ENcontrol s.r.o. , IČO: 290 67 723
Email: info@encontrol.cz

Program spotProcessor v. 2.10

Konfigurační soubory a související nastavení

26.04.2024, ENcontrol s.r.o.

1 Obsah

2 Popis programu spotProcessor

Program spotProcessor je klonem programu encProcessor společnosti ENcontrol. Je **určen pro malé jednotky s operačním systémem Linux a ovládá připojená zařízení (regulátory, střídače, reléové moduly, spotřebiče a čidla)**. Zařízení mohou být **připojena přes LAN (ethernet, WiFi) nebo sériovou linkou RS-232 / RS-485**. Program provádí řízení zařízení podle zadaných časových plánů a reaguje na různé události. Umožňuje **spínání spotřebičů, měření různých elektrických veličin, detekci signálů z čidel, zasílání e-mailů a SMS**.

Program **spotProcessor** je rozšířen o:

- **stahování aktuálních spotových cen** elektřiny a plynu 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/RTU
- komunikaci s dalšími zařízeními přes http protokol Shelly
- komunikace s externími reléovými moduly přes USB, ethernet nebo WiFi.

Program umí reagovat zasíláním signálů z připojených zařízení a 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 otevřených protokolů **Modbus nebo Shelly 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.

Typicky připojovaná (a ovládaná) zařízení jsou:

- regulátory Wattrouter a GreenBonO
- chytré střídače FVE
- reléové WiFi, ethernet nebo USB moduly
- WiFi zásuvky
- elektroměry s otevřeným rozhraním

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. Komentáře se mohou psát také za příkazy na konce řádek.

3.1 Sekce [general]

Typická podoba sekce [general]:

```
[general]
ACTIVE_IFACE_NAME=wlan0  #obvykle wlan0, eth0
UNIT_PORTNUM_TCP=50150
STATION_PORTNUM_TCP=50151
REG_ADDR_FILE=/media/extended/spotProcessor/regAddr.conf
REG_CURRENCY_FILE=/media/extended/spotProcessor/regCurrency.conf
REG_TARIFF_FILE=/media/extended/spotProcessor/regTariff.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
MODBUS_RTU_CRCTYPE=HL

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  #NONE, CRIT, ERRO, WARN, ALL
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
SCENARIO_CONFIGS_DIR=/media/extended/spotProcessor/configs/
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  #ISO kod meny
MIN_SPOT_PRICE=62.0
MAX_SPOT_PRICE=105.5
INCLUDE_TARIFF_PRICE=true
TABLE_FILE_FORMAT=HTML
TABLE_FILE_PREFIX=/media/extended/spotProcessor/denni-trh-tab_
TABLE_FILE_POSTFIX=.txt
MIN_SPOT_RATIOG=1.5
MAX_SPOT_RATIOG=3.5
TABLE_FILE_FORMATG=HTML
TABLE_FILE_PREFIXG=/media/extended/spotProcessor/download/denni-
trhp-tab_
TABLE_FILE_POSTFIXG=.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

REG_CURRENCY_FILE	Cesta k souboru s údaji o měnových kurzech	/media/extended/spotProcessor/regCurrency.conf
REG_TARIFF_FILE	Cesta k souboru s údaji o časových pásmech a cenách regulovaných distribučních poplatků	/media/extended/spotProcessor/regTariff.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
MODBUS_RTU_CRCTYPE	Pořadí Bzte v kontrolním součtu CRC protokolu Modbus RTU. „H“ je vyšší Byte, „L“ pak nižší Byte	HL (běžný standard) LH
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 kritické chyby) - ERRO (navíc nekritické chyby) - WARN (navíc upozornění) - ALL (logování všeho včetně debug zpráv) 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 *schedule*, ne *reaction*) 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
SCENARIO_CONFIGS_DIR	Adresář, do kterého jsou ukládány konfigurační soubory jako individuální scénáře	/media/extended/spotProcessor/configs/
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
INCLUDE_TARIFF_PRICE	Je-li *false*, aktuální ceny elektřiny jsou neměněny – jsou přesně takové, jaké poskytuje OTE. Je-li *true*, je k aktuálním cenám elektřiny připočítávána regulovaná cena za distribuci podle daného tarifu (viz. soubor regTariff.conf)	false

TABLE_FILE_FORMAT	Formát souboru s cenami elektřiny 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
MIN_SPOT_RATIOG	Dolní limit poměru aktuální ceny elektřiny ku ceně plynu (desetinná tečka a max. 3 desetinná místa)	0.95
MAX_SPOT_RATIOG	Horní limit poměru aktuální ceny elektřiny ku ceně plynu (desetinná tečka a max. 3 desetinná místa)	3.5
TABLE_FILE_FORMATG	Formát souboru s cenami plynu stahovaného ze stránek OTE	HTML
TABLE_FILE_PREFIXG	Cesta a první část jména souboru, do kterého se ukládají stahované hodnoty	/media/extended/spotProcessor/denni-trhp-tab_
TABLE_FILE_POSTFIXG	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 nastavuji 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=
DoDateTo=
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:00
FollowFromToTimes=false
ActLogLevel=ALL

[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
FollowFromToTimes=false
ActLogLevel=ALL

[schedule]
#Pravidelne zasilani Modbus prikazu
ScheduleID=210
Active=true
RelationType=Modbus
Action=16
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=7
IndicatorType=TCP
High=1000 0 -100 0xe8ff
Low=
Repeatable=true
RepeatTime=1min
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 06:00:00
DoDateTo=01.01.2033 09:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 12:00:30
FollowFromToTimes=true
ActLogLevel=ERRO
```

Význam a hodnoty parametrů sekce [schedule].

