
Program spotProcessor v. 2.3

Struktura konfiguračního souboru a související nastavení

08.11.2023, ENcontrol s.r.o.

1 Obsah

1	Obsah.....	1
2	Popis programu spotProcessor	3
3	Struktura konfiguračního souboru	4
3.1	Sekce [general]	4
3.2	Sekce [schedule].....	7
3.3	Sekce [reaction].....	11
4	Struktura souboru regAddr.conf	14
5	Implementace protokolu Modbus	15
5.1	Protokol Modbus TCP.....	15
5.2	Protokol Modbus RTU	15
5.3	Implementované funkce Modbus	16
5.4	Čísla registrů pro vyčítání hodnot z programu spotProcessor	16
5.4.1	Digitální výstupy	16
5.4.2	Analogové výstupy – aktuální stav	18
5.4.3	Analogové výstupy – historie a předpovědi	19
5.5	Reagování na zápis digitálních a analogových vstupů.....	21
5.5.1	Digitální vstup.....	21
5.5.2	Analogový vstup	23
6	Použití makrojazyka.....	25
6.1	Identifikace zařízení.....	25
6.2	Průměrování měření.....	25
6.3	Telefonní čísla a emailové adresy.....	25
6.4	Aktivace /deaktivace plánů a reakcí.....	26
6.5	Neimplementované příkazy	26
6.6	Nově implementované příkazy.....	26
6.6.1	Nově implementované příkazy (mimo Modbus).....	27
6.6.2	Nově implementované příkazy pro Modbus.....	31

7	Specifika řízení regulátoru Wattrouter	35
8	Vysvětlení obsahu makra SPOTPRICE_EXEC.mac.....	36
9	Další související soubory.....	38
10	Další související programy	39
10.1	Zasílání emailů.....	39
10.2	Zasílání SMS zpráv	41

2 Popis programu spotProcessor

Program spotProcessor je klonem programu encProcessor společnosti ENcontrol. Program encProcessor je **určen pro malé jednotky s operačním systémem Linux a ovládá připojená zařízení (spotřebiče a čidla)**. Zařízení mohou být **připojena přes LAN (ethernet, WiFi) nebo bezdrátovým protokolem Jennic**. Program provádí řízení zařízení podle zadaných časových plánů a reaguje na různé události. Program umožňuje, **spínání spotřebičů, měření různých elektrických veličin, detekci signálů z čidel, zasílání e-mailů a SMS**.

Program **spotProcessor** je navíc rozšířen o:

- **stahování aktuálních spotových cen** ze stránek operátora trhu s energiemi v ČR (OTE)
- vyhodnocování spotových cen a řízení připojených zařízení na jejich základě
- komunikaci s dalšími zařízeními přes protokoly Modbus TCP nebo Modbus RTU
- komunikace s externími reléovými moduly přes USB, ethernet nebo WiFi.

Program umí reagovat zasíláním signálů řízených zařízení na aktuální spotové ceny na základě předem definovaných podmínek. Těmito podmínkami může být například výše ceny pod dolním limitem, nad horním limitem nebo výše ceny NAD nebo POD jakoukoliv jinou zadanou hodnotou.

Pomocí protokolu **Modbus TCP nebo RTU může jednotka spotProcessor**:

- řídit připojená zařízení (resp. zapisovat hodnoty do jejich registrů); těmito zařízeními může být regulátor Wattrouter, střídače nebo jiná „chytrá“ zařízení
- vyčítat hodnoty registrů z připojených zařízení a podle zjištěných hodnot dále reagovat definovaným způsobem
- poskytovat k případnému vyčítání stavové (diskrétní) a hodnotové (analogové) registry pro jiná zařízení. Například je tak možné vyčítat ceny nebo aktuální stav řízení externích zařízení.

3 Struktura konfiguračního souboru

Konfigurační soubor programu spotProcessor je prostý textový soubor. Jeho název je libovolný, obvykle `spotProcessor.conf`.

Soubor může obsahovat tři druhy sekcí:

- [general]
- [schedule]
- [reaction]

Povinná sekce je pouze [general], ostatní sekce jsou nepovinné. Sekce [general] může být v souboru uvedena pouze jednou, ostatní v libovolném počtu. Na pořadí sekcí nezávisí – mohou se libovolně řadit.

V každé sekci jsou uvedeny jednotlivé parametry ve tvaru `PARAMETR=HODNOTA`. Parametr i hodnota je vždy jedno slovo (bez mezer) a kolem rovnítka nesmí být žádné další bílé znaky (mezery). Na velikosti písmen obecně u názvů parametrů záleží a u hodnot nezáleží. Výjimkou je hodnota udávající název nebo cestu k nějakému souboru (např. makru) – tam na velikosti písmen záleží také. Na pořadí parametrů v rámci jedné sekce nezáleží. Vyskytuje-li se parametr v jedné sekci vícekrát, vždy platí poslední uvedená hodnota.

V souboru se kdekoliv mohou vyskytovat komentáře. Jedná se o řádky, které jsou uvozeny na počátku znakem '#'. Tyto řádky se pak při zpracování ignorují.

3.1 Sekce [general]

Typická podoba sekce [general]:

```
[general]
ACTIVE_IFACE_NAME=wlan0
UNIT_PORTNUM_TCP=50150
STATION_PORTNUM_TCP=50151
REG_ADDR_FILE=/media/extended/spotProcessor/regAddr.conf
NETWORK_CHECK_FILE=/media/extended/spotProcessor/ipAddrStateUp.sh
MODBUS_TCP_ALLOWREQ=true
MODBUS_RTU_ALLOWREQ=true
MODBUS_TCP_SLAVEID=2
MODBUS_RTU_SLAVEID=3

SERIAL_DEVICE=/dev/ttyUSB0
SERIAL_BAUDRATE=115200
SERIAL_VMIN=0
SERIAL_VTIME=4
SERIAL_PARBITS=8N1
SERIAL_HWFLOW=false
SERIAL_SWFLOW=false
```

```

LOG_LEVEL=ERRO
TRIES_COUNT=3
SERIES_COUNT=2
SLEEP_TIME_LOOP=300
SLEEP_TIME_SERIES=3000
TIMEOUT_READ=500
TIME_SHIFT=300
REACTION_DELAY=30
FILE_ERROR_TERMINATE=0
STARTUP_MAC=/media/extended/spotProcessor/STARTUP.mac
SERVICE_MAC=/media/extended/spotProcessor/spotProcessor_SERVICE.mac
SERVICE_LOG=/media/extended/spotProcessor/spotProcessor_SERVICE.log
EMAIL_BODY_FILE=/media/extended/email-body.txt
EMAIL_SEND_FILE=/opt/encontrol/email-send.sh
SMS_BODY_FILE=/media/extended/sms-body.txt
SMS_SEND_FILE=/opt/encontrol/sms-send.sh

SPOT_PRICE_CURRENCY=EUR
MIN_SPOT_PRICE=80.0
MAX_SPOT_PRICE=95.0
TABLE_FILE_FORMAT=HTML
TABLE_FILE_PREFIX=/media/extended/spotProcessor/denni-trh-tab_
TABLE_FILE_POSTFIX=.txt

```

Význam a hodnoty parametrů sekce [general].

Parametr	Popis	Příklad hodnoty
ACTIVE_IFACE_NAME	Název síťového zařízení, interface (jméno speciálního souboru v OS Linux)	wlan0, eth0, eth1
UNIT_PORTNUM_TCP	Výchozí hodnota čísla portu klienta v komunikaci přes protokol ENcontrol	50150
STATION_PORTNUM_TCP	Výchozí hodnota čísla portu serveru v komunikaci přes protokol ENcontrol	50151
REG_ADDR_FILE	Cesta k souboru s údaji o adresách a portech připojených zařízení	/media/extended/spotProcessor/regAddr.conf
NETWORK_CHECK_FILE	Cesta ke skriptu, který kontroluje dostupnost síťového připojení	/media/extended/spotProcessor/ipAddrStateUp.sh
MODBUS_TCP_ALLOWREQ	Umožní nebo zakáže zpracování příchozích požadavků přes Modbus TCP	true false
MODBUS_RTU_ALLOWREQ	Umožní nebo zakáže zpracování příchozích požadavků přes Modbus RTU	true false
MODBUS_TCP_SLAVEID	Číslo SlaveID pro pasivní komunikaci vyvolanou z jiných zařízení přes protokol Modbus TCP	2
MODBUS_RTU_SLAVEID	Číslo SlaveID pro pasivní komunikaci vyvolanou z jiných zařízení přes protokol Modbus RTU	3
SERIAL_DEVICE	Název rozhraní pro sériovou komunikaci (jméno speciálního souboru v OS Linux)	/dev/ttyUSB0 (pro linku RS485 s USB redukcí) /dev/ttyS0 (pro linku RS232)

