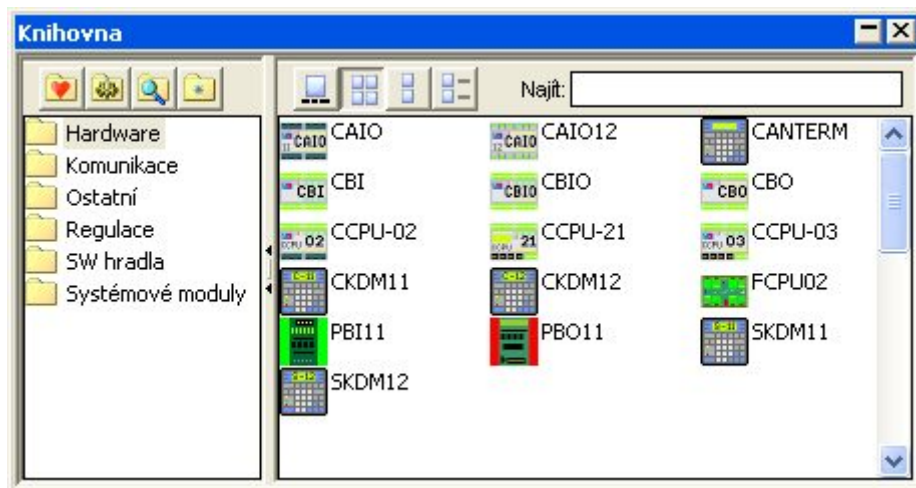
1	Knihovní moduly - obecně . . . . .	7
1.1	Základní vlastnosti modulů. . . . .	7
1.1.1	Rychlost . . . . .	7
1.1.2	Priorita . . . . .	7
1.1.3	Rychlost komunikace . . . . .	8
1.1.4	Vícenásobné použití modulu . . . . .	8
2	Skupina modulů Hardware . . . . .	9
2.1	CCPU-02 . . . . .	10
2.2	CANTERM. . . . .	11
2.3	CAIO . . . . .	12
2.4	CBI . . . . .	14
2.5	CBO . . . . .	15
2.6	CBIO . . . . .	15
2.7	FCPU02 . . . . .	16
2.8	PBI11 . . . . .	17
2.9	PBO11 . . . . .	17
2.10	CKDM11 . . . . .	17
2.11	CKDM12 . . . . .	18
2.12	SKDM11 . . . . .	19
2.13	SKDM12 . . . . .	20
2.14	CCPU-03. . . . .	21
2.15	CAIO12 . . . . .	22
2.16	CCPU-21. . . . .	23
3	Skupina modulů pro komunikaci . . . . .	26
3.1	serialcomm . . . . .	28
3.2	modem . . . . .	29
3.3	svm840/940. . . . .	31
3.4	cf50 . . . . .	31
3.5	danfoss . . . . .	32
3.6	pwpb_main . . . . .	32
3.7	pwpb_rx. . . . .	33
3.8	pwpb_tx. . . . .	33
3.9	pwpbp_rx . . . . .	34
3.10	EmergCall . . . . .	34
3.11	comport . . . . .	35
3.12	cdafoss . . . . .	35
3.13	sdo. . . . .	36
3.14	gsm . . . . .	36
3.15	sms . . . . .	38
3.16	dcf . . . . .	38
3.17	ModbusRX. . . . .	39
3.18	ModbusTX. . . . .	39
3.19	M-Bus . . . . .	40
3.20	EESA. . . . .	41
3.21	SAIOi. . . . .	42
3.22	SAIOo . . . . .	42
3.23	SBI. . . . .	42
3.24	SBO . . . . .	42

3.25 SBIOi . . . . .	43
3.26 SBIOo . . . . .	43
3.27 SFCPUi . . . . .	43
3.28 SFCPUo . . . . .	44
3.29 ReTran . . . . .	44
3.30 PortMix. . . . .	44
3.31 ModbusTXC . . . . .	46
3.32 Inet. . . . .	46
3.33 IEC870 . . . . .	46
3.34 IECinSP . . . . .	48
3.34.1 IECinDP . . . . .	48
3.35 IECinMI . . . . .	49
3.36 IECinMF . . . . .	49
4 OSTATNÍ moduly . . . . .	50
4.1 Vstupní label . . . . .	51
4.2 Výstupní label . . . . .	51
4.3 Textová poznámka . . . . .	51
4.4 Obrázek . . . . .	51
5 Skupina modulů REGULACE . . . . .	53
5.1 ekviterm . . . . .	54
5.2 pid. . . . .	55
5.3 havbin . . . . .	56
5.4 havan . . . . .	57
5.5 body_4 . . . . .	57
5.6 aservo . . . . .	58
5.7 bservo . . . . .	58
5.8 timer. . . . .	59
5.9 stridac. . . . .	60
5.10 urci_tv . . . . .	60
5.11 pockot . . . . .	61
5.12 fastpid . . . . .	61
5.13 minmax . . . . .	62
5.14 R3state. . . . .	63
5.15 SaveEn . . . . .	63
5.16 TimerA . . . . .	63
6 Skupina modulů SW hradel . . . . .	64
6.1 Logická hradla . . . . .	64
6.1.1 and2 . . . . .	64
6.1.2 and10. . . . .	64
6.1.3 or2 . . . . .	65
6.1.4 or10. . . . .	65
6.1.5 xor2. . . . .	65
6.1.6 xor10 . . . . .	65
6.1.7 not . . . . .	65
6.1.8 delay . . . . .	65
6.2 jklop. . . . .	66
6.2.1 rsko. . . . .	66
6.2.2 mono . . . . .	66
6.2.3 ppg . . . . .	66
6.2.4 keyin . . . . .	66

6.2.5 keyout . . . . .	66
6.2.6 setpar. . . . .	67
6.2.7 regf . . . . .	67
6.2.8 regl . . . . .	67
6.2.9 stopw. . . . .	67
6.2.10 State . . . . .	67
6.3 Celočíselná hradla . . . . .	68
6.3.1 imin. . . . .	68
6.3.2 iplus . . . . .	68
6.3.3 idiv . . . . .	68
6.3.4 imul. . . . .	69
6.3.5 table . . . . .	69
6.3.6 cnt . . . . .	69
6.3.7 ctc . . . . .	69
6.3.8 constl. . . . .	70
6.3.9 bin_to_int. . . . .	70
6.3.10 int_to_bin . . . . .	70
6.3.11 int2float . . . . .	70
6.3.12 float2int . . . . .	70
6.3.13 cntud . . . . .	70
6.3.14 l2hwlw. . . . .	71
6.4 Hradla s reálnou aritmetikou . . . . .	71
6.4.1 amul . . . . .	71
6.4.2 adiv . . . . .	71
6.4.3 aplus . . . . .	71
6.4.4 amin . . . . .	71
6.4.5 abig. . . . .	72
6.4.6 asmall . . . . .	72
6.4.7 amux . . . . .	72
6.4.8 amux10. . . . .	72
6.4.9 admx . . . . .	72
6.4.10 admx10 . . . . .	73
6.4.11 almt . . . . .	73
6.4.12 aflt. . . . .	73
6.4.13 compute . . . . .	74
6.4.14 constf . . . . .	75
6.4.15 scale . . . . .	75
7 Skupina Systémových modulů . . . . .	76
7.1 SCREEN . . . . .	77
7.2 sysmon . . . . .	79
7.3 timdat . . . . .	81
7.4 hesla . . . . .	81
7.5 vzorky . . . . .	82
7.6 flash . . . . .	83
7.7 canopen. . . . .	84
7.8 flag . . . . .	84
7.9 journal. . . . .	84
7.10 joornal_txt . . . . .	85
7.11 printer . . . . .	85
7.11.1 HDBmain . . . . .	86
7.11.2 HDBa . . . . .	86

7.11.3 HDBb . . . . .	87
7.11.4 Program. . . . .	88
8 Příloha 1 - Základní funkce regulátoru PL2 . . . . .	89
8.1 Nastavení centrály CCPU-02/03 . . . . .	89
8.2 F1 Download . . . . .	89
8.3 F2 Run . . . . .	89
8.4 F3 Test I/O. . . . .	89
8.4.1 test logických vstupů CBI/CBIO . . . . .	90
8.4.2 test logických výstupů CBO/CBIO . . . . .	90
8.4.3 test CAIO . . . . .	90
8.4.3.1 test AD vstupů . . . . .	90
8.4.3.2 test DA výstupů . . . . .	90
8.4.4 test i/o na centrále CCPU-02. . . . .	90
8.5 F4 Kanal. . . . .	91
8.6 CCPU-21 základní funkce . . . . .	92
9 Příloha 2 - Zásuvné moduly pro analogové i/O . . . . .	94
10 Příloha 3 - Seznam objektů knihovny pro chybová hlášení . . . . .	96
11 Multi Point Communication . . . . .	98
12 Historická databanka HDB. . . . .	99

# 1 Knihovní moduly - obecně



Obr. 1 Okno KNIHOVNY MODULŮ

Knihovní moduly jsou rozděleny do několika skupin. Pro přehled uvedeme i zařazení modulů do jednotlivých skupin.

V levé části okna knihovny je nabídka skupin modulů, v jeho pravé části jsou pak zobrazeny moduly zvolené skupiny (viz obrázek 0).

Všechny knihovní moduly mají dva základní parametry **rychlost** a **priorita**.

Nezapojený vstup modulu má vždy hodnotu 0.

Skupiny KNIHOVNÍCH modulů:

- Hardware
- Komunikace
- Ostatní
- Regulace
- SW hradla
- Systémové moduly

## 1.1 Základní vlastnosti modulů

Všechny knihovní moduly mají dva základní parametry **rychlost** a **priorita**, které jsou přístupné v dialogovém okně modulu.

Toto se otevře po dvojkliku na modul, umístěný na ploše schéma.

### 1.1.1 Rychlost

Parametr **rychlost** určuje programovou smyčku, ve které bude modul vykonáván.

**Platné hodnoty: 1, 2 a 3.** Moduly s neplatnou hodnotou nebudou vykonávány.

- Moduly s parametrem **rychlost=1** jsou vykonávány s periodou **3000 ms** (3 s).
- Moduly s parametrem **rychlost=2** jsou vykonávány s periodou **1000 ms** (1 s).
- Moduly s parametrem **rychlost=3** jsou vykonávány s periodou **100 ms**.

V případě, že systém nestihne všechny moduly v příslušné smyčce vykonat během dané periody, bude perioda vykonávání prodloužena o dobu nezbytnou k vykonání všech

požadovaných modulů. V tomto případě už není možné určit přesně periodu vykonávání modulů.

Moduly, které mají vlastní menu, je vhodné spouštět s periodou 100ms pro zajištění dostatečně rychlé reakce na stisk klávesy. Pokud to není u některých modulů žádoucí (např. modul regulátoru PID, modul Bservo), je potřeba počítat s pomalou reakcí na stisk klávesy, zejména při vstupu a opuštění menu tohoto pomalého modulu.

### 1.1.2 Priorita

Parametr **priorita** určuje pořadí vykonávání modulů v rámci jedné programové smyčky (stejný parametr **rychlost**).

**Moduly s defaultní prioritou 0** jsou vykonávány v pořadí zleva doprava a shora dolů podle jejich grafické polohy ve schématu.

**Moduly s jinou prioritou** jsou vykonávány v pořadí, určeném hodnotou parametru **priorita**.

Tzn. modul s prioritou 1 bude vykonáván první, s prioritou 2 druhý atd. **Předpokladem je postupné zvyšování hodnoty parametru priorita o +1.**

Pokud by bylo postupné zvyšování hodnoty parametru **priorita** porušeno (některé její hodnoty by byly přeskočeny), jsou do uvolněných pozic při vykonávání aplikačního programu vřazeny moduly s nulovou prioritou (a to podle pravidel modulů s prioritou 0, tj. podle polohy modulů v grafickém schématu).

Pokud by byla zadána hodnota priority modulu tak velká, že při vykonávání modulů je modulů s nulovou prioritou na vřazení málo, podstatně je zpomalena celá časová smyčka a mohlo by dojít až k nevykonání činnosti takového modulu.

**Dva moduly** (se stejnou hodnotou parametru **rychlost**) **nesmí mít stejnou a nulovou prioritu.**

V tomto případě by byl ze dvou modulů se stejnou prioritou vykonáván pouze ten, který by byl v defaultním pořadí druhý.

*Poznámka:*

*Pokud využijete parametr **priorita** pro stanovení pořadí vykonávání modulů, neopomeňte, že tyto moduly musí být ve stejné programové smyčce, tj. musí mít stejný parametr **rychlost**. Tzn., že hodnotu parametru **priorita** zadáváte od 1 vždy po každou hodnotu parametru **rychlost**.*

*Jinými slovy - **prioritu definujete zvlášť (samostatně) pro každou rychlost.***

### 1.1.3 Rychlost komunikace

Pokud použijete defaultní object dictionary v centrální jednotce, určený pro systém konfigurovatelný FREDem, jsou zvoleny rychlosti přenosu PDO tak, aby ani při plném osazení periferními jednotkami na sběrnici CAN nedošlo k přetížení sběrnice.

Znamená to, že informace z **binárních vstupů a výstupů jsou komunikovány s periodou 25 ms** (pro adresy 1..4) nebo 50 ms (pro ostatní adresy).

Informace **do analogových výstupů** jsou komunikovány stejně.

Informace z **analogových vstupů 0 až 3** jednotky CAIO-11/12 jsou komunikovány také tak. Proto je vhodné je používat přednostně pro regulační smyčky s velkými nároky na rychlost.

Informace z **ostatních analogových vstupů** je přenášena s **periodou 350 milisekund**. Tyto vstupy je tedy vhodné použít pro regulační smyčky s menšími nároky na rychlost.

### 1.1.4 Vícenásobné použití modulu

**Běžné moduly** jednoho typu lze použít v projektu maximálně 255x.

Počty **HW modulů** odpovídají v projektu praktickým počtům ve skutečnosti:

- jedna centrála pro jeden projekt

- jeden ovládací panel do projektu na sběrnici CAN (CKDM11/12), případně druhý na sériovou linku (SKDM11/12)
- periferní jednotky lze použít tolikrát, dokud stačí jejich adresování na CANu (adresy 1 - 31)

Některé **systémové moduly** (sysmon, flag, timdat, hesla, flash, canopen, journal) pouze 1x.

**Historickou databanku** musíme zvolit jednu ze dvou možných typů, buď VZORKY nebo HDB.

Modul VZORKY lze v jednom projektu použít 4x, ale jen v tom případě, že nepoužijeme banku typu HDB.

Modul HDBmain v případě nepoužití modulů VZORKY může být použit pouze jednou.

Moduly HDBa a HDBb (moduly pro přístup do HDB) pak až 255x.

Některé **komunikační moduly** jen 1x: ReTran, modem, EmergCall, pwpb\_main, dcf, gsm, Inet, IEC870 a chystaný modul mail.

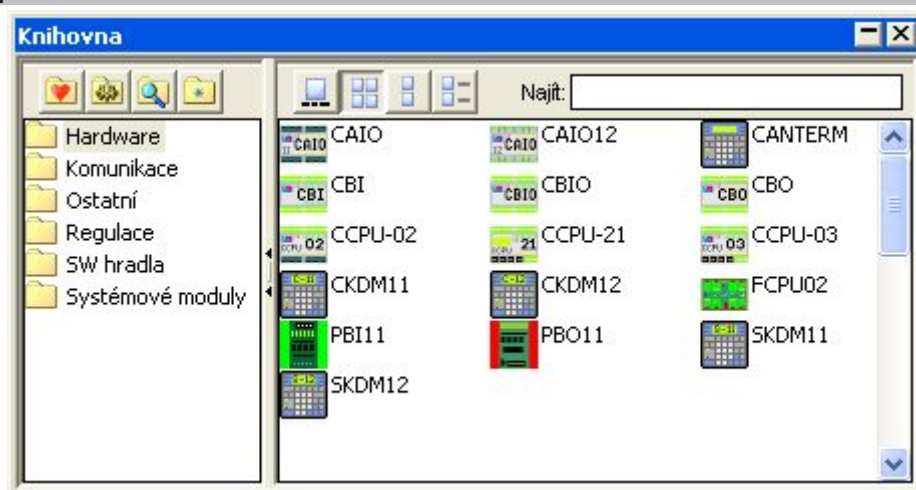
Dále se vylučují moduly na stejné komunikační lince s různými protokoly.

Na stejné komunikační lince lze kombinovat pouze moduly, které mají kompatibilní přístup na COM, tj. moduly typu pwpb...

A dále lze mezi sebou míchat sercom, všechny měřiče tepla, modbustx/rx/txc, M-Bus. Do této skupiny patří i moduly Sxxx, ale míchat na jedné lince různé protokoly není nejlepší nápad. Nedělá se to ...



## 2 Skupina modulů Hardware



Obr. 2 Okno knihovny modulů - aktivní skupina **Hardware**

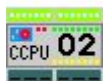
Navolíme-li v levé části okna knihovny modulů **Hardware**, objeví se v jeho pravé části malé ikony jednotlivých HW modulů regulátoru **PROMOS line 2**. Uchopíme-li malou ikonu zvoleného HW modulu myší a tuto přetáhneme a pustíme na plochu listu, změní se ikona modulu na větší. Větší ikona modulu již obsahuje čitelné vstupy a výstupy modulu.

Malé ikony všech modulů jsou stejně velké. Proto se značka / ikona modulu liší na ploše a v knihovně.

Moduly jsou ve skupině řazeny abecedně a odpovídají HW systému PL2:

- **CAIO**  
modul pro analogové I/O  
odpovídá jednotce CAIO-11, která je staršího provedení, proto v nových aplikacích používejte nový typ CAIO-12  
čte až 12 měřených analogových vstupů  
ovládá až 6 analogových výstupů
- **CAIO12**  
modul pro analogové I/O  
čte až 12 měřených analogových vstupů  
ovládá až 6 analogových výstupů
- **CANTERM**  
modul pro zastaralý ovládací panel - nepoužívejte aktualizací FW staršího panelu CKDM-11 lze používat ve FREDovi pak modul CKDM11
- **CBI**  
modul odpovídající jednotkám CBI-11/12  
čte 16 binárních vstupů
- **CBIO**  
modul odpovídající jednotkám CBIO-11/12  
čte 8 binárních vstupů  
ovládá 8 relé
- **CBO**  
modul odpovídající jednotkám CBO-11/12  
ovládá 12 relé
- **CCPU-02**  
centrála s binárními i analogovými I/O
- **CCPU-21**  
kompaktní centrála s I/O i ovládacím panelem
- **CCPU-03**  
centrála s binárními I/O, USB, ethernet
- **CKDM11**  
modul pro ovládací panel CKDM-11
- **CKDM-12**  
modul pro ovládací panel CKDM-12  
proti verzi 11 má navíc panelové I/O
- **FCPU02**  
modul pro jednotku FCPU-02A pro 64 logických linek, volitelných po osmicích moduly PBI a PBO
- **PBI11**  
modul k FCPU02 pro osm binárních vstupů
- **PBO11**  
modul k FCPU02 pro osm relé
- **SKDM11**  
modul pro sériový ovládací panel SKDM-11
- **SKDM12**  
proti SKDM12 obsahuje navíc panelové I/O

## 2.1 CCPU-02



Modul **CCPU-02** umožňuje práci s HW mikro- počítače CCPU-02 včetně jeho i/o.

CCPU-02 je kompaktní mikropočítačová jednotka systému **PROMOS line 2**. Variabilita vstupů / výstupů dovoluje sestavit konfiguraci dle potřeby a maximálně tak využít všech možností jednotky.



Obr. 3 Značka **CCPU-02** na ploše

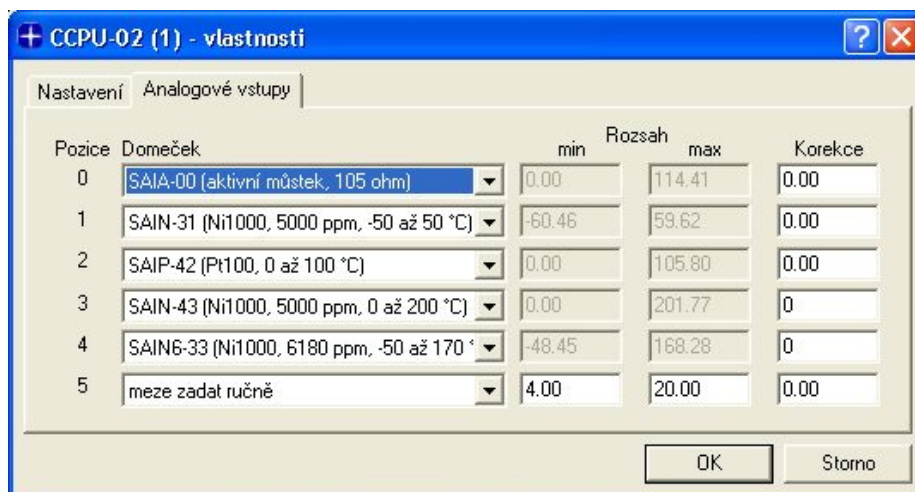
Jednotku tvoří procesorový modul **MCPU-01** a základní deska vstupů/výstupů.

Základní deska i/o má standardně osazeno:

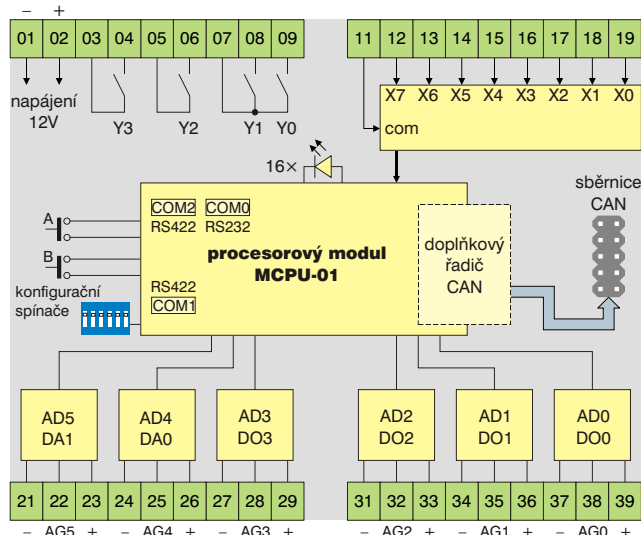
- **4 reléové výstupy** (jazýčkový kontakt) piny sekce **Binary** s označením **Out4..7** jsou určeny pro tyto 4 standardní logické výstupy, piny **Out0..3** jsou určeny **pouze** pro ovládání indikačních LED na panelu CCPU-02 s popisem Y0..3 (v blokovém schéma odpovídá Out0 označení Y0, ... Out3 pak Y3)
- **8 logických vstupů** 24 V s galvanickým oddělením piny sekce **Binary** s označením **In0..7** parametrem **Counter\_switch** v dialogu modulu lze nastavit čítecí funkce každého **In0..7** zvlášť (v blokovém schéma odpovídá In0 označení X0, ... In7 pak X7)

Pak obsahuje **6 dalších univerzálních pozic** (v blokovém schéma značeny **poz. 0..5**), které lze osadit výměnnými moduly:

- **analogového vstupu** (všech 6 pozic) piny **In0..5** sekce **Analog** (v blokovém schéma pozice **poz. 0..5**)
- **analogového výstupu** (0-10 V, 8bit DA převod, max. 2 pozice) piny **Out0..1** sekce **Analog** (v blokovém schéma pozice **poz. 4..5**)



Obr. 5 Dialogové okno modulu **CCPU-02** pro definici mezí analogových vstupů



Obr. 4 Blokové schéma CCPU-02

Výměnný modul pro definici univerzální pozice obsahuje operační zesilovač s odporovou sítí a podle typu umožňuje měření napětí, proudu, odporu nebo přímé připojení odporového čidla Pt100, Ni1000. Typ signálu a rozsah měření je dán typem výměnného piggy modulu (přehled je uveden v poslední kapitole tohoto manuálu). A/D převodník je desetibitový. Pro vlastní A/D převod je využit vnitřní převodník procesoru, který poskytuje rychlou odezvu a umožňuje tak číslicovou filtraci a měření rychle se měnících signálů.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **load** nepoužívejte (ponechejte = 0), měl mít stejný význam jako u modulu CAIO.

**Meze lze zadat pouze v projektu.**

V dialogovém okně modulu CCPU-02 jsou na kartě **Analogové vstupy** přístupné další parametry pro meze (měřený rozsah) jednotlivých analogových vstupů.

Každá univerzální pozice má v tomto okně svůj řádek (0 až 5).

Pro standardní výměnné odporové sítě nemusíte parametry **Hmez** (popis v okně **rozsah max**) a **Dmez** (popis v okně **rozsah min**) zadávat - zvolíte z rozbalovací "roletky" jen označení / typ sítě a hodnoty jsou doplněny automaticky.

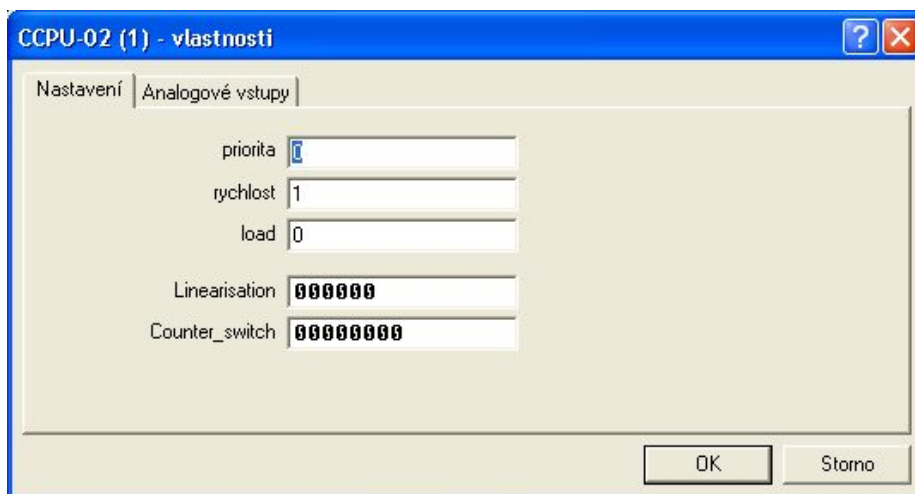
Po takové volbě je pro zvolené čidlo provedena i **linearizace průběhu**.

Pokud zvolíme defaultní "meze zadat ručně" lze meze do políčka dialogového okna zadat ručně, linearizace se neprovádí.

Parametr **korekce** je pak pro každý vstup dodatečná korekce, která se přičte k výstupní hodnotě.

Parametr **counter\_switch** je přístupný v dialogovém okně modulu **CCPU-02** na kartě **Nastavení** a slouží pro volbu funkce vstupů **In0..7** sekce **Binary**.

Lze volit mezi klasickou funkcí logického vstupu nebo určit, že vstup je čítecí. Takto lze definovat každý vstup samostatně.



Obr. 6 Karta Nastavení dialogu pro CCPU-02

Proto parametr **counter\_switch** obsahuje bitovou hodnotu v rozsahu 00000000 až 11111111.

Každá cifra parametru může nabývat pouze hodnot 0/1.

Hodnota 10000000 určuje, že vstup **In7** sekce **Binary** bude fungovat jako čítací, tj. s vazbou na modul **cnt**.

Hodnota 11111111 určuje, že všechny vstupy **In0..7** sekce **Binary** se budou chovat jako čítací.

Pokud se vstup chová jako čítací, předává na svém pinu dvoubajtovou hodnotu vnitřního čítače. Jejím nárůstem umí zpracovat modul **cnt**.

Perioda testování čítacích vstupů je 10 ms, což odpovídá maximální frekvenci cca 50 Hz, kterou lze ještě vstupem teoreticky zpracovat.

Po zapnutí jednotky CCPU-02 je vnitřní hodnota čítačů nulována.

*Poznámka:*

*Na čelním panelu modulu CCPU-02 jsou přístupné přepínače DIL (popsány dále).*

*Tlačítka s popisem A a B nejsou programově využita.*

*Tlačítko RESET s popisem IR je schováno nad tlačítkem A.*

*Je přístupné otvorem nad tlačítkem A a lze stlačit např. pomocí tužky.*

**Adresu modulu CCPU-02 pro komunikaci lze nastavit jednak na DIL přepínačích jednotky, jednak pomocí SW.**

**Přednost má nastavení na DIL přepínačích a je platné až po RESETu (restartu) centrály !**

SW adresa platí, pokud je na DIL přepínačích nastavena adresa 0. Zadávat se jako parametr **address** modulu **system**.

DIL přepínače jsou zleva popsány přímo na přepínačích znaky **1 2 3 4 5 6**.

První dva jsou popsány i na štítku znaky **T E**.

Přepínačem 1 s popisem **T** (terminátor) zařazujeme zakončení sběrnice CAN.

Přepínačem 2 s popisem **E** povolujeme (enable) používání logických výstupů (4 kontakty jazýčkových relé) na modulu CCPU-02.

Přepínače **3 4 5 6** jsou určeny pro nastavení adresy modulu CCPU-02.

V poloze ON přepínačů **3 4 5 6** tak postupně zařazujeme váhy adresy 8 4 2 1.

*Poznámka:*

*Komunikační rychlost kanálu COM0 u PL2 (určeného pro napojení PC s FREDem nebo připojení modemu) je implicitně 38400 Bd.*

*Zatím ji lze měnit pouze použitím (i fiktivním) modulu **modem** a změnou jeho parametru **commspeed**.*

Nová komunikační rychlost je platná po překladu - pozor proto na změnu komunikačních parametrů ProgWiinu.

## Komunikační kanály centrály

Centrála má tři sériové kanály - COM0, COM1 a COM2.

Pokud některý z modulů určených pro komunikaci lze napojit na některý z nich, má tento modul parametr, kterým toto navolíme. Tak je zvolen potřebný komunikační protokol a pokud komunikační modul nemá další komunikační parametry, pak jsou navoleny i ty. Jinak se zadávají jako parametry komunikačního modulu.

Některé komunikační moduly volbu COM0/1/2 nemají, protože je u nich předepsáno povinné napojení na jeden z nich. Přesto mohou mít tyto moduly jako parametry některé z komunikačních parametrů (přenosovou rychlost apod.). Pak jsou jimi nadefinovány. Komunikační protokol je zvolen použitím komunikačního modulu v projektu.

Defaultně je COM0 (hlavní kanál) určen pro tzv. QQ relace, tj. pro napojení k PC s FREDem, pro napojení na dispečink s PC s příslušným SW pro QQ relace, pro napojení k PC s loaderem pro UpGrade firmware centrály.

Centrála CCPU-02 umí na hlavním kanále i zjednodušený protokol **modbus** - bez přepínání. Modbus umožňuje přístup do modulů COMPORT a SCREEN, a to tak, že **Register address Hi** (terminus technikus protokolu) určuje instanci daného modulu, a **Register address Lo** je pro výstupy COMPORT 0..31, vstupy modulu COMPORT 32..63. Pro přístup do modulu COMPORT musí být tato položka násobek 2. **Register address Lo** 64, 66, 68 a 70 jsou vyhrazeny pro přístup do modulu SCREEN, a to na výstupy **edit0..edit3**. Je možné vždy zapisovat i čísl, délka dat musí být násobek 4 (tj. No. of Points musí být násobek 2).

## 2.2 CANTERM

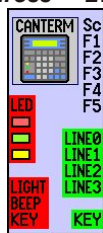


Modul **CANTERM** umožňuje práci s terminálem - ovládacím panelem **CKDM-11** (se starším firmware CANTERM.mhx) na sběrnici CAN. Pro nový firmware v jednotce CKDM-11/12 je ve FREDovi modul **CKDM11**, resp. **CKDM12**. V nových projektech modul **CANTERM nepoužívejte** - ve starých aplikacích proveďte UpG FW z **CANTERMu** na **CKDM11**!

Ovládací panel má čtyřřádkový displej 4 x 20 znaků, 27 kláves a 5 LED. **Pokud je napájen, bliká zelená LED s popisem Run.**

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **address** určuje sběrniceovou adresu modulu. Pro správnou funkci terminálu, zejména v módu regulátoru system je třeba, aby **address = 2**.



Obr. 8 Značka **CANTERM** na ploše

Parametr **access** určuje úroveň přístupu do systémového módu.

- **access=0**  
umožňuje přechod do módu system (regulátoru) kdykoliv, popis ovládání v tomto módu viz kapitola 4 **Základní funkce regulátoru PL2**
- **access=1..4**  
určuje, že pro přechod do módu system musí být přihlášen uživatel alespoň úrovně rovné hodnotě parametru **access**.

Parametr **noterm** běžně = 0.

Je-li =1, neprobíhá CAN komunikace s CKDM-11.

Současně není hlášena chyba CAN v případě, že není terminál vůbec připojen (není v sestavě).

Vstupy modulu **CANTERM**, označené symboly **LED** či popisem **Beep**, **Key** a **Light** ovládají napovídání funkce u HW:

- **LED** - svit jednotlivých LED
- **Beep** -zvukovou signalizaci
- **Key** - simulaci stisku klávesy
- **Light** - ovládání podsvitu displeje

Výstup **sc** je určen pro připojení modulu **SCREEN** pro tvorbu stromové struktury uživatelského menu. První modul **SCREEN** tvoří tzv. hlavní menu.

Výstupy **F1..5** jsou určeny opět pro připojení modulů **SCREEN** (jako výše) pro tzv. rychlý přístup do často používaných menu. Do těchto menu se dostáváme stiskem klávesy F1, F2, F3, F4 nebo F5 na ovládacím panelu CKDM-11.

Výstupy **Line0..3** zobrazují aktuální stav jednotlivých řádků displeje terminálu, bez ohledu na to, je-li terminál fyzicky připojen (hardware) či nikoli.

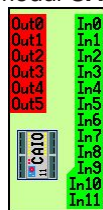
Stačí na ně napojit labely a v režimu Ladění (FREDA) lze aktuální stav řádků v nich jen číst.

Výstup **Key** vrací kód stisknuté klávesy na terminálu.

## 2.3 CAIO



Modul **CAIO** umožňuje práci s hardwarovou jednotkou CAIO-11. V nových aplikacích používejte periferní jednotku CAIO-12 a ve FREDovi pak modul **CAIO-12**.



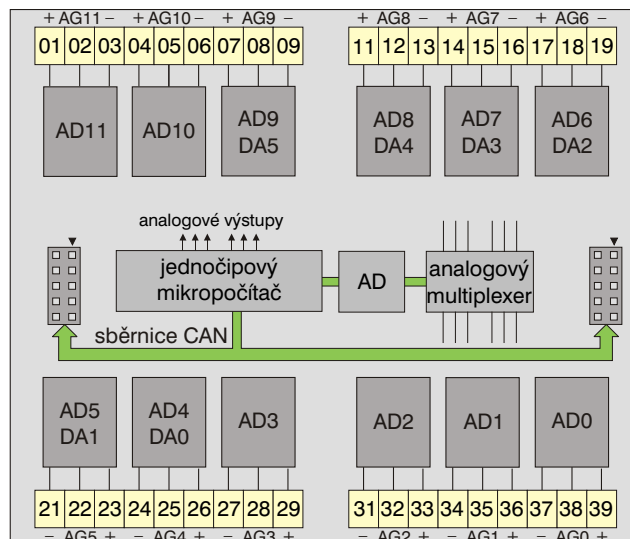
Obr. 7 Značka **CAIO** na ploše

CAIO-11 je periferní modul na sběrnici CAN s 12 univerzálními pozicemi, SW je zpracován pro analogové vstupy/výstupy.

Na **univerzální** pozice základní desky se podle potřeby osazují analogové i/o moduly. Typ signálu a rozsah měření je dán typem výměnného piggy modulu (přehled je uveden v poslední kapitole tohoto manuálu).

Moduly jsou výměnné bez rozebrání jednotky.

**Analogové vstupní moduly** obsahují operační zesilovač



Obr. 9 Blokové schéma CAIO-11

s odporovou sítí a podle modifikace umožňují měření napětí, proudu, odporu nebo přímé připojení odporových čidel Pt100, Ni1000. Typy signálů a rozsahy měření jsou shodné s analogovými vstupy jednotek PAI-01 a analogovými vstupy SBPS-02 pouze s tím rozdílem, že vlastní AD převodník je **čtrnáctibitový**. Protože obsahují i operační zesilovač, nemusi být nepoužité vstupy ošetřovány.

**Moduly analogových výstupů** obsahují filtr a výstupní zesilovač.

**Na všech pozicích (0-11)** mohou být piggy pro analogové vstupy.

Hodnoty z analogových vstupů jednotky CAIO-11 (vstup 0-11) jsou po filtraci, normalizaci a korekci přeneseny na výstupy **In0..In11** modulu **CAIO**.

**Na pozicích 4-9** mohou být piggy pro analogové výstupy. D/A výstupů může být tedy max. 6.

Hodnoty ze vstupů **Out0..Out5** modulu **CAIO** ovládají analogové výstupy jednotky CAIO-01.

Standardní parametry modulu: **rychlost a priorita**.

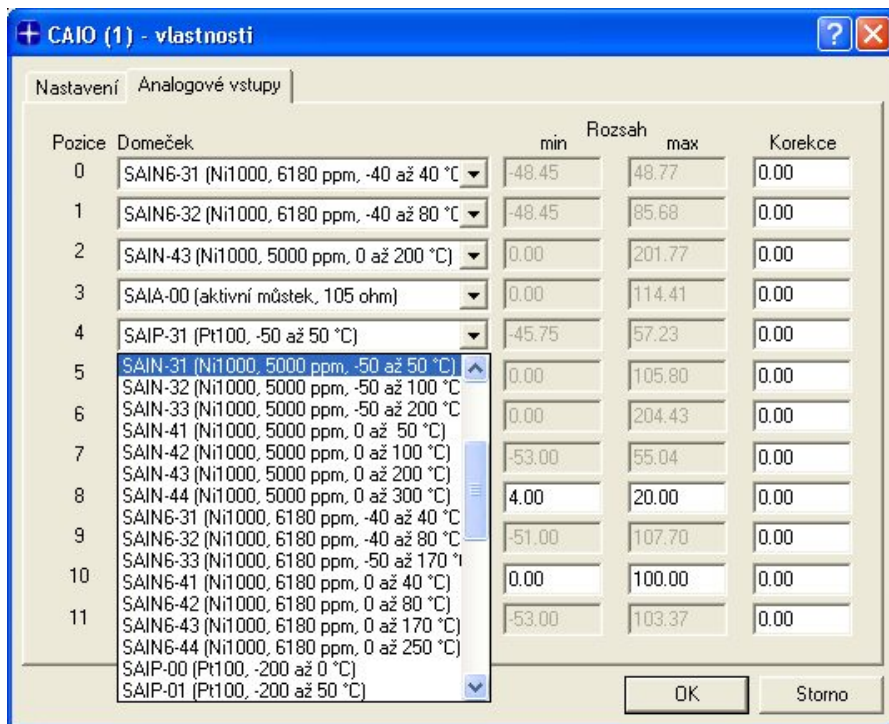
Parametr **address** určuje adresu jednotky na sběrnici CAN. Platné hodnoty jsou 1, 3 až 31.

Parametr **filter** určuje časovou konstantu filtrace analogových vstupů (v ms). Je však ignorován pokud je menší než 60.

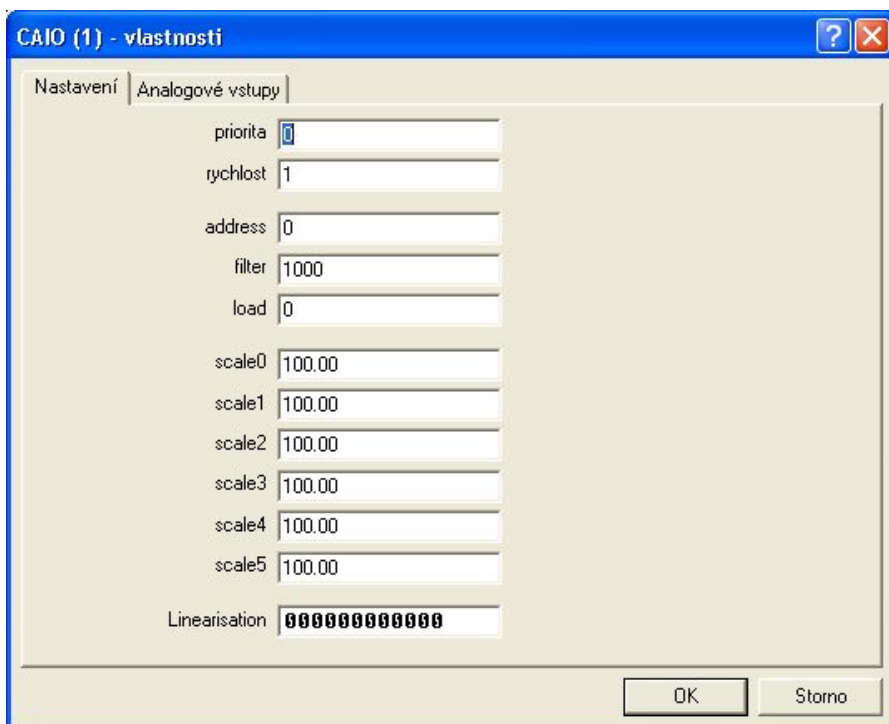
Parametr **load**, pokud=1 způsobí po PŘEKLAdu (a po přechodu regulátoru do jeho režimu RUN - nezaměňujte s ladícím režimem FREDA) přesun hodnot parametrů **Hmez**, **Dmez** a **korekce**, které jsme nastavili v režimu regulátoru **system** (volba **test CAIO**), z části paměti regulátoru pro zadávání do části paměti regulátoru pro aplikační program.

Tyto hodnoty je pak možné z regulátoru načíst do projektu ve FREDovi volbou **načtení konstant modulu**.





Obr. 10 Dialogové okno modulu CAIO, karta Analogové vstupy



Obr. 11 Dialogové okno modulu CAIO, karta Nastavení

Pokud je  $load=0$ , systém použije hodnoty parametrů z FREDa.

Funkce vstupů a výstupů závisí na osazení jednotky zásuvnými moduly. Platná data budou pouze na těch výstupech modulu, které budou mít osazené správné zásuvné moduly.

Parametry **scale** jsou měřítka pro jednotlivé analogové výstupy. Pokud  $scale=100$ , pracuje výstup v rozsahu 0 až 100 %.

V dialogovém okně modulu CAIO na kartě *Analogové vstupy* má každá univerzální pozice v tomto okně svůj řádek (0 až 11).

Pro standardní výměnné odporové sítě nemusíte parametry **Hmez** (popis v okně **Rozsah max**) a **Dmez** (popis v okně **Rozsah min**) zadávat - zvolíte z rozbalovací "roletky" jen označení / typ sítě a hodnoty jsou doplněny automaticky.

Po takové volbě je pro zvolené čidlo provedena i **linearizace průběhu**.

Pokud zvolíme defaultní "zadat meze ručně" lze meze do políčka dialogového okna zadat ručně, linearizace se neprovádí.

Parametr **korekce** je pak pro každý vstup dodatečná korekce, která se přičte k výstupní hodnotě.

**Poznámky:**

Popsané platí od verze 2 firmware v CAIO, pokud je v CAIO starší firmware, volte "bez domečku" a meze zadejte ručně (dle katalogu nebo tabulek pro Pt100 či Ni1000).

Výměnný modul pro analogové vstupy obsahuje propojku, kterou je třeba nastavit podle použití modulu.

Jednotka CAIO-11 má 12 univerzálních pozic pro zásuvné moduly, některé z nich mohou pracovat buďto jako vstupy, nebo jako výstupy.

SW je zpracován tak, že všech 12 pozic může být osazeno pro analogové vstupy a pozice 4-9 mohou být osazeny pro analogové výstupy.

Ve FREDovi pak zapojujeme piny modulu CAIO podle osazení univerzálních pozic, a to od varianty 12 ADin + 0 ADout až po variantu 6 ADin + 6 ADout.

**Příklad:**

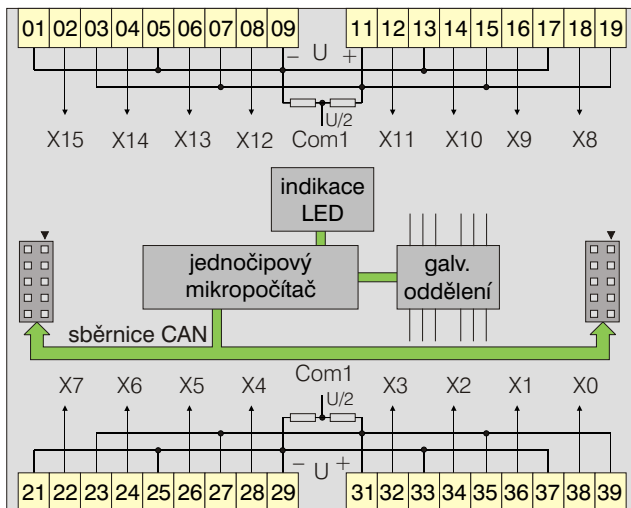
Pokud osadíme univerzální pozici 9 pro analogový výstup, zapojíme ve schématu pin Out5 a nesmíme zapojit pin In9!

Pozice se počítají od 0.

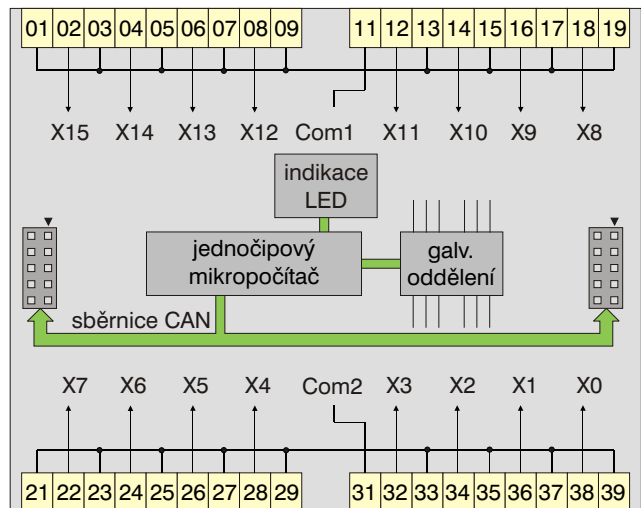
**Vysvětlení k parametru load:**

Parametry mezi a korekcí analogových vstupů, které zadáváme do regulátoru pomocí klávesnice ovládacího panelu CKDM-11, jsou uloženy v jiné oblasti paměti centrály CCPU-02, než parametry Hmez, Dmez a korekce, které jsou používány aplikačním programem.

Pokud tyto parametry zadáme přes klávesnici regulátoru a jsou jednoznačně správné, volíme v projektu ve FREDovi u příslušného modulu CAIO-11 parametr  $load=1$ .



Obr. 13 Blokové schéma CBI-11



Obr. 12 Blokové schéma CBI-12

To způsobí, že nejsou přeneseny po PŘEKLAdu parametry mezi a korekcí modulu CAIO s parametrem  $load=1$  do patřičné oblasti paměti regulátoru, vyhrazené pro aplikační program (tj. projekt ve FREDovi), ale jsou přeneseny parametry mezi a korekcí z oblasti paměti regulátoru, kde jste je zadali ručně přes klávesnici regulátoru, do příslušné části paměti, vyhrazené pro aplikaci (projekt).

Po této akci doporučujeme z FREDa načíst konstanty modulu (modulů) CAIO s parametrem  $load=1$  volbou načtení konstant modulu

(rychlá volba = vyber modul a stiskni CTRL+R nebo klikni na modulu CAIO pravým tlačítkem na myši).

Potom raději změňte parametr  $load$  na 0 u všech modulů CAIO a proveďte PŘEKLAdu znovu.

Rovněž nezapomeňte uložit projekt s načtenými konstantami na HDD či disketu.

## Perioda komunikace i/o modulů na sběrnici CAN

Aplikační program v centrále (vytvořen ve FREDovi) běží ve třech programových smyčkách. Periody vykonávání naprogramovaných modulů v projektu (podle parametru rychlost v každém modulu) jsou 3000 ms, 1000 ms a 100 ms.

Kromě toho centrála s jednotlivými periferními moduly (i/o moduly) komunikuje po sběrnici CAN, aby s nich vyčetla nebo jim předala požadované/vypočtené informace.

Informace z/do binárních (logických) vstupů a výstupů jsou komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy).

Informace do analogových výstupů jsou rovněž komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy).

Informace z analogových vstupů 0..3 jednotky CAIO-11 jsou komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy), ale z ostatních vstupů 4..11 jsou komunikovány s periodou 350 ms.

Proto je vhodné používat vstupy 0..3 modulu CAIO přednostně pro regulační smyčky s velkými nároky na rychlost.

## 2.4 CBI



Modul CBI umožňuje práci s hardwarovou jednotkou CBI-11/12, periferní jednotkou na sběrnici CAN se 16 logickými vstupy s GO. Jednotky se liší



Obr. 14 Značka CBI na ploše

vnitřním zapojením vstupních obvodů, SW ovládání je totožné.

Logické úrovně ze vstupů jednotky CBI-11/12 jsou v jednotce filtrovány a předány na výstupy **In0..15** modulu CBI.

Hodnoty měření periody impulsů ze vstupů **4..7** jsou na výstupech **Per0 až Per3** modulu (v ms).

Hodnoty stavu čítačů na vstupech **0..3** jednotky jsou na výstupech **Cnt0 až Cnt3** modulu. Po znovuzapnutí regulátoru jsou hodnoty na těchto výstupech nulové. V projektu na tento výstup obvykle zapojujeme hradlo **cnt** (akumulátor hodnoty čítače).

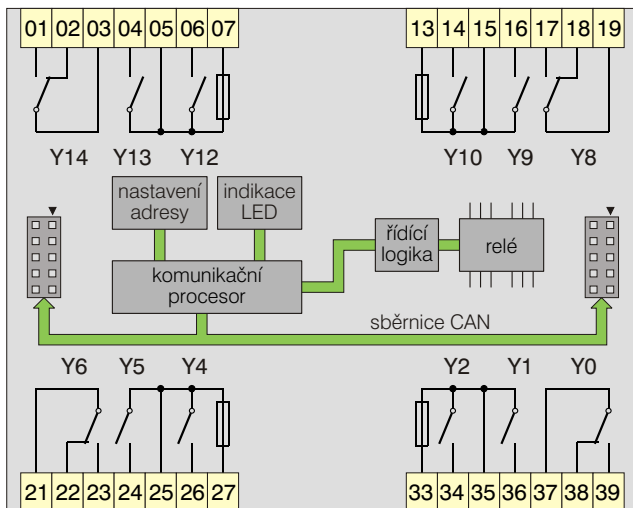
Standardní parametry modulu: **rychlost a priorita**.

Parametr **address** určuje adresu jednotky na sběrnici CAN. Platné hodnoty jsou 1, 3 až 31.

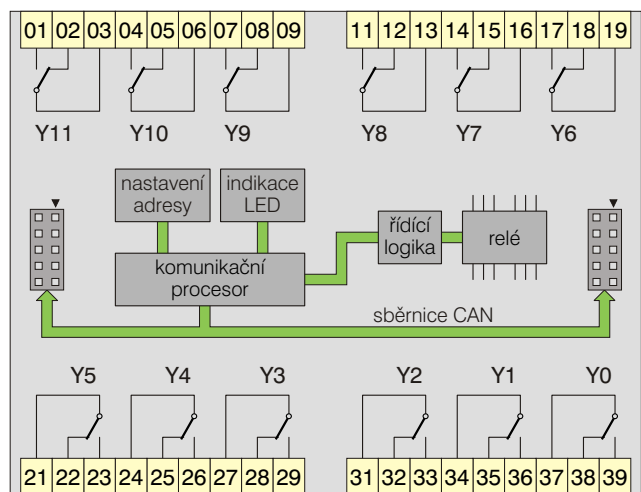
Parametr **filtr** určuje časové konstanty ( $timeh=timel$ ) filtrace binárních vstupů jednotky v milisekundách. Zadává se v rozmezí 0 - 256.

Při hodnotě **filtr = 0** je filtr vyřazen a maximální frekvence čítání se tak zvýší na 500 Hz.

Parametr **CANper** (určoval periodu komunikace pro čítačové vstupy) je ve firmware od verze 2 ignorován. Tento SW jednotky však umožňuje pomocí modulu **sdo** a speciální komunikace vyčítat z jednotek CBI-11/12, a to dokonce ze všech vstupů jednotky, hodnotu načítaných pulzů, změřit jejich periodu i frekvenci.



Obr. 16 Blokové schéma CBO-11



Obr. 15 Blokové schéma CBO-12

## 2.5 CBO

Modul **CBO** umožňuje práci s hardwarovou jednotkou CBO-11/12.



Obr. 19 Značka **CBO** v knihovně a na ploše

CBO-11 je výstupní jednotka na sběrnici CAN. Obsahuje 12 výstupních obvodů s galvanickým oddělením. Spínacím prvkem je relé se síťovým kontaktem 250 V AC, který umožňuje přímé spínání síťových spotřebičů. Kontakty jsou uspořádány do čtyř skupin tak, aby umožňovaly spínání jednofázových spotřebičů (stykače, solenoidové ventily) i obousměrné servopohony. V každé skupině je jedna tavná pojistka.

Jednotky CBO-11/12 se liší zapojením výstupních přepínacích kontaktů, CBO-12 má vyvedeny kontakty samostatně. SW ovládání je totožné.

Logické úrovně ze vstupů modulu **CBO** ovládají relé jednotky CBO-11/12.

Standardní parametry modulu: **rychlost a prioritita**.

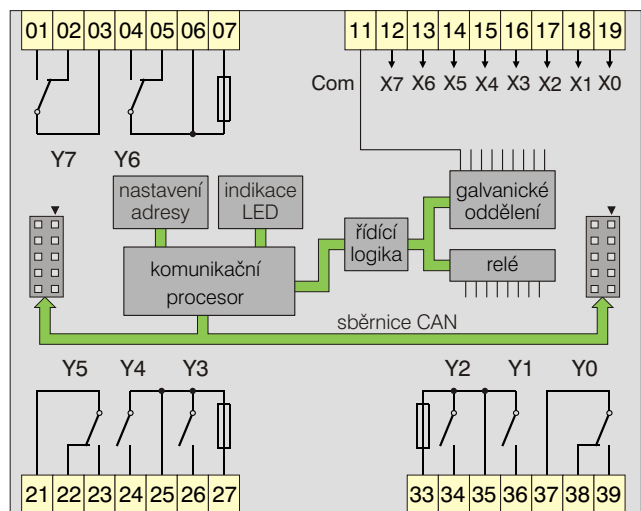
Parametr **address** určuje adresu jednotky na sběrnici CAN. Platné hodnoty jsou 1, 3 až 31.

Po stisku tlačítka DIALOG MODULU v dialogovém okně modulu **CBO** je přístupný parametr **Switch\_off**. Tento řetězec 12ti stavů 0/1 je určen pro definici stavu relé při ztrátě komunikace modulu s centrálou.

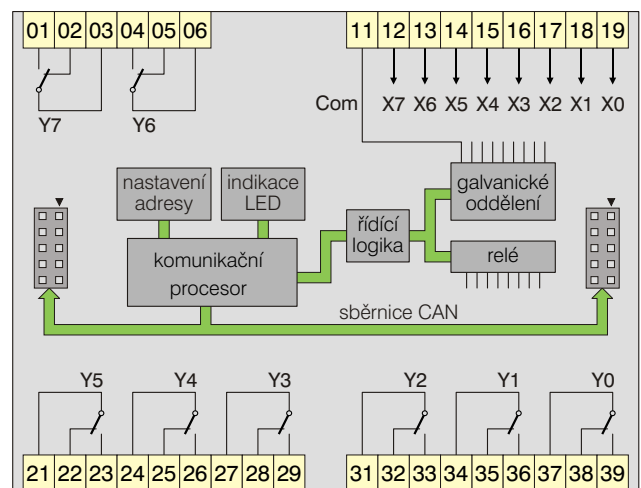
**POZOR!** Od verze firmware 2 jednotky CBO-11/12 je tento parametr nefunkční! Stejně funkce lze docílit pomocí speciální komunikace modulem **sdo**.

*Poznámka:*

*Při zapojování modulu CBO ve schématu zapojte jen označené piny o1 až o12. Nepopsané piny modulu ovládají pouze signalizační LED modulu!*



Obr. 17 Blokové schéma CBIO-11



Obr. 18 Blokové schéma CBIO-12

## 2.6 CBIO

Modul **CBIO** umožňuje práci s hardwarovou jednotkou CBIO-11 nebo CBIO-12.

CBIO-11/12 jsou kombinované periferní jednotky logických vstupů/výstupů na sběrnici CAN s 8



galvanicky oddělenými logickými vstupy a s 8 logickými výstupy, které jsou realizovány jako kontakty relé.

Modul CBIO-12 má vyvedeny výstupní kontakty samostatně. Jednotky CBIO-11/12 se liší zapojením výstupních přepínacích kontaktů, SW ovládání je totožné.



Obr. 22 Značka **CBIO** na ploše

### Logické vstupy

Logické úrovně ze vstupů jednotky CBIO-11/12 jsou v jednotce filtrovány a předány na výstupy **In0-7** modulu **CBIO**.

Hodnoty měření periody impulsů ze vstupů 4-7 jsou na výstupech **Per0** až **Per3** modulu (v ms).

Hodnoty stavu čítačů na vstupech 0-3 jednotky jsou na výstupech **Cnt0** až **Cnt3** modulu. Po znovuzapnutí regulátoru jsou hodnoty na těchto výstupech nulové. V projektu na tento výstup obvykle zapojujeme hradlo **cnt** (akumulátor hodnoty čítače).

