

Čítačové vstupy, měření otáček/impulsů (PseDet)

Abstrakt

U řídicích systémů s operačním systémem NOS a u rozšiřujících modulů typu DM(M)-DI24 lze každý číslicový vstup využít pro čítání impulsů. Jakým způsobem je lze využít je popsáno v této aplikační poznámce.

Autor: Zbyněk Říha
Dokument: ap0017_cz_05.pdf

Příloha

Obsah souboru: ap0017_cz_03.zip

citacovevstupy_p1_cz_03.dso	Příklad č. 1 – programový čítač
citacovevstupy_p2_cz_03.dso	Příklad č. 2 – programový čítač INT, měření otáček
citacovevstupy_p3_cz_03.dso	Příklad č. 3 – hardwarový čítač
citacovevstupy_p4_cz_03.dso	Příklad č. 4 – čítání impulsů na DM-DI24
citacovevstupy_p5_cz_03.dso	Příklad č. 5 – čítání impulsů na DMM-DI24

Obsah

	Obsah	2
	Historie revizí	3
	Související dokumentace.....	3
1	Definice použitých pojmů	4
2	Čítačové vstupy, měření otáček/impulsů	5
3	Programový čítač	6
3.1	Programová obsluha	6
3.1.1	Modul Impln	6
3.1.2	Modul DImp.....	6
3.2	Příklad č. 1 – programový čítač	6
	Proces Quick, perioda 50 ms	7
	Řádný proces, perioda 5 s.....	7
4	Programový čítač INT	8
4.1	Interrupt procesy	8
4.2	Programová obsluha	8
4.2.1	Měření otáček	9
4.3	Příklad č. 2 – programový čítač INT, měření otáček	9
	Proces Interrupt_0.....	9
	Proces Interrupt_1.....	9
	Řádný proces, perioda 1 s.....	9
5	Hardwarový čítač.....	10
5.1	Režimy vstupu.....	10
5.2	Vlastnosti čítačových vstupů	11
5.3	Programová obsluha	12
5.4	Příklad č. 3 – hardwarový čítač.....	12
	Proces INIT	13
	Řádný proces, perioda 10 s.....	13
6	Čítání impulsů pomocí rozšiřujících modulů	14
6.1	Rozšiřující modul DM-DI24.....	14
6.1.1	Příklad č. 4 – čítání impulsů na DM-DI24	14
	Načtení hodnot čítačů ze všech vstupů	14
	Nastavení hodnoty všech čítačů.....	15
6.2	Rozšiřující modul DMM-DI24.....	15
6.2.1	Příklad č. 5 – čítání impulsů na DMM-DI24	15
	Čtení hodnoty čítačů na všech vstupech	16
	Nastavení hodnoty čítačů	16
7	Technická podpora	17
8	Upozornění	18

Historie revizí

Verze	Datum	Autor změny	Změny
001	28. 11. 2008	Říha Z.	Nový dokument
002	08. 09. 2011	Říha Z.	Oprava maximální možné hodnoty čítání modulu DM-DI24 , oprava aplikace. Ukázkové aplikace vytvořeny v DetStudios verze 1.7.0.
003	20. 12. 2012	Říha Z.	Do kapitoly 6.1.1 doplněna informace o problému se zápisem hodnoty čítačů ve firmware 1.65.
004	26. 05. 2015	Říha Z.	Úprava kapitol 4.1, 5, 5.3.1 a 5.3.5.
005	14. 08. 2019	Říha Z.	Změna názvu AP, aktualizace seznamů HW dle aktuální nabídky, úprava HW v ukázkách dle aktuální nabídky, změna struktury dokumentu.

Související dokumentace

1. Návod k části PseDet vývojového prostředí DetStudio
soubor: PseDet_cz.chm
2. Návod na obsluhu řídicích systémů AMiT
soubor: xxx_g_cz_xxx.pdf
3. Aplikační poznámka AP0008 – Komunikace v síti MODBUS RTU (PseDet)
soubor: ap0008_cz_xx.pdf
4. Aplikační poznámka AP0016 – Zásady používání RS485
soubor: ap0016_cz_xx.pdf
5. Aplikační poznámka AP0025 – Komunikace v síti ARION – definice tabulkou
soubor: ap0025_cz_xx.pdf

1 Definice použitých pojmů

DetStudio

Vývojové prostředí firmy AMiT, které slouží pro parametrizaci řídicích systémů. Prostedí je možné stáhnout z amitautomation.cz.

Čítačové vstupy

Jsou číslicové vstupy, které jsou vybavené rychlými čítači a lze je využít pro čítání příchozích impulsů pomocí modulů **IRCxxx**.

Kanál

Skupina až šestnácti signálů (vstupů/výstupů) stejného typu (digitální/analogové).

RS485

Je poloduplexní sériová sběrnice umožňující komunikaci více jednotek na jednom signálovém páru. Více informací nalezete v aplikační poznámce *AP0016 – Zásady používání RS485*.

