



**MIDAM** Control System



**MIDAM**

# Uživatelský manuál

## k modulům řady **MIDAM**



**MIDAM** Control System



**MIDAM**

Tento dokument je chráněn autorskými právy. Všechna práva jsou vyhrazena. Výrobce si vyhrazuje právo upravovat moduly řady **MIDAM**, které jsou popsány v této uživatelské příručce, bez jakéhokoli upozornění při zachování stávající funkce modulů.

únor 2004  
Revize 2.4

**MIKRO**  **KLIMA**

Veverkova 1343  
500 02 Hradec Králové  
Tel.: 495 500 970, fax: 495 500 979  
e-mail: [info@midam.cz](mailto:info@midam.cz)

*Název ADAM je registrovaná obchodní známka firmy Advantech®.  
© 1997 – 2004 Mikroklima s.r.o*



## 1. Obsah

1. OBSAH.....	3
<b>2. MIDAM 180</b> .....	<b>4</b>
2.1. POPIS MODULU.....	5
2.2. TECHNICKÁ SPECIFIKACE MODULU.....	5
2.3. ZAPOJENÍ SVOREK MODULU <b>MIDAM 180</b> .....	6
2.4. PŘEPOČET HEXADECIMÁLNÍCH HODNOT NA TEPLOTU A RELATIVNÍ VLHKOST.....	6
2.5. PŘIPOJENÍ ZAKONČOVACÍHO ODPORU.....	6
2.6. POPIS ŘÍDÍCÍCH PŘÍKAZŮ MODULU.....	8
2.6.1. <i>Konfigurace</i> .....	9
2.6.2. <i>Přečtení analogové vstupní hodnoty</i> .....	11
2.6.3. <i>Span kalibrace (kalibrace plného rozsahu)</i> .....	13
2.6.4. <i>Offset kalibrace</i> .....	14
2.6.5. <i>Vyčtení konfigurace modulu</i> .....	15
2.6.6. <i>Synchronizovaný odměr</i> .....	16
2.6.7. <i>Přečtení data změřených při synchronizovaném odměru</i> .....	17
2.6.8. <i>Přečtení verze firmware</i> .....	18
2.6.9. <i>Přečtení jména modulu</i> .....	19
2.7. UVEDENÍ MODULU DO REŽIMU INIT.....	20



**MIDAM** Control System

**MIDAM**

## **2. MIDAM 180**

čidlo teploty a relativní vlhkosti s digitálním výstupem



## 2.1. Popis modulu

Modul **MIDAM 180** je inteligentní čidlo teploty a relativní vlhkosti řízené mikroprocesorem. Čidlo se sestává z vlastního měřicího elementu a elektroniky umístěné v hlavici čidla. Rozsah měřených hodnot je u teploty  $-40^{\circ}\text{C}$  až  $123,8^{\circ}\text{C}$  a u relativní vlhkosti 0 až 100%. Modul je vyráběn v provedení do interiéru a v kanálovém provedení.

Modul komunikuje a je ovládán výhradně prostřednictvím datové sběrnice RS 485. Komunikační protokol je shodný s moduly ADAM řady 4000 firmy ADVANTECH. Čidlo MIDAM 180 se chová jako **dva** moduly ADAM 4013 (moduly jsou umístěny na dvou po sobě následujících adresách, přičemž adresací modulu na nižší adrese se automaticky předadresuje i čidlo na vyšší adrese) tzn., že pro ovládání je možné použít standardní ovladače pro moduly ADAM v různých řídicích programech. Modul na **nižší adrese udává hodnotu teploty**. Modul na **vyšší adrese udává hodnotu relativní vlhkosti**.

Protože některé komunikační kabely obsahují více párů vedení v jednom kabelu, bylo čidlo konstruováno tak, aby mohlo být napájení k modulu přivedeno dalšími volnými vodiči v kabelu.