### Logické výstupy:

Modul obsahuje 8 výstupních obvodů s galvanickým oddělením. Spínacím prvem je relé se síťovým kontaktem 250V AC, který umožňuje přímé spínání síťových spotřebičů.

Kontakty modulu CBIO-11 jsou uspořádány do dvou skupin tak, aby umožňovaly spínání jednofázových spotřebičů (stykače, solenoidové ventily) i obousměrné servopohony. V každé skupině je jedna tavná pojistka.

Jednotky CBIO-11/12 se liší zapojením výstupních přepínacích kontaktů, CBIO-12 má vyvedeny kontakty samostatně. SW ovládání je totožné.

Logické úrovně ze vstupů **Out0..7** modulu **CBIO** ovládají relé jednotky CBIO-11/12.

### Parametry modulu

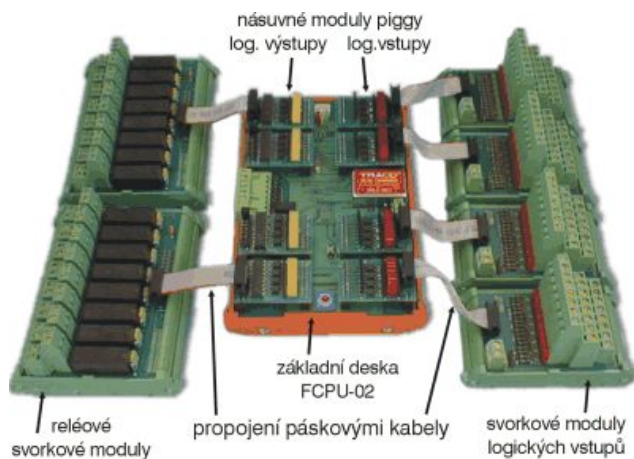
Standardní parametry modulu: **rychlost a priorita**.

Parametr **address** určuje adresu jednotky na sběrnici CAN. Platné hodnoty jsou 1, 3 až 31.

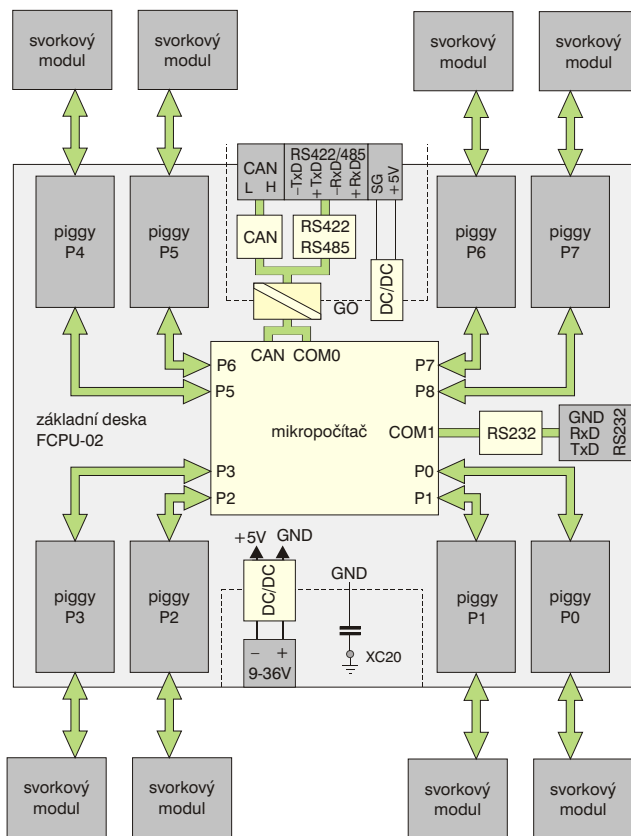
Parametr **filtr** určuje časové konstanty (timeh=timel) filtrace binárních vstupů jednotky v milisekundách. Zadává se v rozmezí 0 - 256. Při hodnotě **filtr = 0** je skutečně filtr vyřazen. Tím je zvýšena maximální frekvence čítání na 500 Hz.

Parametr **CANper** (určoval periodu komunikace pro čítačové vstupy) je ve firmware od verze 2 ignorován. Tento SW jednotky však umožňuje pomocí modulu **sdo** a speciální komunikace vyčítat z jednotek CBI-11/12, a to dokonce ze všech vstupů jednotky, hodnotu načítaných pulzů, změřit jejich periodu i frekvenci.

Po stisku tlačítka DIALOG MODULU v dialogovém okně modulu **CBIO** je přístupný parametr **Switch\_off**. Tento řetězec 8 stavů 0/1 byl určen pro definici stavu relé při ztrátě komunikace modulu s centrálou. Od verze firmware 2 je ne-



Obr. 20 Zapojení FCPU-02A s piggy PBI-11 (vpravo), PBO-11 (vlevo) a plochými vodiči na svorkové moduly.



Obr. 21 Blokové schéma modulu FCPU-02A

funkční. Stejně funkce lze docílit pomocí speciální komunikace modulem **sdo**.

## 2.7 FCPU02



Modul **FCPU02** umožňuje práci s hardwarem modulem FCPU-02A, připojený pomocí sběrnice CAN.

Mikropočítač FCPU-02A je stavebnicový modul, určený především pro úlohy sběru dat a logické řízení. Základní deska FCPU-02A (ve FREDovi modul **FCPU02**) obsahuje jednočipový mikropočítač, obvody sériových rozhraní, napájecí měniče a 8 pozic pro násuvné I/O moduly. Na každé pozici je vyvedeno 8 I/O bitů mikropočítače. Osažením příslušného I/O piggy modulu je možné odpovídající



bránu mikropočítače konfigurovat jako vstupní (piggy PBI-11, ve FREDovi modul **PBI11**) nebo výstupní (piggy PBO-11, ve FREDovi modul **PBO11**). Celá jednotka FCPU-02A tak může distribuovat až 64 logických I/O signálů. Viz blokové schéma modulu FCPU-02A.



Obr. 25 Značka **FCPU02** na ploše

Ve FREDovi musíte dle projektu zvolit počet a typ násuvných piggy (**PBI11**, **PBO11**) a napojit je na modul **FCPU02**.

Na fotce zapojení HW vidíte, že na základní desce FCPU-02A je třeba umístit pro jednotlivé osmice i/o násuvné piggy (PBI-11 pro osmici vstupů, PBO-11 pro osmici výstupů). K nim páskovým vodičem napojíte HW svorkové moduly XBI-11 pro logické vstupy či XBO-11 s osmici výstupních relé. Na svorkové moduly (ty ve FREDovi nejsou třeba) pak napojujete signály z/do technologie.

Standardní parametry modulu: **rychlost** a **priorita**.

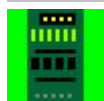
Parametr **address** určuje sběrniceovou adresu modulu **FCPU02** a tato adresa musí být také nastavena na otočném přepínači modulu. Platné adresy jsou 1 až 15, kromě adresy 2, která je přednostně vyhrazena pro modul **CANTERM/CKDM11/CKDM12** (ovládací panel). Tak z jedné adresy sestavy PL2 obsloužíme 64 logických i/o.

Vstupy **In0..3** a parametry **SpecialInput0..3** a **SpecialOutput0..3** byly původně určeny pro budoucí rozšíření využití hardwarových možností modulu. Na komunikaci jiných dat použijte modul **SDO**.

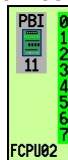
Výstupy **M0 až M7** reprezentují pozice pro zásuvné hardwarové moduly na desce FCPU-02A. Připojení vstupu FCPU-02A modulů PBI-11 nebo PBO-11 na výstup **M** modulu FCPU-02A reprezentuje zasunutí HW zásuvného modulu do příslušné pozice na desce FCPU-02A.

Princip určení i/o brán ve FREDovi zapojením **PBI11**, **PBO11** na **FCPU02** je na obrázku 23.

## 2.8 PBI11



Modul **PBI11** ve spojení s modulem **FCPU02** umožňuje připojení osmi logických vstupů.



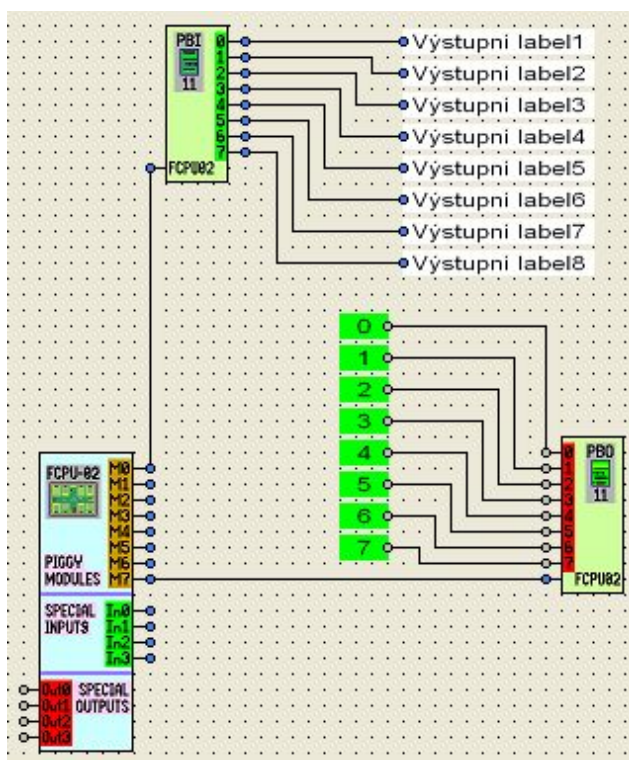
Obr. 24 Značka **PBI11** v knihovně a na ploše

Je reprezentován zásuvným HW (piggy) PBI-11.

Výstupy modulu **PBI11** s označením **0-7** předávají čtenou hodnotu binárních vstupů z HW modulu PBI-11.

Modul **PBI11** lze používat pouze napojený na modul **FCPU02**.

Standardní parametry modulu: **rychlost** a **priorita**.



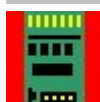
Obr. 23 Zapojení modulů **FCPU02**, **PBI11** a **PBO11**

*Poznámka:*

*U modulů CBI/CBIO je k dispozici parametr filtr pro určení časové konstanty (timeh=timel) filtrace binárních vstupů jednotky v milisekundách.*

*Pro čtení logických vstupů pomocí FCPU-02A a modulů PBI je tento parametr pevný (nezveřejněný) a je přednastaven na 10 ms.*

## 2.9 PBO11



Modul **PBO11** ve spojení s modulem **FCPU02** umožňuje ovládání osmi logických výstupů - relé. Je reprezentován zásuvným HW (piggy) PBO-11.

Stav na vstupech **0-7** modulu **PBO11** určuje stav výstupů HW modulu PBO-11.



Obr. 26 Značka **PBO11** v knihovně a na ploše

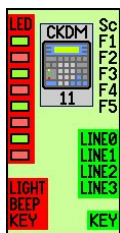
Modul **PBO11** lze používat pouze napojený na modul **FCPU02**.

Standardní parametry modulu: **rychlost** a **priorita**.

## 2.10 CKDM11



Modul **CKDM11** umožňuje práci s terminálem - ovládacím panelem CKDM-11 na sběrnici CAN (s firmware 2.1 a vyšším). V projektu lze použít jeden z modulů terminálů, a to výběrem z modulů **CANTERM**, **CKDM11** či **CKDM12**.



Obr. 28 Značka **CKDM11** v knihovně a na ploše

Ovládací panel jednotky CKDM-11 má čtyřřádkový displej 4 x 20 znaků, 27 kláves a 5 LED.

Jedná se o verzi terminálu, která neobsahuje doplňkové vstupy/výstupy pro připojení vnějších ovladačů a signálů.

Vstupy modulu **CKDM11**, označené symboly **LED** či popisem **Beep**, **Key** a **Light** ovládají napovídané funkce hardware - ovládání 4 dvoubarevných LED, zvukovou signalizaci, simulaci stisku klávesy a ovládání podsvitu alfanumerického displeje.

Výstup **sc** je určen pro připojení modulu **SCREEN** pro tvorbu stromové struktury uživatelského menu. První modul **SCREEN** tvoří tzv. hlavní menu.

Výstupy **F1..5** jsou určeny opět pro připojení modulů **SCREEN** (jako výše) pro tzv. rychlý přístup do často používaných menu.

Do těchto menu se dostáváme stiskem klávesy F1, F2, F3, F4 nebo F5 na ovládacím panelu CKDM-11 (odezva kláves je z hlavního menu).

Výstupy **Line0..3** zobrazují aktuální stav jednotlivých řádků displeje terminálu, bez ohledu na to, je-li terminál fyzicky připojen (hardwarově) či nikoli.

Stačí na ně napojit labely a v ladícím režimu lze aktuální stav řádků v nich jen číst.

Výstup **Key** vrací kód právě stisknuté klávesy na terminálu.

TABULKA kódů kláves modulu **CANTERM**, **CKDM11** i **CKDM12** je přístupná v HELPu.

Standardní parametry modulu: **rychlost** a **priorita**.

Doporučujeme používat parametr **rychlost=3** (defaultní hodnota) pro rychlou odezvu na stisk klávesy.

Parametr **address** určuje sběrniceovou adresu modulu. Pro správnou funkci terminálu, zejména v módu regulátoru system, je třeba, aby **address = 2**.

Parametr **access** určuje úroveň přístupu do systémového módu.

- **access=0** umožňuje přechod do módu system (regulátoru) kdykoli
- **access=1-4** určuje, že pro přechod do módu system musí být přihlášen uživatel alespoň úrovně rovné hodnotě parametru **access**.

Parametr **noterm** běžně = 0.

Je-li =1, neprobíhá CAN komunikace s CKDM-11.

Současně není hlášena chyba CAN v případě, že není terminál vůbec připojen (není v sestavě).

## 2.11 CKDM12



Modul **CKDM12** umožňuje práci s terminálem - ovládacím panelem CKDM-12 na sběrnici CAN. V projektu lze použít jeden z modulů terminálů, a to výběrem z modulů **CANTERM**, **CKDM11** či **CKDM12**.



Obr. 27 Značka **CKDM12** na ploše

Ovládací panel má čtyřřádkový displej 4 x 20 znaků, 27 kláves a 5 LED.

Jedná se o plnou verzi terminálu CKDM-11, která navíc obsahuje doplňkové vstupy/výstupy pro připojení vnějších ovladačů a signálů.

Standardně je osazeno 8 logických vstupů pro bezpotenciálové kontakty, 8 výstupů pro LED, žárovky apod. a 4 napěťové analogové vstupy pro potenciometry. I/O nejsou filtrovány, výstupy jsou při výpadku CAN komunikace nulové.

Standardní parametry modulu: **rychlost** a **priorita**.

Doporučujeme používat parametr **rychlost=3** (defaultní hodnota) pro rychlou odezvu na stisk klávesy.

Parametr **address** určuje sběrniceovou adresu modulu.

Pro správnou funkci terminálu, zejména v módu regulátoru system je třeba, aby adresa = 2.

Parametr **access** určuje úroveň přístupu do systémového módu.

- **access=0** umožňuje přechod do módu system (regulátoru) kdykoli
- **access=1-4** určuje, že pro přechod do módu system musí být přihlášen uživatel alespoň úrovně rovné hodnotě parametru **access**

Parametr **noterm** běžně = 0.

Je-li =1, neprobíhá CAN komunikace s CKDM-12.

Současně není hlášena chyba CAN v případě, že není terminál vůbec připojen (není v sestavě).

Vstupy modulu **CKDM12**, označené symboly **LED** či popisem **Beep**, **Key** a **Light** ovládají napovídané funkce hardware - ovládání 4 dvoubarevných LED, zvukovou signalizaci, simulaci stisku klávesy a ovládání podsvitu alfanumerického displeje.

Výstup **sc** je určen pro připojení modulu **SCREEN** pro tvorbu stromové struktury uživatelského menu. První modul **SCREEN** tvoří tzv. hlavní menu.

Výstupy **F1..5** jsou určeny opět pro připojení modulů **SCREEN** (jako výše) pro tzv. rychlý přístup do často používaných menu.

Do těchto menu se dostáváme stiskem klávesy F1, F2, F3, F4 nebo F5 na ovládacím panelu CKDM-12 (z hlavního menu aplikace).

Výstupy **Line0..3** zobrazují aktuální stav jednotlivých řádků displeje terminálu, bez ohledu na to, je-li terminál fyzicky připojen (hardwarově) či nikoli.

Stačí na ně napojit labely a v ladícím režimu lze aktuální stav řádků v nich jen číst.

Výstup **Key** vrací kód právě stisknuté klávesy na terminálu.

TABULKA kódů kláves modulu **CKDM12** je přístupná v **HELPu**.

Doplňkové vstupy a výstupy terminálu **CKDM-12** jsou zakresleny ve spodní části modulu **CKDM12**.

#### **Logické vstupy:**

Jednotka **CKDM-12** obsahuje 8 logických vstupů pro bezpotenciálové kontakty.

Logické úrovně ze vstupů jednotky **CKDM-12** jsou předány na výstupy **In0..7** modulu **CKDM12**.

#### **Logické výstupy:**

Jednotka **CKDM-12** obsahuje 8 výstupních obvodů pro ovládání dalších LED, signálních žárovek, apod.

Logické úrovně ze vstupů **Out0..7** modulu **CKDM12** ovládají výstupní obvody jednotky **CKDM-12**.

#### **Analogové vstupy:**

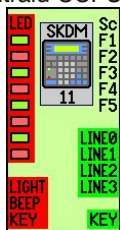
Jednotka **CKDM-12** obsahuje 4 napěťové analogové vstupy (s osmibitovým AD převodem) pro připojení potenciometrů (jako nastavovacích ovladačích prvků s pevným rozsahem 0 až 100).

Naměřená hodnota z těchto vstupů je předána na výstupy **Aln0..3** modulu **CKDM12**.

## 2.12 SKDM11



Modul **SKDM11** umožňuje práci s terminálem - ovládacím panelem **SKDM-11** po sériové lince, napojené na centrálu **CCPU**.



Obr. 29 Značka **SKDM11** na ploše

#### **Poznámka:**

Sériová linka terminálu **SKDM-11** musí být **RS-422** a nastavena:

- 8bit bez parity, 38400 Bd, typ komunikace 73/3 (**COM1**), bez kurzoru.

Lze také použít typ komunikace 73/0 (**COM0**) za předpokladu osazení tohoto kanálu správným piggy (**RS-422**).

V projektu může být použit pouze jeden z terminálů na **CAN** sběrnici (**CANTERM**, **CKDM11**, **CKDM12**) a jeden z terminálů na sériový kanál (**SKDM11**, **SKDM12**).

Ovládací panel má čtyřřádkový displej (4 x 20 znaků), 27 kláves a 5 LED.

Standardní parametry modulu: **rychlost** a **priorita**.

Pro rychlou odezvu na stisk klávesy doporučujeme používat parametr **rychlost=3** (defaultní hodnota) a parametr **priorita=1**.

#### **Poznámka:**

Pokud v projektu použijete některé systémové moduly (např. flash, timdat, hesla) a odezva na stisk klávesy pro vstup do menu těchto systémových modulů bude špatná, nadefinujte prioritu *i* u těchto modulů následně modulo +1 po modulu **SKDM11**.

Ze sériového terminálu se také nedostanete do tzv. systémového menu.

Parametr **COMno** určuje číslo komunikačního kanálu centrály, na který bude terminál připojen.

Defaultně = 2, tedy na **COM2** modulu **CCPU-02**. Tím je tento kanál vyhrazen pouze pro tento terminál.

Parametr **candisp** určuje, zda bude terminál kopírovat terminál na **CANovské** sběrnici (**CANTERM/CKDM11/CKDM12**) nebo zda bude vytvářet vlastní stromovou strukturu menu.

Je-li **candisp=1**, kopíruje, je-li **candisp=0** má **SKDM11** vlastní strom menu.

V případě vlastního stromu menu, nemohou být použity u **SKDM11** a **CANTERM/CKDM11/CKDM12** současně stejné povely (dáno principem funkce modulu **SCREEN**, ale lze vytvořit povely dva a dál je logicky zpracovat).

Rovněž je stejné omezení v používání systémových modulů (některé mohou být v projektu jen jednou).

Editace jedné hodnoty pomocí **SKDM11** a **CANTERM/CKDM11/CKDM12** v jednom projektu se dá udělat pomocí volby **dispedit** modulu **SCREEN**.

**Vstupy** modulu **SKDM11**, označené symboly **LED** či popisem **Beep**, **Key** a **Light** ovládají napovídání funkce hardware

- ovládání 4 dvoubarevných LED, zvukovou signalizaci, simulaci stisku klávesy a ovládání podsvitu alfanumerického displeje.

Podsvit lze ovládat "plynule" hodnotami 0 (bez podsvitu) až 15 (plný podsvit).

**Výstup Sc** je určen pro připojení modulu **SCREEN** pro tvorbu stromové struktury uživatelského menu. První modul **SCREEN** tvoří tzv. hlavní menu.

**Výstupy F1..5** jsou určeny opět pro připojení modulů **SCREEN** (jako výše) pro tzv. rychlý přístup do často používaných menu.

Do těchto menu se dostáváme stiskem klávesy **F1**, **F2**, **F3**, **F4** nebo **F5** na ovládacím panelu **SKDM-11**.

**Výstup Line0-3** zobrazují aktuální stav jednotlivých řádků displeje terminálu, bez ohledu na to, je-li terminál fyzicky připojen (hardwarově) či nikoli.

Stačí na ně napojit labely a v ladícím režimu lze aktuální stav řádků v nich jen číst.

**Výstup Key** vrací krátkodobě kód právě stisknuté klávesy na terminálu.

Pokud je parametr **candisp=1** (pro stejnou činnost terminálu sériového s **CANovským**) je ještě nutno propojit výstup **Key** modulu **SKDM11** (s **prioritou 1**) se vstupem **Key** modulu **CANTERM/CKDM11/CKDM12**.

TABULKA kódů kláves je totožná pro všechny terminály (**CANTERM**, **CKDM11**, **CKDM12**, **SKDM11**, **SKDM12**).

#### **Poznámka:**

Modul **SKDM11** vytváří virtuální modul **CKDM11** na adrese 0. Adresa 0 se na **CAN** sběrnici nesmí nastavit, ale pokud bude nastavena omylem na jiném z modulů projektu, mohla by být funkce modulu **SKDM11** zdánlivě "nevysvětlitelná".

**Pokud v aplikaci použijete jako jediný ovládací panel SKDM-11/12** (bez **CKDM-11/12**) a vyskytnou se problémy se zobrazováním na displeji či při zadávání z klávesnice, pak do projektu zařaďte i modul **CKDM11**, u modulu **SKDM** nastavte parametry **rychlost=3**, **priorita=1**, **candisp=1** a propojte výstup **Key** modulu **SKDM** se vstupem **Key** modulu **CKDM**. Po překladu by mělo pak dojít ke správné funkci **SKDM-11/12**.



## 2.13 SKDM12



Modul **SKDM12** umožňuje práci s terminálem - ovládacím panelem SKDM-12 po sériové lince, napojené na centrálu CCPU-02.



Obr. 30 Značka **SKDM12** na ploše

### Poznámka:

Sériová linka terminálu SKDM-12 musí být RS422 a nastavena:

- 8bit bez parity, 38400 Bd, typ komunikace 73/3 (COM1), bez kurzoru.

Lze také použít typ komunikace 73/0 (COM0) za předpokladu osazení tohoto kanálu správným piggy (RS422).

V projektu může být použit pouze jeden z terminálů na CAN sběrnici (CANTERM, CKDM11, CKDM12) a jeden z terminálů na sériový kanál (SKDM11, SKDM12).

Ovládací panel má čtyřřádkový displej 4 x 20 znaků, 27 kláves a 5 LED.

Jedná se o plnou verzi terminálu SKDM-11, která navíc obsahuje doplňkové vstupy/výstupy pro připojení vnějších ovladačů a signálů.

Standardně je osazeno 8 logických vstupů pro bezpotenciálové kontakty, 8 výstupů pro LED, žárovky apod. a 4 napěťové analogové vstupy pro potenciometry. I/O nejsou filtrovány, stavy výstupů jsou při výpadku komunikace zachovány.

Standardní parametry modulu: **rychlost a priorita**.

Pro rychlou odezvu na stisk klávesy doporučujeme používat parametr rychlost=3 (defaultní hodnota) a parametr priorita=1.

### Poznámka:

Pokud v projektu použijete některé systémové moduly (např. flash, timdat, hesla) a odezva na stisk klávesy pro vstup do menu těchto systémových modulů bude špatná, nadefinujte prioritu *i* u těchto modulů následně modulo +1 po modulu SKDM12.

Ze sériového terminálu se také nedostanete do tzv. systémového menu.

Parametr **COMno** určuje číslo komunikačního kanálu centrály, na který bude terminál připojen.

Defaultně = 2, tedy na COM2 modulu CCPU-02. Tím je tento kanál vyhrazen pouze pro tento terminál.

Parametr **candisp** určuje, zda bude terminál kopírovat terminál na CANovské sběrnici (**CANTERM/CKDM11/CKDM12**) nebo zda bude vytvářet vlastní stromovou strukturu menu.

Je-li **candisp=1**, kopíruje, je-li **candisp=0** má **SKDM12** vlastní strom menu.

V případě vlastního stromu menu, nemohou být použity u **SKDM12** a **CANTERM/CKDM11/CKDM12** současně stejné povely.

Rovněž je stejné omezení v používání systémových modulů (některé mohou být v projektu jen jednou).

Editace jedné hodnoty pomocí **SKDM12** a **CANTERM/CKDM11/CKDM12** v jednom projektu se dá udělat pomocí volby **dispedit** modulu **SCREEN**.

**Vstupy** modulu **SKDM12**, označené symboly **LED** či popisem **Beep**, **Key** a **Light** ovládají napovídáné funkce hardware

- ovládání 4 dvoubarevných LED, zvukovou signalizaci, simulaci stisku klávesy a ovládání podsvitu alfanumerického displeje.

Podsvit lze ovládat "plynule" hodnotami 0 (bez podsvitu) až 15 (plný podsvit).

**Výstup Sc** je určen pro připojení modulu **SCREEN** pro tvorbu stromové struktury uživatelského menu. První modul **SCREEN** tvoří tzv. hlavní menu.

**Výstupy F1..5** jsou určeny opět pro připojení modulů **SCREEN** (jako výše) pro tzv. rychlý přístup do často používaných menu.

Do těchto menu se dostáváme stiskem klávesy F1, F2, F3, F4 nebo F5 na ovládacím panelu SKDM-12.

**Výstupy Line0-3** zobrazují aktuální stav jednotlivých řádků displeje terminálu, bez ohledu na to, je-li terminál fyzicky připojen (hardwareově) či nikoli.

Stačí na ně napojit labely a v ladícím režimu lze aktuální stav řádků v nich jen číst.

**Výstup Key** vrací kód právě stisknuté klávesy na terminálu.

Pokud je parametr **candisp=1** (pro stejnou činnost terminálu sériového s CANovským) je ještě nutno propojit výstup **Key** modulu **SKDM12** (s **prioritou 1**) se vstupem **Key** modulu **CANTERM/CKDM11/CKDM12**.

TABULKA kódů kláves je totožná pro všechny terminály (CANTERM, CKDM11, CKDM12, SKDM11, SKDM12).

Doplňkové vstupy a výstupy terminálu SKDM-12 jsou zakresleny ve spodní části modulu **SKDM12**.

### Logické vstupy:

Jednotka SKDM-12 obsahuje 8 logických vstupů pro bezpotenciálové kontakty.

Logické úrovně ze vstupů jednotky SKDM-12 jsou předány na výstupy **In0..7** modulu **SKDM12**.

### Logické výstupy:

Jednotka SKDM-12 obsahuje 8 výstupních obvodů pro ovládání dalších LED, signálních žárovek, apod.

Logické úrovně ze vstupů **Out0..7** modulu **SKDM12** ovládají výstupní obvody jednotky SKDM-12.

### Analogové vstupy:

Jednotka SKDM-12 obsahuje 4 napěťové analogové vstupy (s osmibitovým AD převodem) pro připojení potenciometrů (jako nastavovacích ovladačích prvků s pevným rozsahem 0 až 100).

Naměřená hodnota z těchto vstupů je předána na výstupy **AIN0..3** modulu **SKDM12**.

Analogové vstupy jsou funkční od verze firmware 2.3, tj. od 26.8.2002.

Poznámka:

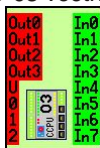
Modul SKDM12 vytváří virtuální modul CKDM12 na adrese 0. Adresa 0 se na CAN sběrnici nesmí nastavit, ale pokud bude nastavena omylem na jiném z modulů projektu, mohla by být funkce modulu SKDM12 zdánlivě "nevysvětlitelná".

**Pokud v aplikaci použijete jako jediný ovládací panel SKDM-11/12 (bez CKDM-11/12) a vyskytnou se problémy se zobrazováním na displeji či při zadávání z klávesnice, pak do projektu zařadte i modul CKDM11, u modulu SKDM nastavte parametry rychlost=3, priorita=1, candisp=1 a propojte výstup Key modulu SKDM se vstupem Key modulu CKDM. Po překladu by mělo pak dojít ke správné funkci SKDM-11/12.**

## 2.14 CCPU-03



Modul CCPU-03 umožňuje práci s HW mikro- počítače CCPU-03 včetně jeho i/o.



Obr. 32 Značka modulu CCPU-03 na ploše.

CCPU-03 je kompaktní mikro počítačová jednotka systému PROMOS line 2 se stejným procesorovým modulem MCPU-01 jako má centrála CCPU-02.

Kromě této desky obsahuje centrála desku logických vstupů a výstupů.

Oproti centrále CCPU-02 má značně rozšířeny komunikační možnosti (krom 3x COM má navíc komunikační kanál USB, Ethernet a volitelně M-Bus), nemá však analogové vstupy a výstupy.

**Základní deska i/o** má standardně osazeny

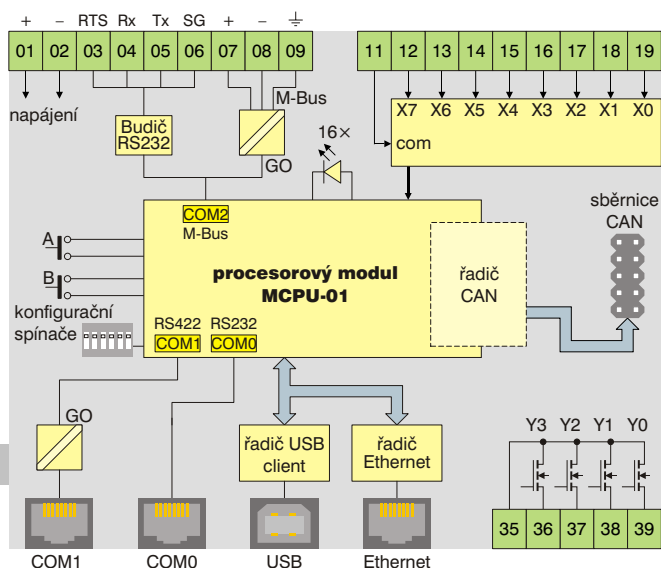
- 8 logických vstupů 24 V s galvanickým oddělením piny sekce **Binary** s označením **In0..7** parametrem **counter\_switch** v dialogu modulu lze nastavit čítací funkce každého **In0..7** zvlášť (v blokovém schéma odpovídá **In0** označení X0, ... **In7** pak X7)  
Pokud se vstup chová jako čítací, předává na svém pinu dvoubajtovou hodnotu vnitřního čítače.  
Její nárůsty umí zpracovat modul **cnt**.  
Perioda testování čítacích vstupů je 10 ms, což odpovídá maximální frekvenci cca 50 Hz, kterou lze ještě vstupem teoreticky zpracovat.  
Po zapnutí jednotky CCPU-03 je vnitřní hodnota čítačů nulována.
- 4 logické výstupy (SSR spínače 50 V / 100 mA, společný mínus) piny sekce **Binary** s označením **Out0..3** jsou určeny pro tyto 4 logické výstupy  
Dále lze pomocí pinů **U 0 1 2** ovládat LEDky na štítku centrály se stejným popisem.  
Standardní parametry modulu: **rychlost** a **priorita**

Poznámka:

Na čelním panelu centrály CCPU-03 jsou přístupné přepínače DIL (popsány dále).

Tlačítka s popisem A a B nelze programově využít.

Tlačítko RESET s popisem IR je schováno nad tlačítkem A.



Obr. 31 Blokové schéma CCPU-03

Je přístupné otvorem nad tlačítkem A a lze stlačit např. pomocí tužky.

Pokud stlačíte a držíte současně tlačítka A+B a na krátkou dobu stisknete RESET, dostanete se do systémového menu centrály.

**Adresu modulu CCPU-03 pro komunikaci** lze nastavit jednak na DIL přepínačích jednotky, jednak pomocí SW.

**Přednost má nastavení na DIL přepínačích a je platné až po RESETu (restartu) centrály!**

SW adresa platí, pokud je na DIL přepínačích nastavena adresa 0. Zadává se jako parametr **address** modulu **system**.

DIL přepínače jsou zleva popsány přímo na přepínačích znaky **1 2 3 4 5 6**.

První dva jsou popsány i na štítku znaky **T E**.

Přepínačem **1** s popisem **T** zařazujeme zakončení sběrnice CAN (terminátor).

Přepínačem **2** s popisem **E** povolujeme (enable) používání logických výstupů (4 SSR spínače) na modulu CCPU-03.

Přepínače **3 4 5 6** jsou určeny pro nastavení adresy modulu CCPU-03.

V poloze **ON** přepínačů **3 4 5 6** tak postupně zařazujeme váhy adresy **8 4 2 1**.

Poznámka:

Komunikační rychlost kanálu COM0 u PL2 (defaultně určeného pro napojení PC s FREDem nebo pro připojení modemu) je implicitně 38400 Bd.

Zatím ji lze měnit pouze použitím (i fiktivním) modulu **modem** a změnou jeho parametru **commspeed**.

Nová komunikační rychlost je platná po překladu - pozor proto na změnu komunikačních parametrů ProgWiinu.

### Komunikační kanály centrály

Centrála má tři klasické sériové kanály - COM0 (RS-232), COM1 (RS-422) a COM2 (RS-232, případně M-Bus).

Dále (např. v systémovém menu centrály) označujeme jako COM3 USB kanál a jako COM4 kanál Ethernet.

Defaultně je COM0 (hlavní kanál) určen pro tzv. QQ relaci, tj. pro napojení k PC s FREDem, pro napojení na dispe-

čink s PC s příslušným SW pro QQ relace, pro napojení k PC s loaderem pro UpGrade firmware centrály.

Centrála CCPU-03 umí na hlavním kanále i zjednodušený protokol **modbus** - bez přepínání. Modbus umožňuje přístup do modulů **COMPORT** a **SCREEN**, a to tak, že **Register address Hi** (terminus technikus protokolu) určuje instanci daného modulu, a **Register address Lo** je pro výstupy **COMPORT 0..31**, vstupy modulu **COMPORT 32..63**. Pro přístup do modulu **COMPORT** musí být tato položka násobek 2. **Register address Lo** 64, 66, 68 a 70 jsou vyhrazeny pro přístup do modulu **SCREEN**, a to na výstupy **edit0..edit3**. Je možné vždy zapisovat i číst, délka dat musí být násobek 4 (tj. No. of Points musí být násobek 2).

Kanál **COM3 = USB** je určen pro stejné funkce jako hlavní kanál (QQ, download, ladící režim, wzorky, epsnet pro ladící režim), není určen pro upgrade FW.

Kanál **COM4 = TCP/IP (Ethernet)** - dtto.

V případě, že je navolen jako hlavní COM4, pak je současně na COM0 implementován protokol TCP/IP přes PPP (dial-up internetové připojení, funkční přes modem na internet providera, nebo bez modemu přímo na COM PC).

*Poznámky:*

- na TCP/IP (zvolen COM4) modul **Actcall** vysílá aktivní volání na Ethernet (nutno nastavit ve WWW konfiguraci - povolit aktivní propojení přes Ethernet)

- na TCP/IP (zvolen COM4) modul **Modem** vysílá aktivní volání přes dial-up internetové připojení zprávu do vzdáleného serveru (nutno nastavit ve WWW konfiguraci - povolit aktivní propojení přes dial-up/PPP)

- je-li COM2 hardwarově M-Bus, pak modul **danfoss** aplikuje na toto rozhraní protokol M-Bus

K dispozici jsou návody na zprovoznění USB, TCP/IP na COMu0 centrály či přes Ethernet. Najdete je na našich stránkách [www.elsaco.cz](http://www.elsaco.cz) (do hledání zadejte *navody.zip*).

## 2.15 CAIO12



Modul **CAIO12** umožňuje práci s hardwarovou jednotkou CAIO-12.



Obr. 34 Značka **CAIO12** na ploše

CAIO-12 je periferní modul na sběrnici CAN s 12 univerzálními pozicemi pro analogové vstupy/výstupy.

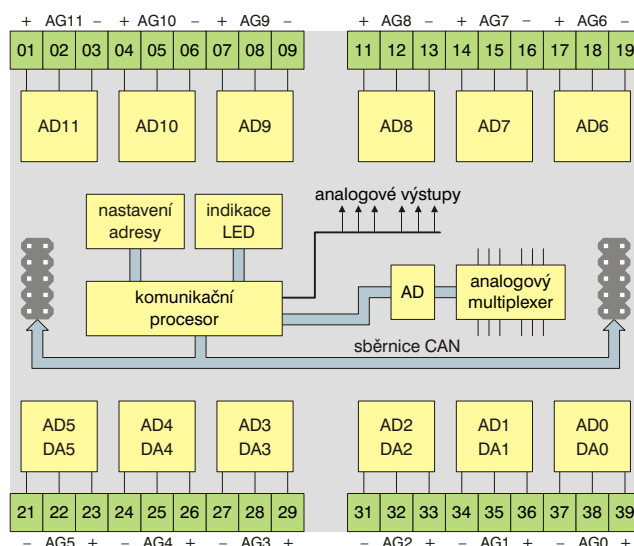
Na univerzální pozice základní desky se podle potřeby osazují analogové I/O moduly.

Moduly jsou v plastových pouzdech a jsou výměnné bez rozebrání jednotky.

**POZOR !**

**I/O moduly (t.zv. domečky) se smí osazovat nebo vyměňovat (tj. zasouvat i vysouvat) pouze při vypnutém napájení sestavy PL2 !**

Analogové vstupní moduly obsahují operační zesilovač s odporovou sítí a podle modifikace umožňují měření napětí,



Obr. 33 Blokové schéma modulu CAIO-12

proudu, odporu nebo přímé připojení odporových čidel Pt100, Ni1000. Typy signálů a rozsahy měření jsou stejné jako u modulu CAIO-11. Protože obsahují i operační zesilovač, nemusí být nepoužité vstupy ošetřovány. Vlastní AD převodník je **šestnáctibitový**. Vestavěný mikroprocesor této jednotky zabezpečuje identifikaci osazeného modulu, automatické nastavení horní a dolní meze rozsahu i linearizaci průběhu teplotních snímačů.

Modulky pro **analogové vstupy** mohou být osazeny na všech pozicích (v blokovém schématu označeno jako AD0..11).

Hodnoty z analogových vstupů jednotky CAIO-12 jsou po filtraci a přepočtu přeneseny na výstupy **In0..11** modulu **CAIO12**.

Pokud je třeba dále měřítkovat (přepočítávat) výstupní hodnotu na některém z výstupů **In0..11**, použijte k tomu modul **scale**.

Moduly **analogových výstupů** obsahují filtr a výstupní zesilovač. Používají se výstupy vestavěného mikroprocesoru s šířkovou modulací (PWM). Rozlišení je osmibitové.

Modulky pro analogové výstupy mohou být osazeny na pozicích DA0..5 (dle blokového schéma), D/A výstupů může být tedy max. 6.

Hodnoty ze vstupů **Out0-Out5** modulu **CAIO12** ovládají analogové výstupy jednotky CAIO-12.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **address** určuje jedinečnou adresu jednotky na sběrnici CAN.

Platné hodnoty jsou 1 až 31.

Parametr **filter** určuje časovou konstantu filtrace analogových vstupů (v ms).

Parametr je ignorován pokud je jeho hodnota < 60.

Parametry **scale0..5** jsou měřítky pro jednotlivé analogové výstupy. Pokud **scale=100**, pracuje výstup v rozsahu 0 až 100 %.



Parametry **Hmez0..11** a **Dmez0.11** jsou vždy vyčteny z modulu CAIO-12 - jejich původní hodnoty jsou vždy přepsány vyčtenými.

Parametr **korekce0..11** je pak pro každý vstup dodatečná korekce, která se přičte k výstupní hodnotě. Tuto má smysl editovat z FREDa.

Funkce parametru **load** je obdobná jako u modulu **CAIO** pro jednotku CAIO-11:

Parametr **load**, pokud=1 způsobí po PŘEKLADu (a po přechodu regulátoru do jeho režimu RUN - nezaměňujte s ladicím režimem FREDa) přesun hodnot parametrů **Hmez**, **Dmez** a **korekce**, které jsme nastavili v režimu regulátoru system (volba test CAIO), z části paměti regulátoru pro zadávání do části paměti regulátoru pro aplikační program.

Tyto hodnoty je pak možné z regulátoru načíst do projektu ve FREDovi volbou *načtení konstant modulu*.

Pokud je load=0, systém použije hodnoty parametrů z FREDa.

Funkce vstupů a výstupů závisí na osazení jednotky zásuvnými moduly. Platná data budou pouze na těch výstupech modulu, které budou mít osazené správné zásuvné moduly.

Vzhledem k tomu, že meze se vždy přepíšou hodnotami načtenými z I/O modulkou, má praktický význam jen načtení parametru **korekce**.

*Poznámky:*

*Jednotka CAIO-12 má 12 univerzálních pozic pro zásuvné moduly, některé z nich mohou pracovat buďto jako vstupy, nebo jako výstupy.*

*SW je zpracován tak, že všech 12 pozic může být osazeno pro analogové vstupy a pozice 0-5 mohou být osazeny pro analogové výstupy.*

*ve FREDovi pak zapojujeme piny modulu CAIO12 podle osazení univerzálních pozic, a to od varianty 12 ADin + 0 ADout až po variantu 6 ADin + 6 ADout.*

*Příklad:*

*Pokud osadíme univerzální pozici 5 pro analogový výstup, zapojíme ve schématu pin Out5 a nesmíme zapojit pin In5! Pozice se počítají od 0.*

*Hodnoty po osazení nebo záměně či výměně domečku jsou načteny jednotkou CAIO-12 až po restartu jednotky. Hodnoty jsou dále načteny centrálou až po restartu aplikace v centrále (reset nebo nahrátí projektu).*

*Je vhodné proto všechny tyto úpravy dělat na vypnutém systému a pak vše najednou zapnout.*

#### Perioda komunikace i/o modulů na sběrnici CAN

Aplikační program v centrále (vytvořený ve FREDovi) běží ve třech programových smyčkách.

Periody vykonávání naprogramovaných modulů v projektu (podle parametru rychlost v každém modulu) jsou 3000 ms, 1000 ms a 100 ms.

Kromě toho centrála s jednotlivými periferními moduly (i/o moduly) komunikuje po sběrnici CAN, aby z nich vyčetla nebo jim předala požadované/vypočtené informace.

Informace z/do binárních (logických) vstupů a výstupů jsou komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy).

Informace do analogových výstupů jsou rovněž komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy).

Informace z analogových vstupů 0..3 jednotek CAIO-11/12 jsou komunikovány s periodou 25 ms (pro adresy modulů 1..4) nebo 50 ms (pro ostatní adresy), ale z ostatních vstupů 4..11 jsou komunikovány s periodou 350 ms.

Proto je vhodné používat vstupy 0..3 modulů **CAIO** a **CAIO12** přednostně pro regulační smyčky s velkými nároky na rychlost.

#### K firmware CAIO-12:

- doplněny čítače na analogových vstupech.

Rozhodovací úroveň pro čítání je polovina rozsahu, hystereze je 7 % (z rozsahu).

Čítače jsou 16 bitové na objektu 0x4100 (stejně jako u binárních jednotek).

Navíc jsou hodnoty čítačů na vstupech 0 a 1 namapovány do PDO4 za hodnoty analogových vstupů.

Minimální detekovaná šířka pulsu je 20ms (= perioda samplování analogových vstupů).

Čítá se z nefiltrovaných analogových hodnot.

## 2.16 CCPU-21



Modul **CCPU-21** umožňuje práci s HW mikropočítače CCPU-21 včetně jeho i/o.



Obr. 35 Značka modulu **CCPU-21** na ploše

**CCPU-21** je kompaktní mikropočítačová jednotka systému PROMOS line 2 s vestavěným ovládacím panelem.

Jednotku tvoří procesorový modul MCPU-01, základní deska vstupů/výstupů a deska panelu s displejem, tlačítky a LED.

Sériové linky jsou vyvedeny na konektory a jsou osazeny rozhraním 1x RS-232, 1x M-Bus/RS-232 a 1x RS-422/485 s GO s vestavěným napájecím měničem. M-Bus umožňuje připojit až 3 slave zařízení.

Displej má 4 řádky o 20 znacích, ovládaný podsvit. Výpisy tedy jako u modulů CKDM.

Tlačítková klávesnice (9 tlačítek) je překryta fólií.

#### Základová deska standardně obsahuje:

##### – 6 binárních vstupů

24 V (společná svorka "com") s GO 1500 V AC

piny sekce **Binary** s označením **In0..5**

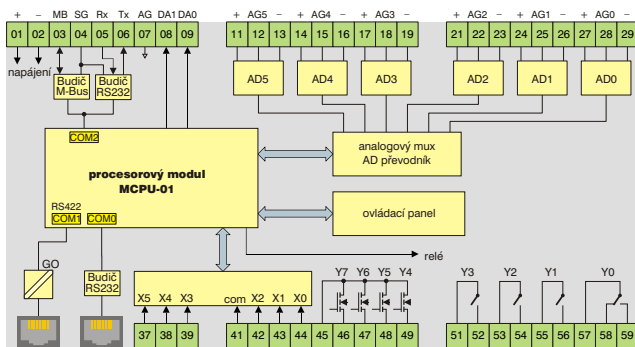
parametrem **counter\_switch** v dialogu modulu lze nastavit čítací funkce každého **In0..5** zvlášť

v blokovém schéma odpovídá In0 označení X0, ... In5 pak X5

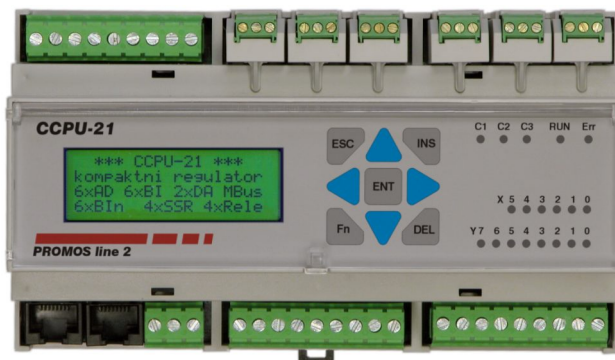
##### – 4 reléové výstupy

kontakt 250 V AC / 5 A, GO 5000 V AC

piny sekce **Binary** s označením **Out0..3**



Obr. 37 Blokové schéma mikropočítače CCPU-21



Obr. 36 Mikropočítač CCPU-21

- **4 MOSFET spínače**  
50 V / 250 mA s GO 1500 V  
piny sekce **Binary** s označením **Out4..7**  
v blokovém schéma odpovídá Out0 označení Y0, ... Out7 pak Y7
- **6 univerzálních pozic pro analogové vstupy**, které lze osadit výměnnými moduly:  
piny sekce **Analog** s označením **In0..5**  
v blokovém schéma pozice **AD0..5**

meze jsou vyčteny z domečku (z výměnného modulu) pokud meze nelze vyčíst z domečku, pak platí meze v příslušném parametru **Dmez0..5** a **Hmez0..5** hodnota parametru **korekce0..5** se přičítá k příslušné výsledné analogové hodnotě

parametr **linearisation**, přístupný přes tlačítko DIALOG MODULU, povoluje generování chybového hlášení pozice bitu v parametru odpovídá pozici analogového vstupu, bit v 1 povoluje kód chyby odpovídá pozici nenalezeného domečku (0..5)

např. při **linearisation** = 00000001 hlášení:

**Er 000,000@044,001** říká, že nebyly načteny meze pro **Analog In0**

parametr **filter** umožňuje nastavit hodnotu filtru pro analogové vstupy  
filtruje v periodách, možné nastavení 1..4000  
(např. **rychlost** = 3 [100ms], **filter** = 10 => časová konstanta  $t = 100ms * 10 = 1 sec$ ), což je default, aby to bylo stejné jako u modulu **CAIO12**

Výměnný modul pro **analogový vstup** obsahuje operační zesilovač s odporovou sítí a paměť s parametry pro konkrétní vstupní modul. Podle typu umožňuje měření napětí, proudu, odporu nebo přímé připojení odporového čidla Pt100, Pt500, Pt1000, Ni1000, KTY... Výměnné moduly jsou stejné jako pro CAIO-12.

Výměnný modul typu **EBI** je určen pro 2 binární vstupy, dochází tak k převodu analogového vstupu na dva vstupy binární. Použijeme-li **EBI-1x**, pak na příslušném pinu **In0..5** je výstupní signál typu integer a připojíme-li na tento pin modul **int\_to\_bin** získáme na výstupech tohoto modulu požadované binární signály rozkódované na jednotlivé bity (X0 a X1 dle popisu na domečku).

A/D převodník je **šestnáctibitový**.

Vstupy **0..2** modulu **CCPU-21** ovládají LED na panelu centrály s popisem Com0..2.

Standardní parametry modulu: **rychlost** a **priorita**.

### Komunikační kanály centrály

Centrála má tři klasické sériové kanály - COM0 (RS-232), COM1 (RS-422/485 s GO) a COM2 (RS-232, případně M-Bus).

Hlavní kanál (defaultně COM 0) je určen pro tzv. QQ relace, tj. pro napojení k PC s FREDem, pro napojení na dispečink s PC s příslušným SW pro QQ relace, pro napojení k PC s loaderem pro UpGrade firmware centrály.

Komunikační rychlost hlavního kanálu (defaultně COM0) je implicitně 38400 Bd.

Lze ji měnit pouze použitím (i fiktivním) modulu **modem** a změnou jeho parametru **commspeed**.

Centrála CCPU-21 umí na hlavním kanále i zjednodušený **protokol modbus** - bez přepínání (pro režim slave). Modbus umožňuje přístup do modulů **COMPORT** a **SCREEN**, a to tak, že **Register address Hi** (terminus technikus protokolu) určuje instanci daného modulu, a **Register address Lo** je pro výstupy COMPORT 0..31, vstupy modulu COMPORT 32..63. Pro přístup do modulu COMPORT musí být tato položka násobek 2. **Register address Lo** 64, 66, 68 a 70 jsou vyhrazeny pro přístup do modulu SCREEN, a to na výstupy **edit0..edit3**. Je možné vždy zapisovat i číst, délka dat musí být násobek 4 (tj. No. of Points musí být násobek 2).

### CCPU-21 - základní funkce

#### Zapnutí

Pokud není v CCPU-21 platná konfigurace z FREDa, přejde CCPU-21 ihned do systémového menu. Tento stav je indikován trvalým svitem LED RUN na panelu (zelená).

Pokud je v CCPU-21 platná konfigurace, spustí se aplikace. Tento stav je indikován blikáním LED RUN (zeleně).

#### Přechod do systémového menu

Pokud během restartu/zapnutí je stisknuto tlačítko **Fn**, centrála po zapnutí přejde vždy do systémového menu (obdoba A+B+reset u CCPU-02-03).

#### Závada hardware

Pokud by došlo závadou HW CCPU-21 k narušení obsahu paměti flash, po startu zůstanou svítit na panelu všechny LED (obdoba rychlého blikání LED RUN+ERR u CCPU-02/03).



## Ovládání

Pomocí devíti tlačítek na panelu je možné ovládat CCPU-21 podobným způsobem, jako z terminálu CKDM.

### Základní význam tlačítek

Základní význam tlačítek je na nich vyznačen černě. Pomocí šipek a tlačítek **ENTER** a **ESCAPE** je možné se pohybovat v menu stejně, jako z ovládacího terminálu CKDM. Dlouhým stiskem (5 sec) tlačítka **ESC** lze přejít do systémového menu.

Tlačítky +/- lze v některých menu nastavovat hodnotu stejným způsobem, jako u CKDM.

### Druhý význam tlačítek

Druhý význam tlačítek se volí stiskem tlačítka **Fn**. V tomto stavu mají tlačítka funkci, která je na nich vyznačena červeně. To je indikováno zobrazením menšího **F** v pravém horním rohu LCD. V tomto režimu jsou k dispozici klávesy **F1..F5**, umožňující přímý přechod do menu, dále tlačítka **DEL**, umožňující mazat chybová hlášení, a dále tlačítka +/-, mající funkci odpovídající Shift+ a Shift- na CKDM (umožňují větší krok nastavování hodnot v některých menu).

### Význam tlačítek v editačním režimu

Editační režim se volí dvojným stiskem tlačítka **Fn**. Druhý stisk tlačítka **Fn** odpovídá stisku klávesy **INS** na CKDM. Tento režim je indikován zobrazením menšího **E** v pravém horním rohu LCD. V editačním režimu mají tlačítka ten význam, který je na nich vytištěn modře. Stiskem tlačítka **Fn** v editačním režimu se zvolí druhý význam tlačítek v editačním režimu. Tento stav je indikován současným zobrazením menších **F** a **E** v pravém horním rohu LCD. V tomto stavu mají tlačítka ten význam, který je na nich vytištěn šedě.

*Poznámka: pokud se vstoupí do editačního režimu omylem (například na řádku menu, kde není nic k editaci), lze editační režim vždy zrušit Escapem (v editačním režimu to je sekvence Fn - ESC).*

## Speciální kombinace

### Blokování binárních výstupů CCPU-21:

stisknout a podržet tlačítka **Fn**. Dále stisknout tlačítka **DEL**. Stav blokování binárních výstupů je indikován LED **BLK** na panelu.

### Reset CCPU-21:

stisknout tlačítka **Fn**. Znovu stisknout a držet tlačítka **Fn**. Stisknout tlačítka **DEL**.

## Zobrazení

LCD zobrazuje obrazovky z modulu **SCREEN** stejným způsobem, jako terminál CKDM. Do projektu ve FREDovi je tedy třeba zařadit modul **CKDM11**. Navíc má možnost zobrazit stav vstupů modulu **CKDM11**, které na CKDM-11 ovládaly LED. Stav těchto vstupů indikují malé symboly **G** (jako Go, místo RUN na CKDM), **1** (odpovídá Mode 1 na CKDM), **2** (odpovídá Mode 2 na CKDM) a **E** (odpovídá Error na CKDM). Trvalé zobrazení těchto symbolů odpovídá svitu příslušných LED zeleně, blikání symbolů odpovídá svitu příslušných LED červeně.

Oproti CKDM umí CCPU-21 zobrazovat malá písmena s diakritikou. Znak stupeň Celsia má stejně jako na CKDM kód 223.

## Download

Režim download je indikován zhasnutím LED **RUN** a svitem LED **ERR** červeně.

## Systémové menu

Systémové menu nemá, na rozdíl od CCPU-02/03, volby pro test periferií na sběrnici CAN. Naopak, navíc má možnost nastavení komunikační adresy v menu **F4 Kanál** (funkce odpovídá nastavení adresy na DILech u centrály CCPU-02/03). Adresa se v tomto menu nastavuje tlačítky šipka nahoru/šipka dolů, potvrzuje se stiskem klávesy **ENTER**. Je možné nastavit adresu 0..15, význam nastavení je stejný jako u centrály CCPU-02/03. Změna nastavení adresy se projeví až po restartu centrály, a to i v menu **F4 kanál**.

Test I/O CCPU-21 je realizován pouze jednoduchým způsobem, obdobně jako test CCPU-02.

## Verze firmware

Číslo verze firmware je shodné s odpovídající verzí firmware pro CCPU-02/03. Rozdíly ve firmware jsou v tom, že firmware pro CCPU-21 nemá podporu sběrnice CAN, a v systémovém menu, viz výše.

Firmware pro CCPU-21 a CCPU-02/03 však nelze vzájemně zaměňovat.

