

# **Modul ALA1**

## **Uživatelská příručka**

**Verze komunikačního protokolu 1.16 a 2.10**

**Rev. B**

Aleš Sekanina, Slovenská 152, 685 01 Bučovice, [ala1@ala1.com](mailto:ala1@ala1.com)

Aktualizováno: 18. listopadu 2008

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Verze komunikačního protokolu . . . . .	1
1.2	Komunikace s modulem . . . . .	1
1.3	Parametry modulu . . . . .	1
<b>2</b>	<b>Použitá označení v této příručce</b>	<b>2</b>
2.1	Označení příkazu a odpovědi . . . . .	2
2.2	Označení parametrů modulu . . . . .	2
2.3	Zvýraznění textu . . . . .	2
<b>3</b>	<b>Obecné vlastnosti příkazu</b>	<b>3</b>
3.1	Složení příkazu . . . . .	3
3.2	Příkazy pro čtení, příkazy pro zápis . . . . .	3
3.2.1	Příkazy pro čtení . . . . .	3
3.2.2	Příkazy pro zápis . . . . .	3
3.3	Odpověď na příkaz . . . . .	3
3.3.1	Odpověď na chybně zadaný příkaz . . . . .	3
3.4	Zadání příkazu . . . . .	3
3.4.1	Odeslání příkazu . . . . .	3
3.4.2	Zpětné zobrazení znaků . . . . .	3
3.4.3	Přerušování zadávání příkazu . . . . .	3
3.4.4	Vymazání posledního zapsaného znaku . . . . .	4
3.4.5	Časový limit pro zadání příkazu . . . . .	4
3.4.6	Zadání příkazu přes síť GSM . . . . .	4
3.4.6.1	Pomocí SMS zprávy . . . . .	4
3.4.6.2	Pomocí e-mailu . . . . .	4
3.4.6.3	Zadání příkazu pomocí modemu . . . . .	4
3.4.6.4	Zadání příkazu pomocí programu telnet . . . . .	4
<b>4</b>	<b>Formát jednotlivých složek příkazu</b>	<b>5</b>
4.1	Příklady zadání příkazu . . . . .	5
4.2	Zápis textových řetězců . . . . .	5
4.2.1	Řetězce o známé délce, řetězce o neznámé délce . . . . .	5
4.2.2	Řetězce o známé délce . . . . .	5
4.2.3	Řetězce o neznámé délce . . . . .	5
4.2.4	Jedna mezera jako univerzální oddělovač . . . . .	6
4.3	Parametry příkazu . . . . .	6
4.3.1	Bod v záznamu . . . . .	6
4.3.2	Datum . . . . .	6
4.3.3	Perioda . . . . .	6
4.3.4	Číslo . . . . .	6
4.3.5	Formát čísla . . . . .	6
<b>5</b>	<b>Struktura příkazu</b>	<b>8</b>
5.1	Hlavička . . . . .	8
5.2	Přepínač . . . . .	8
5.3	Tělo . . . . .	8

<b>6</b>	<b>Tělo příkazu</b>	<b>9</b>
6.1	Strom klíčových slov těla příkazu	9
6.2	date	11
6.3	i	11
6.3.1	version	11
6.3.2	device	11
6.3.3	snumber	11
6.3.4	program	11
6.3.5	name	12
6.3.6	description	12
6.3.7	type	12
6.3.8	address	12
6.4	device	13
6.4.1	type	13
6.4.1.1	MSP430ADC	13
6.4.1.2	MSP430TA	13
6.4.1.3	DS18B20	13
6.4.2	address	14
6.4.3	Verze 2.10	14
6.4.3.1	DS18B20	14
6.4.3.1.1	Zjištění kódu čidla DS18B20	14
6.4.4	cal	15
6.4.4.1	mult	15
6.4.4.2	add	15
6.4.4.3	Kalibrační konstanty při volbě typu zařízení	15
6.5	channel	16
6.5.1	max number	16
6.5.1.1	Verze 2.10	16
6.5.2	Parametry měřených kanálů	16
6.5.3	name	16
6.5.4	device	16
6.5.5	address	17
6.5.5.1	Verze 2.10	17
6.5.5.2	x	17
6.5.6	type	18
6.5.6.1	Typ 1	18
6.5.6.2	Typ 2	18
6.5.6.2.1	Zařízení typu MSP430ADC	18
6.5.6.2.2	Zařízení typu MSP430TA	18
6.5.6.3	Typ 3	18
6.5.6.3.1	Zařízení typu MSP430ADC	18
6.5.6.3.2	Zařízení typu DS18B20	18
6.5.6.4	Typ 6	18
6.5.6.5	Typ 9	18
6.5.6.6	Typ 11	19
6.5.6.7	Typ 12	19
6.5.7	format	19
6.5.8	value	19
6.5.9	sample	19
6.5.10	cal	19
6.5.10.1	mult	20
6.5.10.2	add	20

6.5.10.3	Čtení a zápis kalibračních konstant	20
6.5.10.3.1	value	20
6.5.10.3.2	format	20
6.5.10.4	Vysvětlení funkce kalibračních konstant	20
6.5.11	cal2	21
6.5.12	convert	21
6.5.12.1	file name	21
6.5.12.2	Dvojměrná převodní tabulka	21
6.5.13	switch	22
6.5.13.1	output	22
6.5.13.2	advance	22
6.5.14	sampling	22
6.5.14.1	period	22
6.5.14.2	start	23
6.5.15	average samples	23
6.5.16	record	23
6.5.16.1	change	24
6.5.16.2	samples	24
6.5.16.3	value	24
6.6	record	25
6.6.1	from	25
6.6.1.1	date	25
6.6.1.2	start	25
6.6.1.3	last [number] read hold	25
6.6.1.4	last [number] read shift	25
6.6.2	last [number] read	25
6.6.3	space	25
6.6.4	type	26
6.6.4.1	cyclic	26
6.6.4.2	linear	26
6.6.5	status	26
6.6.5.1	clear	26
6.7	output	26
6.7.1	device	26
6.7.2	address	27
6.7.3	value	27
6.8	command	27
6.8.1	Verze 2.10	27
6.8.2	name	27
6.8.3	value	27
6.8.4	Odesílání dat ve zprávě	28
6.8.4.1	Odesílání záznamu	28
6.8.4.2	Odesílání aktuálních změřených hodnot	29
6.8.5	Vyvolání příkazu u modulu s vlastní adresou	29
6.9	command channel switch hold	29
6.10	address	29
6.10.1	Verze 2.10	29
6.10.2	type	29
6.10.2.1	sms	29
6.10.2.2	email	30
6.10.2.3	ringvoice	30
6.10.2.4	ringdata	30

6.10.3	value . . . . .	30
6.11	com . . . . .	31
6.11.1	Řízení toku dat . . . . .	31
6.11.2	speed . . . . .	31
6.11.3	max speed . . . . .	31
6.11.4	transmitter mode . . . . .	31
6.11.4.1	saving . . . . .	31
6.11.4.2	follow . . . . .	31
6.11.4.3	on . . . . .	31
6.11.5	device . . . . .	31
6.11.5.1	local . . . . .	32
6.11.5.2	wavecom . . . . .	32
6.11.5.2.1	Vlastnosti společné všem variantám . . . . .	32
6.11.5.2.2	wavecom40 . . . . .	32
6.11.5.2.3	wavecom41 . . . . .	32
6.11.5.2.4	wavecom42 . . . . .	32
6.11.5.3	mode . . . . .	32
6.11.5.3.1	saving . . . . .	32
6.11.5.3.2	gsm . . . . .	32
6.11.5.3.3	gprs . . . . .	32
6.11.6	network . . . . .	33
6.11.6.1	smc . . . . .	33
6.11.6.2	apn address . . . . .	33
6.11.6.3	smtp address . . . . .	33
6.11.6.4	pop3 . . . . .	33
6.11.6.4.1	address . . . . .	33
6.11.6.4.2	size . . . . .	33
6.11.6.4.3	time . . . . .	34
6.11.6.5	delay . . . . .	34
6.11.7	status sim credit code . . . . .	34
6.11.8	close . . . . .	34
6.12	sms . . . . .	34
6.12.1	command boundary . . . . .	34
6.12.2	email address . . . . .	34
6.12.2.1	start . . . . .	34
6.12.2.2	end . . . . .	34
6.13	email send . . . . .	35
6.13.1	from . . . . .	35
6.13.2	subject . . . . .	35
6.14	message . . . . .	36
6.14.1	Verze 2.10 . . . . .	36
6.14.2	address . . . . .	36
6.14.2.1	Verze 2.10 . . . . .	36
6.14.3	command . . . . .	36
6.14.4	period . . . . .	36
6.14.5	start . . . . .	36
6.14.6	of record . . . . .	36
6.14.6.1	function . . . . .	36
6.14.7	of channel . . . . .	37
6.14.7.1	function . . . . .	37
6.14.7.2	number . . . . .	37
6.14.7.3	value . . . . .	37

6.14.7.4	hysteresis . . . . .	37
6.14.8	run . . . . .	37
6.14.9	Blokování zprávy . . . . .	37
6.14.9.1	GSM modem . . . . .	37
6.14.9.2	Počítač připojený přes kabel . . . . .	37
6.15	file . . . . .	38
6.15.1	record size . . . . .	38
6.15.2	list size . . . . .	38
6.15.3	data size . . . . .	38
6.15.4	free size . . . . .	38
6.16	special . . . . .	38
6.16.1	Zápis seznamu tabulek . . . . .	39
6.16.2	Zápis dat tabulek . . . . .	39
6.16.3	Čtení souboru record . . . . .	39
6.16.4	Upgrade programu modulu . . . . .	39
6.17	global . . . . .	39
6.17.1	reset . . . . .	39
6.17.2	timing . . . . .	40
6.18	nothing . . . . .	40
6.19	text . . . . .	40
<b>7</b>	<b>Zřetězení příkazu</b>	<b>41</b>
7.1	Zřetězení celého příkazu . . . . .	41
7.2	Zřetězení příkazu za prvním klíčovým slovem . . . . .	41
<b>8</b>	<b>Hlavička příkazu</b>	<b>42</b>
8.1	Struktura hlavičky příkazu . . . . .	42
8.2	Klíčová slova hlavičky příkazu . . . . .	42
8.3	Kontrola správnosti přijatého příkazu . . . . .	42
8.3.1	check . . . . .	42
8.4	Kontrola správnosti přijaté odpovědi . . . . .	42
8.4.1	crcsum . . . . .	43
8.4.2	sum . . . . .	43
8.5	Způsob odpovědi na příkaz . . . . .	43
8.5.1	pause . . . . .	43
8.5.2	comspeed . . . . .	43
8.5.3	quiet . . . . .	44
8.5.4	startline . . . . .	44
8.5.5	endline . . . . .	44
8.5.6	iname . . . . .	44
8.5.7	counter . . . . .	45
8.5.8	number . . . . .	45
8.5.9	date . . . . .	45
8.6	Adresace modulu pro který je příkaz určen . . . . .	45
8.6.1	iaddress . . . . .	45
<b>9</b>	<b>Záznam</b>	<b>46</b>

<b>10</b>	<b>Převodní tabulky</b>	<b>47</b>
10.1	Rozdělení paměti modulu . . . . .	47
10.2	Formát jednorozměrné tabulky . . . . .	47
10.3	Formát dvojrozměrné tabulky . . . . .	48
10.4	Příprava dat pro zápis tabulek do modulu . . . . .	49
10.4.1	Zápis tabulek do modulu . . . . .	49
<b>11</b>	<b>Seznam parametrů modulu a jejich počáteční hodnoty z výroby</b>	<b>50</b>
<b>12</b>	<b>Popis konektorů</b>	<b>52</b>
12.1	Rozmístění konektorů . . . . .	52
12.2	Konektor K1, napájení . . . . .	52
12.3	Konektor K2, připojení měřených signálů nebo výstupy . . . . .	52
12.4	Konektor K6, připojení dalších měřených signálů nebo výstupy . . . . .	53
12.5	Konektor K3, komunikační rozhraní RS-232 . . . . .	53
12.6	Konektor K4, nepoužitý . . . . .	53
12.7	Konektor K5, doplňkový signál pro komunikační rozhraní RS-232 . . . . .	53
<b>13</b>	<b>Schéma zapojení</b>	<b>54</b>

# 1 Úvod

## 1.1 Verze komunikačního protokolu

Tento návod popisuje verze 1.16 a 2.10. Verze 2.10 obsahuje všechny vlastnosti jako verze 1.16, a navíc má některá rozšíření. Rozšíření je uvedeno vždy u popisu každého parametru, kterého se týká.

## 1.2 Komunikace s modulem

Modul vykonává textové příkazy. Uživatel odešle příkaz přes komunikační rozhraní, modul příkaz vykoná a odešle odpověď. Příkazy slouží ke čtení nebo změně vnitřních parametrů modulu, k zobazení aktuálních změřených hodnot vstupních signálů, ke čtení záznamu změřených hodnot z paměti, a k nastavování výstupních signálů.

## 1.3 Parametry modulu

Činnost modulu se ovládá nastavováním jeho vnitřních parametrů. Některé parametry zachovávají po nastavení svou hodnotu, například vzorkovací perioda pro měření vstupních signálů, jiné svou hodnotu mění, například datum a čas. Všechny parametry a záznam se ukládají do paměti nezávislé na napájecím napětí, a proto uchovávají svou hodnotu po vypnutí.



## 2 Použitá označení v této příručce

### 2.1 Označení příkazu a odpovědi

Příkaz odesílaný uživatelem do modulu je označen:

⇒

a odpověď z modulu:

⇐

### 2.2 Označení parametrů modulu

Parametry modulu jsou označeny podle pořadí klíčových slov, kterými se z daného parametru čte nebo do něj zapisuje hodnota, například takto: **channel/1/sampling/period**. Hodnota tohoto parametru, který znamená vzorkovací periodu pro měření prvního kanálu, se přečte z modulu příkazem

```
⇒ read channel 1 sampling period
⇐ 000100
   OK
```

Vzorkovací perioda je 1 minuta. Perioda 2 hodiny se zapíše příkazem:

```
⇒ write channel 1 sampling period 020000
⇐ OK
```

### 2.3 Zvýraznění textu

Pro všeobecné zvýraznění části textu se používá **toto písmo**.

## 3 Obecné vlastnosti příkazu

### 3.1 Složení příkazu

Příkaz se skládá z několika slov, za nimiž následuje ukončení řádku.

### 3.2 Příkazy pro čtení, příkazy pro zápis

Jsou dva druhy příkazů – příkazy pro čtení a příkazy pro zápis.

