

Popis komunikačních protokolů přístrojů SMY33, SMZ33

Příručka pro programátory

(platí pro verzi fw 7.3 a vyšší)

OBSAH

| | |
|--|-----------|
| 1. PŘENOS DAT MEZI PŘÍSTROJI SMY/SMZ A NADŘAZENÝM SYSTÉMEM..... | 3 |
| 2. KOMUNIKAČNÍ PROTOKOLY | 4 |
| 2.1 Komunikační protokol KMB..... | 4 |
| 2.1.1 Popis zpráv..... | 4 |
| 2.1.2 Identifikace 0x01 | 5 |
| 2.1.3 Nastavení RTC 0x10 | 5 |
| 2.1.4 Přečtení RTC 0x11..... | 6 |
| 2.1.5 Přečtení stavu přístroje (Status) – 0x14 | 6 |
| 2.1.6 Přečtení nastavení přístroje (Config) – 0x26 | 6 |
| 2.1.7 Zápis nastavení do přístroje (Config) – 0x27 | 6 |
| 2.1.8 Přečtení nastavení výstupů přístroje (OutputConfig) – 0x30..... | 7 |
| 2.1.9 Zápis nastavení výstupů do přístroje (OutputConfig) – 0x31 | 7 |
| 2.1.10 Přečtení nastavení tarifů elektroměru přístroje (ElmerTarif) – 0x32..... | 7 |
| 2.1.11 Zápis nastavení tarifů elektroměru přístroje (ElmerTarif) – 0x33 | 7 |
| 2.1.12 Přečtení aktuálního stavu elektroměru přístroje (ElmerActData) – 0x34 | 7 |
| 2.1.13 Nulování elektroměru přístroje – 0x35 | 7 |
| 2.1.14 Předání všech aktuálních dat z přístroje (ActAlldata) 0x3A..... | 8 |
| 2.2 Komunikační protokol Modbus-RTU..... | 9 |
| 2.2.1 Status | 10 |
| 2.2.2 Config..... | 10 |
| 2.2.3 OutputConfig | 11 |
| 2.2.4 RTC | 11 |
| 2.2.5 Identifikace | 11 |
| 2.2.6 ElmerTarif..... | 12 |
| 2.2.7 ClrElmer | 12 |
| 2.2.8 Aktuální data měřených veličin..... | 13 |
| 2.2.9 Stav elektroměru | 14 |
| 3. DATOVÉ STRUKTURY | 15 |

1. Přenos dat mezi přístroji SMY/SMZ a nadřazeným systémem

Přístroje SMY33 a SMZ33 mohou být vybaveny dálkovou komunikační linkou s rozhraním RS 232, nebo RS 485. Pomocí této linky mohou být monitorovány a ovládány z nadřazeného řídicího systému (obvykle PC).

Tato příručka popisuje způsob komunikace s přístroji. Předpokládá základní znalost parametrů přístrojů a jazyka C.

Data, která lze přenášet mezi přístrojem a nadřazeným systémem, jsou uspořádána do následujících struktur :

- „Status“ ... obsahuje informace o stavu přístroje (výrobní číslo, chybové příznaky atd.); lze pouze číst
- „Identifikace“ ... obsahuje informace o typu přístroje (výrobní číslo, typ přístroje atd.); lze pouze číst
- „RTC“ ... umožňuje číst a nastavovat obvod reálného času
- „Config“ ... obsahuje stav nastavitelných parametrů přístroje (převod MTN, MTP, atd); lze číst i zapisovat
- „OutputConfig“ ... obsahuje stav nastavitelných parametrů výstupů přístroje (relé, pulsní výstupy, atd...)
- „Elektroměr“ ... obsahuje stav elektroměru a umožňuje číst a nastavovat tarify
- „Profily“ ... umožňuje číst a nastavovat informace o měření proudových profilů
- „ActData“ ... umožňuje přečtení aktuálně změřených dat z přístroje

Informace o stavu přístroje získá nadřazený systém přečtením odpovídající struktury z přístroje, naopak zápisem do odpovídající struktury může nadřazený systém například změnit některý z parametrů, spustit vybranou operaci atd.

Obsah jednotlivých struktur je uveden v samostatné kapitole dále.

2. Komunikační protokoly

Přenos informací mezi přístrojem a nadřazeným systémem se provádí přes sériovou asynchronní komunikační linku (COM) s rozhraním RS-232 nebo RS-485. V případě použití rozhraní RS-485 je možné na jednu komunikační linku připojit více přístrojů a na straně nadřazeného systému je možné použít převodník rozhraní RS-232/RS-485, vybavený automatickým přepínáním směru přenosu dat, případně musí přepínání směru přenosu dat zajistit program nadřazeného systému.

Přístroje jsou standardně dodávány s přednastaveným firemním protokolem „KMB“. U přístrojů, vybavených firmware verze 7.3 a vyšším lze volitelně nastavit i protokol Modbus-RTU.