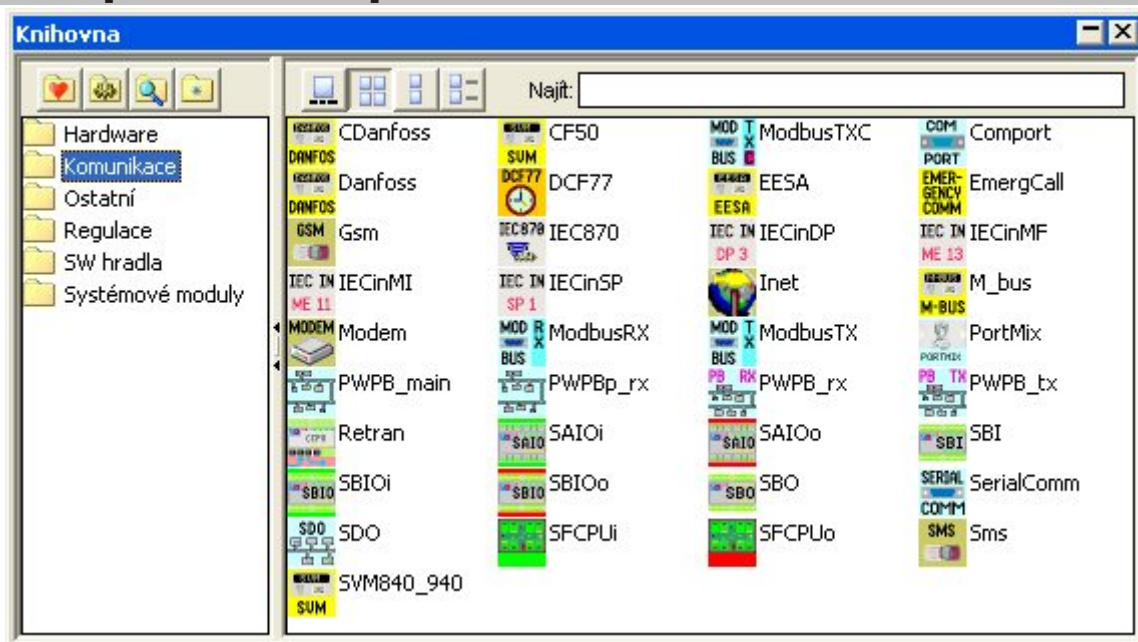
*Poznámky:*

*Označení **FW 3.007** je pro CCPU-02/03 i CCPU-21 totožné.*

*FRED je tedy v příslušné verzi stejný pro všechny centrály, které má v knihovně.*

*Soubory s příponou .S24 pro FW centrály CCPU-02/03 se nadávají zaměňovat se soubory pro FW centrály CCPU-21.*

### 3 Skupina modulů pro komunikaci



Obr. 38 Okno knihovny modulů - aktivní skupina Komunikace

Ve skupině **komunikace** jsou knihovní moduly pro komunikaci mezi regulátorem a ostatními zařízeními:

- **serialcomm**  
pro realizaci jednoduchých komunikačních relací
- **modem**  
ovládání telefonního modemu na COM0
- **emergcall**  
modul pro nouzovou komunikaci
- **svm840/940**  
pro komunikaci s měřičem tepla SVM 840/940
- **cf50**  
pro komunikaci s měřičem tepla CF50
- **danfoss**  
pro komunikaci s měřiči tepla DANFOSS INFOCAL5, SVM-F3 a 2WR5 komunikace M-Bus
- **cdanfoss**  
modul pro vyčítání čítačů z měřiče DANFOSS INFOCAL5 komunikace manufacturer
- **comport**  
modul pro koncentraci dat, určených pro přenos na dispečink
- **pwpb\_main**  
pro základní definici MPC protokol Epsnet/Profibus
- **pwpb\_rx**  
MPC - příjem
- **pwpb\_tx**  
MPC - vysílání
- **pwpbp\_rx**  
MPC - příjem dle objektu, instance, položky
- **sdo**  
pro speciální komunikaci na CANu s periferiemi
- **gsm**  
základní modul pro vysílání a příjem SMS zpráv
- **sms**  
vyhodnocuje přijaté SMS zprávy, odpovídá
- **dcf**  
modul pro přijímač časových značek DCF 77
- **ModbusRX**  
modul protokolu modbus pro příjem
- **ModbusTX**  
modul protokolu modbus pro vysílání
- **M-Bus**  
modul pro dálkové odečty měřidel na sběrnici MBUS
- **EESA**  
modul pro komunikaci s měřiči tepla EESA MT200
- **SAIOi**  
modul pro čtení analogových hodnot z jednotky SAIO-11/12
- **SAIOo**  
modul pro ovládání analogových výstupů jednotky SAIO-11/12
- **SBI**  
modul pro čtení stavů logických vstupů na jednotce SBI-11/12
- **SBO**  
modul pro ovládání relé jednotky SBO-11/12
- **SBIOi**  
modul pro čtení stavů logických vstupů na jednotce SBIO-11/12
- **SBIOo**  
modul pro ovládání relé modulu SBIO-11/12
- **SFCPUi**  
modul pro čtení stavů logických vstupů na jednotce FCPU-02A
- **SFCPUo**  
modul pro ovládání logických výstupů jednotky FCPU-02A
- **ReTran**  
modul pro přepuštění QQ relací na zvolený retranslační kanál centrály
- **PortMix**  
slouží pro komunikaci s modulnetem

- **ModbusTXC**  
modul je určen pro komunikaci protokolem modbus proti modulu ModbusTX je rozšířen
- **Inet**  
modul je určen pro údržbu spojení na internetu
- **IEC870**  
modul je určen pro komunikaci protokolem IEC 870-5-101
- **IECinSP**  
rozšiřující modul k IEC870 pro jednobitové signály
- **IECinDP**  
rozšiřující modul k IEC870 pro dvoubitové signály
- **IECinMI**  
rozšiřující modul k IEC870 pro vstupy typu integer
- **IECinMF**  
rozšiřující modul k IEC870 pro vstupy typu float

*Poznámka:*

*Od verze FW 3.017 je umožněno volit u sériových kanálů rychlost 200 Bd, a to s jistým omezením. V jednom projektu nemohou být současně moduly s navolenou rychlostí 200 a 300 Bd. Lze volit pouze jednu z nich pro celý projekt.*

### 3.1 serialcomm



Modul **serialcomm** je určen pro realizování jednoduchých komunikačních relací na sériovém kanálu COM0/1/2.

Každý modul **serialcomm** uskuteční celou relaci, sestávající z otevření a uzamčení komunikačního portu, vyslání zprávy podle formátovacího řetězce pro vysílání, přijmutí zprávy, dekódování zprávy podle formátovacího řetězce pro příjem, zavření a odemčení komunikačního portu pro použití dalším modulem. Je tedy možné použít více modulů **serialcomm** v jednom projektu a tak komunikovat s více zařízeními (např. několik regulátor PL2 na RS485), připojenými na linku COM0/1/2.

**Formát komunikace:** 7 až 9-bitová data (dle parametru **format**), s nastavením parity (parametr **Parity**) a komunikační rychlosti (parametr **Baudrate**). Parametry lze zvlášť nastavovat pro příjem a zvlášť pro vysílání - viz níže.



Obr. 39 Značka **serialcomm** na ploše

Vstup **strobe** (pin vlevo nahoře) slouží k blokadě komunikace. Blokadě nastane při stavu log. 1.

Pokud bude na tomto vstupu log. 0, bude probíhat komunikace vždy po průchodu programovou smyčkou, tj. při parametru **rychlost=1**, bude probíhat komunikace jednou za 3 vteřiny, ale to pouze tehdy, bude-li v projektu jen jediný modul **serialcomm**.

Bude-li modulů **serialcomm** v projektu více, bude se při správně nastavené prodlevě parametrem **timeout** vykonávat průchodem programovou smyčkou jen jeden, a to postupně po sobě podle umístění v projektu na ploše schéma, resp. podle priority modulu.

Modul **serialcomm** nejprve vyšle požadavek (první průchod programovou smyčkou) a potom očekává odpověď (druhý průchod programovou smyčkou), a to i v případě, že není formátovacím řetězcem předepsána. Proto pokud potřebujeme docílit rychlého vyslání samostatných zpráv modulem **serialcomm**, musíme počítat s tím, že zpráva bude vyslána vždy ob jednu programovou smyčku, tj. např. při **rychlost=3** až po 200 ms. Bude-li takto více modulů **serialcomm** správně po sobě navazovat, proběhne vyslání z každého z nich už po 100ms, ale při přechodu z "posledního" na "první" bude zase až za 200ms.

Vstupy **Tx0..3** slouží pro napojení datových signálů z projektu pro jejich vyslání na sériovou linku modulem **serialcomm**.

Výstupy **Rx0..3** slouží pro připojení získaných hodnot (přijatých dat) do projektu.

Výstup **Error** (vpravo nahoře) je aktivován v případě, že na výzvu modulu **serialcomm** nepříjde do doby **timeout** odpověď, nebo dojde-li k chybě přijímaných dat (nevyjde kontrolní součet, nepříjde očekávaný znak ap.).

#### Poznámka:

Pokud na výstup **Rx0..3** napojíme label a po překladu přejdeme do ladící režimu **FREDA**, bude v něm správně zobrazena pouze hodnota typu *integer*, tj. při definici příjmu bajtu.

Při definici příjmu třibajtového reálného čísla z regulátorů řady **PROMOS RT/RTm/RT40** je sice provedena konverze na čtyřbajtové vyjádření *IEEEfloat*, které používá řada regulátorů **PL2**, ale tato hodnota bude správně zobrazena až za (nebo v) modulem (modulu **SCREEN**) pro float aritmetiku, např. za hradlem *aplus*.

Pro správné obslužení těchto vstupů a výstupů slouží následné formátovací řetězce.

**Formátovací řetězec** (max. 64 znaků) **pro vysílání** rozpoznává tyto příkazy:

– přímo hexadecimální číslo **00** až **FF** (velkými písmeny), např. **3F**.

Tento znak je vyslán na linku.

– příkaz **l0..3** (velké písmeno **l** s indexem pinu) vyšle jeden bajt ze vstupu **Tx0..3**

– příkaz **f0..3** (malé písmeno **f** s indexem pinu) vyšle reálné číslo ze vstupu **Tx0..3** ve formátu 3bytefloat

– příkazy **apz**, **bpz**, **cpz**, **dpz** slouží pro vyslání reálného čísla ze vstupů **Tx0..3** jako ASCII znaky na linku

**a** pro **Tx0**, **b** pro **Tx1**, **c** pro **Tx2**, **d** pro **Tx3**

**p** = počet znaků celkem včetně desetinné tečky

**z** = počet cifer za desetinnou tečkou

maximálně 3 desetinná místa a celkem 8 cifer

je-li cifer pro přenos méně než je udáno ve formátu, jsou zleva doplněny mezerami

při překročení formátu je na linku vysláno **EE.EE** (počet **E** podle formátu)

např. **a52** pro vyslání hodnoty reálného čísla ze vstupu **Tx0** jako ASCII znaky ve tvaru 12.45

– příkazy **hn**, **in**, **jn**, **kn** slouží pro vyslání integer hodnoty ze vstupů **Tx0..3** jako **n** znaků ve formátu ASCII hexa  
např. **h4** vyšle ve formátu ASCII hexa 4bytový string, vyjadřující hexadecimálně integer hodnotu na vstupu **Tx0**.  
Direktivy **h..k** se hodí i pro práci s binárními hodnotami HW modulů **SAM** ve spojení s moduly **BINtoINT** a **INTtoBIN**.

Typický příkaz pro nastavení binárních výstupů **SAMu-02** (adresa=1) bude **403031h10D**,

kde na vstup **Tx0** bude připojen výstup **BINtoINT** (pro **SAM-02** stačí, budou-li funkční první čtyři binární vstupy).

– příkaz **Xn** (velké písmeno **X** s cifrou) spočítá (a vyšle) kontrolní součet znaků logickou funkcí **XOR** od znaku daného parametrem **n** (např. **X3** od třetího znaku, pozor - počítáno od 0. znaku) do konce zprávy. Příkaz **Xn** musí být vždy na konci zprávy. **Pro relace s regulátory PROMOS RT/RTm/RT40 užívejte formát X2 !**

*Poslední příkaz typu **Xn** lze nahradit příkazem **Sn** (pro aritmetické sčítání znaků od n-tého, modulo 256) nebo příkazem **Nn** (doplňk do 0 přechozího **Sn**).*

**Formátovací řetězec** (max. 64 znaků) **pro příjem** rozpoznává tyto příkazy:

– příkaz **##** ignoruje 1 bajt (znak) v příjmu zprávy

– přímo hexadecimální číslo **00** až **FF** (velkými písmeny), např. **A2**

Porovná přijatý znak, souhlasí-li, pokračuje dalším dekódováním, jinak končí a zprávu ignoruje.



- příkaz **O0..3**  
(velké písmeno **O** s indexem pinu)  
předá přijatý bajt (znak) na výstup **Rx0..3**
- příkaz **I0..3**  
(písmeno **I** [malé el] s indexem pinu) = předá 4 bajty (znaky) zprávy na výstup **Rx0..3**
- příkaz **f0..3**  
(malé písmeno **f** s indexem pinu)  
předá reálné číslo na výstup **Rx0..3**, číslo je výsledek konverze ze 3bytefloat formátu ze 3 bajtů přijatých dat
- příkaz **an až dn**  
(malá písmena **a, b, c, d** s cifrou)  
přečte **n** znaků jako ASCII číslo a předá na výstup **Rx0..3** (př. **b5** přečte následujících 5 znaků jako ascii řetězec, konvertuje na reálné číslo a předá na výstup **Rx1**)
- příkaz **hn až kn**  
(malá písmena **h, i, j, k** s cifrou) = přečte **n** znaků ve formátu ASCII hexa a uloží jako hodnotu integer do výstupu **Rx0..3**  
např. **h4** přijme čtyři znaky ve formátu ASCII hexa a uloží je jako int do **Rx0**  
jako první znak čísla může být + nebo - (znaménko) nebo > či < (interpretuje se jako + či -; kvůli modulům SAM-01/02/04).  
Typický formátovací řetězec pro vyčtení čítačů z modulu SAM-02 pak bude  
h52Ci42Cj42Ck40D  
kde první hodnota bude buď kladná nebo záporná podle toho, jestli je na začátku > nebo <.
- příkaz **Xn**  
(velké písmeno **X** s cifrou)  
spočítá kontrolní součet logickou funkcí XOR od znaku daného parametrem **n** (pozor, počítáno od 0. znaku) do konce zprávy a porovná s posledním znakem. Pokud kontrolní součet nesouhlasí, zprávu zahodí. Příkaz **Xn** musí být vždy na konci zprávy. **Pro relace s regulátory PROMOS RT/RTm/RT40 užívejte formát X2 !**

Poslední příkaz typu **Xn** lze nahradit příkazem **Sn** (pro aritmetické sčítání znaků od n-tého, modulo 256) nebo příkazem **Nn** (doplňek do 0 přechozího **Sn**).

Dekódované hodnoty jsou do výstupů uloženy až když je celá zpráva správně dekodovaná. Dojde-li při dekodování k chybě, stav výstupů se nezmění.

Formátovací řetězce jsou přístupné jako parametry **Message\_tx\_format** a **Message\_rx\_format** v dialogovém okně modulu **serialcomm**, a to po kliknutí na tlačítko **DIALOG MODULU**. Každý parametr je zadáván až do dvou řádků po 32 znacích. Aby se bralo formátování i podle druhého řádku, musí být první řádek právě naplněn 32 znaky.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **COMnr** určuje svou hodnotou číslo komunikačního kanálu, na kterém komunikační relace proběhne.

- **0** - pro COM0 (QQ relace přesměrujte na jiný kanál pomocí systémového menu centrály - volba F4 kanál)
- **1** - pro COM1
- **2** - pro COM2

Nesmí být použit kanál, který používají moduly MPC (pro ProfiBus, tj. **pwpb\_main**, **pwpb\_rx** a **pwpb\_tx**).

Parametr **timeout** určuje časový limit pro dobu mezi znaky v ms (Timeout pro celou zprávu je pak 256 \* meziznakový timeout).

Parametr **Baudrate** určuje komunikační rychlost. Jde o dvojciferné dekadické číslo, kde **jednotky** určují rychlost **pro vysílání** a **desítky** rychlost **pro příjem**.

Hodnota **0** odpovídá rychlosti 300 Bd, **2** = 600 Bd .....až **9** odpovídá rychlosti 115 200 Bd.

hodnota	rychlost
0	200 / 300
1	600
2	1 200
3	2 400
4	4 800
5	9 600
6	19 200
7	38 400
8	57 600
9	115 200

Příklad:

Parametr **26** určuje **vysílací rychlost 1200 Bd** a **přijímací 19 200 Bd**.

Parametr **Parity** slouží pro nastavení parity. První cifra určuje paritu vysílané zprávy, druhá paritu přijímané zprávy

0 = žádná, bez parity

1 = lichá

2 = sudá

Parametr **Format** slouží pro nastavení počtu databitů. První cifra určuje počet databitů pro vysílání, druhá pro příjem.

Možné kombinace: 77, 78, 79, 87, 88, 89, 97, 98, 99.

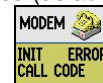
Poznámka:

Praktické příklady formátovacích řetězců jsou zveřejněny v HELPu programu FRED. Jedná se o ovládání číslicového zobrazovače XDM-14 a o komunikaci mezi novým systémem PROMOS line 2 a starším PROMOS RT / RTm / RT40.

## 3.2 modem



Modul **modem** ovládá připojený telefonní modem na sériový kanál, na kterém je nastavena tzv. QQ komunikace (defaultně kanál COM0).



Obr. 40 Značka **modem** na ploše

Vlastní modem musí být předem nakonfigurovaný a tato konfigurace uložena v jeho paměti FLASH.

Použití GPRS modemu - viz dále.

Vstup **Init** umožňuje provést náběžnou hranou reinicializaci modemu.

Vstup **Call code** spouští tzv. aktivní hlášení.

Výstup **Err** se aktivuje pokud nedojde k úspěšné inicializaci modemu nebo pokud nedojde ke spojení víc než **maxcall**-krát za sebou.

V případě požadované inicializace modemu po jeho zapnutí (i po výpadku napájení modemu) je nutno v modulu **modem** zadat konfigurační řetězce pro inicializaci telefonního modemu do parametrů **init\_string\_0..2**.

Poznámka:

Inicializační řetězec je do telefonního modemu vyslán vždy po překladu spuštěním nově nahraného projektu do PL2, po zapnutí regulátoru PL2 nebo po resetu PL2.

Od verze FW 3.001 byl prodloužen timeout pro odpověď na inicializační řetězec modemu na 3 sekundy.

#### **Aktivní hlášení (také jen AH):**

V případě, že hodnota na vstupu **Call code** je různá od hodnoty parametru **actcode**, modem se pokusí navázat spojení vysláním volacího řetězce, uloženého v řetězci parametru **Call string**:

např. ATDT321727753 = modem vytočí telefonní číslo 321727753 tónovou volbou.

Pokud se navázání spojení nezdaří, modul **modem** se pokusí spojení navázat ještě několikrát (podle hodnoty parametru **maxcall**) v intervalech po 8 minutách.

Pokud se nepodaří navázat spojení ani tolikrát, kolikrát je definováno hodnotou parametru **maxcall**, je spojení opakováno až další den a výstup **Err=1**.

Po úspěšném navázání spojení telefonní modem při **comspeed=38400** pošle do centrály zprávu

**CONNECT 38400<CR>**

do 20 vteřin. Pak je vyslána dalším průchodem modulu **modem** programovou smyčkou, tj. po 3 sec, na dispečink modemem QQ relace (zpráva) obsahující **Call code**: s adresou volající stanice PL2 pro objekt 0, instanci 0, položku 1 s daty=hodnotě na vstupu **Call code**. Data jsou 4 bajtová, hodnota typu integer long.

Z této relace ovladač na dispečinku vyhodnotí adresu volající stanice i poslaný chybový kód - hodnotu na vstupu **Call code**.

Dispečink pak musí do 30 sec zabezpečit potvrzení aktivního hlášení, tj. musí odpovědět zápisem hodnoty chybového kódu do parametru **actcode**.

Tím je AH potvrzeno a nebude proto po 8 minutách opakováno.

Je potom na dispečinku, aby si zjistil dalšími relacemi podrobnosti o chybovém stavu.

Zavěsí dispečink, při výpadku spojení zavěsí po určité době modem automaticky.

Neobdrží-li stanice PL2 potvrzení AH, pak modem zavěsí a po 8 minutách učiní další pokus o navázání spojení.

Pokud se spojení zdaří (včetně potvrzení AH), lze vstupem **Call code** inicializovat další spojení ihned (dříve po 8 minutách).

### **Přehled parametrů modulu modem**

Standardní parametry modulu: **rychlost a priorita**.

Parametr **comspeed** definuje komunikační rychlost linky COM0. Další parametry komunikace jsou neměnné (NO parity, 8bitová data, 1 stoPbIt).

Parametr **maxcall** udává maximální počet pokusů o spojení (opakování spojení) před ohlášením Err=1.

Parametr **actcode** slouží pro potvrzování aktivního hlášení, defaultně má hodnotu 0.

Pokud využijeme aktivní hlášení, dojde k jeho potvrzení tehdy, naplníme-li **actcode** hodnotou ze vstupu **Call code**.

Parametr **initbcall** (init before call) způsobí inicializaci modemu před každým zavoláním.

Parametry přístupné přes tlačítko DIALOG MODULU:

Parametr **init\_string\_0** - první konfigurační řetězec pro inicializaci tel. modemu před zavoláním.

Parametr **init\_string\_1** - druhý konfigurační řetězec pro inicializaci tel. modemu před zavoláním.

Parametr **init\_string\_2** - třetí konfigurační řetězec pro inicializaci tel. modemu před zavoláním.

Poznámka:

Pokud chceme využít pro definici konfiguračního řetězce i další řádky (2. a 3.), musí být vždy předchozí řádek plný, tj. musí mít nadefinováno právě 32 znaků.

Obsahuje-li projekt modul **modem**, je po překladu linka COM0 regulátoru PL2 inicializována na komunikační rychlost dle hodnoty parametru **comspeed** modulu **modem**. Pokud chcete dále pracovat s FREDem (ladící režim, další překlad) musíte upravit komunikační rychlost ve FREDovi na stejnou (volbou PROMOSline2 / Parametry komunikace).

### **Zapojení telefonního modemu na COM0 centrály:**

- nastavte hodnotu parametru **pwtout** modulu **sysmon** na doporučených 500 ms

Od verze 2.5 firmware CCPU-02 byly upraveny po zkouškách vnitřní parametry modulu **modem**:

- timeout na potvrzení aktivního hlášení byl prodloužen z 10 na 30 sec  
- doba od vyslání aktivního hlášení do přijetí odpovědi  
- řešeno naplněním parametru **actcode** z dispečinku hodnotou na vstupu **Call code**

- timeout na inicializaci prodloužen na 1000 ms  
- tím se odstraní hlášení výstupu **Err = 1** při inicializaci modemu s **pwtout = 500 ms**

- u modemu propojit RTS+CTS, kabel jen RxD, TxD a GND

- doporučený init pro modem v ČR:

```
ATE0V1X3&C1&R1&D0H0S0=3&W0&Y0
```

zkontrolujte význam dle manuálu modemu:

ATE0 = echo ne

..V1 = modem vypisuje slovní (ne číselnou) formou

..X3 = modem detekuje obsazovací tón, vhodné pro naše podmínky

..&C1 = signál DCD odpovídá skutečnému stavu nosné linky

..&R1 = modem ignoruje RTS, CTS je aktivní

..&D0 = ignoruje DTR

..H0 = zavěsí

..S0=3 = zápis do registru S0 číslo 3, tj. po kolikátém zvonění modem vyzvedne

..&W0 = nové nastavení modemu do paměti č. 0

..&Y0 = konfigurace modemu bude probíhat z paměti č. 0, tj. podle tohoto nastavení

Poznámka:

Poslední úpravy pro verzi 2.5 si vyžádaly rovněž úpravu ovladače pro ControlWeb.

Ovladače pro ControlWeb jsou ke stažení na našich stránkách [www.elsaco.cz](http://www.elsaco.cz).

Od verze FW 3.000 byl timeout pro navázání spojení prodloužen na 90s. Jedná se o dobu odvytočení telefonního čísla do funkčního spojení. Připojení k internetu může trvat dost dlouho, protože se v tomto případě do této doby započítává i doba přípravy PPP protokolu.

### **GPRS modem - zapojení, konfigurace**

- GPRS modem je potřeba připojit na COM 0 centrály

- signály RTS a DTR modemu je třeba propojit s pinem 7 konektoru kanálu COM0, kde je +5V
- je potřeba správně nastavit tyto parametry na konfigurační WWW stránce CCPU:
  - IP a port vzdálené stanice (dispečinku)
  - maska podsítě = 0.0.0.0
  - Active call = PPP/UDP/GPRS
  - na uživatelském jménu ani heslu nezáleží
- je potřeba správně nastavit modul **modem** v projektu
  - comspeed = 38400
  - maxcall = 10
  - ackcode = 0
  - initbcall = 1
- toto nastavení není závazné, pouze doporučené:
  - řetězec pro volání = ATD \*99\*\*\*1#
  - nebo ATD \*99#
  - (podle typu modemu)
  - inicializační řetězec = ATE0 +CGDCONT=1,"IP","internet.t-mobile.cz"
  - popř. ještě další AT konfigurace podle typu modemu
  - místo "internet.t-mobile.cz" zadat APN svého mobilního operátora
- po naběhnutí systému (a přihlášení modemu do sítě GSM, doporučujeme vložit prodlevu několik sekund) nastavit callcode = 1
  - Pozor, v tomto případě se **callcode** po navázání spojení neshazuje, ani se podle něj nenastavuje parametr **ackcode**, jako u standardního modemu.
  - Modem je trvale aktivní a tudíž připojen.
  - Pokud nemá GPRS modem vlastní pevnou IP adresu, je vhodné aktivovat modul **emergcall**, aby měl dispečink trvalou informaci o současné IP adrese stanice na GPRS modemu. Doporučená perioda aktivních hlášení z modulu **emergcall** je 2 minuty.
  - Modul **emergcall** pak může i předávat informaci o stavu systému, takže není nutný pooling ze strany dispečinku.
  - Výstup ! (hw reset) modulu **modem** je možné použít pro HW restart modemu.
  - Je na několik sekund aktivován zhruba 25 sekund před pokusem o znovunavázání spojení v případě jeho výpadku. Napájení modemu je v tomto případě připojeno přes binární výstup, ovládaný výstupem ! (hw reset) modulu **modem**.

### POZOR

- pro GPRS komunikaci musí být v systémovém menu centrály nastavena komunikace na COM4

*Poznámka:*

*Součástí ZIPu FW centrály (na webu ke stažení) je projekt IPGPRSM.TXT.*

## 3.3 svm840/940



Modul **svm840/940** je určen pro komunikaci se stejnojmenným měřičem tepla firmy ABB.

Komunikace probíhá na lince **COM1/2** stejným principem jako u modulu **serialcomm**.

Naměřené hodnoty jsou hodnoty reálných čísel na výstupech:

- **Oper.h** (čas správného chodu v hodinách)
- **Error.h** (čas chybného chodu v hodinách)
- **Energie** (v GJ)
- **Objem** (m<sup>3</sup>)
- **Objem All** (m<sup>3</sup>)

- **Tvstup** (°C)
- **Tvrat** (°C)
- **dT** (°C)
- **Výkon** (kW)
- **Průtok** (m<sup>3</sup>/h)

Standardní parametry modulu: **rychlost a prioritá**.



Obr. 42 Značka **svm840/940** na ploše

Parametr **COMnr** určuje svou hodnotou číslo komunikačního kanálu, na kterém komunikační relace proběhne.

- 1 - pro COM1
- 2 - pro COM2

Nesmí být použit kanál, který používají moduly MPC (pro ProfiBus, tj. **pwpb\_main**, **pwpb\_rx** a **pwpb\_tx**).

Parametr **address** určuje adresu měřiče.

Je-li **address=254** (broadcast), je vysílání zprávy v síti určeno všem stanicím.

Jediný vstup (pin vlevo dole) slouží k blokadě komunikace s měřičem tepla. Blokace nastane při stavu log. 1. Pokud bude na tomto vstupu log. 0, bude probíhat komunikace vždy po průchodu programovou smyčkou. Tzn., že bude-li parametr **rychlost=1**, bude probíhat komunikace jednou za 3 vteřiny.

*Poznámka:*

*Při chybné komunikaci s modulem **svm840/940** (objekt 15 s instancí 1) je vysláno chybové hlášení, zobrazitelné na displeji při použití modulu **sysmon** v projektu:*

**Er: 002, 001@015, 001**

*Chybové hlášení "zmizl" po správné komunikační relaci s měřičem.*

## 3.4 cf50



Modul **cf50** je určen pro komunikaci se stejnojmenným měřičem tepla CF50 firmy Allmess.



Obr. 41 Značka modulu **cf50** na ploše

Od verze SW 1.6 z 29.3.2002 je modul **cf50** rozšířen o komunikaci se starším měřičem tepla Integral firmy Allmess.

V tomto případě jsou funkční jako výstupy modulu pouze hodnoty pro objem, energii, teplotu vstupní a teplotu vratu, ostatní hodnoty jsou nedefinované.

Modul sám pozná, jestli je připojen starý nebo nový typ měřiče.



Komunikace probíhá na lince COM1/2 (dáno parametrem **COMnr**) stejným principem jako u modulu **serialcomm**.

*Poznámka:*

Měřič tepla CF50 je vybaven komunikačním výstupem M-Bus.

Centrála CCPU-02 však rozhraní M-Bus přímo nemá, proto ji je třeba jej doplnit převodníkem na M-Bus nebo použít centrálu CCPU-03 či CCPU-21 s vestavěným převodníkem M-Bus.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **COMnr** určuje svou hodnotou číslo komunikačního kanálu, na kterém komunikační relace proběhne.

- 1 - pro COM1
- 2 - pro COM2

Nesmí být použit kanál, který používají moduly MPC (Profibus).

Parametr **address** určuje adresu měřiče.

Jediný vstup (pin vlevo dole) slouží k blokadě komunikace s měřičem tepla. Blokadě nastane při stavu log.1. Pokud bude na tomto vstupu log. 0, bude probíhat komunikace vždy po průchodu programovou smyčkou. T.zn., že bude-li parametr rychlost=1, bude probíhat komunikace jednou za 3 vteřiny.

Naměřené hodnoty měřičem jsou předány jako hodnoty reálných čísel na výstupech:

- **Energie**
- **Objem** (m<sup>3</sup>)
- **T vstup** (°C)
- **T vrat** (°C)
- **d T** (°C)
- **Výkon** (kW)
- **Průtok** (m<sup>3</sup>/h)

*Poznámka:*

Při chybné komunikaci s modulem cf50 (objekt 21 s instancí 1) je vysláno chybové hlášení, zobrazitelné na displeji při použití modulu sysmon v projektu:

**Er** : 002 , 001@021 , 001

Chybové hlášení "zmizí" po správné komunikační relaci s měřičem.

### 3.5 danfoss



Modul **danfoss** je univerzální modul pro komunikaci s měřiči tepla protokolem Mbus.

Mimo měřiče DANFOSS INFOCAL5 lze zatím také napojit měřič SVM-F3 firmy ABB a měřič 2WR5 firmy Siemens.



Obr. 43 Značka modulu danfoss na ploše

Komunikace probíhá na lince COM1/2 (dáno parametrem **COMnr**) stejným principem jako u modulu **serialcomm**. Komunikační rychlost je 300 Bd.

Komunikační linka musí být použita typu RS422 (ne RS485) nebo speciální, např. přes převodník Danint (ten pouze přes COM2).

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **COMnr** určuje svou hodnotou číslo komunikačního kanálu, na kterém komunikační relace proběhne.

- 1 - pro COM1
- 2 - pro COM2

Nesmí být použit kanál, který používají moduly MPC (Profibus).

Parametr **address** určuje adresu měřiče.

Je-li **address**=254 (broadcast), je vysílání zprávy v síti určeno všem stanicím.

Jediný vstup (pin vlevo dole - **Stop**) slouží k blokadě komunikace s měřičem tepla. Blokadě nastane při stavu log. 1. Pokud bude na tomto vstupu log. 0, bude probíhat komunikace vždy po průchodu programovou smyčkou. T.zn., že bude-li parametr **rychlost**=1, bude probíhat komunikace jednou za 3 vteřiny.

Pravý horní výstup **Error** indikuje chybu komunikace s měřičem.

Naměřené hodnoty měřičem jsou předány jako hodnoty reálných čísel na výstupech:

- **Oper.h**  
(čas správného chodu v hodinách)  
neplatné pro měřič 2WR5, kterému tento údaj chybí
- **Energie**  
(energie v MWh)
- **Objem** (m<sup>3</sup>)  
[defaultně nastaven]
- **Objem All**  
[alternativní objem, u MT ABB F3 je to objem i když se neměří teplo]
- **T vstup** (°C)
- **T vrat** (°C)
- **d T** (°C)
- **Výkon** (kW)
- **Průtok** (m<sup>3</sup>/h)
- **Count1** (hodnoty čítače z SVM-F3)
- **Count2** (hodnoty čítače z SVM-F3)

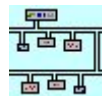
*Poznámka:*

Při chybné komunikaci s modulem danfoss (objekt 30 s instancí 1) je vysláno chybové hlášení, zobrazitelné na displeji při použití modulu **sysmon** v projektu:

**Er** : 002 , 001@030 , 001

Chybové hlášení "zmizí" po správné komunikační relaci s měřičem.

### 3.6 pwpb\_main



Modul **pwpb\_main** umožňuje MPC komunikaci po sériové lince RS485 protokolem Epsnet/Profibus. Každá stanice může být master nebo slave.

Formát používaných zpráv je kompatibilní s automaty Teco a Promos. Tento modul definuje společné parametry pro všechny přijímací (**pwpb\_rx**), vysílací (**pwpb\_tx**) a



sériové (SAIOi, SAIOo, SBIOi, SBIOo, SBI, SBO, SFCPUi, SFCPUo) moduly..



Obr. 44 Značka **pwpb\_main** na ploše

Standardní parametry modulu: **rychlost** a **priorita**.

Ostatní parametry modulu:

- **kanal** - číslo použitého sériového kanálu (Com0 až 2).  
Při použití kanálu 0 je nutno použít převodník RS232/RS485 s automatickým řízením RTS a nastavit parametr **prodleva** podle časové konstanty řízení RTS.
- **comrychlost** - komunikační rychlost v Baudech
- **parita** - zvolená parita:  
0=žádná  
1=lichá  
2=sudá (pro epsnet se volí běžně)
- **mezera** - dovolená meziznaková mezera (počet znaků mezera)
- **prodleva** - dodatečná prodleva mezi dokončením příjmu a zahájením vysílání (v ms)
- **odezva** - maximální doba čekání na odpověď (v ms)
- **maxtoken** - maximální doba držení tokenu (v ms), 0=slave
- **adresa** - adresa této stanice

*Pokud je víc stanic typu master (aktivní stanice), pak definujte adresy těchto stanic od 1 v souvislém bloku (např. 1, 2, 3, 4 - pak parametr **maxadresa** = 4).*

*Adresy pasivních stanic (slave) pro tento režim, volte opět v souvislém bloku následně po adresách stanic master. Pár adres můžete ponechat volných pro případné rozšíření o další stanice master v budoucnu (např. 8, 9, 10, 11.....25, 26, 27).*

*Pokud zvolíte režim monomaster (jediná stanice master v síti), zvolíte adresu stanice master = 1 (a parametr **maxadresa** = 1) !*

*Adresy pasivních stanic (slave) pro režim monomaster, volte v souvislém bloku následně, tj. 2, 3, 4, 5...!*

- **maxadresa** - nejvyšší zkoumaná adresa pro předání tokenu.

Při **maxadresa** = 1 se nepředává token => monomaster, zrychlí se komunikace. Používejte např. tehdy, pokud k centrále připojujete pouze sériové (SAIOi, SAIOo, SBIOi, SBIOo, SBI, SBO, SFCPUi, SFCPUo) moduly.

Poznámka:

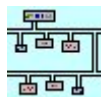
- *nezpomínejte na zakončení sběrnice !*
- *centrály CCPU-21 a CCPU-03 mají zapojeny u COMu1 i zakončovací odpory*
- *schéma a další podrobnosti k RS-485 viz Technický manuál Jednotky s připojením na sběrnici RS-485*

Příklad parametrů pro modul **pwpb\_main** a stanici **monomaster** (při napojení sériových modulů k centrále, vycházejte z toho, že sériové moduly mají z výroby pro epsnet nastavenou rychlost 38400 Bd a sudou paritu):

```
priorita = 0
rychlost = 2
kanal = 1
comrychlost = 38400
parita = 2
```

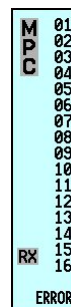
```
mezera = 0
prodleva = 10
odezva = 100
maxtoken = 500
adresa = 1
maxadresa = 1
```

### 3.7 pwpb\_rx



Modul **pwpb\_rx** zajišťuje v MPC komunikaci příjem šestnácti čtyřbajtových údajů prostřednictvím linky RS485 protokolem Epsnet/Profibus.

Komunikační parametry jsou určeny modulem **pwpb\_main**.



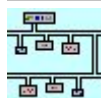
Obr. 45 Značka **pwpb\_rx** na ploše

- **01 až 15** - datové výstupy
- **Error** - výstup indikující jedničkou zastarání dat

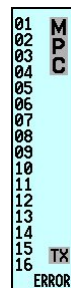
Standardní parametry modulu: **rychlost** a **priorita**.  
Ostatní parametry modulu:

- **perioda** - maximální časová perioda (ms), se kterou mají být data obcerstvována. Parametr je typu longinteger. Pokud je cílová stanice zavčas nezveřejní sama a tato stanice je MASTER, tak si o ně sama požádá.
- **adresa** - od které stanice má data přijímat/požadovat
- **blok** - ze kterého bloku
- **offset** - od jakého offsetu
- **delka** - délka přijímaného bloku v bajtech, obvykle 64
- **initout1..16** - inicializační hodnoty výstupu

### 3.8 pwpb\_tx



Modul **pwpb\_tx** zajišťuje v MPC komunikaci zveřejnění bloku šestnácti čtyřbajtových údajů prostřednictvím linky RS485 protokolem Epsnet/Profibus.



Obr. 46 Značka **pwpb\_tx** na ploše

Komunikační parametry jsou určeny modulem **pwpb\_main**.

- **01 až 15** - datové vstupy

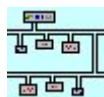
- **Error** - výstup indikující jedničkou výpadek v odesílání

Standardní parametry modulu: **rychlost a priorita**.

Ostatní parametry modulu:

- **perioda** - jak často data odesílat (v ms). Pokud je stanice SLAVE (maxtoken=0), tak pouze odpovídá na žádosti. Parametr je typu longinteger.
- **adresa** - na jakou adresu odesílat. Obvykle 127 (broadcast).
- **blok** - který blok
- **offset** - na jakém offsetu
- **delka** - kolik bajtů, obvykle 64

### 3.9 pwpbp\_rx



Modul **pwpbp\_rx** zajišťuje v MPC komunikaci příjem šestnácti čtyřbajtových údajů prostřednictvím linky RS485 s protokolem Epsnet/Profibus.

Komunikační parametry jsou určeny modulem **pwpb\_main**.

M	01
P	02
C	03
	04
	05
	06
	07
	08
	09
	10
	11
	12
	13
PAR	14
RX	15
	16
	ERROR

obr. 47 Značka modulu **pwpbp\_rx** na ploše

Modul **pwpbp\_rx**:

- obdoba modulu **pwpb\_rx**
- možnost přesměrování komunikace na konkrétní parametr libovolného modulu
- funkční pro čtení i zápis
- např. pro změnu parametrů z dispečinku (posun/sklon u ekvitermu apod.)
- **01..15** - datové výstupy
- **Error** - výstup indikující jedničkou zastarání dat

Standardní parametry modulu: **rychlost a priorita**.

**Ostatní parametry modulu:**

- **perioda**  
maximální časová perioda, se kterou mají být data obcerstkována (v ms). Parametr je typu longinteger. Pokud je cílová stanice zavčas nezveřejní sama a tato stanice je MASTER, tak si o ně sama požádá.
- **adresa**  
od které stanice má data přijímat/požadovat
- **blok**  
ze kterého bloku
- **offset**  
od jakého offsetu
- **delka**  
délka přijímaného bloku v bajtech, obvykle 64

Definice konkrétního parametru, na který se chcete dostat pomocí určení čísla objektu, instance a položky (hodnoty určité z okna *Informace o modulu*, u kterého chcete daný parametr číst/zapisovat):

- **obj0..15**  
číslo objektu
- **inst0..15**  
číslo instance
- **item0..15**  
číslo položky

### 3.10 EmergCall



Modul **emergcall** je modul nouzového hlášení při ztrátě komunikace s dispečinkem.



Obr. 48 Značka **EmergCall** na ploše

Modul detekuje, zda probíhá komunikace s dispečinkem a pokud tato neproběhla po dobu danou parametrem **DispComm** (hodnota je zadávána v minutách), snaží se modul **emergcall** periodicky vysílat zprávu po dispečerské síti do stanice s adresou danou parametrem **DestAddress**.

Perioda vysílání takové zprávy je dána hodnotou parametru **CallPeriod** (hodnota je zadávána opět v minutách).

Zpráva je vysílána na adresu danou parametrem **DestAddress**.

Zpráva obsahuje pouze data, jejichž počet je dán parametrem **NumObjects**.

Jsou vysílána data ze vstupů **Tx0** až **Tx7** (reálná čísla).

*Příklad:*

*Pro NumObjects = 2*

*Vysílány jsou data z prvních dvou vstupů modulu emergcall, tj. ze vstupů Tx0 a Tx1.*

Zpráva je vysílána do vzdálené stanice (na adresu dle **DestAddress**) do objektů (modulů), které jsou určeny parametry **Object0..7**, **Instance0..7** a **Item0..7**.

*Příklad:*

*Chtějme vyslat hodnoty reálných čísel ze vstupů Tx0 a Tx1 do stanice s adresou 2 a do modulů constf o instancích 1 a 2.*

Pak bude **DestAddress=2** (adresa stanice, do které chci posílat zprávu) a **NumObjects=2** (pro první dva vstupy Tx0 a Tx1).

Modul **constf** má číslo objektu 122, proto **Object0=122** i **Object1=122**.

**Instance** (jakési pořadí modulu constf v projektu) bylo zadáno, proto **Instance0=1** a **Instance1=2**.

Hodnota modulu **constf** je dána parametrem **hodnota**, který má položku 4 (viz okno *Informace o modulu*).

Proto **Item0=4** i **Item1=4**.

Obdobně jsou nastaveny defaultní hodnoty modulu **emergcall**.

V případě detekce výpadku komunikace dispečinku je rovněž aktivován výstup **DispFault**.

Je-li však parametr **CallPeriod=0** a je detekován výpadek komunikace dispečinku, stane se sice výstup **DispFault** aktivním (=1), ale zpráva není vysílána.

Toho lze v případě výpadku dispečerské komunikace využít k přepínání hodnot (pomocí modulu **amux** a signálu **DispFault**) mezi hodnotami z dispečinku a hodnotami z nouzové komunikace.

Standardní parametry modulu: **rychlost a priorita**.

V projektu může být použit pouze jeden modul **emercall**.

### 3.11 comport

**COM** Modul **comport** je určen pro koncentraci dat, určených pro přenášení na dispečink.

**PORT**



Obr. 50 Značka **comport** na ploše

mění-li se např. částečně projekt ve FREDovi, nemění komunikovaná data svůj kód (čísla objektů, instancí a itemů). To proto, že data komunikovaná pomocí modulu **comport**, přebírají hodnoty označení objektu, instance a položek z modulu **comport**, na který jsou napojena. Komunikace pomocí tzv. QQ relace se pak dotazuje na objekt modul **comport** a jeho instanci a jeho itemy (položky), které změněny nebyly.

Pomocí modulu **comport** a modulu **SCREEN** lze také vyřešit možnost editace hodnoty jak z ovládacího panelu, tak i z dispečinku. Viz obrázek dále.

Standardní parametry modulu: **rychlost a priorita**.

Parametry **initout0..15** určují výchozí hodnoty výstupních pinů **0..15** po překladu.

Parametry **input0..15** kopírují hodnoty vstupů tohoto modulu.

Určeno pro ladění a pro přenos na dispečink.

### 3.12 cdafo55

Modul **cdafo55** slouží pro vyčítání čítačů měřiče tepla DANFOS INFOCAL5.



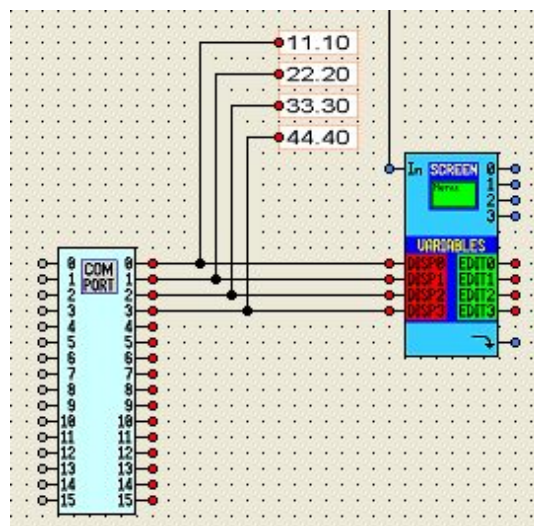
Obr. 51 Značka **cdafo55** na ploše

Komunikace probíhá na lince COM1/2 (dáno parametrem **COMnr**) stejným principem jako u modulu **serialcomm**.

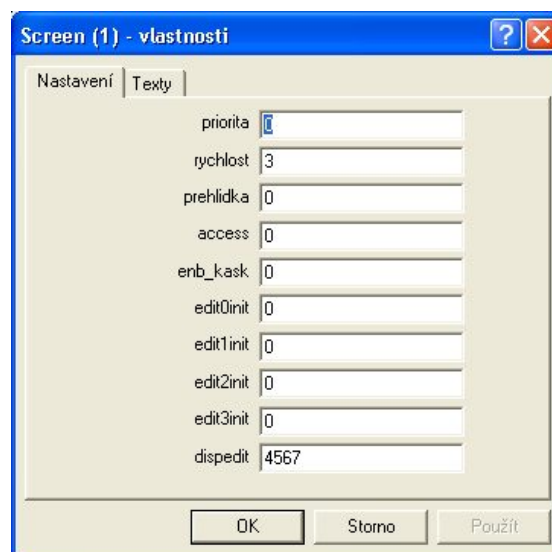
Komunikace využívá nestandardních (výrobce definovaných) služeb sběrnice M-Bus.

Proto lze tento modul použít výhradně s měřičem tepla Infocal5.

Komunikační linka musí být použita typu RS422 (ne RS485) nebo převodník Danint (ten pouze přes COM2).



Obr. 49 Příklad pro možnost editace hodnoty jak z ovládacího panelu, tak i z dispečinku



Obr. 52 Nastavení parametru **dispedit** u modulu **SCREEN**

Standardní parametry modulu: **rychlost a priorita**.

Parametr **COMnr** určuje svou hodnotou číslo komunikačního kanálu, na kterém komunikační relace proběhne.

- 1 - pro COM1
- 2 - pro COM2

Nesmí být použit kanál, který používají moduly MPC (ProfiBus).

Parametr **address** určuje adresu měřiče.

Je-li **address=254** (broadcast), je vysílání zprávy v síti určeno všem stanicím.

Výstupy **CounterA** a **CounterB** předávají načtené hodnoty z čítačů měřiče.

Výstup **Error = 1** při výpadku komunikace s měřičem, jinak = 0.

Jediný vstup (pin vlevo dole - **Stop**) slouží k blokaci komunikace s měřičem tepla.

Blokace nastane při stavu log. 1.

Pokud bude na tomto vstupu log. 0, bude probíhat komunikace vždy po průchodu programovou smyčkou.

T.zn., že bude-li parametr **rychlost=1**, bude probíhat komunikace jednou za 3 vteřiny.

### 3.13 sdo



Modul **sdo** umožňuje vyslat či přijmout speciální data do a z CAN periferií (s firmware verze 2.0 a vyšší).

(SDO = Service Data Object = služební/doplňková/pomocná data).



Obr. 54 Značka **sdo** na ploše

Data, která chceme vyslat přivedeme na vstup **Out** modulu **sdo**.

Data, která přijmeme, odebíráme do projektu na výstupu **In** modulu **sdo**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **address** určuje adresu cílového modulu (např. CKDM-12 s adresou 2).

Parametr **init** určuje, zda se přenos má uskutečnit pouze jednou, při inicializaci (= 1), nebo periodicky za chodu systému. Tento periodický přenos lze dočasně zastavit aktivací vstupu **Disable**.

Parametr **read/write** určuje, zda se mají data číst nebo zapisovat (0 pro read=čtení, 1 pro write=zápis).

Parametry **index**, **subindex** a **length** určují cílový datový objekt v jednotce CAN. Příslušné hodnoty parametrů pro zadaná/požadovaná data lze nalézt v dokumentaci CAN periferií (určeno pro specialisty).

### 3.14 gsm



Modul **gsm** je základním modulem pro vysílání a příjem SMS zpráv pomocí GSM modemu TC35. Tento modul není obsažen ve FW malé centrály (s FLASH 256 kB).



Obr. 53 Značka **gsm** na ploše

Modul **gsm** umožňuje i standardní funkce GSM modemu, tj. napojení dispečinku, nahrávání projektu či ladící režim ve FREDovi (QQ relace).

*Poznámka:*

*Přepínání standardní funkce / SMS režim dělá modem pomocí signálu DSR, který musí být připojen na signál CTS kanálu COM0 (funguje pouze na tomto kanálu).*

*Do inicializačního řetězce GSM modemu je pak nutné přidat AT příkaz AT&S1, který nastaví pro modem požadovanou funkci signálu DSR.*

*(Vyzkoušeno pouze simulováním úrovně na CTS - přepíná komunikace GSM SMS a QQ. CTS=0=SMS režim; CTS=1=QQ relace).*

*Reinicializace modemu GSM je prováděna automaticky každou hodinu, a to pouze tehdy, nejsou-li na modul kladeny další požadavky.*

Modem připojíme na jeden ze tří sériových komunikačních kanálů centrály. Na tento kanál nesmí být připojeno další zařízení, musí sloužit pouze pro GSM modem. Číslo tohoto kanálu 0, 1 nebo 2 zadáme do parametru **comnr** modulu **gsm**.

Parametr **access** omezuje přístup pomocí SMS zpráv. Popis použití - viz dále.

Standardní parametry modulu: **rychlost** a **priorita**.

#### Vysílání SMS

Pokud na vstup **Send** (typu integer) modulu **gsm** dojde k přechodu z 0 do 1, vyšle modul **gsm** SMS zprávu, která obsahuje první čtyři řádky chybových hlášení z modulu **sysmon**.

Spojení se navazuje dle parametrů modulu **gsm**.

To platí, pokud je parametr **lines** = 1. Parametr **lines** určuje počet vysílaných čtveřic (obsah 4 řádků displeje = "obrazovek") chybových hlášení z modulu **sysmon** SMS zprávou. Parametr **lines** může nabývat hodnot 1..3, což znamená, že lze vyslat SMS zprávou až 12 chybových hlášení. Proto nezapomeňte text chybových hlášení volit tak dlouhý, aby nepřesáhl max. povolenou délku SMS zprávy.

Pokud na vstup **Send** přivedeme hodnotu 2, je SMS poslána na telef. číslo poslední příchozí SMS.

Čtyři různá tel. čísla lze zadat do parametrů **tlfnum1..4**.

Hodnotou parametru **numselect** určujeme, které z nadefinovaných tel. čísel bude zvoleno pro navázání spojení.

Od verze firmware 2.502 přibyl vstup **Num**. Hodnota na vstupu **Num** se přičítá k hodnotě parametru **numselect** a tak lze z projektu ovlivňovat, se kterým tel. číslem (podle **tlfnum1..4**) má být navazováno telefonní spojení.

Pokud je **numselect+Num=0 až 3**, dochází k volbě tel. čísla pro vyslání SMS na mobilní telefon (**numselect+Num=0** pro **tlfnum1**, **numselect+Num=1** pro **tlfnum2**, **numselect+Num=2** pro **tlfnum3** a **numselect+Num=3** pro **tlfnum4**).

Pokud je **numselect+Num=4 až 7**, bude vyslána SMS na e-mail, proto příslušné tel. číslo v parametru **tlfnum1..4** musí být číslem operátora mobilní sítě (např. 4616 pro Paegas), (**numselect+Num=4** pro **tlfnum1**, **numselect+Num=5** pro **tlfnum2**, **numselect+Num=6** pro **tlfnum3** a **numselect+Num=7** pro **tlfnum4**).

Další parametry jsou přístupné z dialogového okna modulu **gsm** po kliknutí na tlačítko DIALOG MODULU.

Inicializační řetězec pro GSM modem lze zadat do parametru **init\_string0**. Implicitně je uveden odladěný řetězec pro modem TC35 a síť Paegas:

**ATE0 +CMGF=1**

Do parametru **header** lze uvést řetězec, společný a předcházející všem tel. číslům, zadaných v **tlfnum1..4** (např. +420 pro tel. spojení v ČR). Pokud chceme SMS odesílat e-mailem, musí být **header** prázdný (nesmí obsahovat ani znak).

Do parametru **email** zadáváme e-mailovou adresu ve formátu **#grunc1\*elsaco.cz** (odladěno pro Paegas).

Adrese předchází znak # a namísto @ je užit znak \*.

#### Přijímání SMS

Text přijaté SMS zprávy je odeslán na výstup **Code** modulu **gsm**.

Ten je určen pro napojení vstupu/ů **Code** modulu/ů **sms** pro jeho další zpracování.

Pokud napojíme label na pin **Code** modulu **gsm**, lze v ladícím režimu FREDa sledovat přijímaná data GSM modemem. Pokud však nebude modul **gsm** propojen s modulem **sms**, uvidíme pouze kódy prvních 4 znaků.



Parametr **smsslow** slouží pro zpomalení testování příchozí SMS (od verze FW 3.008).

0 - každé 3 sekundy

(tak to bylo do verze FW 3.007 včetně)

1, 2, 3, ... -> 6, 9, 12, ... sekund

U modemu MC 39i bylo zjištěno, že pokud se testuje příchozí SMS v okamžiku, kdy modem zpracovává příchozí datové volání, tak modem zavěsí. Nastavením **smsslow** na rozumně vyšší hodnotu se pravděpodobnost tohoto jevu výrazně omezí.

## QQ relace typu SMS

Bez nutnosti další konfigurace vykonává modul **gsm** i tzv. QQ relace. Tzn., že pokud modul **gsm** zjistí přijatou SMS zprávu, která formátem odpovídá protokolu QQ (ve verzi protokolu pro SMS, který je níže popsán), tak odešle SMS zprávu, obsahující vyžádaná data (nebo potvrzení zápisu), a to opět ve formátu QQ ve verzi pro SMS.

Tato komunikace je určena pro komunikaci dispečink - PL2 přes GSM modemy.

požadavek:

**Q123,001,004Q**

přečte hodnotu konstanty integer long (z modulu constl) (objekt č. 123), s instancí 1, položka (item) 4, tj. hodnota viz okno *Informace o modulu*

odpověď:

**Q123,001,004Q 555Q**

ve zprávě, která přijde jako odpověď, je zopakován požadavek, následuje mezera a předávaná hodnota 555, nakonec Q

požadavek:

**Q122,001,004.Q**

přečte hodnotu konstanty float (z modulu constf) objekt č. 122, s instancí 1, položka (item) 4, tj. hodnota viz okno *Informace o modulu*

odpověď:

**Q123,001,004. 678.9Q**

ve zprávě, která přijde jako odpověď, je zopakován požadavek, následuje mezera a předávaná hodnota 678.9, nakonec Q

**Q123,001,004 556677Q**

zapiše hodnotu integer long 556677 do modulu constl s instancí 1

**Q123,001,004 556677Q**

odpověď

**Q122,001,004 753.3Q**

zapiše hodnotu reálného čísla (float) 753.3 do modulu constf s instancí 1

**Q122,001,004 753.3Q**

odpověď

## Obecný popis QQ relace, verze SMS

### požadavek pro čtení

První znak Q je povinný, pak postupně následují trojčíselná čísla (bez mezery a oddělená čárkou), a to číslo objektu, číslo instance objektu a číslo itemu (položky) objektu.

Pak (opět bez mezery) je zpráva ukončena znakem Q pro čtení integer long hodnoty nebo je ukončena znaky .Q (tečka a kvé) pro čtení float hodnoty.

### požadavek pro zápis

První znak Q je povinný, pak postupně následují trojčíselná čísla (bez mezery a oddělená čárkou), a to číslo objektu, číslo instance objektu a číslo itemu (položky) objektu.

Pak následuje mezera a po ní hodnota čísla pro zápis. Pokud číslo obsahuje . (tečku), považuje se za typ float, jinak za typ integer long.

Pak opět bez mezery znak Q pro ukončení zprávy.

Přijatá zpráva může obsahovat až 9 libovolných znaků (kromě Q) před prvním Q, které jsou přeskočeny - hledá se první Q.

To je umožněno z důvodu možnosti přijímat SMS zprávy z internetu (taková brána přidává za začátek SMS zprávy: WWW1/1: ).

V odpovědi na požadavek je vždy prvních 13 znaků požadavku. Vracena hodnota čísla je zarovnána na celkový počet 7 znaků (zleva doplní mezery).

Jakákoliv chyba syntaxe, přístupových práv či parametrů způsobí ignorování zprávy bez další kvitace.

Čtení je možné vždy, pro zápis platí stejné restrikce jako pro zápis parametrů z FREDA nebo dispečinku.

Zapsáním příslušné hodnoty do parametru **numselect** modulu **gsm** lze měnit číslo, na které modul **gsm** a moduly **sms** posílají zprávy. Odpověď na změnu tohoto parametru přijde ještě na původní číslo, další odpovědi na nové číslo. Změna čísla se projeví bez nutnosti restartu systému.

### Omezení přístupových práv pomocí parametru **access**

Z telefonních čísel, uvedených v parametrech **tlfnum 1..4**, není přístup omezen.

Z ostatních čísel je možný přístup omezen podle nastavení parametru **access** podle tabulky:

acces	QQwrite	SMSwrite	QQread	SMSread
0	ano	ano	ano	ano
1	ne	ano	ano	ano
2	ne	ne	ano	ano
3	ne	ne	ne	ano
4	ne	ne	ne	ne

V případě pokusu o zakázaný zápis PL2 vrátí SMS s nezměněnou hodnotou.

V případě pokusu o zakázané čtení PL2 nevrátí nic.

