



## Měřič tepla a chladu, vyhodnocovací jednotka průtoku plynu

INMAT 59

## Popis komunikačních protokolů

typ 459

## Obsah:

1. Fyzická vrstva .....	1
2. Implementace M-Bus+ (protokol ZPA) .....	1
2.1. Linková vrstva .....	2
2.2. Aplikační vrstva .....	4
2.2.1. Použité datové typy .....	4
2.2.2. CI-Field codes .....	5
2.2.3. SubCode .....	5
3. Implementace MODBUS RTU v INMATu .....	17
3.1. Linková vrstva .....	17
3.2. Aplikační vrstva .....	17
3.3. Příklady komunikace MODBUS-RTU .....	19
4. Implementace M-Bus v INMATu .....	21
5. Ladění komunikace protokolem M- Bus+ .....	22

## 1. Fyzická vrstva

Využívá pro komunikaci rozhraní M-Bus, RS485, RS232, optická hlava.

INMAT 57S a INMAT 57D využívá současně komunikační M-Bus+ (protokol ZPA) vycházející z M-Bus protokolu, dále protokol M-Bus a protokol MODBUS-RTU. Detekce protokolu v INMATu je automatická. INMAT odpovídá protokolem dotazu. **Vzhledem k autodetekci nelze při MODBUS-RTU dotazu používat adresy 16 (0x10) a 104 (0x68).**

## 2. Implementace M-Bus+ (protokol ZPA)

**Základní popis protokolu:** Aplikační vrstva implementuje rychlý bezstavový protokol ZPA (M-Bus+), jehož základem je **CI-Field code** a **SubCode** podobně jako v protokolu M-Bus. Na rozdíl od protokolu M-Bus protokol ZPA neuchovává stavové informace z předchozí komunikace, což umožňuje současný přenos dat do více master zařízení prostřednictvím síťové brány.

**CI-Field code** (1Byte) pro INMATy se nachází v rezervované oblasti adres 0xC0 – 0xFF a slouží pro základní volbu zpracovávaných dat (sumy, maxima, archivace, ...).

**SubCode** (4Byte) následuje za CI-Field code a slouží pro bližší určení dat. Určuje, jaká data se budou do INMATu zapisovat nebo jaká data se budou z INMATu číst. Udržuje navíc pomocnou informaci umožňující čtení většího množství dat, než kolik se vejde do jednoho telegramu. Každý požadavek na čtení nebo zápis dat musí obsahovat tento čtyřbajtový SubCode. O tom, jestli se jedná o čtení nebo o zápis dat rozhoduje šestý bit v bajtu **C** pole na linkové vrstvě. SubCode může obsahovat adresu, bitové příznaky nebo parametr, kterým se blíže určí požadovaná nebo zapisovaná data. Jsou-li požadovány například sumy ve formátu longWord, single, double nebo extended float, nastaví se parametr v SubCode na příslušnou hodnotu. INMAT v odpovědi vrátí pole float dat ve standardním formátu **IEEE 754** (nebo 4Byte longWord). Zapisuje-li se suma, nastaví se tytéž příznaky v SubCode přičemž v jeho spodním bajtu se odešle index právě zapisované sumy.

INMAT na požádání umí poslat i sumy ve stejném rozlišení jako na displeji (s ořezanými vyššími řády **trimmed single float, trimmed double float**) splňující normu **IEEE 754**. Nedochozí tak ke ztrátě desetinných řádů, vzroste-li velikost sumy nad rozsah zobrazení single float čísla viz. kapitola Použité datové typy.

K polím dat **Inmat poskytuje i příslušné jednotkové textové řetězce** nastaví-li se příznak pro čtení řetězců, čímž se získají jednotky ke každé položce v poli. Celková délka těchto textových řetězců je obvykle delší než 246Byte, proto INMAT si v odpovědi připraví následující SubCode pro následující požadavek. Tento nenulový SubCode z odpovědi se beze změny odešle zpátky do INMATu, aby INMAT mohl vrátit následující blok dat s následujícím předpřipraveným SubCode. Jestliže INMAT spolu s daty vrátí nulový SubCode, explicitně tím ukončuje další čtení dat. Oddělovačem jednotlivých polí textových řetězců je znak LF (0x0A) pro snadné zpracování v TMemoryStream. Data se v cílovém zařízení zřetěží a dále zpracují (jako například jednotky do nějaké tabulky v aplikaci).

Data z archivace a bilancí se čtou podobným způsobem jako řetězce, protože se obvykle jedná o velké množství dat. Na začátku se do SubCode nastaví bitové příznaky a pak se data čtou tak dlouho, dokud není SubCode nulový. Pro zpracování načtených dat v PC lze opět s výhodou použít TMemoryStream bez použití oddělovače záznamů.

**Velkou výhodou přístroje je možnost číst archivovaná data inkrementálně.**

Dotaz pro čtení **Archivace, Bilancí a logu událostí**, může obsahovat (4Byte) časovou známku naposledy čtených dat. Je tedy možné číst data pouze jednou za určité období (například jen jednou za týden přečíst všechny přístroje). Přístroj tedy nevrací celý obsah paměti, ale vrátí pouze data, která přibyla od posledního čtení (někdy před týdnem). Načtená data lze přímo ukládat do databáze a tak je řetězit, protože je zachována časová monotonie. **Několik uživatelů může naráz číst stejná data** z jednoho INMATu s různě nastaveným rozsahem požadovaného období **aniž by o sobě věděli**, protože informace o požadovaném období se přenáší v každém dotazu.

Vnitřní **paměti INMATu jsou zapisovány cyklicky**, tedy nejstarší zaznamenaná data se přepisují nejčerstvějšími daty. Je tedy nutné data z INMATu pravidelně číst (například 1x za týden), aby nedošlo ke ztrátě kontinuity dat. Vyčtená **data jsou časově setříděná**, INMAT tedy nejdříve posílá nejstarší data, naposledy pošle nejčerstvější. Jak už bylo zmíněno, vyčtená data lze snadno zřetězit a postupně získat velkou databázi hodnot i za velmi dlouhé období aniž by se data z jednotlivých přístrojů četla například každou minutu. Uložená data jsou ze zcela přesných časových intervalů na základě nastavení INMATu.

**Maximální délka odesílaného telegramu je nastavitelná.** Pro MODBUS 256Byte, pro M-Bus 262Byte a M-Bus+ až 2056Byte. Pro případ použití bezdrátově komunikace prostřednictvím IQRF nebo ZigBee lze délku telegramu zkrátit na 128Byte. Dokonalejší detekci chyb pro velké telegramy lze zajistit použitím parity.

Obdrží-li INMAT porušený telegram, neodpoví. Obdrží-li INMAT správně utvořený telegram, avšak s nesprávným SubCode nebo s nesprávným množstvím dat, vrátí lidsky čitelnou chybovou zprávu s předřazeným chybovým kódem usnadňujícím strojové zpracování.

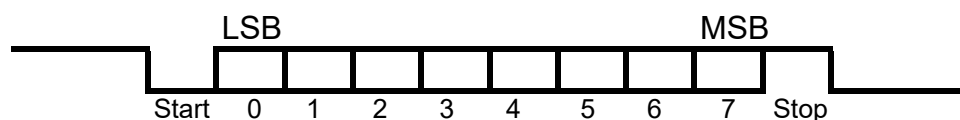
## 2.1. Linková vrstva

Formát telegramu vychází z M-Bus ČSN EN 1434-3, viz. <http://www.m-bus.com/mbusdoc/md5.php>

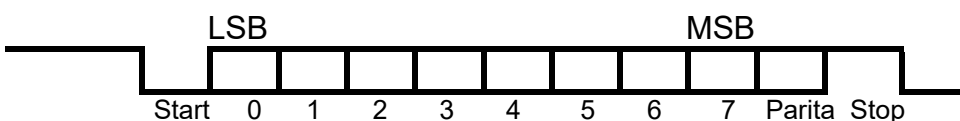
### Znak telegramu

Znak má délku 10, nebo 11 bitů. Začíná start bitem, následuje 8 datových bitů, parita (je-li zvolena), stop bit.

Bez parity



S paritou



### Formáty telegramů:

- Telegram s pevnou délkou bez datového pole (pouze pro vyhledávání přístrojů)

Dotaz : 

SD1	C	A	CS	ED
-----	---	---	----	----

Odpověď: 

ACK
-----

- Telegram s proměnnou délkou datového pole

Dotaz nebo odpověď:

SD2	LE	LEr	SD2	C	A	CI	SUBCODE	DATA	CS	ED
-----	----	-----	-----	---	---	----	---------	------	----	----

Odpověď: 

ACK
-----

V případě chyby přístroj odpoví chybovým hlášením přímo v textové podobě v datové části telegramu. Chybové hlášení přístroj odesílá v přednastaveném jazyce a v přednastavené znakové sadě (Windows-1250, Windows-1251, KOI8-RU, ISO8859-1, ISO8859-2, UTF-8, ASCII).