Rychlost komunikace je nastavitelná v rozsahu 300-19200 Bd (standardně 9600 Bd).

2.1 Komunikační protokol KMB

Komunikační kanál je nastaven na 8 bitů, bez parity, jeden stop-bit.

Komunikace typu „master-slave“. Na základě přijetí řádné zprávy-příkazu přístroj odešle odpovídající zprávu-odpověď.

Zprávy mají jednotný formát :

1. Adresa přístroje
2. Délka zprávy (v bytech) bez závěrečné checksum. Má vždy hodnotu (3+délka těla zprávy).
3. Typ zprávy (1 byte).
4. Tělo zprávy - má různou délku dle typu zprávy.
5. Závěrečná checksum - součet předchozích byte modulo 256 (1 byte).

Pokud přístroj přijme řádnou zprávu, a podaří se mu ji řádně zpracovat (vykonat příkaz), odešle příslušnou odpověď a na místě typu zprávy bude hodnota 0. Pokud je typ zprávy v odpovědi nenulový, příkaz se z nějakých důvodů nepodařilo vykonat.

Některé zprávy nemají tělo.

Přístroj odešle odpověď do 600ms po obdržení zprávy od mastera. Během příjmu příkazu nebo vysílání odpovědi může nastat mezibytová mezera délky odpovídající době přenosu maximálně 2 znaků (bytů).

2.1.1 Popis zpráv

Pro zápis/čtení datových struktur lze použít zprávy dle následující tabulky.

| č. zprávy (hex) | typ zprávy |
|-----------------|---|
| 0x01 | Identifikace přístroje |
| 0x10 | Nastavení RTC |
| 0x11 | Přečtení RTC |
| 0x14 | Přečtení stavu přístroje – Status |
| 0x26 | Přečtení nastavení přístroje – Config |
| 0x27 | Zápis nastavení do přístroje - Config |
| 0x30 | Přečtení nastavení výstupů přístroje – OutputConfig |
| 0x31 | Zápis nastavení výstupů přístroje - OutputConfig |

| | |
|------|--|
| 0x32 | Přečte nastavení tarifů elektroměru |
| 0x33 | Nastaví tarify elektroměru |
| 0x34 | Přečtení stavu elektroměru |
| 0x35 | Nulování elektroměru |
| 0x36 | Předání ProfilStatusu a aktuálních dat profilů |
| 0x37 | Nastavení času pro sledování denního profilu |
| 0x38 | Nulování času pro sledování max. profilu |
| 0x39 | Přečte časy zaznamenaných profilů |
| 0x3A | Předá všechna aktuální data |

V následujících kapitolách je uveden bližší popis těchto zpráv.

2.1.2 Identifikace 0x01

Na dotaz mastera se přístroj „identifikuje“.

Master:

| 0x01| 0x03| 0x01 | 0x05 |

Přístroj odpoví:

| 0x01| 0x11| 0x00 | Identifikace 14 bytů |checksum|

Identifikace (rozepsáno byt po bytu, celkem 14 bytů):

DeviceNo % 256
 DeviceNo / 256
 DeviceType % 256
 DeviceType / 256
 PropsType % 256
 PropsType / 256
 SoftVersion
 Rezerva
 RemoteAdresa % 256
 RemoteAdresa / 256
 Rezerva
 Rezerva
 Rezerva
 Rezerva

2.1.3 Nastavení RTC 0x10

Nadřazený systém předá přístroji čas, dle kterého bude nastaven obvod reálného času v přístroji.

Příklad :

Požadavek na nastavení datumu a času je např.: 2003.8.15 10:29:00. Master odešle následující sekvenci (předpokládáme, že adresa přístroje je 1) :

| 0x01| 0x09 | 0x10 | 0x03|0x08|0x15|0x10|0x29|0x00|checksum|

Struktura odpovědi v případě úspěšného příjmu :

| 0x01| 0x03| 0x00 | checksum |

2.1.4 Přčtení RTC 0x11

Nadřazený systém požaduje předání stav obvodu reálného času z přístroje.

Dotaz mastera:

| 0x01| 0x03| 0x11 | checksum |

Odpověď přístroje

| 0x01| 0x09 | 0x00 | 0x03|0x08|0x15|0x10|0x29|0x00|checksum|

2.1.5 Přčtení stavu přístroje (Status) – 0x14

V odpovědi předá přístroj tzv. Status (52 bytů).

Příklad :

Master odešle následující sekvenci (předpokládáme, že adresa přístroje je 1) :

| 0x01| 0x03 | 0x14 | checksum = 0x18 |

Příkaz nemá tělo.

Struktura odpovědi :

| 0x01| 0x37| 0x00 | tělo zprávy = Status (52 bytů) | checksum |

2.1.6 Přčtení nastavení přístroje (Config) – 0x26

Přístroj předá tzv. Config (28 bytů).