#### 3.2.1 Příkazy pro čtení

Odpověď obsahuje požadovaná data z modulu, což je většinou hodnota některého parametru. Po vykonání příkazu modul odpoví jedním nebo více řádky, obsahujícími data. Posledním řádkem je text

← OK

#### 3.2.2 Příkazy pro zápis

Příkaz obsahuje nová data, která se zapíší do modulu, a tato data mění hodnotu některého parametru. Po vykonání příkazu modul odpoví jediným řádkem

← OK

### 3.3 Odpověď na příkaz

Odpověď na příkaz se skládá z jednotlivých řádků. Při konfiguraci modulu a při zadávání příkazu je nutno počítat s tím, aby délka žádného řádku odpovědi nepřekročila 100 znaků.

#### 3.3.1 Odpověď na chybně zadaný příkaz

Modul zopakuje celý chybný příkaz, a přidá řádek

← ERROR

### 3.4 Zadání příkazu

#### 3.4.1 Odeslání příkazu

Příkaz se zadá zápisem textu příkazu a odešle se ukončením řádku. Maximální délka příkazu je 120 znaků.

#### 3.4.2 Zpětné zobrazení znaků

Zapíše-li se před příkazem znak +, zapne se pro tento příkaz zpětné zobrazování znaků, echo. Modul opakuje a posílá nazpět všechny zapisované znaky.

#### 3.4.3 Přerušování zadávání příkazu

Znakem ESC se zadávání příkazu přerušuje, modul vymaže všechny přijaté znaky a očekává nový příkaz. Pokud je zapnuto zpětné zobrazování znaků, modul pošle text

← BREAK

Znak ESC je možno použít vždy před zápisem příkazu, pokud není jisté zda nebyly do modulu odeslány nepatřičné znaky.

### 3.4.4 Vymazání posledního zapsaného znaku

Je-li zapnuto zpětné zobrazování znaků, pak znakem **Backspace** se vymaže poslední zapsaný znak. Tak lze zadávaný příkaz před jeho odesláním opravovat.

### 3.4.5 Časový limit pro zadání příkazu

Celý příkaz od prvního znaku až do odeslání musí být zadán v průběhu 1 minuty. Po uplynutí této doby modul ukončí čtení příkazu, dosud přijaté znaky vymaže, a očekává nový příkaz. Pokud je zapnuto zpětné zobrazování znaků, pošle text

← BREAK

Zachová se tedy stejně, jako by přijal znak **ESC** pro přerušování zadávání příkazu.

### 3.4.6 Zadání příkazu přes síť GSM

Je-li modul připojen do sítě GSM (6.11.5), lze mu poslat příkazy následujícími způsoby.

#### 3.4.6.1 Pomocí SMS zprávy

Pro odlišení příkazu od ostatního textu ve zprávě se používá řetězec, který musí být uveden před a za příkazem (6.12.1). V SMS zprávě může být kromě příkazu napsána zpáteční e-mailová adresa, na kterou má být odeslána odpověď (6.12.2). Není-li zpáteční e-mailová adresa uvedena, je odpověď odeslána SMS zprávou na telefonní číslo odesílatele.

#### 3.4.6.2 Pomocí e-mailu

Vždy po odeslání každého e-mailu jako odpovědi na příkaz nebo jako zprávy (6.14) se kontroluje e-mailová příkazová schránka, a stahuje se z ní e-mail s příkazem. Pro odlišení příkazu od ostatního textu v e-mailu se používá řetězec, který musí být uveden před a za příkazem (6.12.1).

#### 3.4.6.3 Zadání příkazu pomocí modemu

Pomocí modemu vytočíme telefonní číslo modulu, a jakmile je spojení vytvořeno, můžeme zadávat příkazy stejně jako při kabelovém připojení k počítači. Po skončení činnosti ukončíme spojení k modulu.

Pokud modul po dobu delší než 2 minuty neobdrží žádný příkaz, sám ukončí modemové spojení.

#### 3.4.6.4 Zadání příkazu pomocí programu telnet

Je-li modul připojen přes GPRS do internetu, lze mu poslat příkaz pomocí programu telnet. Otevřeme port 10001 na ip adrese, na které se modul nachází. Jakmile je spojení přes internet vytvořeno, můžeme zadávat příkazy stejně jako při kabelovém připojení k počítači. Po skončení činnosti ukončíme internetové spojení k modulu.

Pokud modul po dobu delší než 2 minuty neobdrží žádný příkaz, sám ukončí internetové spojení.

## 4 Formát jednotlivých složek příkazu

Příkaz se skládá z textových řetězců. Modul rozeznává sadu pevně určených textových řetězců, nazvaných **klíčová slova**. Za některými klíčovými slovy mohou následovat další volitelné řetězce, **parametry příkazu**. Klíčová slova lze psát malými i velkými písmeny. Naproti tomu u parametrů se malá a velká písmena rozlišují.

### 4.1 Příklady zadání příkazu

- *Příklad*

Příkaz pro nastavení data a času na 1. června 2007, 9 hodin, 25 minut, 30 sekund (6.2) můžeme zadat následujícími způsoby:

```
⇒ write date 2007 06 01 09 25 30
⇒ write date 20070601092530
⇒ writedate20070601092530
⇒ write      date      20070601092530
```

- *Příklad*

Příkaz pro označení modulu názvem **Merici bod 1115** (6.3.6):

```
⇒ write i description/Merici bod 1115/
⇒ write   idescription*Merici bod 1115*
⇒ writeidescriptionAMerici bod 1115A
```

- *Příklad*

Příkaz pro čtení data a času zabezpečený kontrolním součtem (8):

```
⇒ check 890 read date
⇒ check826readdate
```

### 4.2 Zápis textových řetězců

#### 4.2.1 Řetězce o známé délce, řetězce o neznámé délce

Pro způsob zápisu řetězce je rozhodující, zda modul může sám poznat délku řetězce, tedy jeho začátek a konec, nebo jestli tuto délku musí určit uživatel při zadávání příkazu. V prvním případě se jedná o řetězec o známé délce, v druhém o neznámé délce.

#### 4.2.2 Řetězce o známé délce

Řetězce o známé délce jsou všechna klíčová slova a některé parametry příkazu. Řetězce o známé délce se mohou v příkazu psát těsně za sebou bez mezery, nebo je lze oddělit vložení jedné nebo více mezer.

#### 4.2.3 Řetězce o neznámé délce

Před řetězcem o neznámé délce se při zadávání příkazu musí napsat hraniční znak. To je libovolný znak, který se v řetězci nevyskytuje. Stejný znak se napíše na konci řetězce, a modul tak pozná začátek a konec. Je-li zadávaný řetězec posledním řetězcem v příkazu, nemusí se na konci před odesláním příkazu hraniční znak psát.

#### 4.2.4 Jedna mezera jako univerzální oddělovač

Z výše uvedeného vyplývá, že pro řetězce o známé i neznámé délce lze použít jako oddělovač jednu mezeru. Podmínkou je, že mezera není součástí řetězce o neznámé délce.

### 4.3 Parametry příkazu

Zde je uveden způsob zápisu některých typů parametrů příkazu.

#### 4.3.1 Bod v záznamu

Bod v záznamu slouží k určení data a času, od kterého se čte záznam z paměti. Skládá se ze sedmi polí – rok, měsíc, den v měsíci, hodina, minuta, sekunda a rozlišovací čítač. Rok má délku 4 znaky, rozlišovací čítač 1 znak, ostatní pole 2 znaky. Rozlišovací čítač je od pole sekund oddělen tečkou a slouží k jednoznačnému určení řádků se stejným datem a časem. Jeho hodnota začíná pro každé nové datum a čas od 0. Jednotlivá pole následují těsně za sebou. Řetězec nemusí být uveden v celé délce, je možno vynechat libovolný počet znaků zprava. Bod v záznamu je řetězec o neznámé délce.

#### 4.3.2 Datum

Udává datum a čas. Skládá se z šesti polí – rok, měsíc, den v měsíci, hodina, minuta, sekunda. Rok má délku 4 znaky, ostatní pole 2 znaky. Jednotlivá pole následují těsně za sebou, nebo mohou být oddělena mezerami. Pokud je zapsaný počet polí menší než šest, pak modul považuje za chybějící pole zleva, a tato pole si doplní podle údaje svých vnitřních hodin. Tak například stačí zadat nové minuty a sekundy, a modul doplní stávající hodinu, den, měsíc a rok. Datum je řetězec o známé délce.

#### 4.3.3 Perioda

Vyjadřuje časovou periodu nebo časový úsek. Skládá se ze tří polí – hodina, minuta, sekunda, každé pole má délku 2 znaky. Jednotlivá pole následují těsně za sebou nebo mohou být oddělena mezerami. Pokud je počet zapsaných polí menší než tři, pak modul považuje za chybějící pole zprava, a tato pole si doplní nulami. Tedy například stačí uvést hodiny, a za minuty a sekundy je automaticky dosazena nula. Maximální nastavitelná perioda v uvedeném formátu je omezena na 96 hodin, tedy na 4 dny.

Pro zápis delší periody se používá následující formát. V prvním poli se uvede číslo 99, které se ignoruje a slouží jako indikátor, že druhé pole obsahuje počet dnů a třetí pole počet hodin. Takto lze nastavit periodu maximálně 90 dnů, například **999000**.

Dále lze nastavit periodu 1 měsíc, ale pouze pro periodu zasílání zpráv (6.14.4). V prvním i druhém poli se uvedou čísla 99, a ve třetím poli číslo 1, tedy **999901**.

Perioda je řetězec o známé délce.

#### 4.3.4 Číslo

Pokud není jinak upřesněno, rozumí se desetinné číslo v zápisu: případné znaménko mínus, pak celá část, desetinná tečka, a část za desetinnou tečkou. Je-li upřesněno, že se jedná o celé číslo, pak je bez desetinné tečky. Číslo je řetězec o známé délce. Jestliže se však v příkazu vyskytnou vedle sebe dvě nebo více čísel, u nichž není stanoven počet číslic, je nutno je oddělit mezerou.

#### 4.3.5 Formát čísla

Určuje, v jakém tvaru bude modul při čtení zobrazovat čísla. Formát čísla se uvádí řetězcem o délce 3 znaky ve tvaru **w.p**, kde **w** je celková šířka textu a **p** počet desetinných míst. Obsahuje-li zobrazovaná hodnota menší počet znaků než určuje **w**, doplní se zleva mezerami, obsahuje-li více znaků, pak se **w** ignoruje, a zobrazí se všechny nezbytné znaky. Je-li **w** rovno nule, není číslo zleva doplňováno žádnými

mezerami, a je tak zobrazováno maximálně úsporně. Je-li  $p$  rovno nule, není desetinná tečka zobrazována, a desetinné číslo je zobrazováno jako celé. Formát čísla je řetězec o známé délce.

## 5 Struktura příkazu

Příkaz se skládá ze tří částí:

1. hlavička příkazu
2. prepínač čtení / zápis
3. tělo příkazu

### 5.1 Hlavička

Hlavička příkazu je nepovinná, nemusí tedy být přítomna. Z důvodu lepší srozumitelnosti výkladu bude nejprve vysvětlen prepínač, pak tělo a nakonec hlavička příkazu.

### 5.2 Prepínač

Prepínač je klíčové slovo **write** nebo **read**. Prepínač **write** znamená, že příkaz bude zapisovat data a měnit parametry modulu. Prepínač **read** znamená čtení z modulu beze změny parametrů.

### 5.3 Tělo

Za prepínačem následují vybraná klíčová slova těla příkazu dle stromu klíčových slov, doplněná potřebnými parametry příkazu. Některé příkazy umožňují pouze čtení nebo pouze zápis.

Jestliže v popisu není u parametru příkazu uvedeno, zda je to řetězec o známé nebo neznámé délce, jedná se o parametr o neznámé délce.

## 6 Tělo příkazu

### 6.1 Strom klíčových slov těla příkazu

date				
i	version			
	device			
	snumber			
	program			
	name			
	description			
	type			
	address			
device	type	msp430adc		
		msp430ta		
		ds18b20		
	address			
	cal	mult		
		add		
channel max number				
channel [n]	name			
	device	all		
	address	all		
	type			
	format			
	value		value	
	sample		format	
	cal	mult		
		add		
	cal2		value	
		mult	format	
		add		
	switch		all	
		output	all	
		advance	all	
			all	
	convert 1   2			
	sampling	file name		
		period		
		start		
	average samples			
	record	change		
		samples		
		value		
record	from			
		date		
		start		
		last [n] read hold   shift		
	last [n] read			
	space			
	type cyclic   linear			
	status			
output		clear		
	device			
	address			
	value			
command max number		no		



command [n]	name value		
command channel switch hold			
address max number			
address [n]	type	sms email ringvoice ringdata	
com	value		
	speed max speed transmitter mode	saving follow on	
	device	local wavecom[n] mode	saving gsm gprs
	network	smc apn address smtp address pop3	address size time
	status sim credit code close	delay	
sms	command boundary email address		
email send		start end	
message max number	from subject		
message [n]	address command period start of record		
	of channel	function no   yes	
		function no   up   down number value hysteresis	
file	run		
special	record   list   data   free size		
	file		
global	program xmodem	record   list   data xmodem	
nothing	reset timing		
text			

## 6.2 date

Datum a čas vnitřních hodin modulu ve formátu dle 4.3.2.

- *Příklad*

Datum a čas na 1. března 2007, 9 hodin nastavíme takto.

```
⇒ write date 20070301090000
← OK
```

Takto nastavíme čas na 9 hodin. Rok, měsíc a den si doplní modul automaticky podle svých vnitřních hodin.

```
⇒ write date 090000
← OK
```

Přečtení data a času provedeme příkazem

```
⇒ read date
← 20070301090000
   OK
```

## 6.3 i

Identifikace modulu.

### 6.3.1 version

Pouze čtení. Verze komunikačního protokolu.

- *Příklad*

Tento návod popisuje protokol verze 1.16 nebo 2.10. To znamená

```
⇒ read i version
← 1.16
   OK
```

nebo

```
← 2.10
   OK
```

### 6.3.2 device

Pouze čtení. Označení verze hardwaru.

### 6.3.3 snumber

Pouze čtení. Výrobní číslo.

### 6.3.4 program

Pouze čtení. Označení verze programu.

### 6.3.5 name

Řetězec o maximální délce 20 znaků, kterým můžeme modul označit. Pokud modul odesílá zprávy se změřenými daty do databáze, slouží tento řetězec pro identifikaci, ze kterého modulu data přicházejí.

### 6.3.6 description

Řetězec o maximální délce 20 znaků, kterým můžeme modul označit.

- *Příklad*

```
⇒ write i description*Varovny system 111*  
← OK
```

```
⇒ read i description  
← Varovny system 111  
OK
```

### 6.3.7 type

Řetězec o maximální délce 20 znaků, který lze použít k označení určité konfigurace parametrů modulu. Moduly stejného typu by měly být záměnné. Typ zahrnuje počet a druh měřených veličin, rychlost na komunikačním rozhraní a další parametry.