*Poznámka:*

*Před připojením nového modemu TC35 k centrále je nutné pomocí AT příkazu nastavit číslo SMS centra. Toto nastavení si modem provždy zapamatuje.*

*Pro Paegas to je:*

**AT+CSCA="+420603052000"<enter>**

Moduly **gsm** a **sms** defaultně odpovídají na telef. číslo poslední příchozí SMS zprávy.

V případě, že se pošle QQ SMS, **gsm** modul odpoví na to číslo, odkud SMSka přišla.

Pokud někdo pošle povel nebo požadavek vyčtení na modul **sms** a jeho parametr **noanswer**=0, pak modul odpoví na číslo, ze kterého ten požadavek přišel.

V tomto případě je generována odpověď automaticky a není potřeba aktivovat vstup **Send**.

Příchozí telefonní číslo je filtrováno parametrem **access** modulu **gsm**.

### 3.15 sms

**SMS** Modul **sms** vyhodnocuje SMS zprávu, přijatou modulem **gsm**, dokáže na ni odpovědět, ale dokáže i samostatně odeslat SMS zprávu s daty, přivedenými na své vstupy **In0..7**. Tento modul není obsažen ve FW malé centrály (s FLASH 256 kB).



Obr. 55 Značka **sms** na ploše

Zprávy odesílá na mobil nebo e-mail podle aktuálního nastavení parametrů v modulu **gsm**.

Parametr **comnr** musí být nastaven stejně jako u modulu **gsm**, který může být v projektu pouze jeden.

Modulů **sms** lze použít v jednom projektu víc, všechny vstupem **Code** lze napojit na výstup **Code** modulu **gsm**.

Parametr **initcmd** určuje výchozí hodnotu výstupu **Cmd** modulu **sms** po překladu.

Parametr **noanswer** způsobí, že modul **sms** neodpovídá na příchozí SMS zprávu.

Výstup **Msg** je aktivován po přijetí SMS zprávy až do doby než je odeslána odpověď

- pokud **noanswer**=0: po dobu trvání jedné programové smyčky
- pokud **noanswer**=1: do odeslání řízeného vstupem **Send**

Parametrem **toint**=1 lze nadefinovat, aby výstup modulu **sms** byl typu integer (celočíslný).

Standardní parametry modulu: **rychlost** a **priorita**.

#### Vyhodnocení přijaté SMS zprávy

Text přijímané SMS zprávy na vstupu **Code** je porovnáván s parametrem **reccode** (pátý řádek v dialogovém okně modulu **sms**, přístupný po stisku tlačítka DIALOG MODULU). Porovnání je činěno detailně, a to i s ohledem na malá / velká písmena! Rovněž včetně případného předznačení ve zprávách, zaslaných e-mailů - např. WWW1/1:Data ...

Po porovnání (od počátku) přijatého textu SMS zprávy s řetězcem v parametru **reccode** je spojení ukončeno.

Pokud je v přijaté SMS zprávě, která odpovídá řetězci v **reccode**, za tímto řetězcem (nanejvýš po mezeře) číslo, je jeho hodnota přenesena na výstup **Cmd** modulu **sms** (a

přepíše se i hodnota **initcmd**, aby po vyp/zap centrály zůstala hodnota na výstupu **Cmd** jako po posledním příjmu).

Pokud je v přijaté SMS zprávě od počátku řetězec, odpovídající parametru **reccode**, za kterým nenásleduje číslo - nastaví se **Cmd**=0.

Pokud při porovnání nedošlo k rovnosti, další spojení se nenavazuje.

Pokud při porovnání došlo ke shodě, je navázáno spojení podle aktuálních parametrů modulu **gsm**.

Spojení lze navázat kdykoliv pomocí vstupu **Send** - jako u modulu **gsm**, přechodem vstupního signálu z 0 do 1.

Pokud na vstup **Send** přivedeme hodnotu **2**, je SMS poslána na telef. číslo poslední příchozí SMS.

Výstup **Cmd** lze kdykoliv vynulovat aktivním signálem na vstupu **Reset**.

*Poznámka:*

*Podle hodnot na výstupu **Cmd** lze porovnáním proti očekávané hodnotě regulátor dálkově ovládat SMS zprávami.*

#### Odeslání dat SMS zprávou

Po navázání spojení (viz výše) je odeslána textová SMS zpráva. Text lze zadat do parametrů **text\_string0..3**, které jsou přístupné v dialogovém okně modulu po stisku tlačítka DIALOG MODULU.

Co parametr, to řádek, řádek může mít max. 20 znaků.

Do textu lze umístit hodnotu ze vstupů **In0..7**, a to pomocí formátovacího řetězce ze znaků # (např.: ###.##).

První formátovací řetězec slouží pro zobrazení hodnoty ze vstupu **In0**, druhý ze vstupu **In1**, ...

Pokud je formát výpisu překročen, jsou cifry nahrazeny ve výpisu znakem E, např. EEE.EE.

Pokud je v textu zprávy použit znak \$, je ve zprávě nahrazen výpisem On/Off, a to podle stavu příslušného bitu na vstupu **In7**.

První užití znaku \$ vypisuje stav bitu 0, druhé bitu 1, ... Maximálně po 16 bitů!

*Poznámky:*

*Moduly **gsm** a **sms** defaultně odpovídají na telef. číslo poslední příchozí SMS zprávy.*

*V případě, že se pošle QQ SMS, **gsm** modul odpoví na to číslo, odkud SMSka přišla.*

*Pokud někdo pošle povel nebo požadavek vyčtení na modul **sms** a jeho parametr **noanswer**=0, pak modul odpoví na číslo, ze kterého ten požadavek přišel.*

*V tomto případě je generována odpověď automaticky a není potřeba aktivovat vstup **Send**.*

*Příchozí telefonní číslo je filtrováno parametrem **access** modulu **gsm**.*

### 3.16 dcf

**DCF77** Modul **dcf** umožňuje připojení přijímače časových značek DCF 77 typu UDCF/T S-RPC-ASCII.



Obr. 56 Značka modulu **dcf** na ploše

Přijímač musí být napojen na COM1 centrály, a to takto:

cannon DCF77	COM1 CCPU
5	9
4	10
2	2
ostatní piny nezapojeny	

Výstup **Err** indikuje stavem 0 přítomnost platného časového údaje.

Vzhledem k tomu, že pro vyhodnocení správnosti přijatého časového údaje se používá jeho porovnání s předchozím časovým údajem, a modul DCF77 požaduje nejméně 5 za sebou jdoucích časových údajů pro uznání jejich správnosti, tak po restartu trvá i při správné funkci přijímače okolo 5 minut (časový údaj je vysílán jednou za minutu) od resetu, než výstup **Err** spadne do nuly.

RTC v CCPU-02 je pak nastaven vždy po restartu (samozřejmě až je přijat první platný časový údaj) a pak jednou za den.

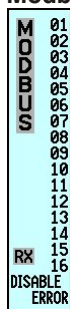
Nastavován je pouze údaj minut a sekund, proto datum a hrubý čas musí být i tak přednastaven obsluhou.

### 3.17 ModbusRX



Modul **ModbusRX** je určen pro komunikaci (příjem zprávy s hodnotami do projektu) na sériovém kanále centrály protokolem modbus.

Modbus je protokol MASTER - SLAVE. Master je implementován moduly **ModbusTX**, **ModbusRX**.



Obr. 57 Značka modulu **ModbusRX** na ploše

Moduly ModbusRX a ModbusTX jsou určeny např. pro komunikaci s PLC Modicon, mohou komunikovat s periferními jednotkami SBI/SBO/SBIO/SAIO, přepnutými na protokol modbus.

Slave může být dále např. měřič tepla, ale i jiná centrála CCPU-02 nebo CCPU-03, protože na jejím hlavním kanále (defaultně COM0 - lze přepnout v systémovém menu na jiný) je implementován protokol modbus (slave). Popsáno u centrály.

Pracuje stejným způsobem jako modul **serialcomm**, tj. vyšle požadavek na data a čeká na odpověď.

Vstup **disb** umožňuje přerušit periodickou komunikaci (parametr **rychlost** = 3, tj. po vyslání požadavku na data testuje po 100 ms zda přišla odpověď s daty nebo zda vypršel timeout).

Výstupy **0..15** nabízí přijatá data z registrů zprávy (dle pravidel protokolu modbus a podle hodnot dalších parametrů) do projektu.

Podle odpovědi (o správném příjmu či vypršení timeoutu) je nastavován výstup **error**. Pokud jsou přijatá data v pořádku => **error** = 0, jinak **error** = 1.

Timeout na odpověď je pevně nastaven na 1000 ms, meziznaková mezera max. 3 znaky.

Standardní parametry modulu: **rychlost** a **priorita**

### Parametry modulu ModbusRX

**function** = kód funkce modbusu

- 1 / 2  
bitová data (bináry) jsou z registrů zprávy skládána po 16ti postupně do výstupů modulu
- 3 / 4  
data typu word nebo long  
pokud **datatype** = 0, bere data jako 16tubitová (default protokolu modbus) a skládá je po jednom do výstupů  
pokud **datatype** = 1, bere data jako 32tubitová (2x16 bitů) a skládá je po 32ti bitech do výstupů modulu

**offset** - viz definice modbus

- pro bitové zprávy udává bitový offset
- jinak offset číselných dat, např. pro :  
0 požaduje data zprávy od nultého registru  
3 požaduje data zprávy od 3. registru

**number** - viz definice modbus

- význam těchto parametrů záleží na volbě kódu funkce
- počet registrů ve zprávě pro číselná data  
- např. při **function**=3, **datatype**=0 a požadavku na příjem 16tubitových dat do všech výstupů **0..15** bude **number**=16 (registr ve zprávě je 16tubitový)  
- pokud při **function**=3 je **datatype**=1, pak určuje, že pro příjem 32tubitových dat do všech výstupů **0..15** bude potřeba dvojnásobek 16tubitových registrů ve zprávě, proto **number**=32
- pro bitové zprávy udává počet bitů

**datatype** = modifikátor datového typu

- u modulu **ModbusRX** má význam pouze při **function** = 3 nebo 4 (číselné vstupy)
- určuje zda na výstupech **0..15** jsou 16 / 32tubitová data

**baud** = definuje rychlost sériového kanálu

- v rozsahu 300...38 400 Bd, na kterém je modbus provozován

**parity** = definuje paritu

- 0, 1, 2 = bez parity, lichá, sudá

**comno** = použitý sériový kanál centrály pro modbus

- 0, 1, 2

**address** = adresa cílové stanice

*Poznámka:*

*U modulů ModbusRX, ModbusTX i ModbusTXC parametr **datatype** = 2 způsobí prohození wordu čtyřbajtových dat jak pro čtení, tak pro zápis.*

*datatype = 0 -> 16bitová data*

*datatype = 1 -> 32bitová data*

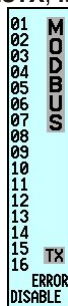
*datatype = 2 -> 32bitová data word swapped*

### 3.18 ModbusTX



Modul **ModbusTX** je určen pro komunikaci (vyslání zprávy s hodnotami z projektu) na sériovém kanále centrály protokolem modbus.

Modbus je protokol MASTER - SLAVE. Master je implementován moduly **ModbusTX**, **ModbusRX**.



Obr. 59 Značka modulu **ModbusTX** na ploše

Moduly **ModbusRX** a **ModbusTX** jsou určeny např. pro komunikaci s PLC Modicon, mohou komunikovat s periferními jednotkami SBI/SBO/SBIO/SAIO, přepnutými na protokol modbus.

Slave může být dále např. měřič tepla, ale i jiná centrála CCPU-02 nebo CCPU-03, protože na jejím hlavním kanále (defaultně COM0 - lze přepnout v systémovém menu na jiný) je implementován protokol modbus (slave). Popsáno u centrály.

Modul **ModbusTX** pracuje obdobným způsobem jako modul **serialcomm**, tj. vyšle zprávu a čeká na odpověď o správném příjmu.

Vstup **disb** umožňuje přerušit periodickou komunikaci (parametr rychlost = 3, tj. po odeslání dat testuje po 100 ms zda přišla odpověď OK nebo zda vypršel timeout).

Vstupy **0..15** nabízí odesílaná data z projektu do registrů zprávy dle pravidel protokolu modbus a podle hodnot dalších parametrů.

Podle odpovědi (o správném příjmu či vypršení timeoutu) je nastavován výstup **error**.

Timeout na odpověď je pevně nastaven na 1000 ms, mezníková mezera max. 3 znaky.

Standardní parametry modulu: **rychlost** a **priorita**

### Parametry modulu ModbusTX

**function** = kód funkce Modbusu

- 5 = zápis jedné binární hodnoty z 1. vstupu
- 6 = zápis jedné 16bitové hodnoty z prvního vstupu modulu
- 15 = zápis skupiny binárů, bere z každého vstupu modulu po 16ti binárech a rovná je za sebou do posílané zprávy
- 16 = zápis skupiny čísel do registrů posílané zprávy pokud datatype = 0, bere hodnoty z každého vstupu 16bitové, jinak 32bitové a rovná je za sebou do registrů posílané zprávy

**offset** - viz definice Modbus

- pro bitové zprávy udává bitový offset
- jinak offset číselných dat, např. pro :  
0 ukládá data zprávy od nultého registru  
3 ukládá data zprávy od 3. registru

**number** - viz definice modbus

- význam těchto parametrů záleží na volbě kódu funkce
- počet registrů ve zprávě pro číselná data  
- např. při **function** = 16, **datatype** = 0 a požadavku na odeslání 16bitových dat ze všech vstupů **0..15** bude **number** = 16 (registr ve zprávě je 16bitový) pokud při **function** = 16 je **datatype** = 1, pak určuje,

že na vstupech **0..15** jsou data 32bitová a bude potřeba dvojnásobek 16bitových registrů ve zprávě, proto **number** = 32

- pro bitové zprávy udává počet bitů
- datatype** = modifikátor datového typu
- u modulu **ModbusTX** má význam pouze při **function** = 16 (číselné vstupy)
  - určuje zda na vstupech **0..15** jsou 16 / 32bitová data
- baud** = definuje rychlost sériového kanálu
- v rozsahu 300...38 400 Bd, na kterém je modbus provozován
- parity** = definuje paritu
- 0, 1, 2 = bez parity, lichá, sudá
- comno** = použitý sériový kanál centrály pro modbus
- 0, 1, 2
- address** = adresa cílové stanice

Poznámka:

U modulů **ModbusRX**, **ModbusTX** i **ModbusTXC** parametr **datatype** = 2 způsobí prohození wordu čtyřbajtových dat jak pro čtení, tak pro zápis.

datatype = 0 -> 16bitová data

datatype = 1 -> 32bitová data

datatype = 2 -> 32bitová data word swapped

## 3.19 M-Bus



Modul **M-Bus** je určen pro dálkové odečty měřidel spotřeby energií na sběrnici MBUS.



Obr. 58 Značka modulu **M-Bus** na ploše

Modul **M-Bus** funguje tak, že vyšle [jako master] požadavek na sériový kanál centrály (podle parametru **COMnr** na adresu měřiče [slave] dle parametru **address**), pak prohledá přijatou zprávu, a pokud v ní nalezne data s kódem hodnoty a dat, odpovídající příslušným parametrům **difo..7** a **vifo..7**, přepoše je na odpovídající výstupy **Value 0..7** modulu **M-Bus**.

### Parametry modulu M-Bus

- standardní parametry **priorita** a **rychlost**
- parametr **address** je adresa cílového zařízení v síti M-Bus, pro testování může být 254 (pak komunikuje se zařízením s libovolnou adresou)
- parametrem **comspeed** se nastavuje komunikační rychlost v Bd, defaultně 300 Bd (od verze FW 3.007)
- parametr **COMnr** je číslo komun. kanálu (0..2) na centrále [300Bd, 8 bit data, even (sudá) parita, jeden stoPBIt]
- parametr **CI** = hodnota kódu CI, implicitně 81 (nastavení CI pole v konfigurační zprávě)
- parametry **vifo..7** jsou kódy hodnot, které mají být zobrazeny na výstupech **Value 0..7**  
VIF = Value Information Field



(informace o hodnotě, určuje, jestli je příslušná hodnota Energie, Objem, hodiny, prutok....)

- parametry **dif 0..7** jsou kódy dat, která mají být zobrazena na výstupech **Value 0..7**

DIF = Data Information Field

(informace o typu dat, určuje, jestli je příslušná hodnota zakódována jako binární, integer, float, 1 byte, 2 byte, 4 byte...)

**Podrobněji viz specifikace protokolu MBUS na [www.M-Bus.com](http://www.M-Bus.com).**

*Např. DIFE, VIFE rozšiřující kódy předchozích - určují, jestli se jedná o maxima, průměry, hodnoty v chybách, odlišují jednotlivé tarify atd.*

Pokud je zadaný **vif0..7 = 0**, do příslušného výstupu **Value 0..7** se uloží první ještě nezobrazená hodnota, obsazená v přijaté zprávě.

Pokud tedy jsou všechny parametry **vif0..7 = 0**, modul **M-Bus** "zobrazí na svých výstupech" prvních 8 hodnot z přijaté zprávy bez ohledu na jejich **dif0..7** kódy.

Vstup **Stop** umožňuje zablokovat komunikaci a výstup **Error** indikuje aktivním stavem, že modul **M-Bus** nedostal platnou odpověď na vyslaný požadavek.

Parametr **Configure\_format** (přístupný přes Dialog modulu) umožňuje zadat kód výběru dat ze vzdáleného zařízení v hexadecimálním formátu.

Zde se zadává vlastně přímo obsah zprávy, kterou přepneme měřič do požadovaného režimu vyčítání.

Podrobnosti - vždy v dokumentaci / manuálu měřiče.

#### **Příklad:**

**Configure\_format = 087E** určuje, že má být provedeno normální čtení všech hodnot, které dává měřič na M-Busu v režimu normálního čtení.

Číslo 08 je dif, určující, že jde o příkaz k přepnutí typu čtení, a číslo 7E je vif, mající význam " všechny kódy vif" (wildcard).

Např. **Configure\_format = C83F05C83F14** udává, že se mají vyčíst všechny hodnoty objemu a energií (dif = C83F -> všechny tarify a třídy data, vif = 05, 14 - objemy, energie).

Parametry **vif** a **dif** umožňují zadat kódy dat a hodnot až se třemi rozšiřujícími položkami (**dife** a **vife**), je však je třeba zadávat v decimálním tvaru.

#### **Příklad:**

**dif = 8C** a **dife = 01** -> hexadecimálně = 0x018C (kód dat 12ciferný BCD, číslo pozice pro uložení = 2) bude dekadicky 396.

Pro vyčtení tohoto údaje musí být tedy parametr **dif** roven 396.

#### **Poznámka:**

Z jednoho měřiče můžeme vyčíst hodnoty i použitím více modulů M-Bus v projektu.

Počet modulů M-Bus v projektu je SW omezen na 250, nezapomínejte však na HW omezení kanálu MBUS na centrále CCPU-03 (cca 40 měřičů, pokud je jejich HW dle normy).

#### **Poznámka:**

V ČR upravuje MBUS protokol norma ČSN/EN 1434-3. Data jsou odečítána elektronicky po datovém kabelu,

všechny měřiče jsou adresovány individuálně (odečet jednotlivých měřidel nebo celé soustavy). Standardizace protokolu umožňuje uživatelům kombinaci systémů s měřidly různých výrobců. Sběrnice umožňuje připojení rádo-  
vě stovek měřidel s celkovou délkou vedení v řádech kilometrů. V současné době je sběrnice MBus využívána při odečtech vodoměrů, měřičů tepla a chladu, plynomě-  
rů, elektroměrů. Systém je investičně nenákladný, mon-  
tážně jednoduchý neboť pro přenos dat je využíván  
jednopárový kabel bez stínění a současně velmi odolný  
proti vnějšímu rušení.

### **Základní charakteristika**

- ⇒ vysoký počet měřidel připojitelný do systému
- ⇒ překonání velkých vzdáleností
- ⇒ měřidla různých výrobců
  - jednotný komunikační protokol
- ⇒ vysoká odolnost proti vnějším vlivům
  - indukce, kapacita
- ⇒ možnost rozšiřování sítě o další měřidla
- ⇒ nízké investiční náklady (1 pár bez polarity, bez stínění)
- ⇒ minimální spotřeba energie měřidel
- ⇒ přijatelná rychlost přenosu dat a vysoká integrita dat - data jsou využívána pro fakturaci !

### **Princip činnosti**

Jedná se o hierarchický systém s řídicí jednotkou a podří-  
zenými přístroji vzájemně propojených datovým kabelem. Komunikace probíhá prostřednictvím dotazů řídicí jednotky a odpovědí podřízených přístrojů. Rozsah předávaných in-  
formačních závisí na typu měřiče jím zpracovávaných dat.

Dotaz – řídicí jednotka => přístroj

- \* napěťová modulace
- \* 1 = 36V; 0 = 24 V

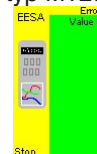
Odpověď – přístroj => řídicí jednotka

- \* proudová modulace
- \* 1 = konst. proud do 1,5 mA
- \* 0 = zvýšení o 11 - 20 mA

## **3.20 EESA**



Modul **EESA** umožňuje komunikovat s měřiči tepla fy EESA, typ MT200.



Obr. 60 Značka modulu **EESA** na ploše

Vstup **Stop** umožňuje pozastavit komunikaci. Výstup **Error** indikuje chybu komunikace. Výstup **Value 0** obsahuje float hodnotu vyčtenou z měřiče tepla.

Parametr **address** určuje adresu měřiče tepla. Parametr **COMnr** definuje číslo kanálu na CCPU,

parametr **baud** rychlost komunikace v Bd.

Parametr **code** určuje kód vysílané zprávy.

Parametr **mode** umožňuje modifikovat způsob komunikace.

Poznámka:

Zatím parametr **mode=1** způsobí, že je obrácena polarita parity ve vysílané zprávě pro případ, že by význam termínu *log1* a *log0* v popise byl opačný (na RS485 nebo RS232 je definován stav aktivní a neaktivní [space-mark], kde aktivní stav je malé, resp. záporné napětí, takže se to dá snadno zaměnit).

Standardní parametry modulu: **rychlost** a **priorita**.

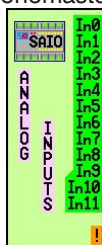
### 3.21 SAIOi



Modul **SAIOi** umožňuje vyčítat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) naměřené analogové hodnoty z jednotky SAIO-11/12.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa=1**).



Obr. 61 Značka SAIOi na ploše

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwppb\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa jednotky SAIO-11/12, nastavená na přepínačích jednotky.

Na výstupech **In0..11** jsou do projektu odevzdávány vyčtené (případně i zlinearizované) hodnoty (float, real.čís.) naměřených veličin jednotkou SAIO-11/12.

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.22 SAIOo



Modul **SAIOo** umožňuje ovládat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) analogové výstupy jednotky SAIO-11/12.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa=1**).



Obr. 62 Značka SAIOo na ploše

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwppb\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa jednotky SAIO-11/12, nastavená na přepínačích jednotky.

Na vstupy **Out0..5** jsou z projektu přiváděny hodnoty v rozsahu 0,0 až 100,0 (float) jako požadavek na nastavení analogového výstupu v rozsahu 0 až 100%.

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.23 SBI



Modul **SBI** umožňuje vyčítat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) logické stavy vstupů (binární vstupy) z jednotky SBI-11/12.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa=1**).



Obr. 63 Značka SBI na ploše

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwppb\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa jednotky SBI-11/12, nastavená na přepínačích jednotky.

Na výstupech **In0..15** jsou do projektu odevzdávány vyčtené a filtrované hodnoty binárních vstupů X0..15 jednotky SBI-11/12, na výstupech **Cnt0..7** jsou do projektu odevzdávány vyčtené hodnoty čítačů impulzů na vstupech X0..7 jednotky SBI-11/12.

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.24 SBO



Modul **SBO** umožňuje ovládat po sériové lince centrály pomocí tzv. MPC komunikace

(protokolem epsnet/profibus) releové výstupy jednotky SBO-11/12.



Obr. 67 Značka SBO na ploše

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa** = 1).

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwpbp\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa jednotky SBO-11/12, nastavená na přepínačích jednotky.

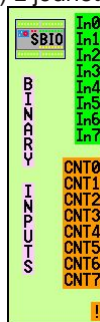
Logické úrovně ze vstupů **00..12** ovládají relé jednotky SBO-11/12, ze zbylých vstupů pak ovládají každou čtvrtou LED na panelu modulu.

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.25 SBIOi



Modul **SBIOi** umožňuje vyčítat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) logické stavy vstupů (binární vstupy) z jednotky SBIO-11/12.



Obr. 64 Značka SBIOi na ploše

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa** = 1).

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwpbp\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa jednotky SBIO-11/12, nastavená na přepínačích jednotky.

Na výstupech **In0..7** jsou do projektu odevzdávány vyčtené a filtrované hodnoty jednotlivých vstupů X0..7 jednotky SBIO-11/12, na výstupech **Cnt0..7** jsou do projektu odevzdávány vyčtené hodnoty čítačů impulzů na vstupech X0..7 jednotky SBIO-11/12.

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.26 SBIOo



Modul **SBIOo** umožňuje ovládat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) releové výstupy jednotky SBIO-11/12.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa** = 1).

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwpbp\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).



Obr. 65 Značka SBIOo na ploše

Parametr **adresa** - komunikační adresa jednotky SBIO-11/12, nastavená na přepínačích jednotky.

Logické úrovně ze vstupů **Out0..7** ovládají relé jednotky SBIO-11/12.

Výstup ! svým stavem = 1 indikuje zastarání dat.

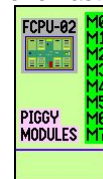
### 3.27 SFCPUi



Modul **SFCPUi** umožňuje vyčítat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) logické stavy vstupů z jednotky FCPU-02A.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa** = 1).



Obr. 66 Značka SFCPUi na ploše

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwpbp\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa sériové jednotky FCPU-02A, nastavená na přepínačích jednotky.

Na výstupech **MO..7** jsou do projektu odevzdávány vyčtené a filtrované hodnoty jednotlivých osmic binárních vstupů jednotky FCPU-02A, přičemž jednotlivé osmice binárních vstupů jsou na jednotce FCPU-02A osazeny HW moduly logických vstupů (I/O piggy PBI-11/12 + případně HW osazení vstupními svorkovými moduly XBI-11). Osmice vyčtených binárních vstupů je do projektu předávána na výstupech **MO..7** jako "bajt" - integer hodnota.



Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.28 SFCPUo



Modul **SFCPUo** umožňuje ovládat po sériové lince centrály pomocí tzv. MPC komunikace (protokolem epsnet/profibus) logické výstupy jednotky FCPU-02A.

V projektu musí být použit modul **pwpb\_main** pro nastavení MPC.

Pokud jsou k centrále připojeny jen sériové moduly, nastavte centrálu jako monomaster (**maxadresa** = 1).



Obr. 69 Značka SFCPUo na ploše

V projektu lze použít zároveň i ostatní moduly - **pwpb\_rx**, **pwpb\_tx** a **pwpbp\_rx**.

Standardní parametry modulu: **rychlost** a **priorita**.

Parametr **perioda** - maximální časová perioda, se kterou mají být data občerstvována (v ms, long).

Parametr **adresa** - komunikační adresa sériové jednotky FCPU-02A, nastavená na přepínačích jednotky.

Na vstupy **MO..7** přivádíme integer hodnoty - požadované stavy jednotlivých osmic binárních výstupů (relé na XBO) jednotky FCPU-02A.

Jednotlivé osmice binárních výstupů jsou na jednotce FCPU-02A osazeny HW moduly logických výstupů (I/O piggy PBO-11/12 + případné HW osazení releovými moduly XBO-11).

Výstup ! svým stavem = 1 indikuje zastarání dat.

### 3.29 ReTran



Modul **ReTran** slouží k přenesení komunikačních zpráv (zejména typu QQ) z hlavního kanálu na jiný COM centrály (pro vyslání na hlavní kanál jiné jiné stanice - a zase QQ zpráv)



Obr. 68 Značka ReTran na ploše

Na hlavním kanálu nesmí být nastavena žádná jiná komunikace (např. MPC). Modul **ReTran** není určen pro retranslování komunikace MPC ani dalších jiných, jako jsou M-Bus, serialcomm, SMS, atd.

Nejedná se o protokolovou retranslaci, ale o jakési "programové nahrazení" komunikačního převodníku (např. RS-485), který je v centrále k dispozici.

Stanice (centrála) předá všechny zprávy, které nejsou určeny pro ni, z hlavního kanálu dál na zvolený retranslační kanál. Relace na retranslačním kanále = QQ protokol.

*Například:*

*Použitím modulu ReTran v projektu prochází dispečerské komunikační zprávy, přijímané modemem na COMu0*

(hlavní kanál) centrály na zvolený COM1 RS-485 a pak na další stanice, které jsou na tuto linku napojeny.

Nebo máme navolen jako hlavní kanál COM4 (komunikace PPP/TCP/IP/GPRS/GSM) a na něm modem GPRS - tak je centrála napojena na dispečink. Použijeme-li v projektu pro tuto centrálu modul ReTran, lze tak na COMu1 RS-485 získat převedené relace z hlavního kanálu, které nejsou určeny pro tuto centrálu, na COMu1 jako tzv. QQ protokol pro síť dalších stanic na komunikační lince RS-485.

Jediný výstup **count** modulu **ReTran** udává počet retranslací (integer long).

Pomocí parametrů modulu **ReTran** lze zvolit číslo COMu pro retranslační kanál, přenosovou rychlost retranslačního kanálu a timeout.

Parametr **rychlost** ponechejte = 3, protože ovlivňuje zpoždění retranslace.

Hodnota parametru **Baudrate** udává přenosovou rychlost na retranslačním kanále. Defaultně 38400 Bd.

Hodnota parametru **Timeout** udává maximální dobu mezi znaky v ms, defaultně 0.

Hodnota parametru **COMnr** udává číslo kanálu (COMu), na který budou relace převedeny (retranslační kanál).

Standardní parametry modulu: **rychlost** a **priorita**.

### 3.30 PortMix



Modul **PortMix** slouží pro komunikaci s modulnetem.

"Tváří se" obdobně jako modul **COMPOR**, až na to, že data přesouvá podle svých vnitřních parametrů přímo z/do projektu, na itemy.



Obr. 70 Značka PortMix na ploše

Funkční jak pro zápis, tak pro čtení.

Pomocí parametrů **target0..63** směřujeme data.

Tento parametr zapisujeme jako řadu cifer ve složení: type/datový modifikátor, objekt, instance, item/položka

- datový modifikátor
  - =0 mapuje jeden byte binárně bez konverze
  - =1 mapuje data jako modulnet int, čili převede na float a vydělí 10
  - =2 mapuje data jako obyčejný int, převede na float bez dělení 10
- číslo objektu  
viz okno *Informace o modulu*
- číslo instance = pořadí téhož objektu v projektu
- item  
viz okno *Informace o modulu*

Například mapujeme jednu editační hodnotu ze **SCREENu** jako modulnet int, pak jeden byte z **konstanty** a dále dva byte z výstupu **ekvitermu** jako modulnet int.

Parametrizace:

1013001058

-> 1 datový modifikátor = modulnet int,

013 = objekt 13, modul SCREEN,

001 = instance 1,



058 = item 58, edit0init

123001003

-> datový modifikátor není, čili byte,

123 = objekt 123, modul constl,

001 = instance 1,

003 = item 3, výstupní hodnota1

1151001003

-> 1 datový modifikátor = modulnet int,

151 = objekt 151, ekviterm

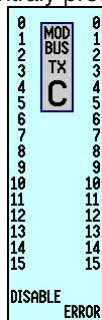
001 = instance 1

003 = item 3, vystup

### 3.31 ModbusTXC



Modul **ModbusTXC** je určen pro komunikaci (vyslání zprávy s hodnotami z projektu) na sériovém kanále centrály protokolem modbus.



Obr. 71 Značka modulu ModbusTXC na ploše

**Modbus TXC** má oproti modulu **ModbusTX** navíc 16 výstupů, které obsahují hodnoty kontrolního čtení zapisovacích hodnot.

Oproti modulu **ModbusTX** má ale jiný způsob komunikace.

Modul **ModbusTXC** může mít parametr **function** buďto 15 (pro binární data) nebo 16 (pro int/float data).

Modul pracuje tak, že periodicky, pomocí zpráv typu 1 (pro binární hodnoty) nebo 3 (pro 16 bitové registry) čte stav dat ve vzdálené jednotce.

V případě, že zjistí nesouhlas hodnot mezi vnitřními registry, indikovaných výstupy **0..15**, načtenými ze vzdálené jednotky, a hodnotami na vstupech modulu, provede zápis těch hodnot, které se liší, a to po jedné postupně, nejdříve pro vstupy s nižším pořadovým číslem.

Význam parametrů modulu **ModbusTXC** je jinak stejný jako u modulu **ModbusTX**.

*Poznámka:*

*U modulů ModbusRX, ModbusTX i ModbusTXC parametr datatype = 2 způsobí prohození wordu čtyřbajtových dat jak pro čtení, tak pro zápis.*

*datatype = 0 -> 16bitová data*

*datatype = 1 -> 32bitová data*

*datatype = 2 -> 32bitová data word swapped*

Standardní parametry modulu: **rychlost** a **priorita**.

### 3.32 Inet



Modul **Inet** je určen pro údržbu spojení na internetu.

Pomocí příkazu PING zkouší a vyhodnocuje funkčnost spojení, pokud spojení selže, dokáže restartovat modem a znovu se připojit.



Obr. 72 Značka modulu Inet na ploše

Vstup **Offline** náběžnou hranou odpojí PPP proces.

Odpojení trvá zhruba 2,5 minuty a není indikováno!

*(vstup Call code modulu modem nemusí být po aktivaci vstupu Offline nulován, po uplynutí 8 minut se pak PPP proces zase připojí)*

Parametr **tcpidle** definuje v sekundách, jak dlouho musí být spojení na TCP nečinné, aby se centrála CCPU odpojila.

Parametr **pingperiod** udává periodu PINGů na remip v sekundách.

*(remip se zadává z www konfigurační stránky centrály)  
pingperiod=0 vypíná příkaz PING*

Výstup "!" je aktivní pokud nepřijde do 20 sec na PING odezva.

*Poznámka:*

*Na nastavovací http stránce centrály zadávejte masku podsítě 0.0.0.0 pro připojení pomocí PPP.*

*Aby byly dodrženy výše uvedené časy, musí být modul Inet ve vteřinové programovací smyčce (parametr rychlost=2).*

*Na nastavovací stránce www nastaven z výroby typ spojení udp/ppp/gprs.*

### 3.33 IEC870



Modul **IEC870** je určen pro komunikaci na sériovém kanále centrály protokolem IEC 870-5-101. V projektu může být použit pouze jeden modul **IEC870**.



Obr. 73 Značka modulu IEC870 na ploše

Implementaci protokolu IEC 870-5-101 ve firmware centrály pro FREDa realizuje modul **IEC870**. Datová napojení jsou realizována pomocí modulů **HDBmain** a **HDBa / HDBb** pro vstupy, modulem **Comport** s instancí 1 pro výstupy. Centrála je z pohledu komunikace protokolem IEC870-5-101 slave.

Od verze FW 3.017 lze pro komunikaci protokolem IEC 870-5-101 využít další podpůrné moduly typu **IECin** (IECinSP, IECinDP, IECinMI a IECinMF). Ty rozšiřují možnosti původního protokolu, proto musí být v projektu k nim použit základní modul IEC870.

Pokud se do projektu přidá jeden nebo více modulů **IECin**, neberou se data pro komunikace z modulu **HDB** (dosavadní funkce), ale z modulů **IECin**. Pokud ne, zůstává dále popsaná funkce modulu **IEC870** zachována.

Když se přidají do projektu moduly **IECin**, pak parametry **offsetxxx** modulu **IEC870** nemají v tomto uspořádání význam a data se berou z modulů **IECin**.

Výstup **Err** je funkční od FW 3.017 a slouží společně pro chybová hlášení jak základního modulu **IEC870**, tak rozšiřujících modulů typu **IECin**.

Výpadek komunikace (a nenulový **Err**) nastane, když:

- od Mastera nepřijde do určité (parametrizovatelné) doby telegram (výzva)
- od Mastera přijde výzva s FC 9 (dotaz na stav linky) - začátek „navazování spojení“
- přijdou chybné (nesmyslné) telegramy od Mastera
- obnovení komunikace nastane po proběhnutí úvodní sekvence (FC 9-11-0-0; nebo volněji po obnovení výzev s FC=10 nebo FC=11)

- výstup **Err**
  - = 0 - bez chyby
  - = 1 - přijetí zprávy link status
  - = 2 - nesouhlasí délka zprávy
  - = 3 - chybný kontrolní součet
  - = 4 - poslední znak není 0x16
  - = 6 - timeout komunikace

## Parametry:

### **linkaddress**

určuje link address (linkovou adresu) komunikace

### **COMnr**

číslo komunikačního kanálu CCPU

### **baud**

rychlost komunikace v Bd

(Od 200 / 300 Bd; v projektu lze použít jen jednu z 200 / 300 i pro ostatní moduly. Nelze tedy v jednom projektu použít zároveň modul IEC870 na 200Bd a zároveň modul Mbus na 300Bd.)

### **parity**

parita (0-bez, 1-lichá, 2-sudá)

### **timeout**

maximální mezníková mezera na příjmu

### **linkaddresssize**

délka linkové adresy v bytech (1 nebo 2)

### **commonaddress**

adresa společná pro všechny komunikační objekty

### **commonaddresssize**

délka common address v bytech (1 nebo 2)

### **causeoftrnsize**

délka cause of transmission (příčiny přenosu) v bytech (1 nebo 2)

### **infoobjaddrsize**

délka pole information object address

(adresy objektu s informací), 1 nebo 2 nebo 3 byty

### **offsetsp**

parametr, určující posun information object address pro jednobitové objekty

(M\_SP\_NA\_1 a M\_SP\_TB\_1) takto:

information object address = offsetsp + index,

kde index je parametr index modulu HDBa,

realizující dané datové napojení

### **offsetdp**

parametr, určující posun information object address pro dvoubitové objekty

(M\_DP\_NA\_1 a M\_DP\_TB\_1),

stejným způsobem, jako v předchozím případě

### **offsetbs**

parametr, určující posun information object address pro objekty typu bitstring

(M\_BO\_NA\_1 a M\_BO\_TB\_1), stejným způsobem

### **offsetana**

parametr, určující posun information object address pro objekty analogových hodnot

(M\_ME\_NA\_1 a M\_ME\_NC\_1 a M\_ME\_TD\_1

a M\_ME\_TF\_1), stejným způsobem

### **offsetcmd**

parametr, určující posun information object address pro commands (povely)

(C\_SC\_NA\_1, C\_DC\_NA\_1, C\_SC\_TA\_1, C\_DC\_TA\_1)

takto:

comport\_output\_num = information object address +  
+ offsetcmd - 3000

kde comport\_output\_num je číslo výstupu modulu comport s instancí 1, na který jsou povely přenášeny

### **offsetanahist**

parametr, určující posun information object address pro objekty analogových hodnot

(M\_ME\_TD\_1), určených pro vyčtení z hist. banky

generálním dotazem, stejným způsobem jako

u offsetana.

### **comtout**

tento parametr přibyl až ve verzi FW 3.017

komunikační timeout

ten v počtech průchodů smyčky (default N x 100ms)

udává, za jak dlouhou nečinnost komunikace se

generuje výstup Err

## Datová napojení - výstupy

Je možné přenášet až 16 povelů. Tyto povely se přenášejí na výstupy modulu **comport** s instancí 1. Information object address pro tyto povely je možné nastavit podle popisu parametru **offsetcmd** modulu **IEC870**. Na výstup modulu **comport** se kopíruje přímo byte SCO, popř. DCO povelu. Jeho další zpracování (hlavně rozlišení, jestli jde o trvalý povel, nebo pouze pulzní) je nutné udělat pomocí aplikace ve FREDovi. Pokud povel obsahuje časovou značku, je povolena maximální tolerance časové značky přijatého povelu 2 minuty oproti aktuálnímu času RTC v centrále.

Při větší odchylce časové značky povelu není povel vykonán.

## Datová napojení - vstupy

Vstupy jsou datově napojeny na vstupy modulů **HDBa** / **HDBb**. Pokud jsou vstupy čteny povelom general interrogation (celkový dotaz), jsou komunikovány hodnoty přímo ze vstupů modulu **HDBa** / **HDBb**, bez časové značky. Pokud jsou vstupy čteny pomocí cyclic polling (cyklické „sbírání hlasů“), je komunikován při každém cyklu jeden vzorek z HDB, a to včetně časové značky ve formátu CP56Time2a.

Datový typ vstupů je určen parametrem **index** modulu **HDBa** takto:

**index** =

1000..1999

vyhrazeno pro sbírání dat z modulů HDBb.

Data jsou interpretována jako jednobitová.

2000..2999

vyhrazeno pro sbírání dat z modulů HDBa.

Data jsou interpretována jako analogová float.

3000..3999

vyhrazeno pro sbírání dat z modulů HDBa.

Data jsou interpretována jako dvoubitová.

4000..4999

vyhrazeno pro sbírání dat z modulů HDBa.

Data jsou interpretována jako 32bitový bitstring.

5000..5999

vyhrazeno pro sbírání dat z modulů HDBa.

Data jsou interpretována jako analogová signed short

(-32768 .. + 32767).

6000..6999

vyhrazeno pro sbírání dat z modulů HDBa.  
Data jsou interpretována jako jednobitová.  
7000..7999  
vyhrazeno pro sbírání dat z modulů HDBb.  
Data jsou interpretována jako analogová signed short  
(-32768 .. + 32767). Aktuální hodnota těchto dat není  
k dispozici v generálním dotazu, a tato data ani  
neregenerují události pro vyčtení v cyklickém pollingu.  
Místo toho jsou určena k vyčítání historických  
záznamů těchto hodnot generálním dotazem. Modul  
HDBa s tímto nastavením musí mít vyšší instanci než  
ostatní moduly HDBa/HDBb.

Do komunikace jsou zahrnovány počáteční obsazené  
(zapojené) vstupy modulů HDBa, popř. HDBb. Prvním  
nezapojeným vstupem se scan vstupů modulu přerušuje a  
přechází na další modul HDBa / HDBb.

### Komunikační služby

**Jsou implementovány tyto komunikační služby:**

- Link reset
- Link status
- End of initialization
- General interrogation command (group 0)
- Podporované typy datových služeb jsou:**
- jednobitové: M\_SP\_NA\_1 a M\_SP\_TB\_1,
- dvoubitové: M\_DP\_NA\_1 a M\_DP\_TB\_1,
- 32bitový bitstring: M\_BO\_NA\_1 a M\_BO\_TB\_1,
- analogové short: M\_ME\_NA\_1 M\_ME\_TD\_1
- analogové float: M\_ME\_NC\_1, M\_ME\_TF\_1 .
- povely: C\_SC\_NA\_1, C\_DC\_NA\_1, C\_SC\_TA\_1,  
C\_DC\_TA\_1.

Cyclic polling

Clock synchronisation command - nastavení RTC  
centrály je pouze v celých minutách.

Všechny časy, používané v komunikaci protokolem IEC  
870-5-101 jsou „zimní“, čili bez uplatnění změn při přechodu  
na „letní“ čas.

Delay command

- služba je komunikačně implementována, ale nemá  
žádný efekt.

#### Poznámka

Ve spojení se CybServerem870M (OPC server pro  
protokol IEC870-5-101) [www.cybectec.com](http://www.cybectec.com), a OPC  
klienta, např. Kassl OPC explorer [www.kassl.de](http://www.kassl.de) je možné  
CCPU nejen dálkově řídit a monitorovat technologii OPC,  
ale i vyčítat a zobrazovat trendy (průběhy) z HDB.

## 3.34 IECinSP

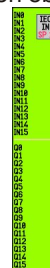
**IEC IN SP 1** Modul IECinSP je rozšiřující modul k modulu  
IEC870, je určen pro jednobitové vstupy.

Modul IEC870 je určen pro komunikaci na  
sériovém kanále centrály protokolem IEC  
870-5-101.

Vstupy In0..15 - binární/jednobitové vstupy.

Vstupy Q0..15 - quality descriptor, nepovinný celočíselný  
vstup. Odpovídající Q0..15 se logicky sečte (OR) se  
vstupem In0..15, výsledný bajt je pak ve zprávě.

Parametr **ioa** (Information object address) pro 0. vstup =



Obr. 74 Značka modulu IECinSP na ploše

počáteční adresa bloku signálů (adresa objektu).

Standardní parametry modulu: **rychlost a priorita**.

#### Poznámky k implementaci:

- komunikace v „unbalanced mode“
- každý modul IECin je vlastně základem jednoho IEC  
telegramu, v němž musí být veličiny téhož typu a s  
toutéž příčinou přenosu.

#### IECin moduly se zpracovávají

- **běžným postupem podle parametrů rychlost a priorita**  
Přitom se pro každý (obsazený) vstup může nastavit  
interní příznak „do telegramu“. U SP a DP se tato paměť  
nahodí při změně stavu. U ME pak vstupem „vyslat  
hodnotu“. Toto probíhá neustále, bez ohledu na stav  
komunikace IEC.
- **při komunikaci s nadřazenou stanicí**  
Komunikační program (reprezentovaný základním  
modulem IEC870) cyklicky prochází jednotlivé moduly  
IECin a - je-li něco k vyslání - vytvoří a pošle telegram.  
Zároveň smaže příznaky „do telegramu“.  
Přesněji řečeno: na každou výzvu s FC 10 (nebo s FC 11)  
se komunikační program „podívá“ do modulu IECin,  
který je právě „na řadě“, a má-li se něco z tohoto modulu  
poslat, vytvoří a pošle příslušný telegram (a smaže  
příznaky u vyslaných veličin). Na další výzvu provede  
totéž s následujícím modulem IECin. Není-li nic k vyslání,  
přeskočí na další modul IECin. Není-li v žádném modulem  
IECin nic k vyslání, odpoví krátkým telegramem s FC=9.  
Příznaky „do telegramu“ se mažou hned - nečeká se na  
to, jestli se telegram skutečně podaří odeslat. Je to proto,  
že neúspěšné odeslání telegramu (FCB od Mastera se  
nemění) musí skončit přerušením komunikace a jejím  
opětným navázáním (úvodní sekvence: >FC9, FC11,  
>FC0, FC0), po němž následuje celkový dotaz od  
Mastera. Nehrozí tedy ztráta dat. (To platí samozřejmě jen  
v našem případě pro nebufferovaná data bez časové  
značky.)  
V odpovědi na celkový dotaz se pošlou aktuální hodnoty  
ze všech (obsazených) vstupů.

Rychlost komunikace se definuje v základním modulem  
IEC870, zrovna tak chybová hlášení jsou generována v  
základním modulem IEC870 (jeho výstup Err) i pro moduly  
IECin, použité v projektu.

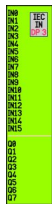
#### 3.34.1 IECinDP

**IEC IN DP 3** Modul IECinDP je rozšiřující modul k modulu  
IEC870, je určen pro dvoubitové vstupy.

Modul IEC870 je určen pro komunikaci na



sériovém kanále centrály protokolem IEC 870-5-101.



Obr. 75 Značka modulu **IECinDP** na ploše

Vstupy **In0..15** - binární/jednobitové vstupy.

Vstupy **Q0..7** - quality descriptor, nepovinný celočíselný vstup.

Vstupy **In0** a **In1** se složí do 1. dvoubitového signálu, vstupy **In2** a **In3** do 2. dvoubitového signálu, ... atd.

(Na rozvodnách je z nějakých důvodů zvykem slučovat signály od koncových spínačů [odpojovače, vypínače] do jednoho dvoubitového signálu.)

Quality descriptor - podobně jako u modulu **IECinSP**. **Q0** se logicky sečte s 1. vytvořeným dvoubitovým signálem (ze vstupů **In0** a **In1**), atd.

Parametr **ioa** (Information object address) pro první dvoubitový signál (bity 0 a 1) = počáteční adresa bloku signálů (adresa objektu).

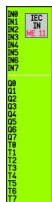
Standardní parametry modulu: **rychlost** a **priorita**.

*Poznámky k implementaci a zpracování modulů IECin jsou uvedeny u modulu IECinSP.*

### 3.35 IECinMI

**IEC IN ME 11** Modul **IECinMI** je rozšiřující modul k modulu **IEC870**, je určen pro celočíselné vstupy.

Modul **IEC870** je určen pro komunikaci na sériovém kanále centrály protokolem IEC 870-5-101.



Obr. 76 Značka modulu **IECinMI** na ploše

Vstupy **In0..7** - celočíselné (integer, 16 bitů) vstupy.

Vstupy **Q0..7** - quality descriptor, nepovinný celočíselný vstup, pro přidání bajtu do zprávy.

Vstupy **T0..7** - spouštěcí vstupy "vyslat hodnotu" (náběžnou hranou).

U nebitových vstupů (tento případ) je ve zprávě pro každou hodnotu jeden bajt navíc. Do něj lze ukládat **Q0..7**.

Parametr **ioa** (Information object address) pro 1. vstup **In0** = počáteční adresa bloku signálů (adresa objektu).

Standardní parametry modulu: **rychlost** a **priorita**.

*Poznámky k implementaci a zpracování modulů IECin jsou uvedeny u modulu IECinSP.*

### 3.36 IECinMF

**IEC IN ME 13** Modul **IECinMF** je rozšiřující modul k modulu **IEC870**, je určen pro vstupy typu float (reál.čís.).

Modul **IEC870** je určen pro komunikaci na sériovém kanále centrály protokolem IEC 870-5-101.

Vstupy **In0..7** - celočíselné (integer, 16 bitů) vstupy.

Vstupy **Q0..7** - quality descriptor, nepovinný celočíselný vstup, pro přidání bajtu do zprávy.

Vstupy **T0..7** - spouštěcí vstupy "vyslat hodnotu" (náběžnou hranou).

U nebitových vstupů (tento případ) je ve zprávě pro každou hodnotu jeden bajt navíc. Do něj lze ukládat **Q0..7**.

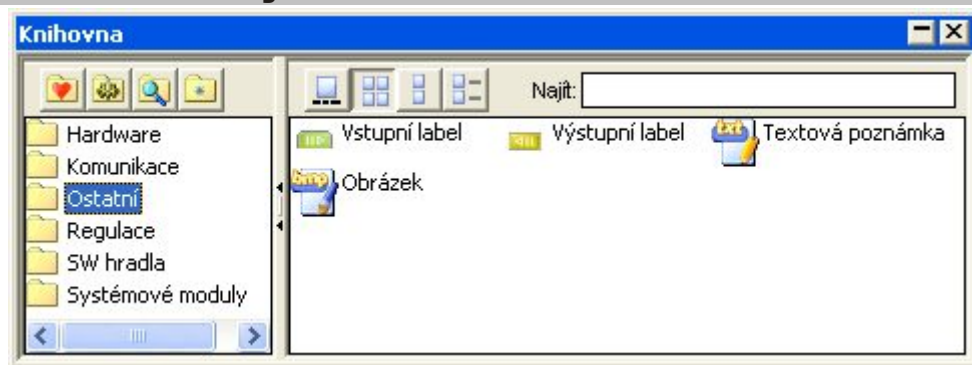
Např. čteme analogovou hodnotu z CAIO-12 do **In0** a stav CANu do **Q0**. Hlásí-li CAN chybu, můžeme tak analogovou hodnotu "zahodit".

Parametr **ioa** (Information object address) pro 1. vstup **In0** = počáteční adresa bloku signálů (adresa objektu).

Standardní parametry modulu: **rychlost** a **priorita**.

*Poznámky k implementaci a zpracování modulů IECin jsou uvedeny u modulu IECinSP.*

## 4 OSTATNÍ moduly



Obr. 77 Okno knihovny modulů  
Aktivní skupina Ostatní

Tato skupina obsahuje moduly pro napojování a pro vizualizaci na ploše listu. Moduly slouží zejména pro ladění aplikace a jako takové se "nepřekládají" a nepřesouvají do aplikačního programu.

Jedná se o moduly:

- **Vstupní label**  
label pro napojení na vstupní pin modulu
- **Výstupní label**  
label pro napojení na výstupní pin modulu
- **Textová poznámka**  
pro zobrazení textu na ploše
- **Obrázek**  
pro zobrazení obrázku na ploše listu
- **GRAF**  
pro grafické zobrazování průběhu zvolené měřené hodnoty
- **TLACITKO**  
pro nastavování napojené logické proměnné
- **POTENCIOMETR**  
pro nastavování napojené proměnné typu float

*Poznámka:*

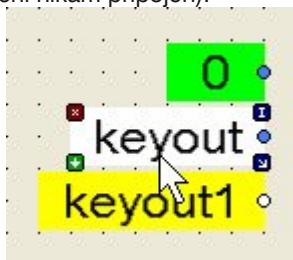
*Poslední tři uvedené se připravují do nové verze FREDa.*

## 4.1 Vstupní label

**III▶** Vstupní label lze vytvořit při tažení spoje a stiskem pravého tlačítka, kopírováním, klonováním nebo ho lze vytáhnout z knihovny. Může sloužit jako *konstanta* nebo *odkaz*.

*Konstanta* je buď číslo (např. 1, 2.57 nebo -59.5e-1 apod.) nebo řetězec uzavřený v uvozovkách (např. "ahoj" atd.). Label - *konstanta* má zelenou barvu (případně červenou, pokud došlo ke kolizi typů).

*Odkaz* je vlastně to, co nelze označit jako konstantu. *Odkaz* má bílou (pokud je někam připojen) nebo žlutou barvu (pokud není nikam připojen).



Obr. 78 Vstupní labely na ploše listu (200%)

Label má 4 rapid-buttony (nebo 3 pokud je to *konstanta*):

- smazat (*levý horní roh*) smaže label z listu
- změna textu/hodnoty (*pravý horní roh*) vyvolá změnu textu/hodnoty labelu
- klonovat objekt (*levý dolní roh*) vytvoří klon objektu poslední číslo názvu inkrementuje (např. z labelu "Teplota\_kotelna\_1" udělá nový label s textem "Teplota\_kotelna\_2" apod.)
- přechod na zdroj/cíl (*pravý dolní roh*) pouze u odkazů zobrazí seznam zdrojů a cílů, se kterými je odkaz spojen seznam obsahuje nejprve zdroje, oddělovač a seznam cílů

## 4.2 Výstupní label

**<II** Výstupní label lze vytvořit při tažení spoje a stiskem pravého tlačítka, kopírováním, klonováním nebo ho lze vytáhnout z knihovny. Může sloužit pouze jako odkaz.



Obr. 79 Výstupní labely na ploše listu (200%)

Label má 4 rapid-buttony:

- smazat (*levý horní roh*) smaže label z listu
- změna textu/hodnoty (*pravý horní roh*) vyvolá změnu textu/hodnoty labelu

- klonovat objekt (*levý dolní roh*) vytvoří klon objektu poslední číslo v názvu inkrementuje
- přechod na zdroj/cíl (*pravý dolní roh*) zobrazí seznam zdrojů a cílů, se kterými je odkaz spojen seznam obsahuje nejprve zdroje, oddělovač a seznam cílů

## 4.3 Textová poznámka



Textová poznámka slouží k vložení textu na list schéma /projektu. Lze vytvářet i víceřádkové poznámky, přechod na nový řádek se provede stiskem CTRL+Enter.

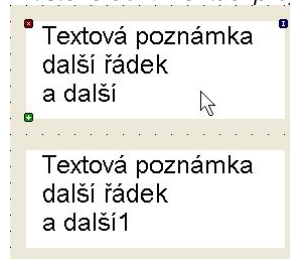
Vytvoříme ji vložením z knihovny nebo dvojklikem na volné ploše listu.

Textová poznámka

Obr. 80 Jednořádková po dvojkliku

Tip

Textovou poznámku lze označit jako komentář, který lze vypustit z tisku, takže je možné vkládat poznámky, které nebudou např. v tištěné dokumentaci projektu apod.