Základem analogového modulu je senzor teploty a relativní vlhkosti a A/D převodník řízený procesorem. Komunikační vstupy jsou chráněny proti přepětí. V případě, že modul je na vedení umístěn jako koncový, je možné zkratováním kontaktů (BUS END vedle svorek pro připojení komunikačních vodičů) připojit k vedení zakončovací odpor.

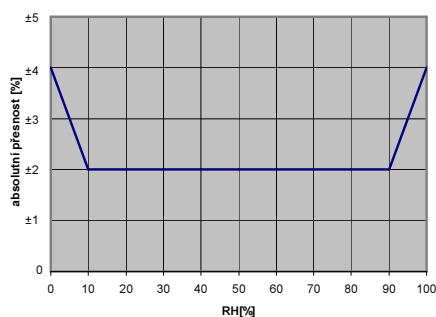
Pro snadnější identifikaci možných závad při zprovoznování je uvnitř modulu indikační LED. Dioda blikne vždy, když čidlo komunikuje.

Veškerá nastavení jsou uložena v paměti EEPROM. Modul je vybaven obvodem WATCHDOG, který dohlíží na správný běh programu v procesoru.

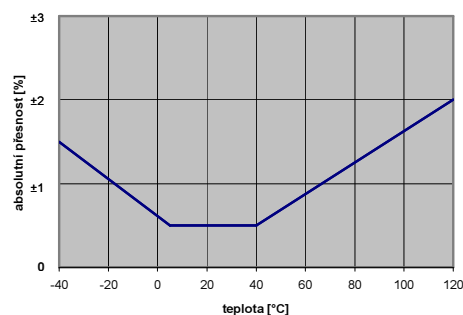
## 2.2. Technická specifikace modulu

Napájecí napětí	10 V ÷ 35 V stejnosměrné nestabilizované 14 V ÷ 24 V střídavé
Příkon	300 mW
Přípustná pracovní teplota v hlavici čidla	0 ÷ 70°C
Měřicí rozsah	teplota $-40^{\circ}\text{C}$ až $123,8^{\circ}\text{C}$ relativní vlhkost 0% až 100%
Komunikační vlastnosti	komunikace po sběrnici RS 485 přenos. rychlosti 1200, 2400, 4800, 9600, 19200 Bd max. délka segmentu 1200 m, asynchronní přenos 256 modulů na jeden sériový port komunikační protokol shodný s moduly ADAM 4000
Přesnost měřených veličin:	teplota, vlhkost viz níže
Rozlišení	teplota 0,01°C (14 bitů) vlhkost 0,03% (12 bitů)

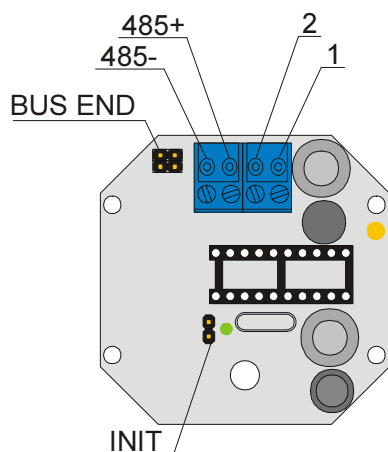
Závislost absolutní přesnosti relativní vlhkosti



Závislost absolutní přesnosti teploty



### 2.3. Zapojení svorek modulu MIDAM 180



Označení	Popis
1	napájení
2	napájení
485+	+ datové sběrnice RS 485
485-	- datové sběrnice RS 485
INIT	uvedení do režimu init
BUS END	pro připojení zakončení sběrnice