### 6.3.8 address

Řetězec o maximální délce 10 znaků pro nastavení vlastní adresy modulu. Pokud je tento parametr prázdný řetězec, přijímá modul všechny příkazy. Zapišeme-li do parametru nějaký text, slouží jako vlastní adresa. V hlavičce každého příkazu je pak nutno tuto adresu uvést (8.6.1), jinak modul příkaz ignoruje. Parametr lze využít jako heslo, nebo v případě kdy je na komunikační linku připojeno současně více modulů pro adresaci vybraného modulu.

## 6.4 device

Konfigurace zvoleného měřicího nebo výstupního zařízení modulu. Za tímto klíčovým slovem následuje parametr, celé číslo v rozmezí 1 až 8, pro určení čísla zařízení, kterého se příkaz týká. Celkem lze tedy nakonfigurovat 8 zařízení.

### 6.4.1 type

Typ zařízení.

#### 6.4.1.1 MSP430ADC

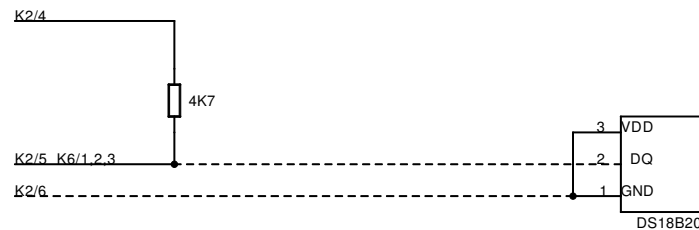
V funkci jako měřicí zařízení je to A/D převodník mikrokontroleru MSP430F149 na pinech portu P6. Při použití jako výstupní zařízení jsou to piny portu P6 ve funkci binárního výstupu.

#### 6.4.1.2 MSP430TA

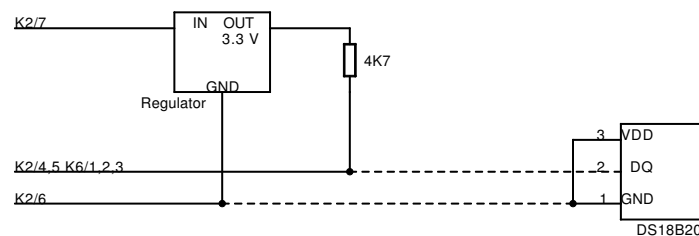
Ve funkci jako měřicí zařízení je to čítač A mikrokontroleru MSP430F149 se vstupem na pinu P1.0. Při použití jako výstupní zařízení je to pin P1.0 ve funkci binárního výstupu.

#### 6.4.1.3 DS18B20

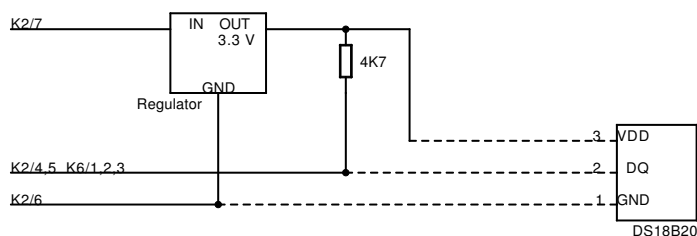
Pouze jako měřicí zařízení. Teplotní čidlo Dallas DS18B20 připojené k některému pinu portu P6. K modulu lze připojit až pět čidel DS18B20. Na následujícím obrázku je příklad připojení.



Toto zapojení využívá dva piny mikrokontroleru, jeden pro připojení výstupu obvodu DS18B20, druhý ve funkci výstupu pro napájení zatěžovacího rezistoru. Použijeme-li vnější stabilizátor 3.3 V, postačí jeden pin. Příklad zapojení je na následujícím obrázku.



Oba výše uvedené příklady používají dvoudrátové připojení obvodu DS18B20. Tento obvod lze připojit i třídrátově například podle následujícího zapojení.



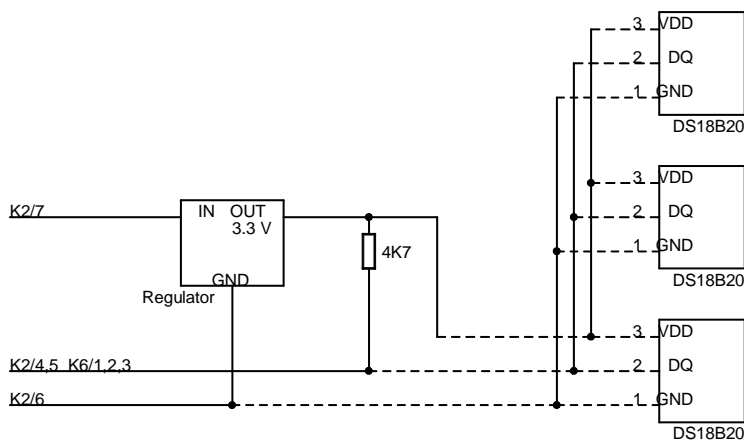
### 6.4.2 address

Celé číslo pro určení adresy, na které se zařízení nachází. Má význam pouze u zařízení typu DS18B20, kde označuje bit portu P6, ke kterému je připojen datový výstup obvodu DS18B20. U ostatních typů zařízení se adresa ignoruje.

### 6.4.3 Verze 2.10

#### 6.4.3.1 DS18B20

K jednotlivým pinům portu P6 lze připojit paralelně více čidel DS18B20. Na následujícím obrázku je příklad připojení.



Všechna čidla připojená paralelně ke stejnému bitu portu P6 tvoří jedno společné měřicí zařízení typu DS18B20. Rozlišení jednotlivých čidel se provádí zadáním adresy kanálu (6.5.5) jako unikátního 64-bitového kódu, kterým je každé čidlo DS18B20 z výroby označeno. Při zapojení více čidel paralelně je nutno použít vždy třídrátové připojení se stabilizátorem 3.3 V z důvodu většího zatížení.

Pokud nejsou čidla připojena paralelně, není nutno rozlišovat jejich kód. V tom případě zadáme adresu kanálu 0.

#### 6.4.3.1.1 Zjištění kódu čidla DS18B20

Pro zjištění kódu neznámého čidla jej připojíme samostatně k modulu, a namísto adresy 0 zadáme jako adresu (6.5.5) kanálu detekční adresu. Detekční adresa je libovolná adresa, která má v hexadecimálním

vyjádření nejnižší byte 0, a alespoň jeden další bit různý od nuly. Vyhoví tedy např. adresa FF00. Pak provedeme měření aktuální hodnoty kanálu (6.5.8). Při tomto měření se provede nejen změření aktuální hodnoty, ale také se zjistí 64-bitový kód čidla a uloží se jako nová adresa kanálu, přičemž před uložením se vynuluje nejnižší byte. Skutečnou adresu čidla zjistíme přečtením adresy kanálu, ve které nahradíme nejnižší byte hodnotou 28 hexadecimálně.

Protože takto uložený kód splňuje i nadále požadavky na detekční adresu, můžeme postupně připojovat další jednotlivá čidla a zjišťovat jejich kódy.

Přestože kanál s detekční adresou poskytuje při měření normální změřené hodnoty, nedoporučujeme používat detekční adresu pro jiný účel než pouze pro zjišťování kódu.

#### 6.4.4 cal

##### 6.4.4.1 mult

Multiplikativní kalibrační konstanta pro zařízení.

Tato konstanta je u zařízení typu **MSP430ADC** rovna vnitřnímu nebo vnějšímu referenčnímu napětí A/D převodníku podle toho, zda kanály používající toto zařízení měří pomocí vnitřního nebo vnějšího referenčního napětí. Výrobní hodnota je zkalibrována pro vnitřní referenční napětí a je nutno ji změnit při použití vnějšího referenčního napětí.

U zařízení typu **MSP430TA** je rovna 1, u **DS18B20** je rovna 3276.8 .

##### 6.4.4.2 add

Aditivní kalibrační konstanta pro zařízení.

Výrobní hodnota u zařízení typu **MSP430ADC** je zkalibrována pro vnitřní referenční napětí A/D převodníku.

U zařízení typu **MSP430TA** a **DS18B20** je rovna 0.

##### 6.4.4.3 Kalibrační konstanty při volbě typu zařízení

Kdykoliv se přiřadí některému zařízení typ, například příkazem

```
⇒ write device 1 type MSP430ADC
```

automaticky se vrátí multiplikativní i aditivní kalibrační konstanta k výrobním hodnotám.

## 6.5 channel

Měřené kanály.

### 6.5.1 max number

Celé číslo 1 až 8. Slouží pro nastavení počtu kanálů. Celkem lze tedy konfigurovat maximálně 8 kanálů.

#### 6.5.1.1 Verze 2.10

Lze konfigurovat maximálně 12 kanálů.

### 6.5.2 Parametry měřených kanálů

Tato skupina parametrů určuje způsob zpracování signálů, přivedených na vstupy modulu.

Za klíčovým slovem **channel** může následovat nepovinný parametr, pole celých čísel, každé o délce 2 znaky, která určují čísla kanálů, kterých se příkaz týká. Pokud nejsou žádná čísla zadána, týká se čtení nebo zápis všech kanálů. Při čtení jsou hodnoty parametrů jednotlivých kanálů odděleny čárkami.

Pokud chceme vybrat pouze některé kanály, uvedeme jejich čísla těsně za sebou nebo je oddělíme mezerami. Následují-li těsně za sebou, musí být doplněna na dvoumístné číslo zleva nulou, jsou-li oddělena mezerami nebo je-li uvedeno pouze jedno číslo, mohou být uvedena i jako jednomístná. Pole čísel je řetězec o známé délce.

### 6.5.3 name

Lze zadat libovolné pojmenování kanálu. Je to řetězec o maximální délce 20 znaků.

#### • *Příklad*

```
⇒ write channel max number 4
⇐ OK

⇒ write channel 2 name/Teplota vody/
⇐ OK

⇒ writechannel010304name/Teplota vzduchu//Hladina vody//Napajeci napeti/
⇐ OK

⇒ readchannelname
⇐ Teplota vzduchu,Teplota vody,Hladina vody,Napajeci napeti
OK
```

### 6.5.4 device

Číslo zařízení pro signálový vstup kanálu. Číslo **0** znamená, že kanál nemá žádný signálový vstup, a jeho změřená hodnota je vždy 0.

**6.5.5 address**

Celé číslo. Adresa vstupu, na kterém se v zařízení nachází měřený signál. Celé číslo.

- *Příklad*

```
⇒ read channel 1 address
← 21
OK
```

Pro zařízení typu **MSP430ADC** platí přiřazení adresy vstupu k číslu pinu na konektoru K2 nebo K6 dle následující tabulky. Měřený signál se připojí kladným pólem k uvedenému pinu, a zem k pinu 3 konektoru K2 – k společné signálové zemi. Adresa v sobě zahrnuje současně informaci, zda se pro měření kanálu použije vnitřní nebo vnější zdroj referenčního napětí A/D převodníku. Pokud je použit, vnější referenční zdroj se připojuje mezi piny 2 a 3 konektoru K2, a v tom případě pin 2 nelze použít jako analogový vstup pro měření signálu.

Pin K2	Adresa pro vnitřní referenční zdroj	Adresa pro vnější referenční zdroj
5	16	32
4	17	33
2	24	nelze

Pin K6	Adresa pro vnitřní referenční zdroj	Adresa pro vnější referenční zdroj
1	19	35
2	20	36
3	18	34

Pro měření napájecího napětí platí tabulka:

Adresa pro vnitřní referenční zdroj	Adresa pro vnější referenční zdroj
18	34

Pro zařízení typu **MSP430TA** se adresa kanálu ignoruje. Pro zařízení typu **DS18B20** je nutno zadat adresu kanálu 0.

**6.5.5.1 Verze 2.10**

Adresa kanálu je celé číslo, které má v hexadecimálním vyjádření délku 64 bitů. Tato zvětšená délka má význam, je-li jako měřicí zařízení kanálu použito zařízení typu DS18B20 s několika čidly připojenými paralelně. Adresa kanálu pak obsahuje unikátní 64-bitového kód, kterým je každé čidlo DS18B20 z výroby označeno (6.4.3.1).

**6.5.5.2 x**

Adresa je zadána jako hexadecimální číslo.

- *Příklad*

```
⇒ read channel 1 address x
← 15
OK
```



### 6.5.6 type

Typ měřeného kanálu. Celé číslo. Určuje způsob zpracování signálu ze vstupu a jeho transformaci do změřené hodnoty.

Jsou tyto typy kanálu:

#### 6.5.6.1 Typ 1

Měření napájecího napětí. Pro zařízení typu **MSP430ADC**.

#### 6.5.6.2 Typ 2

Měření frekvence – počtu impulzů za vzorkovací periodu.

##### 6.5.6.2.1 Zařízení typu MSP430ADC

Maximální měřitelná frekvence závisí na nastavení parametru **global/timing** (6.17.2), a může být až 2 KHz.

##### 6.5.6.2.2 Zařízení typu MSP430TA

Maximální měřitelná frekvence je 50 KHz.

#### 6.5.6.3 Typ 3

Měření analogového signálu.

##### 6.5.6.3.1 Zařízení typu MSP430ADC

Měření napětí. Lze měřit od 0 do hodnoty referenčního napětí A/D převodníku. Při použití vnitřního referenčního napětí je zaručený rozsah 0 až 2 V, a skutečný rozsah 0 až přibližně 2.4 V.

##### 6.5.6.3.2 Zařízení typu DS18B20

Měření teploty v rozsahu  $-50\text{ }^{\circ}\text{C}$  až  $+125\text{ }^{\circ}\text{C}$ .

#### 6.5.6.4 Typ 6

Měření frekvence počtu impulzů za vzorkovací periodu. Používá se pro sledování občas a náhodně se vyskytujících impulzů.

Funkčně podobný jako typ 2. U parametru kanálu **channel/record/change** (6.5.16.1) se však nevyhodnocuje změna oproti poslední zapsané hodnotě v záznamu, ale vezme se přímo načítaný počet impulzů za vzorkovací periodu. Zápis do záznamu se provede, když se za vzorkovací periodu naměří alespoň počet impulzů dle parametru **channel/record/change**.

Program ALAConnect pro tento typ kanálu automaticky zvolí jiný způsob grafické prezentace než pro ostatní typy, a to nespojitý sloupcový graf.

#### 6.5.6.5 Typ 9

Kanál měří dva analogové signály, a výslednou změřenou hodnotu počítá pomocí dvojrozměrné převodní tabulky (6.5.12.2).

### 6.5.6.6 Typ 11

Kanál měří kredit SIM karty v síti GSM (6.11.7). Zjištěná hodnota kreditu je aktualizována vždy v době komunikace v síti GSM bez ohledu na nastavenou vzorkovací periodu kanálu. Vzorkovací perioda určuje pouze četnost zapisování do záznamu. U tohoto typu kanálu se ignoruje parametr **channel/device** (6.5.4).

### 6.5.6.7 Typ 12

Kanál měří sílu signálu v síti GSM. Zjištěná hodnota je aktualizována vždy v době komunikace v síti GSM bez ohledu na nastavenou vzorkovací periodu kanálu. Vzorkovací perioda určuje pouze četnost zapisování do záznamu. U tohoto typu kanálu se ignoruje parametr **channel/device** (6.5.4).