### Význam použitých zkratk:

**SD1** – začátek rámce (Start Delimiter), kód 0x10  
**SD2** – začátek rámce (Start Delimiter), kód 0x68  
**LE** – délka informačního pole (LEngth)  
**LEr** – opakování bajtu délky informačního pole  
**C** – řídicí bajt (Control field)  
**A** – adresa cílové stanice (Address)  
**CI** – řídicí informační bajt (Control Information field)  
**SUBCODE** – řídicí informace pro přenášená data (4Byte)  
**DATA** – přenášená data  
**CS** – (Check Sum) kontrolní součet (1Byte)  
**ED** – konec rámce (End Delimiter), hodnota 0x16  
**ACK** – kladné potvrzení 0xE5

**LE, LEr – délka informačního pole**

Obsahuje počet přenášených bajtů 7 (C + A + CI + SubCode) + DATA. Minimální délka celého informačního pole je 7Byte.

Pro telegram delší jak 255Byte se pro počet přenášených bajtů využívají navíc spodní bity C

Příklady:

Pro směr *Master* → *INMAT* se využívají spodní 4 bity C.

SD2	LE	LEr	SD2	C	délka inf. pole
0x68	0xFF	0xFF	0x68	0x40	... 255Byte
0x68	0x00	0x00	0x68	0x41	... 256Byte
0x68	0x00	0x00	0x68	0x42	... 512Byte
0x68	0x01	0x01	0x68	0x42	... 513Byte
0x68	0x02	0x02	0x68	0x42	... 514Byte
0x68	0xFF	0xFF	0x68	0x4F	... 4095Byte

Pro směr *INMAT* → *Master* se využívají spodní 3 bity C.

SD2	LE	LEr	SD2	C	délka inf. pole
0x68	0x06	0x06	0x68	0x08	... 6Byte
0x68	0xFF	0xFF	0x68	0x08	... 255Byte
0x68	0x00	0x00	0x68	0x09	... 256Byte
0x68	0xFF	0xFF	0x68	0x0F	... 2047Byte

**A – adresa stanice**

Adresa je v rozsahu 0 – 255 přičemž adresy 0, 251 – 255 jsou speciální adresy.

Adresa 0 je určena pro nově připojený přístroj (dosud bez přiřazené adresy).

Adresy 254 a 255 jsou vyhrazeny pro broadcast.

**C – řídicí bajt**

U telegramu s proměnlivou délkou (telegram 2) INMAT tento bajt používá pro odlišení požadavku pro zápis nebo čtení dat. Ostatní bity jsou rezervované.

Dotaz:

0x60, – čtení dat (alternativně 0xE0 při přítomnosti ProfiBus zařízení na lince)  
0x40, – zápis dat (alternativně 0xC0 při přítomnosti ProfiBus zařízení na lince)

Odpověď:

0x08 nebo alternativně 0x88 – při dotazech 0xE0, 0xC0

Krátký telegram (telegram 1):

0x40 (alternativně 0xC0 při přítomnosti ProfiBus zařízení na lince)

**CI** – INMAT protokol **M-Bus+** využívá rezervovanou oblast (0xC0 – 0xFF) viz. popis aplikační vrstvy.

**CS – (Check Sum) kontrolní součet**

Kontrolní součet je dán aritmetickým součtem dat **C, A, CI, SUBCODE, DATA** bez integrace přenosu.

**Chybové kódy**

V případě chyby vrátí INMAT CI-Field code 0x70 následovaný chybovým kódem. Obvykle je připojen i textový řetězec dle předvoleného jazyka a kódování.

**Kód chyby, popis chyby**

0x00	MBUS_UNSPECIFIED,	
0x01	MBUS_UNIMPLEMENTED_CI,	
0x02	MBUS_BUFFER_TOO_LONG,	
0x03	MBUS_TOO_MANY_RECORDS,	
0x04	MBUS_PREMATURE_END_OF_RECORDS,	
0x05	MBUS_MORE_THAN_10DIFE,	
0x06	MBUS_MORE_THAN_10VIFE,	
0x07	MBUS_RESERVED,	
0x08	MBUS_APPLICATION_TOO_BUSY,	// zopakujte požadavek později
0x09	MBUS_TOO_MANY_READOUTS,	
0x0A	ERR_ACCESS_DENIED_CIPHER,	// firmware je určeno pro jiný přístroj
0x0B	ERR_ACCESS_DENIED_JUMPER,	// přístup blokován propojkou
0x0C	ERR_ACCESS_DENIED_METRO,	// přístup blokován metrologickým heslem
0x0D	ERR_ACCESS_DENIED,	// přístup blokován heslem
0x0E	ERR_ACCESS_DENIED_TIMEOUT,	// přístup na 3 minuty blokován

## 2.2. Aplikační vrstva

### 2.2.1. Použité datové typy

**longWord** (unsigned 4Byte) hodnoty provozního času a času mimo parametry, chybové slovo v případě sum, okamžitých hodnot,... vrací hodnotu vynásobenou **100**.

**trimmed longWord** (unsigned 4Byte) – z pohledu PLC se jedná o standardní longWord formát, který byl v INMATu před odesláním vynásoben **100** a byly oříznuty vyšší řády, viz příklad.

**float** (IEEE 754) pro okamžité hodnoty, konstanty atd.

Sumy lze číst a zapisovat (jen při „Security no“) v **extended** (10Byte), **double** (8Byte) a **single** (4Byte) float. Bilance lze v přednastaveném datovém typu pouze číst.

**trimmed single float, trimmed double float** - z pohledu PLC se jedná o standardní float, ze kterého byly v INMATu odstraněny vyšší řády. Ořez je určen formátem čísla na displeji. Nedochozí tak ke ztrátě rozlišení hodnoty po „přetočení“ hodnoty na displeji, viz příklad.

Příklad sum v INMATu:

Hodnota v paměti INMATu		123456789.123456789		
Formát na displeji INMATu		456789.12		
Formát		INMAT vrací hodnotu	Formát v PLC	
longWord * 100	4B	345678912	/100 = 3456789,12	longWord/single/...
single float	4B	123456784.		single float
double float	8B	123456789.123456776		double float
extended float	10B	123456789.123456789		extended float
trimmed longWord * 100	4B	45678912	/100 = 456789.12	longWord/single/...
trimmed single float	4B	456789.12		single float
trimmed double float	8B	456789.123456788948		double float

**pktime** je komprimovaný formát času o velikosti 4Byte obsahující rok, měsíc, den, hodinu, minutu a vteřinu. Vychází ze starého MS DOS Date Time Format posunutý o jeden bit doleva.

| YYYY YYMM | MMDD DDDh | hhhh mmmmm | mmss ssss |

```
typedef struct {
    unsigned sec:6;           // 0-59
    unsigned min:6;          // 0-59
    unsigned hour:5;         // 0-23
    unsigned day:5;          // 1-31
    unsigned month:4;        // 1-12
    unsigned year:6;         // 0-63 (2000-2063)
} pktimeT;
```

## 2.2.2. CI-Field codes

0xC0	XADCONFIG	AD převodník (R/W)
0xC1	XAPPLIC	nevyužívá se
0xC2	<b>XARCHIVEBLOCK1</b>	blok 1 archivace (pouze 57S, 57D) (R/W)
0xC3	<b>XARCHIVEBLOCK2</b>	blok 2 archivace (pouze 57S, 57D) (R/W)
0xC4	<b>XARCHIVEBLOCK3</b>	blok 3 archivace (pouze 57S, 57D) (R/W)
0xC5	<b>XARCHIVEBLOCK4</b>	blok 4 archivace (pouze 57S, 57D) (R/W)
0xC6	<b>XARCHIVECFG</b>	konfigurace archivace + řetězce (pouze 57S, 57D) (R/W)
0xC7	<b>XBALANCE</b>	bilance single/double/extended float, longWord (pouze 57S, 57D), (R/W)
0xC8	<b>XCOMMUNICATION</b>	nastavení komunikace (R/W)
0xC9	<b>XCONFIG</b>	hlavní konfigurace přístroje (R/W)
0xCA	<b>XCONST</b>	konstanty, názvy konstant (R)
0xCB	XDIAGNOSTIC	diagnostika, časové známky chybových zprav přístroje, které se kdy vyskytly, řetězce diagnostiky a data (R/W)
0xCC	<b>XDISPLAY</b>	obsahy displejů provoz/servis/konfig, nastavení menu uživatel (R/W)
0xCD	<b>XERRORS</b>	chybové zprávy, chybové slovo, chybový status, není-li žádná čerstvá zpráva, vrací ACK (R/W)
0xCE	<b>XGASARCHIVE</b>	archiv plynu, čtení, zápis změn, mazání archivu, (pouze 57S, 57D) (R/W)
0xCF	XIMPOUT	impulsní výstup, nastavení impulsu při chybě nebo při nějakém přírůstku sumy (R/W)
0xD0	<b>XINMAT</b>	identifikační řetězce přístroje, použitá výstupní znaková sada a název přístroje (R/W)
0xD1	XIOUT	proudový výstup, kalibrace a nastavení proudového výstupu (R/W)
0xD2	<b>XMAXIMA</b>	vteřinové špičky, minutové průměrované špičky, čtvrt hodinová maxima, nulování maxim, názvy (R/W)
0xD3	<b>XPASSWD</b>	odemyká přístroj, nastavuje uživatelské nebo systémové heslo (W)
0xD4	XETHERNET,	nastavení ethernetu (pouze Inmat 59) (R/W)
0xD5	<b>XSUM</b>	sumy ve formátu single/double/extended float (R/W)
0xD6	<b>XTIME</b>	RTC, časové známky doby provozu a náhradních parametru (R/W)
0xD7	XUPDATER	odtud lze vyčíst/zapsat zašifrovaný FW z přístroje, který se pošle zákazníkovi, aby si mohl sám provést update (pouze 57S, 57D) (R/W)
0xD8	<b>XUSRSUM</b>	uživatelské sumy ve formátu single/double/extended float (R/W)
0xD9	<b>XVARIABLES</b>	proměnné, názvy (R)
0xDA	XWORKLOAD	časové známky z časovače pro výpočet zátěže v klientském SW, aktuální čas (pouze 57S, 57D), (R)

## 2.2.3. SubCode

Tyto příznaky blíže určují požadavek dotazu – rozšiřují „specializované SubCode parametry“.

