

**ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA ELEKTROTECHNICKÁ**

Katedra elektromechaniky a výkonové elektroniky

DIPLOMOVÁ PRÁCE

**Návrh inteligentního řízení fotovoltaického (FV) systému
s akumulátory dle předpovědi slunečního záření**

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta elektrotechnická

Akademický rok: 2017/2018

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Lukáš ČÁP**

Osobní číslo: **E15N0007K**

Studijní program: **N2644 Aplikovaná elektrotechnika**

Studijní obor: **Aplikovaná elektrotechnika**

Název tématu: **Návrh inteligentního řízení fotovoltaického (FV) systému s akumulátory dle předpovědi slunečního záření**

Zadávací katedra: **Katedra elektroenergetiky a ekologie**

Z á s a d y p r o v y p r a c o v á n í :

1. Popište různé typy FV systémů, jejich zapojení a použití.
2. Analyzujte možnosti a způsoby předpovědi slunečního záření i profily spotřeby elektřiny v budově.
3. Zpracujte návrh metodiky a inteligentního řízení FV systému s akumulací dle předpovědi slunečního záření a profilů spotřeby elektřiny v budově.

Rozsah grafických prací: **podle doporučení vedoucího**

Rozsah kvalifikační práce: **40 - 60 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

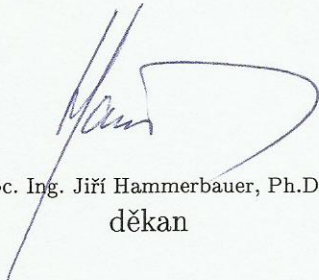
Seznam odborné literatury:

Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.


Vedoucí diplomové práce: **Prof. Ing. Jan Škorpil, CSc.**
Katedra elektroenergetiky a ekologie

Datum zadání diplomové práce: **10. října 2017**

Termín odevzdání diplomové práce: **24. května 2018**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Ing. Karel Noháč, Ph.D.
vedoucí katedry

V Plzni dne 10. října 2017

Abstrakt

Tématem diplomové práce je „Vypracování návrhu inteligentního řízení fotovoltaického (FV) systému s akumulátory dle předpovědi slunečního záření.“. Řešená práce se zabývá popisem součástí, principů a možnosti využití FV systémů. Nedílnou součástí je popis možných druhů předpovědi slunečního záření včetně profilu spotřeby elektřiny v budově. Závěr práce je věnován problematice návrhu metodiky a inteligentního řízení FV systému s akumulací na základě předpovědi slunečního záření a profilů spotřeby elektřiny v budově. Navržený systém řízení byl zrealizován v programu CX-Supervisor od výrobce Omron.

Klíčová slova

elektrotechnika, energetika, ekologie, elektrárna, fotovoltaická elektrárna, FV, FVE, polovodič, PN přechod, fotovoltaický jev, fotovoltaický článek, fotovoltaický panel, polykrystalický panel, monokrystalický panel, amorfní panel, měnič, střídač, akumulátor, regulátor napětí, MTTP, solární systém, , pasivní systém, aktivní systém, grid on systémy, grid off systémy, inteligentní systém, předpověď slunečního záření, sluneční záření, měření, měřicí metody, meteostanice, profil spotřeby, řídicí systém, SCADA, HMI, programovatelný automat, PLC, Omron, CX Supervisor, PV forecast, návrh FV systému, návrh řídicího systému, vizualizace

Abstract

The theme of diploma thesis is „Developing design of the intelligent control photovoltaics system with accumulators according to solar radiation forecast“. Thesis deals with description of the components, principles and using photovoltaics systems. One part is the description of the possible ways of solar radiation forecast, including consumption profile of the building. The conclusion of the thesis is devoted to design of an intelligent managing system of the photovoltaics system with energy accumulation based on the solar radiation forecast and the electricity consumption profiles in the building. Intelligent system was created and developed in the Omron´s program CX-Supervisor

Key words

electrotechnics, energetics, ecology, power plant, photovoltaic power plant, FV, FVE, semiconductor, PN transition, photovoltaic phenomenon, photovoltaic cell, photovoltaic panel, polycrystalline panel, monocrystalline panel, amorphous panel, inverter, inverter, voltage regulator, MTTP, solar system, active systems, passive systems, grid on systems, grid off systems, intelligent system, solar radiation forecast, solar radiation, measurement, measuring methods, weather stations, consumption profile, control system, SCADA, HMI, programmable logic computer, PLC, Omron, CX Supervisor, PV forecast, design of photovoltaic system design of control system, visualization.

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů uvedených v seznamu, který je součástí této diplomové práce.

Dále prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.



.....

podpis

V Plzni dne 19.5.2018

Lukáš Čáp

Poděkování

Tímto bych rád poděkoval vedoucímu diplomové práce Prof. Ing. Janu Škorpilovi, CSc. za podporu, cenné a profesionální rady, připomínky a metodické vedení práce. Dále bych chtěl poděkovat panu Václavu Švábovi za poskytnuté rady a software pro návrh FV systému. Poděkování bych dále rád věnoval mému kolegovi Pavlu Holubovi za cenné rady a připomínky, při návrhu inteligentního SCADA systému v programu CX-Supervisor. V neposlední řadě bych chtěl poděkovat panu Janu Schröpferovi za konzultaci, kterou mi poskytl a také mému zaměstnavateli Aisan Industry Czech s.r.o. za poskytnutí licence k programu CX-Supervisor.

Obsah

OBSAH	7
SEZNAM SYMBOLŮ A ZKRATEK	10
1. SOLÁRNÍ SYSTÉMY A MOŽNOSTI ŘÍZENÍ	12
1.1 OBECNĚ O FV SYTÉMU	12
1.1.1 Historie fotovoltaiky	12
1.1.2 Princip přeměny slunečního záření na elektrickou energii	13
1.1.2.1 Polovodiče	13
1.1.2.2 PN přechod	15
1.1.2.3 Fotočlánek	15
1.1.2.4 Fotovoltaický panel	16
1.1.2.5 Měniče	18
1.1.3 Solární systém	19
1.1.3.1 Aktivní solární systém	19
1.1.3.2 Pasivní solární systém	19
1.1.3.3 Grid-off systémy	20
1.1.3.4 Grid-on systém	22
1.2 INTELIGENTNÍ ŘÍDÍCÍ SYSTÉM	23
1.2.1 Co je to inteligentní systém	23
1.2.2 Možnost implementace	24
1.2.3 Typy inteligentních systémů na základě řízení	25
1.2.4 Stručný přehled výrobců	26
1.2.4.1 Vantage control	26
1.2.4.2 Bticino	27
1.2.4.3 KNX	28
1.2.5 Výhody a nevýhody inteligentních systémů	28

2	MOŽNOSTI PŘEDPOVĚDI SLUNEČNÍHO ZÁŘENÍ VČETNĚ PROFILU SPOTŘEBY	29
2.1	PŘEDPOVĚĎ.....	29
2.1.1	Sluneční záření.....	30
2.1.2	Předpověď slunečního záření.....	31
2.1.3	Měření slunečního záření na povrchu země	32
2.1.4	Predikce slunečního záření na základě známých modelů.....	33
2.1.5	Měřicí metody	33
2.1.5.1	Angstromova - Prescottova	33
2.1.5.2	Klabzubova	33
2.1.5.3	Hargreavesova.....	33
2.1.5.4	Supitova	34
2.1.5.5	Winslowova	34
2.1.5.6	Interpolační metoda.....	34
2.1.6	Měřicí stanice slunečního záření v ČR	35
2.1.6.1	Univerzitní centrum ČVUT (UCEEB).....	35
2.1.6.2	Meteorologické stanice FIEDLER.....	35
2.1.7	Možné zdroje předpovědi	35
2.1.7.1	GEFS Reforecast v2 Server	35
2.1.7.2	ERDDAP Server	36
2.1.7.3	Solar Radiation Database	36
2.1.7.4	ENERGOCENTRUM PLUS, s.r.o	36
2.1.7.5	Projekt MEDARD.....	36
2.1.7.6	Pacific Islands Ocean Observing System.....	36
2.2	PROFIL SPOTŘEBY BUDOVY	37
2.2.1	Jak lze získat profil spotřeby	37
2.2.1.1	Odhadem bez prvotních dat	37
2.2.1.2	Na základě celkové roční spotřeby.....	37
2.2.1.3	Měření profilu spotřeby:	37
2.2.1.4	Poskytnutí profilu spotřeby distributorem elektrické energie	38
2.2.2	Využití profilu spotřeby pro FV systém	38
2.2.2.1	Systémy a tím spojená úspora elektrické energie.....	38
2.2.2.2	Souvislost profilu spotřeby, FV systému a předpovědi počasí.....	39

3	NÁVRH METODIKY A INTELIGENTNÍHO ŘÍZENÍ FV SYSTÉMU....	40
3.1	CÍL PROJEKTU	40
3.2	TOPOLOGIE NAVRŽENÉHO SYSTÉMU	40
3.2.1	Návrh FV systému	41
3.2.1.1	Popis objektu.....	42
3.2.1.2	Tvorba profilu spotřeby	43
3.2.1.3	Možnost volby FVE.....	44
3.2.1.4	Prvky mnou navrženého FV systému.....	45
3.2.1.5	Zapojení elektroinstalace	50
3.2.2	Získání dat o předpovědi slunečního záření a teplotě.....	52
3.2.2.1	HTML dotaz předpovědi:.....	52
3.2.2.2	Formáty dat	53
3.2.3	Tvorba databáze	55
3.2.3.1	Vzhled databáze:	56
3.2.3.2	Skript databáze:.....	57
3.2.4	Návrh inteligentního systému řízení (SCADA a HMI)	59
3.2.4.1	Funkce navrženého systému	60
3.2.4.2	Režimy řízení navrženého systému:.....	61
3.2.4.3	Otevření databáze v programu CX-Supervisor	65
3.2.4.4	Komunikační protokol navrženého systému	69
3.2.4.5	Komunikační řetězec.....	70
3.2.4.6	Algoritmus řízení navrženého systému	72
3.2.4.7	Vizualizace navrženého systému	76
4	ZÁVĚR.....	78
	SEZNAM LITERATURY A INFORMAČNÍCH ZDROJŮ	79
	SEZNAM PŘÍLOH	83

Seznam symbolů a zkratk

FV	Fotovoltaický, fotovoltaické
FVE.....	Fotovoltaická elektrárna
SCADA.....	Supervisory Control And Data Acquisition
HMI.....	Human machine interface
PLC	Programmable Logic Controller
AC	Alternate current
DC	Directive current
PWM.....	Pulsně šířková modulace
TUV	Teplá užitková voda
M	Měření
PC	Osobní počítač
MPPT	Regulátor nabíjení
Wp	Watt peak
W	Watt
kW.....	Kilo watt
kWh	Kilo watt hodina
U	Napětí
I	Proud
P	Výkon
Si	Křemík
Ge.....	Germanium
B.....	Bor
In	Indium
P	Fosfor
As.....	Astat
Sb	Antimon

Úvod

Současný trend ve zvyšování cen za energie, vede řadu lidí k investicím do systémů zabezpečujících podstatnou úsporu energií. Úspory energií jsou zabezpečeny výrobními technologiemi, vysokou mírou integrace elektronických součástí a vhodně zvoleného systému řízení. Podstatný vliv na funkčnost systému, má právě návrh co nejvhodnějšího řízení pro zvolenou aplikaci a následná podpora ze strany prodejce. Jako zástupce systémů zde uvedu například FV systémy, solární systémy pro ohřev vody, pasivní solární systémy, systémy tepelného čerpadla, systémy pro zadržení vody a systémy pro vytápění. Veliké oblibě těchto systémů nahrává dotační politika státu společně s velice příznivými cenami zmiňovaných systémů, které se díky masovému rozšíření staly dostupnými pro širší část obyvatelstva.

Diplomová práce je rozdělena do tří částí. První část se věnuje popisu základních principů fotovoltaiky, FV panelů a veškerých součástí s nimi souvisejícími, včetně popisu různých druhů FV systémů, jejich zapojení a použití.

Druhá část se věnuje problematice slunečního záření a profilů spotřeby. Prvotně je zmíněn obecný popis slunečního záření s možnostmi jeho měření. Následuje problematika předpovědi slunečního záření a metod, které jsou v současné době využívány pro předpověď. Pro přehled jsou uvedeny některé zahraniční měřicí stanice, včetně stanic nacházejících se na území České republiky, zabývající se problematikou předpovědi slunečního záření. Závěr druhé části se soustřeďuje na problematiku profilů spotřeby, kde dochází k základnímu popisu profilů spotřeby, možnostem jejich získání a následného využití spolu s předpovědí počasí a FV systémy.

Závěrečná třetí část práce se zabývá návrhem inteligentního řízení FV systému s akumulací energie na základě předpovědi slunečního záření a profilů spotřeby. V této části je uveden postup návrhu FV systému s akumulátory pro rodinný dům včetně tvorby ovládacího SCADA systému, využívajícího předpovědi počasí a profilu spotřeby k zabezpečení co nejefektivnějšího využití toku energie.

1. Solární systémy a možnosti řízení

1.1 Obecně o FV systému

Jedná se o systém složený z více prvků, které zajistí výrobu, přenos a distribuci elektrické energie na základě přeměny slunečního záření v elektrickou energii. FV systém je složen z mnoha prvků. Hlavní z nich jsou FV panely, měnič (střídač), regulátor napětí, regulátor nabíjení, akumulátory a další v závislosti na realizovaném projektu. Stavba FV systému je individuální záležitostí a je zcela v kompetenci spotřebitele.

1.1.1 Historie fotovoltaiky

Pojem fotovoltaika je složen ze dvou slov: foto a volta. Slovo foto má původ v řečtině a znamená světlo. Slovo Volt je spjaté s italským hrabětem a fyzikem Allessandrem Giussepem Antoniem Anastasiem Voltou. Tento muž je spjatý s vynálezem baterie, což se samotným fotočlánkem nemá v zásadě nic společného, ale dal základ „know-how“, který se využil při dalších pokusech. Po jeho smrti byla jednotka napětí pojmenována „Volt“. Při následných pokusech s bateriemi a jejich exponováním světlem, objevil Francouz Alexandre Edmond Becquerel přírůstek elektrického napětí. Toto už byl jen malý krůček k objevení polovodičů, což se následně v roce 1876 potvrdilo u Selenu. Na základě těchto informací došlo v roce 1883 k vynálezu selenového fotočlánku, tento fotočlánek však nenašel široké uplatnění z důvodu vysokých finančních nákladů na jeho výrobu. Jev, který umožňoval přeměnu světelné energie na elektrickou „fotojev“ nebyl do této doby pochopen. Popsal ho a vysvětlil až později Albert Einstein, za což následně obdržel Nobelovu cenu. V 50. letech se začali hojně využívat polovodiče, hlavně Křemíku a dalších, u kterých se zjistilo, že jsou na povrchu Země velice rozšířené. Následně v roce 1954 došlo v amerických Bellových laboratořích „Bell Laboratories“ k objevení křemíkového fotočlánku, čímž byl položen základ k dalšímu výzkumu v oblasti fotovoltaiky.[1]

1.1.2 Princip přeměny slunečního záření na elektrickou energii

Pro přeměnu je zapotřebí FV systému, což je systém, který využívá přímé přeměny slunečního záření na elektrický proud. Přeměna je zajištěna pomocí polovodičových součástek „fotočlánků“ u kterých vlivem FV jevu dojde k přeměně dopadající slunečního záření na elektrickou energii různých parametrů. Elektrická energie je dále zpracována na požadovaný výstup, přenesena a využita dle požadavků spotřebitele. Většinou však dojde k přímé spotřebě v prvcích, jako jsou ohřívače vody, elektrické kotle nebo dojde k akumulaci energie v akumulátorech. Pro pochopení principu fotočlánku, je nutné, mít povědomí o polovodičích a proto je zde ve stručnosti zmíním.[2]

1.1.2.1 Polovodiče

Jsou látky s rozdílnou hodnotou vodivosti oproti vodičům a izolantům. Vodivostí se nachází právě mezi vodiči a izolanty a jsou vodivé za přesně stanovených podmínek, proto název polovodiče. Mezi nejznámější polovodiče patří prvky IV. A skupiny periodické tabulky prvků (Ge - germanium, Si - křemík). V zásadě rozeznáváme dva druhy polovodičů. Dle vodivosti rozlišujeme polovodiče s vlastní (přirozenou) vodivostí a polovodiče s nevlastní (příměsovou) vodivostí.[3]

1.1.2.1.1 Vlastní vodivost = vodivost samotného polovodiče

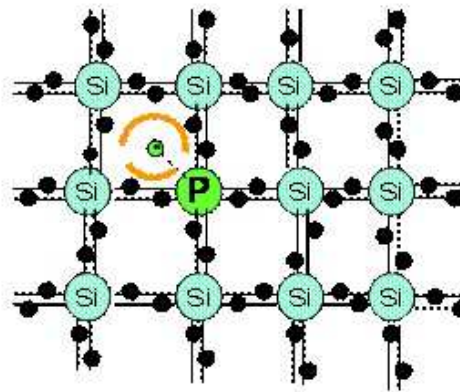
Obecně je samotný prvek nevodivý, jakmile mu však dodáme energie ve formě napětí, tepla, světelného záření či chemického záření. Dojde v jeho struktuře ke změně, která vyústí ve stav, kdy se stane částečně nebo zcela vodivým. Tento princip se nazývá generace párů elektron – díra. Při zahřátí struktury polovodiče dojde k uvolnění vazebních elektronů (záporný náboj), po kterých vzniknou díry (kladný náboj). Díky těmto volným elektronům a dírám jsme již schopni přenášet elektrický proud a látka se stává vodivou. Zároveň však s jevem generace párů elektron – díra, probíhá současně v polovodiči rekombinace, což je zánik uvolněných elektronů na základě rekombinace elektronu s některou ze vzniklých děr. V podstatě se elektron setká s některou z děr a odevzdá jí část své energie a až nebude schopný pokračovat dále, vyplní některou ze vzniklých děr. Samotnou vodivost pak můžeme sledovat po připojení polovodiče k elektrickému zdroji, kde dojde k uspořádanému pohybu děr k zápornému pólu zdroje a elektronů ke kladnému pólu zdroje.[3]

1.1.2.1.2 Nevlastní vodivost (Příměšová vodivost)

Jedná se o vodivost, která je vytvořena zásahem do krystalické mřížky čistého polovodiče. Jsou to nejčastěji prvky IV. A skupiny například (křemík – Si), který je doplněn dalším polovodičem (příměsí), která změní jeho vlastnosti a na základě toho rozeznáváme dva druhy příměšové vodivosti a to vodivost typu N a vodivost typu P.[3]

1.1.2.1.3 Vodivost typu N (elektronová vodivost)

Tento druh vodivosti vytvoříme tak, že do krystalické mřížky čistého polovodiče, nejčastěji křemíku (Si) vložíme atomy prvků IV. Skupiny (P, As, Sb), které v krystalické mřížce vyvolají elektronovou vodivost. Princip je takový, že prvky IV. Skupiny (P, As, Sb) obsahují 5 valenčních elektronů, kde na samotné vazbě s křemíkem se podílí pouze 4 z nich a zbylý pátý elektron (donor) je vázán velmi malou silou, kterou lze snadno porušit přiložením napětí či jiné formy energie čímž vznikne elektronová vodivost.[3]

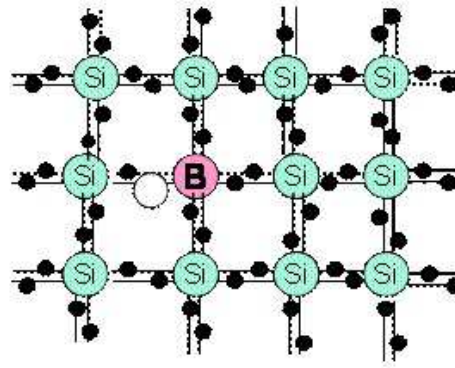


Obr. 1: Příměšová vodivost typu N [3]

1.1.2.1.4 Vodivost typu P (děrová vodivost)

Tento druh vodivosti vytvoříme tak, že do krystalické mřížky čistého polovodiče, nejčastěji křemíku (Si) vložíme atomy prvků III. Skupiny (Ga, B, In), které v krystalické mřížce vyvolají děrovou vodivost.[3]

Princip je takový, že prvky III. Skupiny (Ga, B, In) obsahují pouze 3 valenčních elektrony, takže na vazbě s křemíkem se podílí pouze 3 z nich a místo chybějícího čtvrtého elektronu vznikne díra (akceptor), která může být snadno zaplněna volným elektronem. Proto tuto vodivost nazýváme děrovou.[3]



Obr. 2: Příměsová vodivost typu P [3]

1.1.2.2 PN přechod

Vytvoříme spojením polovodiče typu P a N. PN přechod vznikne právě na jejich rozhraní, kde v místě styku dojde k difuzi částic. Z polovodiče typu N dojde k difuzi elektronů do polovodiče typu P a naopak z polovodiče typu N dojde k difuzi děr do polovodiče typu P a následně rekombinují.[3]

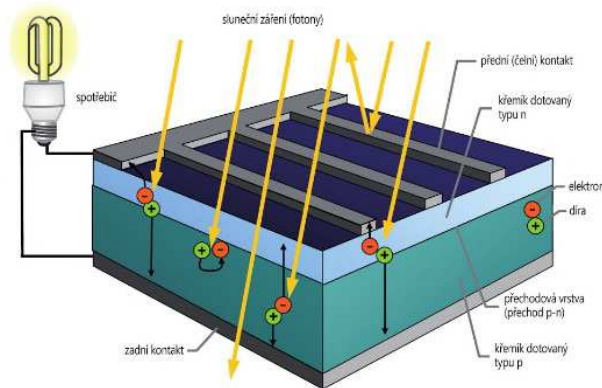
Pokud připojíme polovodič typu P, ke kladnému pólu zdroje a polovodič typu N k zápornému pólu zdroje, docílíme toho, že se PN přechod stane vodivým a začne jím protékat proud. V případě opačného zapojení svorek, dojde k tomu, že se nám na rozhraní PN přechodu odsají nosiče náboje a proud již nebude procházet. Jedná se o zapojení v závěrném směru.[3]

1.1.2.3 Fotočlánek

Zařízení sloužící k přímé přeměně slunečního záření na elektrickou energii. Strukturu si můžeme představit zjednodušeně jako polovodičovou diodu. V podstatě se jedná o PN přechod, který generuje elektrickou energii na základě dopadajícího slunečního záření.[4]

Princip:

Dopadající sluneční záření na atom křemíku, uvolní elektron. Následuje vznik páru elektron díra. V čistém polovodiči by takový pár rekombinoval, a tudíž by zanikl, ale pokud k tomu dojde v blízkosti PN přechodu, je díra vtažena elektrickým polem do polovodiče typu P a elektron do polovodiče typu N. V rekombinaci jim brání prahové napětí, které je přibližně 0,7 V. Pokud uzavřeme elektrický obvod mezi polovodičem typu P a typu N, začne jím protékat proud a fotočlánek pracuje jako zdroj elektrické energie.[4]



Obr. 3: Princip fotočlánku [1]

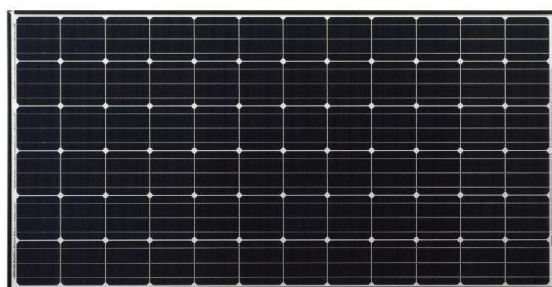
Z důvodu malého napětí dodávaného fotočlánkem, je samotný fotočlánek nepoužitelný. Proto nejčastěji využíváme sérioparalelního spojování fotočlánků. Spojováním fotočlánků do celků, jsou realizovány FV panely o požadovaných vlastnostech (výkonu, napětí, proudu).[4]

1.1.2.4 Fotovoltaický panel

Je zařízení, které se skládá z několika desítek až stovek fotočlánků, které jsou sérioparalelně propojeny, aby nám poskytly požadované vlastnosti (napětí, proud, výkon). Vzhledem ke křehkosti jednotlivých FV článků, musí být panely vyrobeny z kvalitních a odolných materiálů. Dle druhu instalovaných FV článků rozeznáváme následující druhy FV panelů.

1.1.2.4.1 Monokrystalické panely

Monokrystalické solární panely jsou vyrobeny z jednoho monokrystalu, proto název monokrystalický. Vyrobí se jeden monokrystal, který se následně rozřeže na jednotlivé FV články o rozměrech obvykle (156x156mm). Vzhledem k jednodlosti a čistotě struktury lze dosáhnout teoretické účinnosti až 24%, reálně je to 14% až 20%. Tyto panely se vyznačují typicky tmavou barvou nejčastěji hnědou až černou.[5],[6]



Obr. 4: Monokrystalický panel [5]

1.1.2.4.2 Polykrystalické panely

Polykrystalické solární panely jsou vyráběny tak, že se nechá na ploše vykristalizovat větší množství krystalů, které se následně slisují do jednoho kompaktního krystalu. Krystal je opět rozřezán na FV články o rozměrech obvykle (156x156mm). Typická barva pro tyto panely je světlejší či tmavší modrá barva. U panelů si lze všimnout nerovnoměrnosti povrchu a rozdílů jejich odstínu. Panely nejsou již tak technologicky čisté jako monokrystalické, což souvisí s jejich menší účinností.[6],[7]



Obr. 5: Polykrystalický panel [8]

1.1.2.4.3 Amorfní panely

Amorfní solární panely jsou vyráběny napařováním jednotlivých vrstev krystalů na skleněnou tabuli za vysokých teplot a absolutního vakua. Typická barva pro tyto panely je černá. Účinnost těchto panelů je poloviční oproti krystalicky vyráběným panelům. Pokud bychom chtěli stejnou výtěžnost jako u krystalických panelů, pak bychom museli jejich plochu několikanásobně zvětšit, přibližně (2x – 3x).[9],[10]



Obr. 6: Amorfní panel [9]

1.1.2.5 Měniče

Jedná se o zařízení, pomocí kterého, jsme schopni elektrickou energii vstupních parametrů přeměnit na elektrickou energii výstupních (požadovaných) parametrů. Běžně se označují jako inventory či střídače. Jsou nezbytnou součástí FV systémů. Vstupem do měniče je stejnosměrný proud (DC), vyrobený pomocí FV panelů a výstupem měniče je střídavé proud (AC). V dnešní době jsou měniče vyráběny z velice kvalitních výkonových součástek. Jedná se o velice účinné zařízení, čemuž odpovídají i jeho velmi malé ztráty. Podstatný vliv na jeho účinnost má použitý algoritmus řízení.[14]

Měniče menších výkonů jsou jednofázové, větších výkonů pak trojfázové. Výstupní střídavé napětí nemá tvar čisté sinusoidy, proto některé spotřebiče jako například výbojky s ním mohou mít problémy, nicméně tyto problémy jsou již vyladěny. Pro měnič je nutné počítat s výkonovou rezervou a to minimálně 20% z důvodu přetěžování měniče.[14],[33]

1.1.2.5.1 Měniče pracující na analogovém principu

Jedná se o starší typ měniče, kde je použit dvojčinný střídač a klasický transformátor se železným jádrem. Dle způsobu buzení hlavních výkonových prvků (unipolární tranzistory, bipolární tranzistory), dostaneme různé průběhy střídavého napětí a to buď obdélníkové, lichoběžníkové či sinusové. Čím více se u tohoto druhu měniče výstupní napětí podobá sinusovému průběhu, tím více klesá jeho účinnost a rostou ztráty na jeho spínacích prvcích. Tyto měniče jsou konstrukčně velice jednoduché, spolehlivé a nezpůsobují rušení. Jejich výstup je galvanicky oddělen pomocí transformátoru. Lze je využít pro menší výkony a všude tam, kde není žádoucí rušení samotným výkonovým obvodem.[33]

1.1.2.5.2 Měniče pracující na principu spínaného zdroje s tzv. modifikovanou sinusovkou

Jsou modernější. Vstupní stejnosměrný proud o nízkém napětí je převeden na proud o vysokém napětí a následně umělou komutací převeden na proud střídavý. Využije se jedné či dvou spínacích součástek, ovládaných mikroprocesorovým řídicím obvodem, který mění dobu sepnutí prvku (nejčastěji unipolární tranzistor - MOSFET), tím dosáhneme vytvoření schodovité sinusoidy. K řízení je nejčastěji použita pulsně šířkové modulace (PWM). Napětí je zvyšováno pomocí feritových transformátorů, které jsou velice kompaktní díky jejich vysokým frekvencím. Výkonový obvod, je tvořen, čtyřmi tranzistory zapojenými do můstku.