Parametr	Popis	Příklad hodnoty
ScheduleID	Libovolné přirozené číslo	10
Active	Je-li hodnota „true“, plán se vykonává, jinak ne.	True false
RelationType	Hodnota může být: - Device - Indicator (pak musí být zároveň vyplněn i parametr IndicatorType) - Modbus - Macro	Device Indicator Modbus Shelly 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, 06, 15 nebo 16) a musí být vyplněn parametr High, případně i Low 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, 15, 16 (pro Modbus) /media/extended/macro…
MaxTimeOn	Specifikace bezpečnostní konstanty maximální doby sepnutí spotřebiče. Jedná se o přirozené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h
MaxTimeOff	Specifikace bezpečnostní konstanty maximální doby vypnutí spotřebiče. Jedná se o přirozené 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 (znaménkové) nebo 0 až 65535 (neznaménkové) zapisované do registrů (funkce 06 a 16). Může být zadáno i hexadec. Pro funkci 05 platí, že zadaná 0 znamená zapisovaná hodnota bitu 0; cokoliv jiného znamená hodnotu 1. Pro funkci 15 například zadané číslo 26 (binárně 11010) znamená postupný zápis do registrů od nejméně významného bitu 0, do nejvýzn. Bitu 1 Pro funkci 16 se zadává řada čísel oddělených mezerami (maximálně 16 z sebou u 16ti bitových nebo 8 u 32-bitových)	21.5 (desetinná tečka) 1 (nebo 0) pro funkci 05 1000 -1000 0x03e8 1 26 1000 0 -100 0xe8ff
Low	Dolní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound). U zařízení Modbus a funkce 15 se zadává počet registrů (zapisovaných bitů, např. 5 pro zadanou hodnotu 26 (binárně 11010). U funkce 16 je možné zadat číslo 32. V tom případě se zadané hodnoty považují za 32 bitové (jinak 16ti)	20.5 5 32 16
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: - 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. Speciální hodnota „OPTH“ má vazbu na optimalizační příkazy (viz. příkazy makrojazyka SCHOPT_SPHE, SCHOPT_SDHE, SCHOPT_SCHE, SCHOPT_COMP). Znamená, že plán je vykonán pouze v hodinách, kdy předchozí optimalizace je vybrala k vykonání.	PO UT ST CT PA SO NE PO-PA SO-NE PO-NE OPTH
DoDateFrom	Od jakého času je plán platný. Formát datumu a času je závazný a pole je nepovinné. Je-li prázdné, je plán neomezený ve smyslu od kdy.	01.01.2017 00:00:00
DoDateTo	Do jakého času je plán platný. Formát datumu a času je závazný a pole je nepovinné. Je-li prázdné, je plán neomezený ve smyslu do kdy.	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
FollowFromToTimes	Upřesňuje pole DoDateFrom, DoDateTo, ExceptDateFrom a ExceptdateTo. Je-li true, pak se odděleně vyhodnocují datumy a čas v těchto polích. **Pro nastavení „false“ platí:** - Akce se vykoná, pouze pokud aktuální čas je vyšší než DoDateFrom a nižší než DoDateTo. - Jsou-li vyplněny datumy výjimek, pak aktuální čas nesmí být v tomto intervalu **Pro nastavení „true“ a datumové části platí:** - Pokud se datumy OD a DO rovnají nebo datum DO není vyplněn, pak se bere v úvahu pouze datum OD a nebere se v úvahu datum DO - Pokud jsou oba datumy vyplněny a nerovnají se, pak se bere v úvahu datum OD i DO - Pokud není vyplněno jedno z datumů výjimky nebo se datumy výjimky rovnají, pak se výjimky neberou v úvahu **Pro nastavení „true“ a části času platí:** - Pokud se časy OD a DO rovnají, pak se čas nebere v úvahu (pouze datum) - Pokud jsou oba časy vyplněny a nerovnají se, pak se berou v úvahu čas OD i DO jako denní doba; pro neomezený běh je tak vhodné zadávat čas DO 23:59:59 - Pokud se časy výjimek rovnají, pak se nebere v úvahu ani čas OD, ani DO (pouze datumy bez zohlednění času)	true false defaultně je false

ActLogLevel	Globální úroveň logování služby je dána parametrem LOG_LEVEL. Pokud má daný plán úroveň shodnou nebo nižší, pak se bude do logu zapisovat každý jeho běh. Pozor tak na často vykonávané plány a nízkou úroveň, kdy by se log velmi rychle plnil. Standardní úroveň logování časových plánů je ALL.	NONE CRIT ERRO WARN ALL

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 přirozené 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 diskrétních 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 diskrétních 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 Shelly 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 přirozené číslo s písmenem ‘s’ (sekundy), ‘m’ (minuty) nebo ‘h’ (hodiny)	5s 20m 3h
ActionMaxTimeOff	Specifikace bezpečnostní konstanty maximální doby vypnutí spotřebiče. Jedná se o přirozené 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. Platí zde všechny pravidla, co u položky *High* v časových plánech	21.5 1 1000
ActionLow	Dolní mez při nastavení limitů indikátoru – reálné číslo (povinné u příkazu Setbound). Platí zde všechny pravidla, co u položky *Low* v časových plánech	20

4 Struktura dalších konfiguračních souborů

4.1 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
10     wlan0 10.0.1.81         00:0d:b9:27:6e:2c 80           50161
11     wlan0 10.0.1.82         00:0d:b9:27:6e:ef 80           50161
```

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

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

4.2 Struktura souboru regCurrency.conf

Pro účely případných přepočtů z měny, ve které jsou poskytovány ceny na stránkách OTE (obvykle EUR), je možné definovat další lokální měny a jejich kurzy. K tomuto účelu slouží soubor *regCurrency.conf*. Jeho umístění je definováno v parametru REG_CURRENCY_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 encProcessor
#Registration file containing info about additional currency and the rate
Currency/EUR
#
##Table header
```

```
#<CurrencyCode><RateEUR>
##Table records
CZK   24.16
USD   1.13
```

a) *Currency*: ISO kód měny. Obvykle CZK pro Českou Korunu.
b) *RateEUR*: Kurz dané měny ve vztahu k hlavní měně ceny (tedy obvykle CZK/EUR)

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

Ve webové aplikaci je pro účely zobrazení nabízena pouze měna uvedená na první řádce tabulky v konfiguračním souboru. Ve výše uvedeném případě tedy CZK.