### 6.5.7 format

Formát změřené hodnoty, jak je popsán v 4.3.5. V tomto formátu je změřená hodnota zapisována do záznamu i zobrazována aktuální hodnota.

- *Příklad*

```
⇒ write channel 1 2 format 0.1 0.2
```

nebo

```
⇒ writechannel0102format0.10.2
```

```
← OK
```

Změřené hodnoty prvního kanálu budou zobrazovány na 1 desetinné místo, hodnoty druhého kanálu na 2 desetinná místa.

### 6.5.8 value

Modul provede změření aktuální hodnoty vstupního signálu v zadaném kanálu. S přepínačem **read** se aktuální změřená hodnota zobrazí v odpovědi. S přepínačem **write** se aktuální změřená hodnota uloží do záznamu jako řádek typu 2, nebo lze typ řádku v rozmezí 1 až 9 zvolit jako parametr na konci příkazu. (9).

- *Příklad*

```
⇒ readchannelvalue
```

```
← 2.0,10.51,13.8,12.0
```

```
OK
```

```
⇒ readchannel2value
```

```
← 10.51
```

```
OK
```

### 6.5.9 sample

Pouze čtení. Poslední změřená hodnota vstupního signálu dle nastavené vzorkovací periody. Modul neprovádí žádné měření, pouze zobrazí údaj uložený v paměti.

### 6.5.10 cal

Kalibrační konstanta. Číslo (4.3.4), které slouží k úpravě změřených hodnot a pro přizpůsobení k externím čidlům.

**6.5.10.1 mult**

Multiplikační kalibrační konstanta. Tímto číslem se změřená hodnota násobí.

**6.5.10.2 add**

Aditivní kalibrační konstanta. Přičte se k změřené hodnotě.

**6.5.10.3 Čtení a zápis kalibračních konstant****6.5.10.3.1 value**

Čtení nebo zápis hodnoty kalibrační konstanty.

**6.5.10.3.2 format**

Čtení nebo zápis formátu kalibrační konstanty (4.3.5).

**6.5.10.4 Vysvětlení funkce kalibračních konstant**

Označíme-li *PuvodniHodnota* změřenou hodnotu při multiplikační konstantě rovné jedné a aditivní konstantě rovné nule, *Mult* hodnotu multiplikační a *Add* hodnotu aditivní konstanty, pak platí vztah:

$$NovaHodnota = (PuvodniHodnota * Mult) + Add$$

Použití kalibračních konstant tedy umožňuje lineární transformaci původní změřené hodnoty na novou hodnotu.

• Příklad

Předpokládáme že původní změřená hodnota v kanále 3 je 13.8.

```

=> read channel 3 value
<= 13.8
    OK

=> write channel 3 cal mult value 10
<= OK

=> read channel 3 value
<= 138.0
    OK

=> write channel 3 cal add value -0.8
<= OK

=> read channel 3 value
<= 137.2
    OK

```

Při posledním příkazu pro změření aktuální hodnoty třetího kanálu se provádí výpočet

$$10 * 13.8 - 0.8 = 137.2$$

### 6.5.11 cal2

Druhá sada kalibračních konstant kanálu, kterými lze upravit změřené vstupní signály dříve než je aplikován přepoččet pomocí tabulky. U těchto konstant je nastaven pevný formát čísla **0.6** (4.3.4).

- *Příklad*

```
⇒ write channel 2 cal2 add all 7.55 3.5
⇒ write channel 2 cal2 mult all 1.85 2.91
```

### 6.5.12 convert

Převodní tabulky. Za tímto klíčovým slovem se zadá celé číslo 1 nebo 2, které určuje druh tabulky. Pokud se číslo nezadá, modul si doplní číslo 1. Tabulka číslo 1 se aplikuje na změřené vstupní signály po jejich úpravě druhou sadou kalibračních konstant (6.5.11). Poté jsou hodnoty upraveny normálními kalibračními konstantami (6.5.10), a nakonec se aplikuje tabulka číslo 2. Výsledkem je změřená hodnota daného kanálu.

#### 6.5.12.1 file name

Jméno tabulky pro přepoččet změřené hodnoty kanálu pomocí jednorozměrné nebo dvojrozměrné tabulky. Dvojrozměrná tabulka může být pouze tabulka číslo 1.

- *Příklad*

```
⇒ write channel 1 type 3
První kanál je typu 3, což je kanál pro měření analogového signálu.
⇒ write channel 1 convert file name/tabulka1/
```

Hodnota tohoto kanálu bude přepočítávána na výslednou hodnotu podle jednorozměrné tabulky **tabulka1**.

#### 6.5.12.2 Dvojrozměrná převodní tabulka

Modul umožňuje počítat výslednou změřenou hodnotu kanálu ze dvou vstupních signálů. Tento výpočet se provádí, pokud nastavíme typ kanálu **9** (6.5.6.5). V tom případě je dále nutno nastavit některé parametry zvláště pro každý vstupní signál. Jedná se o zdvojení následujících parametrů: **channel/device**, **channel/address**, **channel/cal2/mult**, **channel/cal2/add**, **channel/switch/output** a **channel/switch/advance**.

Pro čtení a zápis do těchto zdvojených parametrů doplníme na konec klíčové slovo **all**, a za ním následují oba parametry oddělené čárkou.

- *Příklad*

Takto připravíme první kanál na měření ze dvou signálových vstupů. Jako první měřicí zařízení nastavíme zařízení číslo 3 a jako druhé číslo 4. Dále nastavíme adresu prvního vstupu signálu 0, a druhého 1.

```
⇒ write channel 1 device all 3,4
⇒ write channel 1 address all 0,1
```

### 6.5.13 switch

Parametry spínače napájecího napětí pro externí čidlo. Ke každému kanálu může být přiřazen spínač, který slouží pro zapnutí čidla před měřením. Po změření je čidlo odpojeno.

Při pravidelném měření dle nastavené vzorkovací periody je spínač automaticky zapnut s potřebným předstihem před měřením kanálu.

Při měření aktuálních hodnot se spínač zapne a modul čeká před změřením hodnoty, až uplyne nastavený předstih. Po změření je spínač vypnut s nastaveným zpožděním dle parametru **command/channel/switch/hold** (6.9), aby při opakovaném měření nedocházelo k zbytečnému vypínání čidla a tím k nutnosti dalšího čekání po dobu předstihu.

#### 6.5.13.1 output

Číslo výstupu použitého pro realizaci spínače. Tento výstup musí mít správně zkonfigurované zařízení a adresu pomocí parametrů **output/device** a **output/address**. Číslo 0 znamená, že kanál nemá přiřazen žádný výstup.

#### 6.5.13.2 advance

Předstih zapnutí spínače před změřením kanálu. Je to číslo udávající předstih v sekundách.

- *Příklad*

```
⇒ write channel 2 switch advance 3
```

Nastavený předstih je 3 sekundy.

### 6.5.14 sampling

Zde se nacházejí parametry pro pravidelné měření kanálů a zápis změřených hodnot do záznamu v paměti.

#### 6.5.14.1 period

Vzorkovací perioda měření kanálu (4.3.3). S touto periodou se provádí pravidelné měření a podle dalších parametrů případně i zápis změřené hodnoty do záznamu. Zadáním hodnoty 000000 se pravidelné měření kanálu zastaví.

- *Příklad*

Nastavíme vzorkovací periodu čtvrtého kanálu na 1 hodinu.

```
⇒ write channel 4 sampling period 010000
```

Dle 4.3.3 lze stejnou periodu zadat též příkazem

```
⇒ write channel 4 sampling period 01
```

Zadáva-li se současně perioda pro několik kanálů, je nutno uvést pro každý kanál všechna tři pole tohoto parametru

- *Příklad*

```
⇒ write channel 04 05 sampling period 001000 003000
```

Nyní je nastavena perioda čtvrtého kanálu na 10 minut, a pátého na 30 minut.

**6.5.14.2 start**

Parametr ve formátu dle 4.3.2. Je to počáteční datum a čas, od kterého se počítá vzorkovací perioda kanálu. Zadáme-li datum a čas už minulý, je skutečný začátek měření posunut o potřebný počet vzorkovacích period do budoucnosti.

- Příklad

```
⇒ write channel 1 sampling start 20070301090000
```

Zde je začátek určen přesným datem a časem.

- Příklad

```
⇒ write channel 1 sampling start 0000
```

Začátek je nejbližší celá hodina. Zadává-li se současně začátek pro několik kanálů, je nutno uvést pro každý kanál všech šest polí tohoto parametru

- Příklad

```
⇒ write channel 4 5 sampling start 20070301090000 20070301090000
```

**6.5.15 average samples**

Kladné celé číslo. Parametr pro výpočet aritmetického průměru ze změřené hodnoty kanálu. Při čtení aktuální změřené hodnoty (6.5.8) se výpočet průměru neprovádí. Aritmetický průměr se počítá tak, že po změření hodnoty kanálu dle nastavené vzorkovací periody se tato nová hodnota přičte do celkové sumy, ale nepoše se do zpracování pro zápis do záznamu dle podmínek 6.5.16. Jakmile je počet změřených vzorků roven parametru, suma se vydělí tímto počtem, a výsledná hodnota teprve může být zapsána do záznamu, jsou-li splněny podmínky dle 6.5.16. Suma se následně vynuluje. To znamená, že zápis do záznamu se provádí ve skutečnosti s periodou, která je minimálně násobkem parametru a vzorkovací periody dle 6.5.14.1. Hodnota parametru 0 znamená, že je počítání průměru vypnuto.

- Příklad

První kanál se měří s periodou 1 minuta, a každou celou hodinu se do záznamu uloží aritmetický průměr.

```
⇒ write channel 1 sampling period 000100
```

```
⇒ write channel 1 average samples 60
```

```
⇒ write channel 1 record change 0
```

**6.5.16 record**

Podmínky pro zápis pravidelně změřené hodnoty kanálu do záznamu. Tyto podmínky jsou dvě, a zápis se provede, pokud je splněna alespoň jedna z nich.

### 6.5.16.1 change

Číslo. Zápis do záznamu se provede při změně změřené hodnoty oproti předchozí hodnotě již uložené v záznamu, je-li tato změna rovna nebo větší než zadané číslo. Změna se počítá jako absolutní hodnota, parametr se zadává jako kladné číslo.

- *Příklad*

⇒ `write channel 4 record change 0.02`

Je-li poslední zapsaná hodnota 1.25, pak nový zápis bude proveden při změřené hodnotě 1.23 a menší, nebo 1.27 a větší. U kanálů, které měří počet událostí za jednotku času, se podmínka vyhodnocuje odlišným způsobem. Zápis se provede, je-li změřená hodnota rovna nebo větší než zadaný parametr. Pokud při dalším měření již hodnota nesplňuje podmínky pro zápis, je do záznamu zapsána hodnota 0, a následná měření se již nezapisují, dokud není opět splněna podmínka. Jsou dva zvláštní případy hodnoty parametru. Při hodnotě 0 se provede zápis vždy. Při hodnotě -1 se tento typ zápisu do záznamu neprovádí vůbec, a provádí se pouze zápis dle parametru **channel/record/samples**, který je popsán dále.

### 6.5.16.2 samples

Celé číslo. Zápis bude prováděn pravidelně, vždy při dosažení určitého počtu změřených vzorků, který se zadává jako celé číslo. Hodnota 0 znamená, že se tento typ zápisu neprovádí, a provádí se pouze zápis dle parametru **channel/record/change**.

- *Příklad*

⇒ `write channel 4 record samples 6`

Je-li nastavena vzorkovací perioda 10 minut, pak bude změřená hodnota čtvrtého kanálu zapsána každou celou hodinu. Kombinací parametrů **channel/record/change** a **channel/record/samples** lze zajistit úsporný zápis do záznamu pouze při změně změřené hodnoty a současně vynutit po určité době zápis, i když je hodnota konstantní.

### 6.5.16.3 value

Pouze čtení. Poslední změřená hodnota vstupního signálu zapsaná do záznamu. Modul neprovádí žádné měření, pouze zobrazí poslední údaj uložený v paměti.

## 6.6 record

Příkazy, které pracují se záznamem změřených hodnot uložených v paměti. V případě, že jde o příkaz pro čtení záznamu, pak následuje parametr, celé číslo, určující maximální počet řádků záznamu, které jsou přečteny. Není-li parametr uveden, přečte se maximálně jeden řádek. Struktura záznamu v paměti je popsána v 9.

### 6.6.1 from

Čtení záznamu od zadaného bodu (4.3.1). Zadá-li se bod v záznamu včetně rozlišovacího čítače, přečte se záznam až od řádku následujícího, neboť se předpokládá, že byl do modulu poslán údaj převzatý z posledního již přečteného řádku. Jestliže se zadá bod bez rozlišovacího čítače, přečte se záznam od nejstaršího řádku, který vyhovuje zadanému řetězci.

#### 6.6.1.1 date

Počáteční bod pro čtení záznamu je zadán jako parametr příkazu.

- *Příklad*

Takto se přečte záznam novější než 9 hodin, 16. července 2007, maximálně 100 řádků.

```
⇒ read record 100 from date/20070716090000.0/
⇐ 20070716100000.0,1,2.8,-1.35,13.2
   20070716110000.0,1,2.2,-1.34,13.2
   20070716120000.0,1,2.2,-1.34,13.3
   OK
```

#### 6.6.1.2 start

Od začátku. Jako bod je dosazen nejstarší řádek v záznamu.

#### 6.6.1.3 last [number] read hold

Od zvoleného bodu posledního čtení (6.6.2). Bod posledního čtení zůstane beze změny.

#### 6.6.1.4 last [number] read shift

Od zvoleného bodu posledního čtení. Bod posledního čtení se posune podle počtu přečtených řádků. Tento příkaz má zvláštní chování v případě odesílání zprávy (6.14.3), kdy k posunutí dojde pouze pokud byla zpráva úspěšně odeslána.

### 6.6.2 last [number] read

Bod posledního čtení ze záznamu. Lze číst i zapisovat řetězec, označující bod v záznamu. Namísto symbolu [number] zapíšeme číslo 1 až 4, které označuje jeden ze čtyř bodů, který jsme zvolili. Nezapíšeme-li žádné číslo, je zvolen bod číslo 1.

- *Příklad*

```
⇒ write record last 1 read/200707160000/
```

### 6.6.3 space

Pouze čtení. Celé číslo. Velikost paměti pro uložení záznamu. Vyjadřuje počet znaků, které se vejdou do záznamu. Tento parametr se přibližně rovná celkovému počtu znaků, které lze přečíst příkazem pro čtení záznamu od začátku (6.6.1.2).



### 6.6.4 type

Typ záznamu do paměti. Lze jej změnit pouze pokud je paměť prázdná.