SERIAL_BAUDRATE	Rychlost datového toku. Možné hodnoty jsou: 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 115200, 230400	115200 (pro většinu zařízení) 9600 (pro USB relé. modul)
SERIAL_VMIN	Minimální počet znaků zprávy. Obvykle 0	0
SERIAL_VTIME	Max. počet desetin sekundy, po které se čeká na další Byty v jedné zprávě (hodnota 10 = 1 sekunda) Ujistěte se, že součin (SERIAL_VTIME * 0,1 * TRIES_COUNT) > (SLEEP_TIME_LOOP / 1000)	4
SERIAL_PARBITS	Nastavení sériového portu (8 datových bitů, žádná parita, 1 stop-bit)	8N1
SERIAL_HWFLOW	Hardwarové řízení toku ano/ne	false
SERIAL_SWFLOW	Softwarové řízení toku ano/ne	false
LOG_LEVEL	Úroveň logování. Možné hodnoty: - NONE (žádné logování) - CRIT (pouze nezbyt. údaje a krit. chyby) - ERRO (navíc nekritické chyby) - WARN (navíc upozornění) - ALL (logování všeho včetně debugování) Pro správný běh programu umožňující řízení podle spotových cen nesmí být úroveň „NONE“	ERRO
TRIES_COUNT	Počet pokusů o získání odpovědi z ovládaného zařízení při vykonávání příkazu v rámci jedné série. Pokud odpověď ze zařízení není přečtena po X pokusech udaných tímto parametrem, znovu se posílá příkaz a zahájí se nová série pokusů.	3
SERIES_COUNT	Počet sérií pokusů o získání odpovědi na vykonání jednoho příkazu. Při neúspěšném získání se daný příkaz posílá tolikrát, kolik udává tento parametr.	2
SLEEP_TIME_LOOP	Zpracování plánovaných akcí a reakcí probíhá v nekonečné smyčce. Parametr udává počet milisekund čekání na konci každé iterace. Hodnota by neměla být 0 (1000 = 1 sekunda).	500
SLEEP_TIME_SERIES	Počet milisekund mezi sériemi při neúspěšném vykonávání příkazu (4000 = 4 sekundy)	4000
TIMEOUT_READ	Počet milisekund pro 1 pokus o čtení ze síťového portu. Po vypršení je předpokládáno, že žádná zpráva není na síťovém rozhraní k dispozici.	500
TIME_SHIFT	Pokud se plánovaná akce (pouze <i>schedule</i> , ne <i>reaction</i>) nepodaří vykonat, akce se ještě max. 2x přeplánuje na později. Parametr udává počet sekund posunu akce do budoucna (300 = 5 minut). Je aplikováno pouze v případě, že parametr je kratší než perioda opakování vlastního plánu	300
REACTION_DELAY	Po startu jednotky a tedy resetu komunikační sítě některá zařízení aktivně posílají hlášení o svém stavu. Na to se mohou vázat definované reakce, které ale při resetu nejsou žádané. Parametr udává počet sekund po startu programu, po který se všechna hlášení pro reakce ignorují.	30

FILE_ERROR_TERMINATE	Počet tolerovaných chyb práce se soubory (např. zámky). Je-li dosaženo limitu, program je ukončen. Jednotka spotProcessor ho pak automaticky po nějaké době opět spustí. Hodnota „0“ znamená, že tyto chyby budou ignorovány a program se nebude ukončovat.	3
STARTUP_MAC	Soubor s příkazy makrojazyka, který se spouští hned po startu programu	/media/extended/STARTUP.mac
SERVICE_MAC	Pomocný soubor s příkazy makrojazyka, který se spouští vždy, když program dostane signál SIGUSR2 (využíván např. při spouštění uživatelských příkazů makrojazyka)	/media/extended/spotProcessor_SERVICE.mac
SERVICE_LOG	Pomocný soubor, kam program zapisuje stav zařízení vždy, když dostane signál SIGUSR1 (využíván při zobrazení aktuálního stavu připojených zařízení)	/media/extended/spotProcessor_SERVICE.log
EMAIL_BODY_FILE	Soubor, kam program zapisuje tělo emailu, který se má odeslat. Adresář a soubor musí být dostupné pro zápis.	/media/extended/email-body.txt
EMAIL_SEND_FILE	Spustitelný soubor se skriptem, který odesílá emaily pomocí externího programu exim4.	/opt/encontrol/email-send.sh
SMS_BODY_FILE	Soubor, kam program zapisuje skript pro odeslání SMS. Adresář a soubor musí být dostupné pro zápis.	/media/extended/email-body.txt
SMS_SEND_FILE	Spustitelný soubor se skriptem, který odesílá SMS zprávy pomocí externího programu minicom.	/opt/encontrol/email-send.sh
SPOT_PRICE_CURRENCY	Měna, ve které jsou na stránkách OTE udávány spotové ceny	EUR
MIN_SPOT_PRICE	Dolní limit spotové ceny pro uplatnění řízení (desetinná tečka a max. 2 desetinná místa)	65.3
MAX_SPOT_PRICE	Horní limit spotové ceny pro uplatnění řízení (desetinná tečka a max. 2 desetinná místa)	87.15
TABLE_FILE_FORMAT	Formát souboru stahovaného ze stránek OTE	HTML
TABLE_FILE_PREFIX	Cesta a první část jména souboru, do kterého se ukládají stahované hodnoty	/media/extended/spotProcessor/denni-trh-tab_
TABLE_FILE_POSTFIX	Druhá část jména souboru. Mezi první a druhou část se vkládá aktuální datum	.txt

3.2 Sekce [schedule]

V konfiguračním souboru může být libovolný počet sekcí [schedule]. Pouze se nesmí opakovat jejich ID (ScheduleID). V nich se nastavují časové plány.

Tři typické příklady sekce [schedule]:

```
[schedule]
#Ziskani aktualni spotove ceny
ScheduleID=10
Active=true
RelationType=Indicator
Action=Measure
MaxTimeOn=
MaxTimeOff=
Satellite=1
```

```
OrderNum=1
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=true
RepeatTime=15min
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 12:00:00
DoDateTo=01.01.2033 12:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:00

[schedule]
#Spusteni makra pro reagovani na cenu
ScheduleID=20
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_EXEC.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=true
RepeatTime=1min
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 12:00:15
DoDateTo=01.01.2033 12:00:15
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:15

[schedule]
#Pravidelne zasilani Modbus prikazu
ScheduleID=210
Active=true
RelationType=Modbus
Action=06
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=7
IndicatorType=TCP
High=1000
Low=
```



```
Repeatable=true
RepeatTime=1min
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 12:00:30
DoDateTo=01.01.2033 12:00:30
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:30
```

Význam a hodnoty parametrů sekce [schedule].

Parametr	Popis	Příklad hodnoty
ScheduleID	Libovolné přiřazené číslo	10
Active	Je-li hodnota „true“, plán se vykonává, jinak ne.	true false
RelationType	Hodnota může být: <ul style="list-style-type: none"> - Device - Indicator (pak musí být zároveň vyplněn i parametr IndicatorType) - Modbus - Macro 	Device Indicator Modbus Macro
Action	Specifikace příkazu. Všechny možné příkazy jsou uvedeny ve vedlejším sloupci. V případě příkazu „Setbound“ musí být vyplněny i parametry High a Low. Trojfázové měření není implementováno. V případě RelationType=Modbus je v tomto parametru uvedeno číslo Modbus funkce (05 nebo 06). V případě RelationType=Macro je v tomto parametru uvedena cesta k souboru s programovým makrem.	TurnOn TurnOff Pulse Measure Clear Checkstate Setbound Regulate CheckRegulation 05, 06 (pro MODBUS) /media/extended/macro
MaxTimeOn	Specifikace bezpečnostní konstanty maximální doby sepnutí spotřebiče. Jedná se o přiřazené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h
MaxTimeOff	Specifikace bezpečnostní konstanty maximální doby vypnutí spotřebiče. Jedná se o přiřazené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h
Satellite	Číslo ovládaného satelitu, pod kterým je v modemu zaregistrován. U zařízení Modbus se jedná o číslo SlaveID	3
OrderNum	Číslo logického zařízení v ovládaném satelitu. U zařízení Modbus se jedná o číslo registru (dekadicky)	2

IndicatorType	Specifikace typu indikátoru. Musí být vyplněn, je-li hodnota RelationType „Indicator“. Možné hodnoty jsou ve vedlejším sloupci. U zařízení Modbus se jedná o specifikaci protokolu (TCP nebo RTU)	Temperature Humidity Light Other Spotprice TCP RTU
High	Horní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound). U zařízení Modbus jde o 16ti bitové celé číslo v rozsahu -32768 až 32767 nebo 0 až 65535 zapisované do registru (funkce 05 a 06). Může být zadáno i hexadecimálně	21.5 (desetinná tečka) 1 (nebo 0) pro funkci 05 1000 -1000 0x03e8
Low	Dolní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound). U zařízení Modbus a funkce 01 ze zadat počet registrů (požadovaných bitů)	20.5 4
Repeatable	Jedná se o jednorázový plán (false) nebo opakovaný (true)?	true false
RepeatTime	Je-li hodnota Repeatable nastavena na „true“, musí být vyplněno. Možné hodnoty jsou uvedeny ve vedlejším sloupci. Nejkratší možný interval je 1 minuta (hodnota „1min“). Je-li nutné akci opakovat v kratších intervalech, je nutné vytvořit více plánů a ty naplánovat na různé vteřiny při periodě spouštění po 1 minutě. Příklad: potřebujeme nějakou akci vykonávat po 15 vteřinách. Založíme 4 samostatné plány s následujícími hodnotami StartDate: <ul style="list-style-type: none"> - StartDate=01.01.2023 12:00:00 - StartDate=01.01.2023 12:00:15 - StartDate=01.01.2023 12:00:30 - StartDate=01.01.2023 12:00:45 Všechny ostatní hodnoty budou mít shodné.	1min 2min 5min 10min 15min 30min 1h 2h 6h 12h 1d 2d 5d 1t 2t 1mes 2mes 6mes 1r
RepeatWeek	Nepovinné pole. Uplatňuje se, je-li hodnota Repeatable nastavena na „true“. Není-li vyplněno, uvažuje se PO-NE. Možné hodnoty jsou uvedeny ve vedlejším sloupci.	PO UT ST CT PA SO NE PO-PA SO-NE PO-NE
DoDateFrom	Od jakého času je plán platný. Formát datumu a času je závazný a pole je povinné .	01.01.2017 00:00:00
DoDateTo	Do jakého času je plán platný. Formát datumu a času je závazný a pole je povinné .	01.01.2020 00:00:00
ExceptDateFrom	Výjimka (vynechání) OD. Formát datumu a času je závazný a pole je nepovinné.	01.07.2016 00:00:00
ExceptDateTo	Výjimka (vynechání) DO. Formát datumu a času je závazný a pole je nepovinné.	01.09.2016 00:00:00

StartDate	Čas vykonání akce. Při opakování čas první akce. Od tohoto času se odvozují intervaly opakování. Lze specifikovat s přesností na vteřiny. Formát datumu a času je závazný a pole je povinné .	10.05.2017 20:15:30
-----------	--	---------------------