4.3 Struktura souboru regTariff.conf

Někteří zákazníci chtějí řídit svá zařízení nikoliv podle holé spotové ceny, ale podle součtu této ceny a patřičných regulovaných poplatků za distribuci. K tomuto účelu slouží soubor *regCurrency.conf*. Obsahuje definice časových pásem zapínání tarifů a jejich distribuční ceny za 1 MWh. Tento plán bývá pro každý region i každého zákazníka s různými povely HDO jiný. Umístění souboru je definováno v parametru REG_TARIFF_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 encProcessor
#Registration file containing info about tariff periods and prices
#
##Table header
#<DayCode><StartTime><Tariff><Currency><Price>
##Table records
#
PO-PA 00:00 L     CZK   179.98
PO-PA 05:30 H     CZK   311.07
PO-PA 06:30 L     CZK   179.98
PO-PA 08:30 H     CZK   311.07
PO-PA 09:30 L     CZK   179.98
PO-PA 12:30 H     CZK   311.07
PO-PA 13:30 L     CZK   179.98
PO-PA 15:35 H     CZK   311.07
PO-PA 16:30 L     CZK   179.98
PO-PA 21:25 H     CZK   311.07
PO-PA 21:30 L     CZK   179.98
#
SO-NE 00:00 H     CZK   311.07
SO-NE 01:30 L     CZK   179.98
SO-NE 09:45 H     CZK   311.07
SO-NE 10:45 L     CZK   179.98
SO-NE 15:00 H     CZK   311.07
SO-NE 16:00 L     CZK   179.98
SO-NE 21:55 H     CZK   311.07
SO-NE 22:55 L     CZK   179.98
```

a) *DayCode*: Kód dne v týdnu nebo bloku. Jedná se o shodné kódy, jaké je možné použít v parametru RepeatWeek při definici časového plánu. Tedy: PO, UT, ST, CT, PA, SO, NE, PO-PA, SO-NE, PO-NE
b) *StartTime*: Čas, kdy se daný tarif **zapíná**. První záznam každého časového bloku tabulky by tak měl začínat s časem 00:00
c) *Tariff*: Kód tarifu. Lze použít pouze hodnoty „H“ (vysoký tarif) a „L“ (nízký tarif)
d) *Currency*: Měna, ve které jsou zadávány hodnoty v posledním sloupci (obvykle CZK)
e) *Price*: Regulovaná cena za distribuci 1 MWh.

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 směrovány 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 směrovány 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.*
- MODBUS_RTU_CRCTYPE: Některá zařízení očekávají opačné pořadí Byte v kontrolním součtu CRC, než je standard. Obvykle je pořadí vyšší Byte, pak nižší Byte (tedy nastavení „HL“).

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:

Adresa = SatNo * 32 + (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:

Adresa = SatNo * 32 + (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 ≈ 105,01 EUR/MWh)
43 – Horní bajt významového Byte – Objem
5E – Dolní bajt významového Byte – Objem (435E hex = 17246 dec ≈ 172,01 MWh)
D3 – Horní bajt významového Byte – Bilance
DC – Dolní bajt významového Byte – Bilance (D3DC hex = -11556 dec ≈ -1152,56 MWh)
00 – Horní bajt významového Byte – Hodina
0B – Dolní bajt významového Byte – Hodina (000B hex = 11 dec ≈ 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:

a) 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
b) Číslo OrdNo jsou vždy 1 a 24 a odpovídají jednotlivým denním hodinám
c) 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 bázového registru se vypočítá podle vzorce:

Adresa = SatNo * 96 + (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 ≈ 88,21 EUR/MWh)
EA – Horní bajt významového Byte – Objem
FE – Dolní bajt významového Byte – Objem (EAFE hex = 256510 dec ≈ 2565,1 MWh)
DC – Horní bajt významového Byte – Bilance
2A – Dolní bajt významového Byte – Bilance (DC2A hex = -9430 dec ≈ -94,30 MWh)
00 – Horní bajt významového Byte – Hodina
0F – Dolní bajt významového Byte – Hodina (000F hex = 15 dec ≈ 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 (zadává 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 (zadává 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 (zadává 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 (zadává 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 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 nenajde, 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 programu 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:

- IFDEVM
- IFINDM

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
```

Seznam klíčových slov pro zjišťování hodnot měření ze spotřebičů a indikátorů je uveden v tabulce u popisu příkazu MDB06 v kapitole 6.6.5.

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 "Testovaci zprava"
SENDSMS MARIE "Testovaci zprava"
```

Tytéž příklady v spotProcessor :

```
SENDMAIL marie.novakova@gmail.com "Testovaci zprava"
SENDSMS +420123456789 "Testovaci 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
- SHEON, SHEOFF, SHEPLS
- SCHOPT_SPHE, SCHOPT_SPHEN
- SCHOPT_SDHE, SCHOPT_SDHEN
- SCHOPT_SCHE, SCHOPT_SCHEN
- SCHOPT_SCHC
- SCEN_SAVE, SCEN_LOAD
- 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
```

Příkazy **SCHOPT_SPHE** a **SCHOPT_SPHEN** provádí optimalizaci časových plánů (jejich přeplánování) podle budoucích spotových cen. Zkratka „SPHE“ znamená „Spot Price in Hours – Electricity“. Písmenko „N“ na konci znamená „negativní“. Příkazy určují podle budoucích spotových cen elektřiny, v jakých následujících maximálně 24 hodinách se má daný plán spouštět a v jakých ne. Příkaz SCHOPT_SPHE vybrané hodiny označuje pro spouštění daného plánu, ostatní jako zakázané. Oproti tomu SCHOPT_SPHEN vybrané hodiny označuje pro daný plán jako zakázané, ostatní jako povolené. První hodina pro optimalizaci je aktuální hodina do nejbližší celé hodiny.

Příklad použití příkazu:

```
SCHOPT_SPHE 211 MIN 5 12
```

Syntaxe:

Význam výše uvedeného příkladu: „Vezmi aktuální spotovou cenu a všechny budoucí ceny v této a následujících 12ti hodinách. Vyhledej mezi nimi 5 hodinových úseků (nemusí být v souvislé řadě za sebou), které mají nejnižší cenu. U existujícího časového plánu s ID=211 označ vyhledané hodiny jako ty, ve kterých se má časový plán spouštět.

Důležité poznámky:

- Časový plán s daným ID (druhý parametr) **musí existovat** a musí mít **v parametru *WeekDay* speciální hodnotu *OPTH***, aby se řídil výsledkem optimalizace.
- Třetí parametr *MIN* nebo *MAX* udává, zda se hledají **minimální nebo maximální ceny**.
- **První hodina je aktuální hodina**. Špičkové hodnoty pro optimalizaci se tedy hledají od času spuštění příkazu zpět od nejbližší celé hodiny. Pokud tedy spustíme příkaz třeba ve 14:13 s výhledem na 12 hodin dopředu, bude zkoumáno časové období mezi dnešními 14:00 hodin a zítřejšími 02:00 (celkem dvanáct hodinových úseků). Tento příkaz je tedy většinou vhodné naplánovat na počátku hodiny.
- Daný plán může mít **jakékoliv parametry včetně opakování**, výjimek, apod. Může být i neaktivní. Například, máme-li nějaký plán s opakováním každých 5 minut a budeme ho optimalizovat pomocí příkazu SCHOPT_SPHE, bude se provádět každých 5 minut pouze v optimalizací vybraných hodinách. Mimo ně se nebude provádět vůbec.
- Minimální hodnota čtvrtého parametru (počet hodin) je 1 a maximální hodnota je hodnota pátého parametru mínus 1.
- Minimální hodnota pátého parametru (v kolika následujících hodinách hledat) je 2 a maximální 24.
- Jako poslední **pátý parametr lze přidat písmeno „C“** („continual“). Je-li zadán, příkaz vyhledá **souvislý úsek zadaného počtu hodin**, který má v součtu nejnižší cenu. Hodí se například v situacích, kdy je nutné dokončit nějaký započatý cyklus (například prací program pračky).

Příkazy **SCHOPT_SDHE** a **SCHOPT_SDHE** provádí optimalizaci časových plánů (jejich přeplánování) podle budoucích spotových cen se zohledněním minimálního rozdílu mezi horní a dolní cenovou hladinou. Tento způsob řízení se může s výhodou uplatňovat například při řízení střídačů, kdy se střídače přepínají mezi různými režimy – například preference využití vlastní energie nebo preference přetoku energie do sítě.

Zkratka „SDHE“ znamená „Spot Difference in Hours Electricity“. Písmenko „N“ na konci znamená „negativní“. Příkazy určují podle budoucích spotových cen elektřiny, v jakých následujících maximálně 24 hodinách se má daný plán spouštět a v jakých ne. Příkaz SCHOPT_SDHE vybrané hodiny označuje pro spouštění daného plánu, ostatní jako zakázané. Oproti tomu SCHOPT_SDHEN vybrané hodiny označuje pro daný plán jako zakázané, ostatní jako povolené. První hodina pro optimalizaci je aktuální hodina do nejbližší celé hodiny.

Příklad použití příkazu:

```
SCHOPT_SDHE 211 MIN 4 12 65.5
```

Min. rozdíl

Syntaxe:

Příkaz *ID čas. plánu* *min/max?* *Počet hledaných hodin* *Hledání v X násl. hodinách*

Význam výše uvedeného příkladu: „Vezmi aktuální spotovou cenu a všechny budoucí ceny v této a následujících 12ti hodinách. Vyhledej mezi nimi 4 hodinové úseky (nemusí být v souvislé řadě za sebou), které mají nejvyšší cenu. Tuto cenu zprůměruj a odečti zadanou hodnotu 65,5 EUR pro získání limitu MIN. U existujícího časového plánu s ID=211 označ jako aktivní ty hodinové úseky, které mají ceny nižší nebo rovnu limitu MIN.

Důležité poznámky:

- Vždy se berou jako základ výpočtu **nejvyšší ceny** (nikoliv nejnižší). Ve výše uvedeném příkladu jsou 4.
- Časový plán s daným ID (druhý parametr) **musí existovat** a musí mít **v parametru *WeekDay* speciální hodnotu *OPTH***, aby se řídil výsledkem optimalizace.
- Třetí parametr *MIN* nebo *MAX* udává, zda se hledají **minimální nebo maximální ceny**.
- **První hodina je aktuální hodina**. Špičkové hodnoty pro optimalizaci se tedy hledají od času spuštění příkazu zpět od nejbližší celé hodiny. Pokud tedy spustíme příkaz třeba ve 14:13 s výhledem na 12 hodin dopředu, bude zkoumáno časové období mezi dnešními 14:00 hodin a zítřejšími 02:00 (celkem dvanáct hodinových úseků). Tento příkaz je tedy většinou vhodné naplánovat na počátku hodiny.
- Poslední parametr je **hodnota minimálního rozdílu** udaná **v EUR / 1 MWh**. Zadává se jako číslo s desetinnou tečkou.
- Daný plán může mít **jakékoliv parametry včetně opakování**, výjimek, apod. Může být i neaktivní. Například, máme-li nějaký plán s opakováním každých 5 minut a budeme ho optimalizovat pomocí příkazu SCHOPT_SCHE, bude se provádět každých 5 minut pouze v optimalizací vybraných hodinách. Mimo ně se nebude provádět vůbec.
- Minimální hodnota čtvrtého parametru (počet hodin) je 1 a maximální hodnota je hodnota pátého parametru mínus 1.
- Minimální hodnota pátého parametru (v kolika následujících hodinách hledat) je 2 a maximální 24.

Příkazy **SCHOPT_SCHE** a **SCHOPT_SCHEN** provádí optimalizaci časových plánů (jejich přeplánování) podle budoucích spotových cen se zohledněním dodatečných nákladů na 1 MWh. Tyto dodatečné náklady se mohou s výhodou uplatňovat například při akumulaci energie (nabíjení / vybíjení baterií), kdy baterie mají svoji životnost určenu předpokládaným počtem nabíjecích cyklů. Každý takový nabíjecí cyklus lze ohodnotit nějakými náklady (amortizace; obvykle pořizovací cena / počtem předpokládaných nabíjecích cyklů). Nabíjet/vybíjet baterie je pak vhodné pouze v situaci, kdy rozdíl spotových cen je vyšší než příslušná amortizace.

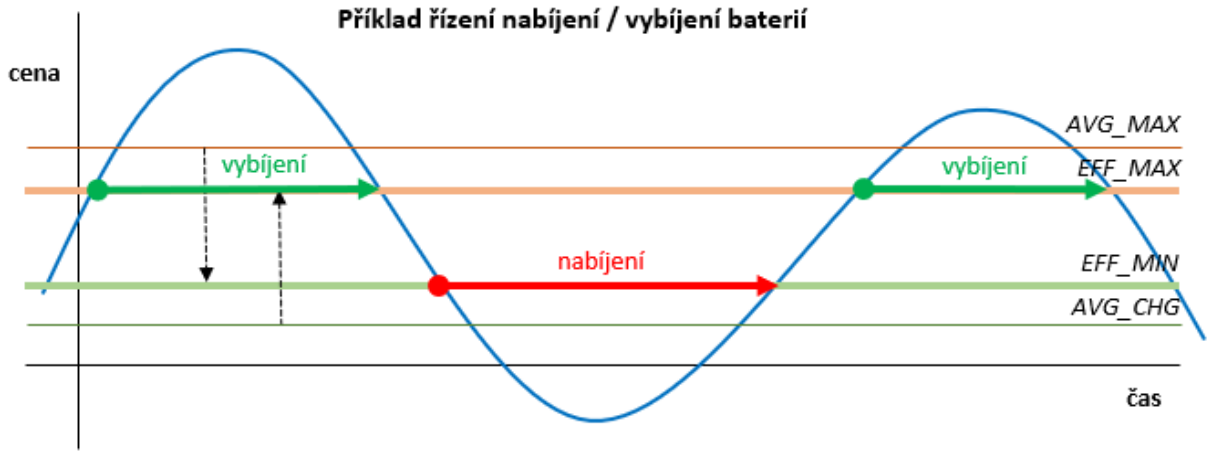
Zkratka „SCHE“ znamená „Spot incl. Costs in Hours Electricity“. Písmenko „N“ na konci znamená „negativní“. Příkazy určují podle budoucích spotových cen elektřiny, v jakých následujících maximálně 24 hodinách se má daný plán spouštět a v jakých ne. Příkaz SCHOPT_SCHE vybrané hodiny označuje pro spouštění daného plánu, ostatní jako zakázané. Oproti tomu SCHOPT_SCHEN vybrané hodiny označuje pro daný plán jako zakázané, ostatní jako povolené. První hodina pro optimalizaci je aktuální hodina do nejbližší celé hodiny.

Příklad výpočtu amortizace: Pořizovací cena baterie = 30.000 Kč, kapacita baterie = 7,1 kWh a předpokládaný počet nabíjecích cyklů = 6.000. Celková uložitelná energie do baterií je tedy 42.600 kWh a vydělením přepokládaným počtem nabíjecích cyklů dostaneme hodnotu amortizace přibližně 0,7 Kč / kWh – tj. 27,5 EUR / MWh. Číslo 27,5 by tak vstupovalo jako parametr do příkazu.

Pro bližší vysvětlení způsobu určování optimalizovaných hodin je nutné vysvětlit některé proměnné, se kterými výpočet pracuje:

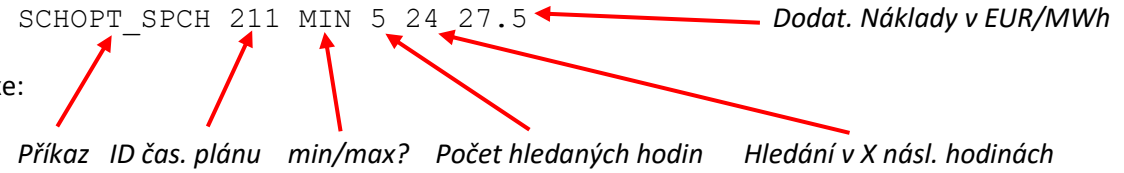
- AMO = hodnota dodatečných nákladů (amortizace) v EUR, která se zadává jako parametr
- AVG_MAX = průměrná cena několika hodin s nejvyšší cenou. Počet těchto špičkových hodin se zadává jako parametr
- EFF_MIN = hodnota vypočítaná jako AVG_MAX – AMO. Pod tímto limitem se vyplatí baterie nabíjet
- AVG_CHG = průměrná cena všech hodin, jejichž cena je pod limitem EFF_MIN. Jedná se o odhad průměrné ceny nabíjení baterií
- EFF_MAX = hodnota vypočítaná jako AVG_CHG + AMO. Jedná se o odhad ceny, za kterou se vyplatí vybíjet baterie.

Pro bližší vysvětlení způsobu určování optimalizovaných hodin si vezměme následující příklady vývoje spotové ceny a různé výše dodatečných nákladů při řízení nabíjení/vybíjení baterií:



Příklad použití optimalizačního příkazu:

Syntaxe:



Význam výše uvedeného příkladu: „Vezmi aktuální spotovou cenu a všechny budoucí ceny v této a následujících 24 hodinách. Vyhledej mezi nimi 5 hodinových úseků s **nejvyšší** cenou a vypočítej průměr AVG_MAX. Od něj odečti hodnotu dodatečných nákladů AMO a vypočítej hodnotu EFF_MIN. Ze všech cen pod limitem EFF_MIN vypočítej průměr jako AVG_CHG. K němu AVG_CHG přičti hodnotu AMO a vypočítej cenu EFF_MAX. Vyhledej všechny hodinové úseky v daném intervalu, které **mají spotovou cenu <= EFF_MIN**. U existujícího časového plánu s ID=211 označ vyhledané hodiny jako ty, ve kterých má být časový plán aktivní.

Důležité poznámky:

- Časový plán s daným ID (druhý parametr) **musí existovat** a musí mít **v parametru *WeekDay* speciální hodnotu *OPTH***, aby se řídil výsledkem optimalizace.
- Třetí parametr *MIN* nebo *MAX* udává, zda se hledají **minimální nebo maximální ceny**.
- **První hodina je aktuální hodina**. Špičkové hodnoty pro optimalizaci se tedy hledají od času spuštění příkazu zpět od nejbližší celé hodiny. Pokud tedy spustíme příkaz třeba ve 14:13 s výhledem na 12 hodin dopředu, bude zkoumáno časové období mezi dnešními 14:00 hodin a zítřejšími 02:00 (celkem dvanáct hodinových úseků). Tento příkaz je tedy většinou vhodné naplánovat na počátku hodiny.
- Poslední parametr je **hodnota dodatečných nákladů** (amortizace) udaných **v EUR / 1 MWh**. Zadává se jako číslo s desetinnou tečkou.
- Daný plán může mít **jakékoliv parametry včetně opakování**, výjimek, apod. Může být i neaktivní. Například, máme-li nějaký plán s opakováním každých 5 minut a budeme ho optimalizovat pomocí příkazu SCHOPT_SCHE, bude se provádět každých 5 minut pouze v optimalizací vybraných hodinách. Mimo ně se nebude provádět vůbec.
- Minimální hodnota čtvrtého parametru (počet hodin) je 1 a maximální hodnota je hodnota pátého parametru mínus 1.
- Minimální hodnota pátého parametru (v kolika následujících hodinách hledat) je 2 a maximální 24.

Příkaz **SCHOPT_COMP** provádí označení hodin daného časového plánu „mezi“ dvěma jinými dříve optimalizovanými plány. Používá se v případech, kdy v časech s maximálními cenami chceme zasílat nějaké příkazy A, v časech s minimálními cenami chceme zasílat nějaké příkazy B a v ostatních časech (tedy „mezi“ nimi) chceme zasílat jiné příkazy C.

Zkratka „COMP“ znamená „Complement“. Příkaz vezme dva před tím optimalizované plány a hodiny, ve kterých není aktivní ani jeden z nich, označí jako aktivní pro právě optimalizovaný plán.

Příklad použití optimalizačního příkazu:

Syntaxe:

```
SCHOPT_COMP 103 101,102
```

Příkaz *ID čas. plánu* *ID plánu 1* *ID plánu 2*

6.6.2 Příklad praktického nastavení optimalizace č. 1

Mějme například následující situaci: Máme do datové sítě připojeno externí zařízení Shelly č. 10 a na ně máme zapojen ohřev teplé vody v bojleru. Bojler chceme zapínat v noci (mezi 20 – 6 hodin) po dobu 5ti hodin a přes den (mezi 6 – 20 hodin) po dobu 3 hodin. Vždy v intervalech s minimálními spotovými cenami. Budeme využívat toho, že zařízení Shelly má implementovány bezpečnostní konstanty a umí se automaticky vypnout po zadané době. Takže nemusíme zařízení aktivně vypínat – pokud přestaneme posílat signály pro zapnutí, do dvou minut se vypne samo. Pak v konfiguraci postupujeme takto:

a) Zadáme do konfiguračního souboru **3 nové plány**:

```
[schedule]
#Optimalizace nahříváni bojleru od 06:00 (14 hodin)
ScheduleID=201
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT1.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 06:00:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 06:00:00

[schedule]
#Optimalizace nahříváni bojleru od 20:00 (10 hodin)
ScheduleID=202
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT2.mac
MaxTimeOn=
```

```
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 20:00:00
DoDateTo=01.01.2033 17:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 20:00:00

[schedule]
#Udržení zapnutého bojleru – optimalizace podle spotových cen
ScheduleID=211
Active=true
RelationType=Shelly
Action=TurnOn
MaxTimeOn=2min
MaxTimeOff=
Satellite=10
OrderNum=0
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1min
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:00:00
```

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