#### 6.6.4.1 cyclic

Po naplnění paměti se začnou přemazávat nejstarší data.

#### 6.6.4.2 linear

Po naplnění paměti se zápis zastaví.

### 6.6.5 status

Stav záznamu do paměti. Při čtení vrací odpověď, která obsahuje na řádku 3 pole, oddělená čárkami.

- *Příklad*

```
⇒ read record status
⇐ 59653,0.06,FILLING
   OK
```

První pole obsahuje údaj o počtu znaků uložených v záznamu, druhé pole přibližné poměrné zaplnění paměti, a třetí pole údaj o probíhající činnosti v záznamu. Údaj ve třetím poli **CLEAR** znamená, že paměť je zcela prázdná, **FILLING** částečně zaplněná, a **FULL** plná. Jakmile se v případě cyklického typu záznamu začnou vymazávat nejstarší data, je odpověď:

```
⇐ 0,1.00,FULL
   OK
```

#### 6.6.5.1 clear

Pouze zápis. Vymazání celého záznamu.

## 6.7 output

Konfigurace zvoleného výstupu.

Za tímto klíčovým slovem následuje parametr, celé číslo v rozmezí 1 až 7, určující číslo výstupu, kterého se příkaz týká.

Výstup může sloužit pro dvě funkce – pro automatické spínání napájecího napětí externího čidla při měření, nebo může být přímo ovládán příkazy. Tyto dvě funkce nelze spojit.

### 6.7.1 device

Číslo zařízení 1 až 8. Určuje zařízení, které se používá pro realizaci výstupu. Zvolené číslo zařízení musí mít přiřazen typ **MSP430ADC** nebo **MSP430TA**. Číslo zařízení **0** znamená, že k výstupu není přiřazeno žádné zařízení, a je tudíž bez funkce.

- *Příklad*

```
⇒ write output 7 device 8
```

### 6.7.2 address

Adresa, na které se nachází zvolený výstup na příslušném zařízení. Celé číslo, které u zařízení typu **MSP430ADC** určuje bit portu P6, u zařízení typu **MSP430TA** se adresa ignoruje.

Adresa 6 slouží pro spínač napájecího napětí externího senzoru. Napájení senzoru se připojí kladným pólem na pin 7 konektoru K2, a zem na pin 6 téhož konektoru.

Adresa 7 slouží pro interní spínač pro měření napájecího napětí.

Adresy 0, 1, 2, 3, 4 lze použít pokud současně není příslušný pin použit jako signálový vstup některého kanálu. Výstupy s adresami 0, 1, jsou vyvedeny na piny 5, 4 konektoru K2, výstupy s adresami 2, 3, 4 na piny 3, 1, 2 konektoru K6.

### 6.7.3 value

Přímý zápis zvolené hodnoty na výstup nebo čtení hodnoty výstupu. Zápisem hodnoty 0 se výstup nuluje, libovolnou jinou hodnotou se aktivuje do log.1. Hodnota zapsaná na výstup se stejně jako ostatní parametry modulu uchovává i po vypnutí modulu.

Zápisem řetězce **NO** se zruší hodnota přímo zapsaná na výstup, a tento výstup lze použít pro automatické spínání externího čidla. Modul jej pak řídí sám dle potřeby.

## 6.8 command

Za tímto klíčovým slovem se zadá celé číslo 1 až 4, které určuje číslo příkazu. Do modulu lze předdefinovat 4 příkazy, každému z nich se přiřadí jméno a hodnota. Číslo předdefinovaného příkazu lze použít v konfiguraci odesílaných zpráv (6.14).

Maximální počet příkazů lze zjistit:

```
⇒ read command max number
```

### 6.8.1 Verze 2.10

Lze zadat maximálně 40 příkazů.

### 6.8.2 name

Jméno příkazu, řetězec o maximální délce 20 znaků. Pojmenovaný příkaz může být vyvolán zapsáním jména.

### 6.8.3 value

Hodnota, libovolný příkaz o maximální délce 100 znaků. Je-li na konci doplněn znakem **&**, je k němu při vyvolání připojen text, který následuje za jménem, jinak se text za jménem ignoruje.

- *Příklad*

```
⇒ write command 1 name zmer
← OK
```

```
⇒ write command 1 value/read channel value/
← OK
```

```
⇒ zmer
← 2.0,-10.5,13.8,12.0
OK
```

- Příklad

```
⇒ write command 1 name ctizaznam
⇐ OK

⇒ write command 1 value/read record 4 from date &/
⇐ OK

⇒ ctizaznam200707
⇐ 20070701000000.0,1,0.0,9.5,13.8,12.0
    20070701010000.0,1,0.4,10.5,13.8,12.0
    20070701020000.0,1,2.0,12.0,14.1,12.0
    20070701030000.0,1,1.7,12.2,14.3,12.0
⇐ OK
```

#### 6.8.4 Odesílání dat ve zprávě

Při konfiguraci modulu pro odesílání zpráv (6.14) je nutné zadat příkaz, který se vykoná před každým odesláním zprávy. Odpověď na tento příkaz pak modul vloží do zprávy.

- Příklad

Takto přiřadíme zprávě číslo 1 příkaz číslo 1. Podle toho, jaká data chceme odesílat, zvolíme hodnotu příkazu číslo 1. Dále této zprávě přiřadíme adresu číslo 1, kterou rovněž musíme zadat podle svých požadavků.

```
⇒ write message 1 command 1
⇒ write message 1 address 1
```

##### 6.8.4.1 Odesílání záznamu

Pro odesílání nových dat ze záznamu ve zprávě je možné použít hodnotu příkazu dle následujícího příkladu.

- Příklad

Jestliže chceme ve zprávě odeslat pomocí e-mailu všechna data ze záznamu (6.6) od posledního odeslání až po současnost, můžeme použít příkaz

```
⇒ write address 1 type email
⇒ write command 1 value/read record 1000 from lastread shift/
```

Takto napsaný příkaz číslo 1 má zvláštní chování, je-li použit pro odeslání zprávy s komunikačním zařízením **wavecom**. V takovém případě dojde k posunutí bodu posledního čtení **record/lastread** pouze tehdy, pokud byla zpráva pomocí SMS nebo e-mailu úspěšně odeslána. Pokud se odeslání nezdařilo například při poruše sítě GSM, zůstane bod posledního čtení na původním místě. Tak je zajištěno, že následující zpráva bude o něco delší, a bude obsahovat všechna dosud neodeslaná data.

#### 6.8.4.2 Odesílání aktuálních změřených hodnot

Pro odesílání aktuální změřené hodnoty prvního kanálu je možné použít hodnotu příkazu dle následujícího příkladu.

- *Příklad*

Jestliže chceme ve zprávě odeslat pomocí SMS aktuální změřenou hodnotu prvního kanálu, můžeme použít příkaz

```
⇒ write address 1 type sms
⇒ write command 1 value/read channel 1 value/
```

#### 6.8.5 Vyvolání příkazu u modulu s vlastní adresou

Pojmenovaný příkaz lze u modulu s vlastní adresou vyvolat přes komunikační rozhraní jménem i bez zadání vlastní adresy v hlavičce příkazu (6.3.8), (8.6.1). To umožňuje povolit užití vybraných příkazů i bez znalosti vlastní adresy či hesla modulu.

### 6.9 command channel switch hold

Parametr udávající počet hodin, minut a sekund ve formátu **hhmmss**. Určuje dobu, po kterou zůstane zapnutý spínač externího čidla po vykonání příkazu pro změření aktuální hodnoty kanálu. Hodnota je společná pro všechny použité spínače. Tímto zpožděným vypnutím lze zajistit trvalé zapnutí čidla po dobu série po sobě jdoucích měření.

- *Příklad*

```
⇒ write command channel switch hold 000100
```

Nastavená doba je 1 minuta.

### 6.10 address

Zadání adresy. Za tímto klíčovým slovem se zapíše celé číslo 1 až 8, které určuje číslo zadávané adresy. Do modulu lze zadat 8 adres, a u každé z nich se zapíše typ a hodnota. Číslo zadané adresy lze použít pro určení cílové adresy odesílané zprávy (6.14). Zadané adresy lze rovněž využít pro spuštění zpráv pomocí příchozího volání.

Maximální počet adres lze zjistit příkazem

```
⇒ read address max number
```

#### 6.10.1 Verze 2.10

Lze zadat maximálně 40 adres.

#### 6.10.2 type

Typ adresy.

##### 6.10.2.1 sms

Adresa je telefonní číslo pro odeslání zprávy pomocí SMS.

### 6.10.2.2 email

Adresa je e-mailová adresa pro odeslání zprávy pomocí e-mailu.

### 6.10.2.3 ringvoice

Adresa je očekávané telefonní číslo příchozího hlasového volání. Za klíčovým slovem **ringvoice** se uvede číslo zvolené zprávy, která bude spuštěna. Jakmile sonda zaregistruje, že telefonní číslo volajícího souhlasí s adresou, spustí se zvolená zpráva.

### 6.10.2.4 ringdata

Adresa je očekávané telefonní číslo příchozího datového volání. Za klíčovým slovem **ringdata** se uvede číslo zvolené zprávy, která bude spuštěna. Jakmile sonda zaregistruje, že telefonní číslo volajícího pomocí datového volání souhlasí s adresou, zablokuje se navázání datové spojení, a místo toho se spustí zvolená zpráva.

### 6.10.3 value

Hodnota, pro typ **sms**, **ringvoice** a **ringdata** telefonní číslo, pro typ **email** e-mailová adresa.

- *Příklad*

⇒ `write address 1 type email`

⇒ `write address 1 value/MyALA@ala1.com/`

Za e-mailovou adresou lze volitelně uvést adresu smtp serveru, přes který se e-mail posílá. V tom případě se ignoruje parametr **com/network/smtpaddress**. Volitelná adresa smtp serveru je oddělena od e-mailové adresy znakem @, a zadává se ve stejném formátu jako parametr **com/network/smtpaddress** (6.11.6.3).

## 6.11 com

Parametry komunikačního rozhraní. Rozhraní RS-232 má tyto pevně nastavené parametry: délka znaku 8 bitů, 1 stop bit, bez parity.

### 6.11.1 Řízení toku dat

Ve směru od uživatele do modulu se nepoužívá žádné řízení. Modul je vždy připraven přijmout příkaz, a po dobu jeho vykonávání není dovoleno posílat další. Další příkaz je připraven přijmout, jakmile odešle odpověď. Ve směru od modulu k uživateli lze použít softwarové i hardwarové řízení. Modul reaguje na přijatý znak **XOFF** zastavením toku dat, a po přijetí znaku **XON** vysílání pokračuje. Obdobně reaguje na signál **RTS** z počítače nebo na signál **CTS** z modemu (12.7). Řízení toku dat není povinné, modul podporuje i rozhraní RS-232 bez řízení. V případě, že je pomocí řízení tok dat zastaven na dobu delší než 60 sekund, modul začne řízení ignorovat, to znamená že pokračuje ve vysílání. K opětovnému zapnutí řízení dojde, jakmile modul obdrží signál k povolení toku dat.

### 6.11.2 speed

Rychlost asynchronního přenosu dat v bitech za sekundu. Zadává se jako celé číslo. Lze nastavit hodnoty 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 a 230400.

- *Příklad*

⇒ `write com speed 9600`

### 6.11.3 max speed

Doporučená maximální rychlost asynchronního přenosu dat v bitech za sekundu. Zadává se jako celé číslo. Lze nastavit stejné hodnoty jako u parametru **com/speed**. Tento parametr lze nastavit libovolně, a nijak neovlivňuje funkci modulu. Slouží pouze pro informaci, jakou maximální rychlost lze použít v dané aplikaci například s ohledem na délku komunikačního kabelu.

### 6.11.4 transmitter mode

Režim vysílače komunikačního rozhraní RS-232.

#### 6.11.4.1 saving

Úsporný režim. Vysílač je zapnutý pouze při odesílání odpovědi na příkaz, nebo při odesílání zprávy. Při rychlosti komunikačního rozhraní vyšší než 19200 se vysílač chová shodně jako v režimu **follow**.

#### 6.11.4.2 follow

Režim sledování. Vysílač je zapnutý, pokud je komunikační rozhraní připojeno k zapnutému rozhraní RS-232 uživatele.

#### 6.11.4.3 on

Vysílač je trvale zapnutý.

### 6.11.5 device

Komunikační zařízení, připojené k modulu.

#### 6.11.5.1 local

Není použito žádné komunikační zařízení. Modul je připojen kabelem přímo k rozhraní RS-232 uživatelského počítače, nebo je použito takové zařízení, které nevyžaduje ze strany modulu žádné ovládání.

#### 6.11.5.2 wavecom

Je použit modul ALA GPRS, GSM modem pro komunikaci v síti GSM. Číslo na konci klíčového slova rozlišuje tři varianty ovládání modemu ze strany modulu.

##### 6.11.5.2.1 Vlastnosti společné všem variantám

Modul je vždy připraven přijmout příkaz SMS zprávou (6.12.1) nebo navázat datové spojení službou CSD při příchozím volání. Pokud je v příkazu přijatém pomocí SMS uvedena zpáteční e-mailová adresa (6.12.2), pak se pomocí služby GPRS připojí do internetu a odešle odpověď jako e-mail. Pokud není e-mailová adresa uvedena, je odpověď odeslána jako SMS. Pokud modul potřebuje odeslat zprávu (6.14) jako e-mail, připojí se pomocí GPRS na internet. GSM modem lze kdykoliv odpojit a modul připojit kabelem k rozhraní RS-232 počítače. Komunikace pak probíhá stejně jako u komunikačního zařízení **local**. Po opětovném připojení modemu jsou obnoveny komunikační možnosti v síti GSM.

Číslo na konci klíčového slova **wavecom** rozlišuje tři varianty chování GSM modemu, pokud je v klidovém stavu, a neprobíhá žádná komunikace.

##### 6.11.5.2.2 wavecom40

Modem je ve spícím režimu a není dostupný v GSM ani GPRS síti.

##### 6.11.5.2.3 wavecom41

Modem je připojen do sítě GSM. Není připojen do sítě GPRS a na internet.

##### 6.11.5.2.4 wavecom42

Modem je připojen do sítě GSM i GPRS, je trvale připojen na internet, poslouchá jako TCP server na portu 10001, a na tomto portu s ním lze navázat datové spojení.

- *Příklad*

⇒ `write com device wavecom41`

#### 6.11.5.3 mode

Jiný způsob nastavení funkce komunikačního zařízení **wavecom**, tedy GSM modemu, v klidovém stavu. Tento parametr nelze použít, je-li nastaveno komunikační zařízení **local**.

##### 6.11.5.3.1 saving

Ekvivalent zápisu dle 6.11.5.2.2.

##### 6.11.5.3.2 gsm

Ekvivalent zápisu dle 6.11.5.2.3.

##### 6.11.5.3.3 gprs

Ekvivalent zápisu dle 6.11.5.2.4.