Obr. 81 Víceřádková

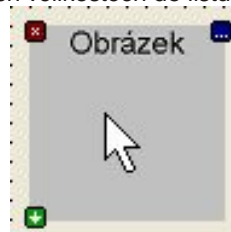
Textová poznámka má 3 rapid-buttony:

- smazat (*levý horní roh*) smaže textovou poznámku z listu
- klonovat (*levý dolní roh*) vytvoří textovou poznámku stejných parametrů, pouze poslední číslice v textu bude inkrementována (např. z textové poznámky "Kotelna 23" udělá textovou poznámku "Kotelna 24" apod.)
- změna textu (*pravý horní roh*) vyvolá změnu textu

## 4.4 Obrázek

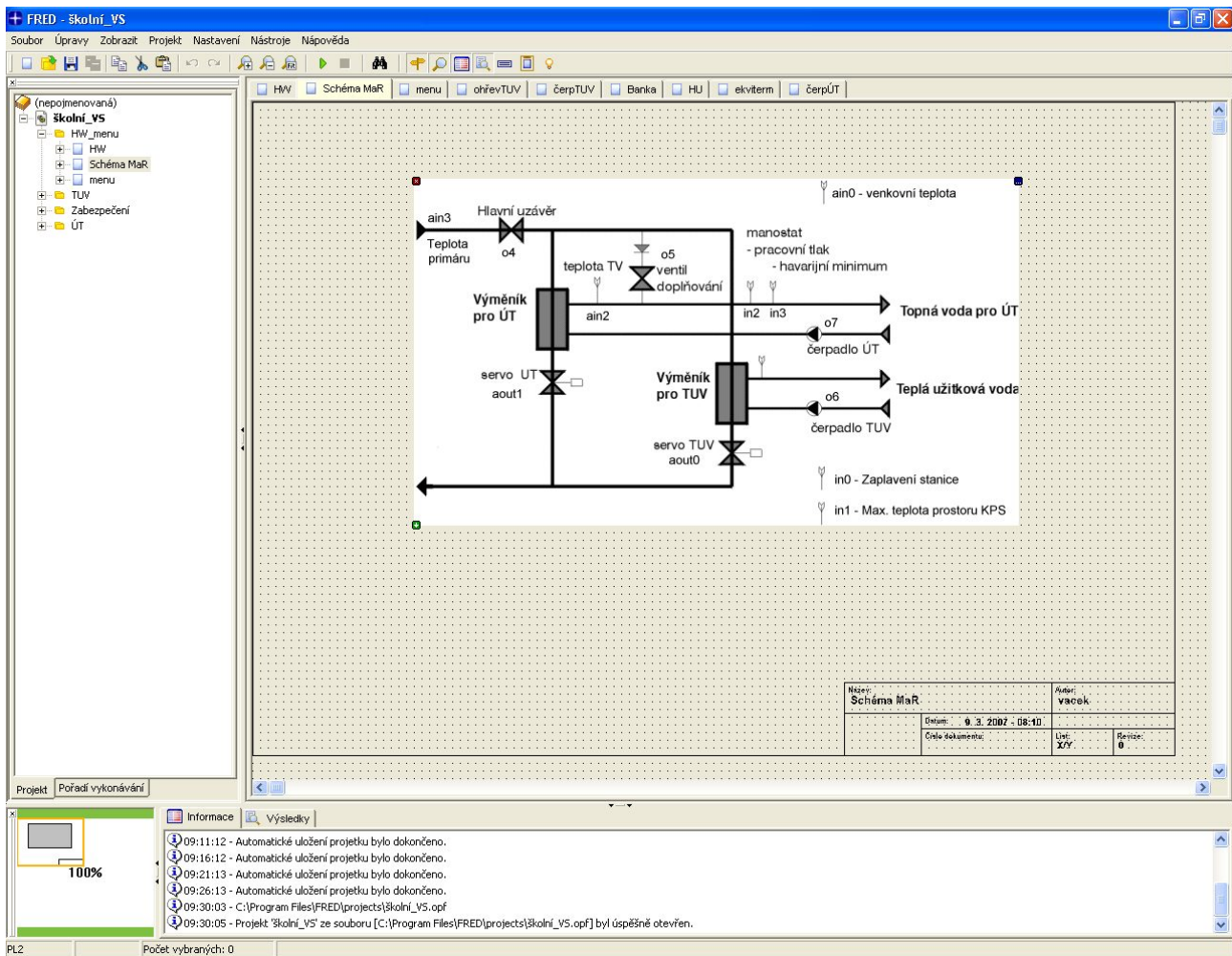


Obrázek slouží k vložení bitmapových obrázků v různých velikostech do listu.



Obr. 82 Po přetažení na plochu listu

Ikonu Obrázku přetáhneme z knihovny na plochu listu - viz obrázek. Kliknutím na šedé pole Obrázku vyvoláme dialog pro nalezení BMP souboru, pomocí kterého skutečný obrázek vložíme na plochu listu a podle potřeby dále na ploše umístíme.



Obr. 83 Obrázek (bitmapa) na ploše listu

**Tip**

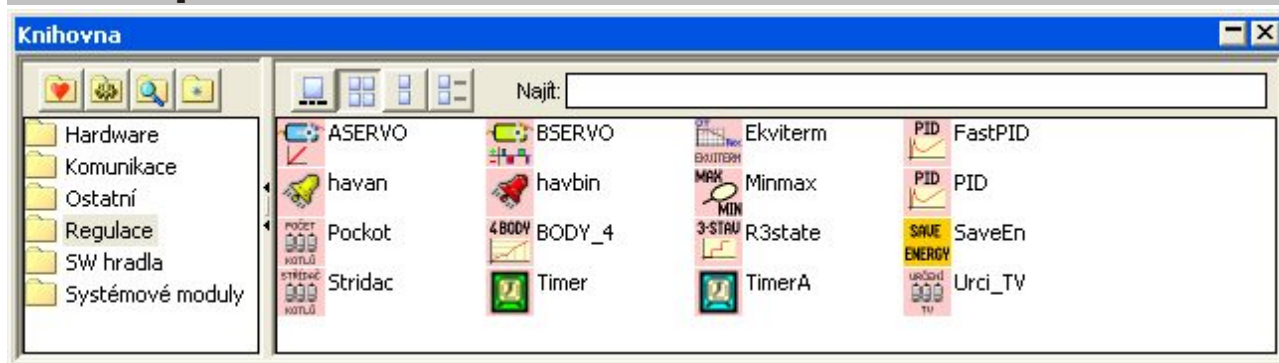
Obrázek lze označit jako komentář, který lze vypustit z tisku, takže je možné vkládat poznámky, které nebudou např. v tištěné dokumentaci projektu apod.

Obrázek má 3 rapid-buttony:

- smazat (levý horní roh)  
smaže obrázek z listu
- klonovat (levý dolní roh)  
vytvoří obrázek stejných parametrů
- změna obrázku (pravý horní roh)  
vyvolá dialog pro výběr souboru na disku, který se má zobrazit



## 5 Skupina modulů REGULACE



Obr. 84 Okno knihovny modulů -aktivní skupina **Regulace**

Ve skupině REGULACE jsou knihovní moduly regulačních smyček a moduly, přímo s regulací související:

- **ekviterm**  
modul pro výpočet teploty topného média  
vstup - hodnota venkovní teploty  
výstup - teplota topného média
- **pid**  
modul PID regulátoru  
vypočítává regulační zásah po porovnání zadané a změřené vstupní veličiny  
jeho výstupy jsou předávány modulům **aservo** nebo **bservo**, které akční zásah provádějí
- **fastpid**  
modul PID regulátoru pro rychlejší děje  
vypočítává regulační zásah po porovnání zadané a změřené vstupní veličiny  
výstupem jsou přímo akční zásahy
- **havbin**  
modul dvoustavové regulační smyčky  
vstup i výstup logická 0/1
- **havan**  
modul dvoustavové regulační smyčky  
vstup hodnota reálného čísla  
horní a dolní mez pro regulaci  
logický výstup 0/1
- **body\_4**  
modul čtyřbodové křivky  
lomená lineární funkce, zadaná 4 body
- **aservo**  
modul pro ovládání servopohonu, řízeného analogo-  
vým signálem  
obsahuje i vstupy pro havarijní zabezpečení
- **bservo**  
modul pro ovládání servopohonu, řízeného povel  
zavírá/otvírá  
obsahuje i vstupy pro havarijní zabezpečení
- **timer**  
modul pro generování hodnoty v určeném čase  
tzv. spínací hodiny, vhodné pro řazení útlumů apod.
- **stridac**  
modul pro střídání a záskoky strojů  
(čerpadel, plynových kotlů,...)
- **urci\_tv**  
modul pro určení teploty topné vody z kotlů
- **pockot**  
modul pro stanovení počtu kotlů v chodu
- **minmax**  
modul pro vyhodnocení poklesu, nárůstu, minima,  
maxima a průměru ze vstupního signálu za definova-  
ný čas.
- **r3state**  
modul třístavového regulátoru
- **SaveEn**  
modul pro odepínání spotřebičů  
(čtvrthodinová maxima)
- **TimerA**  
týdenní spínací hodiny

## 5.1 ekviterm



Modul **ekviterm** vykonává výpočet rovnice ekvitermní křivky, tj. výpočet hodnoty teploty topné vody (topného média) v závislosti na hodnotě venkovní teploty.



Obr. 85 Značka modulu EKVITERM na ploše

**In** je vstup venkovní teploty,

**O** je výstup žádané teploty topného média.

Vstup **Menu** slouží ke připojení na modul **SCREEN** pro tvorbu uživatelských výpisů na displeji (menu ap.).

Standardní parametry modulu: *rychlost a prioritá*.

V menu, poskytovaném modulem **ekviterm** je možné editovat parametry **posun** a **sklon** ekvitermní křivky. V menu je zobrazen i název regulátoru (modulu **ekviterm**) daný parametrem **Text\_line\_0**.

Parametry **posun** (implicitně=20,0) a **sklon** (implicitně=1,0) určují přímo parametry ekvitermní křivky.

Parametrem **korekce** (implicitně=0,0) je možné realizovat dodatečný posuv regulační křivky.

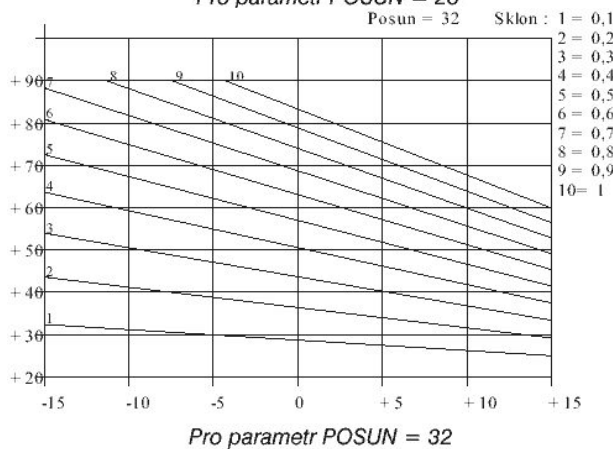
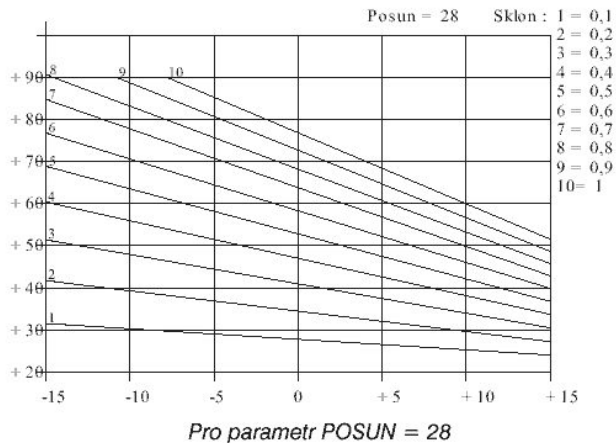
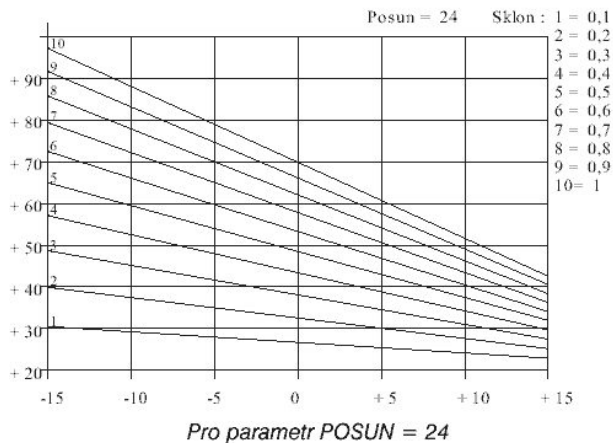
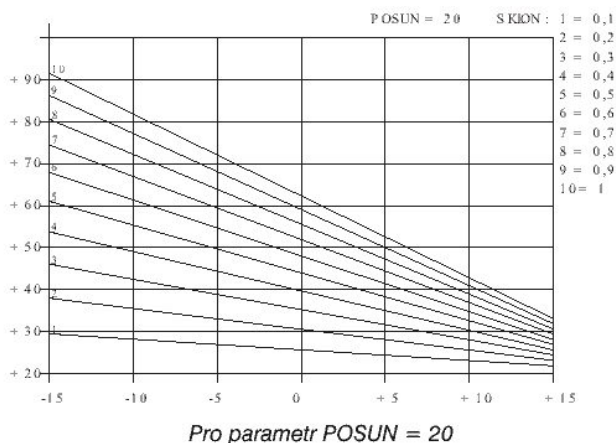
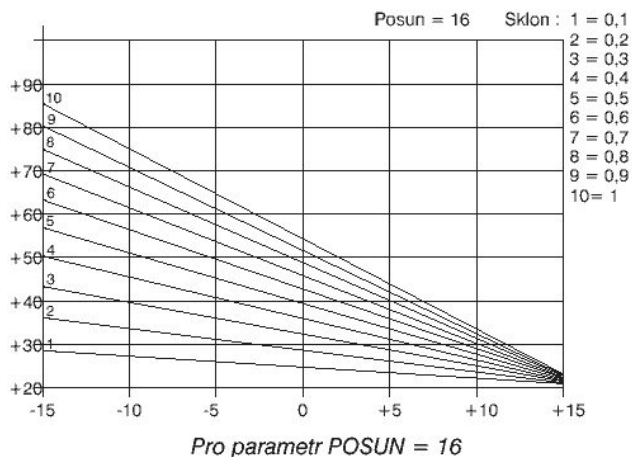
Parametr **TVPM** (implicitně=90,0) určuje maximální hodnotu výstupní hodnoty regulátoru (na výstupu **O**).

Parametr **TVPN** (implicitně=30,0) určuje minimální hodnotu výstupní hodnoty (na výstupu **O**).

Parametry **input** a **output** kopírují hodnoty vstupu a výstupu modulu **ekviterm**. Slouží zejména pro čtení hodnot v režimu RUN, případně pro přetažení do VISUAL modulů **hodnota a průběh**.

Pro standardní hodnoty parametrů **Posuv=20** a **Sklon=1** platí následující tabulka závislosti teploty topné vody na venkovní teplotě:

Teplota [°C]	
venkovní	TV (topné vody)
-15,0	91,5
-10,0	83,8
-5,0	75,3
0	66,0
5,0	55,8
10,0	44,9
15,0	33,1



Průběhy ekvitermní křivky pro různé parametry POSUV a SKLON

## 5.2 pid



**PID**

Modul **pid** vykonává funkci PID regulátoru.

Vstup **Tuz** je vstupem změřené (skutečné) veličiny.

Vstup **Tup** je vstupem požadované (zadané) hodnoty, na kterou regulujeme.



Obr. 87 Značka **pid** na ploše

Výstup **Tau** předává derivaci výstupu PID regulátoru (časově závislý povel pro zavírání/otvírání) a je určen pro ovládání serv, jejichž vstupem je změna regulačního zásahu. Běžně se zapojuje na vstup modulu **bservo**.

Výstup **Stau** je přímým výstupem PID regulátoru (hodnota otevřenosti serva v %) a je určen pro akční členy, jejichž vstupem je přímo regulační zásah. Běžně se zapojuje na vstup modulu **aservo**.

Logický výstup **Ktuv** signalizuje v log.1, že je servo plně otevřeno (viz dále).

Vstup **Menu** slouží ke připojení na modul **SCREEN** pro tvorbu uživatelských výpisů na displeji (menu ap.).

Standardní parametry modulu: *rychlost a priorita*.

Parametr **PIKP** určuje zesílení regulátoru (bezrozměrná veličina). Běžně se tento parametr pohybuje kolem hodnot 10 - 20.

Parametry **PITDplus** a **PITDminus** určují časovou konstantu (v sec) derivační složky pro vzestupnou a sestupnou tendenci vstupního signálu.

Pokud je vstupní hodnota snímána přes modul **CAIO**, je potřeba provozovat PID regulátor s parametrem rychlost=1 pro správnou funkci derivační složky. Pokud je **PITDplus** i **PITDminus** = 0, je možné regulátor provozovat i v rychlejší smyčce. Je třeba vždy brát v úvahu, že pro správné vyhodnocení derivační složky perioda vykonávání PID regulátoru musí být několikanásobně delší než perioda smírání vstupní hodnoty. O periodě snímání dat I/O jednotkami viz rychlost komunikace nastavená pro FREDa.

Tzn., že pokud má modul **pid** parametr *rychlost=1*, měl by mít modul, který zabezpečuje měření vstupní analogové veličiny, přiváděné na vstup **Tuz**, parametr *rychlost=2* nebo 3.

Parametr **PITI** určuje časovou konstantu (v sec) integrační složky. Běžně se tento parametr pohybuje kolem hodnot 100 - 200. Pokud bývá regulační odchylka (**Tuz-Tup**) v dané aplikaci příliš velká, volíme **PITI** podstatně menší (desítky, resp. i jednotky sec).

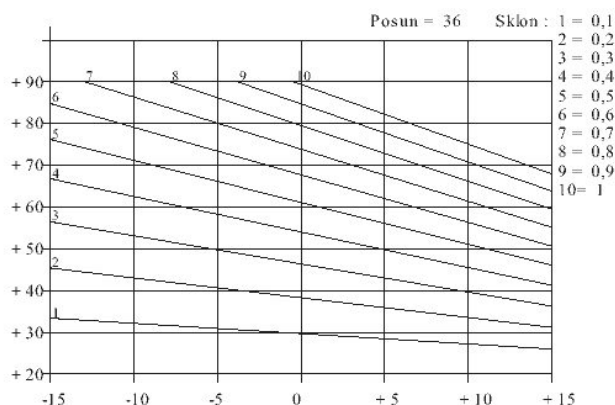
Parametr **TAUM** určuje maximální velikost regulačního zásahu na výstupu **TAU** jak v kladném, tak v záporném směru změny regulačního zásahu.

Parametr **TAUV** určuje maximální hodnotu na výstupu **Stau**.

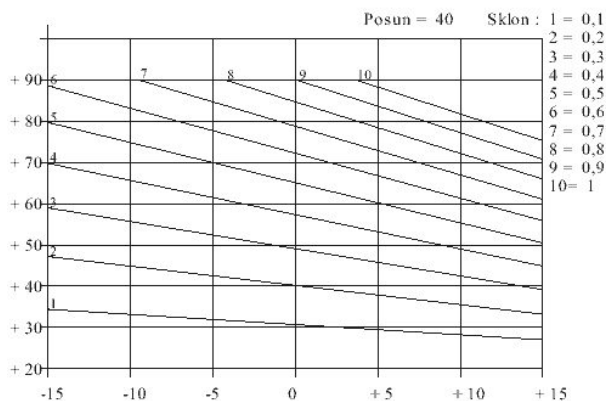
Pokud výstup **Stau** začne být limitován hodnotou parametru **TAUV**, nastaví se binární výstup **Ktuv** = 1, jinak je **Ktuv**=0. Minimální hodnota výstupu **Stau** je nula.

U modulu **pid** je v menu modulu zobrazen text z parametru **Text\_line\_0** a hodnota na vstupu **Tup**.

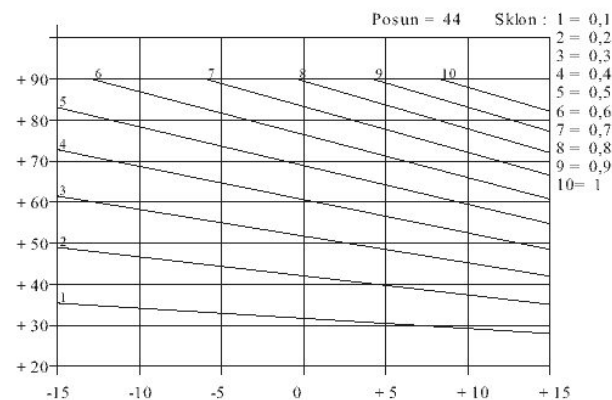
Parametr **Text\_line\_0** je přístupný v dialogovém okně modulu po kliknutí na tlačítko DIALOG MODULU.



Pro parametr POSUN = 36



Pro parametr POSUN = 40



Pro parametr POSUN = 44

Hodnotu **Tup** lze v menu editovat. Tato editace má však smysl **pouze** pokud je tento vstup připojen na modul **Constf**. Toto zapojení se využívá pro regulaci na konstantní hodnotu (ohřev TUV ap.).

Parametry **inTUZ**, **inTUP**, **outTAU** a **outSTAU** kopírují hodnoty vstupů a výstupů modulu **pid**. Slouží zejména pro čtení hodnot v režimu RUN, případně pro přetažení do VISUAL modulů **hodnota** a **průběh**.

### TEORIE PID REGULÁTORU

PID regulátor vychází z tohoto obecného vzorce:

$$\text{vystup} = \text{proporcionalni\_konstanta} * ((\text{zadana} - \text{skutecna}) + \text{derivacni\_konstanta} * \text{derivate}(\text{zadana} - \text{skutecna}) / dt + (1 / \text{Integracni\_konstanta}) * \text{integral\_pres\_dt}(\text{zadana} - \text{skutecna}) * dt)$$

kde

$dt$  je  $\Delta t$  čili časový interval, přes který se dělá výpočet,  
proporcionální konstanta = parametr PIKP,  
zadana = vstup Tup,  
skutečna = vstup Tuz,  
derivacní konstanta = PITDplus, resp. PITDminus.

Po transformaci na diskretní regulátor (což je náš případ) je

$dt$  perioda výpočtu regulátoru,  
derivace = (zadana\_nova - skutečna\_nova) - (zadana\_stara - skutečna\_stara),  
integral = (zadana\_nova - skutečna\_nova) + součet\_vsech\_prechozich (zadana\_stara - skutečna\_stara).

Při použití časového serva (modul **bservo**) je pak výstup pro časové servo derivací výstup regulátoru a ztrácí se stejnosměrná složka, což lze dokompenzovat integrační složkou.

### 5.3 havbin



Modul **havbin** vykonává funkci regulační smyčky dvoustavové regulace (např. detektoru havarijního stavu) s logickým (binárním) vstupem.



Obr. 88 Značka **havbin** na ploše

**Vstup** modulu je vstupem hlídáné logické veličiny.

Logický **výstup** je = 1 pokud je detekován hlídáný stav, jinak je = 0.

Standardní parametry modulu: rychlost a priorita.

Parametr **kvit** určuje, má-li být hlídáný stav kvitován.

Je-li **kvit=0**, pak je výstup aktivní pokud je detekován hlídáný stav, ale pokud hlídáný stav pomine, výstup je opět nula.

Pokud je **kvit=1**, výstup zůstane aktivní i po odeznění hlídáného stavu. Musí se "odkvitovat" = vzít na vědomí (stiskem klávesy DEL v menu).

Pokud je parametr **porucha=1**, modul **havbin** generuje v případě detekování hlídáného stavu chybové hlášení.

Jeho text je určen parametrem **Text\_line\_0**. Ten lze nadefinovat v dialogovém okně modulu **havbin** po kliknutí na tlačítko DIALOG MODULU.

Chybové hlášení je poté zaregistrováno a zobrazeno modulem **sysmon**.

Parametr **SHAH** určuje maximální dobu trvání indikace hlídáného stavu (doba omezení trvání výstupu v log.1).

Je-li **SHAH>99999990**, pak není funkční časové omezení výstupu ve stavu log.1.

Pokud je **SHAH<99999990** a je detekován hlídáný stav, jde výstup do log.1, a to maximálně na dobu danou hodnotou parametru **SHAH**.

Po vypršení času, nastaveného parametrem **SHAH** je výstup vynulován, a to i když hlídáný stav na vstupu trvá (=1).

Generování chybového hlášení, monitorované modulem **sysmon** však trvá.

Po případném odkvitování jde výstup do log.1 jen na velmi krátkou dobu a chybové hlášení je obnoveno. V tomto případě je po odstranění závady, která signalizuje poruchu, nutno provést RESTART regulátoru (vypnout a zapnout).

Měření času se provádí počítáním vykonávání modulu, čili **SHAH** udává počet vykonání modulu pro časový limit. Jinými slovy jednotkou **SHAH** je perioda vykonávání modulu **havbin** (daná parametrem rychlost). Proto hodnoty parametrů **SHAH**, **SHAM** a **N28M** jsou zhruba v sekundách tehdy, pokud zvolíme parametr rychlost = 2.

Parametr **N28M** je délka vyhodnocovacího intervalu v periodách vykonávání modulu **havbin**.

Parametr **SHAM** určuje, kolik detekovaných stavů **vstup = 1** během intervalu **N28M** (z kolika) způsobí indikaci hlídáného stavu.

#### Příklad pro havarijní hlášení

Vstup:

kontaktní čidlo pro zaplavení

Parametry:

priorita = 0

rychlost = 2

(modul vykonáván 1x za vteřinu - vhodné pro stanovení hodnot dalších parametrů)

kvit = 1

(chci kvitovat)

porucha = 1

(chci poruchové hlášení - ALARM)

Text\_line\_0 = "ZAPLAVENI VS"

(text pro poruchové hlášení na displej)

SHAM = 1

(kolik)

N28M = 1

(z kolika, tj. je-li během poslední vteřiny alespoň SHAMkrát = 1x vstup = 1, pak dej výstup = 1)

SHAH = velké implicitně

= nechci časové omezení výstupu v jedničce

Výstup:

Vždy, když vstup = 1 alespoň 1 vteřinu, je výstup = 1 a je vysláno chybové hlášení ZAPLAVENI VS na displej.

I když vstup spadne do 0, zůstává výstup = 1 a je vypisováno chybové hlášení.

Teprve po odkvitování v menu modulu SCREEN, vázaného na modul sysmon, klávesou DEL, spadne výstup do 0 a je zrušeno chybové hlášení ZAPLAVENI VS.

#### Příklad pro doplňování topné vody do systému

Vstup:

kontaktní manometr, kontakt sepnut při nižší tlaku, než je požadovaný

Parametry:

priorita = 0

rychlost = 2

(modul vykonáván 1x za vteřinu - vhodné pro stanovení hodnot dalších parametrů)

kvit = 0

(nechci kvitovat)

porucha = 1

(chci poruchové hlášení - ALARM)

SHAM = 1

(kolik)

N28M = 2

(z kolika, tj. je-li během posledních 2 vteřin alespoň 1 vteřinu vstup = 1, pak dej výstup = 1)

SHAH = 300



(max. doba trvání výstupu v jedničce je 300 sec, tj. 5 minut)

Text\_line\_0 = "porucha dopln. TV"

Výstup:

Bude-li na vstup modulu havbin přivedena log.1, bude do 2 sec výstup převeden z log.0 do log.1.

Tímto výstupem lze pomocí dalších knihovnických modulů ovládat ventil pro doplňování topné vody.

Podaří-li se soustavu dotlakovat do 5 minut, bude pak vstup=0 a výstup přejde rovněž do log.0.

Bude-li vstup=1 déle než 5 minut (protože SHAH=300 při rychlost=2), přejde výstup do stavu log.0 a bude vysláno chybové hlášení porucha dopln. TV.

Tuto poruchu nestačí odkvítovat, po odstranění závady je nutno regulátor vypnout a zapnout (restartovat).

## 5.4 havan



Modul **havan** vykonává funkci regulační smyčky dvoustavové regulace (např. detektoru havarijního stavu) s analogovým vstupem.



Obr. 89 Značka **havan** na ploše

**Vstup** modulu je vstupem hlídáné analogové veličiny.

Logický **výstup** je =1 pokud je detekován hlídáný stav, jinak je nula.

Standardní parametry modulu: *rychlost a priorita*.

Parametr **kvit** určuje, má-li být hlídáný stav kvitován.

Je-li **kvit=0**, pak je výstup aktivní pokud je detekován hlídáný stav, ale pokud hlídáný stav pomine, výstup je opět nula.

Pokud je **kvit=1**, výstup zůstane aktivní i po odeznění hlídáného stavu. Musí se "odkvitovat" = vzít na vědomí (stiskem klávesy DEL v menu).

Pokud je parametr **porucha=1**, modul **havan** generuje v případě detekování hlídáného stavu chybové hlášení.

Jeho text je určen parametrem **Text\_line\_0**. Ten lze nadefinovat v dialogovém okně modulu **havan** po kliknutí na tlačítko DIALOG MODULU.

Chybové hlášení je poté zaregistrováno a zobrazeno modulem **sysmon**.

Do textu hlášení lze zadat výpis aktuální hodnoty na vstupu modulu **havan**, a to formátem ###.## (platí od verze SW v ccpu 2.1).

Pokud je formát výpisu překročen, jsou cifry nahrazeny ve výpisu znakem E, např. EEE.EE.

**Horní mez** detekce je určena součtem parametrů **HAVH**, **deltHAVH** a hodnoty horního vstupu **dH**.

**Dolní mez** detekce je určena součtem parametrů **HAVN**, **deltHAVN** a hodnoty dolního vstupu **dN**.

Pokud je horní mez větší než mez dolní, havárie je detekována pokud překročí hodnota vstupu horní mez. Havárie přestane být detekována, pokud hodnota vstupu klesne pod spodní mez.

Pokud je horní mez menší než dolní, havárie je detekována pokud vstupní hodnota klesne pod horní mez. Havárie pak přestane být detekována, pokud hodnota vstupu stoupne nad dolní mez.

Parametr **N28M** určuje, kolik může být v souvislé řadě detekováno hodnot vstupů mezi horní a dolní mezí. Po překročení tohoto počtu výskytů je indikován havarijní stav.

Parametr **CA10** určuje konstantu exponenciálního zapominání, účinnou při zmenšující se havarijní odchylce.

Doporučená hodnota je v intervalu **<0,1>** a menší než **CA11**.

Parametr **CA11** určuje konstantu exponenciálního zapominání, účinnou při zvyšující se havarijní odchylce.

Doporučená hodnota je v intervalu **<0,1>** a větší než **CA10**.

Parametr **SHAM** určuje mez sumy poměrných překročení hodnoty vstupu hodnotu dolní meze pro vyvolání indikace havarijního stavu. Suma je počítána ze současné hodnoty poměrného překročení a předešlých hodnot poměrných překročení redukovaných příslušnou konstantou exponenciálního zapominání.

## 5.5 body\_4

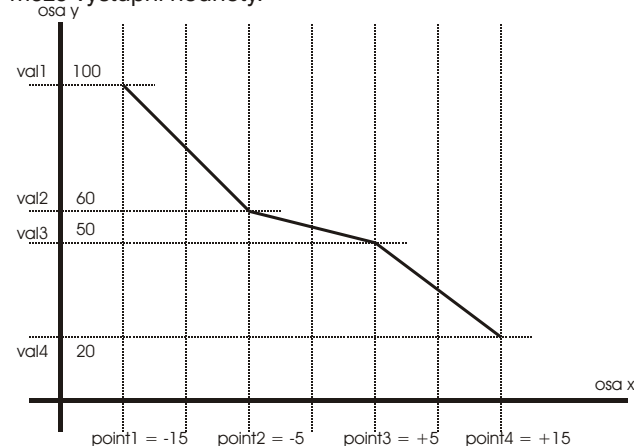


Modul **body\_4** realizuje lomenou lineární funkci na vstupní hodnotu **x** s výsledkem **y** na výstupu modulu.



Obr. 90 Značka **body\_4** na ploše

Lomená lineární funkce je dána 4 body této funkce, a to pomocí parametrů modulu včetně omezení horní a dolní meze výstupní hodnoty.



Obr. 91 Lomená lineární funkce modulu **body\_4**  
Příklad pro vytápění.

Výpočet výstupní hodnoty lze korigovat hodnotou, přivedenou na vstup **Korekce**.

Vstup **Menu** je opět určen k připojení na modul **SCREEN** pro zobrazení menu modulu **body\_4**.

V menu modulu jsou zobrazeny hodnoty polohy všech čtyř bodů funkce. Editovat lze však pouze polohu bodů na ose **y**, tedy výstupní hodnoty.

V menu je zobrazen i název modulu **body\_4**, daný parametrem **Text\_line\_0**.

Ten je přístupný v dialogovém okně modulu po kliknutí na tlačítko DIALOG MODULU.

Standardní parametry modulu: *rychlost a priorita*.

Parametry **point1..4** určují polohu těchto bodů v ose **x**, tedy vstupních hodnot.

Parametry **val1..4** určují polohu v ose **y**, tedy výstupních hodnot.

Parametr **maxout** (implicitně = 90) omezuje svou hodnotou maximální výstupní hodnotu modulu **body\_4**.

Parametr **minout** (implicitně = 10) omezuje svou hodnotou minimální výstupní hodnotu modulu **body\_4**.

Parametry **input** a **output** kopírují hodnoty vstupu a výstupu modulu **body\_4**. Slouží zejména pro čtení hodnot v režimu RUN, případně pro přetažení do VISUAL modulů **hodnota** a **průběh**.

## 5.6 aservo



Modul **aservo** je určen pro připojení serva řízeného analogovou hodnotou.

Hodnota výstupu modulu **aservo** přímo odpovídá hodnotě vstupu, modul je tedy vhodný pro připojení serva, u kterého vstupní hodnota určuje přímo regulační zásah z výstupu **Stau** modulu **pid** (viz obr. dole).



Obr. 92 Značka **aservo** na ploše

Logické vstupy **100%** a **0%** slouží pro přednostní úplné otevření nebo zavření servopohonu na vstupní povel.

Pokud je vstup **100%=1**, výstup modulu = 100 bez ohledu na stav vstupu.

Pokud je vstup **0%=1**, výstup modulu = 0 bez ohledu na stav vstupu.

V případě současné aktivace vstupů 100% a 0% zvítězí vstup 0%.

Vstup **Menu** je určen pro připojení na modul **SCREEN** pro vyvolání speciálního menu modulu **aservo**.

Standardní parametry modulu: *rychlost* a *priorita*.

Parametr **minact** určuje minimální změnu výstupní hodnoty.

Výstupní hodnota se nastaví hodnotou vstupu, pokud se vstupní hodnota odchýlí od hodnoty výstupní o více než hodnotu parametru **minact**.

Text názvu servopohonu je určen parametrem **Text\_line\_0**. Menu modulu **aservo** zobrazuje název serva, uvedený v parametru **Text\_line\_0**, a současnou hodnotu výstupu modulu **aservo**.

Tato hodnota je editovatelná stiskem klávesy INS na ovládacím panelu regulátoru. Tak lze ručně nastavit libovolnou hodnotu výstupu - polohu serva. Tato volba má přednost před hodnotou ze vstupů modulu. Po opuštění menu modulu **aservo** se výstupní hodnota modulu nastaví opět podle stavu vstupů.

## 5.7 bservo



Modul **bservo** je určen pro připojení serva, řízeného přírůstkem regulačního zásahu (binární vstupy otvírá/zavírá), kde přírůstek regulačního zásahu je úměrný době, po jakou je příslušný binární výstup otvírá/zavírá aktivní.



Obr. 93 Značka **bservo** na ploše

Vstup **In** modulu **bservo** je určen pro připojení na výstup **Tau** regulátoru PID (modul **pid** - viz obr. dole). Pro správnou funkci modulu je nezbytné, aby modul **bservo** a modul **pid** byly vykonávány ve stejné smyčce, čili se stejnou periodou (stejná hodnota parametru **rychlost**).

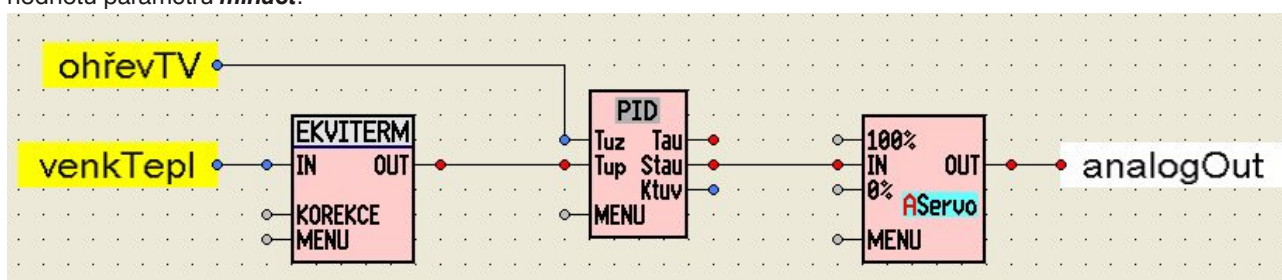
Modul **bservo** pracuje v zásadě tak, že nastaví výstup na dobu (v sekundách) danou hodnotou na vstupu.

Pokud je vstupní hodnota kladná nastaví výstup otvírá (+), pokud je vstupní hodnota záporná, nastaví výstup zavírá (-).

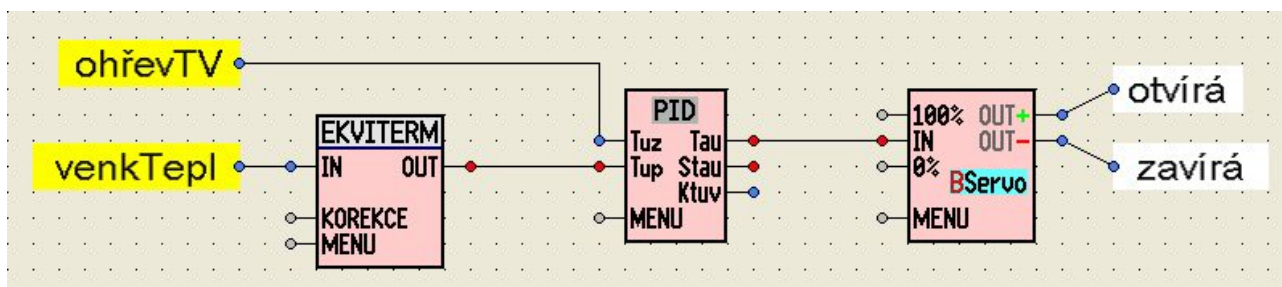
K této základní funkci poskytuje několik dalších funkcí, podporujících funkci připojeného serva.

Logické vstupy **100%** a **0%** slouží pro přednostní úplné otevření nebo zavření servopohonu na vstupní povel.

Pokud je vstup **100%=1**, pak výstup modulu "otevírá" (+) jde do log.1 bez ohledu na stav vstupu, ale pouze na dobu omezenou hodnotou parametru **maxtime**.



Obr. 94 Příklad zapojení modulů **ekviterm**, **pid**, **aservo**



Obr. 95 Příklad zapojení modulů **ekviterm**, **pid**, **bservo**

Pokud je vstup  $0\% = 1$ , pak výstup modulu "zavírá" (-) jde do log.1 bez ohledu na stav vstupu, ale pouze na dobu omezenou hodnotou parametru **maxtime**.

V případě současné aktivace vstupů 100% a 0% zvítězí vstup 0%.

Vstup **Menu** je určen pro připojení na modul **SCREEN** pro vyvolání speciálního menu modulu **bservo**.

Standardní parametry modulu: *rychlost a priorita*.

Výstupní hodnota se nastaví hodnotou vstupu, pokud se vstupní hodnota odchýlí od hodnoty výstupní o více než hodnotu parametru **minact**.

Text názvu servopohonu je určen parametrem **Text\_line\_0**. Ten lze nadefinovat v dialogovém okně modulu aservo po kliknutí na tlačítko DIALOG MODULU.

Menu modulu **bservo** zobrazuje název serva, uvedený v parametru **Text\_line\_0**, a dále umožňuje manuálně aktivovat výstup otevírá nebo zavírá. Toto manuální ovládání má přednost před ovládáním podle stavu vstupů.

Od FW 3.000 bylo menu ručního ovládání doplněno o výpis **AKCE** a zobrazení + nebo - nebo nic, a to podle aktuálního pohybu serva.

Po opuštění menu modulu **bservo** je stav výstupů opět určen hodnotami vstupů modulu.

*Příklad výpisu při ručním ovládání:*

```
RUCNE servo TUV
AKCE
Otevíra +
Zavíra -
```

Po stisku klávesy + servo začne otevírat a v řádce AKCE je zobrazeno +, ...

Pokud stiskneme nejprve klávesu SHIFT a pak klávesu + (nebo SHIFT a -), pak se servo otevírá (nebo zavírá) po dobu 30 sec.

Pokud v tomto případě chcete servo zastavit, stačí ještě jednou stlačit klávesu + nebo -.

Parametr **delay** určuje v sekundách prodlevu mezi přepnutími z jednoho směru otáčení motoru serva na druhý.

Parametr **minact** určuje minimální dobu (v sec) otáčení motoru serva.

Parametr **hystotev** určuje prodloužení změny regulačního zásahu (v sec) při přechodu ze směru změny regulačního zásahu zavírá na otevírá.

Parametr **hystzav** určuje prodloužení změny regulačního zásahu (v sec) při přechodu ze směru změny regulačního zásahu otevírá na zavírá.

Parametr **maxtime** určuje dobu (v sec) přeběhu servopohonu (z krajní polohy otevřeno do krajní polohy zavřeno).

**Znaménko** hodnoty parametru **maxtime** pak určuje, zda a jak bude výstup modulu **bservo** limitován.

#### maxtime > 0

Určuje v sekundách limit doby zásahu serva jedním směrem (tzn., je-li součet zásahů jedním směrem - a po sobě jdoucích - větší než **maxtime**, přestanou se pak vykonávat).

Přijde-li na vstup modulu **bservo** zásah s opačným směrem je opačný směr otáčení serva spuštěn téměř okamžitě.

#### maxtime < 0

Limit zásahu jedním směrem se nekontroluje, tzn. výstup modulu **bservo** není časově omezen.

Pokud jsou zásahy delší než (absolutní hodnota) **maxtime**, jsou v modulu **bservo** zapamatovány a přičteny k požadavku (maximálně však opět o hodnotu **maxtime**).

Např. pokud byl zásah jedním směrem delší než dvojnásobek **maxtime** a pak je na vstup modulu **bservo** přivedena 0, odpadne příslušný výstup po době **maxtime**. Pokud se změní (na vstupu) směr zásahu přes nulu, tak se doba přechodu na otáčení serva opačným směrem úměrně zkrátí.

*Poznámka:*

Obě hystereze by měli mít zadánu hodnotu menší než je hodnota parametru **minact**.

Hodnota parametru **minact** musí být větší než perioda vykonávání modulu, daná parametrem **rychlost**.

## 5.8 timer



Modul **timer** je určen pro generování časových hodnot, případně jako klasické spínací hodiny.

V menu modulu **timer** se nastaví časový interval.



Obr. 96 Značka **timer** na ploše

a) generování časovaných hodnot, např. pro útlumy ÚT apod.

Podmínkou je nenulová hodnota parametru **value**.

Hodnotový výstup **V** = 0, pokud systémový čas je mimo nastavený interval.

Hodnotový výstup **V** je roven hodnotě parametru **value**, pokud je systémový čas uvnitř intervalu daném parametry modulu.

Logický výstup **V < > 0** je aktivní (=1) v případě, že hodnotový výstup **V** je různý od nuly.

b) spínací hodiny

Podmínkou je nulová hodnota parametru **value**.

Hodnotový výstup **V** je vždy nulový.

Logický výstup **V < > 0** je aktivní (=1) v případě, že systémový čas je uvnitř nastaveného časového intervalu modulu **timer**.

Vstupem **Menu** je možné modul připojit na výstup modulu **SCREEN**.

Pak lze měnit nastavení spínacího intervalu a hodnoty **value** přímo z terminálu.

Na **druhém řádku** displeje je v případě nenulového parametru **value** vypisován údaj o hodnotě tohoto parametru.

Pokud je parametr **value=0**, je na tomto řádku vypisován nápis **On/Off**, který nás informuje o stavu logického výstupu **V < > 0**, který je závislý na tom, zda systémový čas regulátoru je uvnitř nastaveného časového intervalu v modulu **timer**.

Nastavování je podobné jako nastavování systémového času:

- klávesy <- -> pohybují kurzorem doleva/doprava
- klávesy + a - mění nastavení času
- nastavení hodnoty **value** je možné klávesou INS
- nastavené hodnoty se uloží stiskem ENTER

**Třetí řádek** je čas počátku intervalu, **čtvrtý řádek** je čas konce intervalu.

Mezi hodnotami nastavení času (při listování šipkou) je vždy jedna hodnota **no** (před nulou), která znamená, že tato položka bude při vyhodnocování ignorována (vždy pravda, **no** znamená libovolná hodnota). Tak je možné určit intervaly s periodou jedna hodina, den, týden, měsíc nebo rok. Pokud



je zadán den v měsíci je hodnota zadaná pro den v týdnu ignorována.

Pokud je interval zadáván parametry ve FREDovi, hodnota **no** odpovídá hodnotě parametru **-1**.

Standardní parametry modulu: *rychlost a priorita*.

Parametry **hodina\_on**, **minuta\_on**, **po\_ne\_on**, **den\_on**, **mesic\_on**, **rok\_on** určují počátek časového intervalu.

Parametry **hodina\_off**, **minuta\_off**, **po\_ne\_off**, **den\_off**, **mesic\_off**, **rok\_off** určují konec intervalu.

*Poznámka:*

*Parametry pro počátek i konec časového intervalu musí být nastaveny tak, aby si vzájemně odpovídaly.*

*Nelze volit počátek jako např. týdenní útlum a konec jako měsíční útlum.*

Parametr **value** určuje hodnotu výstupu v případě, že systémový čas je uvnitř zadaného intervalu.

Parametr **title** určuje popis (název) časovače, který se objeví v menu časovače na terminálu na prvním řádku. Ten je přístupný v dialogovém okně modulu po kliknutí na tlačítko DIALOG MODULU.

Parametr **vtitle** určuje popis (název) hodnoty, kterou časovač odevzdává na výstup (dříve pouze pevný text na displeji VALUE). Objeví se ve druhém řádku menu časovače. Ten je přístupný v dialogovém okně modulu po kliknutí na tlačítko DIALOG MODULU.

## 5.9 stridac



Modul **stridac** je určen pro funkci střídání a záskoků čerpadel, plynových kotlů, ... (strojů).

Modul je z technologie informován o chodu či poruše jednotlivých strojů. Musíme nadefinovat příslušnými vstupy a parametry celkový počet ovládaných strojů a požadovaný počet strojů, které mají být současně v chodu.

STŘÍDAČ	
□□□□	
CHOD 0	OUT 0
POR 0	
CHOD 1	OUT 1
POR 1	
CHOD 2	OUT 2
POR 2	
CHOD 3	OUT 3
POR 3	
CHOD 4	OUT 4
POR 4	
CHOD 5	OUT 5
POR 5	
CHOD 6	OUT 6
POR 6	
CHOD 7	OUT 7
POR 7	
	STATE
MAX NUM	
REQ NUM	
ALL OFF	
SWITCH 2	
SWITCH 1	

Obr. 97 Značka modulu **stridac** na ploše

K záskoku za stroj v poruše dochází automaticky, na povel dochází ke střídání strojů v chodu.

Pokud ovládáme stroje o stejném výkonu (např. čerpadla), platí pro parametry **power0..7** implicitní hodnoty = 1.

Pokud chceme tento modul použít pro řazení strojů o nestejném výkonu (např. různých plynových kotlů), využijeme pro správné řazení parametry **power0..7** s tím, že jejich celkový výkon uvedeme do vstupu **MAX.num**. Do vstupu **Req.num** pak přivádíme požadavek, jaký výkon mají kotle právě odevzdat.

Vstupy typu hodnota (integer):

Maximální celkový počet (výkon) strojů definujeme vstupem **Max.num**.

Požadovaný počet strojů současně v chodu (nebo právě požadovaný výkon strojů) je dán vstupem **Req.num**.

*Pokud je Req.num větší než Max.num je počet strojů, které jsou současně v chodu omezen na Max.num.*

Logické vstupy:

Povel ke střídání: **Switch1** a **Switch2**

Informace o chodu strojů: **Chod0...7**

Informace o poruše strojů: **Por0...7**

Povel k odstavení všech strojů: **All off**

Logické výstupy:

Povely pro zap/vyp strojů: **Out0...7**

Výstup typu hodnota (integer):

Výstup **state** indikuje aktivním bitem 0..n, který stroj (kotel/čerpadlo) je řazen jako první (je aktivní i když jsou všechny Outn=0).

Vhodné využití např. pro další typ doběhu příslušného čerpadla u řazení kotlů.

Standardní parametry modulu: *rychlost a priorita*.

Parametry:

- **number**  
- počet přepínaných strojů algoritmem ve střídači
- **power0..7**  
- výkon (váha) jednotlivých strojů
- **seldev**  
- určuje, který stroj má být střídačem spuštěn po překladi

*Poznámka:*

*Po výpadku el. energie střídač zapíná stroje podle stavu střídače před výpadkem.*

*Pokud zadáváte v projektu pokyn pro střídání pomocí modulu **keyin**, doporučujeme aby tento měl hodnotu parametru **rychlost=3**. Pokud modul **střídač** má parametr **rychlost** o jiné hodnotě, je třeba za modul **keyin** zařadit modul **mono** s nastaveným časem v parametru **time** tak, aby se informace mezi moduly spolehlivě předaly. Teprve potom je odezva na stisk klávesy rychlá a stoprocentní.*

## 5.10 urci\_tv



Modul **urci\_tv** slouží ke stanovení požadované teploty topné vody z kotlů.

TUP 0	TU
TUP 1	
TUP 2	
TUP 3	
TUP 4	
TUP 5	
TUP 6	
TUP 7	
T.REZERVA	
□□□□	

Obr. 98 Značka modulu **urci\_tv** na ploše

Modul vybere nejvyšší teplotu ze vstupů **TVP0..7** (což jsou teploty topných vod z ekvitermů pro jednotlivé topné větve dané plynové kotelnou), k tomu přidá hodnotu dle parametru **tempres** na tepelné ztráty (+ případně ještě hodnotu na vstupu **T.Rezerva**, je-li vstup zapojen) a takto stanovenou hodnotu omezí podle parametrů **mintemp** (spodní mez) a **maxtemp** (horní mez). Výsledek je předán na výstup **TV**.

Všechny vstupy, výstup i uvedené parametry jsou typu float (reál.čís.).



Standardní parametry modulu: **rychlost a priorita**.

Poznámky:

Rezervu na tepelné ztráty lze zadat jednak do parametru **tempres**, jednak i pomocí vstupu **T.Rezerva**, případně i kombinací obou. Jejich hodnoty se přičítají k vybrané nejvyšší hodnotě ze vstupů **TVP0..7**. Tak lze ovlivňovat výstup **TV** i z venčí modulu **urci\_tv**, ale vždy v mezích **mintemp** a **maxtemp**.

Hodnotu parametru **mintemp** lze využít i pro antikorozi ochranu kotle. Např. pro ocelolitinové kotle někteří výrobci doporučují nejnižší teplotu topné vody v kotli 42°C, pro litinové kotle pak 70°C, apod. Proto je implicitní hodnota dána 42°C.

Horní hodnotu teploty topné vody z kotlů zadáváme parametrem **maxtemp**. Tu definujeme podle technologie i druhu použitých kotlů, většinou v rozsahu 95°C až 105°C. Proto je implicitní hodnota dána 95°C.

Na modul **urci\_tv** běžně navazuje modul **pocket**.

## 5.11 pocket



Modul **pocket** slouží ke stanovení počtu kotlů plynové kotelny, které mají být právě v chodu.



Obr. 99 Značka modulu **pocket** na ploše

Modul srovnává teplotu topné vody změřenou na výstupu kotlů (vstup **ZmerTV**) s teplotou požadovanou (vstup **PozTV**, který se běžně napojuje na výstup **TV** modulu **urci\_tv**).

Zároveň změřenou teplotu na výstupu kotlů vzorkuje (za definovaný čas - daný parametrem **testperiod**) a tak si vnitřně určuje strmost nárůstu / poklesu této teploty za čas daný zmíněným parametrem.

Další parametr **tempdiffrsh** určuje meze detekce teploty (změřené), implicitně  $\pm 3^\circ\text{C}$ .

Parametr **tempendtrsh** určuje práh zjišťování poklesu či nárůstu této teploty, implicitně  $0.1^\circ\text{C}$ .

Maximální počet obsluhovaných kotlů je určen parametrem **pocetkotlu**, implicitní je hodnota 8.

Ten slouží k omezení vstupní hodnoty na výstupu **Pocket**.

Výstup **Pocket** se běžně napojuje na vstup **Req.num** modulu **stridac**, který pak vydává povely pro zapínání kotlů.

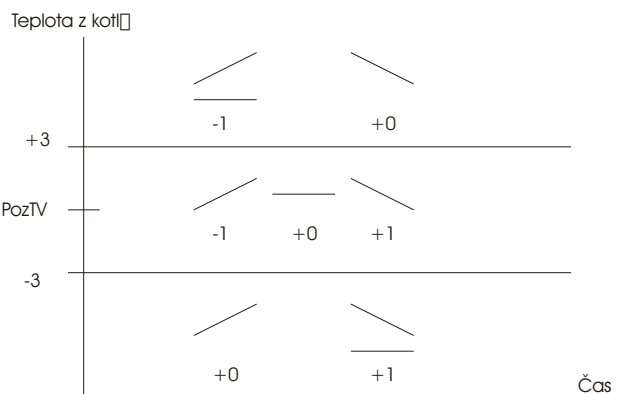
Do modulu **pocket** jsou rovněž přivedena hlášení o poruchách jednotlivých kotlů, a to na vstupy **Por0..7**.

Je-li některý ze vstupů **Por0..7** v "1", modul tak ví, že může řídit méně kotlů než je jeho maximum dle parametru **pocetkotlu**.

Poznámka:

Původně byla maximální hodnota parametru **pocetkotlu** omezena na 8, pokud byla zadána hodnota větší, automaticky byla opravena na 8.

Od verze SW centrály 1.6 z 8.2.2002 není maximální hodnota parametru **pocetkotlu** omezena. Protože má modul k



Obr. 101

Princip vyhodnocování teploty **TV** z kotlů pro přidávání / ubírání počtu kotlů v chodu.

dispozici pouze hlášení o poruchách od prvních 8 kotlů (vstupy **Por0..7**), nemohou být při větším počtu kotlů než 8 brána v potaz hlášení o poruchách dalších kotlů.

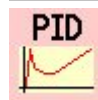
Protože výpočet požadovaného počtu kotlů probíhá cyklicky po době **testperiod**, musí modul znát předchozí požadovaný počet kotlů v chodu.

To zabezpečíte vazbou výstupu **Pocket** na vstup **Pocket-puv**. **Proto nezapomínejte tyto piny na schématu propojit!**

Vstup **Stop kotlu** zabezpečuje řazení kotlů po nuceném stopu od 1. Je-li **Stop kotlu** = 1, je umožněno řazení kotlů až po jejich plný počet (dáno parametrem **pocetkotlu**). Je-li **Stop kotlu** = 0, je řazen pouze jeden kotel, tzn. že výstup **Pocket**=1.

Standardní parametry modulu: **rychlost a priorita**.

## 5.12 fastpid



Modul **fastpid** vykonává funkci PID regulátoru.

Je určen pro regulaci rychlých a špatně regulovatelných dějů, zejména v oblasti limitací a přechodových dějů mimo pásmo proporcionality.



Obr. 100 Značka **fastpid** na ploše

Vstup **Tuz** je vstupem změřené (skutečné) veličiny.

Vstup **Tup** je vstupem požadované (zadané) hodnoty, na kterou regulujeme.

Logické vstupy **100%** a **0%**

- slouží pro přednostní úplné otevření nebo zavření servopohonu na vstupní povel
- vstup **0%** má přednost
- aktivací těchto vstupů se nuluje integrační složka (po jejich uvolnění tedy regulátor nastaví hodnotu podle složky proporcionální)

Horní vstup **menu** slouží po napojení na modul **SCREEN** pro zobrazení a použití menu pro ruční ovládání výstupů **Aout** / + / - modulu **fastpid**.

Název tohoto ovládání zadávejte do parametru **Text\_line\_0** po stlačení tlačítka **DIALOG MODULU** dialogového okna modulu **fastpid**.

Ruční ovládání reaguje na klávesy + a -. Pokud chcete zvolenou klávesou ručního ovládání vydat delší povel, stisk-

něte nejprve klávesu SHIFT a teprve pak klávesu + nebo -. Potom takový povel trvá cca 30 vteřin.

Rychlost ručního ovládání analogového serva lze ovlivnit hodnotou parametru **brange** (viz poznámka dále).

Výstup **Aout** (v rozsahu 0 - 100%) je určen přímo pro ovládání analogového serva (zapojuje se v sekci *Analog* na pin **Outx** modulu **CCPU-02** nebo **CAIO**).

Výstupy + a - odpovídají obdobným výstupům modulu **bservo**, slouží pro ovládání relé pro signály servo otvírá / zavírá.

Standardní parametry modulu: **rychlost a priorita**.

Parametr **PIKP** určuje zesílení regulátoru (bezrozměrná veličina).

Parametr **PITD** určuje časovou konstantu (v sec) derivační složky vstupního signálu.

FastPID regulátor provozujte s parametrem **rychlost=1 nebo 2**.

Je třeba vždy brát v úvahu, že pro správné vyhodnocení derivační složky perioda vykonávání FastPID regulátoru musí být několikanásobně delší než perioda smírání vstupní hodnoty.

Vstupní hodnota změřené veličiny (přiváděná na pin **Tuz**) má být snímána modulem **CAIO**, který má jednak nastaven parametr **rychlost=3**, jednak je využito jeho vstupů **In0...3**, které s centrálou komunikují každých 30 ms (na rozdíl od ostatních, které komunikují pouze každých 300 ms).

Parametr **PITI** určuje časovou konstantu (v sec) integrační složky. Definice **PITI=0** úplně zakáže integrační složku.

Parametr **delay** v ms určuje hodnotu prodlevy mezi přepnutími z jednoho směru otáčení motoru serva na druhý.

Praktická hodnota odpovídá násobkům času vykonávání programové smyčky, do které je modul **fastpid** zařazen.

Např. je-li **rychlost=2** a **delay=500** ms, bude vřazená prodleva při změně směru otáčení serva 1000 ms.

Parametr **brange** slouží pro zadání hodnoty času pro přeběh (z 0% na 100%) servopohonu v sec.

Parametr **mulper** udává, jednou za kolik period programové smyčky se vykoná výpočet modulu **fastpid**. Defaultně roven 1 (tj. každým průchodem programovou smyčkou).

Parametry **inTUZ**, **inTUP**, **analogout** kopírují hodnoty vstupů a výstupů modulu **fastpid**.

Slouží zejména pro čtení hodnot v režimu RUN, případně pro přetažení do VISUAL modulů **hodnota a průběh**.