### SubCode - Obecné bitové příznaky

DEFAULT = 0x00000000; implicitní akce

### SubCode - Formát dat

Popis viz výše.

formát	or SubCode	
0	LONG_INTEGER	0x00000000 4 Byte
1	SINGLE_FLOAT	0x01000000 4 Byte
2	DOUBLE_FLOAT	0x02000000 8 Byte
3	EXTENDED_FLOAT	0x03000000 10 Byte
4	TRIMMED_LONG_INTEGER	0x04000000 4 Byte
5	TRIMMED_SINGLE_FLOAT	0x05000000 4 Byte
6	TRIMMED_DOUBLE_FLOAT	0x06000000 8 Byte
8	STRINGS	0x80000000 Textové řetězce oddělené LF (0x0A)

### SubCode - Kódování zapisovaného textu

zapisuje-li se text, INMATu se musí dát vědět v jakém kódování je text zaslán, aby ho mohl překódovat

Používá se v **XINMAT**, **XARCHIVECFG**

WINDOWS 1250	0x00000000 (DEFAULT)
WINDOWS 1251	0x01000000
ISO 8859 1	0x02000000
ISO 8859 2	0x03000000
KOI8 R	0x04000000

### SubCode - Inkrement do cyklu

SUBCODE\_INCREMENT = 0x01000000

**SubCode - speciální parametry**

Níže uvedený seznam SubCode je společný pro INMAT 57, 57D, 59. Pokud není některá funkce v konkrétním přístroji implementována, vrací INMAT **CI-Field code 0x70** a číslo chyby **0x34** „Neznámá hodnota SubCode“

**XVARIABLES (CI = 0xD9)**

	C	SubCode		Struktura dat
			Dostupný formát	
Systémové proměnné	R	0x00000000	1	pkTime + všechny proměnné dle zvoleného formátu
			8	Názvy proměnných jako strings
Pomocné proměnné	R	0x20000000	1	pkTime + všechny proměnné dle zvoleného formátu
			8	Názvy proměnných jako strings
Okamžité proměnné	R	0x40000000	1	pkTime + všechny proměnné dle zvoleného formátu
			8	Názvy proměnných jako strings

**XSUM (CI = 0xD5)**

	C	SubCode		Struktura dat
			Dostupný formát	
Sumy	R	0x00000000	0..7 (Inmat 57) 3 (Inmat59)	pkTime + všechny sumy dle zvoleného formátu
			8	Názvy sum jako strings
		0x84000000	-	bytové pole s počty celých řádů u jednotlivých sum
	W	0x000000nn	0..7 (Inmat 57) 3 (Inmat 59)	Zápis jedné sumy dle zvoleného formátu. nn = pořadí sumy ve skupině [0..x]. Pouze u „Security: no“ Úspěšné uložení potvrdí INMAT - ACK

**XUSRSUM (CI = 0xD8)**

	C	SubCode		Struktura dat
			Dostupný formát	
Uživatelské sumy	R	0x00000000	0..7 (Inmat 57) 3 (Inmat 59)	pkTime + všechny uživatelské sumy dle zvoleného formátu
			8	Názvy uživatelských sum jako strings
		0x84000000	-	bytové pole s počty celých řádů u jednotlivých sum
	W	0x000000nn	0..7 (Inmat 57) 3 (Inmat 59)	Zápis jedné uživatelské sumy dle zvoleného formátu. nn = pořadí uživatelské sumy ve skupině. [0..x]. Úspěšné uložení potvrdí INMAT - ACK

**XCONST (CI = 0xCA)**

	C	SubCode		Struktura dat
			Dostupný formát	
Pevné konstanty	R	0x00000000	1	pkTime + všechny konstanty dle zvoleného formátu
			8	Názvy konstant jako strings
Metrologické konstanty	R	0x20000000	1	pkTime + všechny konstanty dle zvoleného formátu
	W		8	Názvy konstant jako strings
Uživatelské konstanty	R	0x40000000	1	Všechny uživatelské konstanty dle zvoleného formátu Úspěšné uložení potvrdí INMAT – ACK *)
	W		1	Všechny uživatelské konstanty dle zvoleného formátu Úspěšné uložení potvrdí INMAT - ACK

Pozn : Struktura Pevných a Uživatelských konstant je shodná pro obě skupiny. Každá konstanta je definována jako uživatelská, nebo pevná. Není-li ve skupině definována, vrací hodnotu "NAN" Je-li uživatelská konstanta „NAN“ nelze ji po lince změnit.

\*) Nastavení je možné pouze pokud je INMAT ve stavu „Security No“, případně ve stavu „Security Yes“ a současně „Sumarizace No“. Nastavením současně dojde k vynulování sum a bilancí. Více viz.: Návod k výrobku.

Doplňující informace dostupné k metrologickým a uživatelským konstantám:

CONST\_MAX\_LIMIT 0x08000000 Seznam maximálně možných hodnot (single float seznam)  
CONST\_MIN\_LIMIT 0x04000000 Seznam minimálně možných hodnot (single float seznam)  
*v případě nedefinované meze vrací „NAN“*

Příklad: Metrologické konstanty, Maximální meze konstant, single float  
SubCode → 0x29000000 ← 0x20000000 or 0x08000000 or 0x01000000

CONST\_CONDITIONS 0x02000000 text podmínky mezi CONST\_MIX\_LIMIT a CONST\_MAX\_LIMIT  
INMAT\_HINTS 0x01000000 textová nápověda „Hints“ ke konstantám  
*string seznamy, položky oddělené LF*

Příklad: Metrologické konstanty, nápověda významu konstant „Hints“  
SubCode → 0xA1000000 ← 0x80000000 or 0x20000000 or 0x01000000

### Příklad: (XVARIABLES, XSUM, XUSRSUM, XCONST)

**SubCode**

Dotaz > **na názvy sum**  
0x68 0x07 0x07 0x68 0xE0 0x00 0xD5 0x00 0x00 0x00 0x80 0x35 0x16  
Odpověď > data - strings oddělené LF 0x0A  
0x68 0x25 0x25 0x68 0x88 0x00 0xD5 0x00 0x00 0x00 0x00 0x45 0x31 0x20 0x20 0x20 0x5B 0x47 0x4A 0x5D  
0x0A 0x4D 0x31 0x20 0x20 0x20 0x20 0x5B 0x74 0x5D 0x0A (0x56 0x31 0x20 0x20 0x20 0x5B 0x6D 0x33  
0x5D) 0x0A 0x03 0x16  
ASCII >  
h%h .Ń...E1 [GJ].M1 [t].V1 [m3]...

Dotaz > **na vlastní hodnoty, formát extended float**  
0x68 0x07 0x07 0x68 0xE0 0x00 0xD5 0x00 0x00 0x00 0x03 0xB8 0x16  
Odpověď > čas vyčtení-pkTime 3x extended  
0x68 0x29 0x29 0x68 0x88 0x00 0xD5 0x00 0x00 0x00 0x00 0x7A 0x72 0x96 0x31 0xF5 0xA6 0x5B 0xF3 0xA3  
0xA2 0x79 0xEB 0x19 0x40 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00  
0x00 0x00 0x00 0x00 0x00 0xFB 0x16

Dotaz > **na vlastní hodnoty, formát single float**  
0x68 0x07 0x07 0x68 0xE0 0x00 0xD5 0x00 0x00 0x00 0x01 0xB6 0x16  
Odpověď > čas vyčtení-pkTime 3x single  
0x68 0x17 0x17 0x68 0x88 0x00 0xD5 0x00 0x00 0x00 0x00 0x91 0x80 0x96 0x31 0xA2 0x79 0xEB 0x4C 0x00  
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x87 0x16

### Doporučené formáty dat:

Sumy : trimmed single float  
Okamžité hodnoty : single float

**XMAXIMA (CI = 0xD2)**

	C	SubCode	Struktura dat
Čas posledního nulování	R	0x00000000*)	pkTime
Vynulování maxim	W	0x00000000	Úspěšné vynulování potvrdí INMAT - ACK
¼ maxima	R	0x21000000*)	pkTime, všechny maxima single float, časy dosažení pkTime
		0xA0000000	Názvy a jednotky ¼h maxim
		0xE0000000	Popis sloupců datového pole ¼h maxim
Minutové a vteřinové špičky	R	0x19000000*)	pkTime, minutové maxima single float, maxima, časy dosažení pkTime
		0x98000000	Názvy a jednotky minutových a sekundových špiček
		0xC8000000	Popis sloupců datového pole minutových špiček
		0xD0000000	Popis sloupců datového pole vteřinových špiček

\*) Doplnující informace:

Aktuální           0x00000000  
 Předchozí        0x04000000

Aktuální (v tomto období) dosažené maxima a špičky  
 Předchozí (uzavřené období) dosažené maxima a špičky  
 Funkce „předchozí“ dostupná od dubna 2013 u 57D

**Příklad: (XMAXIMA)****SubCode**

Dotaz &gt; čas posledního nulování

```
0x68 0x07 0x07 0x68 0xE0 0x00 0xD2 0x00 0x00 0x00 0x00 0xB2 0x16
Odpověď > čas nulování-pkTime
0x68 0x0B 0x0B 0x68 0x88 0x00 0xD2 0x00 0x00 0x00 0x00 0x61 0x83 0x96 0x31 0x05 0x16
```

Dotaz &gt; dotaz na maxima, formát single float

```
0x68 0x07 0x07 0x68 0xE0 0x00 0xD2 0x00 0x00 0x00 0x21 0xD3 0x16
Odpověď > čas vyčtení-pkTime data
0x68 0x1B 0x1B 0x68 0x88 0x00 0xD2 0x00 0x00 0x00 0x00 0x9B 0x82 0x96 0x31 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x8A 0xD0 0x8C 0x31 0x8A 0xD0 0x8C 0x31 0x6C 0x16
```

**Data = 2x4B ¼h maximum , 2x4B čas dosažení**

Dotaz &gt; dotaz na minutové a vteřinové špičky, formát single float

```
0x68 0x07 0x07 0x68 0xE0 0x00 0xD2 0x00 0x00 0x00 0x19 0xCB 0x16
Odpověď > čas vyčtení-pkTime data
0x68 0x5B 0x5B 0x68 0x88 0x00 0xD2 0x00 0x00 0x00 0x00 0x9B 0x82 0x96 0x31 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0xB7 0x67 0xAF 0x43 0x62 0x1D 0xEC 0x41 0xE7 0xBC 0xA0 0x43 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xB7 0x67 0xAF 0x43 0x0E 0xA7 0xED 0x41 0xAF 0xC6 0xA0
0x43 0x8A 0xD0 0x8C 0x31 0x8A 0xD0 0x8C 0x31 0x8B 0xD0 0x8C 0x31 0xF7 0xD0 0x90 0x31 0x80 0x53 0x91
0x31 0x4A 0xD0 0x8C 0x31 0x4A 0xD0 0x8C 0x31 0x4C 0xD0 0x8C 0x31 0xAD 0x05 0x95 0x31 0xDF 0xEA 0x8E
0x31 0xB5 0x16
```

**Data = 5 x 4B minutová špička, 5 x 4B vteřinová špička, 5 x 4B čas dosažení-minutová špička, 5 x 4B čas dosažení-vteřinová špička**



**XTIME (CI = 0xD6)**

	C	SubCode	Struktura dat
RTC	R	0x00000000	Aktuální čas INMATu (pkTime)
		0x40000000	Čas posledního nastavení času
		0x80000000	Popis položky RTC
	W	0x00000000	Úspěšné nastavení potvrdí INMAT – ACK
Provozní časy	R	0x20000000	Pole LongWord [sec] dle aplikace [runtime, wet steam, ...]
		0xA0000000	Popis položek časů
	W	0x20000000	Zápis LongWord [sec] podle počtů časů v konkrétním INMATu. Pouze u „Security: no“. Úspěšné uložení potvrdí INMAT - ACK

**Příklad: Nastavení času pomocí Broadcast adresy 255:**

Nastavované INMATy musí mít stejné nastavení linky a heslo. V případě INMATů bez hesla, se odblokování nemusí provádět.

**SubCode**

Dotaz > **Odblokování hesla zápisem platného hesla (odblokuje pouze INMATy se stejným heslem)**  
 0x68 0x0B 0x0B 0x68 0x40 0xFF 0xD3 0x00 0x00 0x00 0x00 0x34 0x34 0x34 0x34 0xE2 0x16  
 odpověď >

Dotaz > **nastavení RTC času do všech INMATů** 13.12.2012 08:19:11  
 0x68 0x0B 0x0B 0x68 0x40 0xFF 0xD6 0x00 0x00 0x00 0x00 0xCB 0x84 0x1A 0x33 0xB1 0x16  
 odpověď >

**XDISPLAY (CI = 0xCC)**

	C	SubCode	Výčet displejů	Struktura dat
Menu uživatel 1	R	0x10000000		Nastavení pořadí displejů menu uživatel 1-XOR seznam
	W			XOR seznam. Úspěšné nastavení potvrdí INMAT - ACK
Menu uživatel 2	R	0x20000000	**)	Nastavení pořadí displejů menu uživatel 2-XOR seznam
	W			XOR seznam. Úspěšné nastavení potvrdí INMAT ACK
Menu uživatel 3	R	0x30000000		Nastavení pořadí displejů menu uživatel 3-XOR seznam
	W			XOR seznam. Úspěšné nastavení potvrdí INMAT - ACK
Menu uživatel (starší verze FW)	R	0x00000000	***)	Nastavení pořadí displejů menu uživatel - Bytové pole
	W			Bytové pole. Úspěšné nastavení potvrdí INMAT - ACK
Provoz	R	0xC0000000 *)	****)	Menu Provoz - řádky oddělené LF <i>mimo INMATu 59</i>
Konfig		0xD0000000 *)	****)	Menu Konfig - řádky oddělené LF <i>mimo INMATu 59</i>
Servis		0xE0000000 *)	****)	Menu Servis - řádky oddělené LF <i>mimo INMATu 59</i>

\*) lze vyčítat pouze horní (první) řádek displeje

FIRST\_DISPLAY\_LINE = 0x04000000;

oddělovače řádků displeje, každý (celý) displej je ale vždy oddělen znakem LF (0x0A)

LINE\_FEED\_DELIMITER = 0x00000000 oddělovač 0x0A (LF)

SPACE\_DELIMITER = 0x02000000 oddělovač 0x20 (mezera)

SEMICOLON\_DELIMITER = 0x01000000 oddělovač 0x3B (středník)

Příklad: Displej provoz, jen první řádek, oddělovač 0x20

SubCode → 0xC6000000 ← 0xC0000000 or 0x04000000 or 0x02000000

Provoz or FIRST\_DISPLAY\_LINE or 0x02000000

\*\*)) XOR provázaný seznam pořadí vybraných položek 0..255[x]. První byte XOR poslední byte zprávy udává index první položky seznamu.

\*\*\*)) Bytové pole pořadí vybraných položek 0..127[prev, next]

\*\*\*\*)) Pokud se za SubCode odešle výčet displejů, INMAT odešle zpět jen požadované řádky displeje v pořadí dle zasláního výčtu. (První řádek = 0x01)

**Příklad:****SubCode Seznam výčtu displejů**

Dotaz > **Požadavek na celé displeje 8, 1, 6, 2 z menu Provoz, oddělené znakem LF**  
 0x68 0x0B 0x0B 0x68 0x60 0x00 0xCC 0x00 0x00 0x00 0x00 0x08 0x01 0x06 0x02 0x0B 0x16  
 odpověď >

**XERRORS (CI = 0xCD)**

	C	SubCode	Struktura dat
Chybové slovo	R	0x80000000	Aktivní chyby v INMATu (string oddělené LF)
	W	0x00000000	Smaže seznam chyb v INMATu. Při úspěšném smazání -ACK
	R	0x10000000	Aktuální chybové slovo (LongWord)
		0x20000000	Read Only Flags (LongWord)
		0x40000000	Bitové příznaky vyhodnocení chyb *)
		0xC0000000	Seznam dostupných hlášení – (string oddělené LF)
	0x90000000	Název proměnné „err word[4B]”– string	

\*) Bitové příznaky vyhodnocení chybových hlášení

Blik \*\*\*\*\*X (mimo INMAT 59)

Hold \*\*\*\*\*X\*

Msg \*\*\*\*\*X\*\*

**XINMAT (CI = 0xD0)**

	C	SubCode	Struktura dat
INFO DATA	R	0x08000000	Interní informace počet displejů, typ displeje atd...
Výrobní číslo		0x80000000	Výrobní číslo (string)
IDENTIFICATION HEADER		0xA0000000	Popis identifikační hlavičky (string oddělený LF)
IDENTIFICATION_DATA		0xB0000000	Data identifikační hlavičky (string oddělený LF)
SOFTWARE_DETAILS		0xB8000000	Detail verze sestavení (string oddělený LF)
APPLIC_DESCRIPTION	W	0x90000000	Popis aplikace (Zobrazení/Identifikace) Zápis pouze při „Security no“. Úspěšné nastavení - ACK
Název měřeného okruhu	R	0x98000000	String dva řádky oddělené LF <i>mimo INMATu 59</i>
	W	or Kódování	Úspěšné nastavení ACK <i>mimo INMATu 59</i>
INPUT_CHARSET_LIST		0x88000000	Seznam dostupných kodování v INMATu (string oddělený LF) <i>mimo INMATu 59</i>

**XETHERNET (CI = 0xD4)** (pouze INMAT 59)

	C	SubCode	Struktura dat
ETHERNET_CONFIG	R	0x00000000	Parametry Ethernetu
	W		Úspěšné nastavení ACK

**t\_ethernet**

```

ip_address: longword;
subnet_mask: longword;
gateway: longword;
dns1: longword;
dns2: longword;
netbios: array[0..15] of byte;
mac: array[0..5] of byte;
flags: word;
end;
```