ARION

Je sériový poloduplexní komunikační protokol pro komunikaci řídicích systémů firmy AMiT s rozšiřujícími V/V moduly. Pomocí těchto rozšiřujících V/V modulů lze navýšit počet vstupů/výstupů řídicího systému.

MODBUS RTU

Je otevřený protokol pro vzájemnou komunikaci různých zařízení, který umožňuje přenášet data po různých sítích a sběrnících. Komunikace funguje na principu předávání datových zpráv mezi klientem a serverem (master a slave).

Vzdálený bod

Jedná se o definici registru/bináru, případně definici skupiny registrů/binárů, které odpovídají vstupům/výstupům na zařízení (slave) komunikujícím prostřednictvím protokolu MODBUS RTU.

Moduly DM-xxx

Moduly umožňující, prostřednictvím komunikační sítě ARION, rozšířit počet vstupů a výstupů řídicího systému. Do jedné sítě ARION lze připojit až 63 těchto modulů.

Moduly DMM-xxx

Moduly umožňující prostřednictvím komunikační sítě MODBUS RTU rozšířit počet vstupů a výstupů u zařízení, které komunikuje prostřednictvím protokolu MODBUS RTU a je masterem na této síti. Do jedné sítě MODBUS RTU lze připojit až 63 těchto modulů.

2 Čítačové vstupy, měření otáček/impulsů

Všechny řídicí systémy firmy AMiT, které jsou osazeny číslicovými vstupy, lze využít pro čítání impulsů resp. měření otáček.

Čítání impulsů resp. měření otáček může být realizováno (dle typu konkrétního řídicího systému) následujícími způsoby:

- ♦ Programový čítač
- ♦ Programový čítač INT
- ♦ Hardwarový čítač

Pokud není řídicí systém vybaven číslicovými vstupy a je vybaven komunikační linkou RS485 nebo RS232 (pro RS232 je nutné použít převodník RS232/RS485), lze pro čítání impulsů (do frekvence 25 Hz) využít rozšiřující moduly **DM-DI24** (komunikace protokolem ARION) nebo **DMM-DI24** (komunikace protokolem MODBUS RTU). Tyto skutečnosti jsou popsány v kapitole 6 „Čítání impulsů pomocí rozšiřujících modulů“.

Programový čítač

Každý číslicový vstup lze programově obsluhovat v Hi_x procesech. Programovou obsluhou lze tyto vstupy použít jako čítačové. Omezení kmitočtu vstupního signálu je dáno programem. Typicky lze použít do kmitočtu 250 Hz. Bližší informace jsou uvedeny v kapitole 3 „Programový čítač“.

Programový čítač INT

Některé číslicové vstupy mohou generovat hardwarové přerušení (interrupt). Programovou obsluhou lze pomocí Interrupt_x procesů tyto vstupy použít jako čítačové, resp. pro měření otáček. Omezení kmitočtu vstupního signálu je dáno programem. Typicky lze použít do kmitočtu 10 kHz. Bližší informace jsou uvedeny v kapitole 4 „Programový čítač INT“.

Hardwarový čítač

Některé řídicí systémy jsou vybaveny hardwarově podporovanými čítačovými vstupy, resp. vstupy pro inkrementální čidla polohy. Programová obsluha je zajišťována pomocí modulů **IRCxxx**. Bližší informace jsou uvedeny v kapitole 5 „Hardwarový čítač“.

3 Programový čítač

Je-li řídicí systém vybaven číslicovými vstupy, je možné tyto vstupy programově obsluhovat v rychlých procesech a využít je tak pro čítání impulsů. Parametrizace se provádí ve vývojovém prostředí DetStudio pomocí modulů **ImpIn** a **DImp**.

Omezení kmitočtu vstupního signálu je dáno programem. Typicky lze použít do kmitočtu 250 Hz.

Modul **DImp** se umísťuje v kterémkoliv řádném procesu. Modul **ImpIn** se umísťuje v rychlém procesu (proces typu Quick nebo Hi_x), jehož perioda se volí v závislosti na délce impulsu. Perioda procesu s modulem **ImpIn** musí být menší nebo rovna polovině délky impulsu, tzn., pokud impuls = 10 ms tak perioda rychlého procesu musí být 5 ms nebo menší.

3.1 Programová obsluha

3.1.1 Modul ImpIn

Modul **ImpIn** ošetřuje až šestnáct impulsních vstupů, načítaných z vybraného logického kanálu DI. Zpracované hodnoty neukládá do databáze, ale uchovává je ve svých vnitřních proměnných. Tyto hodnoty se dále zpracovávají pomocí modulu **DImp**. Modul **ImpIn** musí být umístěn v rychlém procesu (proces typu Quick nebo Hi_x).

Každý ze šestnácti signálů může mít aktivní vzestupnou nebo sestupnou hranu případně obě hrany. U netvarovaných signálů bývá většinou aktivní pouze vzestupná hrana.

3.1.2 Modul DImp

Modul **DImp** čte údaj z impulsního vstupu a přepočte ho na fyzikální rozměr. Modul spolupracuje s modulem pro obsluhu impulsních vstupů **ImpIn**. Modul **DImp** se umísťuje v kterémkoli řádném procesu (Proc00 až Proc15). Při načítání impulsů na více vstupech se modul **DImp** do aplikace vloží vícekrát.