Příkaz :

| 0x01| 0x03 | 0x26 | checksum = 0x2A |

Odpověď :

| 0x01| 0x1F| 0x00 | tělo zprávy = Config (28 bytů) | checksum |

2.1.7 Zápis nastavení do přístroje (Config) – 0x27

Zápis tzv. Config (28 bytů) do přístroje.

Příkaz :

| 0x01| 0x1F | 0x27 | tělo zprávy = Config (28 bytů) | checksum |

Odpověď :

| 0x01| 0x03| 0x00 | checksum = 0x04 |

Poznámka : Proměnné DeviceAddr a RemoteBDRate nelze přes Remote komunikační linku měnit, zapisované hodnoty těchto proměnných mohou být libovolné.

2.1.8 Přečtení nastavení výstupů přístroje (OutputConfig) – 0x30

Přístroj předá tzv. OutputConfig (20 bytů).

Příkaz :

| 0x01| 0x03 | 0x30 | checksum = 0x34 |

Odpověď :

| 0x01| 0x17| 0x00 | tělo zprávy = OutputConfig (20 bytů) | checksum |

2.1.9 Zápis nastavení výstupů do přístroje (OutputConfig) – 0x31

Zápis tzv. OutputConfig (20 bytů) do přístroje.

Příkaz :

| 0x01| 0x17 | 0x31 | tělo zprávy = OutputConfig (20 bytů) | checksum |

Odpověď :

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.10 Přečtení nastavení tarifů elektroměru přístroje (ElmerTarif) – 0x32

Přístroj předá tzv. ElmerTarif (6 bytů).

Příkaz :

| 0x01| 0x03 | 0x32 | checksum = 0x36 |

Odpověď :

| 0x01| 0x09| 0x00 | tělo zprávy = ElmerTarif (6 bytů) | checksum |

2.1.11 Zápis nastavení tarifů elektroměru přístroje (ElmerTarif) – 0x33

Zápis tzv. ElmerTarif (6 bytů) do přístroje.

Příkaz :

| 0x01| 0x09| 0x33 | tělo zprávy = ElmerTarif (6 bytů) | checksum |

Odpověď :

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.12 Přečtení aktuálního stavu elektroměru přístroje (ElmerActData) – 0x34

Přístroj předá tzv. ElmerActData (94 bytů).

Příkaz :

| 0x01| 0x03 | 0x34 | checksum = 0x38 |

Odpověď :

| 0x01| 0x61| 0x00 | tělo zprávy = ElmerActData (94 bytů) | checksum |

2.1.13 Nulování elektroměru přístroje – 0x35

Příkaz :

| 0x01| 0x04 | 0x35 | 0x01| checksum = 0x3B |

Odpověď :

| 0x01| 0x03| 0x00 | checksum = 0x04 |

2.1.14 Předání všech aktuálních dat z přístroje (ActAlldata) 0x3A

Příkaz :

| 0x01| 0x03 | 0x3A| checksum = 0x3E |

Odpověď :

| 0x01| 0xDD| 0x00 | 218 bytů dat | checksum |

2.2 Komunikační protokol Modbus-RTU

Přístroje s firmware 7.3 a vyšší lze nastavit na komunikační protokol Modbus-RTU. Vedle adresy a rychlosti komunikace lze nastavit i funkci paritního bitu (sudá / lichá / žádná parita).

Přístroj odešle odpověď nejpozději do 600ms po obdržení zprávy od mastera. Během příjmu příkazu nebo vysílání odpovědi může nastat mezibytová mezera délky odpovídající době přenosu maximálně 1,5 znaku (bytů).

Režim "broadcast" není podporován.

Jsou implementovány následující funkce :

| kód funkce | název funkce | aplikace |
|------------|--|--|
| 03 | Read Holding Registers | Čtení nastavení a stavu přístroje – Status, Config, OutputConfig, ElmerTarif |
| 04 | Read Input Registers | Čtení aktuálních dat z přístroje – ActAllData, ElmerActData |
| 06 | Preset Single Register | zápis Config, OutputConfig, ElmerTarif, nulování Elmeru |
| 08 | Diagnostics – 00 – Return Query Data 01 – Restart Comm Option 10 – Clear Ctrs & Diag. Register 11 – Return Bus Message Count | základní diagnostické funkce |
| 16 | Preset Multiple Registers | obdobně jako 06 - Preset Single Register |

Přístup ke strukturám je realizován pomocí čtení/zápisu z/do odpovídajících registrů dle tabulek uvedených v následujících kapitolách. Jednobytové údaje jsou v registru uloženy ve tvaru 0x00nn, nn je jednobytový parametr.

Datové struktury, popisující nastavení a stav přístroje, jsou uloženy v poli „holding“-registrů. Aktuální naměřená data lze číst jako stav „input“-registrů.