Z důvodu buzení je obvod provázán s primární i sekundární stranou, což je slabinou těchto měničů. V případě propojení vstupu s výstupem následuje poškození měniče. Další nevýhodou je okolní rušení elektrických přístrojů v případě špatného navržení obvodu měniče. Nicméně i přes značné nevýhody je tento měnič velice výhodný a to zejména svou účinností, hmotností a nízkou vlastní spotřebou. Lze docílit konstantního výstupního napětí a frekvence nezávisle na velikosti zátěže nebo napětí baterie.[33]

1.1.3 Solární systém

Skladba solárního systému je individuální záležitostí, liší se v závislosti na požadavcích uživatele. Nicméně nedílnou součástí každého FV systému jsou FV panely, které při styku se slunečním zářením budou dodávat spotřebičům stejnosměrný proud. Pro následné využití generované energie je zapotřebí dalších prvků například měničů, sledovačů slunce, regulátorů, akumulátorů a dalších součástí.[11]

1.1.3.1 Aktivní solární systém

Pod pojmem aktivní solární systém si představme systém, který využívá přeměnu solární energie na jinou formu energie. V praxi se jedná o solární panely, které slouží k přímému ohřevu vody, nebo FV panely, které přeměňují sluneční záření na elektrickou energii. Hlavní výhodou těchto systémů je možnost modifikace a snadné aplikace na jakémkoliv místě či stavbu. Další nespornou výhodou je možnost akumulace energie. Elektrickou energii můžeme uschovat v akumulátorech, tepelnou energii lze uschovat v podobě teplé vody v akumulacích tepelných nádrží.[12],[13]

1.1.3.2 Pasivní solární systém

Pod pojmem pasivní solární systém, si představme systém, který nekoná žádnou přeměnu energií a jen se pasivně podílí nejčastěji na vzniku tepla. Dnes jsou to běžně užívané systémy zahradních skleníků, zimních zahrad a jakákoliv prosklených součástí staveb. Všechny výše uvedené prvky jsou využívány k zadržování tepelné energie z dopadajícího slunečního záření. V případě, že chceme tento systém použít, musíme počítat s jeho návrhem, již před zahájením stavby.[12],[13]



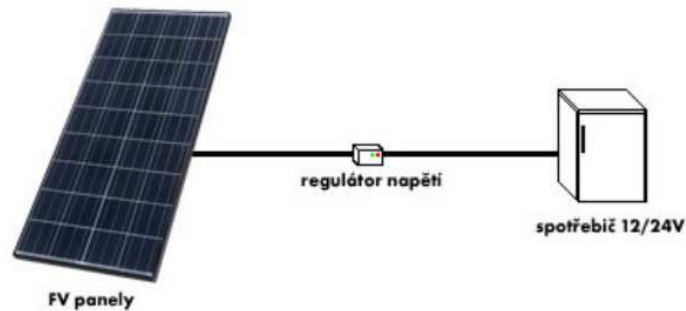
Obr. 7: Rozdělení solárních systémů na aktivní a pasivní [12]

1.1.3.3 Grid-off systémy

Tyto systémy jsou zcela autonomní systémy, častěji se můžeme setkat s názvem ostrovní systémy. Jak již z názvu vyplývá, jsou tyto systémy zcela nezávislé na elektrické napájecí soustavě. Nejčastěji se s instalací těchto systémů, setkáme na místech bez dostupné elektrické sítě, popřípadě tam, kde by vybudování elektrického sítě stálo více, než samotný grid-off systém. Vzhledem ke zmiňovaným vlastnostem, jsou tyto systémy instalovány například u horských chat, samot, telekomunikačních zařízeních, výzkumných stanic, parkovacích automatů či kamerových systémů. V rámci grid-off systémů rozlišujeme následující druhy.[11]

1.1.3.3.1 Grid-off systém s přímým napájením

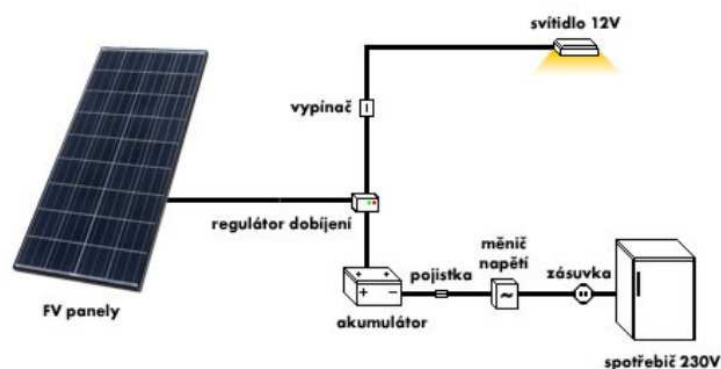
Tento systém je schopný generovat elektrickou energii pouze při dostatečném slunečním záření. Proto se tyto systémy instalují všude tam, kde není elektrická energie vyžadována nepřetržitě. Možnost aplikace najdeme například u napájení čerpadla okrasné fontánky či nabíjecího zařízení. Jedná se o propojení FV panelu, regulátoru a spotřebiče (zátěže). [11]



Obr. 8: Grid-off systém s přímým napájením [14]

1.1.3.3.2 Grid-off systém s akumulací elektrické energie

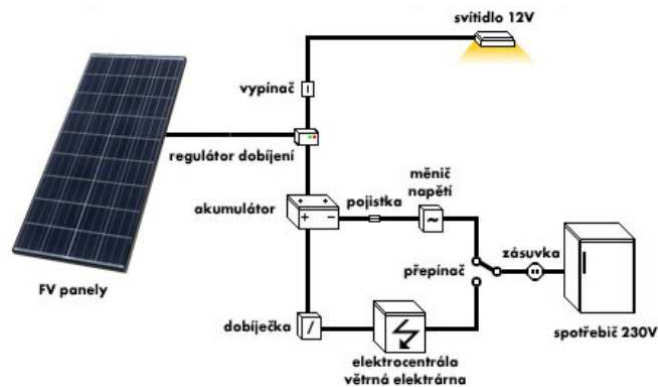
Jedná se o systém s přímým napájením doplněný o akumulační prvek (akumulátorovou baterii). Akumulátor zajišťuje možnost napájení v případě slabého či žádného slunečního záření. Tyto systémy je možno instalovat na místa, kde je potřeba napájení i v případě nedostatečného slunečního záření. Životnost systému je limitována životností akumulátoru, která souvisí se způsobem nabíjení a vybíjení. Proto je systém osazen regulátorem nabíjení. V případě připojení měniče pak lze k tomuto systému připojit i spotřebiče na střídavý proud.[11]



Obr. 9: Grid-off systém s akumulací elektrické energie [14]

1.1.3.3 Grid-off systém - hybridní

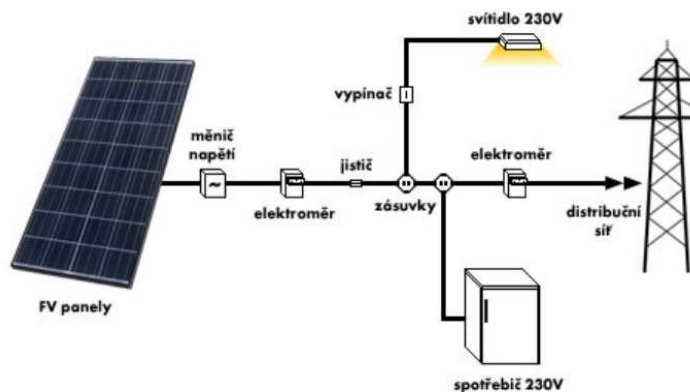
Jedná se systém, který je navíc doplněn o náhradní zdroj energie. Náhradní zdroj energie lze využít k nabíjení akumulátorů v případě nedostatečného slunečního záření popřípadě jako zdroj primární. Pro dostatečný výkon systému je nezbytné jeho správné navržení. Návrh souvisí s množstvím energie určené ke spotřebě, proto návrh systému volíme tak, aby nebyl předdimenzovaný ani poddimenzovaný. Jako náhradní zdroj se využívá diesel agregát, větrná či vodní elektrárna.[11]



Obr. 10: Hybridní grid off systém [14]

1.1.3.4 Grid-on systém

Tento druh systému se nasazuje v husté zástavbě, kvůli možnému přebytku či nedostatku elektrické energie. Systém pracuje zcela autonomně a nezávisle na elektrické síti. V případě přebytků vyrobené elektrické energie, jsou přebytky dodány do elektrické sítě, kde jsou patřičně využity. V případě nedostatku elektrické energie z FV systému, lze nedostatek energie pokrýt z elektrické sítě. Aby měl distributor a zákazník přehled o tocích energií, je systém vybaven dvěma elektroměry umístěnými na FV straně a v místě připojení elektrické sítě.[11]

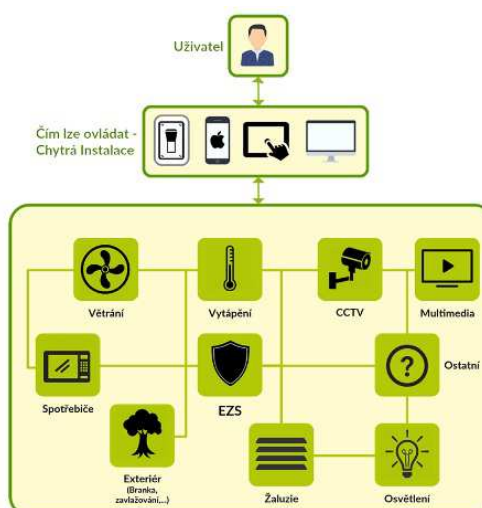


Obr. 11: Grid-on systém [14]

1.2 Inteligentní řídicí systém

1.2.1 Co je to inteligentní systém

Co se nám pod pojmem inteligentní systém vybaví? Představme si, že se jedná o řídicí jednotku a periferie k ní připojené. Řídicí jednotka nám rozhoduje na základě našich požadavků, co se má kde v jaké periférii zapnout. V podstatě vy, jako obyčejný uživatel spatříte pouze přístupové ovládací jednotky, instalované na stěně jako klasické vypínače. Může se jednat o tlačítkové ovladače či dotekové panely. V dnešní době jsou velice rozšířené různé aplikace pro chytré telefony, tablety, televizory a osobní počítače.



Obr. 12: Co lze inteligentním systémem ovládat [15]

Inteligentní systém není pouze ovládání osvětlení, vytápění a žaluzií. Jedná se o propracovaný systém, který v sobě zahrnuje mnoho funkcí. Od základních ovládacích funkcí osvětlení, vytápění, ovládání žaluzií se dostaneme k zabezpečovacímu systému (EZS), systému ventilace domu, kamerovému systému, protipožárnímu systému (EPS), problém není ani AV-technika (televizory, receivery, zesilovače), domácí telefon, a mnoho dalších funkcí, které jsou individuální v závislosti na spotřebiteli. Zjednodušená představa inteligentního systému je na obrázku č. 12.

1.2.2 Možnost implementace

Jak již bylo zmíněno, implementace inteligentního systému je velice rozmanitá. V případě, že chceme, aby inteligentní systém byl součástí naší domácnosti či podniku. Je nutné s jeho návrhem počítat již při samotné realizaci projektové dokumentace, tato možnost nám ušetří mnoho času, peněz a starostí oproti variantě, že by se inteligentní systém dodával až po realizaci stavby.

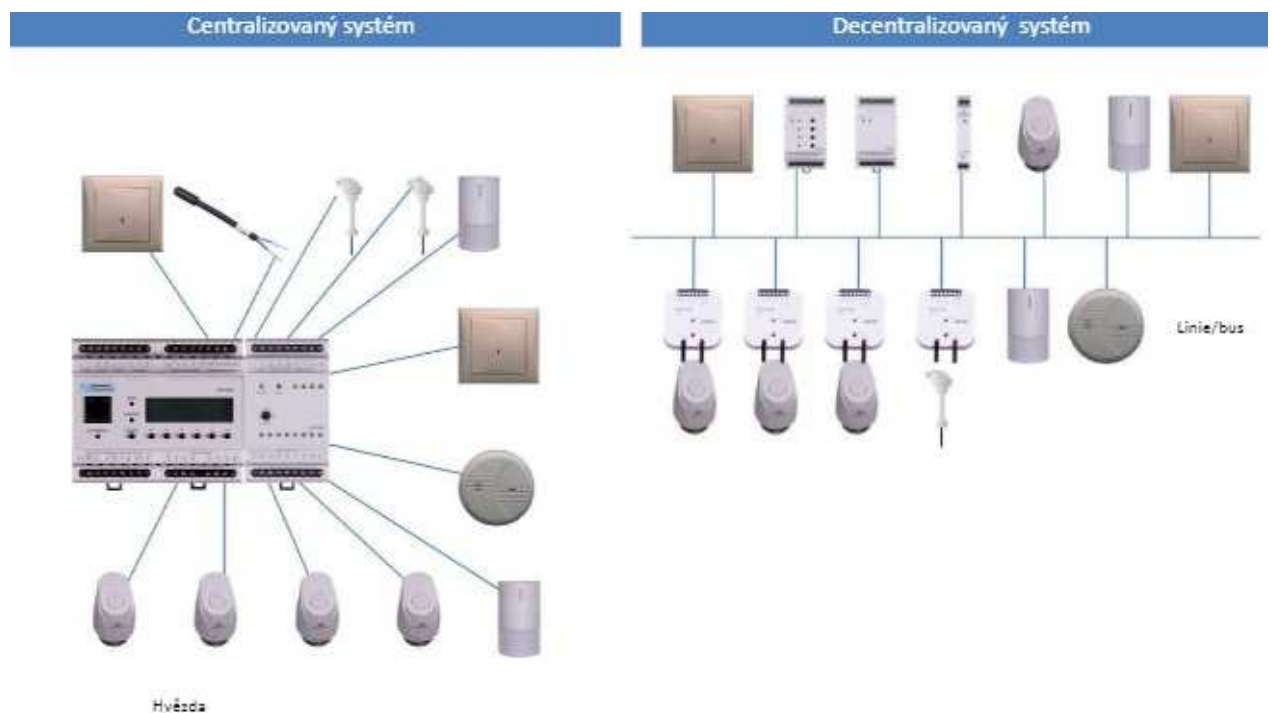
Možnosti využití:

- Ovládání a regulace osvětlení, v dnešní době není problém s jakýmkoliv druhem svítidla (LED, UV, IR, halogenové svítidla, výbojky)
- Ovládání žaluzií se zpětnou vazbou o poloze. Žaluzie je pak možno ovládat manuálně a v rámci propojení se nám zobrazí v softwaru údaj o změně polohy případně naopak
- Ovládání vytápění domu, synergie mezi klimatizačními jednotkami, centrálním vytápěním, tepelným čerpadlem a ventilačními systémy není překážkou
- Centrální zabezpečení domu (EZS), dle požadavku zákazníka. Jedná se o velice rozmanité instalace odlišující se co do výrobce. Systém musí daný výrobce softwarově podporovat, pomocí vhodného ovladače (driveru). V případě, že driver není k dispozici, lze jej získat od technické podpory daného inteligentního systému
- Protipožární systém (EPS) domu. Stejně tak i u této možnosti platí velice rozmanité možnosti aplikace a volby daného systému.
- Kamerové systémy je možné propojit se zabezpečovacím systémem, stejně tak i například s domovním zvonkem
- AV techniku lze těmito systémy také ovládat. Jedná se o různé kombinace zapojení zesilovačů, audio systémů, televizorů, dvd přehrávačů. Lze zrealizovat mnoho možností propojení opět záleží na tom, co požaduje uživatel.

- Další možnosti lze zrealizovat na základě přání budoucího uživatele. Například profily pro ovládání domu při jeho opuštění, plánovaném příjezdu. A mnoho dalších možností od ovládání po ovládání lampičky u nočního stolku až po ovládání venkovní brány.

1.2.3 Typy inteligentních systémů na základě řízení

Rozlišujeme dva druhy systémů. Jedná se buď o centralizovaný systém řízení, nebo decentralizovaný systém řízení. Pod pojmem centralizovaný systém si můžeme představit, že systém je řízený jednou řídicí jednotkou v rámci celého systému. U decentralizovaného systému se jedná o systém, který obsahuje v každé koncové jednotce řídicí jednotku. Komunikace v obou případech probíhá po sběrnici, která je omezena vzdáleností a počtem připojených jednotek. Někdy se setkáme s hybridním systémem řízení, což je kombinace centralizovaného a decentralizovaného systému řízení, kdy samotné sensory jsou připojeny na sběrnici a ovládací prvky jsou připojeny k řídicí jednotce. Představa je na obrázku č. 13.



Obr. 13: Centralizovaný, decentralizovaný systém [20]

1.2.4 Stručný přehled výrobců

1.2.4.1 Vantage control

Jedná se o inteligentní centralizovaný systém amerického výrobce, skládajícího se z řídicí procesorové jednotky, koncových spínacích jednotek (reléových, tranzistorových, stmívacích) a ovládacích prvků v podobě programovatelných tlačítek, dotykových panelů, aplikací použitelných v chytrých elektronických zařízeních. Tento systém lze snadno rozšířit o audio systém, kamerový systém, zabezpečovací systém (EZS) dle požadavku zákazníka. Pro řízení systémů různých výrobců je nutné využít převodníků, které umožní vzájemnou komunikaci.



Obr. 14: Procesorová jednotka [16]



Obr. 15: Koncové spínací jednotky pro ovládání žaluzií, světel popřípadě dalších spotřebičů
[16]



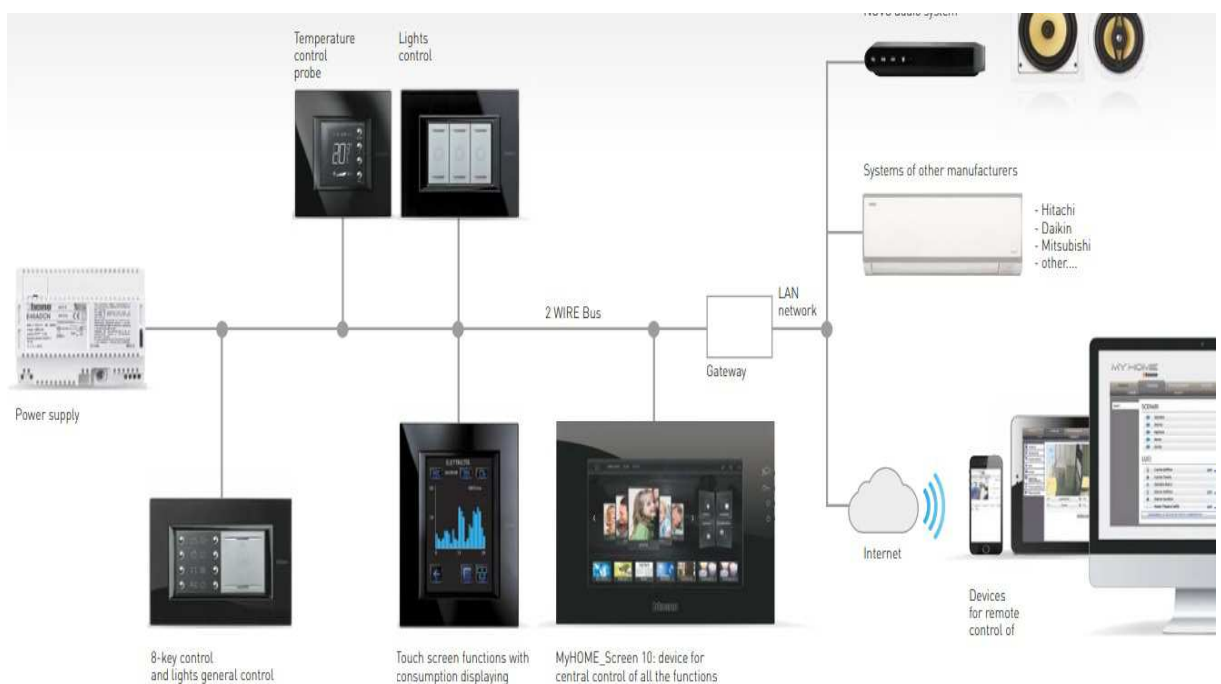
Obr. 16: Ovládací prvky [17], [18]



Obr. 17: Převodník propojující systém KNX se systémem Vantage [19]

1.2.4.2 Bticino

Jedná se o inteligentní systém italského výrobce. Řízení systému je zabezpečeno přes sběrnici, ke které jsou připojeny jednotlivé periferie. Jedná se tedy o decentralizovaný systém. Nastavení tohoto systému je velice jednoduché. Každý prvek v systému má svoji jedinečnou adresu, kterou lze snadno změnit pomocí dodávaných odporů v balení. Po nastavení jednotlivých adres, je nutné jednotlivé prvky zadat do systému, ve kterém lze s nimi dále pracovat. Systém se skládá z ovládacích prvků (tlačítek, dotekových panelů), reléových jednotek pro ovládání světel, žaluzií a dalších jednotek pro vytápění, AV-techniku.



Obr. 18: Přehled prvků a zapojení systému Bticino [21]

1.2.4.3 KNX

Jedná o velice snadno rozšiřitelný systém, který je možné kombinovat s mnoha prvky jiných výrobců podporujících komunikační protokol KNX. Systém jako takový pracuje decentralizovaně. Každá jednotka vykonává předem definovanou činnost. Všechny jednotky komunikují po sběrnici a každá jednotka vykonává činnost nezávisle jiné jednotce. Jak jsem již zmínil, systém je velice modulární a snadno rozšiřitelný i o prvky výrobců jako je ABB, Schneider, Legrand, Siemens. Na obrázku č. 19, můžeme vidět příklad decentralizovaných jednotek.