Modul **DImp** zpracovává údaje z čidel s impulsním výstupem typu „1 impuls = N fyzikálních jednotek“. Na základě této hodnoty N (konstanty čidla) odvozuje stav průběžného počítadla (např. počítadlo vodoměru, plynoměru, elektroměru atd.) a odhaduje okamžitou hodnotu.

Pozor

*Při použití jednoho modulu **ImpIn** nelze přistupovat ke stejnému signálu pomocí více modulů **DImp**!*

3.2 Příklad č. 1 – programový čítač

K řídicímu systému bude na vstup DI0.0 připojeno zařízení, které vysílá impulsy o délce 100 ms. Minimální mezera mezi impulsy je 100 ms. Fyzikální význam jednoho impulsu je 0,25 MJ. Hodnoty čítače budou načítány každých 5 s. Při dosažení hodnoty 1 GJ se bude čítač nulovat.

Modul **ImpIn** je nutné umístit do takového rychlého procesu, aby jeho perioda byla menší nebo rovna polovině délky impulsu. Pro uvedené zadání je to proces s periodou 50 ms.

Požadovanou funkci plní následující kód.

Proces Quick, perioda 50 ms

```
// obsluha 16-tice impulsních vstupů kanálu 0
:10000 ImpIn #0, 100, 30, 0xFFFF, 0x0000
```

Řádný proces, perioda 5 s

```
// načtení a přepočítání impulsního vstupu DI0.0
DImp :10000, 0, Delta, Suma, Okamzita, Konstanta, SyncIn.0, NONE.0
// při dosažení hodnoty 1 milion vynulovat čítač
Let SyncIn = if(Suma >= 1000000, 1, 0)
```

Proměnná *Suma* je průběžným počítadlem, proměnná *Delta* je přírůstek počítadla od minula, proměnná *Okamzita* obsahuje odhad okamžité hodnoty, v proměnné *Konstanta* je konstanta měřidla (0,25 MJ/impuls).

Výše uvedený příklad je součástí přílohy ap0017_cz_03.zip pod názvem citacovevstupy_p1_cz_xx.dso. Aplikace je vytvořená pro řídicí systém **AMiNi4DW2**. Lze ji však změnit pro jakýkoliv jiný řídicí systém, osazený číslicovými vstupy, pomocí volby „Nástroje/Změnit typ stanice“.

4 Programový čítač INT

Některé číslicové vstupy mohou generovat hardwarové přerušení (interrupt). Programovou obsluhou lze pomocí Interrupt_x procesů tyto vstupy použít pro načítání impulsů.

Omezení kmitočtu vstupního signálu je dáno programem. Typicky lze použít do kmitočtu 10 kHz.

4.1 Interrupt procesy

V aplikaci lze vytvořit až 16 interrupt procesů (ve vývojovém prostředí DetStudio označované jako Interrupt_0 až Interrupt_15), které se vyvolávají na základě přerušení od vnější události. Touto vnější událostí se rozumí změna hladiny číslicového signálu přivedeného na vstup řídicího systému. Spuštění procesu může být odvozeno od náběžné hrany signálu nebo od sestupné hrany signálu, případně od obou hran signálu. Počet možných interrupt procesů a jejich navázání na příslušné vstupy je závislé na použitém typu řídicího systému, jak ukazuje následující tabulka.

Proces	ADiS (AD-CPUW2 + 2× AD-FDI8)	AMiNi-ES
Interrupt_0	FDI0.0	DI0.0
Interrupt_1	FDI0.1	DI0.1
Interrupt_2	FDI0.2	DI0.2
Interrupt_3	FDI0.3	DI0.3
Interrupt_4	FDI0.4	DI0.4
Interrupt_5	FDI0.5	DI0.5
Interrupt_6	FDI0.6	DI0.6
Interrupt_7	FDI0.7	DI0.7
Interrupt_8	FDI1.0	–
Interrupt_9	FDI1.1	–
Interrupt_10	FDI1.2	–
Interrupt_11	FDI1.3	–
Interrupt_12	FDI1.4	–
Interrupt_13	FDI1.5	–
Interrupt_14	FDI1.6	–
Interrupt_15	FDI1.7	–

Poznámka

U řídicího systému **ADiS** jsou procesy Interrupt_0 až Interrupt_7 navázány na první modul **AD-FDI8** v sestavě (**AD-FDI8** nejbližší CPU modulu). Procesy Interrupt_8 až Interrupt_15 jsou navázány na druhý modul **AD-FDI8** v sestavě. Na přiřazení logických kanálů těchto modulů nezáleží.

Pozor

Pokud je v sestavě řídicího systému **ADiS** osazen jeden modul **AD-AO8x**, lze použít pouze 8 interrupt procesů. Pokud jsou moduly **AD-AO8x** osazeny dva, nelze využít žádný interrupt proces.