3.3 Sekce [reaction]

Dva příklady typické podoby sekce [reaction]:

```
[reaction]
#Popis reakce - komentar
ReactionID=1
Active=true
StarterEvent=TurnOn
StarterSatellite=3
StarterOrderNum=5
ActionType=Device
ActionEvent=TurnOff
ActionMaxTimeOn=
ActionMaxTimeOff=5m
ActionSatellite=2
ActionOrderNum=1
ActionIndicatorType=Light
ActionHigh=
ActionLow=
```

```
[reaction]
#Popis reakce - komentar
ReactionID=2
Active=true
StarterEvent=05
StarterSatellite=1
StarterOrderNum=120
ActionType=Modbus
ActionEvent=06
ActionMaxTimeOn=
ActionMaxTimeOff=
ActionSatellite=3
ActionOrderNum=8
ActionIndicatorType=RTU
ActionHigh=1000
ActionLow=
```

Význam a hodnoty parametrů:

Parametr	Popis	Příklad hodnoty
ReactionID	Libovolné přiřazené číslo	10
Active	Je-li hodnota „true“, reakce je aktivní. Jinak ne.	true false
StarterEvent	Specifikace typu hlášení, na které se má reagovat. U pasivní komunikace přes Modbus se jedná o číslo funkce zápisu diskretních nebo analogových hodnot, která se testuje (05, 06, 15 nebo 16)	TurnOn TurnOff Pulse 05 06 15 16
StarterSatellite	Číslo satelitu, na který se reaguje (vysílá prvotní signál) U pasivní komunikace přes Modbus se jedná o hodnotu, která se testuje, že má být zapsána (u funkce 06 například 1000; u funkce 05 například 1 = ZAPNUTO)	3 1 0 1000
StarterOrderNum	Číslo logického zařízení satelitu, na který se reaguje (vysílá prvotní signál) U pasivní komunikace přes Modbus se jedná o číslo registru pro zápis diskretních nebo analogových hodnot, které se testuje	2
ActionType	Jakému typu zařízení se má poslat příkaz jako reakce. Hodnota může být „Device“, „Indicator“ nebo „Macro“. Je-li hodnota „Indicator“, musí být zároveň vyplněn i parametr ActionIndicatorType. U komunikace přes Modbus je vyplněno „Modbus“.	Device Indicator Macro Modbus
ActionEvent	Specifikace příkazu, který se posílá jako reakce. V případě příkazu „Setbound“ musí být vyplněny i parametry ActionHigh a ActionLow. Trojfázové měření není implementováno. V případě ActionType=Macro musí být v parametru cesta k souboru s daným makrem. U komunikace přes Modbus je vyplněno číslo funkce, která se má použít (05, 06, 15 nebo 16)	TurnOn, TurnOff, Pulse Measure, Clear Checkstate Setbound Regulate CheckRegulation /media/extended/macro 05, 06, 15, 16
ActionMaxTimeOn	Specifikace bezpečnostní konstanty maximální doby sepnutí spotřebiče. Jedná se o přiřazené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h
ActionMaxTimeOff	Specifikace bezpečnostní konstanty maximální doby vypnutí spotřebiče. Jedná se o přiřazené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h

ActionSatellite	Číslo ovládaného satelitu, pod kterým je v modemu zaregistrován U komunikace Modbus se jedná o SlaveID	3
ActionOrderNum	Číslo logického zařízení v ovládaném satelitu U komunikace Modbus se jedná o číslo (prvního) registru	2 8 120
ActionIndicatorType	Specifikace typu ovládaného indikátoru. Musí být vyplněn, je-li ActionType typu „Indicator“ . Možné hodnoty jsou ve vedlejším sloupci. U komunikace Modbus je vyplněn typ protokolu (TCP nebo RTU)	Temperature Humidity Light Other TCP RTU
ActionHigh	Horní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound) U komunikace Modbus se jedná o zapisovanou hodnotu (dekadicky)	21.5 1 1000
ActionLow	Dolní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound)	20

4 Struktura souboru regAddr.conf

Pro účely komunikace s dalšími zařízeními přes protokol TCP je nutné programu definovat IP adresy a čísla portů, na kterých zařízení naslouchají. K tomuto účely slouží soubor *regAddr.conf*. Jeho umístění je definováno v parametru REG_ADDR_FILE v sekci [general] v hlavním konfiguračním souboru (viz. kapitola 4 Sekce [general]).

Příklad obsahu souboru a význam jednotlivých sloupců tabulky:

```
#Config file for the service spotProcessor
#Registration file containing info about ENcontrol stations
#
##Table header
#<satNo><iface> <IpAddr> <MacAddr> <portNumTCP> <portNumUDP>
##Table records
1 eth0 123.45.67.89 00:0d:b9:27:6e:2c 502 50161
2 wlan0 10.0.1.59 b8:27:eb:6b:67:32 50151 50161
3 eno1 10.0.1.60 b8:27:eb:e9:6a:c9 50151 50161
17 eth0 10.0.1.57 00:0d:b9:27:6e:2c 50151 50161
```

- satNo*: číslo zařízení ENcontrol. V případě komunikace Modbus se jedná o SlaveID připojeného Modbus zařízení.
- iface*: Název síťového rozhraní, přes které se má s připojeným zařízením komunikovat. Obvykle to je při drátovém spojení *eth0* a při bezdrátovém *wlan0*
- IpAddr*: IP adresa zařízení ve formátu IPv4
- MacAddr*: MAC adresa zařízení (používá se pouze při komunikace se zařízeními přes protokol ENcontrol)
- portNumTCP*: Číslo portu pro TCP komunikaci. U ENcontrol zařízení to je obvykle 50151 a u Modbus zařízení 502
- portNumUDP*: Číslo portu pro UDP komunikaci. U ENcontrol zařízení to je obvykle 50161

Řádky uvozeny znakem '#' jsou pomocné komentáře a jsou programem ignorovány.

5 Implementace protokolu Modbus

Program spotProcessor obsahuje podporu protokolu Modbus – a to přes TCP (**Modbus TCP**) a přes sériovou linku (RS-485, **Modbus RTU**). Může komunikovat jak **aktivně**, například při zasílání příkazů do připojených zařízení, tak i **pasivně**, například když připojená zařízení vyčítají z programu stavy řízení nebo jiné hodnoty.

Pro umožnění pasivní komunikace je nutné nastavit v hlavním konfiguračním souboru parametry MODBUS_TCP_ALLOWREQ, resp. MODBUS_RTU_ALLOWREQ na hodnotu „true“ a dále hodnoty SlaveID v parametrech MODBUS_TCP_SLAVEID, resp. MODBUS_RTU_SLAVEID – viz. kapitola 3.1 Sekce [general].

5.1 Protokol Modbus TCP

Nastavení IP adres a portů pro Modbus TCP se provádí v konfiguračním souboru regAddr.conf – viz. kapitola 7 Struktura souboru regAddr.conf. Parametry komunikace přes Modbus TCP se nastavují v hlavním konfiguračním souboru. Jedná se o následující volby:

- MODBUS_TCP_ALLOWREQ: Je-li hodnota *true*, pak program bude reagovat na příchozí požadavky Modbus TCP
- MODBUS_TCP_SLAVEID: Příchozí požadavky, které budou směřovány na jiné SlaveID, budou programem ignorovány.

5.2 Protokol Modbus RTU

Parametry sériové komunikace se nastavují v hlavním konfiguračním souboru. Jedná se o následující volby:

- MODBUS_RTU_ALLOWREQ: Je-li hodnota *true*, pak program bude reagovat na příchozí požadavky Modbus RTU
- MODBUS_RTU_SLAVEID: Příchozí požadavky, které budou směřovány na jiné SlaveID, budou programem ignorovány.
- SERIAL_DEVICE: Jméno speciálního souboru je obvykle */dev/ttyUSB0*
- SERIAL_BAUDRATE: Možné hodnoty rychlosti komunikace jsou 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 115200 a 230400.
- SERIAL_VMIN: Obvykle je nutné nastavit hodnotu 0.
- SERIAL_VTIME: U pomalých zařízení se nastavuje vyšší hodnota, ale obvykle stačí hodnoty v rozmezí 1 – 3.
- SERIAL_PARBITS: V programu je implementováno pouze nastavení „8N1“ – tj. 8 datových bitů, žádná parita, 1 stop-bit.
- SERIAL_HWFLOW: V programu je implementováno pouze nastavení *false*.
- SERIAL_SWFLOW: V programu je implementováno pouze nastavení *false*.

5.3 Implementované funkce Modbus

Pro aktivní komunikaci (tedy zasílání z programu do připojených zařízení) jsou implementovány následující funkce:

- **01 – čtení digitálního výstupu:** používá se pro přečtení stavu výstupu a případnou okamžitou reakci – viz. příkaz MDB01 v kapitole 6.6 Nově implementované příkazy
- **03 – čtení analogového výstupu:** používá se pro přečtení hodnoty výstupu a případnou okamžitou reakci – viz. příkaz MDB03 v kapitole 6.6 Nově implementované příkazy
- **04 – čtení analogového vstupu:** používá se pro přečtení hodnoty vstupu a případnou okamžitou reakci – viz. příkaz MDB04 v kapitole 6.6 Nově implementované příkazy
- **05 – zápis digitálního vstupu** (pouze hodnoty 0 nebo 1) – viz. definice časového plánu v kapitole 3.2 Sekce [schedule] nebo příkaz MDB05 v kapitole 6.6 Nově implementované příkazy
- **06 – zápis analogového vstupu** (pouze 1 dvou-Bytová hodnota představující znaménkové celé číslo) – viz. definice časového plánu v kapitole 3.2 Sekce [schedule] nebo příkaz MDB06 v kapitole 6.6 Nově implementované příkazy
- **15 – zápis více digitálních vstupů** (pouze hodnoty 0 nebo 1). je možné zapsat max. 16 bitů jedním příkazem – viz. příkaz MDB15 v kapitole 6.6 Nově implementované příkazy
- **16 – zápis více analogových vstupů** (více dvou-Bytových hodnot představujících znaménková celá čísla). Je možné zapsat max. 16 hodnot jedním příkazem – viz. příkaz MDB16 v kapitole 6.6 Nově implementované příkazy

Pro pasivní komunikaci (tedy reagování v programu na zprávy z připojených zařízení) jsou implementovány následující funkce:

- **01 – čtení digitálního výstupu** – viz. detaily dále
- **03 – čtení analogového výstupu** – viz. detaily dále
- **05 – zápis digitálního vstupu** – viz. detaily dále
- **06 – zápis analogového vstupu** – viz. detaily dále

5.4 Čísla registrů pro vyčítání hodnot z programu spotProcessor

5.4.1 Digitální výstupy

Používají se pro zasílání hodnot určitých registrů ve funkci 01. Jedná se celkový stav služby a stavy spínání/regulace řízených spotřebičů nebo stavu připojených indikátorů. Číslo prvního (bázového) registru pro konkrétní zařízení je závislé na čísle satelitu (SatNo, musí být větší než 0) a zařízení v něm (OrdNo, musí být mezi 1 a 8). Další hodnoty jsou umístěny v registrech hned za bázovým registrem.