*Poznámka:*

*Modul fastpid generuje analogovou hodnotu 0..100 %.*

*Pomocí parametru brange převádí změny této analogové hodnoty na ± binární výstupy.*

*V ručním režimu to funguje obráceně. Klávesami ± se generují ± binární výstupy.*

*Pomocí parametru brange se tyto převádějí na změnu analogové hodnoty.*

## 5.13 minmax



Modul **minmax** je určen pro analýzu průběhu vstupního (float) signálu. Detekuje nárůst či pokles vstupního signálu, vyhodnocuje maximum, minimum a průměr vstupních hodnot tohoto signálu.

Testovaný signál přivádíme na vstup **Inp** modulu **minmax**.

Periodu taktování lze synchronizovat signálem, přivedeným na vstup **Trig**.



Obr. 102 Značka **minmax** na ploše

Na výstupu **Min** je nabízena minimální hodnota vstupního signálu za testovací periodu.

Na výstupu **Max** je nabízena maximální hodnota vstupního signálu za testovací periodu.

Na výstupu **nárůst** (značen šipkou doprava nahoru) je log. 1, pokud během testované periody došlo k nárůstu vstupní veličiny alespoň o hodnotu danou parametrem **delta**. Jinak je na něm log. 0.

Na výstupu **pokles** (značen šipkou doprava dolů) je log. 1, pokud během testované periody došlo k poklesu vstupní veličiny alespoň o hodnotu danou parametrem **delta**. Jinak je na něm log. 0.

Na výstupu **průměr** je hodnota průměru testovaného vstupního signálu za testovanou periodu.

Standardní parametry modulu: **rychlost a priorita**.

Vyhodnocení se provádí během časového období, které je dáno hodnotou parametru **period** (v ms, defaultně 100 ms). **Hodnota nesmí být nulová !**

Pokud je **period > 0**, je perioda testování nezávislá na vstupu **Trig**.

Pokud je **period < 0**, je vyhodnocení provedeno vždy při přechodu vstupu **Trig** z 0 do 1.

Je ale nutno dodržet periodu signálu na vstupu **Trig** (např. pomocí modulu **timer** - po 15 min - 0:00, 0:15, 0:30, 0:45 atd.) shodnou s periodou zadanou parametrem **period**, aby byly správně vyhodnocovány tendence vstupního signálu, zejména při parametru **cache = 0**.

Pokud parametr **cache = 1**, hodnoty jsou vyhodnocovány během vyhodnocovací periody na pozadí, a na výstupy modulu **minmax** jsou přeneseny na konci testovacího období (např. při aktivaci vstupu **Trig**, nebo po uplynutí doby požadované pro provedení testování vstupu). Čili modul indikuje hodnoty za celou periodu měření.

Pokud je **cache = 0**, jsou všechny hodnoty přenášeny na výstup průběžně už během testovací periody. V tomto módu modul indikuje hodnoty okamžitě, a to za období od počátku probíhající testovací periody.

Parametr **delta** (defaultně 3,00) určuje, o kolik se musí změnit vstupní hodnota během testovací periody, aby byl vyhodnocen nárůst/pokles.

*Poznámka:*

*Při ladění projektu v ladícím režimu FREDa a změně parametru, který má vliv na mód modulu **minmax** (např. přechod z kladných do záporných hodnot parametru **period** a naopak) není zaručena správná funkce modulu **minmax**. Modul bude zpracovávat zpočátku původní data a jeho výsledky budou nesprávné. Musíte vyčkat na další zpracování hodnot podle nových parametrů (další průchod programovou smyčkou, vypršení času v parametru **period**, ...).*

## 5.14 R3state



Modul **R3state** vykonává funkci třístavového regulátoru.

Pokud je odchylka skutečné a požadované hodnoty (TUZ - TUP) v rozsahu hodnoty zadané v parametru **rangedead** (rozsah mrtvé zóny), není ani jeden z výstupů (+/-) aktivní.



Obr. 105 Značka modulu **R3state** na ploše

Pokud je odchylka skutečné a požadované hodnoty v rozsahu hodnoty zadané v parametru **rangeup** (rozsah cyklování nahoru), resp. **rangedown** (rozsah cyklování dolů), cykluje výstup + (up), resp. - (down), s periodou danou parametrem **cperiod**. Cyklování je PWM, čili pokud je odchylka větší je příslušný výstup delší dobu v 1 než v 0, a naopak.

Vstup **Tup** - požadovaná hodnota

Vstup **Tuz** - změřená (skutečná) hodnota

Výstup + je aktivní (v 1) pro zásahy typu servo otevírá, topit, nechládit ... ap. Jinak je v 0.

Výstup - je aktivní (v 1) pro zásahy typu servo zavírá, chládit, netopit ... ap. Jinak je v 0.

Parametr **cperiod** určuje periodu cyklování v programových smyčkách, tzn., že skutečná perioda cyklování záleží na tom, v jaké programové smyčce je modul umístěn (dáno parametrem **rychlost**).

Např.:

při **rychlost=3** a **cperiod=30**,  
tj.  $100ms \text{ smyčka} \times 30 = 3000ms$ .

Výstupy + resp. - jsou trvale aktivní, když je regulační odchylka (TUZ - TUP) mimo oblast danou parametrem **rangeup** resp. **rangedown**.

Standardní parametry modulu: **rychlost** a **priorita**.

Další parametry **inTUZ**, **inTUP**, **outplus** a **outminus** pouze kopírují hodnoty vstupů a výstupů modulu **R3state**. Slouží pro datové napojení na VISUAL moduly **hodnota** a **prubeh**.

## 5.15 SaveEn



Modul **SaveEn** (Save Energy = šetřit energii) je implementací odepínání spotřebičů pro hlídání čtvrt hodinového maxima.

Aby souhlasily veškeré nastavovací časy, musí být tento modul ve vteřinové programovací smyčce, tj. parametr **rychlost=2**.

Parametry **power0..15** jsou příkony ("váhy") jednotlivých strojů.

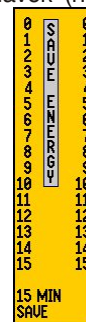
Parametry **maxoff0..15** jsou maximální časy vypnutí jednotlivých strojů v sekundách.

Parametry **priority0..15** určují prioritu vypínání jednotlivých strojů:

0 - vypíná nejdříve ..... až 7 - vypíná naposledy

Vstupy **0..15** jsou požadavky na zapnutí strojů.

Vstup **Save** je požadavek (hodnota kolik) na ušetření



Obr. 103 Značka modulu **SaveEn** na ploše

výkon.

Vstup **15min** vzestupnou hranou (do "1") nuluje počítadla časů odpojení jednotlivých strojů a způsobuje prostřídání strojů (o 1 v každé prioritě).

Výstupy **0..15** jsou požadavky na odpojení jednotlivých strojů.

Výstup **Saved** (poslední dole) indikuje hodnotu skutečně ušetřeného výkonu.

Standardní parametry modulu: **rychlost** a **priorita**.

## 5.16 TimerA



Modul **TimerA** jsou týdenní spínací hodiny.

Časový interval sepnutí, tj. stav kdy výstup **V=1**, se nastavuje pomocí vnitřních parametrů modulu.



Obr. 104 Značka modulu **TimerA** na ploše

Den v týdnu, kdy jsou spínací hodiny funkční, se nastavuje bitově pomocí parametru **po\_ne\_mask**.

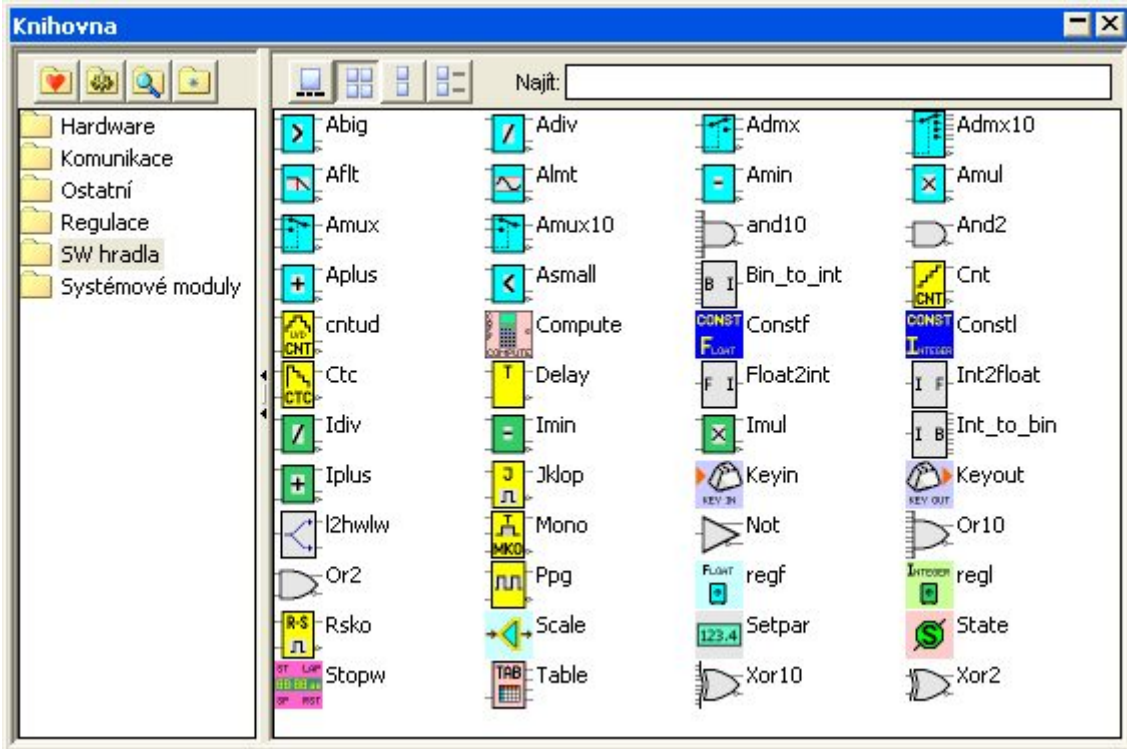
Bit 0 je určen pro neděli, bit 1 pro pondělí, atd.

Čas zapnutí se nastavuje parametry **hodina\_on** a **minuta\_on**.

Čas vypnutí se nastavuje parametry **hodina\_off** a **minuta\_off**.

Standardní parametry modulu: **rychlost** a **priorita**.

## 6 Skupina modulů SW hradel



Obr. 107 Okno knihovny modulů  
Aktivní skupina SW hradla

Moduly softwarových hradel lze rozdělit do skupin podle typu zpracovávaných dat

- **logická hradla**  
vstupní a výstupní data jsou logická
- **celočíslná hradla**  
matematické moduly, pracující s celočíselnou aritmetikou
- **hradla s reálnou aritmetikou**  
matematické moduly, pracující s reálnými čísly

Vyjímkou jsou moduly smíšené, u kterých mají vstupy a výstupy jiný typ. Přesto jsou i tato hradla zařazena do uvedených tří základních skupin. V jejich dalším popisu je tato zvláštnost zdůrazněna.

Všechny knihovní moduly (tedy i hradel) mají parametry **rychlost** a **priorita**.

Nezapojený vstup libovolného modulu (tedy i hradla) má vždy hodnotu 0.

### 6.1 Logická hradla

Vstupní a výstupní data jsou logické stavy 0/1. Kromě standardních logických hradel (barva výpně hradla je bílá) **and2**, **and10**, **or2**, **or10**, **xor2**, **xor10** a **not** zde naleznete i hradlo **delay** (plní funkci filtru) a hradla s funkcemi klopných obvodů **jklop**, **rsko**, **mono** (barva žlutá). Hradla AND, OR a XOR jsou jednak dvouvstupá, jednak desetivstupá.

Mezi logická hradla řadíme rovněž **hradla speciální**, a to hradlo **keyin** pro vyhodnocování stisku klávesy a hradlo **keyout** pro simulaci stisku klávesy, hradlo **ppg** pro generování pulzů, hradlo **setpar** pro obecné nastavení parametrů kteréhokoliv modulu a hradla **regf**, **regl** - registry, paměťové buňky. Řadíme sem i modul **stopw** - stopky a modul **State**, který slouží pro tvorbu stavového automatu.

#### 6.1.1 and2



Modul **and2** je běžným hradlem se dvěma vstupy pro logický součin.



Obr. 106 Značka **and2** na ploše

Jeho vstupy i výstupy nabývají logických hodnot 0/1 podle níže uvedené tabulky.

vstupy		výstupy	
horní A	dolní B	přímý L	nepřímý N
0	0	0	1
1	0	0	1
0	1	0	1
1	1	1	0

Standardní parametry modulu: *rychlost* a *priorita*.

#### 6.1.2 and10



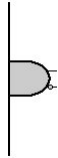
Modul **and10** je běžným hradlem se deseti vstupy pro logický součin.

Jeho vstupy i výstupy nabývají logických hodnot 0/1.

Jedničku na přímém výstupu hradla **and10** dostaneme



pouze v případě, že všechny vstupy budou rovněž jedničkové.



Obr. 114 Značka **and10** na ploše

V opačném případě bude přímý výstup nulový. Negovaný výstup je opakem přímého výstupu. Standardní parametry modulu: *rychlost a priorita*.

### 6.1.3 or2



Modul **or2** je běžným hradlem se dvěma vstupy pro logický součet.



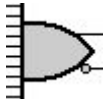
Obr. 113 Značka **or2** v knihovně a na ploše

Jeho vstupy i výstupy nabývají hodnot 0/1 podle níže uvedené tabulky.

vstupy		výstupy	
horní A	dolní B	přímý L	nepřímý N
0	0	0	1
1	0	1	0
0	1	1	0
1	1	1	0

Standardní parametry modulu: *rychlost a priorita*.

### 6.1.4 or10



Modul **or10** je běžným hradlem s deseti vstupy pro logický součet.



Obr. 112 Značka **or10** v knihovně a na ploše

Jeho vstupy i výstupy nabývají binárních hodnot 0/1.

Jedničku na přímém výstupu hradla **or10** dostaneme v případě, je-li jednička alespoň na jednom vstupu hradla.

Nulu na přímém výstupu hradla dostaneme pouze v případě, že všechny vstupy budou nulové.

Negovaný výstup je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.1.5 xor2



Modul **xor2** je logickým hradlem se dvěma vstupy pro vyhodnocení nonekvivalence, nerovnosti vstupů.



Obr. 108 Značka **xor2** v knihovně a na ploše

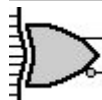
Jeho vstupy i výstupy nabývají binárních hodnot 0/1 podle níže uvedené tabulky.

Rozdílný logický stav vstupů je na přímém výstupu hradla hlášen jako log. 1.

vstupy		výstupy	
horní A	dolní B	přímý L	nepřímý N
0	0	0	1
1	0	1	0
0	1	1	0
1	1	0	1

Standardní parametry modulu: *rychlost a priorita*.

### 6.1.6 xor10



Modul **xor10** je logickým hradlem s deseti vstupy pro vyhodnocení nonekvivalence, nerovnosti vstupů.



Obr. 109 Značka **xor10** na ploše

Jeho vstupy i výstupy nabývají binárních hodnot 0/1.

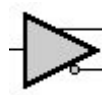
Rozdílný logický stav vstupů je na přímém výstupu hradla hlášen jako log. 1.

Stejný logický stav všech vstupů hradla **xor10** je na přímém výstupu hradla hlášen jako log.0.

Negovaný výstup je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.1.7 not



Modul **not** je logickým hradlem s jedním vstupem, jehož výstup obrací logický stav vstupu.



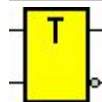
Obr. 110 Značka **not** v knihovně a na ploše

Jeho vstup i výstupy nabývají binárních hodnot 0/1.

Negovaný výstup je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.1.8 delay



Hradlo **delay** plní funkci binárního filtru.

Výstupní úroveň se změní na log.1 v případě, že je na vstupu souvisle log.1 po dobu delší, než je dáno parametrem **timeh** (v ms).



Obr. 111 Značka **delay** v knihovně a na ploše

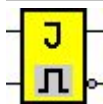
Podobně výstup nabývá log.0 v případě, že je na vstupu souvisle log.0 po dobu delší než **timeh** (v ms).

Praktický rozsah obou parametrů je od 0 do 2147483647 ms.

Negovaný výstup je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

## 6.2 jklop



Modul **jklop** je logickým hradlem s funkcí podobnou klopného obvodu JK.



Obr. 120 Značka **jklop** v knihovně a na ploše

Jeho vstupy i výstupy nabývají binárních hodnot 0/1.

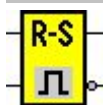
S náběžnou hranou hodinového vstupu **CLK** se přepíše stav na vstupu **DAT** do vnitřního stavu klopného obvodu.

Se sestupnou hranou hodinového vstupu **CLK** se vnitřní stav přepíše na výstup **L**.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.2.1 rsko



Modul **rsko** je logickým hradlem s funkcí klopného obvodu RS.

Jeho vstupy i výstupy nabývají binárních hodnot 0/1.

Pokud je vstup **S** v log.1, výstup **L** je log.1.

Pokud je vstup **R** v log.1 výstup **L** je log. 0.

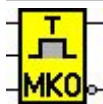


Obr. 115 Značka **rsko** na ploše

Negovaný logický výstup hradla je opakem přímého výstupu **L**.

Standardní parametry modulu: *rychlost a priorita*.

### 6.2.2 mono



Modul **mono** je logickým hradlem s funkcí monostabilního klopného obvodu.



Obr. 116 Značka **mono** v knihovně a na ploše

Jeho vstupy **A**, **B** a výstupy nabývají binárních hodnot 0/1.

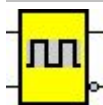
S náběžnou hranou logického součtu vstupů (**A** or **B**) přejde výstup **L** do log.1 na dobu danou parametrem **time** (v ms). Praktický rozsah parametru je od 0 do 2 147 483 647 ms.

Dobu danou parametrem **time** lze korigovat hodnotou, přivedenou na prostřední vstup hradla (od verze SW ccpu 2.1).

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.2.3 ppg



Modul **ppg** je logickým hradlem, které generuje nastavený počet pulzů.



Obr. 118 Značka **ppg** v knihovně a na ploše

Vyslaný počet pulzů je dán součtem hodnoty parametru **numpar** a celočíselné hodnoty, přivedené na vstup **N**.

Doba trvání pulzu na úrovni H (high - v logické 1) je dána součtem hodnoty parametru **highpar** a celočíselné hodnoty, přivedené na vstup **H**.

Doba trvání pulzu na úrovni L (low - v logické 0) je dána součtem hodnoty parametru **lowpar** a celočíselné hodnoty, přivedené na vstup **L**.

Generování pulzů začíná s náběžnou hranou na vstupu **TRG** a trvá po dobu aktivního vstupu **TRG** (v logické 1) do doby, dokud není požadovaný počet pulzů napočítán.

Při nastavení **N=numpar=0** výstup modulu **ppg** pulzuje stále.

Výstup **END** přejde do logické 1 po napočítání požadovaného počtu pulzů.

Standardní parametry modulu: *rychlost a priorita*.

*Poznámka:*

*Doby trvání se neudávají v ms, jsou to počty průchodů programovou smyčkou, do které je modul ppg zařazen.*

*Tzn., že pokud je u modulu ppg parametr rychlost=3 a parametry highpar=lowpar=4, jsou generovány pulzy 400 ms v log.1 a 400 ms v log.0.*

### 6.2.4 keyin



Modul **keyin** slouží k vyhodnocování stisku definované klávesy.

Vstup **Key** musí být napojen na výstup **Key** modulu **CANTERM/CKDM11/CKDM12**.



Obr. 117 Značka **keyin** v knihovně a na ploše

Vstup **Menu** lze připojit na modul **SCREEN** - tak jako u jiných modulů.

Je-li tento pin aktivní, znamená to, že uživatel vstoupil do příslušného menu / podmenu daného modulu **SCREEN**, ke kterému je přiřazeno vyhodnocování stisku definované klávesy modulem **keyin**.

Pokud je vstup **Menu** napojen na konstantu = 1, pak je testován kód klávesy nezávisle na zvoleném (aktivním) menu (modulu **SCREEN**).

Testován je kód klávesy podle parametru **keycode**.

Standardní parametry modulu: *rychlost a priorita*.

Výstup modulu **keyin** je = 1 pouze tehdy, je-li vstup **Menu** aktivní (=1) a byla-li stisknuta klávesa s kódem rovným parametru **keycode**.

### 6.2.5 keyout



Modul **keyout** slouží k simulaci stisku definované klávesy.

Vstup -> je logickým vstupem. Náběžnou hranou vydává výstup **Key** kód klávesy dle hodnoty parametru **keycode**.



Obr. 119 Značka **keyout** v knihovně a na ploše

Výstup **Key** musí být napojen na vstup **Key** modulu **CANTERM/CKDM11/CKDM12**.

Kód klávesy je zadáván jako hodnota do parametru **keycode**.

Standardní parametry modulu: *rychlost a priorita*.

## 6.2.6 setpar

Modul **setpar** slouží pro obecné nastavení parametrů v kterémkoliv modulu. Tento modul není obsažen ve FW malé centrály (s FLASH 256 kB).



Obr. 125 Značka modulu **setpar** na ploše

Lze tak editovat parametr v modulu daném parametry **object** a **instance**.

Parametr **item** odpovídá onomu parametru, který chceme editovat v daném objektu / modulu.

Příslušné hodnoty itemů naleznete ve FREDovi, a to v okně *Informace o modulu* u každého modulu - položka *ident*.

*Např. pro parametr posuv u modulu ekviterm, který byl v projektu na plochu schéma položen jako první, tj. má instanci 1, platí:*

*object=151, instance=1, item=6*

Editujeme hodnotu ze vstupu **In** modulu **setpar** (může být jak typu float, tak typu long integer).

Hodnota se kopíruje na výstup **Out** modulu **setpar** (ale pozor v ladícím režimu FREDa se zobrazuje vždy jako long integer, přitom ji však lze napojit na vstup dalšího modulu, vyžadujícího typ float).

Aktivací vstupu **Block** modulu **setpar** se blokuje zápis parametru, **čtení je nadále funkční** (= > GETPAR).

Dále

- při zápisu hodnoty **itemu** do nového **objectu** se přepočítá kontrolní součet

- zápis se provede při zjištění rozdílu editovaného parametru a vstupu **In** (a neaktivního **Blocku**)

Standardní parametry modulu: *rychlost a priorita*.

Protože by chybná změna parametrů OBJECT, INSTANCE a ITEM v ladícím režimu mohla způsobit až pád aplikace, je editace těchto parametrů v ladícím režimu zakázána.

## 6.2.7 regf

**FLOAT** Modul **regf** plní funkci registru (paměťové buňky) pro hodnotu typu float.

Hodnota ze vstupu **In** při aktivním vstupu **Trig** (aktivní je úroveň 1, ne hrana) je zaregistrována a předána na výstup **Out**.



Obr. 121 Značka modulu **regf** na ploše

Hodnota se pamatuje i po restartu a není chráněna kontrolním součtem.

Parametr **hodnota** definuje také stav výstupu **Out** po překladi.

Standardní parametry modulu: *rychlost a priorita*.

## 6.2.8 regl

**INTEGER** Modul **regl** plní funkci registru (paměťové buňky) pro hodnotu typu long integer.



Hodnota ze vstupu **In** při aktivním vstupu **Trig** (aktivní je úroveň 1, ne hrana) je zaregistrována a předána na výstup



Obr. 122 Značka modulu **regl** na ploše

**Out.**

Hodnota se pamatuje i po restartu a není chráněna kontrolním součtem.

Parametr **hodnota** definuje také stav výstupu **Out** po překladi.

Standardní parametry modulu: *rychlost a priorita*.

## 6.2.9 stopw

**ST LAP** Modul **stopw** jsou v podstatě klasické vteřinové stopky.

**SP RST** Logické vstupy slouží pro jejich ovládání.

Jedničkový impulz na vstup **Strt** spouští stopky, na vstup **Stp** je zastavuje, na vstup **Rst** je nuluje.

Pokud je na vstupu **Lap** jednička, stopky běží, ale hodnota výstupů je zmrazena.



Obr. 123 Značka modulu **stopw** na ploše

Výstupy jsou typu integer. Při dalším zpracování SW hradly s tím proto počítejte a případně použijte převodní hradla.

Naměřený čas je předán na výstupy **Hr** (hodiny), **Min** (minuty) a **Sec** (vteřiny).

Výsledný naměřený čas je pak obvykle převáděn do celkového tvaru Hr:Min:Sec.

Na spodním výstupu **sumaSec** je celkový čas pouze ve vteřinách.

*Příklad:*

*Pokud lze celkový změřený čas zapsat ve tvaru 00:10:10 je na spodním výstupu hodnota 610 (vteřin).*

Standardní parametry modulu: *rychlost a priorita*

Parametr **resinit** určuje, zda po restartu centrály bude dříve naměřený čas snulován nebo ne.

Při **resinit** = 0 čas není nulován, ale mají-li stopky ihned dál automaticky měřit čas, je třeba je po restartu centrály znovu spustit.

## 6.2.10 State

**S** Modul **State** slouží pro vytváření stavového automatu. Jeden modul **State** odpovídá jednomu stavu automatu.



Obr. 124 Značka modulu **State** na ploše

Vstupy **I0..7** se napojují na ty stavy automatu, ze kterých je možné přecházet do daného stavu.

Vstupy **I0..7** nesmí být zapojeny nikam jinam než na výstupy **O0** modulu **State**.

Vstup **SET** slouží pro nastavení aktivního stavu modulu **State**. Vstup **RST** slouží pro nastavení neaktivního stavu modulu **State**.

Výstupy **00** se napojí na vstupy **I0..7** stavu, do kterých lze přecházet ze současného stavu automatu.

Vstup **INP** je pak vstupní hodnota automatu.

V případě, že na vstupu **INP** se objeví integer číslo, rovné podmínce přechodu do současného stavu, dané parametry **state0..7** a je zároveň aktivní příslušný předchozí stav, dojde k přechodu do daného stavu a současně je předchozí stav vynulován. Aktivní stav je indikován výstupem **flag** nastaveným na hodnotu parametru **stnumber**.

Parametry **mask0..7** umožňují maskování (bitové porovnávání) vstupní hodnoty a parametru **state0..7**.

Výchozí hodnota **-1** znamená porovnávat všechno.

*mask1 = 5 například způsobí, že se budou porovnávat pouze 0-tý a 2. bit vstupní hodnoty s parametrem state1, a ostatní bity budou ignorovány.*

Prakticky to umožní přechod do daného stavu za různých situací (některé bity musí mít požadovanou hodnotu a na jiných nezáleží).

*Poznámka:*

*Příklad použití modulu State je v ZIP balíčku pro UpG FW 3.003 (projekt progra2.txt).*

## 6.3 Celočíslná hradla

Jsou to matematické moduly, pracující s celočíselnou aritmetikou.

Celočíselné odečítání lze provádět hradlem **imin**, sčítání hradlem **iplus**, dělení hradlem **idiv** a násobení hradlem **imul** (barva výplně hradel je zelená).

Hradlo **table** (bílé) má funkci tabulky, kdy hodnota vstupu určuje sloupec tabulky, jehož 4 hodnoty jsou vedeny na výstupy hradla.

Hradlo **cnt** (bílé) má funkci akumulátoru hodnoty čítače, hradlo **ctc** (žluté) je časovač.

Pro definici pevné celočíselné hodnoty pro vstup těchto hradel slouží modul **const1** (modrý).

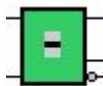
Mezi tato hradla řadíme ještě **převodní hradla** (šedá), a to hradlo **bin\_to\_int**, které převádí 16 binárních vstupů do jednoho celočíselného výstupu a hradlo **int\_to\_bin** s opačnou funkcí. Hradlo **int2float** převádí číslo typu *integer* na *float* a naopak hradlo **float2int** převádí číslo typu *float* na *integer*. Hradlo **l2hwlw** převádí 4 bajt ze vstupu A (long integer) na dva výstupy jako dva "dvojbyty".

*Poznámka:*

*Celočíselná hradla imin, iplus, idiv, imul mají funkci obdobnou hradlům amin, aplus, adiv, amul, a to pouze s tím rozdílem, že pracují s celočíselnou aritmetikou. Nelze je tedy připojovat na vstupy či výstupy s aritmetikou reálnou. Lze je však připojit na vstupy i výstupy logické (binární), za předpokladu že aritmetická 1 odpovídá log.1 (stav H) a aritmetická 0 odpovídá log.0 (stav L).*

*Jiné celočíselné hodnoty nemají z pohledu logických (binárních) smysl a vedou k chybným výsledkům.*

### 6.3.1 imin



Modul **imin** je hradlem s celočíselnou aritmetikou pro odečítání dvou vstupních celých čísel.

Od hodnoty horního vstupu A se odečte hodnota dolního vstupu B.

$$A - B = V$$

Výsledek (celočíslná hodnota) je na výstupu V.



Obr. 127 Značka **imin** v knihovně a na ploše

#### Logické výstupy:

Logické výstupy hradla, vyhodnocují hodnotu výstupního celého čísla (na výstupu **V**). Slouží pro porovnávání vstupních hodnot.

*Poznámka:*

*pro porovnávání logických stavů využijte hradlo XOR.*

Je-li hodnota výsledku **V** rovna nule, je aktivní (v 1) výstup se stejným označením **V=0**.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního čísla, zda je  $V > 0$ .

Je-li hodnota výsledku V větší než nula, je logický výstup (označen jako **V>0**) roven log.1. Je-li hodnota výsledku **V** nula nebo menší než nula, je tento logický výstup roven logické 0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.2 iplus



Modul **iplus** je hradlem s celočíselnou aritmetikou pro sečítání dvou vstupních celých čísel.



Obr. 126 Značka **iplus** v knihovně a na ploše

Hodnota horního vstupu A se sečte s hodnotou dolního vstupu B.

$$A + B = V$$

Výsledek (celočíslná hodnota) je na výstupu V.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního čísla, zda je  $V > 0$ .

Je-li hodnota výsledku V větší než nula, je logický výstup (označen jako **V>0**) roven log.1.

Je-li hodnota výsledku V nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.3 idiv



Modul **idiv** je hradlem s celočíselnou aritmetikou pro dělení dvou vstupních celých čísel.

Hodnota horního vstupu A se dělí hodnotou dolního vstupu B.

$$A / B = V$$

Výsledek (celočíslná hodnota) je na výstupu V.



Obr. 128 Značka **idiv** na ploše

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního čísla, zda je  $V > 0$ .



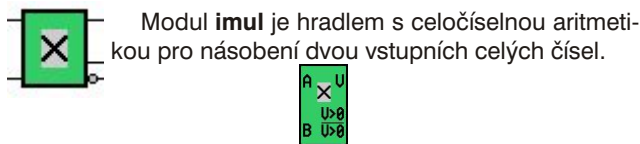
Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.4 imul



Obr. 132 Značka **imul** v knihovně a na ploše

Hodnota horního vstupu  $A$  se násobí hodnotou dolního vstupu  $B$ .

$$A * B = V$$

Výsledek (celočíslná hodnota) je na výstupu  $V$ .

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního čísla, zda je  $V>0$ .

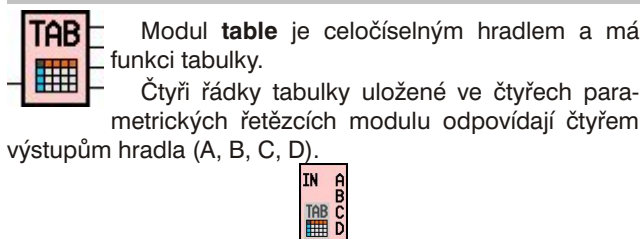
Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.5 table



Obr. 130 Značka **table** v knihovně a na ploše

Hodnota vstupu  $IN$  určuje sloupec tabulky (rozsah 0 až 31).

Definice řetězců je přístupná v dialogovém okně modulu a v něm po stisku tlačítka DIALOG MODULU.

Řetězce parametrů modulu definujeme v řádcích **line\_0**, **line\_1**, **line\_2** a **line\_3**, a to jedním znakem, tj. zapisujeme je bez mezer.

Lze zadávat cifry **0** až **9**, které předávají hodnotu 0 až 9.

Dále lze zadávat písmena **a** až **z** nebo **A** až **Z** (identické zadání), která předávají hodnoty 10 až 35.

Jakýkoliv jiný znak předává hodnotu 0.

Platné hodnoty vstupu  $IN$  jsou 0 = první sloupec tabulky, ... až 31 = 32. sloupec tabulky.

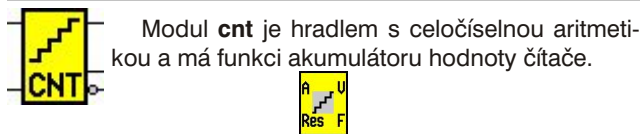
Vstup  $IN$  funguje modulo 32, tzn., že hodnota 32 odpovídá znovu 0, 33 pak 1, ...

Standardní parametry modulu: *rychlost a priorita*.

*Příklad:*

Pokud je na vstupu  $IN$  hodnota 4, modul vybere pátý sloupec tabulky a hodnoty z jednotlivých řádků (vždy páté znaky v rozsahu 0..35) předá na příslušné výstupy ( $A, B, C, D$ ).

### 6.3.6 cnt



Obr. 129 Značka **cnt** v knihovně a na ploše

Modul **cnt** je hradlem s celočíselnou aritmetikou a má funkci akumulátoru hodnoty čítače.

Testuje hodnotu na vstupu  $A$  a přírůstky této hodnoty přičítá k hodnotě výstupní  $V$ .

Vstup **Res** nuluje výstupní hodnotu.

Výstup  $f$  odevzdává načtený počet přírůstků vstupní hodnoty  $A$  za dobu danou násobkem parametru **freqper** (implicitně=100) a délkou zvolené periody vykonávání modulu parametrem **rychlost**.

Standardní parametry modulu: *rychlost a priorita*.

Parametr **max delt** určuje maximální změnu vstupní hodnoty, která je započítána. Pokud se vstupní hodnota nová liší od vstupní hodnoty staré o více než hodnot v parametru **max delt** (např. po výpadku a následném resetu čítače v externí jednotce) není tento zřejmě chybný přírůstek započítán do výstupní hodnoty. Implicitně = 10000.

Parametr **initval** slouží pro definici výstupní hodnoty  $V$  po překladu projektu. Toho využijete při vícenásobném nebo opakovaném pozdějším překladu projektu pro zachování již načtených hodnot do akumulátoru čítače. Implicitně = 0.

Je-li parametr **exflash** = 1, pak se při obnovení aplikačního programu ze záložní paměti Flash nepřepíše stav čítače tohoto modulu.

*Poznámka:*

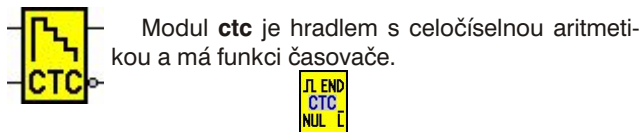
Zvolíme-li **rychlost=2** (tj. perioda vykonávání 1000 ms = 1 sec) a **freqper=1**, bude na výstupu  $f$  hodnota, odpovídající načtení přírůstků vstupu  $A$  za 1000 ms (1 sec), což odpovídá změření frekvence vstupních pulzů.

Zvolíme-li **rychlost=3** (tj. perioda vykonávání modulu 100ms) a **freqper=100**, bude na výstupu  $f$  hodnota, odpovídající načtení přírůstků vstupu  $A$  za 10 000 ms.

Modul **cnt** je určen i pro realizování funkce vodoměru, elektroměru nebo počítadla motohodin.

Na vstup  $A$  se připojí buďto "binární" pulzy z čidla (vodoměr, elektroměr) nebo čítací výstup **Cnt0...3** z jednotky **CBI-11**, jejíž vstup je pak připojen na příslušné čidlo (pro větší frekvence). Modul **cnt** pak spočítá sumu pulzů (výstup  $V$  - objem, spotřebovaná energie), odolný proti výpadku napájení, a okamžitou spotřebu (průtok, el. výkon) na výstupu  $f$ . Potřebné přepočty proveďte podle aktuální potřeby dalšími hradly.

### 6.3.7 ctc



Obr. 131 Značka **ctc** v knihovně a na ploše

Log.1 na vstupu **NUL** načte do vnitřního čítače hodnotu danou parametrem **reldtime** v ms (přesněji v 1/1024s).

Log.1 na vstupu **TRIG** (horní) uvolní čítání - dekrementaci čítače.

Při podtečení čítače, tj. po uplynutí času daného parametrem **reldtime** (od poslední aktivace vstupem **NUL**), se čítání zastaví a výstup **END** se nastaví do log.1.

Během čítání je výstup **END** ve stavu log.0.

Negovaný logický výstup (**notL**) je opakem přímého výstupu **END**.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.8 constl



Modul **constl** slouží pro definici pevné hodnoty celého čísla (**konstanta integer**) pro vstup hradel s celočíselnou aritmetikou, obecně pro zavedení hodnoty celého čísla do knihovnic modulů s příslušným typem vstupu.

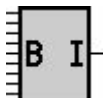


Obr. 138 Značka **CONSTl** na ploše

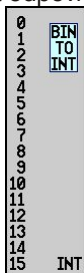
Hodnota této konstanty je daná parametrem **hodnota** (a je typu **LONG** v rozsahu -2147483648 až 2147483647).

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.9 bin\_to\_int



Modul **bin\_to\_int** je hradlem, které zpracovává smíšená data a převádí 16 binárních vstupů do jednoho celočíselného výstupu, kde nejnižší bit výstupu se rovná hodnotě vstupu 0 a postupně další bity výstupního čísla odpovídají stavu dalších vstupů.

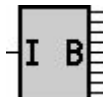


Obr. 136 Značka **bin\_to\_int** v knihovně a na ploše

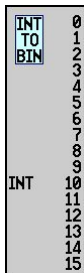
Váhy jednotlivých bitů: postupně 1, 2, 4, 8, 16, 32, ...

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.10 int\_to\_bin



Modul **int\_to\_bin** je hradlem, které zpracovává smíšená data a má opačnou funkci oproti hradlu **bin\_to\_int**.



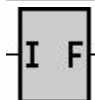
Obr. 135 Značka **int\_to\_bin** na ploše

Převádí 16 bitů celého čísla na 16 oddělených binárních výstupů, kde jednotlivé binární výstupy odpovídají postupně jednotlivým binárním vahám celočíselného vstupu.

Váhy jednotlivých bitů: postupně 1, 2, 4, 8, 16, 32, ...

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.11 int2float



Modul **int2float** slouží k převodu vstupní hodnoty celočíselné (integer) na výstupní reálné číslo (float).



Obr. 134 Značka **int2float** v knihovně a na ploše

Modul je čtyřnásobný, každému vstupu přísluší výstup.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.12 float2int



Modul **float2int** slouží k převodu vstupního reálného čísla (float) na výstupní celočíselnou hodnotu (integer).



Obr. 133 Značka **float2int** v knihovně a na ploše

Modul je čtyřnásobný, každému vstupu přísluší výstup.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.13 cntud



Modul **cntud** je hradlem s funkcí čítače nahoru / dolů s možností nastavit výchozí hodnotu parametrem **hodnota**.

Horní mez čítání je dána součtem hodnoty parametru **hilimpar** a hodnoty na vstupu **HiLim**.

Dolní mez čítání je dána součtem hodnoty parametru **lolimpar** a hodnoty na vstupu **LoLim**.



Obr. 137 Značka modulu **cntud** na ploše

Výstup **HiLim** indikuje jedničkou dosažení horní meze čítání.

Výstup **LoLim** indikuje jedničkou dosažení dolní meze čítání.

Na výstup  $\rightarrow$  (šipka) je posílána aktuální hodnota čítání. Vstup **SetHi** umožňuje nastavení výstupu  $\rightarrow$  na horní mez.

Vstup **SetLo** umožňuje nastavení výstupu  $\rightarrow$  na dolní mez.

Jsou-li oba tyto vstupy aktivní (v 1) má vstup **SetLo** přednost před vstupem **SetHi**.

Hradlo **cntud** čítá vzestupnou hranou na vstup **Ud** nahoru a na vstup **Do** dolů.

Hodnota čítání se pamatuje i po restartu a není chráněna kontrolním součtem.

Standardní parametry modulu: *rychlost a priorita*.

### 6.3.14 I2hwlw

Modul **I2hwlw** (long to high word, low word) je celočíselným hradlem, které převádí vstupní hodnotu **A** (4bajtový longinteger) na dva "dvoubajtové" výstupy, a to **HW** (high word, horní) a **LW** (low word, dolní).



Obr. 143 Značka **I2hwlw** na ploše

Modul je např. vhodný pro přepočet přijatých dat modulem **pwpb\_rx**, kdy je třeba přijatý čtyřbajt rozdělit.

Parametr **sigext**, rozšiřuje výstupní hodnoty na výstupech LW/HW o znaménko:

při **sigext=0** a vstupu **A=0xFFFF** je výstup **LW=65535**

při **sigext=1** a vstupu **A=0xFFFF** je výstup **LW=-1**

Standardní parametry modulu: *rychlost a priorita*.

## 6.4 Hradla s reálnou aritmetikou

Jsou to matematické moduly, pracující s reálnými čísly. Barva výplně těchto hradel je světle modrá.

Tak můžeme realizovat základní matematické operace - násobení (**amul**), dělení (**adiv**), sčítání (**aplus**) a odečítání (**amin**).

Další hradla vybírají ze dvou vstupních reálných čísel větší (**abig**) nebo menší (**asmall**) z nich.

Jiné zase slouží jako přepínač dvou nebo deseti reálných hodnot podle stavu vstupu (**amux**, **amux10**), další posílá na výstup buď 0 nebo vstupní reál. číslo podle stavu logického vstupu (**adm**, **adm10**).

Hradlo **almt** je omezovač s dolním i horním limitem.

Hradlo **aflt** je digitální filtr prvního řádu (dolní propust).

Pomocí modulu **compute** lze zadat rovnici, která pracuje se 4 vstupními hodnotami.

Pro definici pevné hodnoty reálného čísla pro vstup těchto hradel slouží modul **constf**.

### 6.4.1 amul

Modul **amul** je hradlem s reálnou aritmetikou pro násobení dvou vstupních reálných čísel.



Obr. 141 Značka **amul** v knihovně a na ploše

Hodnota horního vstupu **A** se násobí hodnotou dolního vstupu **B**.

$$A * B = V$$

Výsledek (hodnota reálného čísla) je na výstupu **V**.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního reálného čísla, zda je  $V > 0$ .

Je-li hodnota výsledku **V** větší než nula, je logický výstup (označen jako **V>0**) roven log.1.

Je-li hodnota výsledku **V** nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.2 adiv

Modul **adiv** je hradlem s reálnou aritmetikou pro dělení dvou vstupních reálných čísel.

Hodnota horního vstupu **A** se dělí hodnotou dolního vstupu **B**.



Obr. 140 Značka **adiv** v knihovně a na ploše

$$A / B = V$$

Výsledek (hodnota reálného čísla) je na výstupu **V**.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního reálného čísla, zda je  $V > 0$ .

Je-li hodnota výsledku **V** větší než nula, je logický výstup (označen jako **V>0**) roven log.1.

Je-li hodnota výsledku **V** nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.3 aplus

Modul **aplus** je hradlem s reálnou aritmetikou pro sečítání dvou vstupních reálných čísel.

Hodnota horního vstupu **A** se sečte s hodnotou dolního vstupu **B**.

$$A + B = V$$



Obr. 139 Značka **aplus** na ploše

Výsledek (hodnota reálného čísla) je na výstupu **V**.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního reálného čísla, zda je  $V > 0$ .

Je-li hodnota výsledku **V** větší než nula, je logický výstup (označen jako **V>0**) roven log.1.

Je-li hodnota výsledku **V** nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.4 amin

Modul **amin** je hradlem s reálnou aritmetikou pro odečítání dvou vstupních reálných čísel.



Obr. 142 Značka **amin** na ploše

Od hodnoty horního vstupu **A** se odečte hodnota dolního vstupu **B**.

$$A - B = V$$

Výsledek (hodnota reálného čísla) je na výstupu **V**.

Logické výstupy:

Logické výstupy hradla vyhodnocují hodnotu výstupního reálného čísla (na výstupu **V**). Slouží pro porovnávání vstupních hodnot.

Je-li hodnota výsledku  $V$  rovna nule, je aktivní (v 1) výstup se stejným označením  $V=0$ .

Je-li hodnota výsledku  $V$  větší než nula, je logický výstup  $V>0$  roven logické 1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup  $V>0$  roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.5 abig



Modul **abig** je hradlem s reálnou aritmetikou pro výběr většího ze dvou vstupních reálných čísel.

Výsledek (větší hodnota vstupního reálného čísla) je na výstupu  $V$ .

Je-li  $A>B$ , pak  $V=A$ ,

je-li  $B>A$ , pak  $V=B$ .

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního reálného čísla, zda je  $V>0$ .

Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.



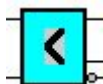
Obr. 147 Značka **abig** na ploše

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.6 asmall



Modul **asmall** je hradlem s reálnou aritmetikou pro výběr menšího ze dvou vstupních reálných čísel.



Obr. 145 Značka **asmall** na ploše

Výsledek (menší hodnota vstupního reálného čísla) je na výstupu  $V$ .

Je-li  $A<B$ , pak  $V=A$ ,

je-li  $B<A$ , pak  $V=B$ .

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního reálného čísla, zda je  $V>0$ .

Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

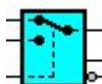
Standardní parametry modulu: *rychlost a priorita*.

### 6.4.7 amux

Modul **amux** je hradlem s reálnou aritmetikou pro výběr jednoho ze dvou vstupních reálných čísel ( $A, B$ ) na výstup hradla  $V$  podle stavu logického vstupu  $X$ .

Je-li  $X=0$ , pak  $V=A$ ,

je-li  $X=1$ , pak  $V=B$ .



Obr. 144 Značka **amux** na ploše

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního reálného čísla, zda je  $V>0$ .

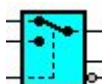
Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

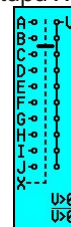
Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.8 amux10



Modul **amux10** je hradlem s reálnou aritmetikou pro výběr jednoho ze deseti vstupních reálných čísel ( $A, B, \dots, J$ ) na výstup hradla  $V$  podle stavu logického vstupu  $X$ .



Obr. 146 Značka **amux10** na ploše

Je-li  $X=0$ , pak  $V=A$ ,

je-li  $X=1$ , pak  $V=B$ ,

je-li  $X=2$ , pak  $V=C$ ,

je-li  $X=3$ , pak  $V=D$ ,

je-li  $X=4$ , pak  $V=E$ ,

je-li  $X=5$ , pak  $V=F$ ,

je-li  $X=6$ , pak  $V=G$ ,

je-li  $X=7$ , pak  $V=H$ ,

je-li  $X=8$ , pak  $V=I$ ,

je-li  $X=9$ , pak  $V=J$ .

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního reálného čísla, zda je  $V>0$ .

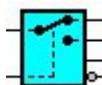
Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

### 6.4.9 admx



Modul **admx** je hradlem s reálnou aritmetikou, které prepíná hodnotu (reálného čísla) ze vstupu  $I$



Obr. 148 Značka **admx** na ploše



na jeden z výstupů hradla (A, B) podle stavu logického vstupu X.

Je-li  $X=0$ , pak  $A=I$  a  $B=0$ ,

je-li  $X=1$ , pak  $B=I$  a  $A=0$ .

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního reálného čísla (na vstupu I), zda je větší než 0.

Je-li hodnota  $I>0$ , je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota  $I=0$  nebo menší než nula, je logický výstup (označen jako  $V>0$ ) roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

#### 6.4.10 admx10

Modul **admx10** je hradlem s reálnou aritmetikou, které přepíná hodnotu (reálného čísla) ze vstupu **In** na jeden z výstupů hradla **A..J** podle stavu integer vstupu **X**, přitom ostatní výstupy jsou nulové.



Obr. 151 Značka modulu **admx10** na ploše

Je-li  $X=0$ , pak  $A=In$  a  $B..J=0$ ,

je-li  $X=1$ , pak  $B=In$  a ostatní výstupy jsou nulové,

je-li  $X=2$ , pak  $C=In$  a ostatní výstupy jsou nulové, atd.

**Poznámka:**

Hradlo lze použít obecně pro přepínání jakýchkoliv hodnot (nejen typu float).

V ladícím režimu FREDa je hodnota z výstupního pinu A..J připojeného na label v něm zobrazována podle definice typu hodnoty (float/int/long).

Logický výstup hradla (přímý), vyhodnocuje hodnotu vstupního čísla (na vstupu **In**), zda je větší než 0.

Je-li hodnota  $In>0$ , je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota  $In=0$  nebo menší než nula, je logický výstup (označen jako  $V>0$ ) roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

#### 6.4.11 almt

Modul **almt** je hradlem s reálnou aritmetikou a funkcí omezovače hodnoty na výstupu hradla.



Obr. 150 Značka **almt** na ploše

Výstupní hodnota je omezena na maximální hodnotu danou parametrem **hlimit** a na minimální hodnotu danou parametrem **llimit**.

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního reálného čísla, zda je  $V>0$ .

Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

Standardní parametry modulu: *rychlost a priorita*.

#### 6.4.12 aflt

Modul **afit** je hradlem s reálnou aritmetikou a funkcí digitálního filtru prvního řádu (dolní propust) pro přivedené hodnoty (reál.č.) na vstupu A.



Obr. 149 Značka **afit** na ploše

Výstupní filtrovaná hodnota je vyvedena na výstup V.

Časová konstanta je daná parametrem **tconst** v sekundách, parametr **slewwrate** (lze zadávat na tři platná desetinná místa) určuje maximální rychlost přeběhu výstupního signálu v 1/s.

Parametr **filtdelay** slouží k definici doby (v ms), po kterou je po inicializaci modulu (tj. po resetu či zapnutí regulátoru) potlačena činnost filtru (vstup je přepouštěn na výstup bez filtrace).

**Poznámka:**

Tato doba zahrnuje inicializaci celého regulátoru PL2 (včetně CAN sběrnice), proto jsou prakticky použitelné hodnoty cca nad 3000 ms.

Použití v praxi zejména tehdy, když požadujete "pomalý" výstup za filtrem, ale pro počáteční regulaci potřebujete reálnou hodnotu.

Pokud **filtdelay=0** je filtr zařazen ihned po překladu, resetu, zapnutí regulátoru ...

Logický výstup hradla (přímý), vyhodnocuje hodnotu výstupního reálného čísla, zda je  $V>0$ .

Je-li hodnota výsledku  $V$  větší než nula, je logický výstup (označen jako  $V>0$ ) roven log.1.

Je-li hodnota výsledku  $V$  nula nebo menší než nula, je logický výstup roven log.0.

Negovaný logický výstup hradla je opakem přímého výstupu.

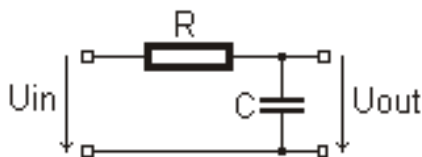
Standardní parametry modulu: *rychlost a priorita*.

**Definice:**

Modul **afit** realizuje filtr *dolní propust prvního řádu*. Na výstup filtru je dále připojen omezovač rychlosti přeběhu výstupního signálu (je součástí modulu **afit**).

Činnost filtru je ekvivalentní elektrickému obvodu "RC integrační článek", kde hodnota na vstupu modulu **afit** odpovídá napětí **Uin** (viz obrázek), a hodnota výstupní odpovídá

napětí **U<sub>out</sub>**. Chování RC filtru na obrázku je definováno časovou konstantou **tau**, kde **tau = R\*C**.



Časová konstanta **tconst** modulu **aflt** je ekvivalentní časové konstantě **tau** článku RC, čili např. modul **aflt** s parametrem **tconst = 1s** se chová stejně jako RC článek se stejnou časovou konstantou, tedy např. **R= 1MΩ, C= 1μF**.

Rozdíl oproti analogovému obvodu je daný pouze principem digitálního zpracování signálu, zejména tedy periodou výpočtu (viz údaje o rychlostech smyček.)

Prakticky to znamená, že modul **aflt** ve smyčce s parametrem **rychlost=3** (to neplatí pouze pro modul **aflt**, ale obecně pro všechny moduly) může zpracovávat vstupní signály s max. frekvencí 5Hz. Pro ostatní smyčky je tato frekvence úměrně nižší. Pokud je třeba filtrovat ještě vyšší frekvence, je možné použít např. filtr v jednotce **CAIO**, který je schopen úspěšně filtrovat až do frekvence 50Hz.

Rovnice výpočtu analog filtru:

$OUT_{nový} = OUT_{starý} + (IN - OUT_{starý}) / (tconst / tper)$ , kde **tconst** je parametr modulu **aflt**, a **tper** je perioda výpočtu modulu.

U takto spočteného výstupu je provedeno omezení rychlosti přeběhu (**slewrates**).

Pokud není potřeba omezení rychlosti přeběhu, je potřeba dát parametr **slewrates** větší než desetinásobek největší očekávané hodnoty na vstupu modulu **aflt**.

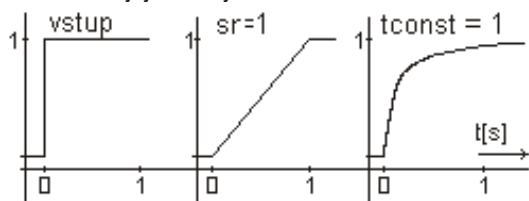
Rychlost přeběhu je dána výrazem

$$SR = \Delta Out / \Delta T,$$

kde **delta Out** je změna výstupní hodnoty modulu **aflt** a **delta T** je perioda výpočtu modulu (odpovídá rychlosti smyčky).

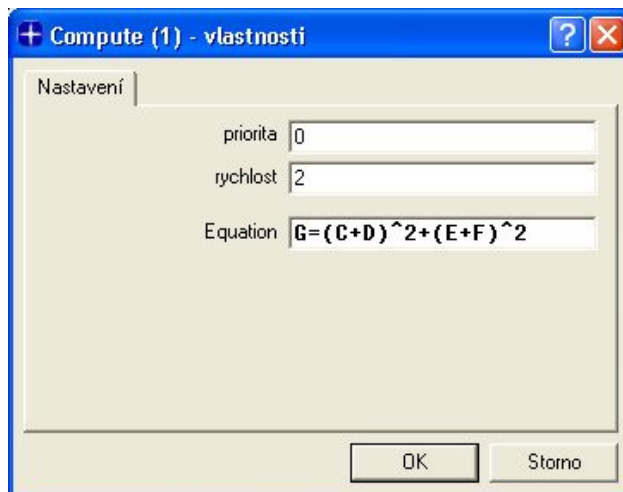
Pokud například máme **slewrates = 10** jednotek za sekundu, a modul **aflt** je ve smyčce č. 2, hodnota na výstupu se mezi dvěma po sobě následujícími výpočty změní maximálně o 10. Pokud byla změna na vstupu větší, dosáhne hodnota na výstupu této hodnoty až po několikátém výpočtu.

Funkce omezení rychlosti je velmi podobná funkci filtrace. Pro ilustraci jejich vzájemného rozdílu viz obrázek.

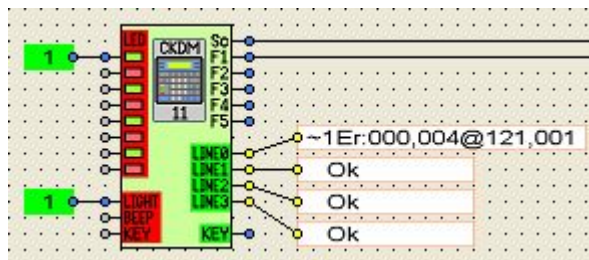


Obecně filtrace dolní propustí je výhodnější při menších amplitudách a nižších frekvencích rušivého signálu (náhodné kolísání), filtrace omezením rychlosti přeběhu je naopak výhodnější při krátkém a intenzivním rušivém signálu (rušivé špičky, krátkodobé výpadky).

Aby modul **aflt** byl stabilní, tak **tconst** musí být > perioda smyčky, tj. např. pro smyčku s parametrem **rychlost = 1** musí být **tconst > 3**.



Obr. 153 Dialogové okno pro zadání rovnice



Obr. 154 Hlášení chyby syntaxe pro objekt 121 tj. pro modul **compute**

### 6.4.13 compute



Modul **compute** je matematický prvek, který pracuje se čtyřmi vstupními hodnotami **C, D, E a F**.



Obr. 152 Značka **compute** na ploše

Hodnota výstupu **G** tohoto modulu je dána rovnicí, která pracuje s uvedenými vstupními hodnotami.

Vlastní rovnici zadejte v dialogovém okně modulu **compute**. Modul **compute** umí pracovat s těmito operátory:

operátor	funkce
+	sčítání
-	odečítání
*	násobení
/	dělení
sin	sinus (hodnota v závorkách a v radiánech)
cos	cosinus (dtto)
atan	arcustangens, tj. $\tan^{-1}$ (dtto)
ln	přirozený logaritmus
^	mocnina
(	levá závorka
)	pravá závorka

Poznámka:

Hodnoty pro funkce **sin, cos, atan** a **ln** doporučujeme zapisovat mezi závorky.

Seznam číselného označení chyb modulu compute (číslo objektu 121):		
chyba	význam	příklad
1	více pravých závorek než levých	$G = (A+B))$
2	nestejný počet levých a pravých závorek	$G = (A+B)*(3*(D)$
3	ve výrazu se vyskytují prázdné závorky	$G = (3*D) + ()$
4	výraz obsahuje identifikátor jiný než C až G	$G = H*3$
5	výraz obsahuje víceznakový identifikátor, který není číslo	$G = d2*C$
6	výraz obsahuje víceznakový identifikátor	$G = \sin(DD)$
7	levá strana výrazu není G	$D = 3*7$

Tyto hodnoty se zadávají v radiánech - ne ve stupních.