Obr. 19: Příklad ovládacích modulů KNX [22]

1.2.5 Výhody a nevýhody inteligentních systémů

Výhody

- +Zvýšení komfortu objektu
- +Úspora energií
- +Modifikace systému
- +Vzdálené řízení

Nevýhody

- Vysoké pořizovací náklady
- Složitost systémů
- Nutná podpora ze strany výrobce
- Stárnutí systému z pohledu neustálého vývoje
- Možná poruchovost
- Možnost snadného zneužití v případě prolomení ochrany systému

2 Možnosti předpovědi slunečního záření včetně profilu spotřeby

2.1 Předpověď

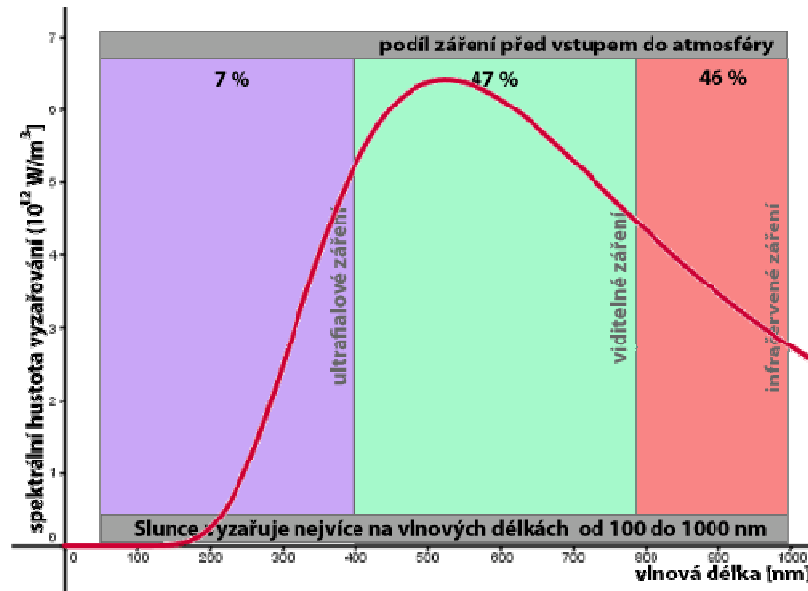
Neboli prognóza nám vyjadřuje budoucí stav atmosféry nad sledovaným územím. K tomu slouží údaje o jejím stavu v současnosti a minulosti, které poskytuje synoptická meteorologie. Předpověď sestavuje meteorolog prognostik. Zpracovává se buď předpověď synoptická, numerická (početní) nebo statistická.[25]

Synoptická předpověď vzniká na základě pravidelných měření meteorologických prvků (tlaku, vzduchu, teploty apod.) nad větším územím meteorolog určí, čím je počasí dané oblasti ovlivněno (fronty, tlakové útvary apod.)[25]

Numerická meteorologie využívá soustavy diferenciálních rovnic, jejichž řešení je silně závislé na volbě počátečních podmínek. Znamená to, že když zadáme do modelu vycházejícího z uvedených rovnic jen mírně pozměněná vstupní data, můžeme dostat zcela odlišné předpovědi. Proto John von Neumann rozdělil procesy v atmosféře do tří kategorií. Do první patří jevy, jejichž vývoj je závislý na počátečních podmínkách, a lze je tedy předpovídat jen na velmi krátkou dobu dopředu. Opakem jsou pak procesy, které jsou na počátečních podmínkách nezávislé a při jejichž předpovědi je vhodné se soustředit na zvláštnosti všeobecné cirkulace atmosféry. Mezi těmito extrémními případy leží skupina procesů, u kterých nelze dost dobře předpovědět, v co se počáteční podmínky vyvinou, jsme-li od nich dostatečně vzdáleni. V souvislosti je předpověď rozčleněna na krátkodobou, střednědobou a dlouhodobou.[25]

2.1.1 Sluneční záření

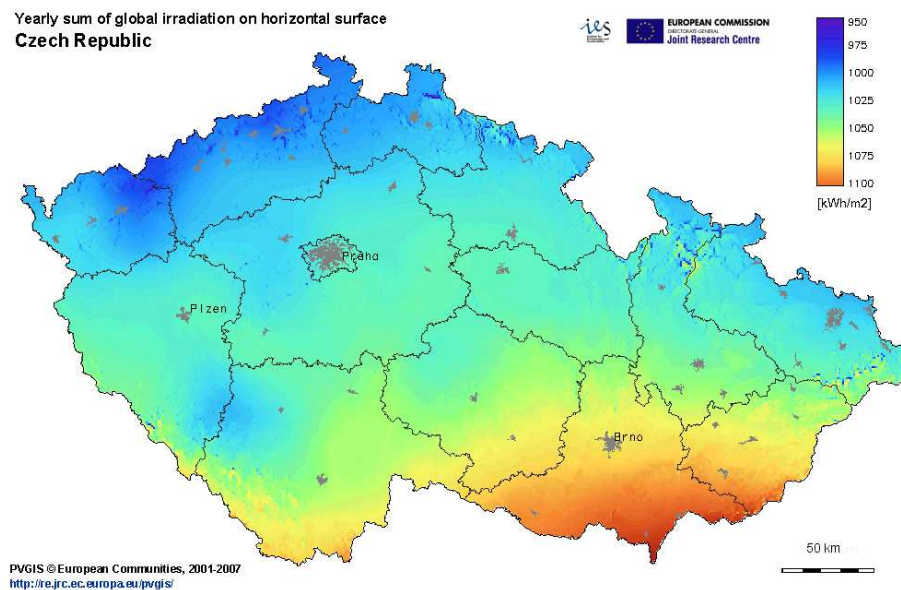
Sluneční záření je elektromagnetické záření o charakteristickém spektru vlnových délek, které před vstupem do zemské atmosféry přibližně odpovídá spektru dokonale černého tělesa o povrchové teplotě 5800 K. Hlavním zdrojem slunečního záření je Slunce. Sluneční spektrum dělíme do tří základních oblastí na: ultrafialové, viditelné a infračervené sluneční záření. [25]



Obr. 20: Sluneční záření před vstupem do atmosféry

Množství slunečního záření dopadající na atmosférický obal Země je v průměru 1365 W/m^2 a nazývá se solární (sluneční) konstanta. Průchodem atmosférou země se výkon slunečního záření zmenšuje z důvodů odražení, pohlcování a rozptylování. Pro odhad jmenovitého výkonu panelů na pokrytí požadované spotřeby energie, můžeme použít vědomost, že z 1 kW instalovaného výkonu dostaneme 800 až 1100 kWh elektrické energie za rok.[14], [25]

Intenzita slunečního záření (záření dopadající na zemský povrch) je součtem přímého slunečního záření a difúzního (rozptýlené) slunečního záření. Sečteme-li přímé a difúzní sluneční záření za celý rok, dostáváme celkové roční ozáření v kWh na $1 \text{ m}^2/\text{rok}$. Které například v Česku dosahuje hodnot 950 až 1340 kWh/m^2 , což je vidět na obrázku č. 21.[14], [25]



Obr. 21: Přehled dopadajícího slunečního záření na povrch České republiky za rok [23]

2.1.2 Předpověď slunečního záření

Nejčastěji využívaná předpověď počasí včetně slunečního záření se rodí v meteorologických stanicích. Tyto stanice se zabývají sběrem dat na základě, kterých predikují vývoj počasí. Dle kritérií můžeme meteorologické stanice rozdělit podle zaměření na synaptické, klimatologické, letecké, zemědělsko-meteorologické, speciální, podle charakteru získaných dat na přízemní, v mezní vrstvě atmosféry, podle umístění na pozemní, námořní, letecké (na letadlech), družicích, podle pozorovatelů na profesionální, amatérské, podle stupně automatizace na automatizované či manuální. [25]

Správa a údržba těchto stanic je v kompetenci Českého hydrometeorologického ústavu, který jim dodává měřicí přístroje a stará se o sběr dat, které stanice produkují a následně je vyhodnocuje. Pokud se na meteorologickou síť zaměříme celosvětově je organizování v kompetenci organizace World Meteorological Organisation (WMO) se sídlem v Ženevě.[25]

2.1.3 Měření slunečního záření na povrchu země

K měření slunečního záření používáme přístroj nazývaný pyranometr, který měří sluneční záření dopadající na vodorovnou plochu. Jedná se buď o záření globální, nebo difuzní. Pyranometry měřící difuzní záření poznáme jednoduše, jsou vybaveny stínícím prstencem nebo černou koulí s automatickým pohonem vybaveným sledovačem slunce (trackerem). Tím se docílí odstínění přímého slunečního záření. [26]



Obr. 22 Druhy pyranometrů [26]

Dalším speciálním přístrojem pro měření slunečního záření je Albedometr, ten je složen ze dvou pyranometrů, z nichž jeden je natočen k zemi a druhý k obloze. Tento měřicí přístroj nám umožňuje zjistit velikost odraženého záření zpět do atmosféry. Pro velmi specializovaná měření je využíván přístroj pyrhelimetr, který zachycuje pouze složky přímého slunečního záření. [26]



Obr. 23 Albenometr a Pyrhelimetr [26]

2.1.4 Predikce slunečního záření na základě známých modelů

Pro predikci slunečního záření rozeznáváme dva základní přístupy. První využívá stochastických generátorů a druhý využívá rovnic o empirických vztazích. Další možností predikce slunečního záření je, využití lineární interpolace či geostacionárních družic. Pomocí stochastických dat jsme schopni, nasimulovat situace, které by mohli teoreticky vzniknout. Avšak nelze těmito daty nahradit reálně měřená data. [29]

Rovnice o empirických vztazích nám poskytují data o slunečním záření na základě výpočtu z jiných již snadněji měřených veličin. Pro tyto odhady bylo vyvinuto nesčetně metod, každá z metod využívá odlišný mechanismus výpočtu a jiná vstupní data. Vstupními parametry jsou často sluneční svit, rozdíl mezi maximální a minimální teplotou, průměrnou teplotu, teplotu půdy, relativní vlhkost vzduchu, počet srážkových dnů, nadmořskou výšku, zeměpisné souřadnice, celkový úhrn srážek, hodnotu albeda, oblačnost či celkovou evaporaci. [29]

2.1.5 Měřicí metody

2.1.5.1 Angstromova - Prescottova

Tato metoda je modifikací původní Angstromovy práce. Je zaměřena na dobu trvání slunečního svitu, jako na nejvhodnější meteorologickou charakteristiku pro účely odhadu slunečního záření. Výpočtový vztah staví do souvislosti hodnotu trvání relativního slunečního svitu a relativního množství dopadajícího slunečního záření. Pro tuto metodu jsou potřebná měřená data, a proto je nepoužitelná u lokalit bez měřených dat. [30]

2.1.5.2 Klabzubova

Metoda vychází z doby trvání slunečního svitu. Avšak výpočet nemá fyzikální základ. Metoda byla odvozena jako regresní matematický vztah mezi relativní dobou slunečního svitu pro daný den v roce a příslušnou sumou slunečního záření. Metoda se omezuje pouze na data z jedné stanice, a tudíž tuto metodu nelze použít u zcela odlišných klimatických podmínek. U této metody jsme omezeni ročním obdobím, v zimních měsících tato metoda generuje velkou chybu, proto je vhodné ji využít v jarních či letních měsících. Metodu lze uplatnit na území střední Evropy. [30]

2.1.5.3 Hargreavesova

Je použitelná v místech, kde máme pouze záznamy denních teplotních extrémů. Metoda

vychází z práce Hargrave. Primárním cílem této práce bylo vyvinout metodiku závlahového systému pro danou část Afriky a jako nejjednodušší pro odhad slunečního záření bylo stanoveno využití vztahů z denních teplotních extrémů. Přestože metoda byla původně vyvinuta pro subtropickou oblast, je pro její jednoduchost v praxi často používána. Na území české republiky jsou denní extrémy teplot relativně stálé, a proto je lze považovat za konstanty. [30]

2.1.5.4 Supitova

Vznikla z hodnocení vlivu počasí na produkci zemědělských plodin. Výpočet metody operuje se čtvercem 50x50km nad územím Evropy. Sluneční záření je jedním vstupem do výpočtu. Vzhledem k obtížnosti získat potřebná data, vznikla zcela nová metoda, která jako vstupy vyžaduje hodnoty denních teplotních extrémů a hodnotu celkové oblačnosti. Rovnice pro výpočet je kombinací rovnic Wornera a Hargave. Pokud máme k dispozici hodnoty oblačnosti je tato metoda kvalitním nástrojem k predikci slunečního záření. [30]

2.1.5.5 Winslowova

Pro tuto metodu je nutné znát tři místně specifické koeficienty, které se určují pouze dlouhodobými měřeními v dané lokalitě, proto lze tuto metodu s obtížemi běžně použít. Základem pro výpočet metody je empiricky obrácený vztah mezi transmitancí a relativní vlhkostí vzduchu. Výsledná hodnota slunečního záření, je pak úměrná poklesu relativní vlhkosti vlivem nárůstu teploty z denního minima na maximum. Vzhledem k lehkému získání měřených hodnot je tato metoda univerzálně použitelná pro všechny zeměpisné šířky. [30]

2.1.5.6 Interpolační metoda

Interpolace je v podstatě odhadem neznámé, náhodné, prostorové hodnoty v určitém bodě, na určitém místě, s ohledem na okolní hodnoty. Pro popis je nutná kvantifikace. Prvotně je nutné stanovit prostor pro výpočet a následně určit jednotlivé složky zvoleného prostoru. Máme na výběr mezi rozptylem či kovariací. Vzhledem ke složitosti je využito většinou rozptylu, u této volby se veličina popisující prostorovou závislost sledovaného procesu nazývá variogram. Variogram popisuje, jakým způsobem se mění rozptyl hodnoty rozdílu mezi dvěma body v závislosti na vzdálenosti těchto bodů. [30]

2.1.6 Měřicí stanice slunečního záření v ČR

Měření slunečního záření provádí meteorologické stanice jen zřídka, z důvodu cenové náročnosti vybavení a následné kalibrace a údržby. Stanice, které měření slunečního záření provádí, neposkytují data nepřetržitě z důvodu problémů s měřicími zařízeními. V následujících odstavcích jsou uvedeny příklady stanic, provádějící měření.

2.1.6.1 Univerzitní centrum ČVUT (UCEEB)

Stanice je provozována univerzitou ČVUT, spíše jejím centrem zabývající se efektivitou energií budov. Stanice leží na okraji města Buštěhrad a skládá se ze tří meteorologických stanic. Stanice měří krátkovlnné a dlouhovlnné sluneční záření, teplotu vzduchu, půdy, rychlost a směr větru, tlak vzduchu, srážky a relativní vlhkost. Měřená data jsou vyhodnocována v minutových intervalech. Za den není problém nasbírat okolo 1400 vzorků dat. Sluneční záření je získáváno průměrováním získaných denních dat. Průměrování se provádí, z důvodů výskytu špičkových hodnot. [27]

2.1.6.2 Meteorologické stanice FIEDLER

O provoz stanic se stará firma FIEDLER AMS s.r.o.. Vlastní dvě stanice zabývající se sběrem dat o krátkovlnném slunečním záření. Jedna z nich je u Třeboně, druhá v Brně. Kromě krátkodobého slunečního záření měří vlhkost půdy a vzduchu, rychlost a směr větru, srážky, teplotu půdy a vzduchu. Intenzita slunečního záření je vyhodnocována v desetiminutových intervalech, výsledné hodnoty jsou zpracovány průměrováním, aby se eliminovala chyba měření. [27]

2.1.7 Možné zdroje předpovědi

Jako zdroj předpovědi krátkodobého slunečního záření, můžeme využít data Národního úřadu pro oceán a atmosféru (National Oceanic and Atmospheric Administration – NOAA) spadající pod Ministerstvo obchodu Spojených států amerických. Organizace zkoumá meteorologii atmosféry a oceánů. Data jsou získávány z následujících modelů předpovědi. [27]

2.1.7.1 GEFS Reforecast v2 Server

Jedná se o model spadající pod Národní úřad pro oceán a atmosféru (NOAA). Umožňuje předpověď slunečního záření s predikcí 72hodin. Data jsou získávány v rámci tříhodinových

intervalů, což je 8 hodnot denně. Předpověď je dodávána ve formě mřížky s rozlišením jeden úhlový stupeň v zeměpisné šířce a délce. Model předpovídá dalších 111 veličin a dalších 164 měří.[27]

2.1.7.2 ERDDAP Server

ERDDAP Server spadá pod Národní úřad pro oceán a atmosféru (NOAA) a poskytuje data s predikcí 48hodin v šestihodinových intervalech. Předpověď, je dodávána ve formě mřížky s rozlišením 0,25, 0,5, nebo 1 úhlový stupeň v zeměpisné šířce a délce. Model je schopný poskytovat dalších 164 měřených veličin.[27]

2.1.7.3 Solar Radiation Database

Jde o francouzský projekt zabývající se zkoumáním klimatických veličin se specializací na veličiny spojené se slunečním zářením. Kromě slunečního záření lze, získat data o relativní i absolutní vlhkosti, teplotě vzduchu. Model předpovídá hodnoty slunečního záření až 72h napřed ve tříhodinových intervalech. Předpověď je opět poskytována ve formě mřížky s rozlišením jeden úhlový stupeň v zeměpisné šířce a délce.[27]

2.1.7.4 ENERGOCENTRUM PLUS, s.r.o

Společnost poskytuje předpověď počasí pro komerční účely. Kromě krátkodobého slunečního záření, poskytuje data o rychlosti větru, teplotě vzduchu, oblačnosti, srážkách, relativní vlhkosti, dlouhodobém slunečním záření. Společnost dodává data s předpovědí 72 hodin s jednohodinovým intervalem aktualizace dat. [27]

2.1.7.5 Projekt MEDARD

Projekt pracuje s numerologickým modelem pro předpověď počasí v rámci dvou domén odečtu dat, první doména zahrnuje většinu Evropy mimo Severských zemí a Islandu. Druhá doména pokrývá Českou Republiku. Předpověď je poskytována 72h napřed s aktualizací po šesti hodinách. Slunečního záření je predikováno 79 hodin napřed s hodinovou aktualizací dat. Projekt Medard spravuje Ústav informatiky Akademie věd České republiky. [27][28]

2.1.7.6 Pacific Islands Ocean Observing System

Projekt vznikl vývojem z organizace (Hawaii Ocean Observing System) na Havaii. Poskytuje předpověď 120 napřed s tříhodinovou aktualizací dat. Předpověď je aktualizována ve formě mřížky s rozlišením 0,5 úhlového stupně v zeměpisné šířce a délce. [27]

2.2 Profil spotřeby budovy

Jde o profil, poskytující přehled o množství spotřebované elektřiny v daném objektu, nejčastěji je v jednotkách kilowatech (kW). Profil spotřeby je nezbytnou součástí návrhu FV systému. Bez něj bychom pouze slepě odhadovali budoucí výkon FV systému, čímž by mohlo dojít k tomu, že navržený systém bude poddimenzovaný nebo předimenzovaný. Profil spotřeby se tvoří na základě odhadu nebo celkové roční spotřeby elektrické energie, popřípadě měření či poskytnutím dat ze strany distributora pro případ, že máme chytrý elektroměr.

2.2.1 Jak lze získat profil spotřeby

2.2.1.1 Odhadem bez prvotních dat

Tato možnost bere v úvahu všechny elektrické spotřebiče, které jsou v domácnosti, provede se součet jejich odběrů a výsledek, který nám vyjde, vynásobíme koeficientem, podobně jak tomu je při návrhu elektroinstalace. Z výsledku, který nám vyjde, vezmeme hodnotu odpovídající maximálně 30% zjištěného výkonu.

2.2.1.2 Na základě celkové roční spotřeby

Celkovou roční spotřebu zjistíme z ročního či meziročního vyúčtování od distributora elektrické energie. Určení profilu spotřeby je pak otázkou snadného výpočtu a následného odhadu. Vezme se celková spotřeba elektrické energie za rok a vydělí se počtem dní v roce. Výsledek je přibližná hodnota spotřeby elektrické energie za jeden den. Denní hodnota se pak vydělí počtem hodin za den a vyjde nám hodnota spotřeby za hodinu. Následně je na nás odhadnout a stanovit profil spotřeby v daném objektu pro jednotlivé dny.

2.2.1.3 Měření profilu spotřeby:

Spíše se jedná o odečet nežli o samotné měření. Podstata spočívá v odečtu spotřebované elektrické energie na elektroměru a následném zápisu do vytvořené tabulky. Tabulka se skládá z časových a odečítaných hodnot. Pro profil spotřeby bude dostatečný odečet každou hodinu, pro případ vyšší přesnosti, si stanovíme kratší cykly odečtu například na patnáct minut. Ve výsledku tedy dostaneme za den 24 hodnot, popřípadě více. Odečet popsany tímto mechanismem provedeme pro jeden pracovní den v týdnu a jeden nepracovní den. Na základě toho si stanovíme spotřebu pro týden, měsíc, rok.

2.2.1.4 Poskytnutí profilu spotřeby distributorem elektrické energie

V případě, instalovaného chytrého elektroměru je získání profilu spotřeby snazší. Stačí pouze kontaktovat distributora elektrické energie s požadavkem na profil spotřeby. Chytrý elektroměr umí odečítat data spotřeby čteněji, než jednou za rok a to klidně každou hodinu či každou čtvrt hodinu. Důvod proč se tomu tak není je ten, že by došlo k rušení komunikačních tras těmito hodnotami.

2.2.2 Využití profilu spotřeby pro FV systém

2.2.2.1 Systémy a tím spojená úspora elektrické energie

V dnešní době chytrých elektroinstalací, pasivních a inteligentních domů. Je kladen vysoký důraz na celkové náklady spojené se spotřebou energií. Ať už ze strany spotřebitelů, tak ze strany podnikatelů a samotného státu, což je dáno jeho legislativou. Primárními důvody jsou neustále se zvyšující platby za energie (vodu, elektřinu, plyn). A proto mnoho lidí uvažuje o tom, jak tyto náklady co nejvíce minimalizovat.

Pro zabezpečení co největší úspory, je nutné s případným řešením přijít v počátku stavby, ale nejlépe ještě před ní. Nyní hovořím pouze o domech, chatách, bytové domy jsou nad rámec možností, ty jsou realizovány pod záštitou distributora, kde jedinec nemá možnost do stavby zasahovat. Počátečním bodem je návrh projektu rodinného domu, podle kterého se stavba zrealizuje. Otázkou zůstává, jaké další systémy bude chtít uživatel do objektu instalovat. Instalované systémy zajistí nezávislost na zavedeném systému. Například je možné si nechat navrtat studnu a k ní připojit vodovodní systém domu, čímž se sníží náklady na zdroj vody. Pokud tento systém rozšíříme o systém solárního ohřevu vody, ušetříme elektrickou energii potřebnou na ohřev TUV. Další možností je například zavedení systému tepelného čerpadla, FV systému. Uvedené možnosti jsou nespornou výhodou při jejich každodenním užívání a vedou k podstatné úspoře nákladů. Výběr systému je v kompetenci uživatele. Obrovskou nevýhodou je pořizovací cena.

2.2.2.2 Souvislost profilu spotřeby, FV systému a předpovědi počasí

V České republice nastal obrovský nárůst instalovaných FV prvků. Jedním důvodem byla, špatně nastavená dotační politika státu, za elektrickou energii dodávanou do sítě pomocí FV systémů. To mělo neblahý vliv na energetickou síť, kterou to strašně zatížilo z pohledu regulace a nutných rezerv pro regulaci. Což se promítlo do koncových cen zákazníkům. Naštěstí se od tohoto trendu ustoupilo. Přebytky z domácností nejsou, vykupovány ba naopak distributoři elektrické energie dotují spotřebitele, které své přebytky energie spotřebují místo toho, aby je dodávali zpět do distribuční sítě.

S ohledem na výše zmiňované, je výhodné využít profil spotřeby při návrhu FV systému a jeho následném řízení. Profil spotřeby využijeme pro saldo hodin, kdy přesně víme, kolik energie se spotřebovává a víme, kolik energie můžeme akumulovat či spálit v topném tělese akumulčního ohřívače či využít v jiných aplikacích. Množství energie dodávané FV systémem je přímo úměrné slunečnímu záření, a proto je vhodné, pro regulaci využívat předpověď počasí. Výslednou kombinací dosáhneme nejefektivnějšího řízení a nejlepšího využití generované energie.

3 Návrh metodiky a inteligentního řízení FV systému

3.1 Cíl projektu

Cílem projektu je návrh inteligentní řízení FV systému s akumulátory na základě předpovědi slunečního záření a profilu spotřeby. Řídicí systém je realizován ve vizualizačním softwaru CX-Supervisor, ke kterému je připojen logický automat (PLC) od výrobce OMRON. Hlavní funkcí systému je efektivní řízení nabíjení a vybíjení akumulátorů, v závislosti na předpovědi slunečního záření a aktuální spotřebě objektu. Dobu předpovědi si uživatel může libovolně měnit od 1 hodiny do 72h. Data o spotřebě a nabíjení baterie jsou měřena interně ve střídači, zbytek pak pomocí bočníků a děličů napětí, které lze dokoupit s konvenčními systémy nebo je snadno navrhnout. Data pro předpověď jsou získávány z webové stránky, zabývající se sběrem dat o počasí. Data jsou čteny pomocí zdrojového kódu, který je ukládá do databázového souboru a každých šest hodin je obnovuje. Databáze je čtena a zpracována SCADA systémem CX-Supervisor. Program CX Supervisor data získává, zpracovává a rozhoduje, co se bude dít na základě napsaného algoritmu. V případě potřeby je možné systém modifikovat dle přání uživatele. Jako možnou modifikaci zde uvedu například spínání výstupů programovatelného automatu pro ovládání osvětlení, vytápění, kde ale pro tento příklad je nutné mít patřičně zavedenou elektroinstalaci.

3.2 Topologie navrženého systému

Systém je složen z více částí, tyto části lze rozdělit na:

- Návrh FV systému
- Získání dat o předpovědi slunečního záření a teplotě
- Tvorba databáze
- Návrh inteligentního systému řízení (SCADA a HMI)

3.2.1 Návrh FV systému

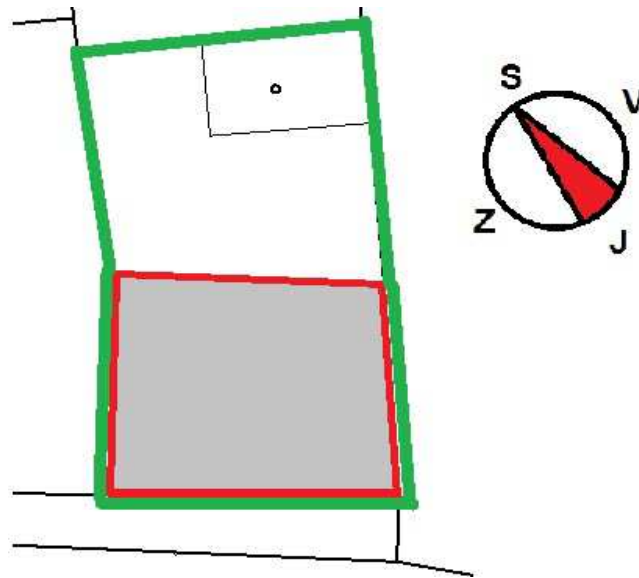
K návrhu FV systému byl použit program Energie 2016 od společnosti K-CAD, s.r.o.. Program lze využít k tepelným výpočtům objektů a tepelných ztrát objektů. Jelikož mým úkolem není vytvoření detailního návrhu FVE, tak zde návrh nebude podrobně popisovat. Níže je uveden stručný popis v bodech, které je nutné dodržet, aby byl návrh FV systému úspěšný.

Návrh FV systému se skládá z následujících bodů:

- Popis objektu
- Vytvoření nového projektu
- Zadání vstupních dat
- Volba režimu provozu FV systému (ostrovní režim/síťový režim)
- Výběr klimatického profilu
- Zadání dat o celkové roční spotřebě
- Vytvoření profilů spotřeby daného objektu
- Zadání dat vytvořených profilů spotřeby
- Výběr FV panelů
- Výběr akumulátorů
- Provedení výpočtu programem
- Vygenerování protokolu o výpočtu
- Volba vhodného hybridního měniče

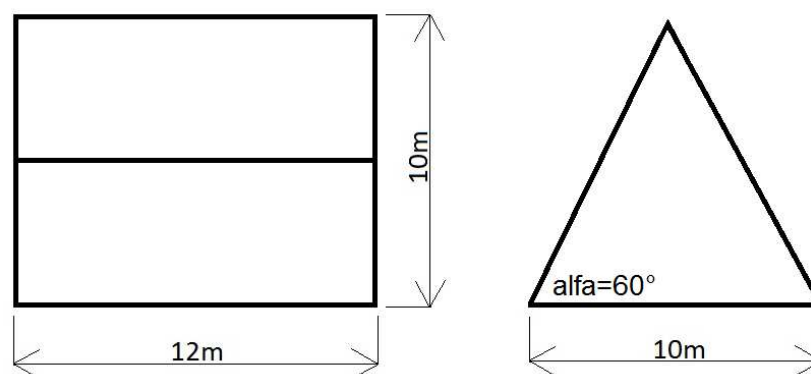
3.2.1.1 Popis objektu

Jedná se o rodinný dům v Ústeckém kraji v obci poblíž města Louny. S celkovou výměrou objektu 240m². Na obrázku č. 24 je znázorněná dispozice objektu, kde zelená barva ohraničuje hranice objektu a červená barva ohraničuje samotný objekt pro návrh. [34]



Obr. 24: Dispozice objektu

Objekt je svou polohou orientovaný na jihovýchod s azimutem -10° . Objekt má sedlovou střechu s plechovou krytinou, tudíž se nemusíme bát o přetížení střechy. Sklon střechy je 60° , což není zcela ideální sklon. V letních měsících je ideální sklon 30° , v zimních měsících pak 60° , proto by se měl volit kompromis, optimálně 45° . Vzhledem k poloze objektu umístím panely na jihozápadní část střechy, kde nebude docházet k žádnému zastínění panelů, nicméně při východu slunce nebudou panely zpočátku plně osvětleny. [34]



Obr. 25: Rozměry a sklon střechy

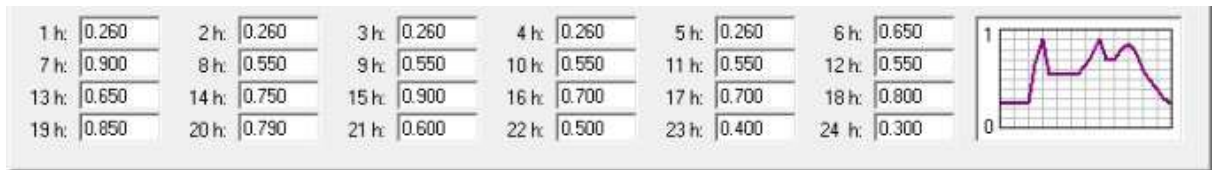
3.2.1.2 Tvorba profilu spotřeby

Základem pro tvorbu profilu spotřeby byla celková roční spotřeba energie, která v mém případě činila 4942 kWh. Tato hodnota je tvořena spotřebou elektrické energie běžných spotřebičů uvedených v tabulce č. 1. Ve spotřebě se uvažuje i vliv užití dalších spotřebičů, které nejsou běžně využívány, jako například ruční dýlenské náradí, kuchyňské spotřebiče. Tyto spotřebiče jsou zahrnuty v položce ostatní souhrn.