Číslo prvního (bázového) registru se vypočítá podle vzorce:

$$\text{Adresa} = \text{SatNo} * 32 + (\text{OrdNo}-1) * 4$$

Seznam možných registrů pro poskytování digitálních výstupů přes Modbus, funkce 01:

Číslo registru	Počet bitů	Popis bitů zprava
0	4	1 – Stav služby (0 = kritická chyba, 1 = běží) 2 – Stav licence (0 = není platná, 1 = platná) 3 – Stav plánovače (0 = plány se neprovádí, 1 = plány se provádí) 4 – Stav reakcí (0 = reakce se neprovádí, 1 = reakce se provádí)
4	28	rezerva
32 Speciální zařízení pro ukládání spotových cen (SatNo=1, OrdNo=1)	4	2 a 1 – Stav regulace (00 = POD limitem, 01 = NAD limitem, 10 = MEZI limity, 11 = NEZNÁMÝ) 3 – Je aktivní? (0 = není, 1 = je) 4 – Poslední akce chybná? (0 = poslední akce úspěšná, 1 = poslední akce chybná)
od 64 dále Příklad: Pro zařízení 2-3 (SatNo=2, OrdNo=3) je číslo registru $2*32 + (3-1)*4 = 72$	4 pro každé zařízení	2 a 1 – Stav sepnutí (00 = OFF, 01 = ON, 10 = REGULATED, 11 = UNKNOWN) 3 – Je aktivní? (0 = není, 1 = je) 4 – Poslední akce chybná? (0 = poslední akce úspěšná, 1 = poslední akce chybná)

Příklad komunikace:

Pokud připojené zařízení chce zjistit stav regulace podle spotových cen přes protokol Modbus RTU, zašle následující zprávu (SlaveID je nastaveno na 2, speciální zařízení pro načítání spotových cen má SatNo=1 a OrdNo=1). Zpráva je uvedena po Bytech v hexadecimálním kódu):

02 – SlaveID
01 – Kód funkce
00 – Horní bajt adresy prvního registru
20 – Dolní bajt adresy prvního registru ($1*32 + (1-1)*4 = 32$ dekadicky)
00 – Horní bajt počtu registrů
04 – Dolní bajt počtu registrů (obvykle 4, max. může být 16)
3C – Kontrolní součet CRC
30 – Kontrolní součet CRC

Zpět se může vrátit následující odpověď:

02 – SlaveID
01 – Kód funkce
02 – Počet následujících Byte
00 – Horní bajt významového Byte
05 – Dolní bajt významového Byte (binárně 0101: žádná chyba, aktivní, zapnuto)
3D – Kontrolní součet CRC
FF – Kontrolní součet CRC

5.4.2 Analogové výstupy – aktuální stav

Používají se pro zasílání hodnot určitých registrů ve funkci 03. Těmito hodnotami jsou měřené veličiny spotřebičů, indikátorů nebo spotové ceny. Všechny zasílané hodnoty jsou typu znaménkového celého čísla. Reálná čísla jsou tak z důvodu přesnosti násobena a při jejich použití na straně masteru je nutné je opět zpětně vydělit. Například napětí je násobeno 1000 a teplota 10.

Číslo prvního (bázového) registru pro konkrétní zařízení je závislé na čísle satelitu (SatNo) a zařízení v něm (OrdNo). Další hodnoty jsou umístěny v registrech hned za tímto prvním registrem.

Číslo bázového registru se vypočítá podle vzorce:

$$\text{Adresa} = \text{SatNo} * 32 + (\text{OrdNo}-1) * 4$$

Při zpracování odpovědi na požadavek 03 se vždy nejprve vypočítá bázový registr. Odpověď pak vrací tolik hodnot, kolik je v požadavku uvedeno jako požadovaný počet. Například pro zařízení 2-2 je bázový registr $2*32 + 1*4 = 68$. Je-li například v požadavku uvedena adresa registru 70 a délka 4, pak se vrátí hodnoty 4 registrů počínaje bázovým registrem 68 (tedy nikoliv registry 70-73, ale 68-71). Toto omezení limituje případné omyly a pomíchání hodnot více různých zařízení.

Pořadí registrů počínaje bázovým registrem a použité násobitele u konkrétních typů informací jsou uvedeny v následujících odstavcích.

Standardní spotřebič ENcontrol

Pořadí registru	Popis	Násobitel
1 = bázový reg.	Aktuální napětí [V]	10
2	Aktuální proud [A]	10
3	Aktuální příkon [W]	10
4	Kumulovaná spotřeba [Wh]	1

Standardní indikátor ENcontrol

Pořadí registru	Popis	Násobitel
1 = bázový reg.	Aktuální teplota [°C]	10
2	Aktuální vlhkost [%]	10
3	Aktuální Intenzita osvětlení [Lux]	1
4	nepoužito	

Speciální zařízení pro ukládání spotových cen

Pořadí registru	Popis	Násobitel
1 = bázový reg.	Aktuální cena [EUR / MWh]	100
2	Aktuální objem [MWh]	1
3	Aktuální bilance [MWh]	1
4	Platná hodina (0-23)	1

Příklad výpočtu čísla registru:

Číslo registru hodnoty aktuální spotové ceny uložené ve speciálním zařízení pro vyhodnocování spotových cen 1-1 (SatNo = 1, OrdNo = 1) má adresu registru $1 * 32 + (1-1) * 4$, tedy 32. Použitý násobitel je 100, takže zjištěnou hodnotu ceny uvedenou jako znaménkové celé číslo je následně nutné vydělit 100.

Příklad komunikace:

Pokud připojené zařízení chce zjistit aktuální spotovou cenu a další údaje přes protokol Modbus RTU, zašle následující zprávu (SlaveID je nastaveno na 2, speciální zařízení pro načítání spotových cen má SatNo=1 a OrdNo=1). Zpráva je uvedena po Bytech v hexadecimálním kódu):

02 – SlaveID
03 – Kód funkce
00 – Horní bajt adresy prvního registru
20 – Dolní bajt adresy prvního registru ($32 * 1 + (1-1) * 4 = 32$ dekadicky)
00 – Horní bajt počtu registrů
04 – Dolní bajt počtu registrů (obvykle 4, max. může být 4)
45 – Kontrolní součet CRC
F0 – Kontrolní součet CRC

Zpět se může vrátit následující odpověď:

02 – SlaveID
03 – Kód funkce
08 – Počet následujících Byte
29 – Horní bajt významového Byte – Cena
05 – Dolní bajt významového Byte – Cena (2905 hex = 10501 dec \approx 105,01 EUR/MWh)
43 – Horní bajt významového Byte – Objem
5E – Dolní bajt významového Byte – Objem (435E hex = 17246 dec \approx 172,01 MWh)
D3 – Horní bajt významového Byte – Bilance
DC – Dolní bajt významového Byte – Bilance (D3DC hex = -11556 dec \approx -1152,56 MWh)
00 – Horní bajt významového Byte – Hodina
0B – Dolní bajt významového Byte – Hodina (000B hex = 11 dec \approx 11 hodin)
12 – Kontrolní součet CRC
66 – Kontrolní součet CRC

5.4.3 Analogové výstupy – historie a předpovědi

Z programu spotProcessor je kromě aktuálních hodnot možné také vyčítat předchozí hodnoty a budoucí předpokládané hodnoty stahovaných spotových cen. Pořadí registrů počínaje bázovým, formát údajů i datové typy jsou shodné s vyčítáním aktuálních spotových cen. Liší se ovšem:

- v číslech SatNo. Zatímco speciální zařízení pro stahování a ukládání spotových cen je 1-1 (SatNo=1, OrdNo=1), historické i budoucí hodnoty jsou uloženy ve virtuálních zařízeních s čísly SatNo 40 až 51
- Číslo OrdNo jsou vždy 1 a 24 a odpovídají jednotlivým denním hodinám
- Způsob výpočtu bázového registru je jiný.

Význam příslušných čísel SatNo udává následující tabulka:

Číslo SatNo	Význam	Poznámka
40 – 48	Uložení hodnot pro 2 (SatNo = 48) až 10 dnů (SatNo = 40) zpátky	Není-li tak staré měření k dispozici, vrátí se chybové hlášení
49	Uložení hodnot pro předchozí den	Není-li tak staré měření k dispozici, vrátí se chybové hlášení
50	Uložení hodnot pro tento den	
51	Uložení hodnot pro následující den	OTE poskytuje údaje na následující den od 13 hodin předchozího dne

Číslo bazového registru se vypočítá podle vzorce:

$$\text{Adresa} = \text{SatNo} * 96 + (\text{OrdNo}-1) * 4$$

Například pro zjištění ceny předchozí den v 15 hodin je adresa registru $49*96 + (15-1)*4 = 4760$. Pořadí registrů je shodné se zprávou pro aktuální údaje o spotové ceně.