**XCOMMUNICATION (CI = 0xC8)**

	C	SubCode	Struktura dat
MBUS_LINE	R	0x00000000	Vyčtení rychlosti, adresy, parity,... <b>TUsart</b> nebo <b>TUsart57d</b>
	W		Nastavení rychlosti, adresy, parity... Úspěšné nastavení - ACK
SPEED_LIST	R	0x80000000	Seznam dostupných přenosových rychlostí (string oddělený LF)
xBUS_LIST	R	0xC0000000	Textový seznam dostupných proměnných pro seznamy MBUS_LIST_x a MODBUS_LIST_x (string oddělený LF)
MBUS_LIST_A	R	0x40000000	M-Bus telegram linky A – standardní M-Bus odpověď na dotaz SND UD2 konfigurace seznamu vysílaných položek (pole byte indexů) na linku A
	W		
MBUS_LIST_B	R	0x41000000	M-Bus telegram linky B – standardní M-Bus odpověď na dotaz SND UD2 konfigurace seznamu vysílaných položek (pole byte indexů) na linku B
	W		
MBUS_LIST_C	R	0x42000000	M-Bus telegram linky C – standardní M-Bus odpověď na dotaz SND UD2 konfigurace seznamu vysílaných položek (pole byte indexů) na linku C
	W		
MBUS_LIST_D	R	0x43000000	M-Bus telegram linky D – standardní M-Bus odpověď na dotaz SND UD2 konfigurace seznamu vysílaných položek (pole byte indexů) na linku D
	W		
MODBUS_LIST_A	R	0x60000000	MODBUS telegram linky A – konfigurace seznamu vysílaných položek na MODBUS RTU adrese 0x1F80 (pole byte indexů)
	W		
MODBUS_LIST_B	R	0x61000000	MODBUS telegram linky B – konfigurace seznamu vysílaných položek na MODBUS RTU adrese 0x1F80 (pole byte indexů)
	W		
MODBUS_LIST_C	R	0x62000000	MODBUS telegram linky C – konfigurace seznamu vysílaných položek na MODBUS RTU adrese 0x1F80 (pole byte indexů)
	W		
MODBUS_LIST_D	R	0x63000000	MODBUS telegram linky D – konfigurace seznamu vysílaných položek na MODBUS RTU adrese 0x1F80 (pole byte indexů)
	W		

*MODBUS\_LIST\_x - dostupné od srpna 2013 pouze v některých INMATech*

**TUsart** (pro INMAT57)  
 speedIdx: byte; index rychlosti ze seznamu SPEED\_LIST  
 address: byte; adresa INMATu  
 flags: byte; |00ee 00pp| parita - spodní bity = [žádná, lichá, sudá]  
 MODBUS endianness - horní bity = [ABCD, CDBA, BADC, DCBA]  
 Padding: byte  
 mbusPlusPtr:word; maximální délka telegramu M-Bus+ = (mbusPlusPtr – basePtr)  
 mbusStdPtr:word; maximální délka telegramu M-Bus = (mbusStdPtr – basePtr)  
 MODBUSPlusPtr:word; maximální délka telegramu MODBUS = (MODBUSPlusPtr - basePtr)  
 endPtr:word;  
 basePtr:word;  
 padding:word;

**TUsart57d** [0..2] linka A,B,rezerva (pro INMAT 57D)  
 speedIdx: byte; index rychlosti ze seznamu SPEED\_LIST  
 address: byte; adresa INMATu  
 flags: byte; |00ee 0app| parita - spodní bity = [žádná, lichá, sudá]  
 a – aktivní linka (na které byl přijat tento dotaz)  
 MODBUS endianness - horní bity = [ABCD, CDBA, BADC, DCBA]  
 mbusStd: byte; maximální délka telegramu M-Bus  
 mbusPlus:word; maximální délka telegramu M-Bus+  
 MODBUSPlus:word; maximální délka telegramu MODBUS  
 end;

Vlastní nastavení parametrů komunikační linky musí být potvrzeno čtením nebo zápisem do INMATu na nově nastavených parametrech. Tím dojde k potvrzení a dokončení přestavení komunikace. V opačném případě se po 2 minutách v Inmatu nastaví původní parametry komunikace.

Pro provedení INMATu s více komunikačními linkami se potvrzení komunikace musí provést pro každou linku samostatně.

**XGASARCHIVE (CI = 0xCE)** (pouze INMAT 57S, 57D)

	C	SubCode	Struktura dat
GAS_ARCHIVE	R	0x00000000	Archiv složení plynu ( <b>TGasArch</b> * počet uložených změn)
	W		Nastavení kompresibility 8 x single float [x1..x8] z TGasArchive Úspěšné nastavení - ACK
	R	0x80000000	Názvy parametrů plynu (string dělený LF)
COMPRESIBILITY	R	0x81000000	Typ kompresibility (string LF)
LAST_RECORD	R	0x01000000	Aktuální parametry složení plynu <b>TGasArch</b>
DELETE_GAS_ARCHIVE	W	0x01000000	Smazání archivu kompresibility (pouze s metrologickým heslem) Úspěšné smazání - ACK

**TAchivRecord**

x1: single;  
x2: single;  
x3: single;  
x4: single;  
x5: single;  
x6: single;  
x7: single;  
x8: single;

pktime: longword;  
kompresibilita: byte;  
cisloZaznamu: byte;  
padding: byte;

Významy proměnných x1 až x8 se mění podle typu kompresibility. Aktuální proměnné lze zjistit dotazem GAS\_ARCHIVE Názvy parametrů plynu.

čas nastavení složení směsi  
index kompresibility  
pořadové číslo záznamu

**XCONFIG (CI = 0xC9)**

	C	SubCode	Struktura dat
MAIN_CONFIGURATION	R	0x00000000	Vyčtení hlavní konfigurace <b>TSysCfg</b>
	W		Nastavení hlavní konfigurace. Úspěšné nastavení-ACK
OUTPUT_CHARSET_LIST	R	0xC0000000	Seznam dostupných kodování v INMATu (string dělený LF)
LANGUAGE_LIST	R	0x80000000	Seznam dostupných jazyků v INMATu (string dělený LF)

**TXConfig**

agingOffset: shortInt;  
tempRate: byte;  
padding: word;  
timeUnit: longword;  
adjtime: longword;  
sysCfg: **TSysCfg**;

nastavení korekce RTC  
perioda teplotní kompenzace RTC (64, 128, 256, 512 sec)

čas posledního nastavení času

**TSysCfg**

language: byte;  
charset: byte;  
flags: byte;  
contrast: shortInt;

index aktuálně vybraného jazyka  
index znakové sady  
0x01 = přepínat letní/zimní čas, 0x02 = rotovat uživatelské menu  
kontrast displeje

*timeout-y pro displej*

backLightingTimeout: longword;

čas podsvětlení displeje = backLightingTimeout / (timeUnit \* 60 \* 3),  
0xFFFFFFFF = trvale

refreshValues: longword;

Refresh displeje

periodicEvents: longword;

Rychlost blikání kurzoru displeje

balanceExchangeTime: longword;

Prohlížení bilancí

menuRotationDelay: longword;

Autorotace menu - čas do spuštění

menuRotationPeriod: longword;

Perioda autorotace menu

**XPASSWD (CI = 0xD3)**

	C	SubCode	Struktura dat
Uživatelské heslo	R	0x00000000	Vrací čas poslední změny hesla
	W	0x01000000	Odemčení INMATu pro zápis na 3 min. Úspěšné odemčení- ACK
Metrologické heslo	R	0x40000000	Vrací čas poslední změny hesla
	W	0x41000000	Odemčení INMATu pro zápis na 30 sec. Úspěšné odemčení- ACK
			Nastavení metrologického hesla

**Příklad (XPASSWD) zápis uživatelské sumy, INMAT chráněn heslem**

```
Dotaz >                               SubCode
0x68 0x11 0x11 0x68 0x40 0x00 0xD8 0x00 0x00 0x00 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x1B 0x16
..... INMAT odpoví „Přístup je blokován uživatelským heslem!...“
Odpověď > Chybová zpráva SubCode kód chyby textový řetězec chyby
0x68 0x31 0x31 0x68 0x08 0x00 0x70 0x00 0x00 0x00 0x00 0x0D 0x50 0xF8 0xED 0x73 0x74 0x75 0x70 0x20
0x6A 0x65 0x20 0x62 0x6C 0x6F 0x6B 0x6F 0x76 0xE1 0x6E 0x20 0x75 0x9E 0x69 0x76 0x61 0x74 0x65 0x6C
0x73 0x6B 0xFD 0x6D 0x20 0x68 0x65 0x73 0x6C 0x65 0x6D 0x21 0x0A 0x00 0x16
ASCII > h1h.p.....Přístup je blokován uživatelským heslem!...
```

**Heslo se odblokuje zápisem platného hesla**

```
Dotaz>                               XPASS SubCode HESLO (zde 2222)
0x68 0x0B 0x0B 0x68 0x40 0x00 0xD3 0x00 0x00 0x00 0x00 0x32 0x32 0x32 0x32 0xDB 0x16
Odpověď > 0xE5 .. Heslo OK
```