Tab. 1: Přehled běžně užívaných spotřebičů

Spotřebič	Spotřeba [W]	Provoz den[h]	Spotřeba za den [W]
Televize	120	6	720
Zesilovač audio	700	0,5	350
Chladnička 2x		24	1050
Kávovar	1200	0,5	600
Rychlovarná konvice	2000	0,2	400
Mrazák		24	630
Pračka	890	0,3	267
Sušička	740	0,3	222
Ohřívač vody	2200	2,5	5500
Osobní počítač	600	2	1200
Ostatní souhrn	2000	1,3	2600

Pro realizovaný návrh, jsem rozdělil dny na všední dny a dny o víkendu. Tyto dny se pak co do spotřeby cyklicky opakují. Pro všední den jsem bral v potaz, že v ranních hodinách do 5 hodit se na spotřebě podílejí jen přístroje, které jsou zapojeny do sítě nepřetržitě. Největší podíl na konstantní spotřebě má lednice, mrazák, bojler, kotel a elektronické přístroje v pohotovostním režimu. Během dne se spotřeba elektrické energie nepatrně zvedne, což je dáno ranním vstáváním do zaměstnání, popřípadě spuštění přístrojů jako jsou myčka, pračka, sušička. Následně spotřeba opět klesá. Zvyšovat se začne po příchodu členů domácnosti ze zaměstnání či školy, to můžeme pozorovat v rozmezí hodin od 15:00 až do 22:00, dále již spotřeba opět klesá. V nepracovní den je profil spotřeby malinko odlišný z důvodu setrvání členů domácnosti. Jednotlivé rozdíly již nebudu popisovat. Přehled toho, jak se dny liší co do spotřeby je uveden na obrázku č. 26 a č. 27.



Obr. 26: Relativní profily spotřeby všední den



Obr. 27: Relativní profily spotřeby den o víkendu

3.2.1.3 Možnost volby FVE

Máme na výběr z dvou možností návrhu FVE. První je zvolit klasickou FVE, kde se využije maximálně 60% vyrobené energie a zbytek se dodá zpět do elektrické sítě za dosti nevýhodných podmínek. Druhá možnost je zvolit hybridní FVE, kde dojde k využití veškeré energie vyprodukované hybridní FVE. Přebytečná energie je akumulována do akumulátorů a lze ji použít v případě potřeby. Výhodou hybridních FVE je možnost dodání přebytečné energii zpětně do sítě stejně jako klasické FV elektrárny. [34]

Hybridní systémy sice umožňují dodávat přebytečnou energii do sítě, ale pouze za podmínky, že ji už nelze nijak lépe využít například (akumulátor je plně nabit, spotřeba objektu je menší než výkon dodávaný FV elektrárnou). Výhodou je, že v případě nedostatku slunečního záření je možnost nedostatky pokrýt z distribuční sítě. [34]

Další výhodou hybridních FVE, je možnost požádat o dotaci z programu Nová zelená Úsporám, což u klasických FVE není možné. O tuto dotaci lze požádat v případě, pokud je spotřeba vyrobené energie u hybridní FVE minimálně 70%. U instalace s ročním ziskem alespoň 1700 kWh, je možné obdržet dotaci ve výši až 70 tisíc korun, u instalace se ziskem alespoň 3000 kWh je možné obdržet dotaci ve výši 100 tisíc korun. V případě většího zisku lze dosáhnout na částku až 160 tisíc korun. Z toho důvodu je dnes většina navrhovaných solárních systémů hybridních. Další z důvodů je nezávislost provozu na síti, kvůli rostoucím cenám elektřiny. [34]

3.2.1.4 Prvky mnou navrženého FV systému

Mnou navržený systém je hybridní a skládá se z dvanácti monokrystalických panelů HIT-N245 od výrobce Panasonic s celkovým špičkovým výkonem 3675 Wp. Střídač byl vybírán dle následné možnosti řízení SCADA systémem, v mém případě padla volba na typ XTH 5000-24 od výrobce Studer-innotec. Regulátor napětí je opět od výrobce Studer Innotec a jedná se o typ VT-80. Volba stejného výrobce u střídače a regulátoru je z důvodu řízení systému jako celku. Aby bylo systém řídit, bylo ho nutné dovybavit komunikační jednotkou Xcom-232i. Veškerá přebytečná energie je akumulována do dvanácti solárních akumulátorů Hoppecke Solar.bloc o napětí 6V a kapacitě 250Ah, které jsou zapojeny sérioparalelně, aby jejich výsledné napětí bylo 12V s kapacitou 1500Ah.

3.2.1.4.1 Fotovoltaický modul Panasonic HIT-N245

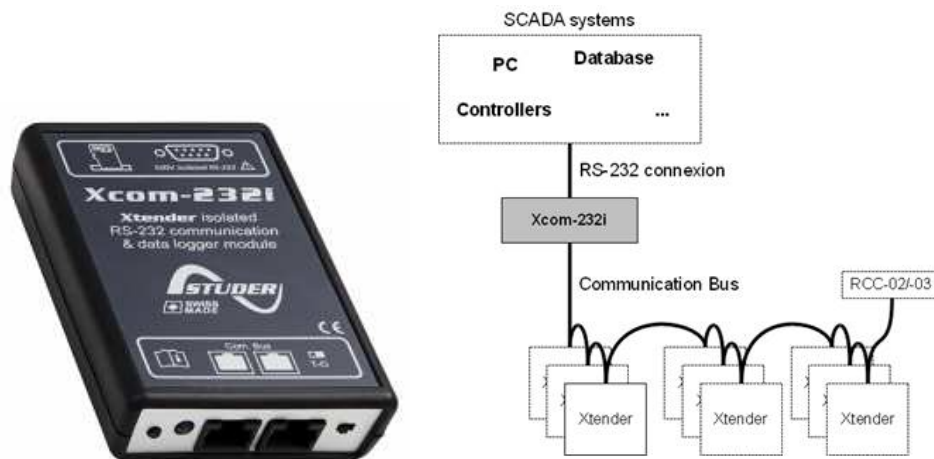
Jedná se o FV panel německého výrobce Panasonic. Panely byly vybrány s ohledem na rozměry střechy, instalovaný výkon a dostupnosti výrobce v daném regionu. Více parametrů je uvedeno v příloze č. 3.[34]



Obr. 28: FV panel Panasonic HIT-N245 [35]

3.2.1.4.2 Komunikační modul Xcom-232i

Jedná se o komunikační modul výrobce Studer innotec. Pomocí tohoto komunikačního modulu jsme schopni monitorovat a ovládat celý systém. Je možné individuální nastavení na běžném ovládacím panelu. Po připojení k nadřazenému SCADA systému (PLC, PC, mikroprocesoru), lze systém řídit zasílanými příkazy pomocí komunikačního protokolu s využitím portu RS232. Jednotlivé prvky střídač, regulátor napětí, ovládací jednotka jsou sériově propojeny na obrázku č. 29.



Obr. 29: Komunikační modul Xcom-232i a jeho zapojení [36]

3.2.1.4.3 Střídač Studer Innotec XTH 5000-24

Jedná se o hybridní střídač švýcarského výrobce Studer innotec. Nespornou výhodou je částečná možnost řízení a monitoringu po připojení ke komunikační jednotce. Střídač je schopný konstantně dodávat do objektu výkon 4500VA. V případě špičky dodá výkon až 12000VA po dobu 5 minut a nebo výkon 5000VA po dobu 30minut. Výstupní napětí je čistá sinusoida. Více parametrů je uvedeno v katalogovém listu v příloze č. 1. [34]



Obr. 30: Střídač XTH 5000-24 [36]

Výhodou u měničů tohoto výrobce je implementovaný algoritmus, který si řídí toky energie, tak aby využití energie bylo co nejefektivnější. Tato funkce se jmenuje smartboost. V případě dostatečně nabité baterie se zdrojem pro střídač stane baterie, v případě částečně nabité baterie se zdrojem pro střídač stane síť. V případě vybitého akumulátoru se zdrojem stane síť a akumulátor se bude nabíjet. Pro ilustraci funkce je uveden obrázek č. 31. [34]



Obr. 31: Funkce smartboost [34]

3.2.1.4.4 Regulátor nabíjení Studer Innotec VT-80

Regulátor je zvolen tak, aby byl kompatibilní se zbytkem FV systému. Po připojení FV systému ke komunikační jednotce Xcom-232i, ho lze monitorovat a ovládat podobně jako střídač. Regulátor napětí umožňuje pracovat s napětím do maximální hodnoty 150V v režimu otevřeného obvodu, pro uzavřený obvod pak jen s hodnotou 145V, proto jsem v mém případě mohl spojit jen tři FV panely do série. Pro vyšší efektivitu přeměny, lze tento regulátor umístit přímo pod střešní krytinu. Technická data jsou uvedeny v příloze č. 2. [34]



Obr. 32: Regulátor nabíjení VT-80 [36]

3.2.1.4.5 Akumulátory HOPPECKE SOLAR BLOC 250AH, 12V

Jedná se o akumulátor, vyvinutý přímo pro solární systémy, který vyniká extrémní cyklickou odolností. Při vybíjení akumulátoru na 40% jeho kapacity, je akumulátor schopný vydržet až 3200 nabíjecích cyklů. Díky použité technologii, se jedná o zcela bezúdržbové akumulátory. Hlavní výhodou je cyklická odolnost, použití u ostrovních systémů, dlouhá životnost, vysoký výkon a kompatibilita s mnoha technologiemi. V mém případě jsem využil kombinaci dvanácti akumulátorů zapojených sérioparalelně pro dosažení napětí 12V a celkové kapacity 1500Ah. Více technických informací je uvedeno v technickém listu v příloze č. 4. [24]



Obr. 33: Solární baterie Hoppecke 12V 250Ah [24]

3.2.1.4.6 Řídící PC

Pro účely řízení je možné použít libovolný počítač s komunikačním portem RS232. Počítač bude v provozu 24 hodin denně, a proto zvolím počítač s co nejmenší spotřebou elektrické energie. Pro případ výpadku elektrické energie a nutného zálohování dat je systém doplněn o zařízení UPS. V případě nedostatku místa v rozváděči či nevíle kupovat a udržovat počítač, je možné použít serverový počítač, připojený kdekoli k ethernetové síti. Výpočty se provádí vzdáleně na síti, v tomto případě je nutný převodník z ethernetové sítě na ovládací port RS232.

3.2.1.4.7 Programovatelný logický automat (PLC)

V mém případě jsem použil PLC od výrobce OMRON, konkrétně řadu CS1G-CPU42H. PLC používám primárně ke sběru měřených dat (napětí, proudů a následně výpočtu výkonů). Měření je zabezpečeno pomocí bočníků a děličů napětí, data z těchto prvků jsou zpracovány pomocí analogové karty CS1W-AD081-V1. Jedná se o analogovou kartu o 8vstupech, její nastavení probíhá interně v PLC. V případě rozšíření funkce systému a tím zvýšení inteligence budovy je PLC možné využít například ke spínání relátek, která by spínala osvětlení, ventilaci, topení, k tomuto účelu by se použily výstupní karty dle potřeby. Sestava PLC by pak vypadala podobně jako na obrázku č. 34.



Obr. 34: Sestava PLC a připojených karet

3.2.1.5 Zapojení elektroinstalace

Jedná se teoretický popis schématu zapojení. Z obrázku č. 35 je vidět, že se jedná o klasickou elektroinstalaci bytu/domu s hlavním domovním rozvaděčem, doplněnou o FV systém a jemu přidružený podružný rozvaděč FVE.

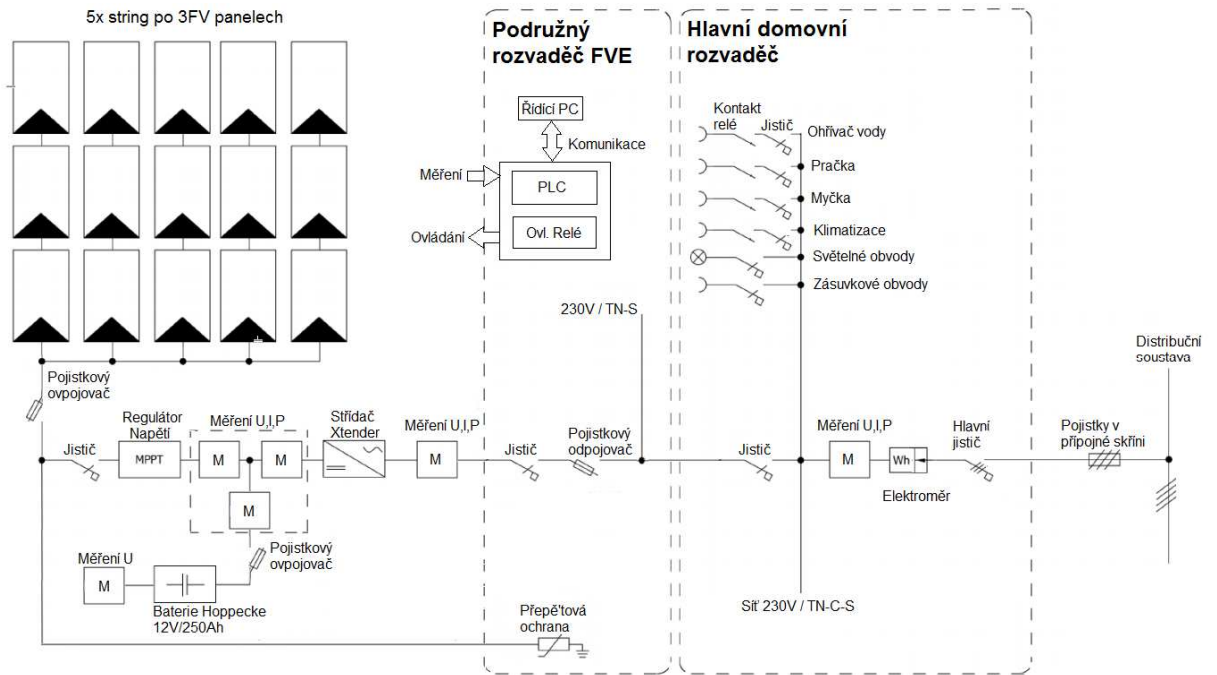
Za zmínku stojí zapojení FV panelů. Volil jsem zapojení maximálně tří panelů do série, takže vzniklo 5 stringů. Jednotlivé stringy jsou spojeny paralelně. Kombinace pouze tří FV panelů je z důvodu na připojení regulátoru nabíjení (MTTP).

Pro ochranu FVE a domácího rozvodu jsou panely připojeny přes pojistkový odpojovač, jistič a přepět'ovou ochranu, FVE je ze strany střídavé sítě připojena přes pojistkový odpojovač a jistič, akumulátor je chráněn pojistkovým odpojovačem.

Celý systém rozvodu elektrické energie včetně FVE, je doplněn o systém měření označený v obrázku symbolem M. Měření je instalováno z důvodu řízení toků energie a možného přepínání mezi FVE a distribuční sítí. Vzhledem k možnosti měření energie, kterou střídač nabízí, by se přídatný systém měření nemusel instalovat na straně baterie a stejnosměrné straně. V přehledovém schématu je uvedeno vnější měření, které je realizováno vhodně dimenzovanými bočníky a děliči napětí připojenými do analogové karty PLC.

Podružný rozvaděč FVE obsahuje řídicí počítač, propojený s komunikačním modulem Xcom-232i, internetovou sítí, programovatelným logickým automatem (PLC), FV střídačem a regulátorem napětí. V řídicím počítači je spuštěn navržený SCADA/HMI systém, který komunikuje s jednotkou PLC, ze které získává měřená data. SCADA systém je propojen s jednotkou střídače a regulátorem napětí skrze komunikační modul Xcom-232i, pomocí kterého se celý systém FVE monitoruje a řídí.

Hlavní domovní rozvaděč je zapojen klasickým způsobem jako běžná elektroinstalace. Jediné co lze možné doplnit jsou kontakty relátek, která jsou spínána řídicím systémem SCADA/HMI. Toto jsem uvedl, jen jako možné rozšíření systému, které nám umožní vyšší míru inteligence v domě. Kde by se pak jednotlivá světla dala ovládat jakýmkoliv vypínačem. S tímto je však nutné počítat už v době realizace projektové dokumentace.



Obr. 35: Přehledové schéma zapojení [34]

3.2.2 Získání dat o předpovědi slunečního záření a teplotě

Data pro předpověď jsou získávány s využitím služby PV forecast na základě html dotazu. Služba poskytuje předpověď slunečního záření ve W/m^2 na 24h, 48h a 72h dopředu stejně tak teplotu a úhrn dešťových srážek, předpověď je aktualizována 2x za den. Záleží na příkazu, který si sestavuje uživatel. Služba pro předpověď využívá více nezávislých zdrojů, ze kterých algoritmu stanoví předpověď. Jediné co je třeba udělat je se registrovat, jako registrovaný uživatel máte možnost přístupu na 14dní s neomezenou možností aktualizace dat, poté je již služba zpoplatněna.

3.2.2.1 HTML dotaz předpovědi:

Jako registrovaný uživatel dostanete licenční klíč, který je nutné zadat do dotazu spolu se zeměpisnou šířkou a délkou. Další možnosti úpravy kódu a sběru dat jsou zobrazeny na obrázku č. 36.

html dotaz pro předpověď slunečního záření a vysvětlením na obrázku č. 36 :

<http://www.pvforecast.cz/api/?key=f7rhf8&lat=50.362&lon=13.845&format=simple&type=hour&number=72>

html dotaz pro předpověď teploty a vysvětlením na obrázku č. 36:

<http://www.pvforecast.cz/api/?key=f7rhf8&lat=50.362&lon=13.845&forecast=temp&format=simple&type=hour&number=72>

Klíč	Význam	Možné hodnoty	Výchozí hodnota
key *	Unikátní uživatelský klíč	Šestimístný kód přidělený při registraci	
lat *	Zeměpisná šířka	Číslo vyplněné při registraci (tři desetinná místa)	
lon *	Zeměpisná délka	Číslo vyplněné při registraci (tři desetinná místa)	
forecast	Určení typu předpovědi. Sluneční svit (pv), teplota (temp), srážkový úhrn (rain)	pv/temp/rain	pv
format	Formát dat	simple/csv/json	simple
type	Hodinové nebo denní sumy	hour/day	hour
number	Délka předpovědi v hodinách nebo dnech (zadané číslo udává počet hodnot v odpovědi) **	24/48/72 *** (type = hour) 1/2/3 *** (type = day)	24 (type = hour) 1 (type = day)
date	Umožňuje vypnout časovou značku u formátu Simple. U formátů JSON a CSV nemá význam.	1 = zapnuto 0 = vypnuto	1
dst	Automatická volba letního času (daylight saving time)	1 = zapnuto 0 = vypnuto	1

* Povinné parametry.

** Pokud předpověď na danou dobu není dostupná, je v odpovědi místo příslušné hodnoty prázdné místo (formáty Simple a CSV) nebo hodnota NULL (formát JSON).

*** Předpověď na 72 hodin (3 dny) není podporována a jedná se o přípravu na možné budoucí rozšíření služby.

Obr. 36: Možnosti dotazu pro získání předpovědi [31]

3.2.2.2 Formáty dat

Data je možné získat ve třech formátech a to formátu:

- simple
- csv
- JSON.

3.2.2.2.1 Formát simple:

Je formát, který poskytuje data v následující podobě:

2016-04-12|0|0|0|0|0|0|15|80|262|512|538|507|481|565|532|235|313|176|108|28|0|0|0

3.2.2.2 Formát csv:

Sloupce dat jsou u tohoto formátu oddělené pomocí středníků. První sloupec obsahuje časový údaj, druhý hodnotu slunečního záření.

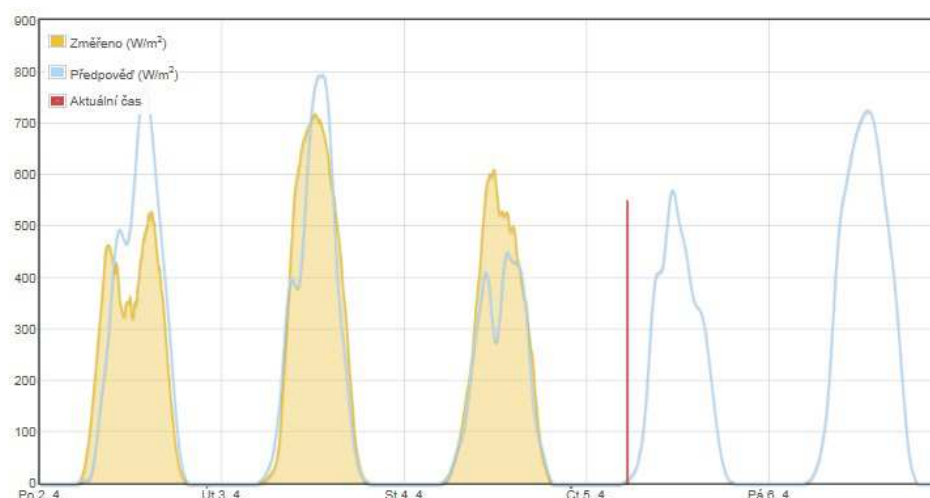
```
Cas;Osvit (W/m2)
2016-04-12 00:00:00;0
2016-04-12 01:00:00;0
2016-04-12 02:00:00;0
2016-04-12 03:00:00;0
2016-04-12 12:00:00;450
2016-04-12 15:00:00;650
2016-04-12 18:00:00;20
```

3.2.2.3 Formát JSON:

Tento formát najde uplatnění ve vyšších programovacích jazycích a má následující formu.

```
[["2016-04-12 00:00:00",0],["2016-04-12 01:00:00",0],["2016-04-12 02:00:00",0],["2016-04-12 03:00:00",0],["2016-04-12 04:00:00",0],...
```

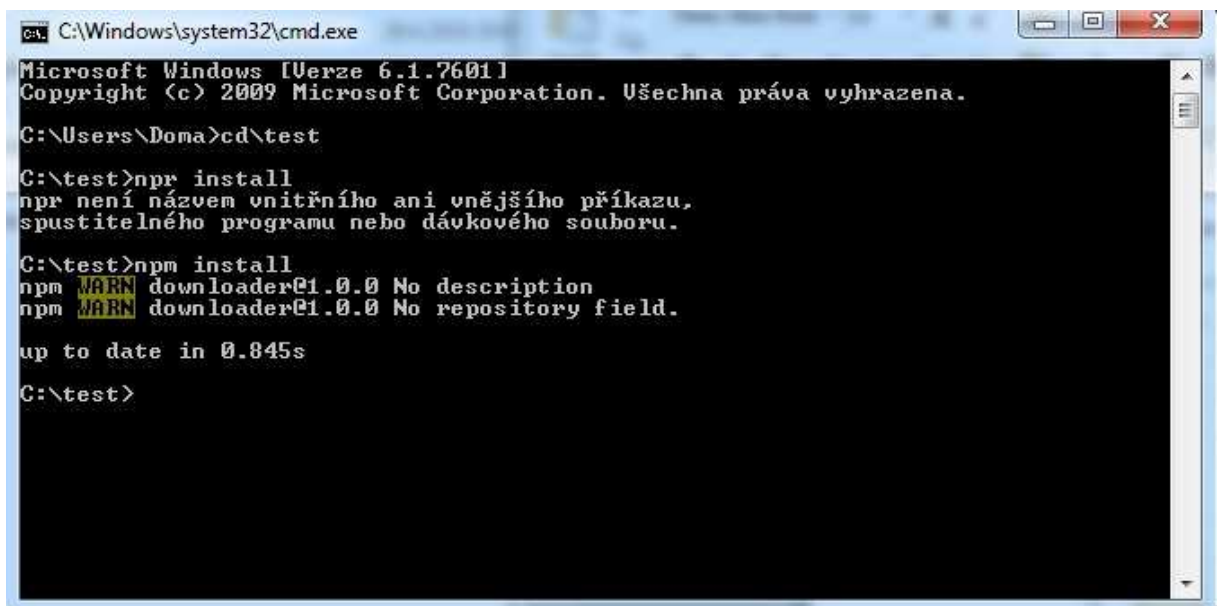
Přesnost předpovědi je vidět na obrázku č. 37. K této službě je možné dokoupit řídicí algoritmus s řídicím PLC, tím lze zvýšit přesnost předpovědi pro danou lokalitu. Data se aktualizují 2x denně a s každou aktualizací se předpověď zpřesňuje. Aktualizace probíhá po poledni aktuálního dne. Před polednem dostanete data z předchozího dne, po poledni pak nová přesnější data.