Každá ze struktur je v poli registrů uložena od různé tzv. bazové adresy dle následující tabulky (platí pro „holding“-registry). Aktuální data jsou uložena v poli „input“-registrů od bazové adresy 0.

| struktura | bázová adresa holding-registru |
|----------------|--------------------------------|
| IDENTIFICATION | 0x0200 |
| RTC | 0x0300 |
| STATUS | 0x0400 |
| CONFIG | 0x0700 |
| OUTPUTCONFIG | 0x0800 |
| ELMERTARIF | 0x0B00 |
| CLRELMER | 0x0B80 |

2.2.1 Status

Bázová adresa 0x0400. Detailnější popis proměnných viz. kapitola Datové struktury.

| Offset | Parametr | Čtení(R)/Zápis(W) | Délka v bytech |
|-----------------|--------------|-------------------|----------------|
| 0x0000 | RamErr | R/W | 1 |
| 0x0001 | WDHafs | R | 1 |
| 0x0002 | PWOffs | R | 1 |
| 0x0003 | RecordMode | R/W | 1 |
| 0x0004 | PwrConf | R/W | 2 |
| 0x0005 | Iconf[0] | R/W | 1 |
| 0x0006 | Iconf[1] | R/W | 1 |
| 0x0007 | Iconf[2] | R/W | 1 |
| 0x0008 | Iconf[3] | R/W | 1 |
| 0x0009 | Uconf | R/W | 1 |
| 0x000A | FrConf | R/W | 1 |
| 0x000B | Tconf | R/W | 1 |
| 0x000C | AuxConf | R/W | 1 |
| 0x000D | HDOTrigLevel | R/W | 1 |
| 0x000E | Reserve | R/W | 1 |
| 0x000F – 0x0010 | MaxRecs | R | 4 |
| 0x0011 – 0x0012 | NoRecs | R | 4 |
| 0x0013 – 0x0014 | CurrRecNo | R | 4 |
| 0x0015 – 0x001A | LastRecTime | R | 6 |
| 0x001B – 0x0020 | StartRecTime | R | 6 |
| 0x0021 | Reclnt | R/W | 1 |
| 0x0022 | DeviceNo | R | 1 |
| 0x0023 | NoteBookSize | R | 1 |
| 0x0024 | DeviceType | R | 1 |
| 0x0025 | SoftVersion | R | 1 |
| 0x0026 | Reserve6 | R/W | 1 |

2.2.2 Config

Bázová adresa 0x0700. Všechny registry jsou R/W.

| Offset | Parametr | Délka v bytech |
|-----------------|---------------|----------------|
| 0x0000 + 0x0001 | MTN | 4 |
| 0x0002 + 0x0003 | MTP | 4 |
| 0x0004 | NomPwr | 2 |
| 0x0005 | InputType | 1 |
| 0x0006 | DeviceAddr | 1 |
| 0x0007 | RemoteBdRate | 1 |
| 0x0008 | Rezerva | 4 |
| 0x000A | CANDeviceAddr | 2 |
| 0x000B | NomU | 2 |
| 0x000C + 0x000D | Rezerva | 4 |
| 0x000E | Temp4mA | 2 |
| 0x000F | Temp20mA | 2 |

2.2.3 OutputConfig

Bázová adresa 0x0800. Všechny registry jsou R/W.

| Offset | Parametr | Délka v bytech |
|-----------------|------------|----------------|
| 0x0000 | PulseOut | 1 |
| 0x0001 + 0x0002 | rkWh | 4 |
| 0x0003 + 0x0004 | Limit | 4 |
| 0x0005 + 0x0006 | Hysteresis | 4 |
| 0x0007 | DelayTime | 2 |
| 0x0008 | ReleCgf | 1 |
| 0x0009 + 0x000A | Rezerva | 4 |

2.2.4 RTC

Bázová adresa 0x0300. Všechny registry jsou R/W.

| Offset | Parametr | Délka v bytech |
|--------|------------------|----------------|
| 0x0000 | Rok a měsíc | 2 |
| 0x0001 | Den a hodina | 2 |
| 0x0002 | Minuta a sekunda | 2 |

2.2.5 Identifikace

Bázová adresa 0x0200.

| Offset | Parametr | Délka v bytech |
|--------|---|----------------|
| 0x0000 | DeviceNo (v. č. přístroje) | 2 |
| 0x0001 | DeviceType (typ přístroje) | 2 |
| 0x0002 | Props Type (pro SMY a SMZ pevně hodnota 0x0030) | 2 |
| 0x0003 | SoftVersion (verze fw) | 1 |
| 0x0004 | Remote Addr | 1 |

Parametr DeviceType je rozepsán v následujících tabulkách.

Přístroj SMY, DeviceType

| Typ přístroje | Hodnota |
|--------------------------------------|---------|
| SMY33 - základní typ | 0900 |
| SMY33T – s teplotním vstupem | 0901 |
| SMY33R – s výstupními relé | 0902 |
| SMY33RT – s teplotním vstupem a relé | 0903 |

Je-li přístroj vybaven dálkovou komunikační linkou (Remote), mění se část čísla 09xx takto:

Přístroj SMY, DeviceType přístrojů s linkou Remote

| Typ přístroje | Hodnota |
|---------------|---------|
| SMY33xx CAN | 0Bxx |
| SMY33xx 485 | 0Dxx |
| SMY33xx COM | 0Fxx |

Pravá polovina čísla zůstává dle tabulky výše.