### 6.11.6 network

Parametry sítí, do kterých je modul připojen přes komunikační rozhraní.

#### 6.11.6.1 smsc

SMS centrum v síti GSM. Zde je možné zadat telefonní číslo centra pro komunikaci pomocí SMS. Pokud není zadáno žádné číslo, modul použije číslo, uložené GSM operátorem na SIM kartě v GSM modemu, což většinou vyhovuje.

#### 6.11.6.2 apn address

Název přístupového bodu do internetu přes službu GPRS. Tento parametr je nutno zadat podle informací GSM operátora.

#### 6.11.6.3 smtp address

Adresa smtp serveru pro odesílání e-mailů z modulu k uživateli. E-maily se odesílají buďto jako odpovědi na příkazy uživatele nebo na základě splněných podmínek jako zprávy. Zpravidla se zadá název smtp serveru podle informací GSM operátora. Pokud je vyžadováno přístupové jméno a heslo, uvádí se před adresou smtp serveru, takže parametr pak obsahuje 3 údaje — první údaj je přístupové jméno, druhý údaj heslo a třetí údaj je adresa smtp serveru. První a druhý údaj jsou odděleny znakem :, druhý a třetí znakem @.

Pro odesílání zpráv (6.14) lze nastavit pro každou cílovou e-mailovou adresu jiný smtp server (6.10.3).

#### 6.11.6.4 pop3

Parametry e-mailové příkazové schránky.

##### 6.11.6.4.1 address

Tři přístupové údaje k e-mailové schránce, která obsahuje e-maily s příkazy pro modul. První údaj je přístupové jméno, druhý údaj je heslo a třetí údaj je adresa pop3 serveru. První a druhý údaj jsou odděleny znakem :, druhý a třetí znakem @.

- *Příklad*

Příkazem

```
⇒ write com network pop3 address/jan.novak@posta.cz:heslo@pop3.posta.cz
```

uložíme následující údaje

jméno: jan.novak@posta.cz

heslo: heslo

pop3 server: pop3.posta.cz

Vždy po odeslání každého e-mailu, tedy buďto po odeslání odpovědi na příkaz (3.4.6) nebo po odeslání zprávy (6.14) testuje modul obsah příkazové schránky. Pokud schránka obsahuje e-maily, modul stáhne nejstarší z nich, vymaže jej, a vykoná příkaz, který je v něm napsán. Případný další e-mail ze schránky bude stažen opět až po příštím odeslání e-mailu s daty.

##### 6.11.6.4.2 size

Maximální povolená délka e-mailu s příkazem. Delší e-mail je vymazán bez stažení.



#### 6.11.6.4.3 time

Maximální povolená doba stahování e-mailu s příkazem. Po jejím překročení je stahování ukončeno, e-mail je ignorován, ale není vymazán. Další pokus o jeho stažení proběhne příště.

#### 6.11.6.5 delay

Doba v sekundách, po kterou modul čeká po registraci do sítě GSM než začne využívat služeb sítě, například odesílat zprávu. Pro tento parametr většinou vyhovuje hodnota **0**, která znamená bez čekání. Čekání nastavíme, pokud to vyžadují vlastnosti sítě GSM v daném místě.

#### 6.11.7 status sim credit code

Kód, který je třeba odeslat do sítě GSM pro zjištění zbývajících kreditu předplacené SIM karty, a očekávaná odpověď sítě. Parametr se skládá ze tří částí oddělených hraničním znakem, přičemž povinná je pouze první část. Struktura parametru je následující.

Jako první znak se uvede hraniční znak, což je libovolný znak, který se v následujících třech částech nevykytuje, takže jej lze použít k jejich oddělení. Za tímto prvním hraničním znakem následuje první část parametru, což je kód odeslaný do GSM sítě pro zjištění kreditu SIM karty. První část parametru je ukončena druhým hraničním znakem. Následující druhá část parametru obsahuje řetězec, za kterým se vyhledává hodnota kreditu SIM karty v odpovědi z GSM sítě. Pokud odpověď sítě obsahuje pouze hodnotu kreditu bez předcházejícího textu, nemusí se druhá část parametru uvádět. Třetí část parametru je oddělena od druhého opět hraničním znakem, a obsahuje alternativní řetězec, pokud může mít odpověď GSM sítě dva různé texty předcházející hodnotě kreditu.

Modul zjišťuje kredit vždy, pokud proběhne komunikace přes síť GSM, například odeslání SMS nebo e-mailu, nebo příjem datového volání.

Pro zobrazení kreditu ve zvoleném měřeném kanále zadáme typ kanálu 11 (6.5.6.6).

#### 6.11.8 close

Pouze zápis. Příkaz, který uvolní komunikační rozhraní pro odesílání zpráv (6.14.9).

### 6.12 sms

Parametry pro činnost v síti GSM.

#### 6.12.1 command boundary

Řetězec o maximální délce 10 znaků, který slouží pro označení příkazu v SMS zprávě odeslané do modulu. Musí být uveden před a za příkazem v SMS zprávě. Na označený příkaz modul odešle odpověď pomocí SMS na telefonní číslo odesílatele. SMS která neobsahuje označený příkaz modul ignoruje.

#### 6.12.2 email address

V SMS zprávě může být kromě příkazu napsána zpáteční e-mailová adresa, na kterou má být odeslána odpověď. Modul rozpozná adresu podle toho, že je na začátku a na konci ohraničena známými řetězci, a na tuto adresu odešle odpověď pomocí e-mailu.

##### 6.12.2.1 start

Řetězec před e-mailovou adresou. Maximální délka je 10 znaků.

##### 6.12.2.2 end

Řetězec za e-mailovou adresou. Maximální délka je 10 znaků.

### **6.13 email send**

Parametry odesílaných e-mailů, které se použijí pro e-maily, obsahující odpověď (6.12.2), i pro e-maily obsahující zprávu (6.14).

#### **6.13.1 from**

Odesílatel e-mailu. Tento parametr musí obsahovat libovolnou platnou e-mailovou adresu.

#### **6.13.2 subject**

Předmět e-mailu o maximální délce 20 znaků.

## 6.14 message

Zpráva, kterou modul odesílá při splnění podmínek. Může být odesílána pravidelně, při zápisu nových dat do paměti, nebo při překročení zadaných změřených hodnot. Je-li použito komunikační zařízení **local** (6.11.5), je zpráva odeslána jako prostý text. Při použití komunikačního zařízení **wavecom**, tedy GSM modemu, je odeslána jako SMS nebo e-mail. Za slovem **message** následuje číslo **1** nebo **2**, které určuje číslo zprávy. Lze tedy nastavit 2 zprávy. Maximální počet zpráv lze zjistit:

⇒ `read message max number`

### 6.14.1 Verze 2.10

Lze nastavit 40 zpráv.

### 6.14.2 address

Adresa (6.10) kam bude zpráva odeslána. Zvolíme číslo adresy.

#### 6.14.2.1 Verze 2.10

Lze zadat až 10 adres pro jednu zprávu, čísla adres se oddělují čárkami.

### 6.14.3 command

Příkaz (6.8), který modul vykoná, a jehož odpověď je vložena do zprávy. Zvolíme číslo příkazu.

### 6.14.4 period

Perioda (4.3.3), se kterou je zpráva odesílána. Hodnota **000000** znamená, že je pravidelné odesílání vypnuto. Je-li perioda 1 měsíc a pro příští zprávu vyjde větší den než je počet dní v daném měsíci (například 30. února), je zpráva odeslána poslední den v měsíci.

- *Příklad*

⇒ `write message 1 period 240000`

Zpráva se odesílá každý den.

### 6.14.5 start

Začátek odesílání zprávy — datum a čas (4.3.2), od kterého se počítá perioda. Je-li zadán jako budoucí, odesílání zpráv začíná právě tímto datem. Je-li zadán jako minulý, posune se skutečný začátek automaticky o nezbytný počet period do budoucna.

### 6.14.6 of record

#### 6.14.6.1 function

Zapnutí nebo vypnutí zprávy odesílané při zápisu nových dat do záznamu.

**yes**

Zpráva se odesílá.

**no**

Zpráva se neodesílá.

- *Příklad*

⇒ `write message of record function yes`

### 6.14.7 of channel

Zpráva odesílaná při překročení zadané změřené hodnoty v určeném kanále.

#### 6.14.7.1 function

##### no

Zpráva se neodesílá.

##### up

Zpráva se odesílá při překročení zadané hodnoty směrem nahoru.

##### down

Při překročení směrem dolů.

- Příklad

⇒ `write message of channel function up`

#### 6.14.7.2 number

Číslo určeného kanálu.

#### 6.14.7.3 value

Zadaná změřená hodnota, jejíž překročení se testuje.

#### 6.14.7.4 hysteresis

Hystereze. Určuje rozdílný způsob testování zadané hodnoty při pohybu změřené hodnoty směrem nahoru a směrem dolů. Je-li předchozí změřená hodnota pod zadanou hodnotou, modul při testování přičte k zadané hodnotě polovinu hystereze, a je-li nad ní, odečte polovinu hystereze. Tím se zamezí častému překračování zadané hodnoty nahoru a dolů, pokud se změřená hodnota pohybuje v její blízkosti, a tím se zabrání častému odesílání zprávy.

#### 6.14.8 run

Zapsáním čísla 1 za toto klíčové slovo zahájíme odesílání zprávy, i když nebyly splněny podmínky. Toho lze využít například pro testování zprávy.

#### 6.14.9 Blokování zprávy

Jestliže jsou splněny podmínky pro odeslání zprávy, ale komunikační rozhraní modulu je zaměstnáno jinou činností, je zpráva pozdržena. Rozhraní může být zaměstnáno buďto komunikací s připojeným GSM modemem nebo komunikací s počítačem připojeným přes kabel.

##### 6.14.9.1 GSM modem

Zpráva je blokována například tehdy, je-li k modulu jako k serveru právě připojen přes internet klient nebo při čtení příchozích SMS zpráv a odesílání odpovědí. Jakmile vnější aktivita skončí, modul zahájí odesílání zprávy.

##### 6.14.9.2 Počítač připojený přes kabel

Zpráva je blokována od okamžiku, kdy modul přijme první příkaz z počítače. Důvod je ten, aby nedocházelo k rušení komunikace odesílanými zprávami. K odblokování dojde buďto odpojením a opětovným připojením komunikačního kabelu nebo příkazem `write com close` (6.11.8).

## 6.15 file

Parametry čtyř oblastí paměti. U každé oblasti se nastavuje jediný parametr, velikost. Velikost musí být zadána jako celistvý násobek 256. Tyto parametry musíme nastavit, pokud chceme do modulu zapisovat převodní tabulky (6.5.12), (10), (6.16).

### 6.15.1 record size

Velikost záznamu. Velikost oblasti záznamu je nutno zapsat jako první v pořadí ze všech oblastí. Při zápisu se automaticky nastaví velikost dvou následujících oblastí rovna nule.

- *Příklad*

```
⇒ write file record size 983040
```

### 6.15.2 list size

Velikost seznamu souborů. Tento parametr je nutno zapsat jako druhý v pořadí. Při zápisu se automaticky nastaví velikost následující oblasti rovná nule.

- *Příklad*

```
⇒ write file list size 256
```

### 6.15.3 data size

Velikost dat souborů. Tento parametr je nutno zapsat jako třetí, poslední v pořadí.

- *Příklad*

```
⇒ write file data size 51200
```

### 6.15.4 free size

Pouze čtení. Velikost nepoužité oblasti.

- *Příklad*

```
⇒ read file free size  
← 14080  
OK
```

## 6.16 special

Čtení a zápis parametrů speciálním komunikačním protokolem. Slouží pro ukládání převodních tabulek pro přepočítání změřených hodnot kanálů (6.5.12) a pro upgrade programu modulu. Jako komunikační protokol se používá Xmodem.

### 6.16.1 Zápis seznamu tabulek

Délka seznamu menší nebo rovna parametru *file/list/size*, tedy musí se vejít do vyhrazené oblasti.

Zadáme modulu příkaz

```
⇒ write special file list xmodem
```

a pomocí protokolu X-modem odešleme do modulu soubor obsahující seznam tabulek.

Čtení oblasti provedeme příkazem

```
⇒ read special file list xmodem
```

a pomocí protokolu X-modem přijmeme z modulu soubor.

### 6.16.2 Zápis dat tabulek

Délka dat musí být menší nebo rovna parametru *file/data/size*.

Zadáme modulu příkaz

```
⇒ write special file data xmodem
```

a pomocí protokolu X-modem odešleme do modulu soubor obsahující datový soubor vzniklý spojením všech tabulek.

Čtení oblasti provedeme příkazem

```
⇒ read special file data xmodem
```

### 6.16.3 Čtení souboru record

Pomocí protokolu Xmodem lze přečíst i oblast paměti, kde je uložen záznam. Tento způsob čtení je možno použít jako nouzovou záchranu dat, pokud jsme vymazali záznam některým z příkazů, které záznam mažou, například **write global reset** (6.17.1), a proto běžné příkazy pro čtení záznamu (6.6) ukazují prázdný záznam. Je nutno si ale uvědomit, že není zaručena záchrana všech dat, a pokud je nastaven cyklický způsob záznamu, není začátek dat na začátku oblasti.

Čtení provedeme příkazem

```
⇒ read special file record xmodem
```

Do oblasti paměti, kde je uložen záznam, lze pomocí protokolu Xmodem i zapisovat, ale taková činnost se nedoporučuje, protože dojde k neobnovitelnému poškození dat v záznamu.

### 6.16.4 Upgrade programu modulu

Zadáme modulu příkaz

```
⇒ write special program xmodem
```

a pomocí protokolu X-modem odešleme do modulu soubor obsahující upgrade programu. Po vykonání příkazu se vrátí hodnoty všech parametrů do výchozích výrobních hodnot a vymaže se záznam. Výjimku tvoří tři parametry, u kterých se neprovede návrat k výrobním hodnotám z důvodu, aby po upgradu nedošlo k přerušení komunikace. Jsou to parametry **com/speed**, **com/device** a **com/transmitter mode** (6.11).

## 6.17 global

### 6.17.1 reset

Vrácení hodnot všech parametrů do výchozích výrobních hodnot a vymazání záznamu. Výjimku tvoří tři parametry, u kterých se neprovede návrat k výrobním hodnotám z důvodu, aby nedošlo k přerušení komunikace. Jsou to parametry **com/speed**, **com/device** a **com/transmitter mode** (6.11).

Tento příkaz použijeme, chceme-li začít konfigurovat modul z definovaného výchozího stavu (11).

### 6.17.2 timing

Nastavení základního časování modulu. Tento parametr má význam v součinnosti s nastaveným měřicím zařízením pro kanál a s typem kanálu. Pokud měříme frekvenci pomocí zařízení **MSP430ADC** (6.5.6.2.1), pak maximální měřitelná frekvence závisí na základním časování modulu. Lze nastavit 3 hodnoty časování: **8**, **64** a **4096**. Hodnota časování znamená, kolikrát za sekundu modul vzorkuje stav vstupního signálu při měření frekvence. Z toho vyplývají maximální měřitelné frekvence v pořadí 4, 32 a 2048 Hz. S rostoucí hodnotou časování se zvětšuje klidový odběr proudu modulu z napájecího zdroje.