Obr. 37: Předpověď slunečního záření ve městě Buštěhrad. [31]

3.2.3 Tvorba databáze

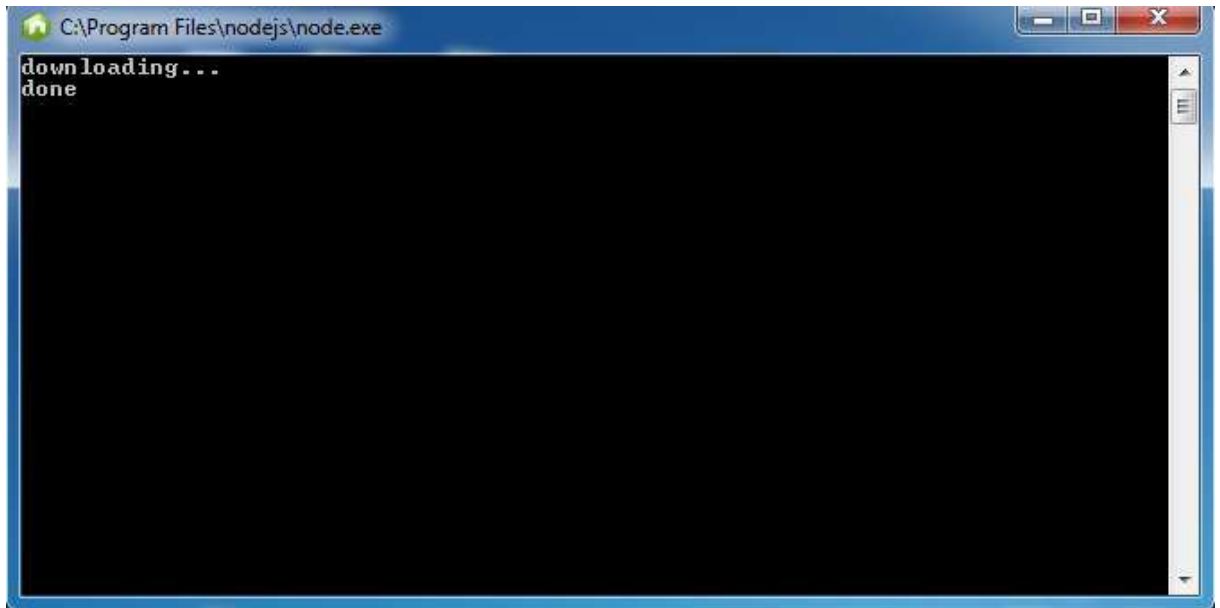
Databáze je vytvořena v programu node.js s využitím java scriptu. Program stáhneme ze stránky <https://nodejs.org/en/download/> a nainstalujeme. Následně rozbalíme přiložená data v příloze pod názvem database. Hlavní část programu se nachází v souboru main.js, kde je i zdrojový kód. Pro první spuštění je nezbytné ve složce se soubory smazat složku node_modules, kde jsou vytvořené závislosti pro skript. Moduly je nutné znovu nainstalovat pomocí příkazového řádku. Spustíme tedy příkazový řádek příkazem cmd, a zadáme cestu k souborům. V mém případě to bylo na disku C ve složce test. Po zadání příkazu „npm install“ se příslušné moduly znovu nainstalují do složky s původními soubory. Příkaz lze vidět na obrázku č. 38.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Verze 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Všechna práva vyhrazena.
C:\Users\Doma>cd\test
C:\test>npr install
npr není názvem vnitřního ani vnějšího příkazu,
spustitelného programu nebo dávkového souboru.
C:\test>npm install
npm WARN downloader@1.0.0 No description
npm WARN downloader@1.0.0 No repository field.
up to date in 0.845s
C:\test>
```

Obr. 38: Příkazový řádek pro instalaci modulů

Následně zbývá spustit program main.js v programu node.js nebo opět v příkazovém řádku. Po spuštění souboru main.js se vám otevře okno obrázek č. 39. Kde je informace o stahovaných datech a informace o dokončení. Data se uloží do databáze pod názvem db.sqlite a bude se pravidelně aktualizovat každých 6 hodin, pro aktualizaci je nezbytné nechat program spuštěný. Vše se děje na základě níže popsaného skriptu.



Obr. 39: Spuštění programu main.js

3.2.3.1 Vzhled databáze:

Vzhled databáze je na obrázku č. 40

	id ↕	zareni ↕	teplota ↕
1	1	55	30.2
2	2	47	29.9
3	4	49	28.6
4	5	45	28.1

Obr. 40: Vzhled databáze db.sqlite

3.2.3.2 Skript databáze:

Skript se nachází v souboru main.js. V adresáři node_modules jsou závislosti nainstalované, v souborech package, json a yarn.lock jsou závislosti zadefinované. Na řádce 4 je funkce pro stažení záření, na řádce 9 pro stažení teploty. Dále na řádce 14 je funkce, která otevírá databázi a stahuje data o záření, teplotě a provede nový zápis dat. Na řádce 27 je definovaná funkce main, která běží v nekonečné smyčce, kde zavolá stažení dat a pak čeká 6hodin do dalšího stažení, což je řádek 32. Na řádce 36 se program spouští s případným výpisem do konzole.

```
-----  
const sqlite = require('sqlite');  
const request = require('superagent');  
  
const getRadiation = async () => {  
  const response = await  
request.get('http://www.pvforecast.cz/api/?key=f7rhf8&lat=50.362&lon=13.845&format  
=json&type=hour&number=72');  
  return JSON.parse(response.text);  
};  
  
const getTemperature = async () => {  
  const response = await  
request.get('http://www.pvforecast.cz/api/?key=f7rhf8&lat=50.362&lon=13.845&forecas  
t=temp&format=json&type=hour&number=72');  
  return JSON.parse(response.text);  
};  
  
const download = async () => {  
  const db = await sqlite.open('./db.sqlite');  
  await db.run('DELETE FROM database');  
  
  const radiation = await getRadiation();  
  const temperature = await getTemperature();
```

```
    await Promise.all(radiation.map((_, index) => db.run(  
      'INSERT INTO database (id, zareni, teplota) VALUES (?, ?, ?)',  
      [index + 1, radiation[index][1], temperature[index][1]]  
    )));  
  };
```

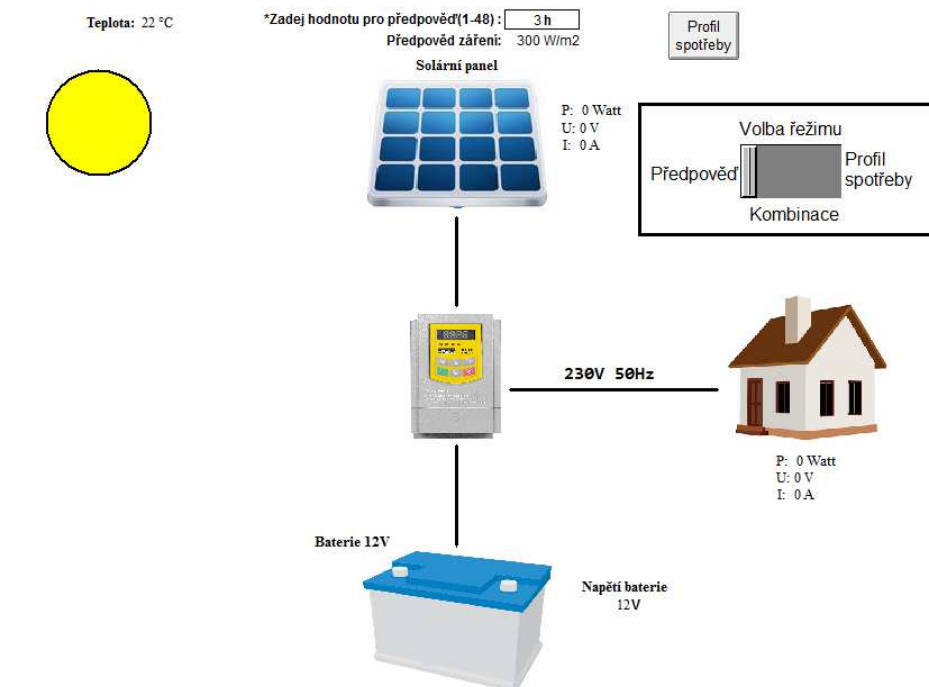
```
    const main = async () => {  
      while (true) {  
        console.log('downloading...');  
        await download();  
        console.log('done');  
        await new Promise((resolve) => setTimeout(resolve, 6 * 60 * 60 * 1000)); // 6 hours  
      }  
    };
```