Obrázek 2.3-1: Schéma zapojení svorek modulu **MIDAM 180**

### 2.4. Přepoččet hexadecimálních hodnot na teplotu a relativní vlhkost

Jestliže je modul nastaven na formát dat typu HEX,

**teplota** se vypočte dle následující rovnice:

$$t[^\circ\text{C}] = 0,01 * N - 40$$

N udává decimální reprezentaci naměřené hodnoty

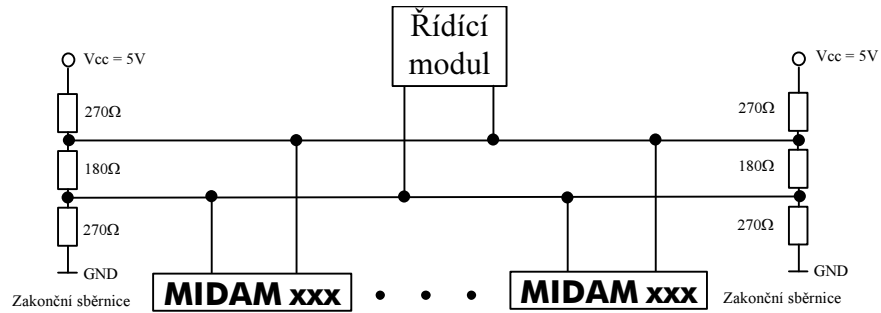
**relativní vlhkost** se vypočte dle následující rovnice:

$$\text{RH}[\%] = (t[^\circ\text{C}] - 25) * (0,01 + 0,00008 * N) - 4 + 0,0405 * N - 2,8 * 10^{(-6)} * N^2$$

N udává decimální reprezentaci naměřené hodnoty

### 2.5. Připojení zakončovacího odporu

Je-li modul připojen na konci komunikační sběrnice RS485, je nutné na sběrnici v tomto místě připojit zakončovací odpor, viz obrázek 2.5-1. Moduly řady **MIDAM** umožňují připojit zakončení sběrnice propojením zkratovacích svorek („jumperů“). Po odkrytí hlavice modulu uživatel získá snadný přístup k jumperům BUS END, viz obrázek 2.3-1. Zkratováním těchto jumperů se připojí na komunikační sběrnici zakončovací odpor.



Poznámka: Pozice řídicího modulu na sběrnici není pevně stanovena.

**Obrázek 2.5-1: Schéma připojení modulů na komunikační sběrnici RS485**

**POZOR:** Na komunikační sběrnici je nutno připojit právě dva zakončovací odpory a to na každém konci sběrnice jeden, viz obrázek 2.5-1.

2.6. Popis řídicích příkazů moduluTabulka 2.8-1 Řídicí příkazy modulu **MIDAM 180**

Syntaxe řídicího příkazu	Název příkazu	Popis příkazu
<b>%AANNTTCCFF(cr)</b>	Konfigurace	Nastaví adresu, vstupní rozsah, přenosovou rychlost, kontrolní součet ano / ne a integrační čas adresovaného modulu
<b>#AA(cr)</b>	Přečtení analogové vstupní hodnoty	Vrátí analogovou vstupní hodnotu adresovaného modulu ve zvoleném formátu dat
<b>\$AA0(cr)</b>	Span kalibrace (kalibrace plného rozsahu)	Kalibrace adresovaného modulu na plný rozsah (odstranění chyby zesílení)
<b>\$AA1(cr)</b>	Offset kalibrace	Kalibrace adresovaného modulu na chyby způsobené offsetem zesilovačů
<b>##*(cr)</b>	Synchronizovaný odměr	Všechny vstupní moduly přečtou hodnotu vstupů a uloží ji do speciálního registru pro pozdější vyčtení
<b>\$AA4(cr)</b>	Přečtení data změřených při synchronizovaném odměru	Vrátí hodnotu uloženou ve speciální registru modulu změřenou po přijetí příkazu <b>##*(cr)</b>
<b>\$AA2(cr)</b>	Vyčtení konfigurace modulu	Vrátí konfigurační parametry adresovaného modulu
<b>\$AAF(cr)</b>	Přečtení verze firmware	Vrátí verzi firmware adresovaného modulu
<b>\$AAM(cr)</b>	Přečtení jména modulu	Vrátí jméno adresovaného modulu