Příklad komunikace:

Pokud připojené zařízení chce zjistit spotovou cenu a další údaje předchozí den v 15 hodin přes protokol Modbus RTU, zašle následující zprávu (SlaveID je nastaveno na 2, SatNo pro předchozí den je 49). Zpráva je uvedena po Bytech v hexadecimálním kódu:

- 02 – SlaveID
- 03 – Kód funkce
- 12 – Horní bajt adresy prvního registru
- 98 – Dolní bajt adresy prvního registru ($96*49 + (15-1)*4 = 4760$ dekadicky)
- 00 – Horní bajt počtu registrů
- 04 – Dolní bajt počtu registrů (obvykle 4, max. může být 4)
- C0 – Kontrolní součet CRC
- AD – Kontrolní součet CRC

Zpět se může vrátit následující odpověď:

- 02 – SlaveID
- 03 – Kód funkce
- 08 – Počet následujících Byte
- 22 – Horní bajt významového Byte – Cena
- 75 – Dolní bajt významového Byte – Cena (2275 hex = 8821 dec \approx 88,21 EUR/MWh)
- EA – Horní bajt významového Byte – Objem
- FE – Dolní bajt významového Byte – Objem (EAFE hex = 256510 dec \approx 2565,1 MWh)
- DC – Horní bajt významového Byte – Bilance
- 2A – Dolní bajt významového Byte – Bilance (DC2A hex = -9430 dec \approx -94,30 MWh)
- 00 – Horní bajt významového Byte – Hodina
- 0F – Dolní bajt významového Byte – Hodina (000F hex = 15 dec \approx 15 hodin)
- 58 – Kontrolní součet CRC
- D7 – Kontrolní součet CRC

5.5 Reagování na zápis digitálních a analogových vstupů

5.5.1 Digitální vstup

Používá se pro reagování programu na zaslání digitálních vstupů ZAPNUTO/VYPNUTO pomocí Modbus funkce 05.

Nejprve je nutné nastavit a povolit příslušnou reakci v hlavním konfiguračním souboru – viz. kapitola 3.3 Sekce [reaction].

Příklad dvou různých konfiguračních sekcí pro reagování na digitální vstup:

```
[reaction]
#Popis reakce - komentar
ReactionID=101
Active=true
StarterEvent=05
StarterSatellite=1
StarterOrderNum=120
ActionType=Modbus
ActionEvent=06
ActionMaxTimeOn=
ActionMaxTimeOff=
ActionSatellite=3
ActionOrderNum=8
ActionIndicatorType=RTU
ActionHigh=1000
ActionLow=

[reaction]
#Popis reakce - komentar
ReactionID=102
Active=true
StarterEvent=05
StarterSatellite=1
StarterOrderNum=140
ActionType=Macro
ActionEvent=/media/extended/spotProcessor/SPOTPRICE_EXEC.mac
ActionMaxTimeOn=
ActionMaxTimeOff=
ActionSatellite=1
ActionOrderNum=1
ActionIndicatorType=
ActionHigh=
ActionLow=
```

První reakce bude každou příchozí zprávu Modbus testovat, jestli:

- Číslo funkce je 05 (parametr *StarterEvent*)

-
- Číslo registru, které se bude testovat, je 120 (parametr *StarterOrderNum*)
 - Hodnota, která akci spustí, musí být logická 1 (zadáva se do parametru *StarterSatellite*)

Jsou-li výše uvedené podmínky splněny, spustí se reakce. Konkrétně se zašle zpráva přes Modbus RTU (parametry *ActionType* a *ActionIndicatorType*) s funkcí 06 (parametr *ActionEvent*), kdy se do registru č. 8 (parametr *ActionOrderNum*) v zařízení se slaveID 3 (parametr *ActionSatellite*) pošle hodnota 1000 (parametr *ActionHigh*).

Druhá reakce bude každou příchozí zprávu Modbus testovat, jestli:

- Číslo funkce je 05 (parametr *StarterEvent*)
- Číslo registru, které se bude testovat, je 140 (parametr *StarterOrderNum*)
- Hodnota, která akci spustí, musí být logická 1 (zadáva se do parametru *StarterSatellite*)

Jsou-li výše uvedené podmínky splněny, spustí se reakce. Konkrétně se spustí programové makro SPOTPRICE_EXEC.mac (parametry *ActionType* a *ActionEvent*).

Najde-li se pro příchozí zprávu aktivní reakce, je zasláno zpět potvrzení (echo, opakování požadavku). Nenajde-li se pro příchozí zprávu žádná aktivní reakce, je zasláno zpět chybové hlášení „Datová adresa uvedená v požadavku není dostupná.“

Příklad komunikace:

Pokud připojené zařízení chce do spotProcessor poslat logickou 1 přes protokol Modbus RTU tak, aby vyhovovala definici první reakce výše, zašle následující zprávu (SlaveID je nastaveno na 2). Zpráva je uvedena po Bytech v hexadecimálním kódu):

- 02 – SlaveID
- 05 – Kód funkce
- 00 – Horní bajt adresy prvního registru
- 78 – Dolní bajt adresy prvního registru (120 dekadicky)
- FF – Horní bajt hodnoty
- 00 – Dolní bajt hodnoty (Logické 1 v protokolu Modbus odpovídá hodnota FF00 hex)
- 0C – Kontrolní součet CRC
- 10 – Kontrolní součet CRC

Pokud se najde aktivní příslušná reakce, zpět se vrátí opakování požadavku. Pokud se nenajde, vrátí se chybové hlášení:

- 02 – SlaveID
- 85 – Kód chybové funkce
- 02 – Kód chyby
- 33 – Kontrolní součet CRC
- 51 – Kontrolní součet CRC

5.5.2 Analogový vstup

Používá se pro reagování programu na zaslání analogových vstupů pomocí Modbus funkce 06, kdy se testuje shoda se zadanou hodnotou. Hodnota testovaného čísla musí být pro uskutečnění reakce přesně shodná s hodnotou v konfiguraci (znaménkové celé číslo).

Nejprve je nutné nastavit a povolit příslušnou reakci v hlavním konfiguračním souboru – viz. kapitola 3.3 Sekce [reaction].

Příklad dvou různých konfiguračních sekcí pro reagování na analogový vstup:

```
[reaction]
#Popis reakce - komentar
ReactionID=103
Active=true
StarterEvent=06
StarterSatellite=500
StarterOrderNum=200
ActionType=Modbus
ActionEvent=06
ActionMaxTimeOn=
ActionMaxTimeOff=
ActionSatellite=3
ActionOrderNum=8
ActionIndicatorType=TCP
ActionHigh=1000
ActionLow=

[reaction]
#Popis reakce - komentar
ReactionID=104
Active=true
StarterEvent=06
StarterSatellite=1000
StarterOrderNum=300
ActionType=Macro
ActionEvent=/media/extended/spotProcessor/SPOTPRICE_EXEC.mac
ActionMaxTimeOn=
ActionMaxTimeOff=
ActionSatellite=1
ActionOrderNum=1
ActionIndicatorType=
ActionHigh=
ActionLow=
```

První reakce bude každou příchozí zprávu Modbus testovat, jestli:

- Číslo funkce je 06 (parametr *StarterEvent*)
- Číslo registru, které se bude testovat, je 200 (parametr *StarterOrderNum*)
- Hodnota, která akci spustí, musí být číslo 500 (zadáva se do parametru *StarterSatellite*)

Jsou-li výše uvedené podmínky splněny, spustí se reakce. Konkrétně se zašle zpráva přes Modbus TCP (parametry *ActionType* a *ActionIndicatorType*) s funkcí 06 (parametr *ActionEvent*), kdy se do registru č. 8 (parametr *ActionOrderNum*) v zařízení se slaveID 3 (parametr *ActionSatellite*) pošle hodnota 1000 (parametr *ActionHigh*).

Druhá reakce bude každou příchozí zprávu Modbus testovat, jestli:

- Číslo funkce je 06 (parametr *StarterEvent*)
- Číslo registru, které se bude testovat je 300 (parametr *StarterOrderNum*)
- Hodnota, která akci spustí, musí být číslo 1000 (zadáva se do parametru *StarterSatellite*)

Jsou-li výše uvedené podmínky splněny, spustí se reakce. Konkrétně se spustí programové makro SPOTPRICE_EXEC.mac (parametry *ActionType* a *ActionEvent*).

Najde-li se pro příchozí zprávu aktivní reakce, je zasláno zpět potvrzení (echo, opakování požadavku). Nenažde-li se pro příchozí zprávu žádná aktivní reakce, je zasláno zpět chybové hlášení „Datová adresa uvedená v požadavku není dostupná.“

Příklad komunikace:

Pokud připojené zařízení chce do spotProcessor poslat hodnotu 500 přes protokol Modbus RTU tak, aby vyhovovala definici první reakce výše, zašle následující zprávu (SlaveID je nastaveno na 2). Zpráva je uvedena po Bytech v hexadecimálním kódu):

02 – SlaveID
06 – Kód funkce
00 – Horní bajt adresy prvního registru
C8 – Dolní bajt adresy prvního registru (200 dekadicky)
01 – Horní bajt hodnoty
F4 – Dolní bajt hodnoty (500 dekadicky)
08 – Kontrolní součet CRC
10 – Kontrolní součet CRC

Pokud se najde aktivní příslušná reakce, zpět se vrátí opakování požadavku. Pokud se nenažde, vrátí se chybové hlášení:

02 – SlaveID
86 – Kód chybové funkce
02 – Kód chyby
33 – Kontrolní součet CRC
A1 – Kontrolní součet CRC

6 Použití makrojazyka

Program spotProcessor zpracovává makrojazyk, který je podobný makrojazyku používanému v plné verzi aplikace ENcontrol. Pro popis tohoto jazyka doporučujeme seznámit se s příručkou k plné verzi aplikace zde:

http://www.encontrol.eu/download/UG-Application-v2.0-FULL_cs.pdf

Mezi makrojazykem plné verze a programem spotProcessor existují rozdíly v implementaci, které popisují následující odstavce.