```
##Optimalizace pro 06:00-20:00
#
##Výběr 3 nejlevnějších hodin v následujících 14ti hodinách
SCHOPT_SPHE 211 MIN 3 14
EXIT
```

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

```
##Optimalizace pro 20:00-06:00
#
##Výběr 5 nejlevnějších hodin v následujících 10ti hodinách
SCHOPT_SPHE 211 MIN 5 10
EXIT
```

Významné hodnoty jsou zvýrazněny žlutě. Po restartu služby se začne automaticky optimalizovat. Ve vybraných minimálních hodinách se každou minutu pošle signál zařízení Shelly, aby drželo bojler zapnutý. Mimo vybrané hodiny se tyto signály přestanou posílat a zařízení se automaticky do dvou minut vypne.

6.6.3 Příklad praktického nastavení optimalizace č. 2

Mějme situaci podobnou předchozí. Nebudeme ale využívat bezpečnostních konstant maximálních časů, ale v každé vybrané hodině pošleme jeden signál pro zapnutí a v ostatních hodinách jeden signál pro vypnutí. Potom postupujeme v konfiguraci takto:

a) Zadáme do konfiguračního souboru **4 nové plány**:

```
[schedule]
#Optimalizace nahříváni bojleru od 06:00 (14 hodin)
ScheduleID=201
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT1.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 06:00:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 06:00:00

[schedule]
#Optimalizace nahříváni bojleru od 20:00 (10 hodin)
ScheduleID=202
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT2.mac
MaxTimeOn=
```

```
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 20:00:00
DoDateTo=01.01.2033 17:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 20:00:00

[schedule]
#Zapínání bojleru - optimalizace podle spotových cen
ScheduleID=211
Active=true
RelationType=Shelly
Action=TurnOn
MaxTimeOn=
MaxTimeOff=
Satellite=10
OrderNum=0
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1h
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:01:00

[schedule]
#Vypínání bojleru - optimalizace podle spotových cen
ScheduleID=212
Active=true
RelationType=Shelly
Action=TurnOff
MaxTimeOn=
MaxTimeOff=
Satellite=10
OrderNum=0
IndicatorType=
```

```
High=
Low=
Repeatable=true
RepeatTime=1h
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 05:50:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:00:30
```

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

```
##Optimalizace pro 06:00-20:00
#
##Interval 3 hodin v nasledujicich 14ti hodinach
SCHOPT_SPHE 211 MIN 3 14
SCHOPT_SPHEN 212 MIN 3 14
EXIT
```

První příkaz naplánuje zapnutí bojleru ve 3 nejlevnějších hodinách. Druhý naplánuje vypnutí mimo tyto 3 hodiny.

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

```
##Optimalizace pro 20:00-06:00
#
##Interval 5ti hodin v nasledujicich 10ti hodinach
SCHOPT_SPHE 211 MIN 5 10
SCHOPT_SPHEN 212 MIN 5 10
EXIT
```

První příkaz naplánuje zapnutí bojleru v 5ti nejlevnějších hodinách. Druhý naplánuje vypnutí mimo těchto 5 hodin.

Významné hodnoty jsou zvýrazněny žlutě. Po restartu služby se začne automaticky optimalizovat. Ve vybraných minimálních hodinách se vždy v danou hodinou a jednu minutu pošle signál zařízení Shelly, aby zapnulo bojler. Mimo vybrané hodiny se v čase 0 minut, 30 vteřin pošle signál pro vypnutí.

6.6.4 Příklad praktického nastavení optimalizace č. 3

Mějme situaci, kdy chceme řídit nabíjení a vybíjení baterie u střídače podle spotové ceny, ale se zohledněním její amortizace. Zde potřebujeme tři různé příkazy (režimy střídače): a) režim prioritního nabíjení baterie; b) režim prioritního vybíjení baterie; c) režim „mezi“, kdy není preferováno ani nabíjení, ani vybíjení. Na počátku každé hodiny pošleme jeden ze tří možných signálů pro případnou

změnu požadovaného režimu. Řízení střídače bude realizováno pomocí protokolu Modbus, zapsáním určitých hodnot do několika registrů.

Postupujeme v konfiguraci takto:

a) Zadáme do konfiguračního souboru **4 nové plány**:

```
[schedule]
#Optimalizace řízení nabíjení/vybíjení baterií – vždy o půlnoci
ScheduleID=201
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/SPOTPRICE_OPT1.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=1d
RepeatWeek=PO-NE
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 00:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:00:00

[schedule]
#Prioritizace nabíjení baterií
ScheduleID=211
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/BATERIE_NAB.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=15min
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 00:00:00
ExceptDateFrom=
```

```
ExceptDateTo=
StartDate=01.01.2023 00:00:30

[schedule]
#Prioritizace vybíjení baterií
ScheduleID=212
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/BATERIE_VYB.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=15min
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 00:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:00:30

[schedule]
#Prioritizace nabíjení baterií
ScheduleID=213
Active=true
RelationType=Macro
Action=/media/extended/spotProcessor/BATERIE_MEZI.mac
MaxTimeOn=
MaxTimeOff=
Satellite=1
OrderNum=1
IndicatorType=
High=
Low=
Repeatable=true
RepeatTime=15min
RepeatWeek=OPTH
DoDateFrom=01.01.2023 00:00:00
DoDateTo=01.01.2033 00:00:00
ExceptDateFrom=
ExceptDateTo=
StartDate=01.01.2023 00:00:30
```

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

```
##Optimalizace pro řízení baterií
#
#Průměr 4 špičkových cen v následujících 24 hodinách, AMO=27,5
SCHOPT_SCHE 211 MIN 4 24 27.5
#
#Průměr 5ti špičkových cen v následujících 24 hodinách, AMO=27,5
SCHOPT_SCHE 212 MAX 5 24 27.5
#
#Doplňkové hodiny
SCHOPT_COMP 213 211,212
EXIT
```

První příkaz vybere hodiny s minimální cenou pro nabíjení baterie se zohledněním nákladů na amortizaci podle průměru 4 špičkových cen a označí je do plánu č. 211. Druhý příkaz vybere hodiny s maximální cenou pro vybíjení baterie se zohledněním nákladů na amortizaci podle průměru 5ti špičkových cen a označí je do plánu č. 212. Třetí příkaz označí všechny hodiny, kde jsou plány 211 a 212 neaktivní jako aktivní pro plán 213.

c) Vytvoříme nový soubor makra BATERIE_NAB.mac s následujícím obsahem (příklad pro střídač DEYE a konkrétní preferované proudy):

```
##Příkazy pro prioritizaci nabíjení baterie
#
MDB16S 1-108 185  # Max A charge
MDB16S 1-109 40   #Max A discharge
MDB16S 1-128 185  #Grid charging start capacity point
MDB16S 1-129 1    #Generator charging enable
MDB16S 1-130 1    #Grid charging enable
```

d) Vytvoříme nový soubor makra BATERIE_VYB.mac s následujícím obsahem (příklad pro střídač DEYE a konkrétní preferované proudy):