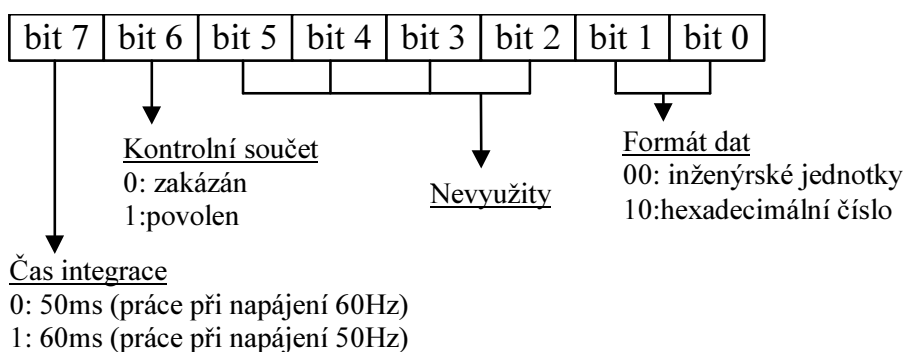


### 2.6.1. Konfigurace

Popis příkazu: Nastaví adresu, vstupní rozsah, přenosovou rychlost, kontrolní součet ano / ne a integrační čas adresovaného modulu.

Syntaxe: %AANNTTCCFF(cr)  
 % určující znak příkazu *konfigurace*.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 NN (rozsah 00 ÷ FF) reprezentuje novou hexadecimální adresu modulu, na kterou bude modul po úspěšném přijetí příkazu předadresován.  
 TT reprezentuje typ vstupního rozsahu modulu, viz tabulka 2.6-2.  
 CC reprezentuje přenosovou rychlost (baud rate), viz tabulka 2.6-3  
 FF hexadecimální číslo odpovídající 8-mi bitovému číslu, které reprezentuje formát dat, povolení / zakázání kontrolního součtu a integrační čas. Význam jednotlivých bitů ukazuje obrázek 2.6-1.  
 (cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Obrázek 2.6-1: Význam jednotlivých bitů parametru FF



Odpověď: !AA(cr) v případě úspěšně přijatého příkazu.  
 ?AA(cr) v případě chybně zadaného parametru příkazu nebo v případě, že se je měněna přenosová rychlost nebo stav kontrolního součtu aniž by byl modul v režimu INIT.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

! určující znak odpovědi na správě přijatý příkaz  
 ? určující znak odpovědi při chybně zadaném parametru konfiguračního příkazu.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 (cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Příklad: příkaz: %01100910(cr)  
odpověď: !10(cr)

Modul **MIDAM 180** na adrese 01hex je konfigurován na novou adresu 10hex, na vstupní rozsah  $-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$  a  $0\% \div 100\%$ , na přenosovou rychlost 9600bd, integrační čas 60ms (50Hz), na „inženýrský“ formát dat a na nezabezpečení přenosu kontrolním součtem.

Poznámka: Všechny parametry modulu mohou být měněny, je-li modul v režimu INIT. Komunikační rychlost a stav posílání kontrolního součtu nemohou být měněny není-li modul v režimu INIT, ostatní parametry modulu mohou být měněny i když modul není v režimu INIT. Uvedení do režimu INIT je popsáno v kapitole 2.7.

**Tabulka 2.6-2 Vstupní rozsahy modulu MIDAM 180**

Kód vstupního rozsahu TT (hex)	Vstupní rozsah
20	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
21	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
22	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
23	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
24	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
25	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
26	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
27	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
28	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$
29	$-40^{\circ}\text{C} \div 123,8^{\circ}\text{C}$ a $0\% \div 100\%$

**Tabulka 2.6-3 Přenosové rychlosti modulu MIDAM 180**

Kód přenosové rychlosti CC (hex)	Přenosová rychlost
03	1200bd
04	2400bd
05	4800bd
06	9600bd
07	19200bd

### 2.6.2. Přečtení analogové vstupní hodnoty

Popis příkazu: Vrábí vstupní hodnotu adresovaného modulu ve zvoleném formátu dat.