### 6.18 nothing

Prázdný příkaz. Nečte ani nezapisuje žádný parametr modulu.

- *Příklad*

```
⇒ read nothing
← OK
```

### 6.19 text

Tento příkaz nečte ani nemění žádný parametr modulu. Modul vrátí v odpovědi parametr příkazu – řetězec – uvedený za klíčovým slovem. S přepínačem **read** modul vrátí v odpovědi parametr příkazu – řetězec uvedený za klíčovým slovem. S přepínačem **write** se parametr příkazu – řetězec – uloží do záznamu jako řádek typu 3, nebo lze typ řádku v rozmezí 1 až 9 zvolit jako parametr za klíčovým slovem **text**, před řetězcem. (9).

- *Příklad*

```
⇒ read text/Dobry den./
← Dobry den.
OK
```

## 7 Zřetězení příkazu

Za prvním příkazem mohou následovat na stejném řádku další příkazy. Celková maximální délka zřetězeného příkazu je 120 znaků.

Výjimkou je příkaz pro návrat všech parametrů k výchozím hodnotám, **write global reset**. Za ním již nemohou na řádku další příkazy následovat.

### 7.1 Zřetězení celého příkazu

V místě, kde by normálně následovalo ukončení řádku příkazu, se zapíše další příkaz.

- *Příklad*

```
⇒ readtext/Zde je zmerena hodnota prvnio kanalu/readchannel1value
← Zde je zmerena hodnota prvnio kanalu
  18.52
  OK
```

### 7.2 Zřetězení příkazu za prvním klíčovým slovem

Klíčová slova, která následují jako první za prepínačem **read** nebo **write** umožňují zřetězení příkazů se stejným klíčovým slovem. Znamená to, že za prvním příkazem mohou následovat na stejném řádku další příkazy, u kterých se první, tedy stejné klíčové slovo vynechává.

- *Příklad*

```
⇒ number1 read channel 1 name 1 value
← 1,Teplota vzduchu
  2,24.2
  3,OK
```



## 8 Hlavička příkazu

Hlavička umožňuje zadat:

1. kontrolu správnosti přijatého příkazu
2. kontrolu správnosti přijaté odpovědi
3. způsob odpovědi na příkaz
4. adresaci – volbu modulu, pro který je příkaz určen

Hlavička nemá žádný vliv na funkci příkazu, ta je určena jeho tělem.

### 8.1 Struktura hlavičky příkazu

Hlavička se skládá z klíčových slov hlavičky příkazu, z nichž za některými může následovat parametr. Pořadí klíčových slov může být libovolné.

### 8.2 Klíčová slova hlavičky příkazu

check  
 crcsum  
 sum  
 pause  
 comspeed  
 quiet  
 startline  
 endline  
 number  
 counter  
 date  
 iaddress

### 8.3 Kontrola správnosti přijatého příkazu

check

#### 8.3.1 check

Příkaz je opatřen kontrolním součtem. Za klíčovým slovem následuje parametr, celé číslo. Modul toto číslo přečte, a počínaje následujícím znakem sečte ASCII hodnoty všech znaků až do konce příkazu jako kladná čísla. Pokud se součet rovná přečtenému číslu, pokračuje se ve zpracovávání příkazu. V opačném případě je příkaz tiše odmítnut, to znamená, že modul nepošle žádnou odpověď, ani řádek **ERROR**.

- *Příklad*

⇒ check 890 read date

### 8.4 Kontrola správnosti přijaté odpovědi

crcsum  
 sum

### 8.4.1 crcsum

Kontrolní součet odpovědi. Modul opatří odpověď kontrolním součtem CRC-32 dle standardu POSIX. Kontrolní součet zahrnuje všechny znaky v odpovědi s výjimkou posledního řádku **OK**, a výsledek kontrolního součtu je uveden na samostatném řádku, před řádkem **OK**.

- *Příklad*

```
⇒  crcsum read record 6 from start
⇐  20050501070000.0,1,2.95,15.6,12.6
    20050501073000.0,1,2.96,18.3,12.6
    20050501080000.0,1,2.98,19.7,12.6
    20050501083000.0,1,3.09,20.1,12.6
    20050501090000.0,1,3.12,20.2,12.6
    20050501093000.0,1,3.13,20.2,12.6
    1139834219,210
    OK
```

### 8.4.2 sum

Kontrolní součet každého řádku odpovědi. Modul sečte hodnoty všech znaků v řádku odpovědi v kódování ASCII jako kladná čísla, a na začátek řádku umístí výsledek zakončený čárkou, doplněný zleva nulami na pětimístné číslo.

- *Příklad*

```
⇒  sum read date
⇐  00713,20070618111806
    00154,OK
```

## 8.5 Způsob odpovědi na příkaz

pause  
comspeed  
quiet  
startline  
endline  
iname  
number  
counter  
date

### 8.5.1 pause

Časová prodleva před vykonáním příkazu. Následuje parametr, celé číslo, které určuje počet sekund, po které modul čeká před vykonáním příkazu. Zadané číslo musí být menší než 3600, lze tedy nastavit dobu kratší než jedna hodina.

### 8.5.2 comspeed

Přechodné přepnutí rychlosti na sériovém komunikačním rozhraní po dobu odpovědi na tento příkaz. Následuje parametr, celé číslo, které určuje rychlost v bitech za sekundu, na které modul pošle odpověď. Lze zvolit hodnoty 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 a 230400.

- Příklad

```

⇒ pause 10 comspeed 230400 crcsum read record 6 from start
⇐ 20050501070000.0,1,2.95,15.6,12.6
   20050501073000.0,1,2.96,18.3,12.6
   20050501080000.0,1,2.98,19.7,12.6
   20050501083000.0,1,3.09,20.1,12.6
   20050501090000.0,1,3.12,20.2,12.6
   20050501093000.0,1,3.13,20.2,12.6
   1139834219,210
   OK

```

Tento příklad ukazuje, jak přechodně přepnout modul na vyšší rychlost na sériovém portu pro čtení záznamu. Časová prodleva 10 sekund daná klíčovým slovem **pause** umožní pohodlně přepnout komunikační program na vyšší rychlost před přijetím odpovědi. Po odeslání řádku **OK** se modul vrátí zpět k původní rychlosti.

### 8.5.3 quiet

Tichá odpověď. Po vykonání příkazu je v odpovědi potlačen řádek **OK**, a také je potlačeno zopakování chybného příkazu následované řádkem **ERROR**.

### 8.5.4 startline

Za tímto slovem následuje parametr, který bude uveden na začátku každého řádku odpovědi. Maximální délka je 20 znaků.

### 8.5.5 endline

Za tímto slovem následuje parametr, který bude uveden na konci každého řádku odpovědi. Maximální délka je 10 znaků.

- Příklad

```

⇒ quiet startline/Teplota: /endline/ stupnu/ read channel 2 value
⇐ Teplota: 10.5 stupnu

```

Takto lze doplnit změřenou hodnotu doprovodným textem před a za zobrazenou hodnotou.

- Příklad

```

⇒ quiet endline/*/ read i description
⇐ Stanoviste 1 *

```

Na tomto příkladě je ukázána možnost zobrazit zřejmě omylem zapsané mezery na konci označení modulu.

### 8.5.6 iname

Na začátku každého řádku odpovědi bude uveden řetězec **i/name** (6.3.5).

### 8.5.7 counter

Za klíčovým slovem následuje parametr, celé číslo **1** až **4**, které označuje zvolený čítač. Řádky odpovědi budou číslovány pomocí jednoho ze čtyř čítačů, který si uchovává hodnotu mezi jednotlivými odpověďmi. Tak lze zkontrolovat, zda byla přijata všechna data z odesílaných zpráv (6.14). Po zapnutí modulu je hodnota čítače 0, a inkrementuje se po odeslání každého řádku.

### 8.5.8 number

Jednotlivé řádky odpovědi budou číslovány od zadaného čísla. Tak lze zkontrolovat, zda byla přijata celá odpověď. Za klíčovým slovem následuje parametr, celé číslo od kterého začíná číslování řádků odpovědi.

- *Příklad*

```
⇒ number1readchannel2value
⇐ 1,10.5
   2,0K
```

### 8.5.9 date

Jednotlivé řádky odpovědi budou na začátku opatřeny datem, odděleným čárkou.

- *Příklad*

```
⇒ date read channel 2 value
⇐ 20060701090532,10.5
   20060701090532,0K
```

## 8.6 Adresace modulu pro který je příkaz určen

iaddress

### 8.6.1 iaddress

Za tímto klíčovým slovem následuje parametr o neznámé délce – adresa modulu (6.3.8). Na takový příkaz odpoví pouze modul, jehož adresa – parametr **i/address** – je totožná s parametrem zadaným v hlavičce. Toto klíčové slovo se použije v případě, že je na komunikační linku připojeno více modulů. Pak není při zadávání příkazu vhodné zapínat zpětné zobrazení znaků, neboť znaky vracejí nazpět všechny moduly, připojené na linku, což může způsobit kolizi.

Další možností je použít tento parametr k zabezpečení přístupu k modulu, tedy jako heslo.

## 9 Záznam

Záznam změřených hodnot se skládá z řádků. Každý řádek se skládá z několika polí, oddělených čárkami. Formát řádku je následující:

**YYYYMMDDhhmmss.c,i,ch1,ch2,ch3,...,chn**

V poli

**YYYYMMDDhhmmss.c**

označuje **YYYY** rok, **MM** měsíc, **DD** den, **hh** hodinu, **mm** minutu a **ss** sekundu, kdy byl řádek zapsán do záznamu. **c** je rozlišující čítač, který rozlišuje řádky zapsané ve stejné sekundě. Podle tohoto pole se zadává bod v záznamu (4.3.1) v příkazech pro čtení záznamu (6.6).

Pole

**i**

označuje typ řádku. Řádky, které obsahují změřené hodnoty kanálů dle nastavených vzorkovacích period, jsou vždy typu **1**. Do záznamu lze kromě toho zapisovat pomocí příkazů uvedených v 6.5.8 a 6.19, u kterých lze zvolit typ řádku v rozmezí 1 až 9.

U řádků změřených hodnot kanálů dle nastavených vzorkovacích period pole

**ch1**

obsahuje změřenou hodnotu prvního kanálu ve formátu dle 6.5.7. Obdobně všechna následující pole obsahují postupně změřené hodnoty dalších kanálů. Pokud nejsou v daném řádku přítomny změřené hodnoty ze všech kanálů, ale jen z některých, jsou pole neuvedených kanálů prázdná.

- *Příklad*

Příkazem (6.6.1.1)

⇒ `readrecord100fromdate*2005071609*`

přečteme z modulu, který obsahuje pět kanálů, například tato data:

⇐ `200507160940.0,1,2.8,-1.3,5.23,8.18,13.2`  
`200507161000.0,1,2.2,, ,8.20,`  
`200507162030.0,1,, , ,13.3`  
`OK`

## 10 Převodní tabulky

Převodní tabulky umožňují modulu provádět pokročilý převod měřených vstupních signálů na výsledné hodnoty.

Jednorozměrná převodní tabulka slouží k převodu jednoho měřeného vstupního signálu na jednu výslednou hodnotu, dvojrozměrná převodní tabulka slouží k převodu dvou měřených vstupních signálů na jednu výslednou hodnotu.

Počet a velikost tabulek jsou omezeny pouze velikostí paměti pro záznam, ve které se pro tabulky vyhradí místo, a zmenší se tím skutečné místo pro záznam změřených hodnot.

Převodní tabulky jsou textové soubory. Zapisují se do modulu standardním komunikačním protokolem X-modem.

### 10.1 Rozdělení paměti modulu

Aby bylo možno do modulu zapisovat tabulky, je nutno provést rozdělení paměti na 3 oblasti.

Jsou to:

#### 1. Záznam

Oblast pro ukládání záznamu změřených hodnot. Tato oblast je povinná, a slouží k funkci dataloggeru. Výrobní nastavení modulu je takové, že tato oblast zabírá celou paměť.

#### 2. Seznam souborů

Oblast pro uložení seznamu souborů. Zde se uloží jmenný seznam a délky jednotlivých tabulek.

#### 3. Data souborů

Oblast pro uložení dat souborů. Zde se uloží za sebou data jednotlivých tabulek.

Čtvrtá oblast, **Volná**, je nevyužitá část paměti, pokud je součet velikostí předchozích tří oblastí menší než velikost paměti modulu. Velikost každé oblasti je celistvým násobkem 256. Příkazy a parametry pro rozdělení paměti jsou popsány v 6.15. Příkazem dle 6.6.3 zjistíme celkovou velikost paměti modulu. Je to součet velikostí všech čtyř oblastí.

```
⇒ read record space
⇐ 1048576
OK
```

### 10.2 Formát jednorozměrné tabulky

Jednorozměrná tabulka slouží pro převod jednoho vstupního signálu na jednu výslednou hodnotu.

Tabulka se skládá z řádků. Na každém řádku jsou dvě čísla. První číslo znamená změřenou hodnotu vstupního signálu, druhé číslo znamená k němu příslušnou výslednou hodnotu. Tabulka musí být sestavena tak, aby první čísla na řádku, tedy vstupní změřené hodnoty, za sebou následovaly po řádcích vzestupně.

Modul počítá výslednou hodnotu interpolací mezi dvěma sousedními hodnotami. Vstupním hodnotám nižším než je na prvním řádku je přiřazena konstantní výsledná hodnota dle prvního řádku. Vstupním hodnotám vyšším než je na posledním řádku je přiřazena konstantní výsledná hodnota dle posledního řádku.

#### • Příklad

```
100.8 550.9
351.65 1000
893 1790.53
```

V tomto příkladě, bude-li mít vstupní signál hodnotu v rozmezí 100.8 až 351.65, bude výsledná hodnota vypočtena lineární interpolací mezi hodnotami 550.9 a 1000. Při vstupním signálu v rozmezí 351.65 až 893 bude výsledná hodnota interpolována mezi 1000 a 1790.53.

Všechny vstupní signály menší než 100.8 budou převedeny na výslednou hodnotu 550.9 a signály větší než 893 na výslednou hodnotu 1790.53.

### 10.3 Formát dvojrozměrné tabulky

Dvojrozměrná tabulka slouží pro převod dvou vstupních signálů na jednu výslednou hodnotu. Převodní tabulka používá hodnotu druhého signálu jako parametr, a podle prvního signálu se provádí lineární interpolace na výslednou hodnotu.