Příklad: SMY33RT 485 má DeviceType = 0D03.

Přístroj SMZ, DeviceType

| Typ přístroje | Hodnota |
|---------------------------------|---------|
| SMZ33 - základní typ | 1100 |
| SMZ33T – s teplotním vstupem | 1101 |
| SMZ33R – s výstupními relé | 1102 |
| SMZ33E – s elektroměrem | 1104 |
| SMZ33ERT – elměr+relé + teplota | 1107 |

Je-li přístroj vybaven Remote komunikační linkou mění se část čísla 11xx takto:

Přístroj SMZ, DeviceType přístrojů s linkou Remote

| Typ přístroje | Hodnota |
|---------------|---------|
| SMZ33xx CAN | 13xx |
| SMZ33xx 485 | 15xx |
| SMZ33xx COM | 17xx |

2.2.6 ElmerTarif

Bázová adresa 0x0B00. Všechny registry jsou R/W.

| Offset | Parametr | Délka v bytech |
|-----------------|----------------------------------|----------------|
| 0x0000 + 0x0005 | ElmerTarif[0] + ElmerTarif[5] | 6 |

2.2.7 ClrElmer

Bázová adresa 0x0B80. Pouze zápis.

| Offset | Parametr | Délka v bytech |
|--------|---|----------------|
| 0x0000 | zápisem konstanty 0x0100 se nuluje elektroměr | 1 |

2.2.8 Aktuální data měřených veličin

Aktuální naměřená data lze číst jako stav „input“-registru (funkce „Read Input Registers“).

| Adresa registru | Parametr | Délka v bytech |
|-----------------|---------------------|----------------|
| 0x0000 | U1 | 2 |
| 0x0001 | U2 | 2 |
| 0x0002 | U3 | 2 |
| 0x0003 | LU | 1 |
| 0x0004 | I1 | 2 |
| 0x0005 | I2 | 2 |
| 0x0006 | I3 | 2 |
| 0x0007 | I4 | 2 |
| 0x0008 | Cos fi 1 | 1 |
| 0x0009 | Cos fi 2 | 1 |
| 0x000A | Cos fi 3 | 1 |
| 0x000B | Fr + Contacts | 1 |
| 0x000C | Temperature | 1 |
| 0x000D | PF1 | 1 |
| 0x000E | PF2 | 1 |
| 0x000F | PF3 | 1 |
| 0x0010 | U12 | 2 |
| 0x0011 | U23 | 2 |
| 0x0012 | U31 | 2 |
| 0x0100+ 0x0101 | P1 | 4 |
| 0x0102 + 0x0103 | P2 | 4 |
| 0x01040+ 0x0105 | P3 | 4 |
| 0x0106 + 0x0107 | Q1 | 4 |
| 0x0108 + 0x0109 | Q2 | 4 |
| 0x010A + 0x010B | Q3 | 4 |
| 0x010C + 0x010D | S1 | 4 |
| 0x010E + 0x010F | S2 | 4 |
| 0x0110 + 0x0111 | S3 | 4 |
| 0x0200 | THDU1 | 1 |
| 0x0201 | THDU2 | 1 |
| 0x0202 | THDU3 | 1 |
| 0x0210 | HarU fáze 1, řád 2 | 1 |
| 0x0211 | HarU fáze 1, řád 3 | 1 |
| ... | ... | ... |
| 0x0227 | HarU fáze 1, řád 25 | 1 |
| 0x0228 + 0x022F | reserved | 16 |
| 0x0230 | HarU fáze 2, řád 2 | 1 |
| 0x0231 | HarU fáze 2, řád 3 | 1 |
| ... | ... | ... |
| 0x0247 | HarU fáze 2, řád 25 | 1 |

| | | |
|-----------------|---------------------|-----|
| 0x0248 + 0x024F | reserved | 16 |
| 0x0250 | HarU fáze 3, řád 2 | 1 |
| 0x0251 | HarU fáze 3, řád 3 | 1 |
| ... | ... | ... |
| 0x0267 | HarU fáze 3, řád 25 | 1 |
| 0x0268 + 0x026F | reserved | 16 |
| 0x0300 | THDI1 | 1 |
| 0x0301 | THDI2 | 1 |
| 0x0302 | THDI3 | 1 |
| 0x0310 | HarI fáze 1, řád 2 | 1 |
| 0x0311 | HarI fáze 1, řád 3 | 1 |
| ... | ... | ... |
| 0x0327 | HarI fáze 1, řád 25 | 1 |
| 0x0328 + 0x032F | reserved | 16 |
| 0x0330 | HarI fáze 2, řád 2 | 1 |
| 0x0331 | HarI fáze 2, řád 3 | 1 |
| ... | ... | ... |
| 0x0347 | HarI fáze 2, řád 25 | 1 |
| 0x0348 + 0x034F | reserved | 16 |
| 0x0350 | HarI fáze 3, řád 2 | 1 |
| 0x0351 | HarI fáze 3, řád 3 | 1 |
| ... | ... | ... |
| 0x0367 | HarI fáze 3, řád 25 | 1 |
| 0x0368 + 0x036F | reserved | 16 |

Bližší popis a kódování jednotlivých parametrů viz popis datové struktury ActAllData.