180 stupňů = 3,141592654 v radiánech

(tj. Ludolfovo číslo =  $\pi = 180^\circ$ )

C, D, E, F a G musí být zapsány velkými písmeny.

Ve výrazu smí být maximálně 32 znaků včetně mezer.

Modul **compute** zpracuje maximálně 16 dočasných proměnných výpočtu a 15 instrukcí mikrokódu.

Syntakticky správné výrazy jsou např.:

$$G = (C + \sin(D)) * 3.45$$

$$G = (C+D)^2 - (E/F)$$

$$G = \cos(C) + \ln(D) - (E^3 * 1.4 - F)$$

Chyby syntaxe jsou hlášeny do modulu **sysmon**. Této skutečnosti využijte pro ladění výrazu v ladícím režimu prostředí FRED. Skladba chybového hlášení - viz **sysmon**.

Standardní parametry modulu: *rychlost a priorita*.

#### 6.4.14 constf

**CONST** Modul **constf** slouží pro definici pevné hodnoty reálného čísla (**konstanta float**) pro vstup hradel s reálnou aritmetikou, obecně pro zavedení pevné hodnoty reálného čísla do knihovnic modulů se vstupem typu float (např. PID regulátor, od něhož požadujeme regulovat na konstantní hodnotu - regulace TUV apod.).

**FLOAT**

Obr. Značka modulu CONSTF na ploše

Hodnota této konstanty je daná parametrem **hodnota**.

Standardní parametry modulu: *rychlost a priorita*.

Poznámka:

Rozsah hodnot reálného čísla je +/- 1.84466 x 10<sup>19</sup>.

Hodnotu lze zadávat i ve tvaru 1.2345e-5, což odpovídá 1.2345 \* 10<sup>-5</sup>, tedy 0.000012345.

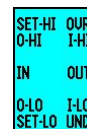
Zobrazení hodnoty v ladícím režimu je dáno formátem labelu [např. 3.2], bude tedy v tomto případě zobrazeno 0.00, ale systém bude pracovat opravdu s námi zapsanou hodnotou. Pokud je nutné zadávat spíše velmi malá čísla, lze formát změnit.

#### 6.4.15 scale



Modul **scale** měřítku hodnotu ze vstupu **In** (float) na výstupní hodnotu **Out** (float).

Rozsah vstupní hodnoty je dán parametry **inhipar** a **inlopar**.



Obr. 155 Značka modulu **scale** na ploše

Rozsah výstupní hodnoty je dán parametry **outhipar** a **outlopar**.

K hodnotám parametru **outhipar** a **outlopar** se pro určité skutečných mezí přičítají ještě hodnoty vstupu **OHi** a **OLo** (take float).

Hodnota výstupní se přepočítává z hodnoty vstupní i pokud jsou tyto mimo zadané meze.

Tento stav je pak indikován výstupy **Ovr** (výstupní hodnota je větší než outhipar + OHi) a **Und** (výstupní hodnota je menší než outlopar + OLo). Tyto výstupy jsou typu long.

Modul také umožňuje editaci rozsahu vstupní hodnoty, a to propojením vstupu **Disp0..3** modulu **SCREEN** s výstupy **IHi**, resp. **ILo** (oba typu float), a povolením editace "přes vstup" parametrem **dispedit** modulu **SCREEN**. Pak je možné zobrazit a editovat hodnoty parametru **inhipar**, resp. **inlopar**.

Další funkcí je kalibrace mezi rozsahu vstupu.

Náběžnou hranou (0->1) na vstupu **SetHi**, resp. **SetLo** se nastaví aktuální hodnota vstupu do parametru **inhipar**, resp. **inlopar**.

Postup kalibrace:

- měřená veličina se nastaví na horní, resp. dolní, rozsah výstupní hodnoty

- aktivací vstupu **SetHi**, resp. **SetLo**, se výstupní hodnota přizpůsobí skutečnosti

V tomto případě není nutné, aby **outhipar** a **outlopar** byly skutečné mezní hodnoty rozsahu, mohou to být dva kalibrační body měřicího zařízení.

Modul **scale** je hlavně určen pro zpracování měřených hodnot z modulu **CAIO12**, které mají vždy nastaven rozsah podle použitého domečku a ten nelze jinak měnit.

## 7 Skupina Systémových modulů



Obr. 157 Okno knihovny - aktivní skupina Systémové moduly

K dispozici jsou tyto systémové moduly:

- **SCREEN**  
pro tvorbu uživatelských menu, zobrazování textu či datových údajů  
návaznost na modul CANTERM/CKDM11/CKDM12
- **sysmon**  
systémový monitor regulátoru PROMOS line 2  
pro zachycování a zobrazení chybových hlášení z modulů havbin, havan, canopen aj.
- **timdat**  
pro nastavení systémového data a času  
návaznost na modul SCREEN
- **hesla**  
pro nastavení přístupových kódů jednotlivých uživatelů a jejich hesel  
návaznost na modul SCREEN
- **vzorky**  
pro definici historické databanky  
návaznost na modul SCREEN
- **flash**  
pro ukládání zálohy konfiguračního programu do paměti FLASH  
návaznost na modul SCREEN
- **canopen**  
pro možnou realizaci chybových hlášení o stavu systémové sběrnice CAN  
možnost vypnutí komunikace na CANu
- **flag**  
pro vyhodnocení a čtení systémových událostí
- **journal**  
pro zobrazení položek historické databanky
- **journal\_txt**  
pro definici textů k položkám hist. databanky
- **printer**  
modul pro ovládání sériové tiskárny
- **HDBmain**  
pro definici historické databanky typu HDM
- **HDBa**  
pro analogové signály (float) HDB
- **HDBb**  
pro binární signály HDM
- **PROGRAM**  
modul pro realizaci programového řízení



## 7.1 SCREEN

**SCREEN** Modul **SCREEN** je určen pro tvorbu uživatelského menu, povelů, zobrazování textu či datových údajů a pracuje v návaznosti na modul **CANTERM**, **CKDM11**, **CKDM12** nebo jiný modul **SCREEN**.



Obr. 158 Značka **SCREEN** na ploše

**In** je vstup modulu a je určen pro připojení na modul **CANTERM/CKDM11/CKDM12** (jeho výstup **Sc** či výstup **F1..5**) nebo na výstup **0..3** jiného modulu **SCREEN**.

Modul **SCREEN**, připojený svým vstupem **In** na výstup **Sc** modulu **CANTERM/CKDM11/CKDM12** se začne vykonávat ihned po zapnutí režimu **RUN** regulátoru (tzv. hlavní menu).

Ostatní moduly **SCREEN** (stromové struktury menu, která začíná tzv. hlavním menu) se budou vykonávat (zobrazí se informace jim příslušející) pokud budou aktivovány výstupem **0..3** jiného **SCREEN** modulu.

Pro rychlejší přístup do často používaných menu nepoužijeme výše uvedené vazby, ale využijeme pro tyto moduly **SCREEN** napojené svým vstupem **In** na výstupy **F1**, **F2**, **F3**, **F4** nebo **F5** modulu **CANTERM / CKDM11 / CKDM12**.

Pak po stisku kláves **F1..5** na ovládacím panelu **CKDM-11** lze aktivovat další "stromové struktury" menu.

Např. na výstup **F1** modulu **CANTERM / CKDM11 / CKDM12** napojujeme pin menu modulu **sysmon**, čímž docílíme výpisu chybových hlášení po stisku klávesy **F1 - ALARM** na ovládacím panelu **CKDM-11**.

Jeden modul **SCREEN** slouží pro definici zobrazení 4 řádků displeje / menu.

Pokud je třeba, aby dané menu bylo více než čtyřřádkové, řadíme pro další 4 řádky další modul **SCREEN** do kaskády, a to pomocí vazby, znázorněné v dolní části modulu výstupním pinem - malou šipkou, který napojujeme na pin **In** dalšího **SCREEN** modulu. Modul **SCREEN**, řazený do kaskády (jako následný), musí mít parametr **enb\_kask=1**. **POZOR** - první modul **SCREEN** v kaskádě musí mít pro správnou funkci klávesy **ESC** parametr **enb\_kask=0**.

Výstupy **0..3** jsou určeny pro přechod do podmenu - menu obsaženého v jiném modulu **SCREEN**, připojeného vstupem **In** na tento výstup.

Např. výstup **0** je aktivován, pokud v menu modulu **SCREEN** je kurzor na prvním řádku a stiskne se klávesa **ENTER** na ovládacím panelu regulátoru. Tím se přejde z menu tohoto modulu **SCREEN** do menu následujícího modulu **SCREEN**, připojeného vstupem **In** na výstup **0**.

Podobně z druhého řádku současného menu lze přejít do menu modulu **SCREEN**, připojeného na výstup **1** ... až z řádku 4. na menu modulu připojeného na výstup **3**.

Na výstupy **0..3** lze také připojit vstupy **Menu** některých dalších modulů, např. regulátorů, a tak vyvolat speciální menu příslušející těmto jednotkám.

Tak lze tvořit z modulů **SCREEN** stromovou strukturu menu, ve kterém se do "hloubky" dá pohybovat výběrem řádku a stiskem **ENTER**, přitom návrat o úroveň výše je vždy možný klávesou **ESC**.

Déletrvajícím stiskem **ESC** v nejvyšším menu stromu (menu modulu připojeném na **CANTERM / CKDM11 / CKDM12**) se přejde do módu **SYSTEM** regulátoru, ve kterém jsou přístupné volby módu regulátoru (**Download**, **Run**, **Test I/O**).

Krátkým stiskem **ESC** v hlavním menu se dostanete do módu přehlídka (funkce **slideshow**) - budou cyklicky zobrazovány údaje modulů **SCREEN** s parametrem **prehlidka=1**.

Vstupy **disp0..3** mají přiřazeny identifikátory **4..7** a slouží pro zobrazení hodnoty.

Výstupy **edit 0..3** mají přiřazeny identifikátory **0..3** a slouží k výstupu hodnoty, editovatelné na terminálu (jejich počáteční hodnota je dána parametry **edit0init ... edit3init**).

Modul **SCREEN** kopíruje (od verze **FW 2.302/3.000**) hodnoty parametrů **edit0init..edit3init** (ident **58..61** v okně **Informace o modulu SCREEN**) do výstupů **edit0..3** (itemy **edit0value..edit3value** s identy **7..10**). To znamená, že displejk by měl nastavovat povel a hodnoty na itemech **edit0init..edit3init**, místo na výstupních **edit0value..edit3value**.

### Formátovací řetězce

Čtyři formátovací řetězce určují samotný obsah každého ze čtyř řádků menu.

Jejich definice je přístupná v dialogovém okně modulu **SCREEN**, a to po kliknutí na tlačítko **DIALOG MODULU**.

Ty pak zadáváme do parametrů **Text\_line\_0 ... 3**.

Text je zobrazen tak, jak je zadán do formátovacího řetězce.

Zápisem **#T** definujeme výpis data a času do celého řádku ve tvaru:

```
12.11.01 Po 17:04:14
```

```
datum | cas (hodiny,minuty,sekundy)
den v tydnu (Po=pondeli, ...)
```

### Formátovací příkazy

Formát zobrazeného čísla definujeme zápisem znaku **#** a desetinné tečky. Např. chceme-li číslo (hodnotu) zobrazovat maximálně ve stovkách a na jedno desetinné místo, bude pro tento požadavek vypadat formátovací příkaz takto:

```
###.#
```

V tomto formátovacím příkazu znak **#** představuje zobrazení jedné cifry. Požadovanou hodnotu (údaj měřený či vypočtený) signálu přivádíme na vstupy **DISP0..3** modulu **SCREEN**. Zobrazení hodnoty z příslušného vstupu definujeme jejím identifikátorem. Např. **###.#6** je příkaz pro zobrazení čísla ve formátu **###.#** z pozice dané identifikátorem **6**. To odpovídá vstupu **DISP2** (identifikátory pro vstupy **DISP0..3** jsou v rozsahu **4..7**). Na místo tohoto formátovacího příkazu se tedy ve skutečnosti zobrazí hodnota ze vstupu **DISP2**.

Standardně jsou na vstupy **DISP0..3** přiváděny hodnoty **FLOAT**. Běžně tedy **INTEGER** čísla musíme nejprve převést modulem **INT2FLOAT** na typ **FLOAT**. Při zobrazování velmi velkých celých čísel však při převodu dochází k "bitové

chybě (nestačí 4bajtové výjádření čísla => nepřesnost). Proto od verze FW 3.020 centrální byl formát rozšířen pro přímé zobrazení velkého celého čísla. Signál s hodnotou velkého celého čísla připojíme přímo (bez převodního modulu INT2FLOAT) na vstup **DISP0..3** a použijeme formát **#####4..7**

(přesně 10 znaků # a identifikátor 4 až 7 pro vstupy **DISP0..3**).

**Formát zadávaného čísla** (pomocí klávesnice ovládacího panelu) definujeme stejně. Představme si, že modul **SCREEN** obsahuje 4 paměťové buňky, pomocí kterých můžeme do projektu (do aplikace) posílat uživatelem nadefinované hodnoty. Hodnoty (obsah) těchto buněk je vyveden na výstupy **EDIT0..3** modulu **SCREEN** a odkazujeme se na ně ve formátovacím příkazu identifikátorem **0..3**. Chceme-li tedy zobrazit obsah této buňky na displeji, použijeme např. formátovací příkaz

**TUV: ##.#1**

čímž zobrazujeme nastavenou (požadovanou) teplotu teplé užitkové vody. Vlastní hodnotu lze po stisku klávesy **INS** editovat a poslat tak do aplikace nově požadovanou hodnotu pro regulaci **TUV**, v tomto případě z výstupu **EDIT1**. Hodnota se zobrazuje a edituje v předem nadefinovaném formátu, schvaluje klávesou **ENTER**.

Od verze FW 3.021 lze tímto způsobem zadávat nejen čísla typu **FLOAT**, ale i typu **LONG\_INTEGER**, a to tehdy, použijeme-li formát zápisu

**#####0..3**

(opět právě 10 znaků # a identifikátor 0 až 3 pro výstupy **EDIT0..3**). Příslušný výstup **EDIT0..3** je pak typu **integer**, proto hodnotu v ladícím režimu sledujte až za hradlem s celočíselnou aritmetikou.

Pokud je zobrazovaná hodnota mimo rozsah daný formátovacím příkazem, zobrazí se místo čísla znaky **E**, a to dle nadefinovaného formátu čísla (např. **EEE.EE** pro formát **###.##**).

## Povely

Jako formátovací příkaz lze dále použít direktivu **\$**, která slouží pro **POVELY**.

Nastavuje výstupy **edit0..3** podle stisku kláves **+** (do 1) nebo **-** (do 0).

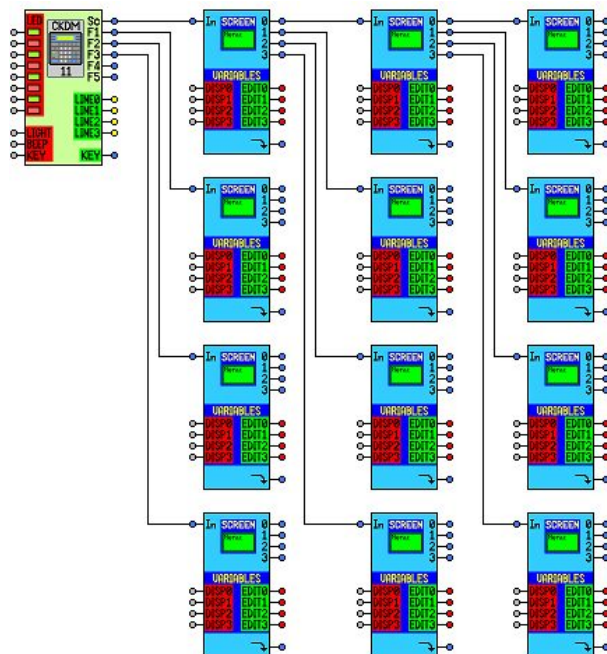
Nastavován je výstup, odpovídající aktivnímu řádku na displeji (musí v něm být kurzor).

Aby byl pokyn pro vykonání povelu jednoznačný, zadáváme text pro zapnutí povelu do jednoho řádku a text pro vypnutí povelu do druhého řádku.

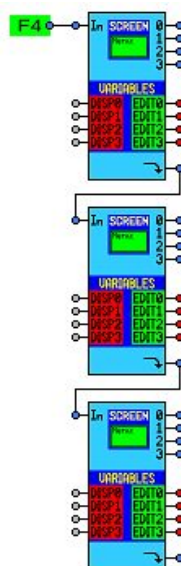
Proto jsou v případě povelů k dispozici další 4 parametry **Text\_line\_4..7**, které jsou vázány na původní parametry **Text\_line\_0 ... 3**.

Pokud chceme nadefinovat pomocí **SCREEN** modulu 4 povely, pak

- první povel definujeme parametrem **Text\_line\_0** pro zapnutí a parametrem **Text\_line\_4** pro vypnutí (a odpovídá mu výstup **Edit0**)
- druhý povel definujeme parametrem **Text\_line\_1** pro zapnutí a parametrem **Text\_line\_5** pro vypnutí (a odpovídá mu výstup **Edit1**)
- třetí povel definujeme parametrem **Text\_line\_2** pro zapnutí



Obr. 159 Příklad klasického zapojení modulů **SCREEN**



Obr. 160 **SCREENy** v kaskádě **Příklad definice povelu s výstupem Edit3:**

**Text\_line\_3: \$+START cernadla**

**Text\_line\_7: -STOP cernadla**

**POZOR!** Výstupy **edit0..3** jsou typu **float**.

Pokud chcete příslušný výstup **edit0..3 = 1,0** (float) použít do logiky hradel, musíte provést konverzi výstupní hodnoty float na integer, a to pomocí hradla **float2int**.

Stav těchto povelů je po překladač dán hodnotou parametrů **edit0init...edit3init**.

Stav povelů po výpadku napájení regulátoru odpovídá jejich stavu před výpadkem.

## Zobrazení hodnot v řádku s povelu

V textu k povelu může být použit formátovací příkaz typu **#####4 až 7** pro zobrazení hodnot ze vstupů **disp0..3**.

a parametrem **Text\_line\_6** pro vypnutí (a odpovídá mu výstup **Edit2**)

- čtvrtý povel definujeme parametrem **Text\_line\_3** pro zapnutí a parametrem **Text\_line\_7** pro vypnutí (a odpovídá mu výstup **Edit3**).

Zobrazuje se text povelu, který chceme vykonat.

Je-li výstup **Edit0..3 = 0**, zobrazuje se text z parametru **Text\_line\_0 ... 3**.

Je-li výstup **Edit0..3 = 1**, zobrazuje se text z parametru **Text\_line\_4 ... 7**.

## Změna textu v řádku v závislosti na stavu vstupu disp0..3

Přepínání textu v řádku využívá principu výše uvedených povelů a znaku @.

Je-li vstup **disp0=0.0** je na prvním řádku displeje zobrazován text dle parametru **Text\_line\_0**, který musí začínat znakem @. Jinak je zobrazován text dle parametru **Text\_line\_4**.

Je-li vstup **disp1=0.0** je na druhém řádku displeje zobrazován text dle parametru **Text\_line\_1**, který musí začínat znakem @. Jinak je zobrazován text dle parametru **Text\_line\_5**.

Je-li vstup **disp2=0.0** je na třetím řádku displeje zobrazován text dle parametru **Text\_line\_2**, který musí začínat znakem @. Jinak je zobrazován text dle parametru **Text\_line\_6**.

Je-li vstup **disp3=0.0** je na čtvrtém řádku displeje zobrazován text dle parametru **Text\_line\_3**, který musí začínat znakem @. Jinak je zobrazován text dle parametru **Text\_line\_7**.

## Parametry

Standardní parametry modulu: *rychlost a priorita*.

Doporučujeme používat parametr **rychlost=3** (defaultní hodnota) pro rychlou odezvu na stisk klávesy.

Parametry **edit0init** až **edit3init** určují počáteční hodnoty výstupů **Edit0..3**. Editováním výstupu **Editn** se také mění hodnota příslušného parametru v paměti RAM regulátoru, aby se uschovala při následujícím spuštění regulátoru poslední editovaná hodnota.

Parametr **dispedit** je určen pro povolení editace hodnot **disp0..3** i z dispečinku, a to pomocí modulu **comport**.

Je-li **dispedit=0** nelze editovat ni jeden z **disp0..3**.

Je-li **dispedit=764** lze i z dispečinku editovat **disp0** (identifikátor 4), **disp2** (identifikátor 6) a **disp3** (identifikátor 7).

Cifra v parametru odpovídá identifikátoru, na pořadí nezáleží.

Povolujte editaci pouze u zapojeného pinu **disp0..3** u modulu **comport**.

Parametr **access** určuje úroveň přístupu do menu modulu **SCREEN** (obdobně jako u modulu **CANTERM / CKDM11 / CKDM12** přístup do módu *system* regulátoru).

- **access=0**  
umožňuje přechod do tohoto kdykoli
- **access=1 až 4**  
určuje, že pro přechod do menu musí být přihlášen uživatel alespoň úrovně rovné hodnotě parametru **access**.

Parametr **prehlidka** zapíná (pokud je roven 1) funkci *slideshow*. Funkce *slideshow*, pokud je aktivovaná, periodicky na několik sekund zobrazuje informace (menu) z modulů **SCREEN**, které mají nastaven parametr **prehlidka=1**. Terminál v režimu *slideshow* nezobrazuje kurzor. Po stisku libovolné klávesy se funkce *slideshow* vypne a kurzor se opět objeví.

Funkce *slideshow* se aktivuje pouze z hlavního menu, a to automaticky pokud nebyla po dobu 30 minut stisknuta žádná klávesa nebo z hlavního menu po krátkém stisku klávesy ESC. Funkce *slideshow* se ruší stiskem libovolné klávesy,

doporučuje se klávesa ESC z toho důvodu, že na jinou klávesu (pokud je to dáno vašim projektem) může SW reagovat.

Do přehlídky jsou moduly **SCREEN** zařazovány podle své polohy na schématu, tj. shora dolů a zleva doprava, tak jako vykonávání ostatních modulů v programu.

V módu přehlídka není zobrazován na displeji kurzor (->).

*Poznámka:*

*Pokud nechcete využít módu přehlídka (funkce slideshow), doporučujeme u modulu SCREEN, který je určen pro hlavní menu (první, který je napojen na výstup Sc modulu CANTERM/CKDM11/CKDM12), nadefinovat parametr prehlidka = 1, a to z toho důvodu, aby po uplynutí doby 30 minut po nestlačení klávesy pro přechod do módu přehlídka, bylo hlavní menu občerstvováno. To je nutné, zejména zobrazujete-li v hlavním menu údaj o datu a čase.*

Parametr **enb\_kask** je běžně = 0.

Pokud však zařadíte modul **SCREEN** do kaskády (pod jiný **SCREEN**, na který se napojuje vstupem **In** na jeho dolní výstup "šipka"), musíte nadefinovat parametr **enb\_kask=1**, a to v modulu, který je zařazen v kaskádě jako následný.

Pokud tak neučiníte, nelze bezchybně rolovat více než 4řádkové menu, v některých případech není funkční klávesa ESC.

### Tip na přehlídku:

Do přehlídky lze zařadit i moduly **SCREEN**, které nenapojíte ani na modul **CANTERM/CKDM11/CKDM12**, ani do stromové struktury modulů **SCREEN**.

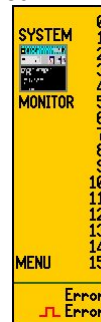
Docílíte tak zobrazování pouze opravdu důležitých informací.

Pokud nejsou všechny moduly **SCREEN** zapojeny na modul **CANTERM/CKDM11/CKDM12**, přechází se do módu přehlídka po 30 minutách (krátký stisk klávesy ESC v tomto případě není funkční).

## 7.2 sysmon



Modul **sysmon** (SystemMonitor) je určen pro zachycování a zobrazení chybových hlášení z jednotlivých modulů, zejména modulů **havbin**, **havan**, **vzorky** a **cdkm**.



Obr. 161 Značka **sysmon** na ploše

Logický výstup **Error** je nastaven do 1 pokud modul zaregistruje alespoň jedno chybové hlášení. Pokud není registrováno žádné chybové hlášení je výstup **Error** = 0.

Od verze FW 3.003 přibyl pulzní výstup **ErrorPulse**, který je aktivní (=1) vždy příchodem nového alarmu, a to po dobu jedné periody programu.



Výstupy **0..15** jsou určeny pro zobrazení seznamu chybových hlášení v režimu RUN ve FREDovi. Chybová hlášení jsou zobrazována labels, připojenými na tyto výstupy.

Vstup **Menu** je určen pro připojení na modul **SCREEN** a tak umožňuje vstup do speciálního menu zobrazení chybových hlášení. V tomto menu klávesami **1..4** volíme zobrazení 1. až 4. čtveřice chybových hlášení v seznamu.

Přepínání stránek s chybami lze provádět **i pomocí šipek** (nahoru a dolů) na CKDM-11.

Při prvním přístupu do výpisu chyb se nejprve zobrazí první stránka.

*Číslo právě zobrazované čtveřice je zobrazováno v levém horním rohu displeje.*

*Řádek, neobsahující chybové hlášení obsahuje text **Ok**.*

Chybové hlášení je možné kvitovat najetím kurzoru na hlášení, které se pokusíme kvitovat (vymazat) a stisknout klávesu DEL.

Pokud je toto chybové hlášení stále aktuální, po stisku DEL se znovu objeví.

Pokud již chybové hlášení není aktuální a bylo v seznamu pouze proto, že bylo nastaveno jako kvitovací, stiskem klávesy DEL se vymaže.

**Výmaz všech chybových hlášení najednou** lze provést v menu prohlížení chyb stiskem klávesy **F2**.

**Obecné chybové hlášení** od modulu, který nemá zadávání textu pro chybové hlášení je ve formátu:

**Er: chybovy\_kod\_MSB, chybovy\_kod\_LSB@obj, inst**  
a znamená, že objekt číslo **obj**, s instancí **inst** způsobil chybu **chybovy\_kod**.

Typickým příkladem jsou chyby od modulu č. 164, instance 1 - **sysmon**, kde chybový kód

1 je určen pro hlášení "nestíhá smyčka rychlosti 1"

2 je určen pro hlášení "nestíhá smyčka rychlosti 2"

4 je určen pro hlášení "nestíhá smyčka rychlosti 3"

Chyby od jednotlivých smyček nastavují příslušné bity chybového kódu, takže např. pokud nebude stíhat smyčka s rychlostí 3 a zároveň nestíhá i smyčka s rychlostí 2 bude chybový kód 4+2=6.

*Příklad chybového hlášení při umístění 4. modulu vzorky do projektu:*

**Er: 000, 001@155, 004**

*znamená chybový kód 1, objekt 155 (vzorky), instance 4.*

Seznam čísel objektů v knihovně pro snadnou orientaci směrem k modulu, který chybu hlásí, naleznete v tabulce obj. č. v **Příloze č. 3** tohoto manuálu.

Standardní parametry modulu: *rychlost a priorita*.

Parametr **Version** zobrazuje v režimu RUN verzi software systému ve formátu **ddmmyyvv**, kde

- **dd** je den v měsíci data verze
- **mm** je měsíc data verze
- **yy** je poslední dvojčíslí roku verze
- **vv** je verze software.

*Příklad:*

*26060221 je verze software ze dne 26.6. 2002, verze 2.1.*

Parametr **CRC** zobrazuje v režimu RUN kontrolní součet (CRC kód) konfiguračního programu z FREDa, uložený v systému.

Parametr **pwout** svou hodnotou určuje časovou prodlevu v ms při komunikaci regulátoru s FREDem (případně Control Webem) na komunikační lince (defaultně COM0), a

to mezi přijetím posledního znaku požadavku a začátkem vysílání odpovědi (timeout přijímače).

Parametr **address** je funkční pro HW jednotku CCPU, pokud je na jejím přepínači nastavena pro komunikaci s FREDem/ControlWebem adresa = 0. Pak hodnota parametru **address** určuje softwarově adresu pro komunikaci.

Parametr **memory** zobrazuje (v ladící režimu FREDa) procento využití paměti, vyhrazené pro aplikační program.

Parametry **loop1..3** zobrazují (v ladícím režimu) procento využití času programových smyček s rychlostmi 1..3.

Parametr **outlock** sloužil pro "zamčení" výstupů při překladu. Od verzí FW 2.0 periférií nefunkční. Nyní řešeno pomocí komunikačního ovladače mezi vývojovým prostředím a centrálou a jeho funkcí CAN\_STOP. Užitím této funkce (a novějšího FW periférií) v ladícím režimu FREDa zůstanou výstupy "zamčené" v aktuálním stavu, v editačním režimu lze projekt upravit, provést překlad, přejít do ladícího režimu a funkci CAN\_STOP zrušit. Tím se aplikace rozběhne ze "zamčeného" stavu a pokračuje v provádění podle aktuálních vstupů...

*Dřívější funkce parametru outlock:*

*Pro správnou funkci (aby relátka neklapla) musí být inicializace modulu SYSMON provedena dříve než modulů s relé. Projekt poprvé nahrajte s outlock = 0.*

*Ladíte běžící aplikaci a v momentě, kdy budete chtít dělat nový překlad, změňte nejprve v RUN režimu parametr outlock = 1.*

*Teprve pak (za předpokladu, že v opraveném projektu outlock = 1) proveďte nový překlad. Po nahrání opraveného projektu (a případně po dosažení některých požadovaných stavů v technologii) přepínejte do RUN režimu a změnou outlock = 0 funkci relé odblokujete.*

Parametr **writecheck** zakazuje/povoluje zápis z dispečinku/FREDa mimo zónu výstupních pinů modulu **comport**. Samozřejmě, že lze však stále měnit parametr **writecheck** tohoto modulu **sysmon**.

Nastavením **writecheck=1** lze zapisovat z FREDa či z ControlWebu pouze do výstupů modulů **comport** v projektu a do parametru **writecheck** modulu **sysmon**. V ostatních případech sice komunikace proběhne, ale zápis se neprovede.

Čtení je možné vždy bez omezení, tj. i když je **writecheck=1**.

Pokud je čteno komunikační relací mimo existující objekt či instanci (v projektu), tak se v komunikačních datech objeví hodnota 0xBEBEBEBE.

Toto slouží pro ochranu proti chybným a pirátským zápisům, které by zbořily aplikační SW v centrále.

Parametr **erase\_error**

- nastavením do 1 v ladícím režimu (nebo z dispečinku) smaže (kvituje) všechna chybová hlášení
- do 0 se vrací "sám"

Parametr **overdrive** zrychlí 3. smyčku o zadaný počet milisekund, záporné číslo smyčku zpomalí.

Platné hodnoty od 90 do -100.

Pro časově kritické aplikace lze jít u smyčky 3 až na vytížení 150% . Hodnota vytížení je uvedena v parametru **loop3**. 50% navíc se bere z času, původně vyhrazeného pro smyčky 2 a 1.



## 7.3 timdat



Modul **timdat** je určen pro nastavení systémového data a času v regulátoru.

Vstupem **Menu** se modul připojuje na výstup modulu **SCREEN** stejně jako jiný modul **SCREEN**.



Obr. 163 Značka **timdat** na ploše

Po vstoupení do menu modulu **timdat** je možné nastavit systémový čas a datum.

Nastavené hodnoty se uloží stiskem ENTER.

Stiskem ESC se menu opustí bez uložení změněných hodnot.

Nastavování systémového času:

- klávesy <- -> pohybují kurzorem doleva/doprava
  - klávesy + a - mění nastavení hodnoty příslušného údaje
- Standardní parametry modulu: *rychlost a priorit*a.

Modul **timdat** musí mít parametr **rychlost = 3**.

Výstupy modulu **timdat** (jako hodnota float):

**Tim** - aktuální čas ve tvaru HH.MM

**Dat** - aktuální datum ve tvaru DD.MM

**Sec** - aktuální vteřiny ve tvaru SS.00

**Day** - aktuální den v týdnu ve tvaru d.00 (0.00 pro neděli až 6.00 pro sobotu)

**Year** - aktuální rok ve tvaru RRRR.00

Parametry:

- **settime**  
pro nastavení systémového času regulátoru po překladu nebo z dispečinku  
hhmmssDD  
(dekadicky hodiny, minuty, sec a den v týdnu)
- **setdate**  
dtto pro nastavení data  
yyyymmdd  
(dekadicky rok, měsíc, den)
- **w2ldate**  
datum přechodu na letní čas
- **l2wdate**  
datum přechodu na zimní čas

*Poznámka:*

*Pokud chcete původní systémový čas zachovat, zadejte parametry pro nastavení data a času nulové.*

*Přechod na letní čas ovlivní čas zobrazovaný direktivou #T modulem **SCREEN** a čas pro moduly **timer**.*

*Čas pro moduly **vzorky** zůstává vždy zimní.*

*Datum přechodu na čas letní/zimní neodpovídá přesně zadaným datům (dáno předpisem pro celou Evropu od r. 1996). Defaultní hodnoty 2503/2510 jsou pak vhodné pro logiku skriptu.*

*K přepnutí na jiný čas dochází nejbližší neděli po zadaném datu v parametru, a to vždy v neděli ráno z 02:00 na 03:00 nebo z 03:00 na 02:00. Hodnoty parametrů w2ldate a l2wdate neměňte - využijí se pouze při změně předpisu.*

## 7.4 hesla



Modul **hesla** je určen pro přihlášení/odhlášení uživatele a pro editaci hesel první skupiny uživatelů.

Standardní parametry modulu: *rychlost a priorit*a.



Obr. 162 Značka **hesla** na ploše

Modul **hesla** musí mít parametr **rychlost = 3**.

Modul **hesla** umožňuje přihlášení až 4 typů uživatelů s různými přístupovými právy.

Parametry **user1 / user1a / user1b** určují počáteční hesla pro 3 uživatele úrovně 1.

Parametry **user2 / user2a / user2b** určují počáteční hesla pro 3 uživatele úrovně 2.

Parametry **user3 / user3a / user3b** určují počáteční hesla pro 3 uživatele úrovně 3.

Parametry **user4 / user4a / user4b** určují počáteční hesla pro 3 uživatele úrovně 4.

Heslo je vždy čtyřmístné celé kladné číslo.

První skupinu hesel (parametry **user1, user2, user3 a user4**) lze editovat přímo z ovládacího panelu CKDM-11/12 dle dalšího popisu níže.

Další skupiny hesel (parametrů s indexy **a a b**) lze editovat pouze z FREDa nebo z dispečerského prostředí.

Vstup **Menu login** je určen pro připojení na modul **SCREEN** a umožňuje vstup do speciálního *menu login*. V *menu login* se může uživatel přihlásit vložím hesla klávesou INS.

Stav *login* je indikován za slovem **Login**:

Pokud se uživatel úspěšně přihlásí, na displeji se zobrazí stav **Login:1**. Pokud uživatel zadá heslo znovu, provede se odhlášení a stav je opět **Login:0**, indikující úspěšné odhlášení uživatele.

Rychlé odhlášení lze provést stiskem klávesy F1 (bez zadání hesla je všem uživatelům přiřazen *Login=0 a userlevel=0*).

Přihlášením se změní stav *userlevel* v systému.

*Příklad:*

*Přihlášení User3 nastaví userlevel=3.*

*Pokud není nikdo přihlášen, userlevel=0.*

Moduly **SCREEN** mají nastaven parametrem **access** minimální *userlevel*, pro který je možné vstoupit do menu daného modulu **SCREEN**.

*Např. uživatel User3 pokud se nepřihlásí, má přístup pouze do menu modulů **SCREEN**, které mají parametr **access=0**. Pokud se přihlásí jako User3, uvolní se mu přístup do menu modulů **SCREEN**, které mají parametr **access<=3** (3 a menší). User3 však nikdy nemá přístup do menu modulů **SCREEN**, které mají parametr **access=4**.*

Vstup **Menu edit** je určen pro připojení na modul **SCREEN** a umožňuje vstup do speciálního menu *edit password*.

Klávesou INS může uživatel vložit nové heslo pro příslušnou skupinu uživatelů **user1..4**. Změna bude však provedena pouze pokud je už přihlášen uživatel té skupiny, pro kterou má být změna provedena.

*Příklad:*

*uživatel User2 může změnit pouze heslo User2, pokud se před tím přihlásí jako User2. Pokud změna proběhla úspěšně, musí se odhlásit už pod novým heslem. Pokud se uživatel pokusil změnit nepatřičné heslo, musí se odhlásit pod původním heslem. Nepatřičná změna hesla pak nebude provedena (např. pokud se User2 pokusí změnit heslo pro User3 nebo třeba pro User1).*

**Výstup modulu hesla** nabízí hodnotu hesla právě přihlášeného uživatele.

Pokud není nikdo přihlášen, je na výstupu 0.

**Výstup Level** indikuje hodnotu úrovně, právě přihlášeného uživatele.

Pokud není nikdo přihlášen, je na výstupu 0.

Jinak nabývá hodnot 1..4.

Po načtení aplikačního programu z paměti Flash je automaticky provedeno odhlášení všech dříve přihlášených uživatelů (Level=0).

*Poznámka:*

*Pokud je kterýkoliv uživatel přihlášen správným heslem, neprovádí se kontrola na správnost kontrolního součtu aplikace. Po odhlášení uživatele se provede přepočítání kontrolního součtu aplikace a začne se provádět jeho kontrola každé 3,5 sec.*

*Pokud je uživatel přihlášen např. jako user4, musí se po provedení změn v systému odhlásit, jinak se pro provedené změny nespočítá kontrolní součet a po restartu se provede obnova z flash. Po provedené změně je vždy potřeba vyčkat, dokud je zobrazen v levém horním rohu na displeji terminálu vykřičník. Teprve pak lze systém restartovat.*

## 7.5 vzorky



Modul **vzorky** realizuje funkci historické databanky hodnot regulátoru.



Obr. 164 Značka **vzorky** na ploše

**Pozor, v jednom projektu je možné použít maximálně 4 moduly vzorky.** Pokud je počet modulů **vzorky** v projektu překročen (větší než 4), je generováno chybové hlášení pro modul **sysmon**.

Modul **vzorky** s instancí 1 zapisuje data do BANKY1, s instancí 2 do BANKY 2, ... s instancí 4 do BANKY 4.

Vstupy **Vz0..7** jsou vstupy monitorovaných hodnot.

Vstup **Record** je logický vstup pro povel k zápisu do banky (vzestupnou hranou dochází k zápisu).

Vstup **Erase** je logický vstup pro povel ke smazání obsahu banky (maže po celou dobu, kdy je log. 1 na tomto vstupu).

Standardní parametry modulu: *rychlost a prioritá*.

Ovládání zápisu do banky:

– Vzestupnou hranou na vstupu **Record** se provede zápis aktuálního stavu monitorovaných vstupů **Vz0..7**.

– Zápis lze také provést, pokud od předešlého zápisu uběhlo více času (v sekundách), než je určeno parametrem **delta\_t**.

Pokud **delta\_t=0** je tento způsob zápisu do banky zablokovan. Vzorky se pak nebudou ukládat periodicky, ale jiným z dalších možných způsobů zápisu.

– Zápis všech vstupů lze provést i při změně některého z monitorovacích vstupů.

Způsob monitorování pro každý vstup je dán parametry **bf30..bf37**.

Pokud **bf3n=0**, uložení se při změně vstupní hodnoty **Vzn** neprovede. Hodnota je brána jako celočíselná a je ukládáno dolních 24bitů vstupní hodnoty.

Pokud **bf3n=1**, uložení se provede, pokud se vstupní hodnota **Vzn**, považovaná za reálné číslo, změní o více než o hodnotu **deltin**. Pak je hodnota vstupu uložena jako reálné číslo konvertované do formátu 3bytefloat.

Pokud **bf3n=2**, uložení se provede, pokud se hodnota vstupu **Vzn**, považovaná za celočíselné číslo, jakkoli změní. Uloží se pak dolních 24 bitů vstupní hodnoty. Ve spojení s modulem **bin\_to\_int** je tato volba vhodná pro monitorování binárních výstupů.

Parametry **redir0..7** souží pro nasměrování vstupů modulu **vzorky** na vstupy modulu **comport**.

Do historie není pak ukládána hodnota příslušného vstupu modulu **vzorky**, ale hodnota vstupu modulu **comport**.

Změnou parametru **redir0..7** pak docílíte překonfiguraci ukládané historie z jiných vstupů/modulů **comport**, např. dálkově z dispečinku.

Samozřejmě musíte zachovat typy vstupů s ohledem na definici pomocí parametru **bf30..7**.

Hodnoty parametru **redir0..7**:

- 0 nula vypíná (čili na vstup 0 modulu **comport** s instancí 1 se nedostanete)
- 1..15 odpovídají vstupům 1..15 modulu **comport** s instancí 1
- 16..31 odpovídají vstupům 0..15 modulu **comport** s instancí 2
- atd.

Parametr **pbankcom** - indikuje index, kde se bude zapisovat další vzorek (vrchol banky).

Když je menší než 0, banka ještě není plná = platné vzorky jsou od indexu 0 do -pbankcom.

Když je kladný, je celá banka plná - nejstarší vzorek je na indexu pbankcom, nejnovější je o jednu před ním.

Slouží pro zdokonalení vyčítání vzorků - možno vyčíst jen nově přibývší vzorky.

*Podrobněji v poznámce:*

Je to ukazatel indexu. Na začátku je -1 -> první vzorek se zapíše na index jedna, pak je -2, -3, -4 atd.

Pouze kvůli indikaci, že banka ještě není plná, se zobrazuje záporně, tedy s mínusem. Čili např. pbankcom = -133 indikuje, že v bance je 132 vzorků, a to od indexu 1 do indexu 132.

Dále např. pro malou centrálu až vystoupá na hodnotu -527, pak spadne do 1 a znovu roste 2, 3, 4 ...

-527 znamená, že je v bance právě 526 vzorků od 1 do 526.

Pak je 1 -> banka je právě plná (na rozdíl od -1, což by znamenalo, že je úplně prázdná).

Takže potom až nabude třeba hodnoty 63, znamená to, že v bance je 527 vzorků, nejstarší na indexu 63, a nejmladší na indexu 62.

Index 0 se používá na identifikaci typu vzorku.

Vždy je to index pro další zápis. V absolutní hodnotě vždy roste 1, 2, 3, .. 527, 1, 2, 3 ...

Pouze znaménkem se rozlišuje, jestli banka ještě není plná, a vzorky jsou tudíž jen od 1 do hodnoty |pbankcom| (absolutní hodnota), nebo je plná, vzorky jsou všude a podle pbankcom se dá zjistit, kolik jich přibylo od posledního čtení.

Kapacita jednoho modulu **vzorky** je 528 záznamů pro "malou" centrálu nebo 2112 záznamů pro "velkou" centrálu. Jedním záznamem se vždy uloží všech 8 monitorovaných hodnot s datem a časem záznamu.

Historickou databanku (BANKY1-4) lze z regulátoru PL2 (z centrály CCPU-02) vyčíst po sériové lince do nadřazeného PC pomocí programu WZORKY, který slouží pouze pro práci s těmito historickými daty nebo pomocí komunikačního ovladače do dispečerského prostředí (ControlWeb).

*Poznámka:*

Banky jsou po překladu vymazány. Důvodem je možná změna definice pro ukládání dat do bank.

Proto pokud provádíte překlad opakovaně a požadujete původní data zachovat, musíte je nejprve vyčíst programem WZORKY a uložit pro jejich archivaci. V regulátoru nebudou zachována!

## 7.6 flash



Modul **flash** je v projektu v podstatě povinný (viz dále) a umožňuje práci se záložní pamětí FLASH:

- uložení konfiguračního programu a konfiguračních dat do paměti FLASH
- načtení konfigurace z paměti FLASH do pracovní paměti RAM



Obr. 165 Značka **flash** na ploše

Po nahrání konfigurace (projektu) z FREDa do regulátoru se tato uloží do paměti pracovní - RAM. Také veškeré další zásahy - např. změna editovatelných parametrů, se provádí v paměti RAM.

Pomocí modulu **flash** musíme aktuální stav systému (konfigurační program, konfigurační data, procesní data) trvale uložit do paměti FLASH.

Toto uložené nastavení regulátoru je možné později obnovit načtením konfigurace z paměti FLASH.

Konfigurace se také čte z paměti FLASH pokud je chybný kontrolní součet konfiguračních dat v paměti RAM.

Kontrolní součet se kontroluje automaticky každé 3,5 sec, proto musí být uložen, aby při chybném kontrolním součtu bylo co z FLASH načíst.

Při změně některého z parametrů z klávesnice nebo z dispečinku či z FREDa dojde automaticky k přepočítání kontrolního součtu aplikace. Od změny parametru do ukončení přepočtu se na displeji v prvním řádku na první pozici zobrazí vykřičník.

Vstupem **Menu** se modul připojuje na výstup modulu **SCREEN**.

V menu je pak nabídka:

- **F1 - Save**  
pro uložení aktuálního stavu konfiguračního programu a dat do paměti FLASH
- **F2 - Load**  
pro vyčtení konfigurace z FLASH do RAM

Po volbě **F1** na terminálu regulátoru dojde během 15 sec (na displeji nápis **WAIT**) k uložení konfigurace do paměti FLASH. Po dobu ukládání pracuje systém regulátoru pomaleji (např. sběrnice CAN asi 4x) a není funkční komunikace po sériových linkách.

Po volbě **F2** na terminálu regulátoru je na displeji zobrazen nápis **WAIT**, načtena paměť FLASH do paměti RAM a proveden RESET centrály (kvůli obnovení mezí z CAIO-12).

Výstup **Saving** je po dobu ukládání (F1 - Save) v 1, jinak je v 0.

Toho lze využít např. pro blokování zvolených činností regulátoru po dobu SAVE, k např. akustické signalizaci beepem ovládacího panelu CKDM pro obsluhu (že má čekat a raději nic neovládat ...).

Parametr **flash** se využívá k aktivaci funkce SAVE (ukládání do FLASH) v případě, že v projektu (a v praxi) není osazen ovládací panel CKDM.

Nastavením do 1 dojde ke zmíněné aktivaci funkce SAVE, téměř ihned se parametr automaticky snuluje.

Na výstupu **Saving** lze sledovat, zda ještě probíhá ukládání dat do paměti FLASH.

Parametry **userload** (pro LOAD z paměti FLASH) a **usersave** (pro SAVE do paměti FLASH) nastavují potřebnou úroveň přihlášení pro práci z FLASH. (Obdobné jako u modulu **SCREEN** parametr **access**.)

0 = operaci může provádět každý user

1-4 = určuje, že pro přechod do menu pro LOAD / SAVE musí být přihlášen uživatel alespoň úrovně rovné hodnotě parametru **userload** / **usersave**.

Standardní parametry modulu: *rychlost* a *priorita*.

Modul **flash** musí mít parametr **rychlost = 3**.

*Poznámka:*

Po upgrade firmware v CCPU je smazán původní projekt (i v RAM i ve Flash). Zároveň je aktivována funkce, která způsobí smazání projektu v RAM vždy po restartu CCPU. Tato funkce se zruší prvním uložením projektu do paměti Flash. Tímto je zabráněno používání (resp. spuštění) PL2 bez uložení projektu do Flash).

Od verze firmware 2.5 je při inicializaci systému prováděna kontrola dat v paměti Flash (včetně BIOSu).



Pokud dojde k chybě dat (nesouhlasný kontrolní součet), je činnost centrály zablokována, na CCPU-21 svítí všechny LED, na CCPU-02/03 všechny LED blikají. Po resetu (nebo po vyp/zap) bliká LED RUN zeleně a současně LED ERR červeně, na COM0 je současně vyslán skutečný kontrolní součet (4 bajty). **V praxi to znamená provést UpG firmware centrály (s ručním resetem) a aplikaci přeložit a nahrát znovu.**

## 7.7 canopen



Modul **canopen** je vhodné zařadit do projektu pokud chceme, aby byl reportován v chybových hlášeních stav sběrnici CAN.



Obr. 168 Značka **canopen** na ploše

Logický vstup **Off** umožňuje (pokud **Off=1**) vypnout komunikaci na sběrnici CAN.

Opětovné zapnutí komunikace na sběrnici CAN je možné až po nastavení **Off=0** a restartování systému.

Standardní parametry modulu: **rychlost a prioritá**.

Od firmware verze 2 (pro periferní jednotky na sběrnici CAN) je hlášení upraveno tak, že cifra v hlášení určuje adresu jednotky, která je sice v projektu, ale nekomunikuje.

Např.: **CANopen 03** znamená, že nekomunikuje periferní jednotka s adresou 3.

Logický výstup **E** (error) svým stavem hlásí, že na CAN sběrnici chybí jednotka/y.

Např.:

$E=1$  při odpojení všech CAN jednotek v sestavě

$E=1$  při zapojené nekomunikující jednotce (např. má jinou adresu než v projektu)

## 7.8 flag



Modul **flag** indukuje počet daných systémových událostí od aktivace vstupem **Del**.

Výstup **Reset** (bit 0, váha 1) počítá u centrály průchody resetem.



Obr. 166 Značka **flag** na ploše

Výstup **Fsave** (bit 1, váha 2) počítá záznamy aplikačního SW do paměti FLASH.

Výstup **Fload** (bit 2, váha 4) počítá načtení aplikačního SW z paměti FLASH.

Výstup **COM** (bit 3, váha 8) počítá komunikační relace na sériové lince, po které probíhají tzv. QQ relace (běžně COM0).

Výstup **SysMenu** (bit 4, váha 16) počítá návraty ze systémového menu do aplikace.

Výstup **Download** (bit 5, váha 32) počítá překlady z FREDa a jejich přenos po sériovém kanálu do paměti CCPU.

Výstup **Rx0** (bit 6, váha 64) počítá přijaté zprávy na kanále COM0.

Výstup **Rx1** (bit 7, váha 128) počítá přijaté zprávy na kanále COM1.

Výstup **Rx2** (bit 8, váha 256) počítá přijaté zprávy na kanále COM2.

Výstup **Bat** (bit 9, váha 512) signalizuje stav zálohovací baterie (1=dobrá).

*Pozor! Při špatném kontaktu nebo uplně chybějící baterii ukazuje "1=dobrá" (HW omezení).*

*Poznámka:*

*Výstupy Rx0..2 indikují příjem zprávy pro libovolný typ komunikace (QQ, MT, Sercocom, GSM, MPC), a to bez ohledu na to, zda je zpráva dále SW zpracována (je-li správný checksum, adresa atd.). Vhodné pro ověření HW funkce COM kanálu.*

Výstup **BitFlags** potom aktivuje příslušný bit (bity) při změně (inkrementaci) daného příznaku (flagu). V klidu sledovaných systémových událostí je = 0. Pak např. při zakomunikování se na výstupu **BitFlags** objeví hodnota 8, apod. To je vhodné pro účely archivace - připojit **BitFlags** k modulu **vzorky** a při každé systémové události je generován bitový (binární) impulz registrující danou událost.

Standardní parametry modulu: **rychlost a prioritá**.

## 7.9 journal



Modul **journal** je jakýsi přímý prohlížeč historické databanky (na displeji terminálu), do které se ukládají informace, definované pomocí modulu **vzorky**.



Obr. 167 Značka **journal** na ploše

Definice zobrazování průvodních textů k historickým datům v příslušné bance se provádí pomocí modulu **journal\_txt**.

Vstup **Menu** modulu **journal** se připojuje na modul **SCREEN** (zabezpečení vazby na displej terminálu v sestavě PL2) do příslušného menu.

Parametrem **bankno** nastavíme číslo banky 0..3, kterou chceme prohlížet.

Banka 0 odpovídá modulu **vzorky** s instancí 1, ...

Lze prohlížet pouze jednu banku, ale libovolnou banku 0..3.

Od FW 3.006 lze tento parametr měnit (v ladicím režimu FREDa nebo v projektu modulem **setpar**) a prohlížet tak i další banky (definované modulem **vzorky**), a to pokud jsou v projektu zapojeny obdobně jejich vstupy **Vz0..7**. Platí pro ně totiž stejné názvy hodnot jako pro původní vstupy **Vz0..7** modulu **vzorky** (definované modulem **journal\_txt**).

V projektu může být pouze jeden modul **journal** !

Standardní parametry modulu: **rychlost a prioritá**.

Na displeji terminálu jsou pak vidět ty položky banky, ke kterým jsou přiřazeny texty pomocí modulu **journal\_txt**.

Pokud je v bance uložena celočíselná hodnota (u modulu **vzorky** příslušný parametr **bf3x=0/2**), bere se tato tříbajtová hodnota jako 24 binárních položek.

Pokud je v bance uloženo reálné číslo, bere se jako analogová položka, jejíž hodnota se pak zobrazuje v pevném



formátu 6.1 (6 cifer, des. tečka, jedno desetinné místo). Zpracovávají jsou hodnoty +/- 8 000 000.01.

Pro kolik položek je zadán modulem **journal\_txt** text (popis/název položky), tolik se jich popořadě zobrazuje v prohlížení.

Zobrazovány jsou pouze změny monitorovaných položek, u binárních každá změna, u analogových pouze skoková změna větší než parametr **deltx** daného vstupu modulu **vzorky**.

První záznam v bance je brán vždy jako změna.

Položky jsou ve výpisu uspořádány (v případě, že je více změn současně v jednom vzorku) podle pořadí textu/hodnot v modulech **journal\_txt**/modulu **vzorky**.

V položkách výpisu na displeji terminálu sestavy PL2 je možné listovat klávesami šipka nahoru/dolů ob dvě položky (= jednu obrazovku), klávesami shift+šipka nahoru / shift+šipka dolů o 20 položek, a klávesami šipka doleva / šipka doprava na začátek / konec seznamu.

Pokud je banka už plná (dochází k přepisování nejstarších hodnot hodnotami novými) a ukládání událostí je časté, je možné, že při použití klávesy šipka doleva (skok na začátek seznamu) dojde při prohlížení nejstarší hodnoty k jejímu přepsání hodnotou novou a při dalším rolování se objeví už hodnota nová (protože staré hodnoty byly mezitím přepsány).

Protože plná databanka obsahuje 2112 vzorků (záznam 8mi položek s datem a časem) nebylo by praktické prohlížet na displeji CKDM banku celou. Proto pro zobrazování na displeji jsou vybrány pouze změny, jak bylo popsáno výše. To vyžaduje při prohlížení všech záznamů porovnávání, které zabere čas. Proto ovládání prohlížení banky modulem **journal** šipkami, zejména skoky na začátek či konec banky, zabere od stisku šipky až po zobrazení až 4 sec.

Aby bylo zřejmé, kde se v prohlížení banky pohybujeme, bylo od verze FW 3.017 zavedeno zobrazení pozice v bance, a to na prvním řádku displeje. Místo celého data záznamu/vzorku je zobrazen jen den a měsíc ve tvaru DD.MM, následuje mezera a po ní pořadí záznamu v bance (zhruba odpovídá parametru **pbankcom** modulu **vzorky**).

Aby bylo jasné, že se jedná o první záznam pro zobrazení modulem **journal**, jsou místo pozice v bance zobrazeny znaky [**<**, znázorňující "levý doraz".

Pokud se jedná o poslední záznam pro zobrazení, jsou místo pozice v bance zobrazeny **+2** mezery a znaky **>**], znázorňující "pravý doraz".

Při prvním vstupu do zobrazení ze zobrazuje poslední (tj. nejaktuálnější) položka.

Příklady výpisů do verze FW 3.016 včetně:

```
26.08.02 15:55:28
Zaplaveni - OFF
26.08.02 15:55:28
Prehrati VS - OFF
```

nebo

```
26.08.02 15:27:10
teplota TUV 55.2
26.08.02 15:27:14
teplota UT 64.1
```

a od verze FW 3.017 pro první zobrazený záznam

```
15.08 [< 12:03:46
Ttuv: 40.9
15.08.2006 12:03:46
Stuv: 96.9
```

a další záznam

```
15.08 2047 12:03:47
```

```
Ttuv: 45.3
```

```
15.08.2006 12:03:47
```

```
Stuv: 70.8
```

nebo poslední záznam

```
15.08 >] 12:05:08
```

```
Stuv: 0.0
```

## 7.10 journal\_txt



Module **journal\_txt** umožňují zadat zobrazení textu s hodnotou / stavem v prohlížení historické databanky (pomocí modulu **journal**).



Obr. 169 Značka **journal\_txt** na ploše

Každý modul **journal\_txt** umožňuje zadat texty až pro čtyři po sobě jdoucí položky.

Pro analogové položky definujeme text (popis / název položky) parametry **Text\_line\_0..3**.

Pro binární položky ve stavu H (high=logická 1) pak také v parametrech **Text\_line\_0..3**, ale pro binární hodnoty ve stavu L v parametrech **Text\_line\_4..7**.

Modulů **journal\_txt** může být v projektu několik, a texty se řadí za sebe do seznamu podle instancí modulů.

Standardní parametry modulu: **rychlost** a **priorita**.

*Poznámka:*

*Protože jedné analogové položce odpovídá jeden řádek textu pro její popis / název parametrem **Text\_line\_0..3**, je vhodné řadit tyto položky do historické databanky jako první (první vstupy modulu vzorky).*

*Jeden celočíselný vstup modulu vzorky (třibajtový) je brán jako 24 binárních položek. Každá binární položka se pro text definuje dvěma řádky (pro ON a pro OFF). Pro úplnou definici těchto po sobě jdoucích 24 binárních stavů je tedy potřeba nadefinovat 48 textových řádků - a to pomocí 12-ti modulů **journal\_txt** s instancemi po sobě jdoucími a navázanými případně na předchozí analogové položky.*

## 7.11 printer



Modul **printer** slouží pro ovládání výpisů na tiskárně, připojené na sériový kanál centrály.