4.2 Programová obsluha

Interrupt_x proces lze jednoduše využít pro čítání impulsů tak, že se impulsy budou přivádět na vstup řídicího systému, od kterého se vyvolává přerušení (tedy i příslušný proces Interrupt_x, viz tabulka výše). V tomto procesu se pak bude provádět čítání příchozích impulsů.

4.2.1 Měření otáček

Pro měření otáček za minutu pomocí impulsního čidla lze využít modul **RPM1**. Modul měří otáčky na jednom číslicovém vstupu (který vyvolává přerušení). Při měření otáček na více vstupech se modul **RPM1** do aplikace vloží vícekrát. Modul se umísťuje do Interrupt_x procesu příslušného k vstupu, na kterém chceme měřit otáčky. Jazyk procesu musí být typu LA nebo RS.

V parametrech příslušného procesu Interrupt_x se nastaví, zda se má proces (a tím i modul **RPM1**) spouštět při náběžné nebo sestupné hraně signálu. Až na výjimečné případy se důrazně nedoporučuje nastavovat spouštění Interrupt_x procesu na obě hrany – zpravidla by to mělo neblahý vliv na přesnost měření. Do Interrupt_x procesu s modulem **RPM1** se nedoporučuje vkládat jakékoliv další funkční moduly.

Bez ohledu na četnost impulsů je obsah výstupní proměnné aktualizován každých 10 ms, výstupní proměnná může být čtena v jakémkoliv procesu.

Celkový počet impulsů za jednotku času pro všechny moduly **RPM1** použité v aplikaci nesmí dohromady přesáhnout 80000 impulsů za minutu. Je však třeba dále uvážit, že počínaje hranicí 40000 impulsů za minutu může způsobená spotřeba času procesoru způsobit chyby v sériové komunikaci. Výše uvedená čísla se mohou lišit v závislosti na typu řídicího systému.

4.3 Příklad č. 2 – programový čítač INT, měření otáček

K řídicímu systému bude připojeno na vstup DI0.0 zařízení, které vysílá impulsy. Fyzikální význam jednoho impulsu je 1 m³. Při dosažení hodnoty 1 mil. m³ se bude čítač nulovat.

Dále bude na vstup DI0.1 přiveden signál z impulsního čidla pro měření otáček. Čidlo generuje v průběhu každé otáčky 8 impulsů.

Požadovanou funkci plní následující kód.

Proces Interrupt_0

```
// čítání impulsů přivedených na vstup DI0.0, aktivní je vzestupná hrana
IncDec cit_hodnota, 1, 1.000
```

Celkový počet načtených impulsů (resp. hodnota v m³) je v proměnné `cit_hodnota`.

Proces Interrupt_1

```
// měření otáček z impulsů přivedených na vstup DI0.1, aktivní je vzestupná hrana
RPM1 0x0808, 10.000, Otacky
```

Pro eliminaci vibrací a nepřesností v umístění osmi značek na měřicím kotouči se používá klouzavý průměr osmi posledních délek impulsů. Měří se otáčky od hodnoty 10 ot./min. Naměřené otáčky se ukládají do proměnné `Otacky`.

Řádný proces, perioda 1 s

```
// vyhodnocení načtené hodnoty čítače, případné nulování
Let cit_hodnota = if (cit_hodnota > 1000000, 0, cit_hodnota)
```

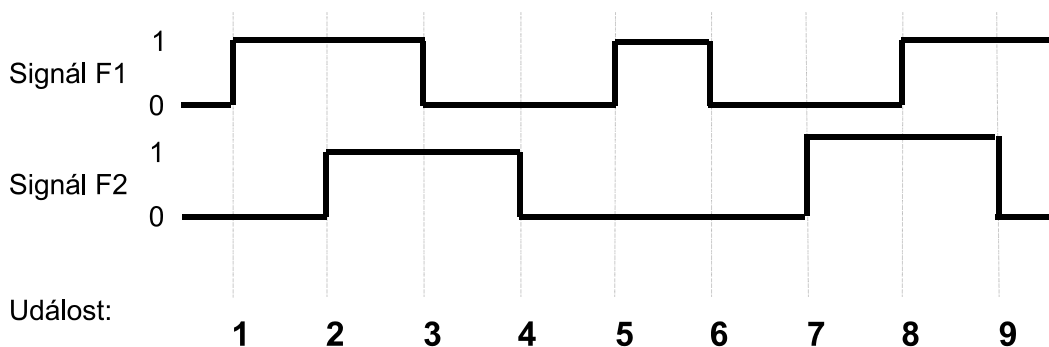
Výše uvedený příklad je součástí přílohy `ap0017_cz_03.zip` pod názvem `citacovevstupy_p2_cz_xx.dso`. Aplikace je vytvořena pro řídicí systém **AMiNi-ES**.