2.2.9 Stav elektroměru

Stav elektroměru lze číst jako stav „input“-registru (funkce „Read Input Registers“).

| Adresa registru | Parametr | Délka v bytech |
|-----------------|--------------|----------------|
| 0x0400+ 0x0417 | ElmerEnergy | 48 |
| 0x0418+ 0x041A | ElmerClrTime | 6 |
| 0x041B+ 0x0422 | QHMaxPwr | 16 |
| 0x0423+ 0x042E | QHMaxPwrTime | 24 |

Bližší popis a kódování jednotlivých parametrů viz popis datové struktury ElmerActData.

3. Datové struktury

Forma popisu odpovídá konvenci jazyka C. Vícebytové proměnné jsou uloženy v pořadí high-low (nejprve nejvyšší byte, poslední nejnižší byte). Vzhledem k tomu, že struktury obsahují veškeré parametry potřebné k funkci přístroje, nejen v režimu on-line, jsou komentáře jen u těch parametrů, které mají s režimem on-line přímou souvislost.

```
//=====
```

Status :

```
typedef struct {
    uchar RamErr;
        /* struktura pole RamErr:
        /* -----
        /* D7...chyba zalohovani RAM
        /* D6...chyba zalohovani RTC
        /* D5...chyba checksum EPROM (zatim nefunkcni)
        /* D4...
        /* D3...
        /* D2...chyba zalozniho Configu
        /* D1...chyba kal. konstant v EEPROM
        /* D0...chyba checksum EEPROM

    uchar WDHafs;
    uchar PWOffs;
    uchar RecordMode;
    uint PwrConf;

    uchar IConf[4];
    uchar UConf;
    uchar FrConf;
    uchar TConf;
    uchar AuxConf;
    uchar HDOTrigLevel;

    uchar Reserve5;

    ulong MaxRecs;
    ulong NoRecs;
    ulong CurrRecNo;

    uchar LastRecTime[6];
    uchar StartRecTime[6];

    uchar RecInt[2];
    uint DeviceNo;
    uchar NoteBookSize;
    uchar DeviceType; /* SMY, SMZ nepouziva */
    uchar SoftVersion;
    uchar Reserve6;
    ulong THDConf;

} SType; /*Statustype*/
```

```
//=====
```

Config :

```

typedef struct {
    /*pokud neni uvedeno jinak, je v SEEprom */
    ulong Mtn; /*nominalni primarni napeti MTN */
    ulong Mtp; /*prevod MTP, kodovani: */
                /* bity 30-0: nominalni primarni proud MTP */
                /* bit 31: 0...nominalni sekundarni proud MTP 1A */
                /* 1...nominalni sekundarni proud MTP 5A */
    uint NomPwr; /* vykon trafa v kVA */
    uchar InputType; /* konfigurace mericich vstupu */
                /* b0.....RemoteComType : 0 = stand. asynchr. protokol pres COM */
                /* 1 = MODBUS RTU */
                /* b1.....Modem : 0 = neni */
                /* : 1 = je */
                /* b3,2 Parita pro MODBUS 00 - neni */
                /* 01 - suda */
                /* 10 - licha */
                /* b4..... 0 - mereni plne trifazove */
                /* 1 - mereni dle Aarona */
                /* b5..... 0 - mereni plne trifazove - zapojeni hvězda */
                /* 1 - mereni plne trifazove - zapojeni trojuhelnik */
                /* b6.....0 - mereni s MTN */
                /* 1 - primo */
                /* b7..... 1 - prepínatelný přístroj(read-only bit) */
    uchar DeviceAddr; /** adresa (2-253) (pro CAN nema vyznam) *
    uchar RemoteBdRate; // spodni nibble definuje kom. rychlost *
                        // pro standard. as. komunikaci,dle tab *
                        // ; Tabulka rychlosti (pro COM) : *
                        // ; 0...50 Bd - *
                        // ; 1...150 Bd - *
                        // ; 2...300 Bd - *
                        // ; 3...600 Bd - *
                        // ; 4...1200 Bd - *
                        // ; 5...2400 Bd - *
                        // ; 6...4800 Bd - *
                        // ; 7...9600 Bd - *
                        // ; 8...19200 Bd - *
                        // ; 9...38400 Bd - atd. *
                        // ; Tabulka rychlosti (pro CAN) : *
                        // ; 0...5 kbit/s *
                        // ; 1...10 kbit/s *
                        // ; 2...20 kbit/s *
                        // ; 3...50 kbit/s *
                        // ; 4...100 kbit/s *
                        // ; 5...125 kbit/s *
                        // ; 6...200 kbit/s *
                        // ; 7...250 kbit/s *
                        // ; 8...500 kbit/s *
                        // ; 9...800 kbit/s *
                        // ; 10...1000 kbit/s *
    ulong Res0; /* u SMD MTP2 */
    uint CANDeviceAddr; /** adresa (2-1023) pro CAN
    uint NomU; /* nominalni napeti
                // pri primem napeti obsahuje volne zadanou hodnotu v instalaci pristroje
                // pri mereni pres MTN obsahuje informaci o sekundaru trafa, coz je vlastne
                // pristrojem mereny nominal, a zaroven nam umoznuje nastavit jiny sekundar
                // nez 100V -> 100,110,115,120
    uchar Res3;