6.1 Identifikace zařízení

Základním rozdílem mezi makrojazykem plné verze aplikace ENcontrol a programem spotProcessor je, že v plné verzi se k identifikaci spotřebičů a indikátorů používají jejich názvy, kdežto v programu spotProcessor se používá identifikace *Satelit–číslo zařízení* (tedy SatNum–OrdNum).

Příklad příkazu v plné verzi aplikace:

```
DEVON PRACKA      (PRACKA je definovaný spotřebič se SatNum=8 a OrdNum=1)
```

Tentýž příklad v programu spotProcessor:

```
DEVON 8-1        (satelit s č. 8 a číslo zařízení č. 1 v tomto satelitu)
```

6.2 Průměrování měření

Program spotProcessor nemá relační databázi a program si pamatuje pouze poslední hodnoty měření (v závislosti na délce uloženého logovacího souboru). Proto není možné průměrovat měřené údaje spotřebičů nebo čidel v určitém časovém intervalu a specifikace časových intervalů pro průměrování se tak nebere v úvahu. **Vždy se použije pouze hodnota posledního měření.** Týká se to příkazů:

- IFDEV M
- IFIND M

Příkaz IFINDM lze s výhodou použít například pro reagování na výši spotové ceny – jiné, než jsou zadané hodnoty dolního a horního limitu v konfiguračním souboru. Příklad nuceného zapnutí výstupu zařízení Modbus SSR 6 (č. registru = 5), je-li cena nižší než 43,5 EUR:

```
IFINDM 1-1 PRIC 1H < 43.5 MDB06 1-5 1000
```

6.3 Telefonní čísla a emailové adresy

Program spotProcessor nemá databázi telefonních čísel a emailových adres. Proto se musejí uvádět konkrétní čísla a adresy přímo v definici příkazu.

Příklady příkazů v plné verzi aplikace:

```
SENDMAIL MARIE "Testovací zprava"  
SENDSMS MARIE "Testovací zprava"
```

Tytěž příklady v spotProcessor :

```
SENDMAIL marie.novakova@gmail.com "Testovací zprava"  
SENDSMS +420123456789 "Testovací zprava"
```

6.4 Aktivace /deaktivace plánů a reakcí

Příkazy makrojazyka pro aktivaci a deaktivaci časových plánů a reakcí fungují shodně jako v plné verzi, ale při ukončení programu se stav aktivace zpět nezapisuje do konfiguračního souboru. Proto při novém spuštění programu spotProcessor je stav aktivace nastaven opět podle údajů v konfiguračním souboru. Týká se příkazů:

- SCHACT
- SCHDEACT
- RCTACT
- RCTDEACT

6.5 Neimplementované příkazy

V programu spotProcessor není implementována kompletní množina příkazů makrojazyka plné verze. Následující seznam uvádí příkazy, které nejsou implementovány:

- DEVCALCL
- DEVCALCN
- INDCALCL
- INDCALCN
- OPTIMIZE
- OPTSET
- BACKUP
- IFLERROR
- IFNLDBMDF

6.6 Nově implementované příkazy

V programu spotProcessor jsou implementovány následující příkazy:

- EXECOMMAND
- IFDEVLE
- IFINDLE
- MDB01, MDB01S
- MDB03, MDB03S

- MDB04, MDB04S
- MDB05, MDB05S
- MDB06, MDB06S
- MDB15, MDB15S
- MDB16, MDB16S
- MDB32, MDB32S
- SCHOPT_SPOT
- SLEEP

6.6.1 Nově implementované příkazy (mimo Modbus)

Příkaz **EXECOMMAND** spustí jakýkoli externí program, který je uložen ve shodném adresáři, jako je soubor servisního makra (/media/extended/spotProcessor/).

Příklad zapnutí relé č. 3 v externím reléovém modulu:

```
EXECOMMAND REL03-ON
```

Příkazy **IFDEVLE** a **IFINDLE** znamenají vyhledání poslední chyby („If Device Last Error“ a „If Indicator Last Error“). Syntaxe je podobná příkazu IFDEVLA.

Příklady:

```
IFDEVLE 3-5 1H DEVON 1-2  
IFDEVLE 3-5 1H EXIT
```

Příkaz **SCHOPT_SPOT** provádí optimalizaci časových plánů (jejich přeplánování) podle budoucích spotových cen.

Příklad použití příkazu:

```
SCHOPT_SPOT 211,212 MIN 5 12
```

Syntaxe:

Příkaz ID pl. START ID pl. STOP min./max.? Délka intervalu Hledání v X násl. hodinách

Význam výše uvedeného příkladu: „Přeplánuj existující plán s ID=211 a existující plán s ID=212 podle budoucích spotových cen. Hledej minimální ceny v souvislé řadě 5ti hodin v následujících 12ti hodinách. Plán s ID=211 se naplánuje na začátek identifikovaného intervalu a plán s ID=212 se naplánuje na konec identifikovaného intervalu.“

Důležité poznámky:

- Hledá se vždy **souvislá řada** hodin, která má součet hodinových cen vždy minimální nebo maximální (podle třetího parametru).

- Optimální řada se hledá **od následující celé hodiny**. Je-li optimalizace spouštěna v 5:59, hledá se od 6:00. Je-li spouštěna v 6:00, hledá se od 7:00.
- **ID plánů**, na které se odvolává příkaz SCHOPT_SPOT, **musí existovat**. Mohou mít jakékoliv parametry včetně opakování, výjimek, apod. Mohou být i neaktivní.
- Mezi ID plánů je vždy čárka.
- Třetím parametrem je buď text „MIN“ nebo „MAX“.
- Minimální hodnota čtvrtého parametru (interval) je 1.
- Minimální hodnota pátého parametru (v kolika následujících hodinách hledat) je 2, maximální 24.

Příklad praktického nastavení optimalizace: Chceme zapínat bojler v noci (mezi 18 – 6 hodin) po dobu 5ti hodin a přes den (mezi 6 – 18 hodin) po dobu 2 hodin. Vždy v intervalech s minimálními spotovými cenami.

a) Zadáme do konfiguračního souboru následující 4 plány:

```
[schedule]
#Optimalizace intervalu od 06:00
ScheduleID=201
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT1.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 05:50:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 05:50:00

[schedule]
#Optimalizace intervalu od 18:00
ScheduleID=202
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT2.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
```

```
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 17:50:00
DoDateTo=01.01.2033 17:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 17:50:00

[schedule]
#Spusteni pri nejnizsi cene - START
ScheduleID=211
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_MIN_START.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=false
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 12:00:00
DoDateTo=01.01.2033 12:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:00

[schedule]
#Spusteni pri nejnizsi cene - STOP
ScheduleID=212
Active=false
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_MIN_STOP.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=SPOTPRICE
High=
Low=
Repeatable=false
```

```
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 12:00:00
DoDateTo=01.01.2033 12:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:00
```

b) Aktualizujeme soubor makra SPOTPRICE_OPT1.mac s následujícím obsahem:

```
##Optimalizace pro 18:00-06:00
#
##Interval 5 hodin v nasledujicich 12 hodinach
SCHOPT_SPOT 211,212 MIN 5 12
EXIT
```

c) Aktualizujeme soubor makra SPOTPRICE_OPT2.mac s následujícím obsahem:

```
##Optimalizace pro 06:00-18:00
#
##Interval 2 hodin v nasledujicich 12 hodinach
SCHOPT_SPOT 211,212 MIN 2 12
EXIT
```

d) Aktualizujeme soubor programového makra SPOTPRICE_MIN_START.mac, do kterého zadáme všechny příkazy, které se mají spustit na začátku optimálního intervalu. Například pro bojler, který má SatNo=4 a OrdNo=1:

```
##Zapnuti spotrebicu
#
##Bojler
DEVON 4-1
EXIT
```

e) Aktualizujeme soubor programového makra SPOTPRICE_MIN_STOP.mac, do kterého zadáme všechny příkazy, které se mají spustit na konci optimálního intervalu. Například pro bojler, který má SatNo=4 a OrdNo=1:

```
##Vypnuti spotrebicu
#
##Bojler
DEVOFF 4-1
EXIT
```

Příkaz **SLEEP** počká s prováděním příkazů zadaný počet vteřin. Tento příkaz je vhodné použít pouze v případě nutnosti, například pokud jednotka komunikuje s pomalým Modbus zařízením, které potřebuje mezi jednotlivými příkazy prodlevy.

Příklad čekání 2 vteřiny:

```
SLEEP 2
```

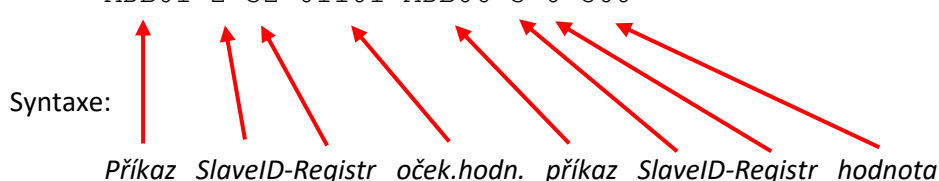
6.6.2 Nově implementované příkazy pro Modbus

Příkaz **MDB01** (alt. **MDB01S**) je určen pro zjištění diskrétních hodnot z určitých registrů (max. 16) přes protokol Modbus, jejich porovnání s očekávanými hodnotami a případné provedení navazujícího příkazu. Očekávaná hodnota je zadávána **binárně**. Příkaz MDB01 používá protokol Modbus TCP, příkaz MDB01S pak protokol Modbus RTU.