**Druhý pokus o zápis sumy**

```
Dotaz >
0x68 0x11 0x11 0x68 0x40 0x00 0xD8 0x00 0x00 0x00 0x03 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x1B 0x16
Odpověď > 0xE5 Suma zapsána
```

**XBALANCE (CI = 0xC7) (pouze INMAT 57S, 57D)**

	C	SubCode		Struktura dat		
			Dostupný formát			
Konfigurace	R	0x70000000		Vyčtení rozložení TControlData *)		
	W	0x70000000		Nastavení hourAlarm z TControlData *). Úspěšné nastavení: ACK		
Vyčtení bilancí	R	0x00000000	0..7	FROM < [pkTime] **)	TO <= [pkTime] **)	Roky ***)
		0x10000000				Měsíce ***)
		0x20000000				Dny ***)
		0x30000000				Hodiny ***)
		0x40000000				Čtvrt hodiny ***)

\*)

**TControlData**

```
hourAlarm : longword; (R/W) hodina uložení pro roky, měsíce, dny. Např. 6 hodin - energetický den
Year      : longword; (R) jednotlivé počty záznamů, rozložení nelze změnit po lince
Month     : longword; (R)      "--
Day       : longword; (R)      "--
Hour      : longword; (R)      "--
quartHour : longword; (R)      "--
```

\*\*)

**FROM, TO** - nepovinný parametry umožňující specifikovat z jakého časového období požadujeme data. Neuvede-li se žádný parametr, INMAT vrátí vzestupně seřazené všechny dostupné záznamy.

**FROM** – INMAT vrátí všechny novější záznamy

**FROM, TO** – INMAT vrátí záznamy v intervalu FROM < záznamy <= TO

\*\*\*)

Počet, pořadí a názvy záznamů jsou shodné jako sumy v aplikaci INMATu. Délky jsou dle typu a požadovaného formátu proměnných.

Data jsou v telegramu řazena za sebou:

Záznam 1 [pkTime, suma 1, suma 2, suma 3, ...], Záznam 2 [pkTime, suma 1, suma 2, suma 3, ...], ...

**XARCHIVEBLOCK (CI = 0xC2, 0xC3, 0xC4, 0xC5) vyčtení archivace (pouze INMAT 57S, 57D)**

	CI	C	SubCode	Struktura dat		
Vyčtení archivace	0xC2	R	0x00000000	FROM < [pkTime] **)	TO <= [pkTime] **)	Blok 1 *)
	0xC3					Blok 2 *)
	0xC4					Blok 3 *)
	0xC5					Blok 4 *)

\*)

Data jsou v telegramu řazena za sebou:

Záznam 1 [pkTime, runtime[s], item1,item 2, item 3, ...], Záznam 2 [pkTime, runtime[s], item1,item 2, ...

Formát hodnot 4B (pkTime, jiné časy, ErrWord = LongWord, proměnné = single float, sumy = TrimmedSingle Float)

\*\*)

**FROM, TO** - nepovinný parametry umožňující specifikovat z jakého časového období požadujeme data. Neuvede-li se žádný parametr, INMAT vrátí vzestupně seřazené všechny dostupné záznamy.**FROM** – INMAT vrátí všechny novější záznamy**FROM, TO** – INMAT vrátí záznamy v intervalu FROM < záznamy <= TO**Příklad: (XBALANCE, XARCHIVEBLOCK)**

	SubCode
Dotaz > na všechny data single float dny	
0x68 0x07 0x07 0x68 0xE0 0x00 0xC7	0x00 0x00 0x00 0x21 0xCA 0x16
Dotaz > na single float od .. do	cist od < cist do <=
0x68 0x0F 0x0F 0x68 0xE0 0x00 0xC7	0x00 0x00 0x00 0x21 0xFF 0xFF 0x81 0x31 0x00 0x00 0xBC 0x31 0xE6 0x16
Dotaz > na single float od .. až po poslední záznam	cist od <
0x68 0x0C 0x0C 0x68 0xE0 0x00 0xC7	0x00 0x00 0x00 0x21 0x00 0x00 0x8A 0x31 0x83 0x16
<b>Dotaz na všechny data extended float hodiny – odpověď ve více telegramech</b>	
Dotaz1 > telegram 1	
0x68 0x07 0x07 0x68 0xE0 0x00 0xC7	0x00 0x00 0x00 0x33 0xDA 0x16
Odpověď > telegram 1	SubCode<0
0x68 0xF3 0xF3 0x68 0x8A 0x00 0xC7	0x16 0x00 0x00 0x33 0xDB 0xCE .....
	↓
Dotaz > telegram 2	SubCode z minulé odpovědi
0x68 0x07 0x07 0x68 0xE0 0x00 0xC7	0x16 0x00 0x00 0x33 0xF0 0x16
Odpověď > telegram 2	SubCode<0
0x68 0xF3 0xF3 0x68 0x8A 0x00 0xC7	0x2C 0x00 0x00 0x33 0x00 0xA0 .....
	↓
Dotaz > telegram 3	SubCode z minulé odpovědi
0x68 0x07 0x07 0x68 0xE0 0x00 0xC7	0x2C 0x00 0x00 0x33 0x06 0x16
Odpověď > telegram 3	SubCode = 0 → data jsou vyčtena
0x68 0xF3 0xF3 0x68 0x8A 0x00 0xC7	0x00 0x00 0x00 0x00 0x00 0x80 .....

**XARCHIVECFG (CI = 0xC6)** (pouze INMAT 57S, 57D)

	C	SubCode	Struktura dat
ARCHIVE_CONFIG	R	0x00000000	Vyčtení konfigurace archivace <b>TArchCfg</b>
	W		Nastavení hodiny uložení <b>TArchCfg</b> Úspěšné nastavení: ACK
ARCHIVE_RECORDS	R	0x10000000	Pole všech archivovaných hodnot <b>TArchRec</b>
	W		Úspěšné nastavení: ACK
ARCHIVE_EEPROM	R	0x18000000	Dostupná archivační paměť <b>LongWord</b>
ARCHIVE_RAM	R	0x1C000000	Paměťová náročnost výpočtu : array[0..15] of byte
ARCHIVE_DATA_TYPES or BLOCKx		0x14000000	Pole s datovými typy archivovaných hodnot <b>     iitt</b> ve vyšších bitech je unikátní identifikátor skupiny proměnných ve spodních 2 bitech je datový typ Single (4B) = 0, bitmapa (stavové slovo 4B) = 1, časy LongWord (4B) = 2, pkTime (4B) = 3 Délka pole = počet hodnot v bloku
ARCHIVE_RAM_DEMANDS		0x08000000	Pole s datovou náročností archivace na RAM
VARIABLES_COMPUTING		0x04000000	Typ výpočtu pro skupinu archivace
ARCHIVE_NAMES		R W	0x80000000 or Kódování
GROUPS_OF_VARIABLES	R	0xC0000000	Názvy dostupných skupin proměnných (string dělený LF)
ITEMS_OF_VARIABLES		0xE?000000	Názvy archivaci dostupných proměnných (string dělený LF) ? = index skupiny GROUPS_OF_VARIABLES
ARCHIVE_TYPES		0xA0000000	Dostupné typy archivace (perioda, po zapnutí,...)
ARCHIVE_BASIC COMPUTING		0xB0000000	Popis - typy výpočtu (string dělený LF)
ARCHIVE_CUSTOM COMPUTING II		0xB8000000	Rezerva (string dělený LF)
ARCHIVE_CUSTOM COMPUTING III		0xBC000000	Rezerva (string dělený LF)
ARCHIVE_CUSTOM COMPUTING I		0xB4000000	Rezerva (string dělený LF)
CHOSEN_GROUPS_OF_VARS or BLOCKx		0xA8000000	Jména groups archivovaných položek v jednotlivých blocích (string dělený LF)
CHOSEN_ITEMS_OF_VARS or BLOCKx	R	0xAC000000	Názvy archivovaných položek v jednotlivých blocích (string dělený LF)
ARCHIVE_PERIODS		0xA4000000	Dostupné periody archivace (string LF)
or BLOCK1		0x00000000	určení bloku (default blok 1)
or BLOCK2		0x01000000	
or BLOCK3		0x02000000	
or BLOCK4		0x03000000	

**TArchCfg**

Block : array[0..3] of **TArchBlock**;  
 Pktime : longWord; čas posledního nastavení archivace pkTime  
 seStartem : byte; hodnota nastavení se startem v X:00 hodin (synchronizace s RTC)  
 focusBlock : byte; aktuální číslo bloku v okamžiku uložení  
 flags : byte; konfigurace případné speciální verze archivace  
 version : byte; verze archivace (rezerva pro speciální verze)

**TArchBlock**

Flags : byte; skupina obecných bitových příznaků bloku  
 0x01 blok povolen  
 0x02 relační podmínky (AND = 1 OR = 0)  
 0x04 (cyklicky přepisovat záznamy  
 0x08 (po záznamu vynulovat maxima  
 0x10 povolit cyklus se startem v X:00 hodin (synchronizace s RTC)  
 archType : byte; typ archivace (perioda | změna podmínek | stavové slovo | po zapnutí)  
 hodnotaPeriody: byte; perioda archivace (až 255)  
 typPeriody : byte; sec, min, hod, den, dtyd, měs, dměs, rok...