5 Hardwarový čítač

Hardwarovými čítačovými vstupy je vybaven řídicí systém **AMiNi-ES**. Řídicí systém je osazen třemi čítačovými vstupy (očíslovanými 0, 1 a 2). Napěťové úrovně signálu musí být 24 V ss. Maximální možná frekvence čítání je 160 kHz. Jedná se o rychlé (hardwarově podporované) čítačové vstupy, resp. vstupy pro inkrementální čidla polohy. Pro každý čítačový vstup lze zvolit, pro jaký konkrétní účel má být používán, popřípadě jaký typ čidla je k němu připojen.

5.1 Režimy vstupu

Čítačový vstup v jednotlivých režimech reaguje odlišně na různé kombinace vstupních signálů (F1 a F2) a hran.



Obr. 1 – Definice událostí

V následující tabulce znamená „+“ zvýšení hodnoty čítače, „-“ znamená snížení hodnoty čítače, prázdné políčko znamená, že čítač na událost nereaguje.

Reakce na události

Režim vstupu	Volba aktivní hrany	Reakce na událost:								
		1	2	3	4	5	6	7	8	9
IRC F1-F2	Nebere se v úvahu	+	+	+	+	+	-	-	-	-
IRC F2-F1	Nebere se v úvahu	-	-	-	-	-	+	+	+	+
Směr +/-	Náběžná		-					+		
	Sestupná				+					-
	Obě hrany		-		+			+		-
Směr -/+	Náběžná		+					-		
	Sestupná				-					+
	Obě hrany		+		-			-		+
Nahoru	Náběžná		+					+		
	Sestupná				+					+
	Obě hrany		+		+			+		+
Dolů	Náběžná		-					-		
	Sestupná				-					-
	Obě hrany		-		-			-		-

5.2 Vlastnosti čítačových vstupů

Parametry a režimy čítačových vstupů u řídicího systému **AMiNi-ES**.

Vstup „0“		
Číslo vstupu	0	
Signál F1	DI0.1	
Signál F2	DI0.0	
Rozsah interního čítače	32 bitů (-2.147.483.648 až 2.147.483.647)	
Vstupy pro poziční (indexové) značky v automatickém režimu	Nejsou. Použijte libovolný volný číslicový vstup pro poloautomatický režim obsluhy pozičních značek.	
Podporované kombinace režimu a volby aktivních hran:		
Režim	Hrany	Poznámka
IRC F1-F2	Nebere se v úvahu	Jedné úplné periodě vstupních signálů odpovídají 4 jednotkové změny interního čítače. (Čítač se mění při každé hraně kteréhokoliv vstupního signálu.)
IRC F2-F1	Nebere se v úvahu	
Směr +/-	Náběžná Sestupná Obě hrany	
Směr -/+	Náběžná Sestupná Obě hrany	
Nahoru	Náběžná Sestupná Obě hrany	Vstupní signál F1 nemá vliv na činnost vstupu, odpovídající číslicový vstup může být použit obvyklým způsobem.
Dolů	Náběžná Sestupná Obě hrany	

Vstup „1“		
Číslo vstupu	1	
Signál F1	DI0.3	
Signál F2	DI0.2	
Rozsah interního čítače	32 bitů (-2.147.483.648 až 2.147.483.647)	
Vstupy pro poziční (indexové) značky v automatickém režimu	Nejsou. Použijte libovolný volný číslicový vstup pro poloautomatický režim obsluhy pozičních značek.	
Podporované kombinace režimu a volby aktivních hran:		
Režim	Hrany	Poznámka
Stejně jako pro vstup „0“.		

Vstup „2“		
Číslo vstupu	2	
Signál F1	DI0.5	
Signál F2	DI0.4	
Rozsah interního čítače	32 bitů (-2.147.483.648 až 2.147.483.647)	
Vstupy pro poziční (indexové) značky v automatickém režimu	Nejsou. Použijte libovolný volný číslicový vstup pro poloautomatický režim obsluhy pozičních značek.	
Podporované kombinace režimu a volby aktivních hran:		
Režim	Hrany	Poznámka
Stejně jako pro vstup „0“.		

Poznámka

Bez ohledu na to, jestli se použijí funkční moduly pro využití vstupů DI0.0 až DI0.5 jako čítačových vstupů/vstupů pro inkrementální čidla polohy, jsou signály z těchto vstupů přístupné v odpovídajících logických kanálech DI0 a DI0AC (čísla 0 a 1). Nic tedy nebrání využití těchto signálů jako obyčejných číslicových vstupů obvyklým způsobem.

5.3 Programová obsluha

Pro obsluhu čítačových vstupů, resp. vstupů pro inkrementální čidla polohy slouží moduly:

- ◆ **IRCM_ode** – Nastavení režimu a konstant hardwarového čítačového vstupu.
- ◆ **IRCP_reset** – Nastavení výchozí hodnoty (poziční značky) hardwarového čítačového vstupu.
- ◆ **IRC_set** – Nastavení hodnoty hardwarového čítačového vstupu.
- ◆ **IRC_In** – Čtení hodnoty z hardwarového čítačového vstupu.

Podrobný popis modulů lze nalézt v nápovědě k části PseDet vývojového prostředí DetStudio.