Příklad příkazu:

MDB01 2-32 01101 MDB06 3-0 500

Syntaxe:



Příkaz SlaveID-Registr oček.hodn. příkaz SlaveID-Registr hodnota

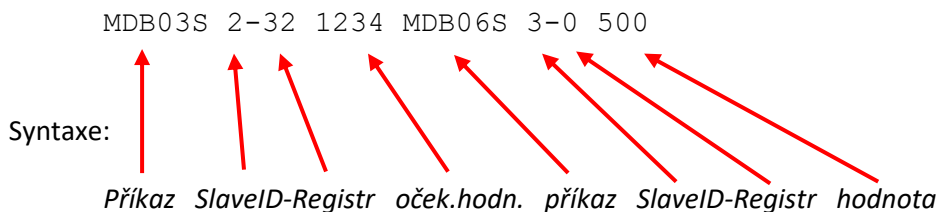
Význam: Přečti 16 bitů počínaje adresou 32 ze zařízení se SlaveID=2 pomocí Modbus funkce 01. Přečtených 16 bitů porovnej se zadanou **binární** hodnotou **zprava**. Pokud se všechny uvedené bity rovnají, spust navazující příkaz (MDB06 3-0 500).

Příklad porovnání:

- a) Zadaná hodnota: 01101
Přečtená hodnota: 00110011000**01101**
Výsledek: **PRAVDA**
- b) Zadaná hodnota: 00001101
Přečtená hodnota: 01010101**00001001**
Výsledek: **NEPRAVDA**
- a) Zadaná hodnota: 0
Přečtená hodnota: 001100110000110**1**
Výsledek: **NEPRAVDA**

Příkazy **MDB03** a **MDB04** (alt. **MDB03S** a **MDB04S**) jsou určeny pro přečtení analogové hodnoty z určitého registru (2 Byte) přes protokol Modbus, její porovnání s očekávanou hodnotou a případné provedení navazujícího příkazu. Očekávaná hodnota je zadávána jako **celé číslo** s případným znaménkem (rozsah -32.768 až 32.767). Přečtená hodnota je považována také za celé znaménkové číslo (16 bitů). Příkazy MDB03 a MDB04 používají protokol Modbus TCP, příkazy MDB03S a MDB04S pak protokol Modbus RTU.

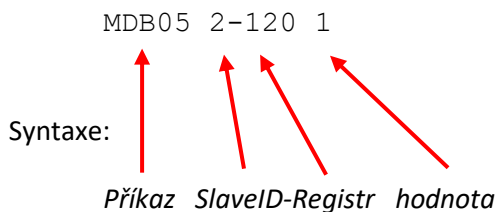
Příklad příkazu:



Význam: Přečti 2 Byte počínaje adresou 32 ze zařízení se SlaveID=2 pomocí Modbus funkce 03. Přečtené 2 Byte převed' na znaménkové celé číslo a to porovnej se zadanou hodnotou 1.234. Pokud se čísla rovnají, spust' navazující příkaz (MDB06 3-0 500).

Příkaz **MDB05** (alt. **MDB05S**) je určen pro zápis jedné diskrétní hodnoty do určitého registru (1 bit) přes protokol Modbus. Příkaz MDB05 používá protokol Modbus TCP, příkaz MDB05S pak protokol Modbus RTU. Hodnota může být pouze 0 nebo 1.

Příklad příkazu:

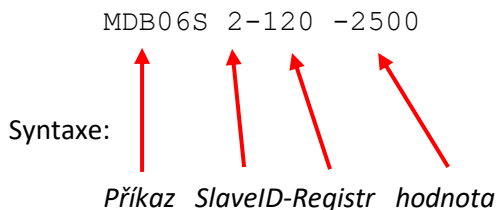


Význam: Zapiš logickou 1 (=ZAPNUTO) do registru s adresou 120 v zařízení se SlaveID=2 pomocí Modbus funkce 05.

Příkaz **MDB06** (alt. **MDB06S**) je určen pro zápis jedné analogové hodnoty do určitého registru (2 Byte, znaménkové celočíselné hodnoty -32.768 až 32.767) přes Modbus. Příkaz MDB06 používá protokol TCP, příkaz MDB06S pak RTU.

Do připojeného zařízení (slave) je možné zasílat i neznaménková čísla v rozsahu 0 až 65.535. Je-li zadáno číslo větší než 32.767, je program jej pošle jako neznaménkové 16-ti bitové číslo.

Příklad příkazu:



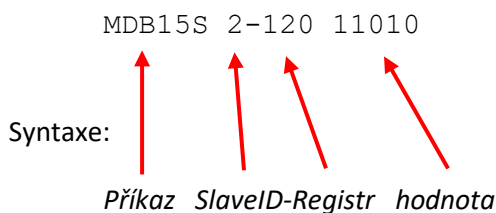
Význam: Zapiš číslo -2500 do registru s adresou 120 v zařízení se SlaveID=2 pomocí Modbus funkce 06.

Příkaz **MDB15** (alt. **MDB15S**) je určen pro zápis více diskrétních hodnot do určitých registrů (po 1 bitu) přes protokol Modbus. Příkaz MDB15 používá protokol Modbus TCP, příkaz MDB15S pak protokol Modbus RTU. Bity jsou zapisovány od prvního registru zprava doleva. Nezádané bity jsou doplněny zleva nulami.

Příklad příkazu:

MDB15S 2-120 11010

Syntaxe:



Příkaz SlaveID-Registr hodnota

Význam: Zapiš logickou 0 do registru 120, 1 do 121, 0 do 122 a 1 do 123 a 124 v zařízení se SlaveID=2 pomocí Modbus funkce 15.

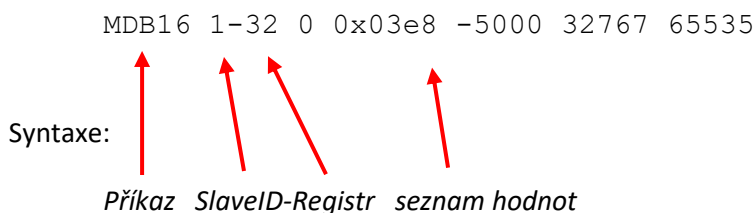
Příkaz **MDB16** (alt. **MDB16S**) je určen pro zápis více analogových hodnot (2 Byte, -32.768 až 32.767) do určitých registrů přes protokol Modbus. Příkaz MDB16 používá protokol Modbus TCP, příkaz MDB16S pak protokol Modbus RTU. Zapisované hodnoty jsou dekadické nebo hexadecimální a jsou odděleny mezerami.

Do připojeného zařízení (slave) je možné zasílat i neznaménková čísla v rozsahu 0 až 65.535. Je-li zadáno číslo větší než 32.767, je program jej pošle jako neznaménkové 16-ti bitové číslo.

Příklad příkazu:

MDB16 1-32 0 0x03e8 -5000 32767 65535

Syntaxe:



Příkaz SlaveID-Registr seznam hodnot

Význam: Zapiš hodnotu 0 do registru 32, 1.000 do registru 33 (03E8 hexadecimálně), -5.000 do registru 34, 32767 (nejvyšší možné znaménkové číslo) do registru 35 a 65535 (nejvyšší možné neznaménkové číslo) do registru 36 v zařízení se SlaveID=2 pomocí Modbus funkce 16.

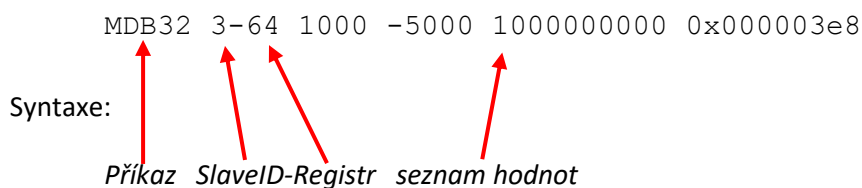
Příkaz **MDB32** (alt. **MDB32S**) je podobný příkazu MDB16, ale zapisuje více 32-bitových analogových hodnot (4 Byte, pouze znaménkové celočíselné hodnoty -2.147.483.648 až 2.147.483.647) do určitých registrů (po 2 Bytech do dvou sousedních registrů) pomocí funkce 16. Příkaz MDB32 používá Modbus TCP, příkaz MDB32S pak Modbus RTU. Zapisované hodnoty jsou dekadické nebo hexadecimální a jsou odděleny mezerami.

Příklad příkazu:

MDB32 3-64 1000 -5000 1000000000 0x000003e8

Syntaxe:

Příkaz SlaveID-Registr seznam hodnot



Význam: Zapiš hodnotu 1000 do registrů 64 a 65, -5.000 do registrů 66 a 67, 1.000.000.000 do registrů 68 a 69 a 1000 do registrů 70 a 71 v zařízení se SlaveID=3 pomocí Modbus funkce 16.

7 Specifika řízení regulátoru Wattrouter

Příkazy MDB06 a MDB16 jsou s výhodou využívány pro řízení regulátoru Wattrouter od společnosti SolarControls.

Tabulka řízení výstupů (Modbus registrů) zařízení Wattrouter:

Příkaz	SatNum	OrdNum	Hodnota	Význam
MDB06 / MDB16	1	0	1000	Nucené ZAPNUTÍ SSR 1
MDB06 / MDB16	1	1	1000	Nucené ZAPNUTÍ SSR 2
MDB06 / MDB16	1	2	1000	Nucené ZAPNUTÍ SSR 3
MDB06 / MDB16	1	3	1000	Nucené ZAPNUTÍ SSR 4
MDB06 / MDB16	1	4	1000	Nucené ZAPNUTÍ SSR 5
MDB06 / MDB16	1	5	1000	Nucené ZAPNUTÍ SSR 6
MDB06 / MDB16	1	6	1000	Nucené ZAPNUTÍ relé 1
MDB06 / MDB16	1	7	1000	Nucené ZAPNUTÍ relé 2
MDB06 / MDB16	1	8	1000	Nucené VYPNUTÍ SSR 1
MDB06 / MDB16	1	9	1000	Nucené VYPNUTÍ SSR 2
MDB06 / MDB16	1	10	1000	Nucené VYPNUTÍ SSR 3
MDB06 / MDB16	1	11	1000	Nucené VYPNUTÍ SSR 4
MDB06 / MDB16	1	12	1000	Nucené VYPNUTÍ SSR 5
MDB06 / MDB16	1	13	1000	Nucené VYPNUTÍ SSR 6
MDB06 / MDB16	1	14	1000	Nucené VYPNUTÍ relé 1
MDB06 / MDB16	1	15	1000	Nucené VYPNUTÍ relé 2

Registry 0-7 slouží k nucenému zapínání SSR a relé; registry 8-15 pak slouží k nucenému vypínání SSR a relé. Hodnota 1000 pro Wattrouter odpovídá 100% dané hodnoty (tedy úplné zapnutí nebo úplné vypnutí). Hodnota 0 je ignorována, protože znamená „nulovou“ regulaci.