Tabulka se skládá z dvou typů řádků. Řádky obsahující jedno číslo jsou řádky druhého signálu jako parametru. Každým takovým řádkem začíná oblast platnosti hodnoty parametru. V řádcích obsahujících dvě čísla je první číslo na řádku hodnota prvního vstupního signálu a druhé číslo k němu příslušná výsledná hodnota v oblasti platnosti parametru. Tabulka musí být sestavena tak, aby vstupní hodnoty druhého signálu jako parametru i vstupní hodnoty prvního signálu jako prvního čísla na řádku, za sebou následovaly po řádcích vzestupně.

Modul spočítá výslednou hodnotu celkem třemi interpolacemi. Nejprve najde dvě po sobě jdoucí oblasti platnosti parametru, mezi kterými se nachází druhý vstupní signál. Pak provede dvě interpolace – spočítá hodnotu interpolací mezi dvěma sousedními hodnotami prvního vstupního signálu pro obě oblasti. Nakonec provede třetí interpolaci mezi oběma hodnotami parametru podle druhého vstupního signálu. Vstupním hodnotám prvního signálu nižším než je na prvním řádku v oblasti platnosti hodnoty parametru je přiřazena konstantní výsledná hodnota dle tohoto prvního řádku. Vstupním hodnotám prvního signálu vyšším než je na posledním řádku v oblasti platnosti hodnoty parametru je přiřazena konstantní výsledná hodnota dle tohoto posledního řádku.

- *Příklad*

```

3
100.8 550.9
351.65 1000
893 1790.53
4
100.7 560
351 1008
890 1800.2

```

V tomto příkladě, bude-li mít druhý vstupní signál hodnotu mezi 3 a 4, například 3.2, platí pro výpočet první interpolace 3 řádky následující za řádkem s hodnotou parametru 3. Při prvním vstupním signálu v rozmezí 100.8 až 351.65 bude první lineární interpolace provedena mezi hodnotami 550.9 a 1000. Při prvním vstupním signálu v rozmezí 351.65 až 893 bude interpolace provedena mezi 1000 a 1790.53. Pro první vstupní signál menší než 100.8 bude výsledek první interpolace konstantní hodnota 550.9 a pro první vstupní signál větší než 893 bude výsledek konstantní hodnota 1790.53.

Druhá interpolace se provede obdobně pro 3 řádky následující za řádkem s hodnotou parametru 4.

Třetí interpolace se provede dle hodnoty druhého vstupního signálu, tedy dle hodnoty 3.2 mezi hodnotami parametrů 3 a 4.

Pro hodnoty druhého vstupního signálu menší než 3 se výsledná hodnota počítá jedinou interpolací dle prvního vstupního signálu v oblasti platnosti parametru s hodnotou 3.

Pro hodnoty druhého vstupního signálu větší než 4 se výsledná hodnota počítá jedinou interpolací dle prvního vstupního signálu v oblasti platnosti parametru s hodnotou 4.

## 10.4 Příprava dat pro zápis tabulek do modulu

Nejprve připravíme všechny potřebné tabulky jako textové soubory na počítači.

Pak vytvoříme seznam tabulek. To je textový soubor, který obsahuje seznam tabulek, které se budou zapisovat do modulu. Tento seznam se skládá z řádků. Na každém řádku jsou 2 údaje. První údaj je jméno tabulky, druhý údaj je délka tabulky. Jméno tabulky může mít maximálně 10 znaků. Nakonec vytvoříme datový soubor spojením všech tabulek v pořadí, v jakém jsme je uvedli v seznamu.

- *Příklad*

Jednorozměrnou tabulku pojmenujeme **tabulka1**, dvojrozměrnou tabulku **tabulka2**. Seznam bude vypadat následovně:

```
tabulka1 36  
tabulka2 70
```

Datový soubor bude:

```
100.8 550.9  
351.65 1000  
893 1790.53  
3  
100.8 550.9  
351.65 1000  
893 1790.53  
4  
100.7 560  
351 1008  
890 1800.2
```

### 10.4.1 Zápis tabulek do modulu

Před zápisem tabulek je nutno zastavit pravidelné měření kanálů, které používají tabulky, jinak může dojít v průběhu zápisu tabulek k chybnému výpočtu změřené hodnoty kanálu. Zápis tabulek je popsán v 6.16.



## 11 Seznam parametrů modulu a jejich počáteční hodnoty z výroby

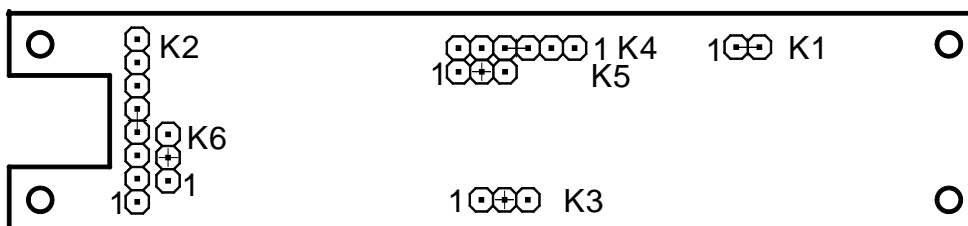
Parametr	Hodnota nebo (popis)
i/ version device snumber program name description type address	(verze příkazů a parametrů) (verze hardwaru) (výrobní číslo) (verze programu) (prázdný řetězec) (výrobní číslo) (výrobní číslo) (prázdný řetězec)
date	20000101000000
device {1,2,...n}/ type cal/ mult/ add/	MSP430ADC  (zkalibrováno pro vnitřní referenční napětí) (zkalibrováno pro vnitřní referenční napětí)
channel {1,2,...n}/ max number name device address type format value sample cal/ mult/ value format add/ value format cal2/ mult add switch/ output advance convert/ file name sampling/ period start average samples record/ change samples value	1 Supply[V] 1 21 1 0.1 (napájecí napětí) (prázdný řetězec)  1 0.0  0 0.0  1.000000 0.000000  1 1  (prázdný řetězec)  000000 20000101000000 0  0.0 0 (prázdný řetězec)
record/ from/ date start lastread/ hold shift lastread space type status	(prázdný řetězec) (prázdný řetězec)   (prázdný řetězec) (prázdný řetězec) (prázdný řetězec) (celková velikost paměti) CYCLIC 0,0.00,CLEAR

Parametr	Hodnota nebo (popis)
output 1/ device address value output {2,3,...n}/ device address value	1 7 NO 0 0 NO
command {1,2,...n}/ name value command channel switch hold	(prázdný řetězec) (prázdný řetězec) 000005
address {1,2,...n}/ type value	SMS (prázdný řetězec)
com/ speed max speed transmitter mode device network/ smc apn address smtp address pop3/ address size time delay status sim credit code	9600 230400 SAVING LOCAL (prázdný řetězec) (prázdný řetězec) (prázdný řetězec) (prázdný řetězec) 10000 120 0 (prázdný řetězec)
sms/ command boundary email address/ start end email send/ from subject	##*# startemail endemail (prázdný řetězec) (prázdný řetězec)
message {1,2,...n}/ address command period start of record/ of channel/ function function number value hysteresis	1 1 000000 20000101000000 NO NO 1 0.0 0.0
file/ record size list size data size free size	(celková velikost paměti) 0 0 0
special file/ record xmodem list size data size	(soubor s velikostí rovnou celkové velikosti paměti) (soubor s velikostí 0) (soubor s velikostí 0)
global/ timing	8

## 12 Popis konektorů

### 12.1 Rozmístění konektorů

Na následujícím obrázku je rozmístění konektorů na modulu.



### 12.2 Konektor K1, napájení

Slouží k připojení napájecího napětí. Napájecí napětí musí být v rozmezí 3.5 až 28.0 V. Typický proudový odběr modulu je 20  $\mu$ A.

Pin K1	Popis
1	Vstup napájecího napětí +3.5 až +28.0 V
2	Zem

### 12.3 Konektor K2, připojení měřených signálů nebo výstupy

Napětí přivedené na piny konektoru K2 proti zemi nesmí překročit povolené meze. Rovněž je nutno se vyvarovat připojení indukovaného elektrostatického napětí. Správná funkce modulu je zaručena v pracovním rozsahu napětí.

Pin K2	Popis	Pracovní rozsah napětí	Mezní rozsah napětí
1	Vstup signálu čítače	0 až 3.3 V	-0.3 až 3.6 V
2	Vstup analogového signálu nebo vnějšího referenčního napětí	Jako vstup signálu: 0 až 2 V Jako vstup referenčního napětí: 0 až 2.5 V	-0.3 až 3.6 V
3	Společná zem pro signálové vstupy a pro vstup vnějšího referenčního napětí	-	-
4,5	Vstup analogového signálu nebo teploměru DS18B20	Při vnitřním referenčním napětí: 0 až 2 V Při vnějším referenčním napětí: 0 až referenční napětí	-0.3 až 3.6 V
6	Výkonová zem pro napájení externích senzorů	-	-
7	Výstup spínače napájení externích senzorů	Nelze připojit vstupní napětí	Nelze připojit vstupní napětí
8	Trvalý výstup napájecího napětí	Nelze připojit vstupní napětí	Nelze připojit vstupní napětí

## 12.4 Konektor K6, připojení dalších měřených signálů nebo výstupy

Napětí přivedené na piny konektoru K6 proti zemi nesmí překročit povolené meze. Rovněž je nutno se vyvarovat připojení indukovaného elektrostatického napětí. Správná funkce modulu je zaručena v pracovním rozsahu napětí. Jako společná zem pro signály slouží pin 3 konektoru K2.

Pin K6	Popis	Pracovní rozsah napětí	Mezní rozsah napětí
1,2,3	Vstupy analogového signálu nebo teploměru DS18B20	Při vnitřním referenčním napětí: 0 až 2 V Při vnějším referenčním napětí: 0 až referenční napětí	-0.3 až 3.6 V

## 12.5 Konektor K3, komunikační rozhraní RS-232

Slouží pro připojení modulu k rozhraní RS-232 řídicího počítače nebo modulu ALA GPRS. Pro komunikaci se používají 3 signály – RX, TX a zem. Volitelně je možno použít vstup pro hardwarové řízení toku dat na konektoru K5.

Pin K3	Popis
1	Zem
2	RX, vstup
3	TX, výstup

## 12.6 Konektor K4, nepoužitý

Konektor K4 je v současné době bez funkce.

## 12.7 Konektor K5, doplňkový signál pro komunikační rozhraní RS-232

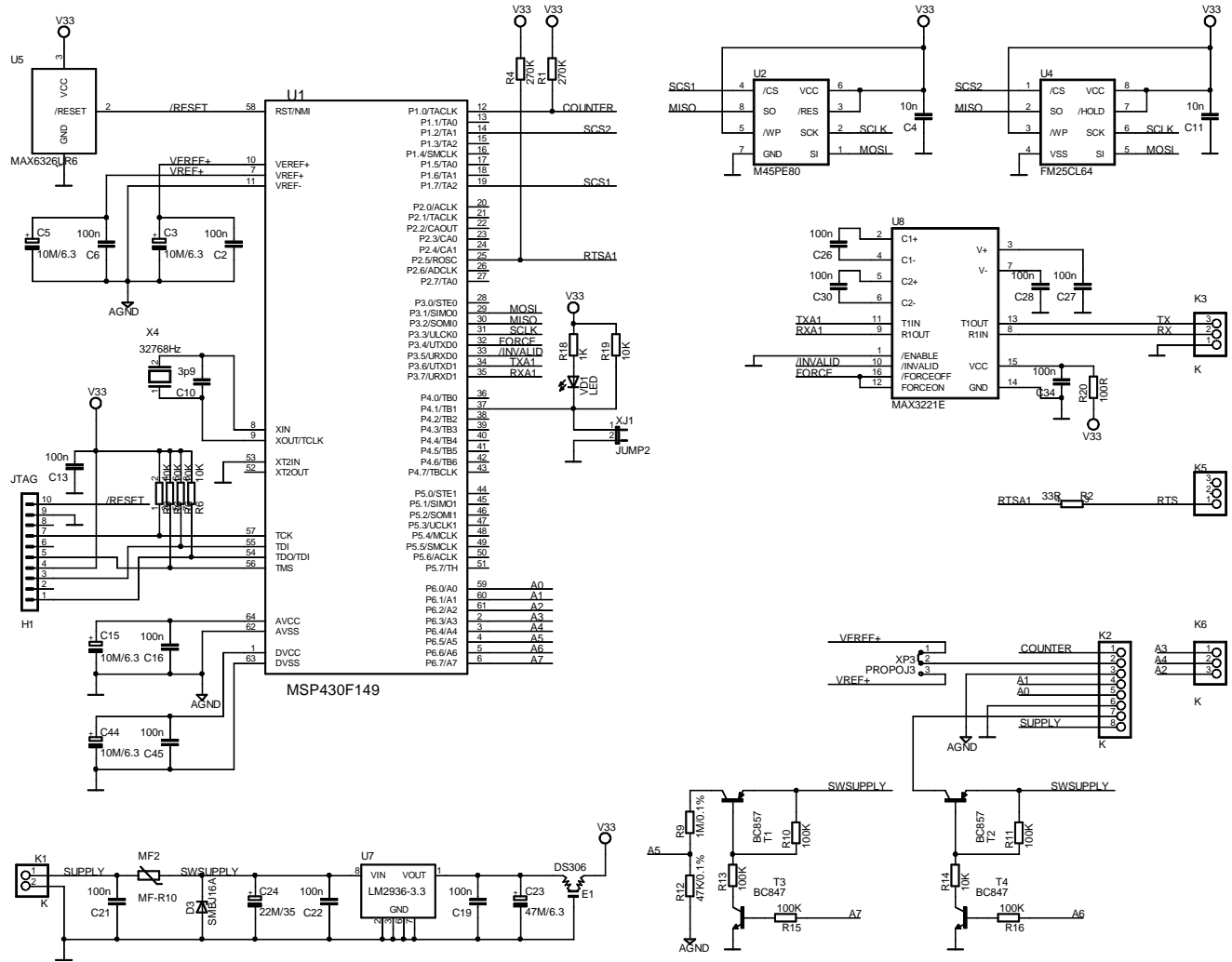
Tento konektor obsahuje signály s napěťovou úrovní CMOS 3.3 V, mezní rozsah napětí je -0.3 až 3.6 V. Pro připojení k lince RS-232 je nutno použít vhodný převodník.

Pin K5	Popis
1	RTS, vstup. Volitelné hardwarové řízení toku dat pro komunikační rozhraní. Ve stavu bez připojení nebo logické 1 je tok dat povolen, ve stavu logické 0 zakázán.

Tento signál se připojí přes vhodný převodník úrovní k signálu RTS na rozhraní RS-232 počítače nebo k signálu CTS na rozhraní RS-232 modulu ALA GPRS.

Poznámka: Logické úrovně signálu RTS jsou oproti standardu RS-232 invertovány.

# 13 Schéma zapojení



TITLE: ALA1SchemaU2	
Document Number:	REV: D
Date: 14.06.2007 09:15:33	Sheet: 1/1