Čítačové vstupy, resp. vstupy pro inkrementální čidla polohy jsou definovány dvěma vstupními signály F1 a F2, případně vstupy pro poziční značky, viz kapitola 5.2 „Vlastnosti čítačových vstupů“.

5.4 Příklad č. 3 – hardwarový čítač

K řídicímu systému bude připojeno zařízení, které vysílá impulsy (pouze 1 signál). Hodnoty z čítače budou načítány každých 10 s. Zároveň bude vyžadováno počítat rozdíl počtu impulsů od předešlého načtení. Při dosažení hodnoty čítače 5 milionů nebo po restartu řídicího systému bude čítač nulován.

Parametry čítače

Zvolený řídicí systém disponuje třemi hardwarově podporovanými čítačovými vstupy. Pro čítání impulsů z jednoho připojeného signálu bude pro náš účel nejvhodnější režim „nahoru“, kdy s každou aktivní hranou (náběžná/ sestupná/ obě hrany) na vstupu F2 dochází ke zvětšování hodnoty čítače. Vstupní signál F1 nemá v tomto režimu na činnost čítače vliv. Uvedený režim je možné použít u obou čítačových vstupů. Použijeme čítač „0“, kdy signál ze zařízení bude přiveden na vstup F2, tj. vstup DI0.0.

Vlastní program je tvořen následujícím kódem.

Proces INIT

V procesu INIT je nastaven čítač „0“ do režimu „nahoru“. Čítačový vstup bude reagovat na náběžnou hranu. S každým příchozím impulsem se zvětší o 1. Pokud nastavení režimu čítače proběhne úspěšně, je nastaven 0. bit proměnné **nastaveno** na hodnotu 1. Dále je v tomto procesu nastavení počáteční hodnoty čítače na 0.

```
// inicializace režimu čítačového vstupu
IRCMoDe 0, 4, 1, 0.000, 1.000, nastaveno.0
// vynulování čítače
IRCSeT 0, 0.000
Let cit_old = 0
```

Řádný proces, perioda 10 s

V tomto procesu se načítá hodnota z čítače. Z této hodnoty se pak počítá rozdíl oproti předchozí hodnotě čítače. V případě překročení stanovené hodnoty je čítač vynulován.

```
// načtení hodnoty čítače
IRCIn 0, cit_hodnota
// vyhodnocení načtené hodnoty
Let ok = if ((cit_old < cit_hodnota) AND (cit_hodnota < 5000000), 1, 0)
If ok.0
    // výpočet rozdílu
    Let cit_delta = cit_hodnota - cit_old
    Let cit_old = cit_hodnota
Else
    // vynulování čítače
    IRCSeT 0, 0.000
    Let cit_old = 0
EndIf
```

Výše uvedený příklad je součástí přílohy ap0017_cz_03.zip pod názvem citacovevstupy_p3_cz_xx.dso. Projekt je vytvořen pro řídicí systém **AMiNi-ES**.

6 Čítání impulsů pomocí rozšiřujících modulů

Pokud je řídicí systém vybaven komunikační linkou RS485 nebo RS232, lze pro čítání impulsů využít rozšiřující moduly. Rozšiřující moduly **DM-DI24** nebo **DMM-DI24** umožňují využívat funkci čítání příchozích impulsů na kterémkoliv z jejich vstupů. Tato funkce alespoň částečně řeší problémy se zachycováním krátkých impulsů. Při použití této funkce je ovšem nutno brát v potaz následující vlastnosti:

- ♦ Maximální možná načítaná hodnota je 16383 (číslo 14 bit) pro modul **DM-DI24** a pro modul **DMM-DI24** je maximální možná načítaná hodnota 32767 (číslo 15 bit). Po přičtení dalšího impulsu se začíná čítat opět od nuly.
- ♦ Maximální možná frekvence příchozích impulsů může být 25 Hz. Při vyšší frekvenci není možné zaručit, že všechny příchozí impulsy budou zaznamenány.
- ♦ Interní čítač modulu je nulován při odpojení napájecího napětí, případně je možno jej nulovat programově.
- ♦ Programově je nutno ošetřit přetečení interního čítače.

6.1 Rozšiřující modul DM-DI24

Komunikace s rozšiřujícím modulem **DM-DI24** se děje pomocí protokolu ARION. Více informací naleznete v aplikační poznámce *AP0025 – Komunikace v síti ARION – definice tabulkou*.

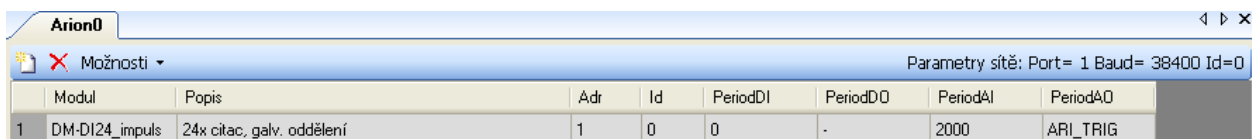
Poznámka

*Funkce čítání příchozích impulsů na kterémkoli ze vstupů rozšiřujícího modulu **DM-DI24** je k dispozici od firmware verze 1.65.*

6.1.1 Příklad č. 4 – čítání impulsů na DM-DI24

Příklad realizace komunikace protokolem ARION (38400 bps, port 1) s **DM-DI24** (adresa 1).