Hodnota SatNum odpovídá SlaveID v protokolu Modbus. V nastavení Wattrouteru se najde v jeho konfiguraci na záložce DALŠÍ NASTAVENÍ – MODBUS adresa.

Příklad nuceného zapnutí relé 2 (slave = 1, registr = 7):

```
MDB06 1-7 1000
```

Příklad nuceného vypnutí SSR 3 (slave = 1, registr = 10):

```
MDB06 1-10 1000
```

Příklad hromadného nuceného zapnutí SSR2, SSR4, SSR6 a relé 2 (slave = 1, 16 registrů od 0):

```
MDB16 1-0 0 1000 0 1000 0 1000 0 1000 0 0 0 0 0 0 0
```

8 Vysvětlení obsahu makra SPOTPRICE_EXEC.mac

Programu spotProcessor je od výrobce nastaven tak, že při vyhodnocování spotové ceny spouští makro SPOTPRICE_EXEC.mac (viz. konfigurační soubor, časové plány).

Zde je příklad obsahu souboru SPOTPRICE_EXEC.mac s příkazy makrojazyka s vysvětlením jejich významu:

Č.ř.	Obsah souboru	Vysvětlení
1	#Vyhodnoceni spotove ceny a reakce	Komentář, přeskakuje se
2	#	
3	#Podminky splneni definovanych limitu	
4	IFINDS 1-1 OFF GOTO 13	Je-li cena nad/pod limitem,
5	IFINDS 1-1 ON GOTO 33	skoč na řádku č...
6	#	
7	#Prikazy pro rizeni mezi limity	
8	IFINDM 1-1 PRIC 1H > 95 EXECOMMAND REL03-OFF	Při cene > 95 vyp. rele 3
9	IFINDM 1-1 PRIC 1H < 106.5 MDB06 1-13 1000	Při cene < 106,5 vyp. SSR 6
10	#EXECOMMAND SPOTPRICE_between.sh	
11	EXIT	Dále nepokračovat
12	#	
13	##<MIN> - Prikazy pro pri cene nizsi nez MIN	
14	MDB16 1-0 1000 0 1000 0 0 0 0 0 0 0 0 0 0 0 0	Zapnout SSR 1 a SSR 3
15	MDB06S 2-12 500	Zapsat 500 do reg. 12
16	MDB06 1-1 1000	Zapnout SSR 2
17	MDB06 1-4 1000	Zapnout SSR 5
18	#MDB06S 2-4 1000	Neprovádí se
19	#MDB06S 2-5 1000	
20	#MDB06S 2-6 1000	
21	#MDB06S 2-7 1000	
22	EXECOMMAND REL01-ON	Zapnout ext. relé 1
23	EXECOMMAND REL02-ON	Zapnout ext. relé 2
24	#EXECOMMAND REL03-ON	Neprovádí se
25	#EXECOMMAND REL04-ON	
26	#EXECOMMAND REL05-ON	
27	#EXECOMMAND REL06-ON	
28	#EXECOMMAND REL07-ON	
29	#EXECOMMAND REL08-ON	
30	#EXECOMMAND SPOTPRICE_bellow.sh	
31	EXIT	Dále nepokračovat
32	#	
33	##<MAX> - Prikazy pro pri cene vyssi nez MAX	
34	MDB16 1-0 0 0 0 0 0 0 0 0 1000 0 1000 0 0 0 0 0	Vypnout SSR 1 a SSR 3
35	MDB06S 2-12 0	Zapsat 0 do reg. 12
36	MDB06 1-10 1000	Vypnout SSR 3
37	MDB06 1-11 1000	Vypnout SSR 4
38	#MDB06S 2-12 1000	Neprovádí se
39	#MDB06S 2-13 1000	
40	#MDB06S 2-14 1000	
41	#MDB06S 2-15 1000	
42	EXECOMMAND REL01-OFF	Vypnout ext. relé 1
43	EXECOMMAND REL02-OFF	Vypnout ext. relé 2
44	#EXECOMMAND REL03-OFF	Neprovádí se
45	#EXECOMMAND REL04-OFF	
46	#EXECOMMAND REL05-OFF	
47	#EXECOMMAND REL06-OFF	
48	#EXECOMMAND REL07-OFF	

49	#EXECOMMAND REL08-OFF	
50	#EXECOMMAND SPOTPRICE_above.sh	
51	EXIT	Dále nepokračovat

Řádky uvozené znaky „#“ se přeskakují – jedná se o komentáře.

Řádky uvozené „##<MIN>“ nebo „##<MAX>“ jsou pomocné značkovací řádky, které program spotProcessor nepoužívá, ale podle kterých se orientuje webová aplikace spotProcessor Web App. Tyto řádky tedy nemažte.

9 Další související soubory

Následující tabulky obsahují seznam obvykle používaných další souborů pro účely plného fungování programu spotProcessor.

Adresář /opt/encontrol/spotProcessor/ (stačí přístup pouze pro čtení)

Soubor	Účel
email-footer.txt	Patička emailu doplňovaná do odesílaného souboru skriptem email-send.sh
email-footer2.txt	Patička emailu doplňovaná do odesílaného souboru skriptem email-status.sh
email-header.txt	Hlavička emailu doplňovaná do odesílaného souboru skriptem email-send.sh
email-header2.txt	Hlavička emailu doplňovaná do odesílaného souboru skriptem email-status.sh
email-send.sh	Skript spouštěný programem při realizaci makropříkazu SENDMAIL
email-status.sh	Skript obvykle spouštěný cronem
encProcess-log.sh	Skript pro zkracování logu obvykle volaný cronem
encProcess-watch.sh	Skript pro kontrolu běhu programu obvykle volaný cronem
reset-usb2.sh	Skript pro unbind/bind USB volaný skriptem encProcess-watch.sh

Adresář /media/extended/spotProcessor/ (musí být přístup pro čtení i zápis)

Soubor	Účel
spotProcessor .log	Hlavní logovací soubor programu spotProcessor
spotProcessor_SERVICE.log	Výstupní soubor programu při zaslání signálu –SIGUSR1
spotProcessor_SERVICE.mac	Programové makro spouštěné při zaslání signálu –SIGUSR2
spotProcessor_STARTUP.mac	Programové makro spouštěné vždy při startu programu

10 Další související programy

10.1 Zasílání emailů

Pro zasílání emailů se používá externí program *exim4*. Následující řádky popisují jeho nastavení pro administrátory systému:

Konfigurační program

Spustit z příkazové řádky:

```
dpkg-reconfigure exim4-config
```

```
-----
Option                               Choice
-----
Configuration type                   mail sent by smarthost; received via
                                      SMTP or fetchmail
System mail name                       encontrol.cz
IP-addresses to listen on             127.0.0.1 (refuse external connections)
Other destinations                     leave empty
Machines to relay mail for            leave empty
IP address or host name                smtp.gmail.com:587 /
                                      smtp.powernet.cz:587 /
                                      172.16.10.254:25
Hide local mail name in outgoing?     no
Keep number of DNS-queries min?       no
Delivery method for local mail        mbox format in /var/mail/
Split configuration into small f?     yes
Root and postmaster mail recipient    root
-----
```

Editovat soubor `/etc/exim4/passwd.client`:

```
-----
# password file used when the local exim is authenticating to a remote
# host as a client.
#
# see exim4_passwd_client(5) for more documentation
#
# Example:
### target.mail.server.example:login:password

gmail-smtp.l.google.com:karel.novak@gmail.com:heslo
*.google.com: karel.novak@gmail.com:heslo
smtp.gmail.com: karel.novak@gmail.com:heslo

smtp.powernet.cz:novakk:heslo
*.powernet.cz: novakk:heslo
-----
```

Spustit z příkazové řádky:

```
cd /var/tmp/  
mkdir spool  
cd spool/  
mkdir exim4  
chmod a+x /var/tmp/spool/exim4  
chmod a+w /var/tmp/spool/exim4
```

Editovat soubor `/etc/exim4/conf.d/main/02_exim4-config_options` a upravit SPOOLDIR definici:

```
SPOOLDIR=/var/tmp/spool/exim4
```

Spustit z příkazové řádky:

```
chown Debian-exim:root /etc/exim4/passwd.client  
chmod 640 /etc/exim4/passwd.client  
update-exim4.conf  
invoke-rc.d exim4 restart  
exim4 -qff
```

Editovat `/etc/crontab` a přidat řádku:

```
MAILTO=""
```

Otestovat odeslání emailu vytvořením souboru `/root/mail-body.txt` s následujícím obsahem:

```
to : info@encontrol.cz  
from : noreply  
subject : Test mail
```

```
This is the first mail sent by my server's sendmail !
```

Spustit z příkazové řádky vlastní odeslání emailu:

```
cat /root/mail-body.txt | sendmail -t
```

10.2 Zasílání SMS zpráv

Pro zasílání SMS zpráv je nutné použít jakýkoliv externí USB modem s AT příkazy a program *minicom*. Následující řádky popisují jeho typickou instalaci a nastavení:

Konfigurační program

Spustit z příkazové řádky:

```
minicom -s
```

V nastavení programu *minicom* je pro USB modem typické nastavení:

```
PORT_NAME = /dev/ttyACM0    (nebo /dev/ttyUSB0)
```

```
BAUD_RATE = 15200
```