```
    main().catch((error) => {  
      console.error(error);  
      process.exit(1);  
    });
```

3.2.4 Návrh inteligentního systému řízení (SCADA a HMI)

Návrh řídicího SCADA systému je proveden v programu CX Supervisor. CX Supervisor je program vybavený nesčetnou řadou funkcí pro využití v oblasti SCADA systémů. Lze v něm vytvořit nesčetné množství aplikací pro definované účely, od jednoduchých přes složitější až po ty nejzapeklitější. Pro zjednodušení návrhu lze využít četné množství předdefinovaných funkcí a knihoven. [32]

Před programováním je nutné vytvořit nový projekt, vzhled operačního prostředí pro uživatele obrázek č. 41. Pokračujeme zanesením ovládacích prvků. Po umístění prvků je nutné definovat jednotlivé proměnné obrázek č. 50, a přiřadit jim definované vlastnosti obrázek č. 51, Proměnné je následně nutné propojit s vloženými prvky. Dále následuje programování. Je na výběr ze dvou možných programovacích jazyků. VBScriptu, což je zjednodušený Visual Basic, nebo je možno použít přímo Visual Basic. Programování je velice individuální, programovací skript je možné vkládat do jednotlivých prvků a volat je zvlášť, což nevyužívám nebo psát jeden skript pro celý projekt, což následně zlepšuje přehlednost programu. Po spuštění vytvořeného projektu je daný skript spuštěn v nekonečné smyčce, kde se aktualizuje programátorem nastavenou dobou, v mém případě to je každou sekundu.



*Hodiny pro předpověď jsou počítané od aktuálního času. Například je 12:32 chci předpověď 3 hodiny napřed, zadám tedy 3. Algoritmus vezme hodnotu aktuální hodiny 12 a přičte k ní 3, výsledek je 15. Předpověď je tedy brána pro 15:00. Obdobně je to pro teplotu.

Obr. 41: Vytvořené operační prostředí včetně ovládacích prvků

3.2.4.1 Funkce navrženého systému

3.2.4.1.1 Nabíjení baterie

Měřicí systém střídače hlídá úroveň nabití baterie, úroveň nabití pak lze sledovat i přidaným systémem nejčastěji pomocí děliče napětí. Nabíjení je zajištěno regulátorem napětí Studer innotec VT-80. Výhodou tohoto regulátoru je, že po připojení na sběrnici s komunikační jednotkou z něj lze snadno získat potřebná data.

Baterie je nabíjena za jakéhokoliv vhodného okamžiku s dostatečným slunečním zářením či dle nastaveného profilu spotřeby nebo kombinací zmíněných možností. Baterie může být nabíjena i z distribuční sítě, tato možnost se využije pouze v případě vybité baterie a nepříznivého slunečního záření.

3.2.4.1.2 Nabíjení baterie v kombinaci se sítí

Navržený měnič má v sobě implementovanou funkci smartboost, pomocí které je co nejvýhodněji řízen tok elektrické energie ze systému ke koncovému zákazníkovi. Děje se to zcela automaticky, tento algoritmus lze ovlivnit zasílanými příkazy měniči, který pak upřednostní uživatelský požadavek. V případě nabití baterie a dostatečného slunečního záření se největší část energie vezme z FV panelů a baterie a v případě nedostatku potřebné energie se zbytek vezme z elektrické sítě.

U mnou navrženého systému je funkce obdobná. Pokud je zvolen režim předpovědi, vím přibližně velikost výkonu FVE. Dalším krokem je změření úrovně nabití baterie a úrovně odebíraného výkonu objektem. Pokud z předpovědi vím, že výkon FVE bude pro provoz budovy dostatečný, zašlu příkaz měniči, aby nabíjel baterie, technologie smartboost, pak vyhodnotí vhodné řízení toku. V případě zjištěného nedostatku energie z FVE se měnič zašle příkaz aby se připojil na síť, kde případný nedostatek kompenzuje.

3.2.4.1.3 Ostrovní provoz

Ostrovní provoz, čili provoz systému nezávisle na síti je provozován do 60% stavu vybití baterie. Tento stav se využívá převážně v noci pro pokrytí spotřeby objektu. Energie je využita zejména pro provoz spotřebičů, které vyžadují napájení neustále jako lednice, mrazák a veškeré spotřebiče v úsporném režimu. V nutném případě je možné energii spálit

v akumulacním ohříváči.

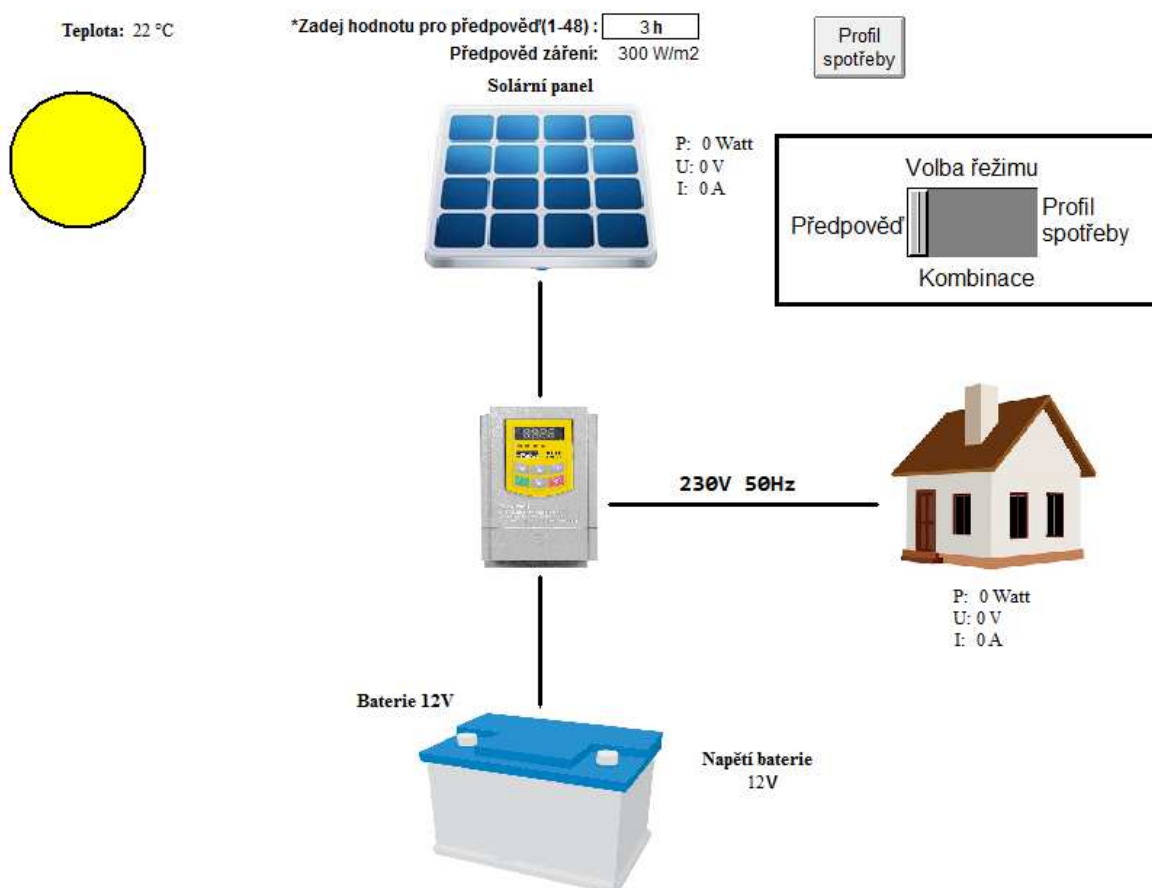
3.2.4.1.4 Provoz na síť

Provoz čistě na síť je využit pouze v případě nedostatečného výkonu FVE. Režim je volen automaticky měničem, dle uživatelem zvoleného režimu. V tomto režimu je nabíjena i baterie. Toho se využije v případě nepříznivé předpovědi slunečního záření na x hodin napřed.

3.2.4.2 Režimy řízení navrženého systému:

3.2.4.2.1 Řízení na základě předpovědi

Tento režim je volen uživatelem za pomoci přepínače volby režimu. Režim je, vybrán pokud se přepínač nachází v levé pozici u popisu předpověď. V tomto režimu je nutné navolit hodiny předpovědi. Je možnost zadat hodnotu od 1h do 72h.



*Hodiny pro předpověď jsou počítané od aktuálního času. Například je 12:32 chci předpověď 3 hodiny napřed, zadám tedy 3. Algoritmus vezme hodnotu aktuální hodiny 12 a přičte k ní 3, výsledek je 15. Předpověď je tedy brána pro 15:00. Obdobně je to pro teplotu.

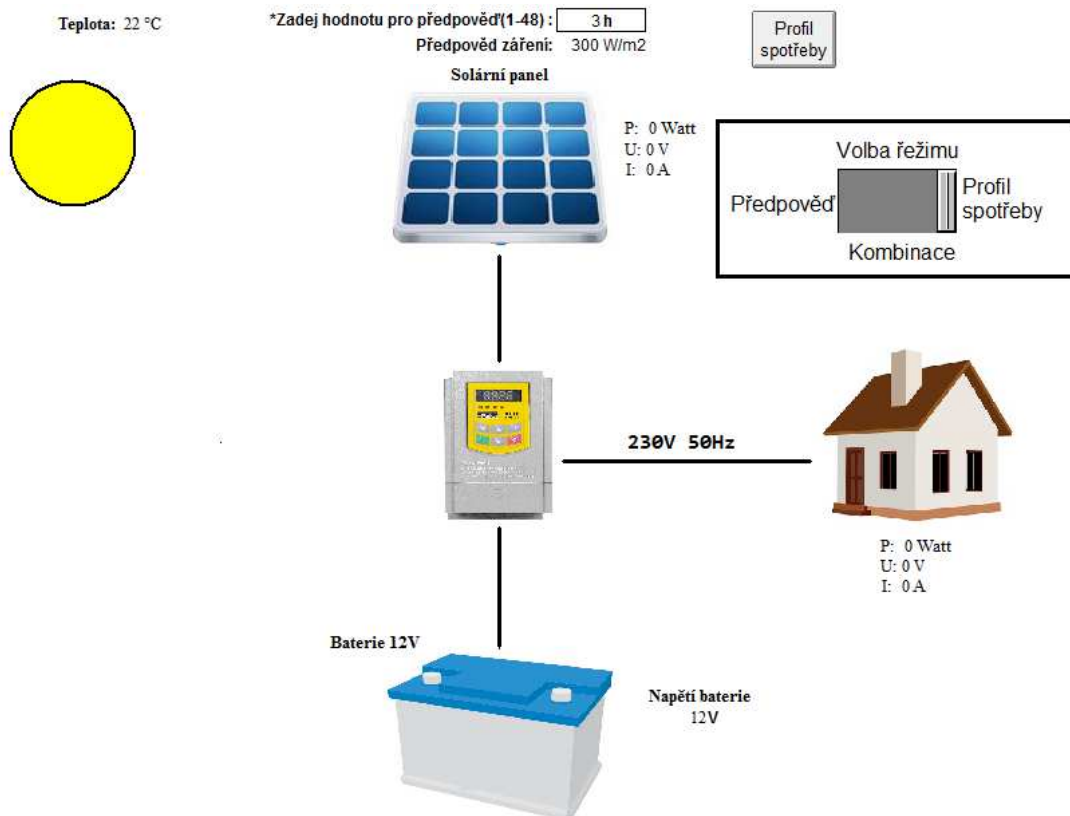
Obr. 42: Obrazovka SCADA systému se zvoleným režimem předpovědi

Princip řízení:

V tomto režimu se nejprve z databáze vyčte hodnota teploty a slunečního záření pro zvolenou hodinu a změří se aktuální odběr objektu. Na základě předvídaného záření se vezme možný výkon, který bude FVE poskytovat. Výsledná hodnota je porovnána s aktuálně odebíranou hodnotou, na základě které dojde k vyhodnocení činnosti měniče. Bude vybíjet či nabíjet? V případě dostatečného výkonu FVE, je měniči zaslán příkaz pro vybíjení baterie, což znamená provoz čistě na baterii, pouze v případě nedostatečného výkonu se k systému připojí síť. V opačném případě je měniči zaslán příkaz nabíjet, kde dojde alespoň k částečnému či úplnému nabití baterie.

3.2.4.2.2 Řízení na základě profilu spotřeby

Tento režim je volen uživatelem za pomoci přepínače volby režimu. Režim je vybrán, pokud je přepínač v pravé pozici u popisu profil spotřeby. K tomuto režimu se váže tlačítko profil spotřeby, který uživatel musí vyplnit. Jedná se o 24hodnot předpokládané spotřeby na celý den.



*Hodiny pro předpověď jsou počítané od aktuálního času. Například je 12:32 chci předpověď 3 hodiny napřed, zadám tedy 3. Algoritmus vezme hodnotu aktuální hodiny 12 a přičte k ní 3, výsledek je 15. Předpověď je tedy braná pro 15:00. Obdobně je to pro teplotu.

Obr. 43: Obrazovka SCADA systému se zvoleným režimem spotřeby

Spotřeba												
Regulátor napětí			Vstup střídače		Baterie		Výstup střídače		Sít'		Spotřeba domu	
P: 0 Watt U: 0 V I: 0 A			P: 0 Watt U: 0 V I: 0 A		P: 0 Watt U: 0 V I: 0 A		P: 0 Watt U: 0 V I: 0 A		P: 0 Watt U: 0 V I: 0 A		P: 0 Watt U: 0 V I: 0 A	
1h	2h	3h	4h	5h	6h	7h	8h	9h	10h	11h	12h	
0kW	0,3kW	0,3kW	0,3kW	0,5kW	0,6kW	0,8kW	0,8kW	0,4kW	0,3kW	0,3kW	0,3kW	
13h	14h	15h	16h	17h	18h	19h	20h	21h	22h	23h	0h	
0,4kW	0,5kW	0,8kW	0,8kW	0,8kW	0,8kW	0,8kW	0,8kW	0,8kW	0,8kW	0,3kW	0,3kW	

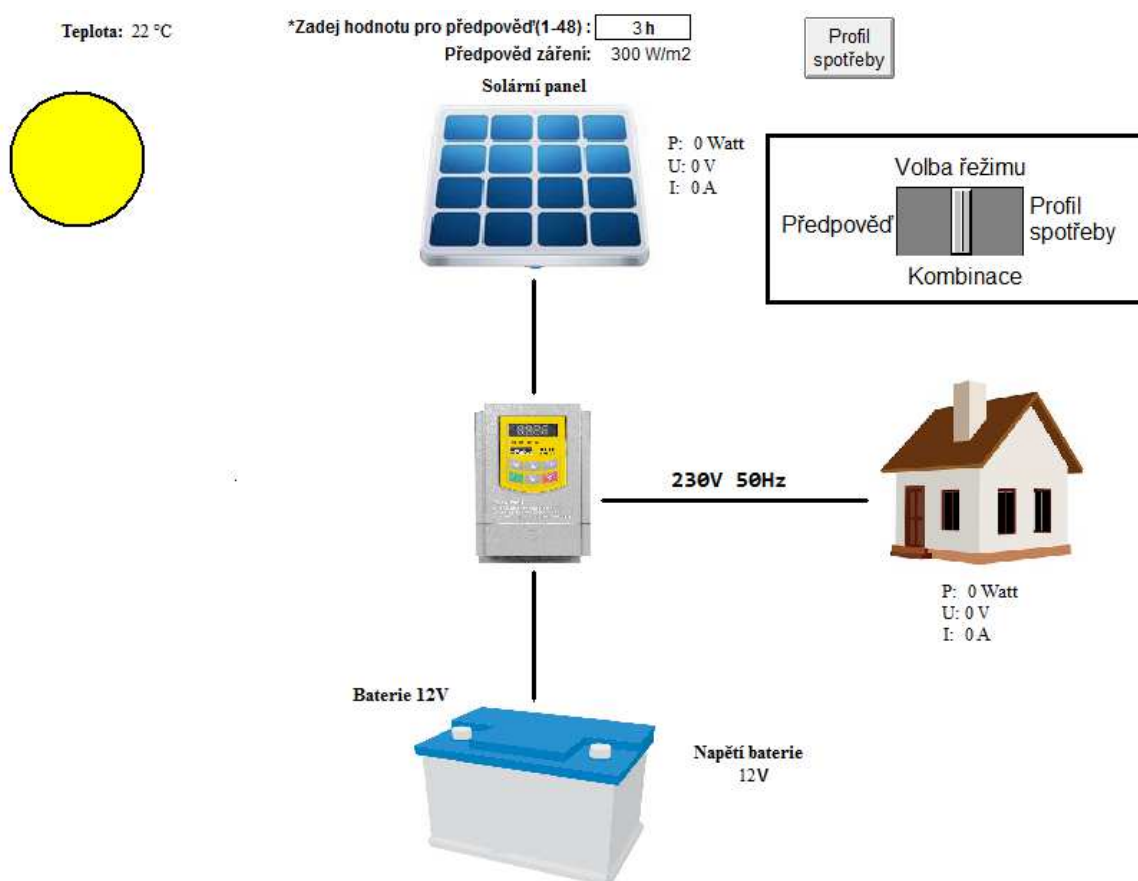
Obr. 44: Obrazovka profilu spotřeby a přehledu výkonů

Princip řízení:

Režim pracuje na základě porovnávání měřených hodnot a hodnot zadaných do profilu spotřeby. Například vezmu hodnotu profilu spotřeby pro 1h ráno a hodnotu si uložím do proměnné hod1, změřím si odebíraný výkon objektem proměnná (Spotřeba), výkon dodávaný sítí (Psít') a výkon poskytovaný FVE (Pstřídač). Algoritmus vezme hodnotu výkonu pro danou hodinu a porovná ji s aktuálními daty a dojde k vyhodnocení nejhodnějšího využití. V případě dostatečného výkonu FVE, se FVE stane zdrojem a případné přebytky energie budou nabíjet baterii. V opačném případě se baterie stane zdrojem pro FVE. V případě nedostatečného výkonu se k bateriím přidá sít', popřípadě se zdrojem stane pouze elektrická sít'.

3.2.4.2.3 Řízení dle kombinace předchozích dvou

Tento režim je volen uživatelem za pomoci přepínače volby režimu. Režim je vybrán, pokud je přepínač v prostřední pozici u popisu kombinace. Režim je spojený s tlačítkem profilu spotřeby a políčkem pro předpověď, které je nutné vyplnit. V podstatě se jedná o řízení založené na kombinaci předchozích dvou metod popsanych výše.



*Hodiny pro předpověď jsou počítané od aktuálního času. Například je 12:32 chci předpověď 3 hodiny napřed, zadám tedy 3. Algoritmus vezme hodnotu aktuální hodiny 12 a přičte k ní 3, výsledek je 15. Předpověď je tedy braná pro 15:00. Obdobně je to pro teplotu.

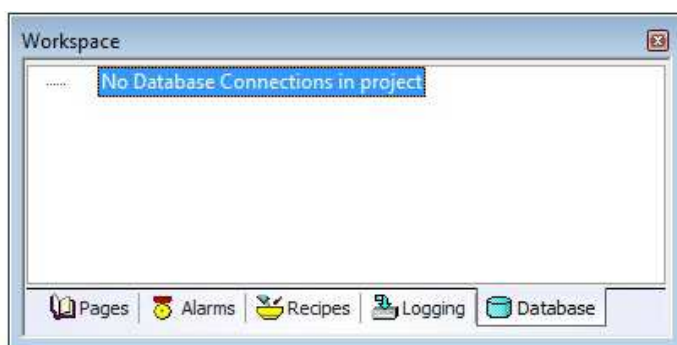
Obr. 45: Obrazovka SCADA systému se zvoleným kombinovaným režimem

Princip řízení:

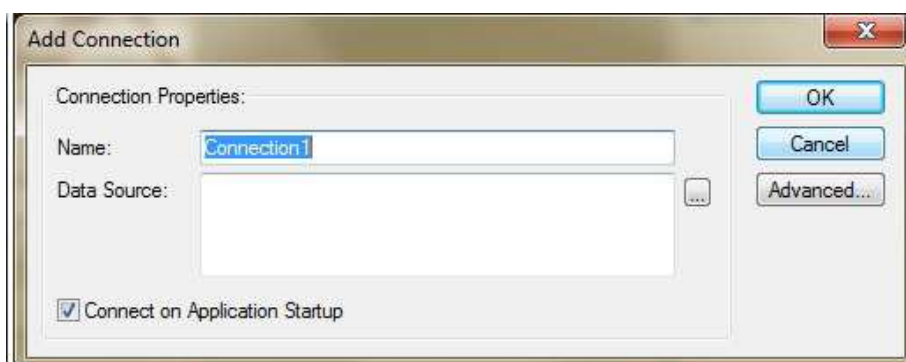
Tento režim je kombinací obou již popsanych metod. Kde na základě porovnání dojde algoritmem ke zvolení nejvhodnějšího využití energie. Prvotně se odečtou aktuální hodnoty výkonů pro objekt (Spotřeba), střídač (Pstřídač) a síť (Psíť). Poté se vezme hodnota předvídaného záření a hodnota z profilu spotřeby, například pro 1h ráno proměnná hod1. Tyto hodnoty se v algoritmu porovnají a dojde k volbě vhodného režimu střídače.

3.2.4.3 Otevření databáze v programu CX-Supervisor

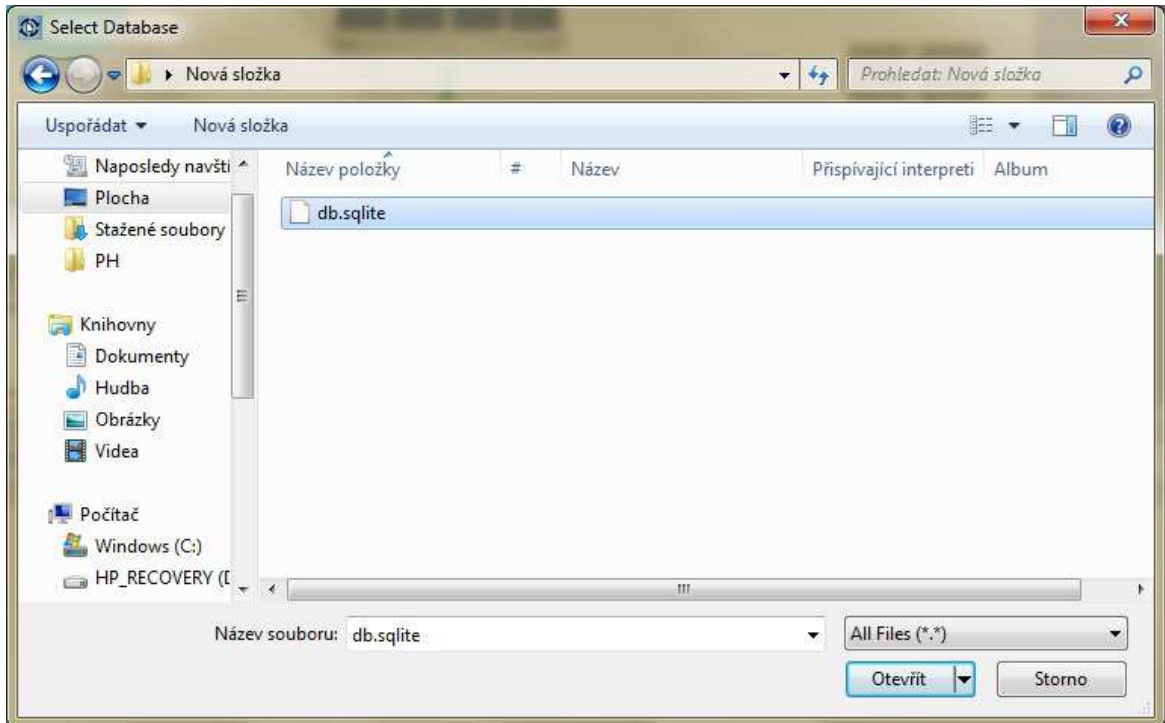
V okně work space obrázek č. 46 se zvolí záložka databáze, kde se následně se vytvoří propojení databázového souboru s programem CX-Supervisor obrázek č. 47. Pro propojení volím databázový soubor db.sqlite obrázek č. 48. Následuje přidání recordsetu obrázek č. 49, na základě kterého budu číst data dle vybraného ID. ID mi představuje hodiny od 0h do 72h dle předpovědi. V CX-Supervisoru položím dotaz na hodnotu ID a čtu hodnotu slunečního záření a teploty pro zvolený ID. Pro stanovení recordsetu je nutné si vytvořit proměnné (pointy) obrázek č. 50, ideálně se stejným názvem. Recordset je zapsán příkazem SELECT ID, Zareni, Temperature from DATATABLE, kde DATATABLE je název tabulky v databázi.



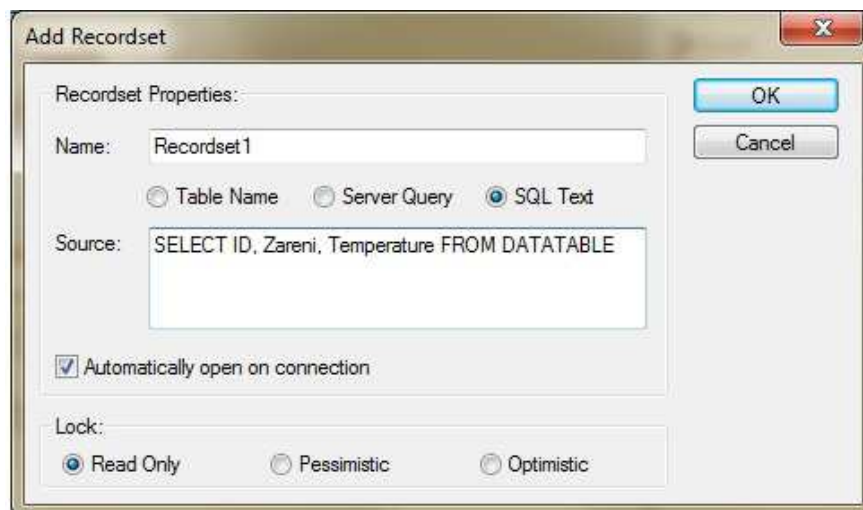
Obr. 46: Workspace



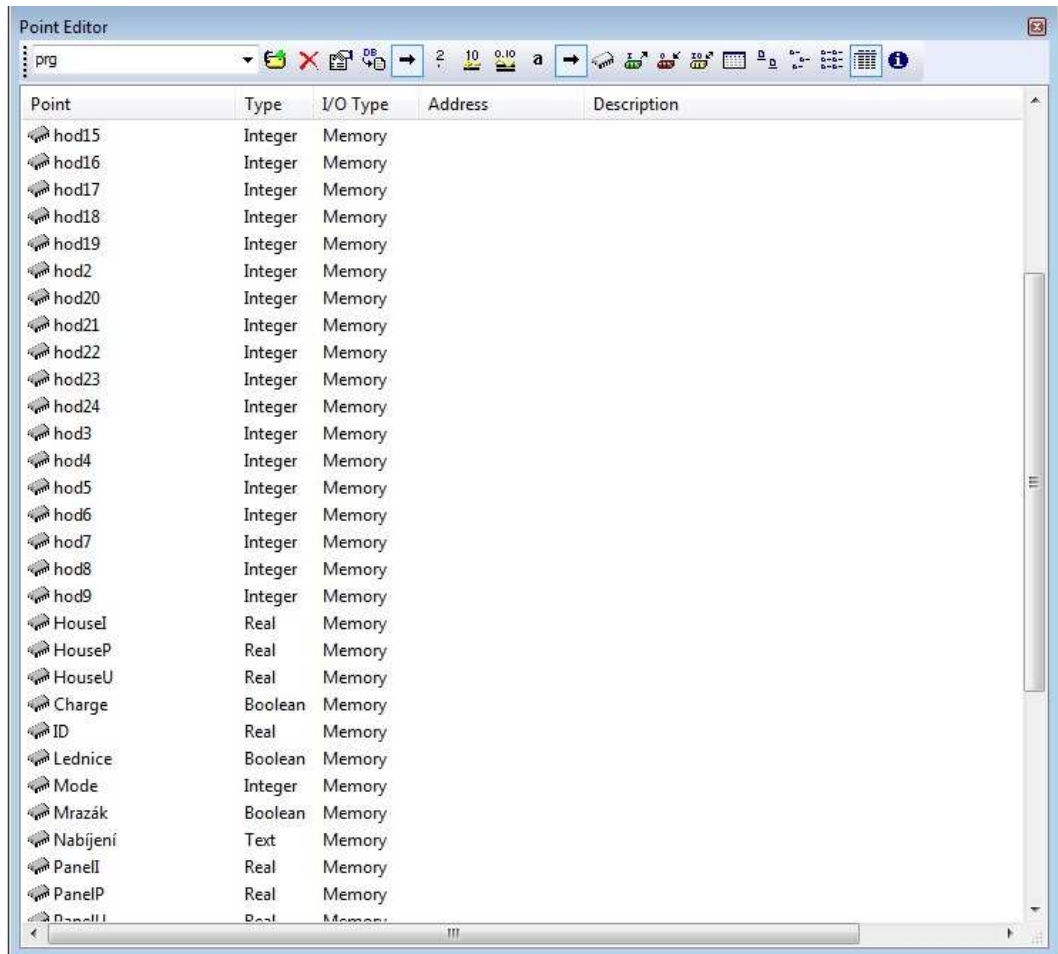
Obr. 47: Propojení s databází



Obr. 48: Výběr databáze



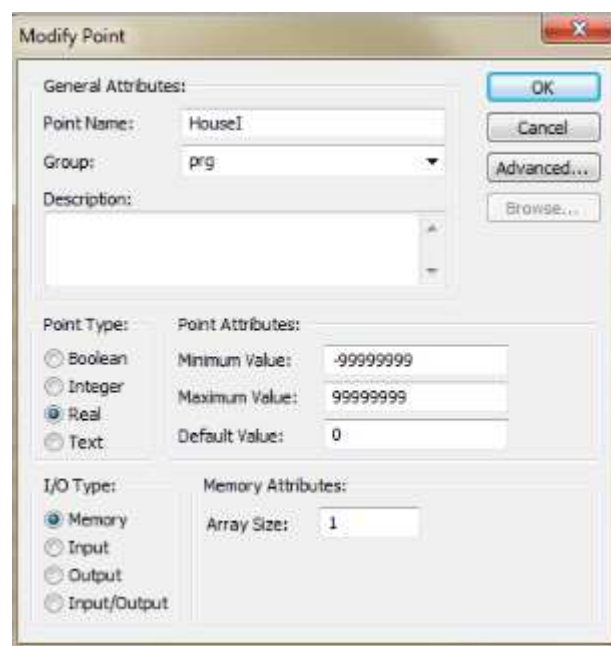
Obr. 49: Přidání recordsetu



The screenshot shows the 'Point Editor' window with a table of variables. The table has five columns: Point, Type, I/O Type, Address, and Description. The variables listed include 'hod15' through 'hod24', 'HouseI', 'HouseP', 'HouseU', 'Charge', 'ID', 'Lednice', 'Mode', 'Mrázák', 'Nabíjení', 'PanelI', and 'PanelP'. Each variable is associated with a specific data type and an I/O type, all of which are 'Memory'.

Point	Type	I/O Type	Address	Description
hod15	Integer	Memory		
hod16	Integer	Memory		
hod17	Integer	Memory		
hod18	Integer	Memory		
hod19	Integer	Memory		
hod2	Integer	Memory		
hod20	Integer	Memory		
hod21	Integer	Memory		
hod22	Integer	Memory		
hod23	Integer	Memory		
hod24	Integer	Memory		
hod3	Integer	Memory		
hod4	Integer	Memory		
hod5	Integer	Memory		
hod6	Integer	Memory		
hod7	Integer	Memory		
hod8	Integer	Memory		
hod9	Integer	Memory		
HouseI	Real	Memory		
HouseP	Real	Memory		
HouseU	Real	Memory		
Charge	Boolean	Memory		
ID	Real	Memory		
Lednice	Boolean	Memory		
Mode	Integer	Memory		
Mrázák	Boolean	Memory		
Nabíjení	Text	Memory		
PanelI	Real	Memory		
PanelP	Real	Memory		
PanelU	Real	Memory		

Obr. 50: Přehled proměnných v point editoru



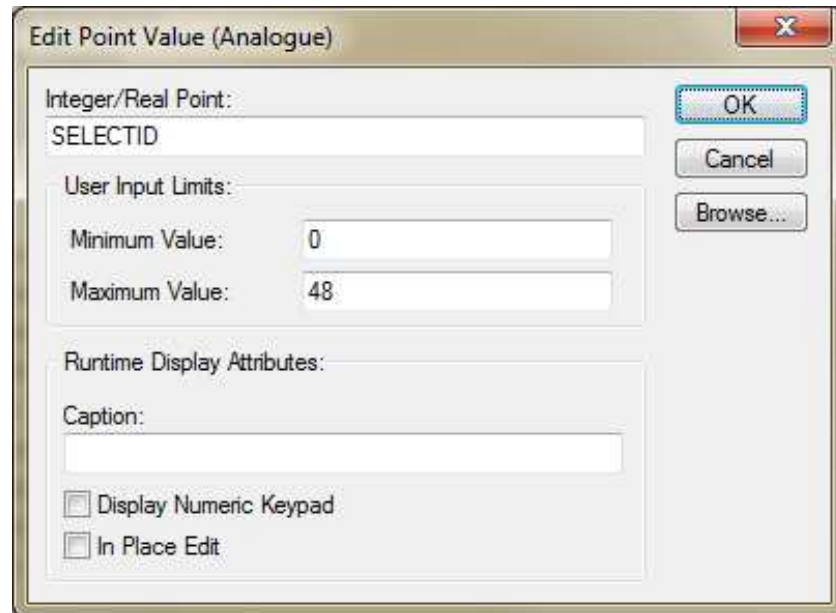
The 'Modify Point' dialog box shows the configuration for the variable 'HouseI'. It includes fields for 'Point Name' (HouseI), 'Group' (prg), and 'Description'. The 'Point Type' is set to 'Real', and the 'I/O Type' is set to 'Memory'. The 'Point Attributes' section shows 'Minimum Value' as -99999999, 'Maximum Value' as 99999999, and 'Default Value' as 0. The 'Memory Attributes' section shows 'Array Size' as 1.

General Attributes:	
Point Name:	HouseI
Group:	prg
Description:	

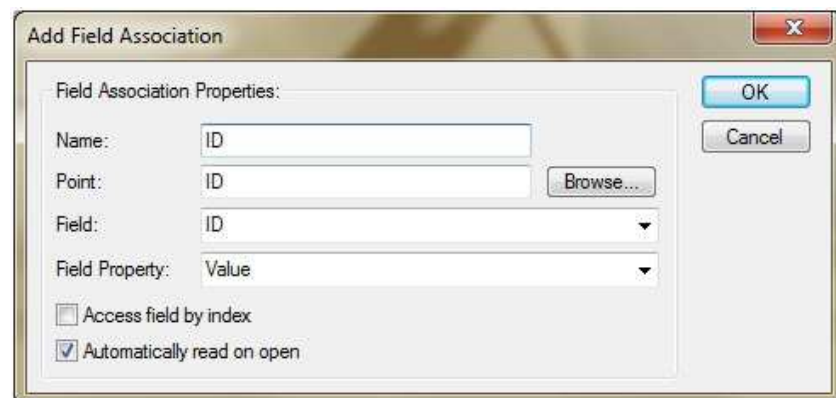
Point Type:	Point Attributes:
<input type="radio"/> Boolean	Minimum Value: -99999999
<input type="radio"/> Integer	Maximum Value: 99999999
<input checked="" type="radio"/> Real	Default Value: 0
<input type="radio"/> Text	

I/O Type:	Memory Attributes:
<input checked="" type="radio"/> Memory	Array Size: 1
<input type="radio"/> Input	
<input type="radio"/> Output	
<input type="radio"/> Input/Output	

Obr. 51: Vlastnosti proměnné House1



Obr. 52: Úprava proměnné ID



Obr. 53: Propojení ID s proměnou ID

3.2.4.4 Komunikační protokol navrženého systému

Komunikace probíhá mezi COM portovou kartou v PC a komunikačním zařízením Xcom-232i. Pro realizaci úspěšného přenosu dat musí být nastavení obou portů shodné. Prvotní nastavení komunikačního portu Xcom-232i, je fixně nastaveno na rychlost 38400bps, 1 startovací bit, 8 datových bitů, 1 paritní bit, sudá parita, 1 stop bit. Pro změnu parametrů je nutné stáhnout software pro úpravu parametrů a následně přes SD kartu parametry nahrát do střídače. Hlavní vlastnosti komunikačního protokolu, jsou popsány dále, zbylé vlastnosti jsou uvedené v technickém listu komunikační jednotky Xcom-232i v příloze číslo 5.

3.2.4.4.1 Zapojení konektoru Xcom-RS232i:

Jedná se o klasický 9 pinový konektor, používaný pro COM port. Popis jednotlivých částí konektoru je vidět na obrázku č. 54.

pin number	usage
1	not connected
2	RxD
3	TxD
4	not connected
5	GND
6	not connected
7	not connected
8	not connected
9	not connected

Obr. 54: Zapojení COM portu komunikátoru Xcom-R232i [36]

3.2.4.5 Komunikační řetězec

Níže je popsána forma komunikačního řetězce, která je využívána k přenosu příkazů z řídicího počítače do střídače a ke čtení hodnot ze střídače. Základ komunikačního řetězce je tvořen směrnicí hexadecimálních znaků ve formátu little endian, popsaných níže na obrázku č. 55. [36]

start_ byte	frame_ flags	src_ addr	dst_ addr	data_ length	header_ checksum	frame_ data	data_ checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes = N	2 bytes	N bytes	2 bytes

Obr. 55: Skladba komunikačního řetězce [36]

- start_byte : Startovací byte řetězce je vždy 0xAA
- frame_flags : Je bytová proměnná složená z osmi bytů, které můžeme konfigurovat dle požadavku.
- src_addr : Je zdrojová adresa o velikosti 32bitů
- dst_addr : Je cílová adresa, opět o velikosti 32bitů
- data_length : Udává délku řetězce, maximální délka je 256 znaků
- header_checksum : Kontrolní součet od frame_flagu po data_length
- frame_data : Data komunikačního řetězce
- data_checksum : Kontrolní součet celého komunikačního řetězce

3.2.4.5.1 Zápis dat

Zápis dat se provádí dle následujícího řetězce složeného z položek na obrázku č. 56. Jednotlivé položky jsou součástí položky frame_data z obrázku č. 55.

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02 or 0x03	1 byte 0x02	2 bytes	4 bytes	2 bytes	0 or 2 bytes

Obr. 56: Řetězec pro zápis dat [36]

service_flags : příznak odpovědi = 1, error= 0 or 1
 service_id : 0x02 adresa pro zápis
 object_type : zadání požadovaného objektu střídač, regulátor napětí
 object_id : zadání požadovaného parametru pro objekt
 property_id : zadání vlastnosti požadovaného parametru
 property_data : Pokud hodnota service_flag=0 pak property_data má velikost 0 bytů, pokud ne jsou chyby zaznamenávány do proměnné o velikosti 2byty.

3.2.4.5.2 Čtení dat

Čtení dat se provádí dle následujícího řetězce složeného z položek na obrázku č. 57. Jednotlivé položky jsou součástí položky frame_data z obrázku č. 55.

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02 or 0x03	1 byte 0x01	2 bytes	4 bytes	2 bytes	N bytes or 2 bytes

Obr. 57: Řetězec pro čtení dat[36]

service_flags : příznak odpovědi = 1, error= 0 or 1
 service_id : 0x01 adresa pro čtení
 object_type : zadání požadovaného objektu střídač, regulátor napětí
 object_id : zadání požadovaného parametru pro objekt
 property_id : zadání vlastnosti požadovaného parametru
 property_data : Pokud hodnota service_flag=0 pak property_data má velikost 0-bytů, pokud ne jsou chyby v proměnné o velikosti 2bytů.

3.2.4.6 Algoritmus řízení navrženého systému

Jak již bylo zmíněno program funguje na principu podmínek ve smyčce, která se neustále opakuje. Přepínačem se zvolí režim. Všechny tyto režimy vychází, ze stejného základu podmínek, které se ve skriptu opakují, nicméně podmínky pro každý režim jsou odlišné.

Pro přehled zde uvedu a popíši nejdůležitější části programu. Nebudu zde uvádět celý program z důvodu rozsáhlosti. Celý program je přiložen v příloze č. 6.

-
- Příkaz pro nastavení, propojení a otevření databáze

```
DBOpen( "Connection1" )  
DBOpen( "Connection1.Recordset1" )
```

- Zde je část programu, která se stará o výběr dat z databáze na základě zadání hodin pro předpověď. Program si na základě zadání hodin vybere z databáze pod políčkem id řádek, ze kterého se přečte hodnota záření a teploty pro danou hodinu

```
id=$hour + Tpredpoved  
SQLDotaz = "id, Zareni, Teplota FROM DATATABLE where ID = " + id  
DBExecute( "Connection1.Recordset1", "Source", SQLDotaz)  
DBExecute( "Connection1.Recordset1", "Requery" )
```

- Zde je část zabezpečující nastavené komunikačního portu RS232

```
SetupCOMPort(1,set)  
baudrate=38400  
startbit=1  
bytesize=8  
paritybyte=1  
parity= even  
stopbits=1
```

- Níže jsou uvedeny příkazy pro čtení hodnot napětí a proudu baterie, střídavého proudu střídače a výkonu střídače.

```
OpenCOMPort1(1, Open)
```

```
Napeti_baterie='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xB8\x0B\x00\x00\x01\x00\xC5\x90'
```

```
InputCOMPort(1,write, Status)
```

```
Napeti_bat=readCOMPort(1,read)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Proud_baterie='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xBD\x0B\x00\x00\x01\x00\xCA\xAE'
```

```
InputCOMPort(1,write, Status)
```

```
Proud_bat=readCOMPort(1,read)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Vstupni_proudAC='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xC4\x0B\x00\x00\x01\x00\xD1\xD8'
```

```
InputCOMPort(1,write, Status)
```

```
ProudAC_vst=readCOMPort(1,read)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Vykon_stridac='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xCF\x0B\x00\x00\x01\x00\xDC\x1A'
```

```
InputCOMPort(1,write, Status)
```

```
Pstrid=readCOMPort(1,read)
```

```
CloseCOMPort(1,close)
```

-
- Příkaz pro připojení stříd sítě. V případě potřeby dojde k připojení střídavé sítě. Tento příkaz urychlí rozhodování pro funkci smartboost.