Syntaxe: #AA(cr)  
 # určující znak příkazu *přečtení analogové vstupní hodnoty z kanálu N*.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 (cr) carriage return je ukončující znak rámce, je roven (0Dhex).

Odpověď: >(data)(cr) v případě úspěšně přijatého příkazu.  
 ?AA(cr) v případě chybně zadaného čísla kanálu (je-li číslo kanálu mimo rozsah hodnot 0÷7).

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

> určující znak odpovědi na správně přijatý příkaz.  
 (data) reprezentují analogovou vstupní hodnotu. Formát dat je uveden v tabulce 2.6-4.  
 ? určující znak odpovědi při chybně zadaném parametru konfiguračního příkazu.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 (cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Příklad: příkaz: #11(cr)  
 odpověď: >+028.25(cr)

Modul **MIDAM 180** na adrese 11hex je dotazován na analogovou vstupní hodnotu.

Analogový vstupní modul odpovídá, že měří teplotu +28.25°C.

Jestliže je modul nastaven na formát dat typu HEX,

**teplota** se vypočte dle následující rovnice:

$$t[°C] = 0,01 * N - 40$$

N udává decimální reprezentaci naměřené hodnoty

**relativní vlhkost** se vypočte dle následující rovnice:

$$RH[\%] = (t[°C] - 25) * (0,01 + 0,00008 * N) - 4 + 0,0405 * N - 2,8 * 10^{(-6)} * N^2$$

N udává decimální reprezentaci naměřené hodnoty

Tabulka 2.6-4 Formát dat modulu MIDAM 180

Kód vstupního rozsahu TT (hex)	Hodnota vstupního rozsahu	Datový formát	Minimální hodnota	Maximální hodnota	Zobrazitelné rozlišení
20 ÷ 29	Čidlo teploty a relativní vlhkosti -40°C ÷ 123,8°C a 0% ÷ 100%	inženýrský	-40°C a 0%	123,8°C a 100%	0,01°C a 0,03%
		hexadecimální	0000 a 0000	3FFF a 0FFF	1 LSB

### 2.6.3. Span kalibrace (kalibrace plného rozsahu)

Popis příkazu: Kalibrace adresovaného modulu na plný rozsah (odstranění chyby zesílení).

Syntaxe: \$AA0(cr)  
\$ určující znak příkazu.  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
0 určuje, že se jedná o příkaz *span kalibrace*.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď: !AA(cr) v případě úspěšně přijatého příkazu.  
?AA(cr) v případě chybně zadaného příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

! určující znak odpovědi na správně přijatý příkaz  
? určující znak odpovědi při chybně zadaném příkaze.  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Poznámka: Modul **MIDAM 180** se **nekalibruje**. Tento příkaz je implementován pouze pro kompatibilitu s moduly řady **MIDAM**.

#### 2.6.4. Offset kalibrace

Popis příkazu: Kalibrace adresovaného modulu na chyby způsobené offsetem zesilovačů.

Syntaxe: \$AA1(cr)  
\$ určující znak příkazu.  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
1 určuje, že se jedná o příkaz *offset kalibrace*.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď: !AA(cr) v případě úspěšně přijatého příkazu.  
?AA(cr) v případě chybně zadaného příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

! určující znak odpovědi na správně přijatý příkaz  
? určující znak odpovědi při chybně zadaném příkaze.  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Poznámka: Modul **MIDAM 180** se **nekalibruje**. Tento příkaz je implementován pouze pro kompatibilitu s moduly řady **MIDAM**.

### 2.6.5. Vyčtení konfigurace modulu

Popis příkazu: Vrátí konfigurační parametry adresovaného modulu.