Síť ARION s rozšiřujícím modulem **DM-DI24** je definována dle následujícího obrázku.



Modul	Popis	Adr	Id	PeriodDI	PeriodDO	PeriodAI	PeriodAO
1	DM-DI24_impuls 24x citac, galv. oddeleni	1	0	0	-	2000	ARI_TRIG

Obr. 2 Definice sítě ARION s rozšiřujícím modulem **DM-DI24**

Pro použití modulu **DM-DI24** v režimu čítání impulsů, je nutné v tabulce Arion0 nadefinovat modul s názvem **DM-DI24_impuls**. Hodnota parametru **PeriodDI** se nastaví na 0 a parametr **PeriodAI** se nastaví na požadovanou hodnotu. Modul lze současně použít pro načítání stavu číslicových vstupů. V tomto případě se parametry **PeriodDI** a **PeriodAI** nastaví na požadované hodnoty. Poté je možné modulem **ARI_DigIn** číst aktuální stav vstupů **DI** rozšiřujícího modulu **DM-DI24** a modulem **ARI_AnIn** číst stav čítačů ze vstupů **DI**.

Načtení hodnot čítačů ze všech vstupů

```
ARI_AnIn 1, 0, 24, DataCit_all[0,0], NONE[0,0], 16384.0, 0.0, 16383.0, 0.0, 16383.0
```

Poznámka

*U čítačů je nutno ošetřit jejich přetečení! Rozsah čítače je 0 až 16383. Čítač tedy vytváří řadu 0, 1, 2, ..., 16382, 16383, 0, 1, 2, ... Bude tedy třeba čítat počet přetečení „p“ čítače a výsledná suma impulsů pak bude rovna: **suma = p × 16384 + čítač**.*

Nastavení hodnoty všech čítačů

```
ARI_AnOut 1, 0, 24, NastavCit[0,0], NONE[0,0], 16384.0, 0.0, 16383.0, 0.0, 16383.0
// nastavení hodnot čítačů
If Nastav.0
    // vyvolání vlastního přenosu do DM-DI24
    ARI_Trig 1, 1
    // vynulování příznaku pro nastavení
    Let Nastav = 0
EndIf
```

Po nastavení hodnot do vyrovnávací paměti pomocí modulu **ARI_AnOut** je nutné vyvolat vlastní přenos modulem **ARI_Trig**, protože se hodnoty do uzlu typu AO u **DM-DI24** nepřenáší periodicky.

Pozor!

Ve firmware verze 1.65 se vyskytl problém se zápisem požadované hodnoty do jednotlivých čítačů. Hodnota zapisovaná do AO0 se zapisuje do AO7, hodnota zapisovaná do AO1 se zapisuje do AO6, atd. Tento problém se vyskytuje vždy v osmici, tzn., že hodnota zapisovaná na AO8 se zapisuje do AO15, hodnota zapisovaná na AO16 se zapisuje na AO23, atd. Problém řeší firmware verze 1.66. Pokud tedy provádíte výměnu modulu DM-DI24 je nutná dle verze firmware v DM-DI24 také úprava SW v řídicím systému!

Výše uvedený příklad je součástí přílohy ap0017_cz_03.zip pod názvem citacovevstupy_p4_cz_xx.dso. Projekt je vytvořen pro řídicí systém **AMiNi4DW2**. Lze jej však změnit pro jakýkoliv jiný řídicí systém, osazený sériovou komunikační linkou, pomocí menu DetStudia „Nástroje/Změnit typ stanice“.

6.2 Rozšiřující modul DMM-DI24

Komunikace s rozšiřujícím modulem **DMM-DI24** se děje pomocí protokolu MODBUS RTU. Více informací naleznete v aplikační poznámce AP0008 – *Komunikace v síti MODBUS RTU (PseDet)*.

6.2.1 Příklad č. 5 – čítání impulsů na DMM-DI24

Příklad realizace komunikace protokolem MODBUS RTU (38400 bps, sudá parita, jeden stop bit, port 1) s **DMM-DI24** (adresa 1).

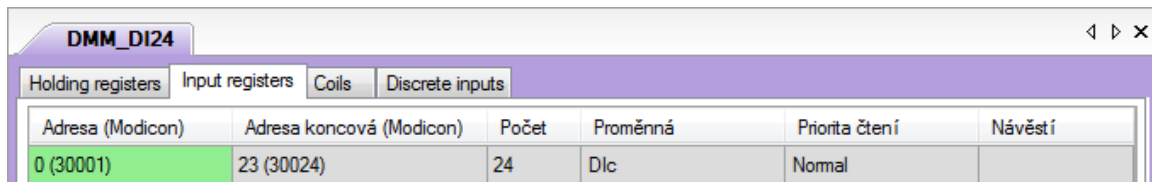
Pozice vzdálených bodů (registrů/binárů) v jednotlivých modulech je vždy dána číslem daného vstupu/výstupu modulu **DMM-DI24**.