```
##Příkazy pro prioritizaci vybíjení baterie
#
MDB16S 1-108 0    # Max A charge
MDB16S 1-109 185  #Max A discharge
MDB16S 1-128 0    #Grid charging start capacity point
MDB16S 1-129 1    #Generator charging enable
MDB16S 1-130 0    #Grid charging enable
```

e) Vytvoříme nový soubor makra BATERIE_MEZI.mac s následujícím obsahem (příklad pro střídač DEYE a konkrétní preferované proudy):

```
##Příkazy pro režim řízení baterie podle výkonu
```

```
#
MDB16S 1-108 40   # Max A charge
MDB16S 1-109 40   #Max A discharge
MDB16S 1-128 40   #Grid charging start capacity point
MDB16S 1-129 1    #Generator charging enable
MDB16S 1-130 1    #Grid charging enable
```

Významné hodnoty jsou zvýrazněny žlutě. Po restartu služby se začne automaticky optimalizovat. V optimalizovaných minimálních hodinách se ve střídači nastaví režim prioritního nabíjení, v maximálních prioritního vybíjení a mimo tyto hodiny se nechá střídač pracovat podle svého algoritmu.

Namísto optimalizace plánů s jednoduchým příkazem zapnutí a vypnutí lze samozřejmě používat i časové plány spouštějící programová makra a obsahují mnoho různých příkazů a podmínek. Možností a kombinací je mnoho – ty už ale záleží na konkrétních potřebách dané implementace.

Příkazy **SCEN_SAVE** a **SCEN_LOAD** slouží k uložení úplné konfigurace služby spotProcessor pod nějakým číslem a poté její případné obnovení (zkratka „SCEN“ znamená „Scenario“). Používá se například v situacích, kdy jiný způsob a parametry řízení jsou vyžadovány o víkendech, jiné v pracovních dnech a jiné třeba o prázdninách.

Příklad uložení úplné konfigurace (scénáře) pod číslem 3:

```
SCEN_SAVE 3 Můj název scénáře
```

Syntaxe:

Příkaz *Č. scénáře* *Uživatelský název*

Příklad obnovení úplné konfigurace (scénáře) ze zálohy pod číslem 3:

```
SCEN_LOAD 3
```

Příklad obnovení úplné konfigurace ze zálohy pod číslem 3 za podmínky, že spotová cena v dané hodině poklesne pod 5,5 EUR:

```
IFINDM 1-1 PRIC 1H < 5.5 SCEN_LOAD 3
```

Důležité poznámky:

- **Čísla scénářů mohou být mezi 1 a 9**.
- Obnovení scénáře znamená kompletní obnovení konfiguračních souborů a **restart celé jednotky**. Tato akce tak může trvat desítky sekund.

- Uživatelský popis je pouze informativní popisek, který se ke scénáři ukládá jako jeho název. Ve webové aplikaci je pak vidět včetně podrobnějšího popisu. Může obsahovat mezery, čárky, pomlčky, apod.
- Je-li aktivní nějaký konkrétní scénář a má se zároveň tentýž obnovit, akce se neprovede.
- Při pokusu o obnovení z čísla, které neosahuje zálohu, se nic neprovede.

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
```

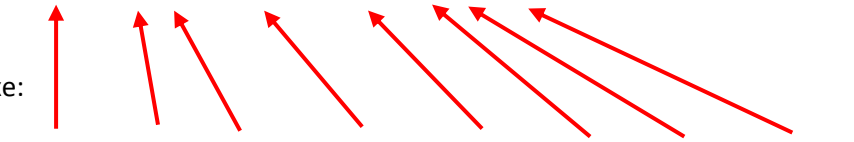
6.6.5 Nově implementované příkazy pro Modbus

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, spusť navazující příkaz (`MDB06 3-0 500`).

Příklad porovnání:

a) Zadaná hodnota: `01101`
Přečtená hodnota: `0011001100001101`
Výsledek: **PRAVDA**

b) Zadaná hodnota: `00001101`
Přečtená hodnota: `0101010100001001`
Výsledek: **NEPRAVDA**

c) Zadaná hodnota: 0
Přečtená hodnota: 0011001100001101
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). Před očekávanou hodnotu je možné uvést jedno ze znamének „<“, „>“ nebo „=“. Není-li žádné znaménko uvedeno, vyhodnocuje se přesná shoda (tedy jakoby tam bylo „=“).

Příkazy MDB03 a MDB04 používají protokol Modbus TCP, příkazy MDB03S a MDB04S pak protokol Modbus RTU.

Příklad příkazu:

```
MDB03S 2-32 >1234 MDB06S 3-0 500
```

Syntaxe:

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

Význam: Přečti 2 Byte počínaje adresou 32 ze zařízení se SlaveID=2 pomocí Modbus funkce 03. Přečtené 2 Byte převeď na znaménkové celé číslo a to porovnej se zadanou hodnotou 1.234. Pokud bude přečtené číslo větší, spusť 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:

```
MDB05 2-120 1
```

Syntaxe:

Příkaz SlaveID-Registr hodnota

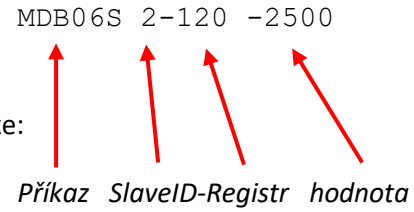
Význam: Zapiš logickou 1 (=ZAPNUTO) do registru s adresou 120 v zařízení se SlaveID=2 pomocí Modbus funkce 05.

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:

Syntaxe:



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

Příkazy MDB06 a MDB06S mohou **zasílat také hodnoty měřených veličin z indikátorů a spotřebičů** násobené příslušných násobitelem, včetně speciálních indikátorů pro měření spotových cen elektřiny a plynu. Hodnoty jsou zasílány jako celá znaménková čísla.

Typy měřených hodnot mohou být následující:

Kód typu	Popis	Násobitel
POWE	Elektrický výkon [W]	1
CURR	Elektrický proud [A]	1000
VOLT	Elektrické napětí [V]	10
PEAKC	Špičkový (maximální) proud [A]	1000
PEAKV	Špičkové (minimální) napětí [V]	10
CONS	Kumulovaná spotřeba [KWh]	1
TEMP	Teplota [°C]	10
HUMI	Vlhkost [%]	10
LIGH	Intenzita osvětlení [Lux]	1
PRIC	Cena (například spotová pro elektřinu a plyn)	100
PRID	Cena včetně distribučního poplatku	100
AMOU	Množství obchodované elektřiny	100
BALA	Saldo obchodované elektřiny	100
EXPO	Hodnota exportu elektřiny	100
IMPO	Hodnota importu elektřiny	100
MINP	Minimální cena plynu	100
MAXP	Maximální cena plynu	100
LAST	Poslední cena plynu	100
INDX	Index OTE pro obchodování s plynem	100

Násobitel je konstanta, kterou je nutné přečtenou hodnotu vydělit. Například cenové údaje je nutné vydělit 100, aby se dostala cena v EUR (zasílá se v centech, setinách EUR).