condCfgA : **TCondCfg**; podmínka A  
 condCfgB : **TCondCfg**; podmínka B  
 errWordBitMask : longword; bitová maska chyby (max 32)  
 pocetZaznamuVExterniPameti : longword; Počet záznamů v bloku  
 pocetHodnotVBloku : byte; počet hodnot v bloku (bez pkTime a času provozu – ty jsou povinné vždy)  
 padding1: byte;  
 padding2: byte;  
 padding3: byte;

**TCondCfg**

flags: byte; skupina příznaků pro podmínku  
 0x01 podmínka povolena  
 0x02 (1=konstanta , 0=proměnná)  
 0x04 konstanta je celočíselná  
 padding1: byte;  
 padding2: byte;  
 relop: byte; relační operátor podmínky AND = 1 OR = 0  
 leftGroup: byte; skupina (sumy, uživatelské sumy, systémové proměnné, ...)  
 leftItem: byte; pořadí položky ve skupině  
 rightGroup: byte; skupina (sumy, uživatelské sumy, systémové proměnné,...)  
 rightItem: byte; pořadí položky ve skupině  
 constant: **TypeCast**; hodnota konstanty (4B float) nebo LongWord (provoz, nebo pktime)

**TArchRec**

item: byte; |zzpp pppp| Pořadí konkrétní položky ve skupině proměnných GroupCalc  
 |zzpp pppp| rezerva - bitový příznak pro zákaznické výpočty  
 groupCalc: byte; |gggg cccc| index skupiny proměnných  
 |gggg cccc| obsahuje typ výpočtu (0-3 pro sumy, 4-7 pro hodnoty, 8-11 pro casy, 12-15 rezervováno)

**Dotazy potřebné pro vyčtení archivace s neznámou strukturou:**

- 1.) Dotaz na typ položek „XARCHIVECFG, ARCHIVE\_DATA\_TYPES or BLOCK“  
Počet archivovaných hodnot je daný velikostí vráceného datového pole
- 2.) Názvy proměnných „XARCHIVECFG, CHOSEN\_ITEMS\_OF\_VARS or BLOCK“
- 3.) Vlastní vyčtení dat „XARCHIVEBLOCK or BLOCK,[FROM, TO]“

**Interní specializované dotazy komunikace (seřizování, flash, diagnostika)**

XAPPLIC	rezerva	
XIMPOUT		
IMPULSE_OUTPUT_CONFIG	= DEFAULT;	
XIOUT		
IOUT_CONFIG	= DEFAULT;	(R/W)
IOUT_CALIBRATION	= 0x40000000;	(W)
XADCONFIG		
KONFIG_PREVODNIKU	= DEFAULT;	(R/W)
XRESET		
XRESET_USR_CONST	= 0x08000000;	
XRESET_MCU	= 0x80000000;	
XRESET_ALL	= 0xFF000000;	
XDIAGNOSTIC		
DIAGN_DATA	= DEFAULT;	(R/W/RS)
STOP_STRINGS	= 0x40000000;	(R/W/RS)
STATISTICS_DATA	= 0x20000000;	(R/RS)
ERROR_DETAILS	= 0x10000000;	(R)
READ_REGISTERS	= 0x02000000;	(RS)
SAVE_REGISTERS	= 0x01000000;	(R)
XUPDATER		
ALL_FIRMWARE	= DEFAULT;	(RS/WS);
GET_NUMBER_OF_BLOCK	= 0x40000000;	(R)
MCU_FIRMWARE	= 0x20000000;	
AD_CONFIG_OVERWRITE	= 0x02000000;	
SYSTEM_CONFIG_OVERWRITE	= 0x01000000;	
XWORKLOAD		
WORKLOAD_DATA	= DEFAULT;	(R/RS)
XVARIABLES		
SIMULATOR_OFF	= 0x10000000;	(W)



### 3. Implementace MODBUS RTU v INMATu

#### 3.1. Linková vrstva

Struktura splňuje MODBUS RTU specifikaci

Slave adresa [1B]	Funkce [1B]	„Base“ Adresa [2B]	Počet registrů [2B]	CRC [2B]
-------------------	-------------	--------------------	---------------------	----------

#### 3.2. Aplikační vrstva

**Čtení dat:** Proměnné jsou dostupné v Input Registerch - funkce 0x04 Read Input Registers

Proměnné jsou uloženy v 16 bitových registrech (2Byte).

Formát je možné nastavit jako **big endian, mixed endian, little endian**. (Nastavuje se pomocí SWK45702)

Počáteční „Base“ adresa se vytvoří bitovým součtem položky "t", "s", "p"

| tttt ssss | sPPP pppp |

kde:

"t" požadovaný datový typ

"s" vybraný seznam

"p" pořadí proměnné v seznamu

**INMAT 57S, INMAT 57D Datový typ** | tttt ???? | ???? ???? |

t	Použité datové typy	Adresa	Počet registrů
0	longWord (v případě sum longWord *100)	0x0000	2
1	single float	0x1000	2
2	double float	0x2000	4
3	extended float	0x3000	5
4	trimmed longWord (u sum longWord * 100)	0x4000	2
5	trimmed single float	0x5000	2
6	trimmed double float	0x6000	4
7	extended float	0x7000	5
8	strings (oddělené LF) vrací všechny dostupné řetězce seznamu	0x8000	<>0

Pozn: Použité datové typy - viz popis u protokolu M-Bus+

**INMAT 57S, INMAT 57D Skupina proměnných** | ???? ssss | s??? ???? |

Skupina/seznam proměnných	Adresa	Dostupný datový typ
sumy	0x0000	0 až 8
uživatelské sumy	0x0080	0 až 8
systémové proměnné	0x0100	1 a 8
pomocné proměnné	0x0180	1 a 8
okamžité proměnné	0x0200	1 a 8
uživatelské konstanty	0x0280	1 a 8
qhmax - čtvrt hodinové maximum	0x0300	1 a 8
qhmax – časy dosažení	0x0380	0 (pkTime) a 8
avgmax – 1 minutové maximum	0x0400	1 a 8
avgmax – 1 minutové maximum časy dosažení	0x0480	0 (pkTime) a 8
max – maximum	0x0500	1 a 8
max – maximum - časy dosažení	0x0580	0 (pkTime) a 8
RTC	0x0600	0 (pkTime) a 8
doby provozu	0x0680	0 (LongWord [s]) a 8
chybové slovo	0x0700	0 (bitová maska) a 8

Pro vyčítání sum je vhodné používat trimmed single float.

**INMAT 57S, INMAT 57D Konkrétní proměnná ze skupiny** | ???? ???? | ?ppp pppp |

Pořadí proměnné	Adresa	Výpočet adresy proměnné
adresa konkrétní proměnné	0x0000 - 0x007F	Adresace verze 1 *) [ppp pppp] = (pořadí proměnné -1) * počet registrů
		Adresace verze 2 *) Popis viz viz níže [ppp pppp] = pořadí proměnné -1

**INMAT 59 Datový typ | tttt ???? | ???? ???? |**

t	Použité datové typy	Adresa	Počet registrů
0	LongWord	0x0000	2
1	single float	0x1000	2
2	double float	0x2000	4
3	extended float	0x3000	5

Pozn: Použité datové typy - viz popis u protokolu M-Bus+

**INMAT 59 Skupina proměnných | ???? ssss | s??? ???? |**

Skupina/seznam proměnných	Adresa	Dostupný datový typ, struktura dat
sumy	0xt000	single float, double, extended
uživatelské sumy	0xt080	single float, double, extended
systémové proměnné	0x1100	single float
pomocné proměnné	0x1180	single float
okamžité proměnné	0x1200	single float
uživatelské konstanty	0x1280	single float
qhmax - čtvrt hodinové maximum	0x1300	Čas vyčtení (longWord - pkTime), maximum 1 (single float), čas maxima 1 (longWord - pkTime), maximum 2 ... atd....
avgmax - 1 minutové maximum	0x1400	Čas vyčtení (longWord - pkTime), maximum 1 (single float), čas maxima 1 (longWord -pkTime), maximum 2... atd....
max - maximum	0x1500	Čas vyčtení (longWord - pkTime), maximum 1 (single float), čas maxima 1 (longWord - pkTime), maximum 2... atd....
RTC	0x0600	longWord (pkTime)
doby provozu	0x0680	longWord [s]
chybové slovo	0x0700	longWord (bitová maska)

**INMAT 59 Konkrétní proměnná ze skupiny | ???? ???? | ?ppp pppp |**

Pořadí proměnné	Adresa	Výpočet adresy proměnné
adresa konkrétní proměnné	0x0000 - 0x007F	Adresace verze 2 *) Popis viz viz níže [ppp pppp] = pořadí proměnné -1

Pozn. Pořadí proměnných, jejich názvy, počty a význam chybového slova se liší podle aplikace. Jejich popis je součástí dokumentace dodávané spolu s INMATem.

\*) Aktuální seznam adres lze vyčíst z INMATu pomocí programu SWK45702 „Testy/Adresy MODBUS RTU“

**Změny Verze 2 MODBUS - RTU komunikace INMATu (Obě verze adresace se vyrábějí současně.)**

**Adresace verze 2 - Adresa konkrétní proměnné je dána pořadím ve skupině [0 až n].**



**Neuplatňuje se datová náročnost proměnné !**

**Nastavitelný seznam proměnných: (Inmat 57D, Inmat 59)**

Na adrese **0x1F80** je dostupný seznam proměnných připravený v aplikaci INMATu. Seznam lze v rámci předpřipraveného seznamu upravit (měnit pořadí, omezit výběr). Nastavuje se pomocí programu SWK45702 „Nastavení/MODBUS-RTU Linka A (nebo B, C, D)“

Data jsou ve formátu single float, 4Byte (2 registry), longWord 4Byte (2 registry) nebo pkTime 4Byte (2 registry).