Od verze firmware 2.501 lze v projektu použít modul **printer** vícenásobně.



Obr. 170 Značka modulu **printer** na ploše

Vstup **Prn** bity 0..3 zachytí požadavek na tisk 1..4 řádku.

Pokud **Prn**=1, tiskne se pouze 1. řádek; pokud **Prn**=2, tiskne se pouze 2. řádek; pokud **Prn**=3, tiskne se 1. a 2. řádek, a to vždy jeden průchodem programovou smyčkou, atd.

Pokud **Prn**=15, tisknou se postupně každým průchodem programovou smyčkou všechny 4 řádky.

Pokud je současně s požadavkem na tisk zachycen odpovídající bit 0..3 na vstupu **Time**, je před text do řádku vloženo datum a čas, a to ve formátu, v jakém je zobrazován na ovládacím panelu CKDM.

Text jednotlivých řádků je zadán v parametrech **Text\_line\_0..3** a pokud je v něm zadán jeden formátovací příkaz typu **##.##**, je místo znaků **#** vytištěna hodnota ze vstupů **Val0..3**.

Pokud hodnota překročí definovaný formát, je vytištěno EE.EE.

Od verze firmware 2.502 jsou vstupy **Prn** i **Time** "zachycovací", tzn., že si požadavek pamatují až do jeho vykonání.

Na výstup **Error** je posílána hodnota chybového hlášení tiskárny, např. 0x19 = není papír, apod. (viz návod k tiskárně).

Pokud tiskárna pracuje bez problémů, pak je výstup nulový.

Parametrem **comno** definujeme, na který sériový kanál centrály je tiskárna připojena.

Defaultně je nabízen kanál 1 (COM1), pokud je tiskárna připojena na jiný (COM0 / COM2), je potřeba ošetřit signál CTS (musí být pro povolení tisku aktivní).

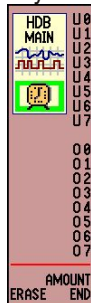
Parametrem **comspeed** definujeme rychlost komunikace na zvoleném sériovém kanále s tiskárnou, defaultně 9600 Bd.

Standardní parametry modulu: **rychlost** a **priorita**.

### 7.11.1 HDBmain

**HDB** Informace o uložených datech v databance typu HDB (popis HDB viz str. 99) poskytuje modul **Hdbmain**.

Jedná se o objekt číslo 186 s instancí 1, který musí být součástí projektu systému PL2.



Obr. 171 Značka modulu **HDBmain** na ploše

Datový objekt **item 13** určuje počet platných vzorků v bance. Tato hodnota se smazáním banky nuluje. Banka se nuluje aktivací 0-tého bitu vstupu **Erase**. Banka se také nuluje automaticky při nahrátí nového projektu.

**Item 14** určuje vrchol banky. Je to index, na který bude zapsán příští vzorek. Jeho hodnota se vymazáním banky nuluje.

Dále **itemy 5..12** (userpos) umožňují uložit pozici (index) až pro 8 uživatelů, určující, kterým vzorkem je potřeba začít příští čtení. Jejich hodnota se vymazáním banky nuluje.

Dále je jejich hodnota posouvána dopředu v případě, že daný uživatel nevyzvedl včas data a tato data jsou přepisována daty novými. Inkrementace probíhá tak, že tyto objekty v případě vymazání nevyzvednutých hodnot ukazují vždy na nejstarší platný vzorek v databázi.

Pro každého uživatele je dále udržováno počítadlo přetečení (**itemy 15..22**). V těchto itemech se zaznamenává po-

čet nevyzvednutých ztracených vzorků daného uživatele. Tato počítadla se dají vynulovat aktivací bitu 1..8 (po řadě každý bit pro každého uživatele) vstupu **Erase**.

Vztah mezi indexem vzorku a fyzickým offsetem paměti databáze je:

$$\text{offset} = 10 * \text{index}$$

Vstup **Erase** umožňuje mazání HDB a mazání počítadel přetečení, viz výše.

Výstupy **U0..7** jsou výstupy uživatelských pozic.

Výstupy **O0..7** indikují stavy počítadel přetečení.

Výstup **Amount** indikuje počet platných vzorků v HDB.

Výstup **End** indikuje pozici indexu pro zápis příštího vzorku (vrcholu HDB).

Standardní parametry modulu: **rychlost** a **priorita**.

### 7.11.2 HDBa

**HDB** Modul **Hdba** zpracovává analogové signály a ukládá je do banky typu HDB.



Obr. 172 Značka modulu **HDBa** na ploše

Vstupy **A 0..7** jsou analogové vstupy typu float, jejich stav je ukládán do HDB.

Výstup **AI** (AlarmOn červený) indikuje příznak vzniku alarmu.

Výstup **AO** (AlarmOff zelený) indikuje příznak zániku alarmu.

Vstup **CI** (ClearOn červený) nuluje příznak vzniku alarmu.

Vstup **CO** (ClearOff zelený) nuluje příznak zániku alarmu.

Parametry **flags0..7** parametrizují svými váhami (bity) způsob ukládání daného analogového vstupu.

Váhy parametru flags0..7 modulu HDBa		
bit	flag	význam
0	wmn	zápis do HDB 1 x denně v 00:00 hod.
1	wtm	zápis do HDB vždy v určený čas (daný parametry <b>fixhour</b> a <b>fixmin</b> ), 1x denně
2	hsn	po zápisu do HDB se suma a počet vzorků nuluje
3	han	po zápisu do HDB se aktuální hodnota nuluje
4	err	vyvolání alarmu při překročení stupně povodňové aktivity a dolní meze
5	hy2	zápis do HDB se provede mimořádně i při překročení <b>_hy2</b>
6	hy1	zápis do HDB se provádí jen při překročení <b>_hy1</b>
7	noz	zápis do HDB se provádí jen pokud je hodnota nenulová

Váhy parametru flags0..7 modulu HDBa		
bit	flag	význam
8	xch	signál se zapisuje do HDB při každé změně aktuální hodnoty
9	avr	analogový signál se při zápisu do HDB průměruje [jinak se bere aktuální hodnota]
10	on	je povolen zápis signálu do HDB
11	ctc	signál se chová čítačově
12	aint	aktuální hodnota se integruje (vyhlazují se prudké změny)
13	isn	při resetu PLC se suma a počet vzorků nuluje
14	ian	při resetu PLC se aktuální hodnota nuluje, jinak beze změny
15	serr	způsobí uložení hodnoty do HDB při vzniku a ukončení alarmu

Např. pro povolení ukládání vstupu **A 0** při každé změně musí být aktivní flagy *xch* a *on*, tj. bity 8 a 10, tj. parametr **flags0** bude mít hodnotu 1280.

Parametry **interval0..7** určují pro každý analogový vstup periodu ukládání v minutách.

Parametry **hyst1 0..7** určují hysterezi 1 pro hlášení alarmu nebo jejíž překročení povolí zápis do HDB.

Parametry **hyst2 0..7** určují hysterezi 2, jejíž překročení způsobí zápis do HDB.

Parametry **lolim 0..7** určují dolní mez analogové veličiny.

Parametry **flood1 1..7** určují meze 1. stupně povodňové aktivity - poruchy.

Parametry **flood2 1..7** určují meze 2. stupně povodňové aktivity - poruchy.

Parametry **flood3 1..7** určují meze 3. stupně povodňové aktivity - poruchy.

Parametry **maxctc 1..7** určují maximální akceptovatelnou diferenci pro jeden krok čítačového vstupu - default 10.

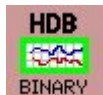
Parametr **index** určuje počátek hodnot identifikátoru vzorku.

Např. *index = 40* znamená, že analogové vstupy **A 0..7** tohoto modulu budou ukládány s identifikátory po řadě 40 ..47.

Parametry **fixhour** a **fixmin** společně určují hodinu a minutu dne pro pravidelné denní uložení vzorku.

Standardní parametry modulu: **rychlost** a **priorita**.

### 7.11.3 HDBb



Modul **Hdbb** zpracovává binární signály a ukládá je do banky typu HDB.

Vstupy **B 0..15** jsou binární vstupy, jejich stav je ukládán do HDB.

Výstup **AI** (AlarmOn červený) indikuje příznak vzniku alarmu.

Výstup **AO** (AlarmOff zelený) indikuje příznak zániku alarmu.

Vstup **Clr** (ClearOn červený) nuluje příznak vzniku alarmu.



Obr. 173 Značka modulu **HDBb** na ploše

Vstup **Clr** (ClearOff zelený) nuluje příznak zániku alarmu.

Parametry **flags0..15** parametrizují svými váhami (bity) způsob ukládání daného binárního vstupu:

Váhy parametru flags0..15 modulu HDBb		
bit	flag	význam
0	n10	odhlášení alarmu při změně z 1 do 0
1	n01	odhlášení alarmu při změně z 0 do 1
2	a10	nastavení alarmu při změně z 1 do 0
3	a01	nastavení alarmu při změně z 0 do 1
4	wmn	zápis do HDB 1 x denně v 00:00 hod.
5	wtm	zápis do HDB vždy v určený čas (daný parametry <b>fixhour</b> a <b>fixmin</b> ) 1 x denně
6	h10	zápis do HDB při změně z 1 do 0
7	h01	zápis do HDB při změně z 0 do 1
8	han	po zápisu do HDB se aktuální hodnota nuluje, 0.. bin act je beze změny
9	on	je povolen zápis signálu do HDB
10	set	aktuální hodnota se při nenulovém vstupu nastaví do 1 [jinak beze změny]
11	ian	při resetu systému se aktuální hodnota nuluje [jinak beze změny]

Např. pro povolení ukládání vstupu **B 0** při každé změně do log.1 i do log.0 musí být aktivní flagy *h10*, *h01* a *on*, tj. bity 6, 7, 9, tzn. že parametr **flags0** bude mít hodnotu 704.

Parametry **interval 0..15** určují pro každý bin. vstup periodu ukládání v minutách.

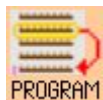
Parametr **index** určuje počátek hodnot identifikátoru vzorku.

Např. *index = 20* znamená, že bin. vstupy **B 0..15** tohoto modulu budou ukládány s identifikátory po řadě 20 ..35.

Parametry **fixhour** a **fixmin** společně určují hodinu a minutu dne pro pravidelné denní uložení vzorku.

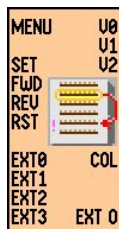
Standardní parametry modulu: **rychlost** a **priorita**.

## 7.11.4 Program



Modul **Program** slouží pro realizaci programového řízení.

Modul **Program** umožňuje zadat program až se 32 kroky, každý krok jednoho modulu **Program** přepouští na výstupy **V0..2** hodnoty tří parametrů **value0..2col0..31**.



Obr. 174 Značka modulu Program na ploše

Definovaným propojením až pěti modulů **Program** lze dosáhnout maximálně 15ti výstupních hodnot na jeden krok.

Umožňuje spouštění, zastavování, přednastavení, každý krok může být časovaný a může přeskakovat na libovolný jiný krok.

Všechny parametry **Programu** lze editovat z menu modulu.

Jedna z možností je realizace týdeních hodin (ve spolupráci s modulem **timdat**) nebo generování programových průběhů řídicích veličin pro testovací procesy.

Parametrem **columns** definujeme počet sloupců/kroků (max.32) programovací tabulky.

Text pro tři řádky tabulky (např. názvy veličin) zadáváme přes **DIALOG MODULU** do parametrů **Text\_value\_0..2**.

Za text lze zapsat formát zobrazované hodnoty s identifikátorem 0..2 (0 pro hodnoty z prvního řádku tabulky ... 2 pro hodnoty třetího řádku).

Např.:

```
Text_value0: Teplota ###.##1
```

Hodnoty do tabulky před překladem lze nadefinovat pomocí parametru **value0..2col0..31**.

Aktuální sloupec hodnot je na výstupech **V0..2**, aktuální hodnota sloupce/kroku je na výstupu **Col**.

Posun na další sloupec hodnot provádíme pulzem na vstup **Fwd**.

Posun na předchozí sloupec hodnot provádíme pulzem na vstup **Rev**.

Pozicovat se za poslední sloupec (dle parametru **columns**) nelze.

Pulzem na vstup **Rst** skočíme na sloupec 0.

Hodnotou na vstup **Set** se nastavíme na požadovaný sloupec (pak musíme dát Set=0).

Vstup **Menu** napojujeme na výstupy **0..3** modulu **SCREEN** a tak zpřístupníme editaci hodnot tabulky.

Pozicování pak pomocí šipek.

Vstupy **Ext0..3** slouží pro napojení dalších modulů **Program** (výstup **Ext 0**). Tak jsou přidány další 3 řádky do sloupce, pokud je ovládání vstupu modulu totožné.



## 8 Příloha 1 - Základní funkce regulátoru PL2

Sestavíte-li regulátor z HW modulů systému **PROMOS line 2**, neobsahuje aplikační program, tj. program, který má danou technologii řídit. Pokud však máte moduly správně propojeny, naadresovány (periferní moduly mají adresový přepínač, adresa 0 je rezervována pro centrálu, adresa 2 pro ovládací panel CKDM-11) a napájeny ze zdroje, můžete vytvořený regulátor zapnout a odzkoušet funkci jednotlivých vstupů a výstupů.

### 8.1 Nastavení centrály CCPU-02/03

Z výroby je na centrále CCPU nastavena komunikační adresa 1 (na DILech) a komunikační rychlost 38400 Bd. Pro komunikaci s FREDem je určen na centrále kanál COM0. Ten je zároveň určen pro komunikaci s nadřazeným pracovištěm (PC s ControlWebem a příslušným ovladačem pro stanice PROMOS).

**Komunikační adresu** lze nastavit na přepínačích DIL přímo na centrále CCPU-02/03, a to v rozsahu 1 až 15. **POZOR!** Toto nastavení se projeví až po RESETu centrály, tzn., že buď musíte po přestavení adresy stlačit na centrále tlačítko RESET (ukryté nad tlačítkem A) nebo musíte vypnout a zapnout napájení centrály.

Pokud nastavíte na DILech adresu 0, lze ji pak nastavovat programově, a to změnou parametru ADDRESS modulu SYSMON (přes FREDa).

Nezapomeňte, že i po přestavení DILů na adresu 0 musíte RESETovat centrálu.

Programově lze nastavit adresu v rozsahu 1 až 255.

**Komunikační rychlost kanálu COM0** centrály CCPU lze nastavit i fiktivním použitím modulu **modem**, a to změnou jeho parametru COMSPEED.

Po RESETu zůstává komunikační rychlost zachována.

Nezapomeňte po provedené změně adresy či rychlosti na COMu0 centrály změnit stejně komunikační parametry ve FREDovi (volba PROMOSline2/komunikační parametry).

**Pro ovládání základních funkcí** regulátoru PL2 slouží tzv. **systémové menu**.

Do něj se dostanete po zapnutí regulátoru ihned, pokud regulátor neobsahuje aplikační program (ten se vkládá po sériové lince z grafického vývojového prostředí FRED).

Pokud regulátor PL2 aplikační program obsahuje, stačí cca 5 sec držet stisknutou klávesu **ESC** na ovládacím panelu CKDM-11 (z hlavního menu).

V případě problémů podržte na centrále tlačítka A+B a stiskněte RESET. Tlačítka A+B uvolněte až po přechodu do systémového menu.

Na displeji se objeví nabídka:

```
F1 Download
F2 Run      F4 Kanal 0
F3 Test I/O
No prog loaded
```

s tím, že poslední řádek je prázdný, pokud je v regulátoru aplikační program nahrán.

V tomto okamžiku je možné nahrávání aplikačního programu z FREDa, tzn., že není nutno přecházet z tohoto menu do menu F1 - Download (je tedy preferováno nahrávání bez stisku klávesy F1)..

### 8.2 F1 Download

Po stisku klávesy F1 je na displeji vypsáno **Guard Error**

a regulátor PL2 očekává naplnění aplikačním programem po sériové lince z FREDa.

#### **Poznámka 1:**

*Od tohoto okamžiku terminál nereaguje na stisk žádné klávesy a musí se do něj nahrát aplikační program. V této chvíli nesmí být modul připojen k dispečinku ani k prostředí FRED v ladícím režimu. Data přicházející z dispečinku nebo z FREDa budou považována za aplikační program a mohlo by tedy dojít ke zhroucení systému.*

*Pokud dojde k přechodu do tohoto menu omylem, stiskněte na centrále současně tlačítka A+B a držte je. Potom krátce stiskněte tlačítko reset a po vypsání systémového menu na displeji tlačítka A+B uvolněte.*

#### **Poznámka 2:**

*Z výroby je komunikační adresa implicitně nastavena na jedničku.*

*Komunikační rychlost kanálu COM0 regulátoru (určeného pro FREDa) je nastavena na 38 400 Bd.*

#### **Poznámka 3:**

*Po upgrade firmware v CCPU je smazán původní projekt (v RAM i ve FLASH). Zároveň je aktivována funkce, která způsobí smazání projektu v RAM vždy po restartu CCPU. Tato funkce se zruší prvním uložením projektu do Flash. Tímto je zabráněno používání (resp. spuštění) PL2 bez uložení projektu do FLASH).*

*Zároveň je umožněno nahrát z FREDa aplikační program (překlad) do centrály, která je v systémovém menu, a to bez nutnosti stisku F1 pro download (to aby se daly dělat dálkové upgrady).*

*Toto je funkční od verze firmware 2.1.*

### 8.3 F2 Run

Po stisku klávesy F2 přechází regulátor do režimu, který je dán aplikačním programem, který již je uložen v paměti regulátoru.

V paměti regulátoru musí být platný projekt. Není-li, je ve spodním řádku displeje vypsáno

**No prog loaded**

a stisk klávesy F2 je bez odezvy.

### 8.4 F3 Test I/O

Po stisku klávesy F3 je nabídnuto menu pro test jednotlivých periferních modulů:

```
F1 CBI      F4 CCPU
F2 CBO
F3 CAIO
+/- Adresa=01
```

Nejprve je nutno nastavit adresu modulu, který chceme testovat, a to pomocí napovídáných kláves + nebo -.

*Např. chceme-li testovat modul CBI-11 s adresou 4 (ta musí být nastavena na otočném přepínači v modulu), postupným stiskem klávesy + přestavíme zobrazovanou adresu (poslední řádek displeje) na hodnotu 4. Potom stiskem klávesy F1 přecházíme na vlastní test modulu CBI-11.*

Obecně - po nastavení adresy (pro F4 pro test i/o na centrále je jedno jaká je adresa nastavena) testovaného modulu stiskneme klávesu:

- **F1** pro test logických vstupů modulu typu CBI a CBIO
- **F2** pro test logických výstupů modulu typu CBO a CBIO
- **F3** pro test modulu typu CAIO (analogové vstupy, výstupy)
- **F4** pro test i/o na centrále

#### 8.4.1 test logických vstupů CBI/CBIO

Na displeji je vypsanó:

```
CBI vstupy
0000 0000 0000 0000
```

pokud na všech 16 vstupů modulu CBI není přivedeno snímané vstupní napětí, např. přes kontakty čidel z technologie.

Zleva pak je zobrazen stav prvního vstupu modulu 0/1 podle skutečné situace, atd.

Test ukončíme stiskem klávesy **ESC**.

Obdobně lze otestovat vstupy na CBIO.

#### 8.4.2 test logických výstupů CBO/CBIO

Na displeji je vypsanó:

```
CBO vystupy
0000 0000 0000 0000
```

a kurzor bliká pod první nulou.

Testujeme výstupy modulu CBO, jejichž hodnotu 0/1 (VYP/ZYP) volíme pomocí kláves "-" a "+".

Na další výstupy se dostáváme pomocí šipek **doprava/doleva** a tak postupně testujeme všech 16 ovládaných bitů výstupního registru.

Ke každému bitu přísluší LED na panelu modulu, každá čtvrtá LED je pouze signalizační a není na ni vázáno výstupní relé.

Výstupních relé je pouze 12 a při testu spínají pouze tehdy, je-li na modulu CBO povolena jejich funkce konfiguračním přepínačem E.

Test ukončíme stiskem klávesy **ESC**, všechny výstupy jdou pak do 0.

Obdobně lze otestovat výstupy na CBIO.

#### 8.4.3 test CAIO

Na displeji je vypsanó:

```
CAIO Line:
```

a je očekáváno zadání hodnoty v rozsahu 0 až 17, a to pro test a nastavení AD vstupů platí hodnoty 0..11 a pro test DA výstupů platí hodnoty 12..17.

##### 8.4.3.1 test AD vstupů

Po zadání čísla vstupu 0 až 11 je pro test zvoleného AD vstupu na displeji např. napsáno:

```
Value Ain01: 0.0
F2 H mez: 0.0
F3 D mez: 0.0
F4 Korekce: 0.0
```

V horním řádku je uvedena naměřená hodnota zvoleného AD vstupu.

- Po stisku **F2** lze zadat horní mez měřené hodnoty.

- Po stisku **F3** lze zadat dolní mez měřené hodnoty.

- Po stisku **F4** lze zadat korekci pro měřenou hodnotu.

Hodnoty mezi měřené veličiny jsou určeny zvoleným pigydomečkem, zadáváme přesné katalogové údaje.

Např. pro měření teploty čidlem Pt100 a zvoleným piggy SAIP-32 zadáme

- horní mez 106,74

- dolní mez -45,75

Korekci zadáváme jako posun (+/-) zobrazované hodnoty v prvním řádku proti momentálně naměřené hodnotě kontrolním měřičem.

Např. je-li zobrazována hodnota 54,3°C a kontrolní teploměr ukazuje 53,3°C, zadáme korekci = - 1,0.

Obecně:

Klávesou **ENTER** hodnotu schválíme, klávesou **ESC** se vracíme zpět na volbu testu nové pozice CAIO modulu.

Při zadávání hodnoty si musíme uvědomit, že je akceptováno pouze prvních 6 znaků !

Rozsah zadávaných hodnot je omezen v rozsahu -9999,9 až 99999,9.

Poznámka:

Parametry mezí a korekcí analogových vstupů zadávané do regulátoru pomocí klávesnice ovládacího panelu CKDM-11 jsou uloženy v jiné oblasti paměti centrály CCPU-02 než parametry mezí a korekcí, které jsou používány aplikačním programem.

Jsou-li parametry zadané pomocí klávesnice ovládacího panelu CKDM-11 regulátoru jednoznačně správné, je třeba zvolit v projektu v prostředí FRED u příslušného modulu CAIO-11 parametr **load=1**. Tato volba způsobí, že po PŘEKLADu nejsou přeneseny parametry mezí a korekcí modulu CAIO s tímto parametrem do oblasti paměti regulátoru vyhrazené pro aplikační program (projekt ve FREDovi). Přeneseny jsou parametry mezí a korekcí z oblasti paměti regulátoru, kam byly zadány ručně pomocí klávesnice ovládacího panelu CKDM-11, do oblasti paměti vyhrazené pro aplikaci (projekt ve FREDovi).

##### 8.4.3.2 test DA výstupů

Po zadání hodnot 12 až 17 volíme test příslušného DA výstupu, a to na skutečných pozicích 4 až 9 modulu CAIO, kde se předpokládá osazený modul pro analogový výstup.

Na displeji je předepsáno:

```
F2 Value Ao0: 0.0
```

Po stisku klávesy F2 lze zadat hodnotu výstupního analogového signálu v procentech, tj. v rozsahu 0 až 99 %.

Klávesou **ENTER** hodnotu schválíme, pak se dostane na zvolený DA výstup.

Klávesou **ESC** se vracíme zpět na volbu testu nové pozice CAIO modulu.

#### 8.4.4 test i/o na centrále CCPU-02

Ze systémového menu se klávesou F3 dostaneme do testů i/o a po stisku F4 volíme test i/o na centrále.

Na displeji ovládacího panelu se objeví:

```
Ao CCPU 00 00
Ai 26 41 53 28 64 07
Bo 0000
Bi 0000 0000
```

Přivedeme-li na **logický vstup** In0 modulu ccpu log.1, zobrazí se jako 1 v řádku Bi namísto první 0.

Obdobně lze otestovat připojení ostatních logických vstupů.

Kurzorovými šipkami lze kurzor nastavit do řádku Bo a klávesami + nebo - sepnout či rozepnout příslušný **logický výstup**.

Jeho stav lze sledovat v tomto řádku jako výpis 0/1.

Pokud je všech 6 univerzálních pozic pro **analogové vstupy** osazeno a zapojeno, lze změnou vstupní analogové veličiny měnit hodnoty ve výpisu na řádku pro test analogových vstupů.

První vstup je vlevo, následují ostatní. Hodnota je znázorněna v %, a to od 0 do 99. Pro test funkčnosti a celého zapojení vstupu to stačí.

Dvě univerzální pozice lze osadit pro **analogové výstupy**. Pokud jsou osazeny, otestujeme jejich funkci a celkové zapojení tak, že kurzor napozicujeme do zobrazovacího pole pro daný analogový výstup (první je vlevo). Potom klávesou + zvětšujeme analogový výstup po 5 %, klávesou - pak snížíme jeho hodnotu po 5 %.

## 8.5 F4 Kanal

Po stisku klávesy F4 máme možnost nadefinovat, na kterém z kanálů bude funkční komunikace tzv. QQ relací a zároveň se dozvědět aktuální nastavení komunikační adresy a rychlosti.

Standardně (z výroby) je nastaven kanál 0, komunikační rychlost 38400 Bd a adresa 1 pro FREDa (adresa 127 pro BIOS).

Tak získáváme možnost propojení mezi PC s FREDem a centrálou i na jiný kanál centrály.

Protože může být dispečink vybaven komunikačním ovladačem se stejným typem QQ relací, platí pak totéž i pro komunikaci centrály CCPU-02 s dispečinkem.

Po stisku klávesy F4 je na displeji (od verze firmware 2.502) zobrazeno:

```
Baud 38400 v2.506
Addr      1 28.01.03
VELKA     F1 Reset
+/- Kanal = 0
```

Údaj v prvním řádku je zobrazení hodnoty aktuální rychlosti kanálu pro QQ relace a zobrazení verze firmware.

Údaj ve druhém řádku je zobrazení hodnoty aktuální adresy stanice pro komunikaci a zobrazení data verze firmware.

Ve třetím řádku je zobrazena "velikost" centrály mala nebo VELKA a nápověda na stisk klávesy F1 pro reset = základní komunikační nastavení centrály.

Po stisku F1 je nastaven kanál 0, 38 400 Bd, prodleva 10 ms, adresa zůstává nezměněna (a lze kdykoliv nastavit na DILech centrály).

Rovněž je smazán aplikační program z RAM i FLASH.

POZOR - toto nastavení se projeví až po restartu centrály!

Proto centrálu vypněte a znovu připojte na napájení - dostanete se systémového menu, ve kterém bude ve spodním řádku výpis **No prog loaded**.

Velikost centrály malá/velká je odvozena od velikostí pamětí RAM a FLASH (viz poznámka dále).

Ve čtvrtém řádku je nápověda pro změnu kanálu klávesami + a -.

Klávesami + nebo - navolte požadovaný kanál COM0 / COM1 / COM2, na který chcete přesunout tzv. QQ relace.

Schvalte klávesou ENTER.

*Poznámka:*

*Zobrazená hodnota komunikační rychlosti zůstane zachována.*

*Tu lze měnit pouze změnou aplikačního SW, tj. překladem z FREDa, kde v projektu je použit modul **modem** s parametrem **comspeed**, který tuto rychlost definuje.*

*Adresu stanice lze měnit buď na DILech centrály v rozsahu 1-15 a je platná po resetu centrály, nebo opět změnou aplikačního programu, a to pomocí modulu **sysmon**.*

### **Zobrazování adresy v systémovém menu**

Když se nastaví adresa na DILech, tak se po spuštění firmware přepíše do BIOSu a platí pro FREDa ihned a pro BIOS po restartu, v systémovém menu se objeví ihned.

Když na DILech=0, pak se nastaví pro FREDa adresa 0, dokud se nenahraje projekt; v okamžiku nahrání projektu se nastaví pro FREDa adresa z projektu a přepíše se do nastavení biosu - v systémovém menu se objeví okamžitě, pro BIOS se projeví po restartu.

Když se nenahraje projekt, nebo v projektu není modul sysmon, zůstává pro BIOS adresa nezměněna, tj. původní, která tam byla naposled.v

V systémovém menu je stav při DILech=0 indikován:

"0 / skutečná adresa BIOSu".

**F1 reset** potom nastaví zpět kanál, kom. rychlost a prodlevu, adresa zůstává nezměněna, lze vždy nastavit na DILech.

### **Velikosti centrály:**

Malá =

256kB RAM + 256kB FLASH (EI5712.80)

nebo 1 MB RAM + 256kB FLASH (EI5712.82)

nebo 256kB RAM + 512kB FLASH (EI5712.84)

V malé může být až 1000 modulů v projektu, 64 kB dat projektu a 64 kB dat vzorků (4 banky \* 8 kanálů \* 528 vzorků).

Velká =

1MB RAM + 512 kB FLASH (pouze EI5712.85)

Ve velké může být až 2000 modulů v projektu, 128 kB dat projektu a 256 kB dat vzorků (4 banky \* 8 kanálů \* 2112 vzorků).

### **Komunikační kanály centrály**

Centrála má tři sériové kanály - COM0, COM1 a COM2.

Pokud některý z modulů určených pro komunikaci lze napojit na některý z nich, má tento modul parametr, kterým toto navolíme. Tak je zvolen potřebný komunikační protokol a pokud komunikační modul nemá další komunikační parametry, pak jsou navoleny i ty. Jinak se zadávají jako parametry komunikačního modulu.

Některé komunikační moduly volbu COM0/1/2 nemají, protože je u nich předepsáno povinné napojení na jeden z nich. Přesto mohou mít tyto moduly jako parametry některé z komunikačních parametrů (přenosovou rychlost apod.). Pak jsou jimi nadefinovány. Komunikační protokol je zvolen použitím komunikačního modulu v projektu.



Defaultně je COM0 (hlavní kanál) určen pro tzv. QQ relace, tj. pro napojení k PC s FREDem, pro napojení na dispečink s PC s příslušným SW pro QQ relace, pro napojení k PC s loaderem pro UpGrade firmware centrály.

Centrála CCPU umí na hlavním kanále i zjednodušený protokol **modbus** - bez přepínání. Modbus umožňuje přístup do modulů **COMPORT** a **SCREEN**, a to tak, že **Register address Hi** (terminus technikus protokolu) určuje instanci daného modulu, a **Register address Lo** je pro výstupy **COMPORT 0..31**, vstupy modulu **COMPORT 32..63**. Pro přístup do modulu **COMPORT** musí být tato položka násobek 2. **Register address Lo** 64, 66, 68 a 70 jsou vyhrazeny pro přístup do modulu **SCREEN**, a to na výstupy **edit0..edit3**. Je možné vždy zapisovat i číst, délka dat musí být násobek 4 (tj. No. of Points musí být násobek 2).

### Od verze FW 3.002

- změna nastavení výchozí komunikační rychlosti a adresy bez projektu
- nastavení rychlosti se přebírá z BIOSu - výhodné pro dálkový UpG FW
- při problémech: v SYSTEMOVEM MENU centrály a po F4 - KANAL volit F1 = RESET (smaže i projekt), čímž se nastaví výchozí komunikační rychlost na 38400Bd
- výchozí adresa (pokud je na DILech 0) se bere z BIOSu
- při problémech nastavit adresu na DILech, nahrát projekt a pak vrátit DILy na 0, tím se nastaví adresa z projektu jak pro FREDa, tak pro BIOS (pro UpG)

### Od verze FW 3.004

- na USB jsou QQ relace při přepnutí hlavního kanálu na kterýkoliv COM centrály
- byla přidána USB komunikace (pouze protokol QQ) při zvolení hlavního komunikačního kanálu 0, 1, 2 nebo 4.
- pozor na pokus o současné nahrání projektu ze dvou kanálů najednou - kanál USB a zvolený hlavní kanál se navzájem vylučují (pokud se nahrává jedním, nejde nahrát druhým a naopak).

**Od této verze FW lze do CCPU-03 nahrávat projekt po USB při zvolení jakéhokoliv hlavního kanálu.**

## 8.6 CCPU-21 základní funkce

### Zapnutí

Pokud není v CCPU-21 platná konfigurace z FREDa, přejde CCPU-21 ihned do systémového menu. Tento stav je indikován trvalým svitem LED RUN na panelu (zelená).

Pokud je v CCPU-21 platná konfigurace, spustí se aplikace. Tento stav je indikován blikáním LED RUN (zeleně).

### Přechod do systémového menu

Pokud během restartu/zapnutí je stisknuto tlačítko **Fn**, centrála po zapnutí přejde vždy do systémového menu (obdoba A+B+reset u CCPU-02/03).

### Závada hardware

Pokud by došlo závadou HW CCPU-21 k narušení obsahu paměti flash, po startu zůstanou svítit na panelu všechny LED (obdoba rychlého blikání LED RUN+ERR u CCPU-02/03).

## Ovládání

Pomocí devíti tlačítek na panelu je možné ovládat CCPU-21 podobným způsobem, jako z terminálu CKDM.

### Základní význam tlačítek

Základní význam tlačítek je na nich vyznačen černě. Pomocí šipek a tlačítek **ENTER** a **ESCAPE** je možné se pohybovat v menu stejně, jako z ovládacího terminálu CKDM. Dlouhým stiskem (5 sec) tlačítka **ESC** lze přejít do systémového menu.

Tlačítka +/- lze v některých menu nastavovat hodnotu stejným způsobem, jako u CKDM.

### Druhý význam tlačítek

Druhý význam tlačítek se volí stiskem tlačítka **Fn**. V tomto stavu mají tlačítka funkci, která je na nich vyznačena červeně. To je indikováno zobrazením menšího **F** v pravém horním rohu LCD. V tomto režimu jsou k dispozici klávesy **F1..F5**, umožňující přímý přechod do menu, dále tlačítko **DEL**, umožňující mazat chybová hlášení, a dále tlačítka +/-, mající funkci odpovídající Shift+ a Shift- na CKDM (umožňují větší krok nastavování hodnot v některých menu).

### Význam tlačítek v editačním režimu

Editační režim se volí dvojitým stiskem tlačítka **Fn**. Druhý stisk tlačítka **Fn** odpovídá stisku klávesy INS na CKDM. Tento režim je indikován zobrazením menšího **E** v pravém horním rohu LCD. V editačním režimu mají tlačítka ten význam, který je na nich vytištěn modře. Stiskem tlačítka **Fn** v editačním režimu se zvolí druhý význam tlačítek v editačním režimu. Tento stav je indikován současným zobrazením menších **F** a **E** v pravém horním rohu LCD. V tomto stavu mají tlačítka ten význam, který je na nich vytištěn šedě.

*Poznámka: pokud se vstoupí do editačního režimu omylem (například na řádku menu, kde není nic k editaci), lze editační režim vždy zrušit Escapem (v editačním režimu to je sekvence Fn - ESC).*

## Speciální kombinace

### Blokování binárních výstupů CCPU-21:

stisknout a podržet tlačítko **Fn**. Dále stisknout tlačítko **DEL**. Stav blokování binárních výstupů je indikován LED BLK na panelu.

### Reset CCPU-21:

stisknout tlačítko **Fn**. Znovu stisknout a držet tlačítko **Fn**. Stisknout tlačítko **DEL**.

### Zobrazení

LCD zobrazuje obrazovky z modulu **SCREEN** stejným způsobem, jako terminál CKDM. Do projektu ve FREDovi je tedy třeba zařadit modul **CKDM11**. Navíc má možnost zobrazit stav vstupů modulu **CKDM11**, které na CKDM-11 ovládaly LED. Stav těchto vstupů indikují malé symboly **G** (jako Go, místo RUN na CKDM), **1** (odpovídá Mode 1 na CKDM), **2** (odpovídá Mode 2 na CKDM) a **E** (odpovídá Error na CKDM). Trvalé zobrazení těchto symbolů odpovídá svitu příslušných LED zeleně, blikání symbolů odpovídá svitu příslušných LED červeně.

Oproti CKDM umí CCPU-21 zobrazovat malá písmena s diakritikou. Znak stupeň Celsia má stejně jako na CKDM kód 223.



## Download

Režim download je indikován zhasnutím LED RUN a svitem LED ERR červeně.

## Systemové menu

Systemové menu nemá, na rozdíl od CCPU-02/03, volby pro test periferií na sběrnici CAN. Naopak, navíc má možnost nastavení komunikační adresy v menu **F4 Kanál** (funkce odpovídá nastavení adresy na DILech u centrály CCPU-02/03). Adresa se v tomto menu nastavuje tlačítky šipka nahoru/šipka dolů, potvrzuje se stiskem klávesy **ENTER**. Je možné nastavit adresu 0..15, význam nastavení je stejný jako u centrály CCPU-02/03. Změna nastavení adresy se projeví až po restartu centrály, a to i v menu **F4 kanál**.

Test I/O CCPU-21 je realizován pouze jednoduchým způsobem, obdobně jako test CCPU-02.

## Verze firmware

Číslo verze firmware je shodné s odpovídající verzí firmware pro CCPU-02/03. Rozdíly ve firmware jsou v tom, že firmware pro CCPU-21 nemá podporu sběrnice CAN, a v systemovém menu, viz výše.

Firmware pro CCPU-21 a CCPU-02/03 však nelze vzájemně zaměňovat.

*Poznámky:*

*Označení **FW 3.007** je pro CCPU-02/03 i CCPU-21 totožné.*

*FRED je tedy v příslušné verzi stejný pro všechny centrály, které má v knihovně.*

Soubory s příponou .S24 pro FW centrály CCPU-02/03 se nadají zaměňovat se soubory pro FW centrály CCPU-21.

## 9 Příloha 2 - Zásuvné moduly pro analogové i/o

Výměnný modul pro definici univerzální pozice obsahuje operační zesilovač s odporovou sítí a podle typu umožňuje měření napětí, proudu, odporu nebo přímé připojení odporového čidla Pt100, Ni1000. Typ signálu a rozsah měření je dán typem výměnného piggymodulu. Tyto moduly se používají pro definici analogových i/o HW modulů **CCPU-02** a **CAIO-11**.

Tyto piggymodulky jsou stejné pro CAIO-11 i CCPU-02 **kromě jediné výjimky**. Piggymoduly **pro analogové výstupy** se musí používat přesně podle jejich určení. Jedna typová řada je určena pro CCPU-02, druhá typová řada je určena pro CAIO-11. Obvodově jsou sice stejné, ale liší se časovou konstantou filtru.

V dalších tabulkách jsou uvedeny typy jednotlivých zásuvných modulů pro měření teplot a pro každý modul pak přesné horní a dolní meze hodnot, které se zadávají jako SW parametry. Protože se HW moduly liší i AD převodníky (10/14-bitový), liší se u stejného typu piggymodulu i meze pro jednotlivé HW moduly, ve kterých jej použijete. Proto zadávání SW parametrů (mezi) věnujte zvýšenou pozornost.

Moduly pro teploměry Pt100					
obj. číslo	typ	rozsah měření	dolní mez	horní mez pro	
				CAIO	CCPU
EI5907.00	SAIP-00	-200÷0°C	-208,49 °C	6,66 °C	6,45 °C
EI5907.01	SAIP-01	-200÷50°C	-208,49 °C	51,37 °C	51,11 °C
EI5907.21	SAIP-21	-100÷50°C	-110,77 °C	52,17 °C	52,02 °C
EI5907.22	SAIP-22	-100÷100°C	-110,77 °C	114,09 °C	113,87 °C
EI5907.23	SAIP-23	-100÷200°C	-110,77 °C	301,70 °C	301,28 °C
EI5907.31	SAIP-31	-50÷50°C	-45,75 °C	57,23 °C	57,13 °C
EI5907.32	SAIP-32	-50÷100°C	-45,75 °C	106,74 °C	106,60 °C
EI5907.33	SAIP-33	-50÷200°C	-45,75 °C	210,77 °C	210,52 °C
EI5907.34	SAIP-34	-50÷300°C	-45,75 °C	302,67 °C	302,33 °C
EI5907.42	SAIP-42	0÷100°C	0,00 °C	105,79 °C	105,69 °C
EI5907.43	SAIP-43	0÷200°C	0,00 °C	204,42 °C	204,22 °C
EI5907.44	SAIP-44	0÷300°C	0,00 °C	318,32 °C	318,00 °C
EI5907.45	SAIP-45	0÷400°C	0,00 °C	434,67 °C	434,23 °C
EI5907.46	SAIP-46	0÷600°C	0,00 °C	615,95 °C	615,30 °C

Moduly pro teploměry Ni1000 s 5000 ppm					
obj. číslo	typ	rozsah měření	dolní mez	horní mez pro	
				CAIO	CCPU
EI5906.31	SAIN-31	-50÷50°C	-60,46 °C	59,62 °C	59,51 °C
EI5906.32	SAIN-32	-50÷100°C	-60,46 °C	103,58 °C	103,45 °C
EI5906.33	SAIN-33	-50÷200°C	-60,46 °C	199,55 °C	199,35 °C
EI5906.41	SAIN-41	0÷50°C	0,00 °C	52,57 °C	52,52 °C
EI5906.42	SAIN-42	0÷100°C	0,00 °C	104,86 °C	104,77 °C
EI5906.43	SAIN-43	0÷200°C	0,00 °C	201,77 °C	201,61 °C
EI5906.44	SAIN-44	0÷300°C	0,00 °C	309,85 °C	309,63 °C

Moduly pro teploměry Ni1000 s 6180 ppm					
obj. číslo	typ	rozsah měření	dolní mez	horní mez pro	
				CAIO	CCPU
EI5906.31	SAIN-31	-50÷50°C	-48,42 °C	48,80 °C	48,71 °C
EI5906.32	SAIN-32	-50÷100°C	-48,42 °C	85,90 °C	85,79 °C
EI5906.33	SAIN-33	-50÷200°C	-48,42 °C	171,15 °C	170,97 °C
EI5906.41	SAIN-41	0÷50°C	0,00 °C	42,93 °C	42,89 °C
EI5906.42	SAIN-42	0÷100°C	0,00 °C	87,00 °C	86,92 °C
EI5906.43	SAIN-43	0÷200°C	0,00 °C	173,20 °C	173,06 °C
EI5906.44	SAIN-44	0÷300°C	0,00 °C	277,46 °C	277,23 °C

Moduly pro měření odporu - aktivní můstek					
obj. číslo	typ	rozsah měření	dolní mez	horní mez pro	
				CAIO	CCPU
EI5905.00	SAIA-00	0 ÷ 105 Ohm	0,00 Ohm	114,40 Ohm	114,29 Ohm
EI5905.01	SAIA-01	0 ÷ 130 Ohm	0,00 Ohm	134,18 Ohm	134,06 Ohm
EI5905.02	SAIA-02	0 ÷ 600 Ohm	0,00 Ohm	609,62 Ohm	609,06 Ohm
EI5905.03	SAIA-03	0 ÷ 1000 Ohm	0,00 Ohm	1026,96 Ohm	1026,01 Ohm

Moduly pro měření odporu - pasivní můstek					
obj. číslo	typ	rozsah měření	dolní mez	horní mez pro	
				CAIO	CCPU
EI5904.00	SAIB-00	0 ÷ 105 Ohm	0,00 Ohm	111,73 Ohm	111,63 Ohm
EI5904.01	SAIB-01	0 ÷ 130 Ohm	0,00 Ohm	134,45 Ohm	134,32 Ohm
EI5904.02	SAIB-02	0 ÷ 600 Ohm	0,00 Ohm	605,00 Ohm	604,42 Ohm
EI5904.03	SAIB-03	0 ÷ 1000 Ohm	0,00 Ohm	1011,70 Ohm	1010,71 Ohm

**Poznámka:**

*U modulů pro měření napětí a proudů horní a dolní meze odpovídají přesně katalogovým údajům.*

*Uvedené údaje pro meze nemusíte po úpravách SW zadávat ve FREDovi "ručně", protože jak u CCPU-02, tak i u CAIO-11 stačí otevřít dialogové okno modulu a v něm přímo navolit typ zásuvného modulu. Meze se tak dosadí automaticky.*

*V prodeji je rovněž nový modul pro analogové vstupy / výstupy, a to typ CAIO-12. Tento modul se vyznačuje 16bitových AD převodníkem a novými zásuvnými moduly, které obsahují i paměť EEPROM, ve které jsou obsaženy pro každý I/O jak meze měřeného rozsahu, tak i linearizační parametry. Po zapnutí modulu si tento potřebné údaje vyčte a zařadí do SW.*

**Proto pro návrh nových regulátorů používejte již modul CAIO-12.**

## 10 Příloha 3 - Seznam objektů knihovny pro chybová hlášení

objekt	modul	objekt	modul
2	ModbusTX	107	and10
3	CCPU-03	108	amul
4	cntud	109	adiv
5	regl	110	aplus
6	regf	111	amin
7	PBI11	112	asmall
8	FCPU02	113	abig
9	PBO11	114	amux
10	flash	115	admx
11	CCPU-02	116	table
12	CANTERM	117	modem
13	SCREEN	118	<b>canopen</b>
14	<b>serialcomm</b>	119	int_to_bin
15	svm840/940	120	bin_to_int
17	CBO	121	<b>compute</b>
18	CAIO	122	constf
19	CBI	123	constl
20	amux10	124	idiv
21	cf50	125	imin
22	keyin	126	imul
23	keyout	127	iplus
24	int2float	128	aftt
25	float2int	129	almt
26	stridac	130	cnt
27	urci_tv	131	ctc
28	pockot	132	mono
29	CBIO	133	rsko
30	danfoss	134	jklop
31	EmergCall	135	pwpb_rx
32	cdanfoss	136	pwpb_tx
33	comport	137	pwpb_main
34	fastpid	138	flag
35	<b>M-Bus</b>	139	minmax
36	R3state	140	sdo
37	Program	141	gsm
38	admx10	142	sms
39	State	143	pwpbr
40	CAIO12	144	SKDM12
41	scale	145	SKDM11
42	EESA	146	ppg
43	IEC870	147	<b>journal</b>
44	<b>CCPU-21</b>	148	journal_txt
45	Inet	149	print
46	PortMix	150	dcf
47	SaveEn	151	ekviterm
48	TimerA	152	pid
49		153	<b>havbin</b>
50		154	<b>havan</b>
100	and2	155	<b>vzorky</b>
101	or2	156	aservo
102	xor2	157	bservo
103	not	158	body_4
104	delay	159	setpar
105	or10	160	timer
106	xor10	161	timdat



objekt	modul
162	CKDM12
163	hesla
164	<b>sysmon</b>
165	CKDM11
166	ReTran
167	I2hwIw
168	SAIOi
169	SBIOi
170	SBI
171	SFCPUi
172	SAIOo
173	SBIOo
174	SBO
175	SFCPUo
176	
177	
178	
179	
180	
181	
182	
183	
184	Hdbb
185	Hdba
186	Hdbmain
187	ModbusTXC
188	IECinSP
189	IECinDP
190	IECinMI
191	IECinMF
192	
193	
194	
195	
255	ModbusRX

Názvy modulů, uvedené tučným písem, vyvolávají chybová hlášení, a to buď v textovém tvaru, definovaném jako parametr modulu (např. moduly HAVAN, HAVBIN) nebo ve tvaru

**Er: chybovy\_kod\_MSB, chybovy\_kod\_LSB @ obj, inst**  
Hodnoty MSB a LSB jsou uvedeny v HELPu FREDa.

## 11 Multi Point Communication

Jedná se o vícebodovou /mezicentrálovou/ komunikaci stanic PROMOS line 2 protokolem EPSNET / ProfiBus, kterou lze využít s příslušnými ovladači i pro komunikaci s nadřízeným dispečinkem (PC + dispečerský SW + ovladač).

Každá stanice může být řídicí (master) nebo podřízená (slave).

Stanice master aktivně vysílá data nebo žádá o data.

Stanice slave pouze odpovídá na dotazy. Dotaz může být jak žádost o čtení dat ze slave stanice, tak žádost o uložení dat do slave stanice.

Společné komunikační parametry se definují v modulu **PWPB\_MAIN**.

Délka bloku přenášených dat je 64 bajtů (lze definovat parametrem **DELKA**, 64 bajtů právě odpovídá množství dat zpřístupňovaných moduly **pwpb\_rx** a **pwpb\_tx**). V případě potřeby komunikace s jiným zařízením než s centrálou CCPU-02 (například se sériovými periferními jednotkami SBI, SBO, SBIO nebo SAIO), které může být do sítě připojeno jako slave zařízení, je možné nastavit délku bloku přenášených dat i menší, podle specifikace komunikačních dat připojeného zařízení.

Data v síti Epsnet jsou definována číslem bloku, který má maximální délku 64 kB.

Parametrem **OFFSET** se dále definuje posun komunikovaných dat v daném bloku.

Definice jedné přenášených dat na přijímací a vysílací straně si musí odpovídat (stejně číslo bloku a stejný offset v bloku). Jiná přenášená data (jiné veličiny) naopak musí mít definici (určení) jinou, aby se nepřekrývaly už s jinde definovanými daty a nedocházelo k jejich vzájemnému přepisování.

Sama vysílat umí pouze stanice MASTER (tedy stanice, u které je parametr **MAXTOKEN >0**).

Po spuštění komunikace zahajuje stanice MASTER vysílání přebráním tokenu (vysílacího oprávnění).

Pokud je stanic MASTER více, začíná ta s nejmenší adresou.

Prakticky je to po počáteční prodlevě ta stanice MASTER, jejíž hodnota výrazu **ADRESA\*ODEZVA** (parametry stanice) je nejnižší. Aby tento systém fungoval v síti s více stanicemi master, je třeba, aby měly všechny stanice MASTER parametr **ODEZVA** stejný.

**TOKEN** (vysílací oprávnění) si posílají stanice mezi sebou pomocí speciální krátké zprávy, a to postupně po adresách (modulo + 1, tj. např. stanice s adresou 1 stanici s adresou 2, stanice 2 stanici 3, ...). Poslední takto obsluhovaná adresa je dána parametrem **MAXADRESA** v modulu **PWPB\_MAIN**.

Stanice mající **TOKEN** vyřizuje po dobu trvání svého **TOKENU** komunikační požadavky podle definice svých modulů **PWPB\_RX** a **PWPB\_TX**.

Je tedy vhodné, aby doba trvání tokenu stanice (**MAXTOKEN**) byla alespoň tak dlouhá, aby byla stanice během jejího trvání schopna vyřídit všechny svoje komunikační požadavky.

**Podle definice stanic master/slave v komunikační síti lze rozdělit síť na:**

### - multislave

Všechny stanice jsou podřízené, většinou dispečerskému PC, který je master.

SLAVE odpovídá na požadavky dispečinku.

Stanice slave musí mít nastaven parametr **ODEZVA** co nejdelší, nejlépe 30000.

### - monomaster

Jediná stanice MASTER, ostatní SLAVE.

Síť stanic PL2, jedna z nich je řídicí (tedy MASTER) - tato stanice vyřizuje centrálně všechny komunikační požadavky, tedy předávání dat mezi SLAVE stanicemi přímo není možné, pouze zprostředkovaně přenosem přes MASTER stanicí.

MASTER stanice by neměla mít nastaven parametr **ODEZVA** na dobu podstatně delší než na dobu nezbytně nutnou k dokončení příjmu odpovědi na komunikační požadavek, aby nebyla komunikace při náhodném výpadku zbytečně bržděna.

### - multimaster

Více stanic řídicích.

Takto lze nadefinovat, že několik stanic může aktivně vyřizovat komunikační požadavky.

To umožňuje předávání dat nejen mezi libovolnou MASTER a SLAVE stanicí, a také mezi MASTER stanicemi mezi sebou navzájem. MASTER stanice obsluhuje jak požadavky na čtení, tak na zápis. Každá MASTER stanice si hlídá čerstvost svých komunikačních dat a v případě zjištěné prodlevy delší než polovina doby timeoutu (parametr **PERIODA**) pro daná data iniciuje komunikační relaci přenosu těchto dat.

## 12 Historická databanka HDB

### Informace o historických databankách typu HDB v centrálních CCPU-02 a CCPU-03:

Od verze FW 3.003 lze použít v projektu jako historické databanky buď moduly **VZORKY** nebo moduly **HDB**, a to proto, protože používají stejnou oblast paměti.

Databanky se liší vnitřním uspořádáním dat, ovládním, .... další informace jsou pro banky typu HDB.

**U velké centrály** má velikost 50000 vzorků x 10byte na vzorek, celkem zabere dohromady 500 000 bytů paměti.

**U malé centrály** má velikost 6500 vzorků x10 byte na vzorek, celkem tedy zabere 65 000 bytů.

Jeden vzorek obsahuje po řadě 2 byte identifikátor dat, 4 byte počet sekund v okamžiku uložení vzorku od 0:00:00 1.1. 2000, a 4 byte dat.

Data jsou ukládána jako 4 bytový float, v případě, že uložená hodnota je binární, pak log. 1 odpovídá číslu 1.00 a log. 0 odpovídá číslu 0.00.

Historická banka HDB je uspořádána jako kruhový seznam, nové vzorky jsou ukládány s rostoucím indexem.

První vzorek je ukládán s indexem 0 (na počatek banky). V případě, že dojde k zaplnění banky, nové vzorky přepisují ty nejstarší. Ty jsou potom ztraceny.

### Čtení vzorků banky.

Ke čtení vzorků bank HDB se používají zprávy QN. Banka pro malou centrálu je 0xF0, pro velkou centrálu je paměť hist. banky zobrazena postupně v bankách 0xF0, 0xF1, 0xF2, 0xF3, 0xF4, 0xF5, 0xF6 a 0xF7. Využit je prostor 500 000 bytů od offsetu 0 první banky.

Při vyčítání dat blízko vrcholu datového prostoru poslední banky je potřeba zarovnat počet bytů čtené zprávy s koncem datového prostoru (poslední platný byte je v bance 0xF7 a offsetu 0xA11F) a pokračovat ve čtení znovu od začátku datového prostoru.

Maximální počet přenášených datových bytů ve zprávě je omezen na 192 byte.

### Informace o uložených datech v HDB.

Informace poskytuje modul **HDBmain** (objekt No. = 186, inst. = 1), který musí být součástí projektu systému PL2.

Datový objekt **item 13** určuje počet platných vzorků v bance. Tato hodnota se smazáním banky nuluje. Banka se nuluje aktivací 0-tého bitu vstupu **Erase**. Banka se také nuluje automaticky při nahrání nového projektu.

**Item 14** určuje vrchol banky. Je to index, na který bude zapsán příští vzorek. Jeho hodnota se vymazáním banky nuluje.

Dále **itemy 5..12** (userpos) umožňují uložit pozici (index) až pro 8 uživatelů, určující, kterým vzorkem je potřeba začít příští čtení. Jejich hodnota se vymazáním banky nuluje.

Dále je jejich hodnota posouvána dopředu v případě, že daný uživatel nevyzvedl včas data a tato data jsou přepisována daty novými. Inkrementace probíhá tak, že tyto objekty

v případě vymazání nevyzvednutých hodnot ukazují vždy na nejstarší platný vzorek v databázi.

Pro každého uživatele je dále udržováno počítadlo přetečení (**itemy 15..22**). V těchto itemech se zaznamenává počet nevyzvednutých ztracených vzorků daného uživatele. Tato počítadla se dají vynulovat aktivací bitu 1..8 (po řadě každý bit pro každého uživatele) vstupu **Erase** modulu **HDBmain**.

Vztah mezi indexem vzorku a fyzickým offsetem paměti databáze je:

$$\text{offset} = 10 * \text{index}$$

### Postup vyčtení nových vzorků.

Nejdříve daný uživatel vyčte pozici vrcholu databáze (item 14), a svůj ukazatel počátku čtení (userpos N, itemy 5..12). Tyto dvě položky je nutné vyčíst jednou QQ zprávou, aby byla zajištěna jejich konzistence, v případě, že současně probíhá zápis do HDB.

Poté neprodleně začne vyčítat vzorky od indexu daném svým userpos (včetně), do indexu daném vrcholem HDB (avšak mimo něj). Po ukončení čtení uloží do svého parametru **userpos** hodnotu na počátku vyčteného vrcholu databáze (item 14).

Popisovanou sekvenci je potřeba dodržet, aby nedocházelo k přeskokování vzorků při následném vyčítání dat z HDB a současném kontinuálním zápisu dat (vzorků do HDB).

Je potřeba brát v úvahu to, že pokud uživatel začíná vyčítat vzorky z banky, která právě přetéká a jsou do ní dále zapisovány nové vzorky, které "vytlačují" **userpos** daného uživatele směrem k vzorkům novějším, je možné, že prvních několik vyčtených nejstarších vzorků bude v době mezi vyčtením stavu banky a vyčtením prvních dat přepsány vzorky nejnovějšími.

Je potřeba, aby aplikace tento stav dokázala ošetřit. Vzhledem k tomu, že k tomuto nebezpečí dochází až když už jsou některé nejstarší vzorky stejně ztraceny, a o nové se nepřichází (budou vyčteny při příštím čtení bez nutnosti dalšího ošetřování) není toto problém.

Nejjednodušší způsob ošetření je testovat, zda  $\text{userpos} = \text{vrchol} + 1$

a když ano, začít číst až o zvolený počet vzorků dále, resp. zvolený počet nejstarších vzorků ignorovat.

Exaktní způsob je vyčíst **userpos** znovu na konci čtení, a v případě, že se liší od hodnoty vyčtené na počátku čtení, prohlásit za platné vzorky až od této nové hodnoty **userpos**. Tímto způsobem se ale mohou zahodit i vzorky platné, pokud tempo vyčítání není podstatně rychlejší než tempo ukládání vzorků.

Při vyčítání je potřeba správně přecházet z konce datového prostoru na začátek (viz výše).