Příklad příkazu pro aktivní zasílání aktuální spotové ceny elektřiny:

```
MDB06 1-32 INDVAL 1-1 PRIC
```

Syntaxe:

Příkaz SlaveID-Registr Hodnota zařízení SatNo-OrdNo Typ hodnoty

Význam: Zapiš aktuální hodnotu ceny ze zařízení 1-1 (speciální indikátor pro měření spotové ceny) do registru s adresou 32 v zařízení se SlaveID=1 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. Nezadané 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.

Příkazy **SHEON**, **SHEOFF** a **SHEPLS** jsou podobné příkazům DEVON, DEVOFF a DEVPLS. Ovšem používají http aplikační protokol Shelly. První příkaz zapíná, druhý vypíná a třetí přepíná stav zařízení.

Příklad příkazu:

```
SHEON 10-0
```

Syntaxe:

Příkaz SatNo-OrdNum

Význam: Zašli na IP adresu (např. 10.0.1.81 podle definice satelitu č. 10 v seznamu IP adres) příkaz zapnutí prvního logického zařízení (URL: http://10.0.1.81/relay/0?turn=on)

7 Specifika řízení komerčních regulátorů

SpotProcessor může řídit mimo jiné i komerční regulátory Wattrouter a GreenBonO. Specifika jejich řízení jsou uvedeny níže.

7.1 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 0
```

7.2 Specifika řízení regulátoru GreenBonO

Příkaz MDB16S může být využíván pro řízení regulátoru GreenBonO od společnosti Yorix. K fyzickému propojení je nutné použít linku RS-485 a následujícími parametry přenosu 9600-8-N-1:

```
SERIAL_BAUDRATE=9600
SERIAL_PARBITS=8N1
```

K externímu ovládání přes Modbus slouží registr č. 14. Pomocí zápisu určité hodnoty lze 6 ovládaných relé:

- Nuceně zapnout **vybrané výstupy**
- Nuceně vypnout **všechny výstupy**
- Povolit opět vlastní regulaci regulátorem

Obsahem 16ti bitového řídícího slova jsou ve vyšším Byte řídící bity dle následující tabulky. Nižší Byte je vždy nulový:

Tabulka bitů v řídícím Byte:

Číslo bitu	Význam	HEX hodnota	DEC hodnota
1	Maska SSR1	0x01	1
2	Maska SSR2	0x02	2
3	Maska SSR3	0x04	4
4	Maska SSR4	0x08	8
5	Maska SSR5	0x10	16
6	Maska SSR6	0x20	32
7	1 = VYPNOUT všechny výstupy; 0 = nic	0x40	64
8	1 = ZAPNOUT vybrané výstupy dle masky; 0 = nic	0x80	128

Příklad nuceného zapnutí relé 1, 3 a 5 (slave = 1, registr = 14):

```
MDB16S 1-14 0x9500
```

Příklad nuceného vypnutí relé 1, 3 a 5 (slave = 1, registr = 14):

```
MDB16S 1-14 0x5500
```

Příklad návratu k vlastní regulaci (slave = 1, registr = 14):

```
MDB16S 1-14 0x3f00
```

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	##<ALW> - Prikazy, ktere se vykonavaji vzdy	Návěští sekce ALWAYS
4	#EXIT	Podmíněné vypnutí regulace
5	MDB16 1-32 INDVAL 1-1 PRIC	Zasílání spotové ceny do WR
6	#	
7	##<COND> - Podminky splneni definovanych limitu	Návěští sekce CONDITION
8	IFINDS 1-1 OFF GOTO 22	Je-li cena nad/pod limitem,
9	IFINDS 1-1 ON GOTO 36	skoč na řádku č...
10	#IFINDS AND(1-1,1-2) OFF GOTO 22	
11	#IFINDS AND(1-1,1-2) ON GOTO 36	
12	#IFINDS OR(1-1,1-2) OFF GOTO 22	
13	#IFINDS OR(1-1,1-2) ON GOTO 36	
14	#	
15	##<ELSE> - Prikazy pro rizeni mezi limity	Návěští sekce ELSE
16	#MDB16S 1-14 0x3f00	
17	IFINDM 1-1 PRIC 1H > 95 EXECOMMAND REL03-OFF	Při cene > 95 vyp. rele 3
18	IFINDM 1-1 PRIC 1H < 106.5 MDB06 1-13 1000	Při cene < 106,5 vyp. SSR 6
19	#EXECOMMAND SPOTPRICE_between.sh	
20	EXIT	Dále nepokračovat
21	#	
22	##<MIN> - Prikazy pro pri cene nizsi nez MIN	Návěští sekce MIN
23	MDB16 1-0 1000 0 1000 0 0 0 0 0 0 0 0 0 0 0 0	Zapnout SSR 1 a SSR 3
24	MDB06S 2-12 500	Zapsat 500 do reg. 12
25	MDB06 1-1 1000	Zapnout SSR 2
26	MDB06 1-4 1000	Zapnout SSR 5
27	SHEON 10-0	Zapnout Shelly zásuvku
28	#MDB06S 2-5 1000	Neprovádí se
29	#MDB06S 2-6 1000	
30	#MDB06S 2-7 1000	
31	EXECOMMAND REL01-ON	Zapnout ext. relé 1
32	EXECOMMAND REL02-ON	Zapnout ext. relé 2
33	#EXECOMMAND REL03-ON	Neprovádí se
34	EXIT	Dále nepokračovat
35	#	
36	##<MAX> - Prikazy pro pri cene vyssi nez MAX	Návěští sekce MAX
37	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
38	MDB06S 2-12 0	Zapsat 0 do reg. 12
39	MDB06 1-10 1000	Vypnout SSR 3
40	MDB06 1-11 1000	Vypnout SSR 4
41	SHEOFF 10-0	Vypnout Shelly zásuvku
42	#MDB06S 2-13 1000	Neprovádí se
43	#MDB06S 2-14 1000	
44	#MDB06S 2-15 1000	
45	EXECOMMAND REL01-OFF	Vypnout ext. relé 1
46	EXECOMMAND REL02-OFF	Vypnout ext. relé 2
	#EXECOMMAND REL03-OFF	Neprovádí se
	EXIT	Dále nepokračovat

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

Řádky uvozené „##<ALW>", „##<COND>", „##<ELSE>", „##<MIN>" a „##<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
```