```

```

    uchar   Res4;
    uchar   Res5;           /* az sem 24 bytu */
    int     Temp4mA;       /* teplota pri 4 ma ve st. C*/
    int     Temp20mA;      /* teplota pri 20 mA ve st. C */

} CType;           /*Configuration 28 byte*/
// =====

OutputConfig :

typedef struct { /*pokud neni uvedeno jinak, je v SEEPROM*/
    uchar PulseOut; /* b1,b0... typ prace PO1*/
                    /* b5,b4... typ prace PO2*/
                    /* 00... vysilat praci CINNY ODBER */
                    /* 01... vysilat praci CINNA DODAVKA */
                    /* 10... vysilat praci JALOVA INDUKTIVNI */
                    /* 11... vysilat praci JALOVA KAPACITNI */

                    /* b3... 1 vysilat, 0 nevysilat PulseOut1*/
                    /* b7... 1 vysilat, 0 nevysilat PulseOut2*/
                    /* b2... 1 - povolit synchronizaci casu od vstupu (jen SMZ)*/
                    /* b2... 0 - zakazat synchronizaci casu od vstupu (jen SMZ)*/

    uint rkWh[2]; /* pocet impulsu na 100Wh */
    uint Limit[2]; /* hodnota limitu v setinach procenta */
                    /* je-li hodnota 0xFFFF je funkce rele neaktivni */

    uint Hysteresis[2]; /* hystereze v setinach % */
                    /* hysterezni pasmo je Limit +- Hysteresis */

    uchar DelayTime[2]; /* casova prodleva */
                    /* 0-96 v 5-ti sekundach */
                    /* 97-200 v 15-ti sekundach */
                    /* 201-252 ve 30-ti sekundach */

    uchar ReleCfg[2]; /* b1,b0 - cislo faze */
                    /* 00 - faze 1 */
                    /* 01 - faze 2 */
                    /* 10 - faze 3 */
                    /* 11 - trifazova hodnota ( ma smysl pro vykony, mozna 3PF) */
                    /* b5,b4,b3,b2 - tabulka velicin */
                    /* 0000 - U */
                    /* 0001 - I */
                    /* 0010 - PF */
                    /* 0011 - F */
                    /* 0100 - T */
                    /* 0101 - P vykon */
                    /* 0110 - Q vykon */
                    /* 0111 - S vykon */
                    /* 1000 - cos */
                    /* 1001 - THDU */
                    /* 1010 - THDI */
                    /* b7,b6 - stav sepnuti */
                    /* 00 - nad limitem sepnout */
                    /* 01 - nad limitem rozepnout */
                    /* 10 - pod limitem sepnout */
                    /* 11 - pod limitem rozepnout */

    uchar Res[3];
} OType; /*Outputs Configuration - 20 byte*/
// =====

```

ElmerTarif :

```

uchar ElmerTarif[6];
    /* kazdy byte obsahuje dvojici bitu, urcujici c.tarifu dane hodiny */
    /* pro hodinu 0 ...byte 0, bity 1,0....00=tarif0, 01=tarif1, 10= tarif2 */
    /* pro hodinu 1 ...byte 0, bity 3,2....00=tarif0, 01=tarif1, 10= tarif2 */
    /* .....*/
    /* pro hodinu 4 ...byte 1, bity 1,0....00=tarif0, 01=tarif1, 10= tarif2 */
    /* .....*/
    /* .....*/
    /* .....*/
    /* .....*/
    /* pro hodinu 23 ...byte 5, bity 7,6....00=tarif0, 01=tarif1, 10= tarif2 */
    /* pokud je tarif=11, znamena to tarify nespecifikovany, */

//=====