Pro načtení/zápis hodnot čítače z/do modulu **DMM-DI24** lze použít následující funkce MODBUS.

Funkce	Využití u DMM-DI24
3	Čtení hodnot čítačů.
4	Čtení hodnot čítačů (stejně jako funkce 3).
6	Zápis hodnoty jednoho čítače.
16	Zápis hodnot všech čítačů.

Čtení hodnoty čítačů na všech vstupech

Modul **DMM-DI24** umožňuje čtení stavu čítačů jak pomocí vstupních registrů, tak pomocí uchovávacích registrů. V rámci aplikační poznámky budou pro čtení využity vstupní registry dle následujícího obrázku.



Adresa (Modicon)	Adresa koncová (Modicon)	Počet	Proměnná	Priorita čtení	Návěstí
0 (30001)	23 (30024)	24	Dlc	Normal	

Obr. 3 – Definice čtení hodnot všech čítačů

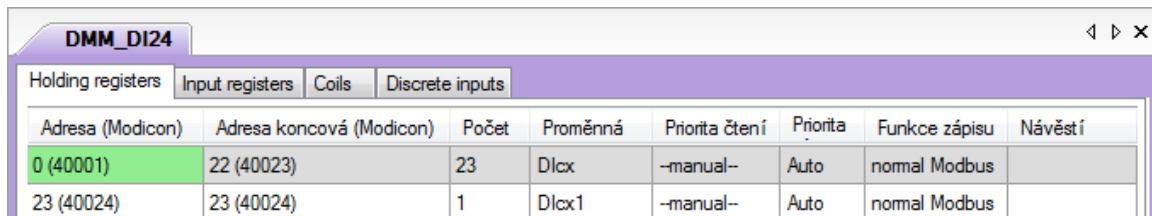
Proměnná **Dlc** je matice typu MI o velikosti [1, 24], tato konstrukce zabezpečí čtení hodnot všech čítačů rozšiřujícího modulu s periodou 1 s (priorita „Normal“).

Poznámka

*U čítačů je nutno ošetřit jejich přetečení! Rozsah čítače je 0 až 32767. Čítač tedy vytváří řadu 0, 1, 2, ..., 32766, 32767, 0, 1, 2, ... Bude tedy třeba čítat počet přetečení „p“ čítače a výsledná suma impulsů pak bude rovna: **suma = p × 32768 + čítač**.*

Nastavení hodnoty čítačů

Nastavení požadovaných hodnot čítačů lze nadefinovat prostřednictvím uchovávacích registrů dle následujícího obrázku.



Adresa (Modicon)	Adresa koncová (Modicon)	Počet	Proměnná	Priorita čtení	Priorita	Funkce zápisu	Návěstí
0 (40001)	22 (40023)	23	Dlcx	-manual-	Auto	normal Modbus	
23 (40024)	23 (40024)	1	Dlcx1	-manual-	Auto	normal Modbus	

Obr. 4 – Definice nastavení hodnoty více čítačům a jednomu čítači

Proměnná **Dlcx** je matice MI o velikosti [1, 23]. Při zápisu do kterékoliv buňky matice dojde k vyvolání zápisu všech hodnot v matici do čítačů na vstupech DI0 až DI22. **Zápis je proveden i v případě, kdy je do buňky matice zapsána stejná hodnota.**

Proměnná **Dlcx1** je skalární proměnná typu Integer. Při zápisu do proměnné dojde k vyvolání zápisu hodnoty proměnné do čítače na vstupu DI23. **Zápis je proveden i v případě, kdy je do proměnné zapsána stejná hodnota.**

Výše uvedený příklad je součástí přílohy ap0017_cz_03.zip pod názvem citacovestupy_p5_cz_xx.dso. Projekt je vytvořen pro řídicí systém **AMiNi4DW2**. Lze jej však změnit pro jakýkoliv jiný řídicí systém, osazený sériovou komunikační linkou, pomocí menu DetStudia „Nástroje/Změnit typ stanice“.

7 Technická podpora

Veškeré informace ohledně čítačových vstupů, Vám poskytne oddělení technické podpory firmy AMiT. Technickou podporu můžete kontaktovat nejlépe prostřednictvím emailu na adrese **support@amit.cz**.

8 Upozornění

AMiT, spol. s r.o. poskytuje informace v tomto dokumentu, tak jak jsou, nepřijímá žádné záruky, pokud se týče obsahu tohoto dokumentu a vyhrazuje si právo měnit obsah dokumentu bez závazku tyto změny oznámit jakékoli osobě či organizaci.

Tento dokument může být kopírován a rozšiřován za následujících podmínek:

1. Celý text musí být kopírován bez úprav a se zahrnutím všech stránek.
2. Všechny kopie musí obsahovat označení autorského práva společnosti AMiT, spol. s r.o. a veškerá další upozornění v dokumentu uvedená.
3. Tento dokument nesmí být distribuován za účelem dosažení zisku.

V publikaci použité názvy produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.