V případě maxim a bilancí INMAT vrací 4Byte single float hodnotu a 4Byte čas dosažení ve formátu pkTime (celkem 4 registry).

Položku čas dosažení - pkTime u maxim a bilancí nelze samostatně adresovat.

**Zápis dat (INMAT 57D)**

**funkce 0x10** Write Multiple Registers

**Adresa 0** - Nastavení času RTC ve formátu pkTime (4Byte). INMAT kontroluje korektnost hodnoty data a času.

Pozor: Adresy INMATu 16 (0x10) a 104 (0x68) nelze použít při komunikaci pomocí protokolu MODBUS. Využívají se k autodekci protokolu – INMAT je vyhodnotí jako dotaz M-Bus normy.

### 3.3. Příklady komunikace MODBUS-RTU

Příklad celého telegramu:

	Inmat	Funkce	Adresa	Počet registrů	CRC
Dotaz >	0x01	0x04	0x11 0x00	0x00 0x02	0x74 0xF7
	Inmat	Funkce	Délka	Data	CRC
Odpověď >	0x01	0x04	0x04	0x00 0x00 0x00 0x00	0xFB 0x84

#### 1.) všechny sumy jako trimmed single

**Funkce** 1B „Read Input Registers“ 0x04  
**Počáteční adresa** 2B = 0x5000 or 0x0000 or 0x0000 => 0x5000  
**Počet registrů** 2B = 2 \* počet sum dostupných v konkrétním INMATu

#### 2.) Druhá suma jako single float

**Funkce** 1B „Read Input Registers“ 0x04  
*Pro adresaci verze 1*  
**Počáteční adresa** 2B = 0x1000 or 0x0000 or 0x0002 => 0x1002  
*Pro adresaci verze 2*  
**Počáteční adresa** 2B = 0x1000 or 0x0000 or 0x0001 => 0x1001  
**Počet registrů** 2B = 2

#### 3.) Okamžitá hodnota (šestá proměnná)

**Funkce** 1B „Read Input Registers“ 0x04  
*Pro adresaci verze 1*  
**Počáteční adresa** 2B = 0x1000 or 0x0200 or 0x000A => 0x120A  
*Pro adresaci verze 2*  
**Počáteční adresa** 2B = 0x1000 or 0x0200 or 0x0005 => 0x1205  
**Počet registrů** 2B = 2

#### 4.) Názvy systémových proměnných

**Funkce** 1B „Read Input Registers“ 0x04  
**Počáteční adresa** 2B = 0x8000 or 0x0100 or 0x0000 => 0x8100  
**Počet registrů** 2B = 1 (počet registrů v dotazu zadat rozdílný od 0)

#### 5.) Čtení z nastavitelného seznamu proměnných

Aktuální nastavení seznamu:

Vytvořený seznam

```
1: E1 [GJ]
30: t1 [°C]
31: ^t1 [°C]
43: runtime
44: M-Bus time point [date + time]
45: -1 month Mc1 [t]
```

E1 - suma (formát trimovaný single float 4Byte/2registry)  
t1 - teplota (formát single float 4Byte/2registry)  
^t1 - maximum teploty (formát single float + pkTime čas dosažení 4Byte+4Byte/ 4 registry)  
runtime - Provozní čas (longWord 4Byte/ 2 registry)  
M-Bus time point (date + time) - RTC čas (formát pkTime 4Byte / 2 registry)  
-1 month Mc1 – bilance formát single float + pkTime 4Byte+4Byte / 4 registry)

Čtení celého seznamu od první adresy

**Funkce** 1Byte „Read Input Registers“ 0x04  
**Počáteční adresa** 2Byte = 0x1F80  
**Počet registrů** 2Byte = 0x0010

Tag : Adresa první (base) hodnoty 8064 (0x1F80) Pts : Počet hodnot 8 (16 registrů)

ID: ?	Function: ?	Type: ?	Tag #: ?	# pts: ?
1	Input Register	Float	8064	8
8064:	1.757	E1		
8066:	322.397	t1		
8068:	599.976	maximum t1		
8070:	1811495172	čas dosažení maxima t1		
8072:	1475936257	provozní čas		
8074:	2787448003	RTC čas		
8076:	-1.#Q0	bilance Mc1		
8078:	0.000	čas dosažení bilancí Mc1		

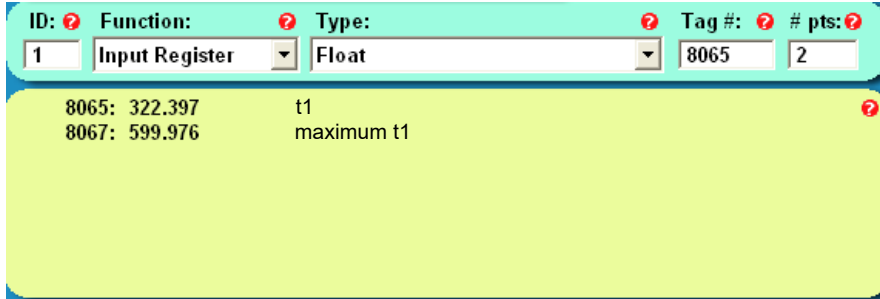
	Inmat	Funkce	Adresa	Počet registrů	CRC
Dotaz >	0x01	0x04	0x1F 0x80	0x00 0x10	0xF7 0xFA

	Inmat	Funkce	Délka	Data	CRC
Odpověď >	0x01	0x04	0x20	0x41 ..... (20 byte)...	0xE1 0x4C

Čtení druhé a třetí adresy „t1“ a „maxima t1“ stejného seznamu.

**Funkce** 1Byte „Read Input Registers“ 0x04  
**Počáteční adresa** 2Byte = 0x1F81  
**Počet registrů** 2Byte = 0x0004

Tag : Adresa první (base) hodnoty 8065 (0x1F81) Pts : Počet hodnot 2 (4 registry)



Čas dosažení maxima se v tomto případě ignoruje, protože bylo požádáno pouze o 4 registry. Dva registry teplota a dva maximum teploty.

	Inmat	Funkce	Adresa	Počet registrů	CRC
Dotaz >	0x01	0x04	0x1F 0x81	0x00 0x04	0xA6 0x35
	Inmat	Funkce	Délka	Data	CRC
Odpověď >	0x01	0x04	0x08	0x41 ..... (8 byte)...	0xE1 0x4C

### 6.) Čtení čtvrt hodinového maxima INMAT 59 – všechny proměnné

**Funkce** 1B „Read Input Registers“ 0x04  
**Počáteční adresa** 2B = 0x1300 or 0x0000 => 0x1300  
**Počet registrů** 2B = 1 + 2 \* počet čtvrt hodinových maxim dostupných v konkrétním INMATu

	Inmat	Funkce	Adresa	Počet registrů	CRC
Dotaz >	0x01	0x04	0x13 0x00	0x00 0x0A	0x74 0x89
	Inmat	Funkce	Délka	Data	CRC
Odpověď >	0x01	0x04	0x14	0x4B ..... (20 byte)...	0xD4 0x6F

[0xaa00] : 18.12.2018 13:58:30 // čas vyčtení telegramu  
 [0xaa01] : 125,356 // hodnota prvního maxima  
 [0xaa02] : 18.12.2018 13:58:17 // čas dosažení prvního maxima  
 [0xaa03] : 75,1547 // hodnota druhého maxima  
 [0xaa04] : 18.12.2018 13:58:17 // čas dosažení druhého maxima

*Příklady vytvořeny v demo verzi programu DAQFactory.*

## 4. Implementace M-Bus v INMATu

Linková i aplikační vrstva vychází z ČSN EN 1434-3 -3, viz <http://www.m-bus.com/mbusdoc/md5.php>

Příkaz	C	CI	Odpověď
SND_NKE	0x40	-	Inicializace účastníka 0xE5
SND_UD	0x73	0x51	0xE5
SND_UD1	0x7A	-	0xE5
SND_UD2	0x5B	-	C=0x08, CI=0x72 + data (Telegram připravený v INMATu, lze upravit viz. M-Bus+ XCOMMUNICATION, MBUS_LIST)

Ostatní komunikace není implementována.

Datová částí odpovědi odpovídá standardu M-Bus. (DIF [DIFE] + VIF [VIFE] + data)

RTC odpovídá normě M-Bus komunikace.

Příklad odpovědi:

**C=08 A=00 CI=72**

```
68 BD BD 68 08 00 72 08 00 06 12 01 6A 57 05 6E ← identifikace 12060008 ZPA 57 Steam
00 00 00 | 05 FB 09 67 E3 A0 40 | 05 1E 54 E7 C1 3F ← DIF, VIF, VIFE, vlastní hodnota
```

typ = „real4B“, tarif = „0“, storage = „0“, „Okamžitá hodnota“, jednotka = „Energy[GJ]“, hodnota = „5.027759e+000“

...