Syntaxe: \$AA2(cr)  
 \$ určující znak příkazu.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 2 určuje, že se jedná o příkaz *vyčtení konfigurace modulu*.  
 (cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď: !AATTCCFF(cr) v případě úspěšně přijatého příkazu.  
 ?AA(cr) v případě chybně zadaného příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

! určující znak odpovědi na správně přijatý příkaz  
 ? určující znak odpovědi při chybně zadaném příkaze.  
 AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
 TT reprezentuje typ vstupního rozsahu modulu, viz tabulka 2.6-2.  
 CC reprezentuje přenosovou rychlost (baud rate), viz tabulka 2.6-3.  
 FF je hexadecimální číslo odpovídající 8-mi bitovému číslu, které reprezentuje formát dat, povolení / zakázání kontrolního součtu a integrační čas. Význam jednotlivých bitů ukazuje obrázek 2.6-1.  
 (cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Příklad: příkaz: \$362(cr)  
 odpověď: !36200610(cr)

Modul **MIDAM 180** na adrese 36hex je dotazován na vlastní konfiguraci. Analogový vstupní modul na adrese 36hex je nastaven na vstupní rozsah -40°C ÷ 123,8°C a 0% ÷ 100%, na přenosovou rychlost 9600bd, na integrační čas 60ms, na „inženýrský“ formát dat a na nezabezpečení protokolu kontrolním součtem.

Poznámka: Viz též příkaz *konfigurace* v kapitole 2.6.1.

### 2.6.6. Synchronizovaný odměr

Popis příkazu: Všechny vstupní moduly přečtou hodnotu vstupů a uloží ji do speciálního registru pro pozdější vyčtení

Syntaxe: #\*\*(cr)  
# určující znak příkazu.  
\*\* určuje, že se jedná o *synchronizovaný odměr*.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex). Carriage return je v tomto příkaze nepovinným znakem příkazu.

Odpověď: Modul nepošle žádnou odpověď. Pro vyčtení hodnot provedených *synchronizovaným odměrem* je třeba použít příkaz \$AA4(cr)



**2.6.7. Přečtení data změřených při synchronizovaném odměru**

Popis příkazu: Vrátí hodnotu uloženou ve speciální registru modulu změřenou po přijetí příkazu #\*\*\*(cr).

Syntaxe: **\$AA4(cr)**  
**\$** určující znak příkazu.  
**AA** (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
**4** určuje, že se jedná o příkaz *přečtení data změřených při synchronizovaném odměru*.  
**(cr)** „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď: **>AA(status)(data)(cr)** v případě úspěšně přijatého příkazu.  
**?AA(cr)** v případě chybně zadaného příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

**>** určující znak odpovědi na správě přijatý příkaz  
**?** určující znak odpovědi při chybně zadaném příkaze.  
**AA** (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu  
**(status)** určuje zda jsou (data) čtena jako první po provedení synchronizovaného odměru (#\*\*). Status má hodnotu **1** jestliže jsou data čtena poprvé po provedení příkazu synchronizovaného odměru. (status) má hodnotu **0** jestliže data již byla z modulu alespoň jednou vyčtena po provedení synchronizovaného odměru.  
**(data)** reprezentují analogovou vstupní hodnotu. Formát dat je uveden v tabulce 2.6-4.  
**(cr)** „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Příklad: příkaz: **\$574(cr)**  
odpověď: **!571+029.56(cr)**

Modul na adrese 57hex je dotazován na data změřená při synchronizovaném odměru.

Analogový vstupní modul odpovídá s hodnotou (status)=1, což znamená, že tyto data jsou první přečtená data po provedení synchronizovaného odměru. Hodnota analogového vstupu při provedení synchronizovaného odměru byla +29,56°C.

### 2.6.8. Přečtení verze firmware

Popis příkazu: Vrátí verzi firmware adresovaného modulu.