```

ElmerActData :

```

xdata union
{
    uchar Arr[48];                /* pro nulovani */
    ulong St[3][4];              /* indexy -[tarif][velicina] */
} ElmerEnergy; // 48 bytes
    /* obsahuje stav 3f-elmeru od posledniho vynulovani */
    /* ve Wh v poradi AE,AI,RL,RC pro tri ruzne tarify */
    /* pozor, MTP a MTN zde neni zahrnuto */

xdata union
{
    uchar Arr[6];                /*pro SetTime Rac, 6bytes(Y,Mo,D,H,Mi,S*/
} ElmerClrTime;                /* datum nulovani elektromeru a 1/4hP maxim */
/* delka 6 byte */

xdata long QHMaxPwr[4];          /*soucasne hodnoty maxim výkonu P1,P2,P3,3P*/
                                /* hodnota v sobě nezahrnuje MTP a MTN */
                                /* delka 16 byte */

xdata union
{
    uchar Arr[6];                /*pro SetTime Rac, 6bytes(Y,Mo,D,H,Mi,S*/
    TimeType St;                /*pro C*/
} QHMaxPwrTime[4];             /*cas vyskytu maxima výkonu*/
                                /* delka 24 byte */

/* celkem delka 94 byte */
//=====

```

ActAllData :

```

typedef struct
{
    uchar RamErr;           // RamErr je kopii ze Statusu
    uint U[3];             // fazova napeti
                          // U je v desetinach voltu 0x0000 - 0V, 0x0001 - 0,1V, .... 0xFFFF - PwrOff
                          // skutecne napeti je dane obsahem promenne Mtn (CType)
                          // if(Mtn == 0xFFFFFFFF) je v U[n] obsazena skutecna hodnota napeti v 0,1V
                          // if(Mtn != 0xFFFFFFFF) je hodnotu v U[n] nutne vynasobit hodnotou Mtn (Config)
                          // a podelit hodnotou (NomU)
    uint LU;               // pro SMZ, SMY nemá význam
    int I[4];
                          // Zmerene I se predava takto:
                          // if(I[n] == 0x0000) -> 0A
                          // if(I[n] == 0x3E80) -> Inom(5A)
                          // if(I[n] == 0x7FFF) -> PwrOff
                          // obdobne pro zaporne hodnoty, PwrOff prezentovan vzdy jako 0x7FFF
                          // hodnota proudu nezahrnuje převod Mtp, nutno vynásobit
                          // převodem Mtp (struktura Config)
                          // ctvrtá hodnota nemá pro SMZ, SMY význam
    char PF[3];           // fazove power-factors
                          // Zmereny PF se predava v takto:
                          // if(PF[] == 0) -> cos fi = 0L
                          // .
                          // .
                          // if(PF[] == 90) -> cos fi = 0.90L
                          // .
                          // .
                          // if(PF[] == 99) -> cos fi = 0.99L
                          // if(PF[] == 100) -> cos fi = 1.00
                          // if(PF[] == -99) -> cos fi = 0.99C
                          // .
                          // .
                          // if(PF[] == -90) -> cos fi = 0.90C
                          // .
                          // .
                          // if(PF[] == -1) -> cos fi = 0.01C
                          // if(PF[] == -100) -> cos fi = 0C
    uchar Fr;
                          // Zmerena Fr se predava v takto:
                          // if(Fr == 0) -> 37.2Hz po 0,1 Hz
                          // if(Fr == 1) -> 37.3Hz
                          // .
                          // .
                          // if(Fr == 177) -> 54.9Hz
                          // if(Fr == 178) -> 55.0Hz po 0,5 Hz
                          // if(Fr == 179) -> 55.5Hz
                          // if(Fr == 180) -> 56.0Hz
                          // .
                          // .
                          // if(Fr == 254) -> 93.0Hz
                          // if(Fr == 255) -> PwrOff
    uchar T;             // predava se hodnota v desetinach mA, skutecna hodnota teploty
                          // se musi spočítat s pomoci hodnot Temp4mA a Temp20mA ze struktury Config

    uchar Contacts;     // bity 0,1 signalizují stav relé 1,2
    char Kos[3];        // fazove cos-phi, kodovani viz PF
    uint Upp[3];        // hodnota sdruzeného napeti, kodovano jako U

```

```

long P[3];      // kódování : 1W ~ 0x4E200
                // je-li hodnota rovna 0x7FFFFFFF, jedná se o neplatnou hodnotu
                // hodnota nezahrnuje převody MTP, MTN-nutno vynásobit těmito převody
long Q[3];      // obdobně jako P
long S[3];      // obdobně jako P
THDAHar thdHar[2]; // THD a harmonické, [0]...pro napětí , [1]... pro proud, kódování níže
} ActAllData;   // total 218 bytes

```

```

typedef struct{
    uchar THD[3];      /* Kodovani : */
                    /* 0 az 100 ... po 0.5%, tj. 0 az 50% */
                    /* 101 az 200... po 2,5%, tj. 50,5 az 300% */
                    /* 201 az 254... po 10%, tj. 310 az 840% */

    uchar Har[3][24]; /* Kodovani : */
                    /* 0 az 50 ... po 0.1%, tj. 0 az 5% */
                    /* 51 az 70... po 0.5% tj. 5.5% az 15% */
                    /* 71 az 90... po 2.5%, tj. 17.5 az 65% */
                    /* 91 az 126... po 5%, tj. 70 az 240% */

}THDAHar;

```

```
//=====
```