Syntaxe: \$AAF(cr)  
\$ určující znak příkazu.  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
F určuje, že se jedná o příkaz *přečtení verze firmware*.  
(cr) „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď: !AA(verze)(cr) v případě úspěšně přijatého příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

! určující znak odpovědi na právě přijatý příkaz  
AA (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu  
(verze) určuje kód verze firmware modulu  
například: V1.3  
(cr) carriage return je ukončující znak rámce, je roven (0Dhex).

### 2.6.9. Přečtení jména modulu

Popis příkazu: Vrátí jméno adresovaného modulu.

Syntaxe:     **\$AAM(cr)**  
**\$**           určující znak příkazu.  
**AA**       (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu.  
**M**       určuje, že se jedná o příkaz *přečtení verze firmware*  
**(cr)**      „carriage return“ je ukončující znak rámce, je roven (0Dhex).

Odpověď:    **!AA(jméno)(cr)**      v případě úspěšně přijatého příkazu.

Jestliže modul detekuje špatnou syntaxi příkazu, komunikační chybu nebo se jeho adresa neshoduje s adresou v řídicím příkaze nepošle **žádnou odpověď**.

**!**           určující znak odpovědi na právě přijatý příkaz  
**AA**       (rozsah 00 ÷ FF) reprezentuje dvouznakovou hexadecimální adresu modulu  
**(jméno)**   určuje kód jména modulu  
            například: 4013  
**(cr)**      „carriage return“ je ukončující znak rámce, je roven 0Dhex.

## **2.7. Uvedení modulu do režimu INIT**

Modul se uvede do režimu INIT:

- a) Jestliže je po připojení napájení zkratován jumper INIT. Nemůžete-li modul připojený na sběrnici najít (má nastavenou jinou přenosovou rychlost, neznáte jeho adresu aj.) využijte tento způsob uvedení modulu do režimu init.
- b) Jestliže je po připojení napájení je na adrese 255dec = FFhex. Hlavní výhodou tohoto systému uvedení modulu do režimu INIT je, že modul může být uveden do režimu INIT i v případě, kdy k modulu není fyzicky přístup.

### **Postup při uvedení modulu do režimu INIT**

#### **ad a)**

1. Zkratujeme jumper INIT při odpojení napájení.
2. Připojíme napájení modulu, tím se modul uvedeme do režimu INIT. V tomto režimu se modul hlásí na adrese 00, komunikuje rychlostí 9600 bd a komunikace probíhá bez kontrolního součtu. V režimu INIT je možné nastavovat všechny parametry modulu, tj. i komunikační rychlost modulu a zabezpečení nebo nezabezpečení protokolu kontrolním součtem (tyto dva parametry nelze nastavit není-li modul v režimu INIT).
3. Provedeme žádoucí nastavení.
4. Odpojíme napájení modulu a vyndáme propojku INIT. Připojíme napájení a tím modul uvedeme zpět do normálního pracovního režimu (nelze nastavovat komunikační rychlost a zabezpečení nebo nezabezpečení protokolu kontrolním součtem).

#### **ad b)**

1. Modul předadresujeme na adresu 255dec = FFhex.
2. Odpojíme a opět připojíme napájení modulu, tím se modul uvedeme do režimu INIT. V tomto režimu se modul hlásí na adrese 00, komunikuje rychlostí 9600 bd a komunikace probíhá bez kontrolního součtu. V režimu INIT je možné nastavovat všechny parametry modulu, tj. i komunikační rychlost modulu a zabezpečení nebo nezabezpečení protokolu kontrolním součtem (tyto dva parametry nelze nastavit není-li modul v režimu INIT).
3. Modul nastavíme na adresu jinou než 255dec = FFhex.
4. Odpojíme a opět připojíme napájení modulu, tím se modul uvedeme zpět do normálního pracovního režimu (nelze nastavovat komunikační rychlost a zabezpečení nebo nezabezpečení protokolu kontrolním součtem).