```
OpenCOMPort1(1, Open)
```

```
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
```

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

- Příkaz pro zapnutí střídače, dojde k zapnutí střídače na základě níže napsaného příkazu

OpenCOMPort1(1, Open)

Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

- Příkaz pro zapnutí funkce smartboost, pomocí které si střídač sám na základě zabudovaného algoritmu řídí nejefektivnější využití energie ze solárních panelů, baterie a střídavé sítě.

OpenCOMPort1(1, Open)

Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

- Příkaz pro nabíjení baterie se změnou velikosti nabíjecího proudu

OpenCOMPort1(1, Open)

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

OpenCOMPort1(1, Open)

If ProudAC_vst >= 2 AND <3

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

```
Elseif ProudAC_vst >= 3 AND <4
    OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x
02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
    InputCOMPort(1,write, Status)
    CloseCOMPort(1,close)

Elseif ProudAC_vst >= 4 AND <5
    OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x
02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
    InputCOMPort(1,write, Status)
    CloseCOMPort(1,close)

Elseif ProudAC_vst >= 5 AND <6
    OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x
02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
    InputCOMPort(1,write, Status)
    CloseCOMPort(1,close)

Elseif ProudAC_vst >= 6 AND <7
    OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x
02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
    InputCOMPort(1,write, Status)
    CloseCOMPort(1,close)
End If
```

-
- Příkaz ovládající vizualizaci měsíce či slunce na základě hodin

```
IF &hour>="6" AND &minute>="0" AND &hour<="19" AND &minute<="59"
```

```
SUN=TRUE
```

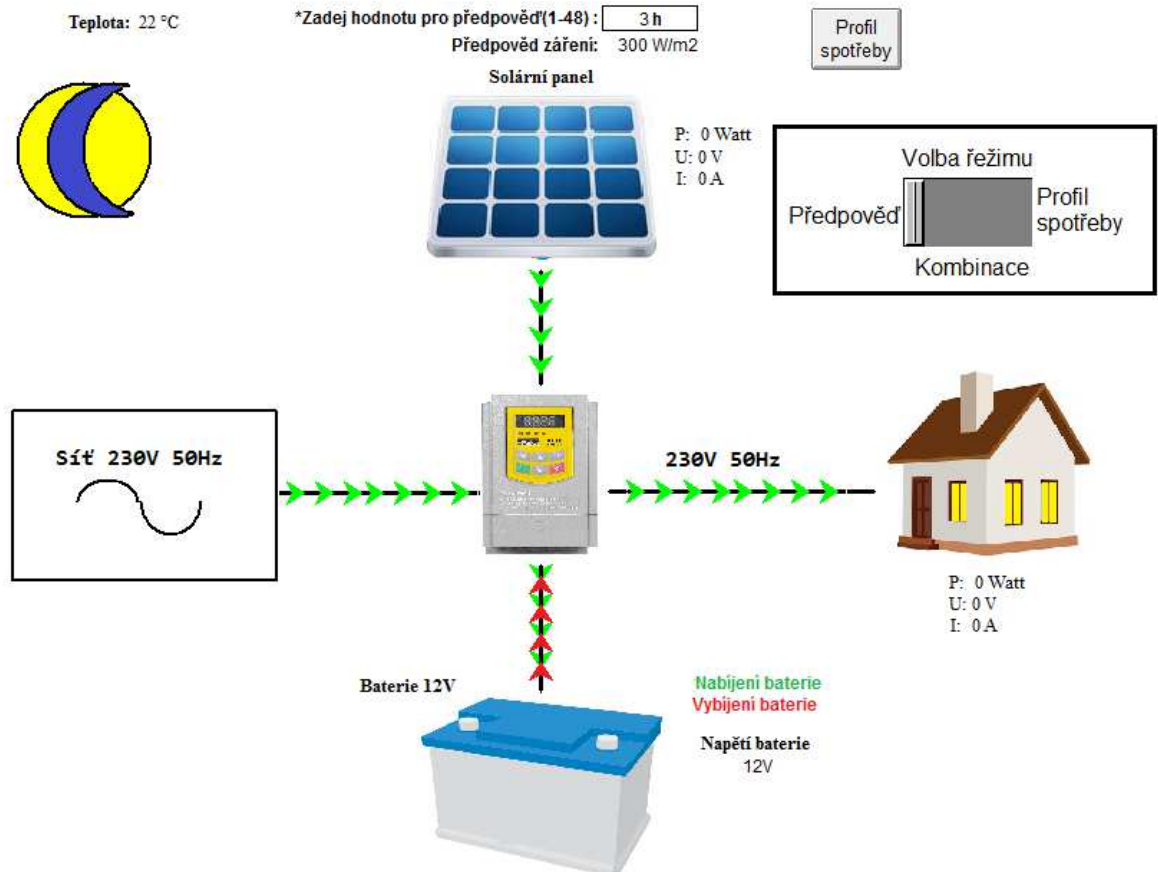
```
MOON=FALSE
```

```
ELSEIF &hour>="20" AND &minute>="0" AND &hour<="5"AND &minute<="59"  
SUN=FALSE  
MOON=TRUE  
END IF
```

3.2.4.7 Vizualizace navrženého systému

V programu je zakomponovaná velice jednoduchá vizualizace, která je na obrázku č. 58, Ve vizualizaci je zobrazení aktuální času, teploty, s detekcí dne a noci, velikostí výkonů, napětí a proudů, stavu baterie, zdroje v podobě připojené střídavé sítě, slunečního záření. Toky energií jsou zobrazeny pomocí šipek. Odběr domu signalizují rozsvícená okna. Rozlišení dne a noci, se děje automaticky dle hodin, zobrazí se symbol slunce či měsíce.

Ovládací prostředí, které vidí uživatel, je na obrázku č. 45. Prostředí s veškerou vizualizací je na obrázku č. 58. Čtvereček se značkou vlnovky a šipek představuje připojení střídavé sítě. V případě provozu solárního systému dojde k zobrazení šipek směřujících z FV panelu do střídače. Zobrazení ostatních věcí je závislé na algoritmu, na základě kterého dojde ve vizualizaci k zobrazení toku energie od střídače k baterii či naopak a od střídače do objektu s následným rozsvícením oken v případě odběru. Symbol slunce a měsíce je zobrazen na základě aktuálního času. V rozmezí hodin od 6:00 do 20:00, je zobrazeno slunce od 20:00 do 6:00, je zobrazen měsíc. Zbylá vizualizace výkonů, napětí a proudů je zobrazena na základě propojení měřicího systému střídače a jednotlivých proměnných. Vizualizační prvky jsou zobrazovány pomocí booleovských proměnných buď v režimu true=viditelné prvky a false=neviditelné prvky.



*Hodiny pro předpověď jsou počítané od aktuálního času. Například je 12:32 chci předpověď 3 hodiny napřed, zadám tedy 3. Algoritmus vezme hodnotu aktuální hodiny 12 a přičte k ní 3, výsledek je 15. Předpověď je tedy brána pro 15:00. Obdobně je to pro teplotu.

Obr. 58: Vzhled vizualizačního prostředí SCADA systému

3.2.4.7.1 Zobrazení vizualizace pomocí šipek

BatteryState = "Nabíjení"; zobrazení stavu nabíjení baterie

BatteryState = "Vybíjení"; zobrazení stavu vybíjení baterie

SipkyBS= TRUE/FALSE ; zobrazení šipek baterie -> střídač

SipkySB= TRUE/FALSE ; zobrazení šipek střídač -> baterie

SipkySD= TRUE/FALSE ; zobrazení šipek střídač -> dům

SipkyPS= TRUE/FALSE; zobrazení šipek panel -> střídač

Dum= TRUE/FALSE ; zobrazení vizualizace oken u domu

SipkyAC=TRUE/FALSE ; zobrazení vizualizace připojení sítě (čtvereček vlnovka a šipky)

4 Závěr

Inteligentní řízení pro FVE bylo vypracováno v programu CX-Supervisor od výrobce OMRON, tento program jsem volil vzhledem k dostupnosti licence u mého zaměstnavatele. V rámci návrhu inteligentního systému bylo současně nutné řešit návrh FVE, kterou by inteligentní systém řídil a monitoroval. Podklad pro návrh FVE byl vypracován v programu Energie 2016 od společnosti K-CAD. Vzhledem k tomu, že primárním úkolem bylo navrhnout inteligentní řídicí systém, je návrhu FVE věnován jen stručný popis nejdůležitějších součástí včetně popisu budovy, tvorby profilu spotřeby a jednotlivých prvků navrženého FV systému zahrnujícího přehledového schéma s teoretickým popisem zapojení elektroinstalace.

Při návrhu inteligentního systému řízení jsem řešil problematiku sběru dat slunečního záření a venkovních teplot. Data jsou získávány na základě html dotazu ze stránek PVforecast v kapitole 3.2.2 a jsou následně ukládána do databáze sq.lite, pomocí skriptu v kapitole 3.2.3.2, který je každých šest hodin aktualizuje. Čtení uložených hodnot a práce s databází již probíhá interně v programu CX-Supervisor na základě uvedených příkazů v kapitole 3.2.4.6. Navržený algoritmus pracuje s příkazy, které zasílá střídači skrze komunikační zařízení uvedené v kapitole 3.2.1.4 s využitím komunikačního protokolu z kapitoly 3.2.4.4.

Algoritmus inteligentního řízení pracuje v nekonečné smyčce, která se obnovuje každou vteřinu. Ve skriptu, který je uveden v příloze č. 6 dochází na základě vhodných podmínek k porovnávání hodnot výkonů, napětí, slunečního záření a hodnot z profilu spotřeby, kde výsledkem je balík dat, složený z příkazů a zasílaných střídači spolu se zobrazením aktuální situace v podobě vizualizace. Popis používaných příkazů, které jsou zasílané střídači, jsou uvedeny v kapitole 3.2.4.6. Skladbu příkazů vhodných k ovládní měniče jsem sestavoval na základě konzultace s panem Janem Schröpferem, který již podobné projekty realizoval. V návrhu FVE jsou použity výhradně výrobky švýcarského výrobce Studer-innotech z důvodu řízení navrženého systému. Ostatní výrobci ovládní vnějšími příkazy nepodporují vůbec, nebo jen velice omezeně a to ve formě monitorování systému.

Seznam literatury a informačních zdrojů

- [1] Quaschnig . V.: Obnovitelné zdroje energií, Grada Publishing, a.s, Praha, 2008
- [2] Fotovoltaické systémy [online]. 2017 [cit. 2017-02-12]. Dostupné z:
<http://www.cne.cz/fotovoltaicke-systemy/uvod-do-fv-systemu>
- [3] Polovodiče [online]. 2017 [cit. 2017-03-12]. Dostupné z:
https://spscl.cz/public/MatFyz/Soubory/Fyzika/14_proud_v_polovodicich/polovodice.htm
- [4] Měření na fotovoltaickém článku [online]. 2017 [cit. 2017-04-12]. Dostupné z:
http://fei1.vsb.cz/kat420/vyuka/Bakalarske/STB/4_LU_fotoclanek.pdf
- [5] Monokrystalické panely [online]. 2017 [cit. 2017-05-12]. Dostupné z:
<http://eshop.terms.eu/cz/e-shop/c68830/monokrystalicke-panely.html>
- [6] Jaký je rozdíl mezi monokrystalickým a polykrystalickým panelem [online]. 2018 [cit. 2018-05-12]. Dostupné z: https://www.joyce-energie.cz/files/joyce-energie.cz/files/Mono_poly_rozdl.pdf
- [7] Polykrystalické panely [online]. 2017 [cit. 2017-05-12]. Dostupné z:
<http://eshop.terms.eu/cz/e-shop/c68829/polykrystalicke-panely.html>
- [8] Solární panel VICTRON ENERGY [online]. 2017 [cit. 2017-07-12]. Dostupné z:
<http://eshop.terms.eu/cz/e-shop/1245483/c68829-polykrystalicke-panely/solarni-panel-spanvictron-energy-bluesolar-spp190-16v-span.html>
- [9] Tenkovrstvé panely [online]. 2017 [cit. 2017-07-12]. Dostupné z:
<http://eshop.terms.eu/cz/e-shop/c68831/tenkovrstve-panely.html>
- [10] Amorfnní panely [online]. 2017 [cit. 2017-09-12]. Dostupné z:
<https://shop.iftech.cz/15-amorfni-panely>

- [11] Fotovoltaika [online]. 2017 [cit. 2017-09-12]. Dostupné z:
<http://rccv.vsb.cz/Island/docs/Fotovoltaika.pdf>
- [12] Rozdělení solárních systémů [online]. 2017 [cit. 2017-011-12]. Dostupné z:
<http://www.obnovitelne-energie.cz/solarni-systemy.php>
- [13] Aktivní a pasivní solární systém [online]. 2017 [cit. 2017-012-12]. Dostupné z:
<https://www.lepebydlet.cz/aktivni-pasivni-solarni-system>
- [14] Mareček, Jan. NÁVRH SYSTÉMU FOTOVOLTAICKÝCH PANELŮ PRO RODINNÝ DŮM [online]. Brno, 2013 [cit. 2017-14-12]. Dostupné z:
<https://www.vutbr.cz/>. Bakalářská práce. Vysoké učení technické v Brně.
- [15] Chytrá elektroinstalace [online]. 2017 [cit. 2017-015-12]. Dostupné z:
<http://www.chytrainstalace.cz/chytra-elektroinstalace/>
- [16] Vantage [online]. 2017 [cit. 2017-015-12]. Dostupné z:
<http://www.vantagecontrols.com>
- [17] Kruinc [online]. 2017 [cit. 2017-017-12]. Dostupné z:
<https://kruinc.com/?page=systems-vantage-controls-home-automation>
- [18] Quantum Systém Design Ltd. [online]. 2017 [cit. 2017-017-12]. Dostupné z:
<http://www.quantumav.net/automated-lighting.html>
- [19] Habitech [online]. 2017 [cit. 2017-017-12]. Dostupné z:
http://www.habitech.co.uk/Price+List/Vantage/Control+Systems/Vantage+control+interface+for+KNX+building+management+system_LEG-KNXINTFPOINT.htm
- [20] KLABAN, Jaromír. Připojování technických zařízení budov pod řídicí systém Tecomat Foxtrot [online]. 2017 [cit. 2017-018-12]. Dostupné z:
<http://docplayer.cz/3890653-Pripojovani-technickyh-zarizeni-budov-pod-ridici-system-tecomat-foxtrot.html>

- [21] Bticino [online]. 2017 [cit. 2017-018-12]. Dostupné z:
<https://bticino.assetbank-server.com/bticino/assetfile/72488.pdf>
- [22] Legrand [online]. 2017 [cit. 2017-018-12]. Dostupné z:
<https://legrand.hu>
- [23] České slunce [online]. 2017 [cit. 2017-019-12]. Dostupné z:
<https://www.ceskeslunce.cz/solarni-mapa.html>
- [24] AKUMULÁTOR Hoppecke [online]. 2018 [cit. 2018-015-02]. Dostupné z:
<https://www.obchodsolar.cz/hoppecke/solar-bloc-250Ah>
- [25] METEOROLOGIE [online]. 2017 [cit. 2017-022-12]. Dostupné z:
<http://edu.techmania.cz/cs/encyklopedie/fyzika/meteorologie>
- [26] VRTEK, Mojmír. Sluneční energie [online]. Ostrava, 2012 [cit. 2017-025-12].
Dostupné z:
https://kke.zcu.cz/export/sites/kke/old_web/files/projekty/enazp/21/IUT/140_Slunecni_energie_-_Vrtek_-_P3.pdf. Článek k odbornému vzdělávání. Vysoká škola báňská v Ostravě.
- [27] Procházka, Martin. Predikce krátkovlnného slunečního záření pro malé fotovoltaické systémy [online]. Praha, 2016 [cit. 2017-27-12]. Dostupné z: <https://dspace.cvut.cz/>.
Bakalářská práce. České vysoké učení technické v Praze.
- [28] Medard [online]. 2018 [cit. 2018-02-01]. Dostupné z:
<http://www.medard-online.cz/>
- [29] Šrahůlková, Karolína. PŘÍSPĚVEK KE STUDIU KLIMATU MĚSTA OLOMOUCE – SLUNEČNÍ ZÁŘENÍ [online]. Olomouc, 2010 [cit. 2018-05-01]. Dostupné z: <https://geography.upol.cz/>.
Bakalářská práce. Univerzita Palackého v Olomouci.

- [30] Hnilica, Jan. Software pro interpolaci hodnot slunečního záření [online]. Praha, 2010 [cit. 2018-05-01]. Dostupné z: <https://anzdoc.com/software-pro-interpolaci-hodnot-sluneniho-zaeni.html>. Diplomová práce. Česká zemědělská univerzita v Praze.
- [31] PVFORECAST [online]. 2018 [cit. 2018-09-01]. Dostupné z: <http://www.pvforecast.cz/>
- [32] OMRON [online]. 2018 [cit. 2018-010-01]. Dostupné z: <https://industrial.omron.cz/cs/products/cx-supervisor>
- [33] Měníče DC/AC [online]. 2018 [cit. 2018-015-01]. Dostupné z: <http://mve.energetika.cz/sikovneruce/menice.htm>
- [34] Darebný, Tomáš. NÁVRH FOTOVOLTAICKÉHO SYSTÉMU RODINNÉHO DOMU [online]. Brno, 2017 [cit. 2018-20-01]. Dostupné z: <https://dspace.vutbr.cz/>. Diplomová práce. Vysoké učení technické v Brně.
- [35] Panasonic [online]. 2018 [cit. 2018-02-02]. Dostupné z: <https://eu-solar.panasonic.net/en/solar-panel-vbhn240sj25-vbhn245sj25.htm>
- [36] Studer innotec [online]. 2018 [cit. 2018-06-02]. Dostupné z: <http://www.studer-innotec.com/en/#>

Seznam Příloh

- Příloha č.1: Technická specifikace střídače
- Příloha č.2: Technická specifikace regulátoru napětí
- Příloha č.3: Technická specifikace solárního panelu
- Příloha č.4: Technická specifikace akumulátoru
- Příloha č.5: Technická specifikace komunikačního protokolu
- Příloha č.6: Navržený algoritmus řízení

Příloha č. 1: Technická specifikace střídače

Xtender XTH



Model	XTH 3000-12	XTH 5000-24	XTH 6000-48	XTH 8000-48
Inverter				
Nominal battery voltage	12 Vdc	24 Vdc	48 Vdc	
Input voltage range	9.5 - 17 Vdc	19 - 34 Vdc	38 - 68 Vdc	
Continuous power @ 25°C	2500 VA	4500 VA	5000 VA	7000 VA
Power 30 min. @ 25°C	3000 VA	5000 VA	6000 VA	8000 VA
Power 5 sec. @ 25°C	7.5 kVA	12 kVA	15 kVA	21 kVA
Maximum load	Up to short-circuit			
Asymmetric load	Up to Pcont.			
* Load detection (stand-by)	2 to 25 W			
Cos φ	0.1-1			
Maximum efficiency	93 %	94 %	96 %	
Consumption OFF/Stand-by/ON	1.2 W / 1.4 W / 14 W	1.4 W / 1.8 W / 18 W	1.8 W / 2.2 W / 22 W	1.8 W / 2.4 W / 30 W
* Output voltage	Pure sine wave 230 Vac (± 2 %) / 120 Vac ⁽¹⁾			
* Output frequency	Adjustable 45 - 60 Hz ⁽¹⁾ ± 0.05 % (crystal controlled)			
Harmonic distortion	< 2 %			
Overload and short-circuit protection	Automatic disconnection with 3 time restart attempt			
Overheat protection	Warning before shut-off - with automatic restart			
Battery charger				
* Charge Characteristic	6 steps: Bulk, Absorption, Floating, Equalization, reduced floating, periodic absorption Number of steps, thresholds, end current and times completely adjustable with the RCC-02/-03			
* Maximum charging current	160 A	140 A	100 A	120 A
* Temperature compensation	With BTS-01 or BSP 500/1200			
Power Factor Correction (PFC)	EN 61000-3-2			
General data				
* Input voltage range	150 to 265 Vac / 50 to 140 Vac ⁽¹⁾			
Input frequency	45 to 65 Hz			
Input current max. (transfer relay) / Output current max.	50 Aac / 56 Aac			50 Aac / 80 Aac
Transfer time	< 15 ms			
Multifunction contacts	2 independent contacts (potential free 3 points, 16 Aac / 5 Adc)			
Weight	34 kg	40 kg	42 kg	46 kg
Dimensions h/w/l [mm]	230 / 300 / 500	230 / 300 / 500	230 / 300 / 500	
Protection index	IP20			
EU declaration of conformity	Low Voltage Directive (LVD) 2014/35/EU: - EN 50178:1997 Electromagnetic Compliance (EMC) Directive 2014/30/EU: - EN 62040-2:2006, EN 61000-3-2:2014, EN 61000-3-12:2011			
Operating temperature range	-20 to 55°C			
Relative humidity in operation	95 % without condensation			
Ventilation	Forced from 55°C			
Acoustic level	<40 dB / <45 dB (without/with ventilation)			
Warranty	5 years			
ISO Certification	9001:2008 / 14001:2004			
Accessories				
Remote control RCC-02 or RCC-03	•	•	•	•
Module Xcom-232i	•	•	•	•
Communication sets Xcom-LAN /w Xcom-GSM / Xcom-SMS	•	•	•	•
Battery Status Processor BSP	•	•	•	•
Remote Control Module RCM-10 (3 m)				
2 aux. contacts module ARM-02				
Cooling Module ECF-01				
Battery temp. sensor BTS-01 (3 m)	•	•	•	•
Communication cable for 3ph and // CAB-RJ45-8-2	•	•	•	•
Mounting frame X-Connect	•	•	•	•

* Adjustable with the RCC-02/-03

** These features are valid only when using the cooling module ECF-01.

⁽¹⁾ With -01 at the end of the reference, means 120V/60Hz. Available for all Xtenders except XTH 8000-48

Data may change without any notice

Příloha č. 2: Technická specifikace regulátoru napětí

VarioTrack

VT-80



Model	VT-80		
Electrical characteristics PV array side			
At nominal battery voltage	12 V	24 V	48 V
Maximum solar power recommended (@STC)	1250 W	2500 W	5000 W
Maximum solar open circuit voltage	75 V	150 V	
Maximum solar functional circuit voltage	75 V	145 V	
Minimum solar functional circuit voltage	Above battery voltage		
Electrical characteristics battery side			
Maximum output current	80 A		
Nominal battery voltages	Automatic / manual set to 12, 24 or 48 V		
Operating voltage range	7 - 68 V		
Performances of the device			
Power conversion efficiency (in a 48 V typical-system)	> 99 %		
Maximum stand-by self-consumption (48 V)	25 mA > 1.2 W		
Maximum stand-by self-consumption (24 V)	30 mA > 0.8 W		
Maximum stand-by self-consumption (12 V)	35 mA > 0.5 W		
Charging stages	4 stages: Bulk, Absorption, Float, Equalization		
Battery temperature compensation (available with accessory BTS-01)	-3 mV /°C / cell (25°C ref) default value adjustable -8 to 0 mV /°C		
Electronic protections			
PV reverse polarity	Up to -150 V		
Battery reverse polarity	Up to -150 V		
Battery overvoltage	Up to 150 V		
Over temperature	Protected		
Reverse current at night	Prevented by relays		
Environment			
Operating ambient temperature range	-20 to 55°C		
Humidity	100 %		
Ingress protection of enclosures	IP54, IEC/EN 60529:2001		
Mounting location	indoor		
General data			
Warranty	5 years		
ISO Certification	9001:2008 / 14001:2004		
Weight	5.5 kg		
Dimensions h/w/l [mm]	120 / 220 / 350		
Parallel operation (separated PV arrays)	Up to 15 devices		
Max wire size	35 mm ²		
Glands	M 20 x 1,5		
Communication			
Network cabling	STUDER communication BUS		
Remote control & Communication	RCC-02/03, Xcom-232i / Xcom-LAN / Xcom-GSM / Xcom-SMS		
Menu languages	English / French / German / Spanish		
Data logging	With RCC-02/03, Xcom-232i on SD card · One point every minute		
Accordance to standards			
EU declaration of conformity	Low Voltage Directive (LVD) 2014/35/EU: - EN 50178:1997 Electromagnetic Compliance (EMC) Directive 2014/30/EU: - EN 61000-6-2:2005, - EN 61000-6-4:2007/A1:2011		
Accessories			
Remote control RCC-02 or RCC-03	•		
Module Xcom-232i	•		
Communication sets Xcom-LAN / Xcom-GSM / Xcom-SMS	•		
Battery Status Processor BSP	•		
2 aux. contacts module ARM-02	•		
Cooling Module ECF-01	Included		
Battery temp. sensor BTS-01 (3 m)	•		
Communication cable CAB-RJ45-8-2	•		

Příloha č. 3: Technická specifikace solárního panelu

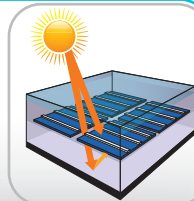


Water drainage frame

- Rain water is drained off the module surface.
- This avoids not only water accumulation, but also water stains after drying.
- Even in low-angle installations, water drainage corners keep the module clean.

Power from both sides

- HIT[®] cells generate solar electricity simultaneously on the front and on the back side.
- This additional amount of light is combined with the light taken up by the front side of the module.



19.4%*
194 W/m²



Vertically integrated factory

- Efficient production flow improves product quality as entire process from wafer to cell is done at the same location.
- No risk of damage of individual components during transportation between factories.

* For N245

HIT[®] cell technology

The HIT[®] solar cell is made of a thin monocrystalline silicon wafer surrounded by ultra-thin amorphous silicon layers. This product offers the industry's leading performance and value, using state-of-the-art manufacturing techniques. The development of the HIT[®] solar cell was supported in part by the New Energy and Industrial Technology Development Organization (NEDO).

Quality

Panasonic is truly committed to quality since it began developing and manufacturing solar PV technology in 1975. Our long track record is supported by our claim-rate of only 0.0036% in our European factory in Dorog, Hungary (as of September 2013).

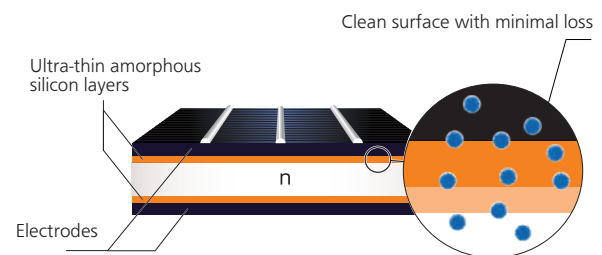
Special features

HIT[®] solar modules are 100% emission free, have no moving parts and produce no noise. The dimensions of the HIT[®] modules enable a space saving installation and the achievement of maximum output power possible on a given roof area.

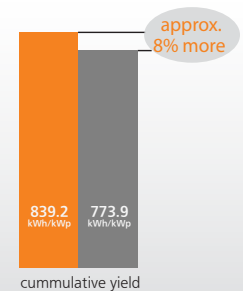
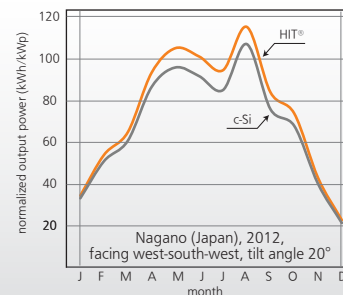
High performance at high temperatures

With its very low temperature coefficient of only -0.29%/°C, the HIT[®] solar cell can maintain a higher efficiency than a conventional crystalline silicon solar cell, even at high temperatures.

HIT[®] solar cell structure



Yield comparison



HIT[®]
Photovoltaic Module

*HIT[®] is a registered trademark of Panasonic Group.

Model	Cell efficiency	Module efficiency	Output/m ²
N245	22.0%	19.4%	194 W/m ²
N240	21.6%	19.0%	190 W/m ²

Electrical data (at STC)

	VBHN245SJ25	VBHN240SJ25
Max. power (Pmax) [W]	245	240
Max. power voltage (Vmp) [V]	44.3	43.6
Max. power current (Imp) [A]	5.54	5.51
Open circuit voltage (Voc) [V]	53.0	52.4
Short circuit current (Isc) [A]	5.86	5.85
Max. over current rating [A]	15	
Production tolerance power [%]	+10/-5*	
Max. system voltage [V]	1000	

Note: Standard Test Conditions: Air mass 1.5; Irradiance = 1000W/m²; cell temp. 25°C
 * All modules measured by Panasonic facilities have an output with positive tolerance.

Temperature characteristics

	VBHN245SJ25	VBHN240SJ25
Temperature (NOCT) [°C]	44.0	44.0
Temp. coefficient of Pmax [%/°C]	-0.29	-0.29
Temp. coefficient of Voc [V/°C]	-0.133	-0.131
Temp. coefficient of Isc [mA/°C]	1.76	1.76

At NOCT (Normal Operating Conditions)

	VBHN245SJ25	VBHN240SJ25
Max. power (Pmax) [W]	187.4	183.2
Max. power voltage (Vmp) [V]	42.5	41.7
Max. power current (Imp) [A]	4.41	4.39
Open circuit voltage (Voc) [V]	50.3	49.7
Short circuit current (Isc) [A]	4.71	4.71

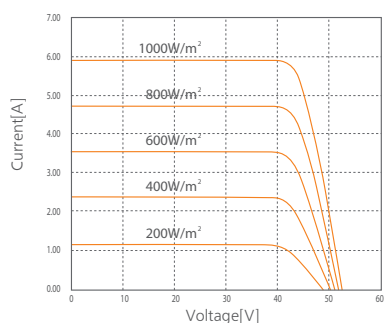
Note: Nominal Operating Cell Temp.: Air mass 1.5; Irradiance = 800W/m²;
 Air temperature 20°C, wind speed 1 m/s

At low irradiance (20%)

	VBHN245SJ25	VBHN240SJ25
Max. power (Pmax) [W]	47.0	45.9
Max. power voltage (Vmp) [V]	43.2	42.2
Max. power current (Imp) [A]	1.09	1.09
Open circuit voltage (Voc) [V]	49.6	49.0
Short circuit current (Isc) [A]	1.17	1.17

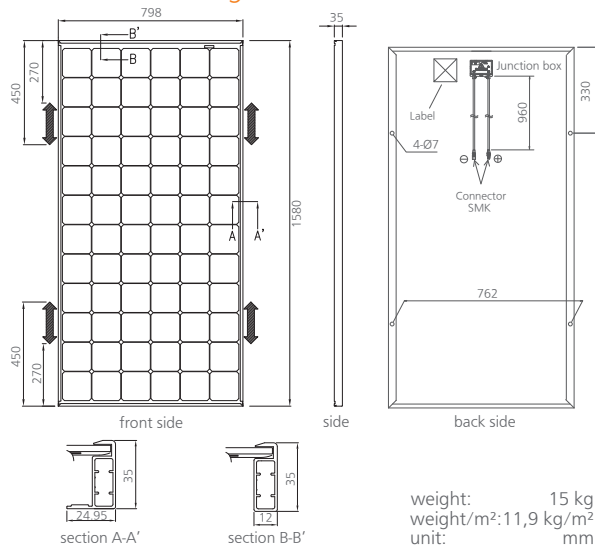
Note: Low irradiance: Air mass 1.5; Irradiance = 200W/m²; cell temp. = 25°C

Dependence on irradiance



Reference data for model VBHN245SJ25 (Cell temperature: 25°C)

Dimensions and weight



weight: 15 kg
 weight/m²: 11,9 kg/m²
 unit: mm

Guarantee

Power output: 10 years (90% of Pmin), 25 years (80% of Pmin)
 Product workmanship: 10 years (based on guarantee document)

Materials

Cell material: 5 inch HIT® cells
 Glass material: AR coated tempered glass
 Frame materials: Black anodized aluminium
 Connectors type: SMK

Certificates



IEC61215
 IEC61730-1
 IEC61730-2



Member of



Please consult your local dealer for more information.

CAUTION! Please read the installation manual carefully before using the products.

Used electrical and electronic products must not be mixed with general household waste. For proper treatment, recovery and recycling of old products, please take them to applicable collection points in accordance with your national legislation.



Panasonic Eco Solutions Energy Management Europe
 SANYO Component Europe GmbH

Stahlgruberring 4
 81829 Munich, Germany
 Tel +49-(0)89-460095-0
 Fax +49-(0)89-460095-170
<http://www.eu-solar.panasonic.net>
info.solar@eu.panasonic.com

All Rights Reserved © 2014 COPYRIGHT SANYO Component Europe GmbH
 Specifications are subject to change without notice.

01/2014

Příloha č. 4: Technická specifikace akumulátoru



sun | power VR M

Valve regulated lead acid batteries for cyclic applications

Typical applications:

- Solar home storage systems
- Street lighting
- Medical care facilities
- Signalling systems
- Leisure applications

Your benefits:

- Maintenance-free monobloc battery - due to Absorbent Glass Mat-technology
- Optimized cycle stability - due to improved electrode design for efficiently charge current acceptance
- Optimum operational safety - integrated backfire protection
- Higher short-circuit safety even during the installation - based on HOPPECKE system connectors

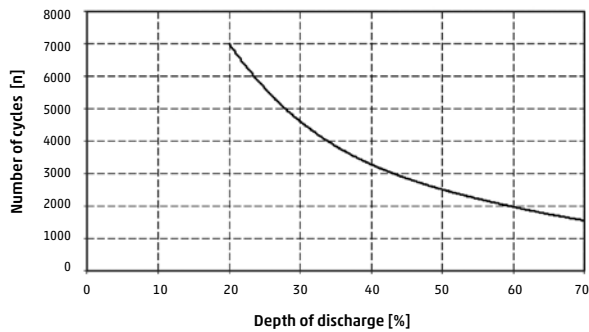
Type overview **sun | power VR M**

Capacities, dimensions and weights

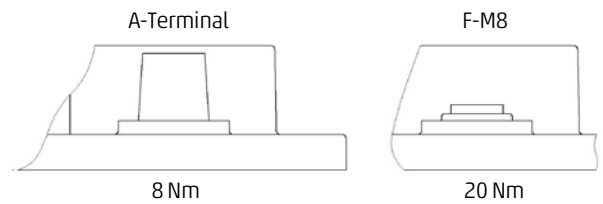
Type	Nominal Voltage V	C ₁₀₀ /1.85 V Ah	C ₅₀ /1.80 V Ah	C ₂₄ /1.80 V Ah	C ₁₀ /1.80 V Ah	Length L mm	Width W mm	Height H mm	Weight kg	Connection	Handle	Terminal layout
sun power VR M 12-58	12	56	58	56	48	232	177	190	19.0	A-Terminal	yes	B
sun power VR M 12-70	12	69	71	68	58	267	177	190	23.0	A-Terminal	yes	B
sun power VR M 12-80	12	76	78	74	66	303	177	190	24.0	A-Terminal	yes	B
sun power VR M 12-90	12	88	89	85	76	342	177	190	28.0	A-Terminal	yes	B
sun power VR M 12-105	12	101	103	98	87	344	177	230	38.0	F-M8	no	A
sun power VR M 12-135	12	125	128	122	111	344	170	275	46.0	F-M8	no	A
sun power VR M 12-150	12	146	151	146	133	498	177	230	55.0	F-M8	no	A
sun power VR M 6-200	6	186	190	183	167	242	170	275	32.0	F-M8	no	C
sun power VR M 6-250	6	247	253	243	229	308	170	275	41.0	F-M8	no	C

C₁₀₀, C₄₈, C₂₄ and C₁₀ = Capacity at 100 h, 48 h, 24 h and 10 h discharge

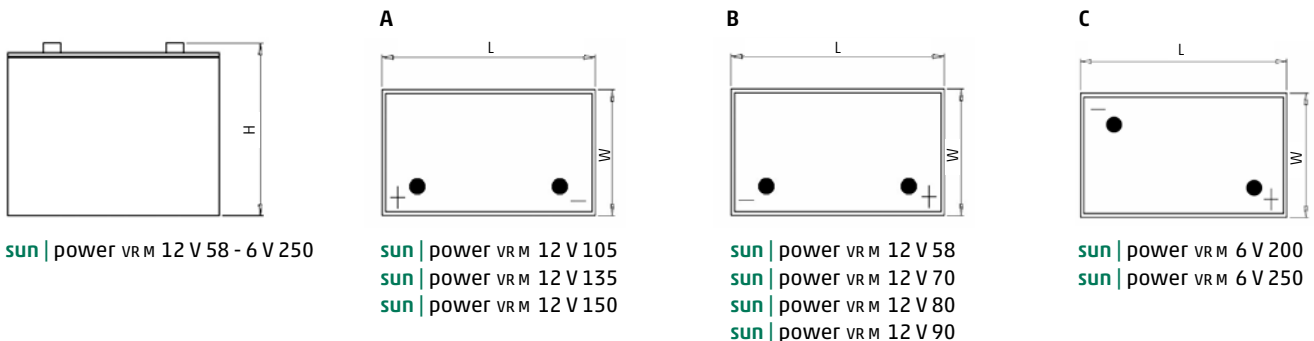
Service life in cycles and Depth of Discharge



Connection and torque



Terminal layout



Optimal environmental compatibility - closed loop for recovery of materials in an accredited recycling system

IEC 60896-21

IEC 61427



Příloha č. 5: Technická specifikace komunikačního protokolu



Technical specification

Xtender serial protocol

Date : 04.12.17

Version : V1.6.30 (R644)

403C\Technical specification - Xtender serial protocol.pdf

Contents

1. INTRODUCTION	4
1.1 CONVENTIONS USED IN THIS DOCUMENT	4
1.2 LIST OF ACRONYMS	4
2. PHYSICAL LAYER	5
2.1 CONNECTOR PINNING	5
2.2 CABLE TO USE	5
3. DATA LINK LAYER	6
3.1 RS-232 CONFIGURATION.....	6
3.2 BYTE ENDIANNESS.....	6
3.3 FRAME	7
3.4 CHECKSUM ALGORITHM.....	8
3.5 ADDRESSING THE DEVICES	8
3.6 MULTICAST ADDRESSES	9
3.7 RESPONSE DELAY.....	9
3.8 HARDWARE WATCHDOG	9
3.9 SCOM WATCHDOG	9
3.10 XTENDER WATCHDOG	10
3.11 VARIOTRACK WATCHDOG.....	10
3.12 VARIOSTRING WATCHDOG.....	10
3.13 DEVICE IDENTIFICATION	10
4. APPLICATION LAYER	11
4.1 SERVICES	11
4.2 OBJECT MODEL	11
4.2.1 <i>READ_PROPERTY</i> service.....	12
4.2.2 <i>WRITE_PROPERTY</i> service	13
4.2.3 <i>Format</i>	13
4.3 ERROR CODES	14
4.4 USER INFO OBJECTS	16
4.4.1 <i>Properties</i>	16
4.4.2 <i>Available user info</i>	16
4.4.3 <i>Software version encoding</i>	16
4.4.4 <i>FID encoding</i>	16
4.5 PARAMETER OBJECTS	17
4.5.1 <i>Properties</i>	17
4.5.2 <i>Values of level properties</i>	17
4.5.3 <i>Available parameters on the Xtender Inverter</i>	17
4.5.4 <i>Cyclic write of parameters on the Xtender Inverter</i>	18
Deactivating the save in flash with {1550}	18
Use of the <i>unsaved_value_qsp</i> property	18
4.5.5 <i>Cyclic write of parameters on VarioTrack and VarioString</i>	18
4.5.6 <i>Cyclic write of parameters on other devices</i>	19
4.5.7 <i>Hours encoding</i>	19
4.5.8 <i>Days of the week encoding</i>	19
4.5.9 <i>Month of the year encoding</i>	19
4.5.10 <i>Date and time encoding</i>	19

4.5.11 <i>Signal encoding</i>	19
4.6 MESSAGE OBJECTS.....	20
4.6.1 <i>Description of the reading function:</i>	20
4.6.2 <i>Notification of new messages:</i>	20
4.6.3 <i>Sorting the messages</i>	20
4.6.4 <i>Request frame</i> :	20
4.6.5 <i>Response frame</i> :	20
4.6.6 <i>Content of property_data</i>	21
4.7 CUSTOM DATALOG FIELD OBJECT	22
4.7.1 <i>Format of the string</i>	22
4.7.2 <i>Importing into Data Analysis Tool</i>	22
4.7.3 <i>Property</i>	22
4.8 FILE TRANSFER OBJECT	23
4.8.1 <i>Request frame</i>	23
4.8.2 <i>Response frame</i>	24
4.8.3 <i>Transfer sequence</i>	25
5. EXAMPLES OF FRAMES	26
5.1 C LIBRARY	26
5.2 COMMAND LINE TOOL	26
5.3 READ THE VALUE OF A USER INFO.....	27
5.4 READ THE QSP_VALUE OF A PARAMETER	28
5.5 WRITE THE QSP_VALUE OF A PARAMETER.....	29
5.6 READ MESSAGES	30
5.7 FILE TRANSFER OBJECT.....	32
5.7.1 <i>Reading a directory</i>	32
5.7.2 <i>Reading a datalog file</i>	33

1. Introduction

This technical specification describes the protocol used to communicate with the Studer Innotec Xcom-232i communication module. It is also valid for the discontinued RCC-02/-03 special execution ES N° 32 (RCC-02/-03-32).

1.1 Conventions used in this document

- Numbers that start with "0x" are in hexadecimal, like in the C integer literals.
- Byte streams are represented with hexadecimal separated by space like "00 01 1F 48"
- constant values are usually represented in UPPER CASE
- field names are in lower_case_with_underscore

1.2 List of acronyms

RCC	The Studer Innotec remote control used to configure the Xtender system
Xcom-232i	The Studer Innotec RS-232 communication module that has the function of a DCE, Data Communications Equipment
DTE	Data Terminal Equipment, the PC or controller system that wants to communicate with the Xcom-232i
SCOM	Naming prefix used for the Studer Innotec serial protocol

2. Physical layer

The physical layer is RS-232. The Xcom-232i is equipped with a DE-9 (also known as DB-9) Female connector which provides this interface.

The serial port is galvanically separated with an isolation of 500 V DC relative to the negative battery potential.

2.1 Connector Pinning

On the female connector of the Xcom-232i, only the wires "receive data", "transmitted data" and ground are connected. The other wires are not connected, and the DTE must ignore signals such as CTS, DTR or DCD.

pin number	usage
1	not connected
2	RxD
3	TxD
4	not connected
5	GND
6	not connected
7	not connected
8	not connected
9	not connected

2.2 Cable to use

The cable to be used with a PC is a Female-Male, straight.

3. Data link layer

The data link layer, as defined in the OSI model, is used to send and receive frame on the RS-232.

3.1 RS-232 configuration

The default RS-232 configuration is defined as :

- A fixed baudrate of 38400 bps
- 1 start bit
- 8 bit of data, LSB first
- 1 parity bit
- even parity
- 1 stop bit

If you require a higher speed, it is possible to increase the Xcom-232i baudrate to 115200 bps. To do that, you require the last version of Xcom Configurator (available to download on the Studer official website under Download center->Software and Updates) in order to create the new configuration file. After the configuration file generation, insert the SD card in the powered Xcom-232i to upgrade his baudrate.

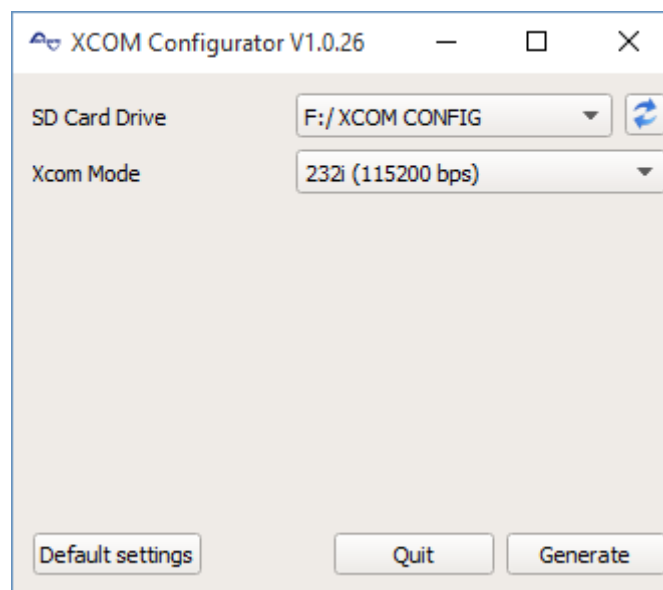


Figure 1: Xcom Configurator

As the baudrate is 3 times bigger, the cable length is 3 times shorter. The maximal cable length is about 1.3 meters.

The Xcom-232i MUST have at least the software version 1.5.88 or 1.6.20 in order to support the configuration file.

3.2 Byte Endianness

All values are in little endian, i.e. LSB bytes are sent on the Physical layer first.

3.3 Frame

The Xcom-232i and the DTE exchange frames consist of a header of 14 bytes followed by a variable number of data bytes and 2 bytes of checksum.

start_- byte	frame_- flags	src_- addr	dst_- addr	data_- length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes = N	2 bytes	N bytes	2 bytes

start byte

The start byte is always 0xAA

frame flags:

- BIT7-BIT6** : reserved.
- BIT5** : is_datalog_supported, 1 if the datalogger is supported.
- BIT4** : is_new_datalogger_file_present, 1 if there is a new datalog file on the SD card. This bit is reseted by the datalog read command through the SCOM or by extracting the SD card.
- BIT3** : is_sd_card_full, 1 if the SD card is full. This bit is reseted by extracting the SD card.
- BIT2** : is_sd_card_present, 1 if the SD card is present
- BIT1** : 1 at each start or restart of the Xcom-232i, also after a WD Reset. This bit can be cleared with the RCC/Xcom-232i signal parameter {5104}.
- BIT0** : is_message_pending flag, 1 if there are some messages pendings.

src addr

src_addr is the source address, 32 bit little endian

dst addr

dest_addr is the destination address, 32 bit little endian

data length

the length of the frame's data, in byte

The maximum number of frame_data is 240 (so that 14+240+2 = 256)

header checksum

the checksum of the header, from frame_flags to data_length (included)

frame data

the data bytes

data checksum

the checksum of all the data bytes of frame_data

3.4 Checksum algorithm

The checksum is computed with the following algorithm:

```

A = 0xFF
B = 0
For I FROM 0 TO number_of_bytes -1 DO
  A := (A + DATA[I]) mod 0x100;
  B := (B + A) mod 0x100;
END
checksum[0] := A
checksum[1] := B
  
```

A and B are byte values and the addition is made modulo 256.

After an invalid parity bit, header or data checksum, the data link layer is reseted and waits for an other frame.

3.5 Addressing the devices

Address	Devices	Remarks
0	Broadcast	
100	a virtual address to access all XTH, XTM and XTS	see section "multicast addresses"
101 to 109	a single XTH, XTM or XTS inverter	ordered by the index displayed on the RCC
191 to 193	virtual address to access properties on all inverters on a phase : 191 for L1, 192 for L2 and 193 for L3	a read access return the value of the master of the phase
300	a virtual address to access all VarioTrack	see section "multicast addresses"
301 to 315	VarioTrack	ordered by the index displayed on the RCC
501	Xcom-232i	alias for the gateway that the DTE uses to communicate (the Xcom-232i to which you speak with RS-232)
601	BSP	
700	a virtual address to access all VarioString	see section "multicast addresses"
701 to 715	VarioString	ordered by the index displayed on the RCC

3.6 Multicast addresses

A WRITE_PROPERTY to this kind of address will have the effect to change the property value on all devices of the same kind. READ_PROPERTY operations are not supported.

3.7 Response delay

The response delay of the Xcom-232i can be up to 2 seconds. This is a good value for a timeout in the DTE implementation.

The response delay depends on the bus load (number of devices, number of RCC or Xcom-232i, values displayed on the RCC). The use of the datalogger on the Xcom-232i or on other RCC can cause a periodic increase of the response delay every 60 seconds.

3.8 Hardware Watchdog

The remote control parameter {5103} "Activation of the watchdog hardware (deactivation restarts the Xcom-232i)" allows activation or deactivation of the hardware watchdog, initially disabled. In case the Xcom-232i is not working properly a Reset will be initiated automatically. The bit 1 of the frame_flags (see 3.3) reflects a start or restart of the Xcom-232i. This bit can be cleared with the RCC/Xcom-232i parameter {5104} "Clears the restart flag of Xcom-232i".

This function reset only the Xcom-232i. For the Xtender, see the parameters {1628} and {1629} in section "XTENDER Watchdog".

3.9 SCOM Watchdog

The RCC/Xcom-232i parameters {5095} "Enable SCOM watchdog" and {5096} "SCOM watchdog delay before reset of Xcom-232i" allow configuration of the SCOM watchdog, initially disabled. There is two kind of security activated by this parameter. First, when a request is received a timer is started. If the response is not send after {5096} seconds, for example if an inverter present does not respond, the Xcom-232i is reset. A second timer is also activated when the first request arrives. It is set to zero every time a request is received. When it reaches {5096} seconds the Xcom-232i is reset. Before both kind of reset a message will be sent. Like the hardware watchdog, the bit 1 of the frame_flags (see 3.3) reflects a start or restart of the Xcom-232i and it can be cleared with the RCC/Xcom-232i parameter {5104} "Clears the restart flag of Xcom-232i".

3.10 XTENDER Watchdog

The parameters {1628} and {1629} make possible to set a watchdog function inside the inverter. The watchdog is a monitoring software that restarts the Xtender in case the communication is lost.

This system is active when the parameter {1628} (Xtender watchdog enable) is enabled and the parameter {1550} (parameters saved in flash memory) is disabled or if you write parameter with the property *unsaved_value_qsp*. Each time the CAN receives a parameter the counter, whose duration is set in seconds by parameter {1629}, will be restarted. If no parameter is received during this period of time the Xtender will stop and a RESET will take place. The device restarts in the configuration determined by the parameter settings before {1550} was set to "no".

In a multi-unit system, each Xtender will handle this function independently. Each device must therefore receive a parameter within the time-frame set by {1629}.

By default this level is deactivated and the time period is set to 60s, adjustable from 10s to 300s.

3.11 VarioTrack Watchdog

The parameters {10342} and {10343} make possible to set a watchdog function inside the VarioTrack. The watchdog is a monitoring software that restarts the VarioTrack in case the communication is lost.

3.12 VarioString Watchdog

The parameters {14218} and {14219} make possible to set a watchdog function inside the VarioString. The watchdog is a monitoring software that restarts the VarioString in case the communication is lost.

3.13 Device identification

The RCC/Xcom-232i parameter {5119} allows to identify a device with the signaling system by flashing all of its LEDs. The written value corresponds to the Xtender's or VarioTrack' SCOM addresses. Unicast and multicast addresses are supported. For the Xtender, the phase addresses (191, 192 and 193) enable the signalling of all Xtenders on a given phase at the same time. The value 0 disables all signals. Sending a new value disables the previous value. If there is no more writing, all signals turn off after 45 seconds.

4. Application layer

The OSI layers 3 to 6 are not used. The application layer defines a number of « services ». A DTE sends a request frame and waits for a response frame from the Xcom-232i. If an error in the header checksum or data checksum is detected, there is no response from the application layer and the Xcom-232i waits for another request as if nothing has been received.

The Xcom-232i copies the src_addr of the request in the response dst_addr.

4.1 Services

The first two bytes of frame_data define the type of service and different flags for this service.

service_flags	service_id	service_data
1 byte	1 byte	N bytes

service_flags:

BIT7-BIT4 : reserved.

BIT1 : is_response flag, 0 if it is a request from the DTE to the Xcom-232i, 1 if it is response from the Xcom-232i.

BIT0 : error flag, 0 in case of success, 1 if an error occurred. In case of a request, error is always 0.

service_id:

One of the following services, described later in this document:

READ_PROPERTY = 0x01

service_data:

The data specific to the service. In case of a problem the errors are reported in a service-specific way, but the response has to include the error code described in the next section.

4.2 Object model

The different data accessible on each device are organized in object classes. Every object class has a number of properties. The service READ_PROPERTY is used to read the object's properties.

4.2.1 READ_PROPERTY service

This service is used to read an object's property.

The DTE sends a request frame with the following frame_data:

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x00	1 byte 0x01	2 bytes	4 bytes	2 bytes	0 byte

service_flags : is_response =0, error=0

service_id : 0x01 for READ_PROPERTY

object_type : the object type identifier, defined later in this document

object_id : the object identifier, specific to each object type, i.e. two objects with different type can have the same id

property_id : identify the property in the object

property_data : no property data

The Xcom-232i responds with a frame with the following frame_data:

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02 or 0x03	1 byte 0x01	2 bytes	4 bytes	2 bytes	N bytes or 2 bytes

service_flags : flags_response = 1, error= 0 or 1

service_id : 0x01 for READ_PROPERTY

object_type : same as the request

object_id : same as the request

property_id : same as the request

property_data : If error in service_flags is 0, the value of the property with the number of bytes of its type. If not, 2 bytes of type ERROR identifying the error code.

4.2.2 WRITE_PROPERTY service

This service is used to write an object's property.

The DTE sends a request frame with the following frame_data:

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x00	1 byte 0x02	2 bytes	4 bytes	2 bytes	n byte

service_flags : is_response = 0, error = 0

service_id : 0x02 for WRITE_PROPERTY

object_type : the object type identifier, defined later in this document

object_id : the object identifier, specific to each object type, i.e. two objects with different types can have the same id

property_id : identify the property in the object

property_data : the data in the right data type.

The Xcom-232i responds with a frame with the following frame_data:

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02 or 0x03	1 byte 0x02	2 bytes	4 bytes	2 bytes	0 or 2 bytes

service_flags : flags_response = 1, error= 0 or 1

service_id : 0x01 for READ_PROPERTY

object_type : same as the request

object_id : same as the request

property_id : same as the request

property_data : If error in service_flags is 0, 0 byte of data. If not, 2 bytes of type ERROR identifying the error code.

4.2.3 Format

The property data are encoded in different formats described below. Some properties have a format that can be different from one object to another for the same object_type. For example, the value_qsp of a parameter can be an ENUM or a FLOAT depending on the parameter id (identified by the object_id). In this case it is described here as type DYNAMIC. The DTE must then know the exact type of the property for each object to decode it.

BOOL : binary data, 1 byte, 0 = false, 1 = true, other values are invalid

FORMAT : a property that defines the format of another property, 16 bit integer

- SHORT_ENUM** : a value that is part of a enumeration of possible values, represented with a 16 bit integer
- LONG_ENUM** : a value that is part of a enumeration of possible values, represented with a 32 bit integer
- ERROR** : 16 bit error code
- INT32** : 32 bit signed value
- FLOAT** : float in 32 bit IEEE 754 format, little endian
- STRING** : ISO_8859-15 string of 8 bit characters
- DYNAMIC** : a property with a different format for each object id
- BYTE_STREAM** : a stream a byte of abitrary length

example of dynamic property:

for the object type 1 and object id 3000 (XT batery voltage), the format is FLOAT and "value" is a 4 byte IEEE 754 little endian float.

4.3 Error codes

The following error codes of type ERROR can be returned:

name	error_id	meaning
INVALID_FRAME	0x0001	malformed frame
DEVICE_NOT_FOUND	0x0002	wrong dst_addr field
RESPONSE_TIMEOUT	0x0003	no response of the server
SERVICE_NOT_SUPPORTED	0x0011	wrong service_id field
INVALID_SERVICE_ARGUMENT	0x0012	wrong service_data
SCOM_ERROR_GATEWAY_BUSY	0x0013	gateway (for example XCOM-232i) busy
TYPE_NOT_SUPPORTED	0x0021	the object_type requested doesn't exist
OBJECT_ID_NOT_FOUND	0x0022	no object with this object_id was found
PROPERTY_NOT_SUPPORTED	0x0023	the property identified by property_id doesn't exist
INVALID_DATA_LENGTH	0x0024	the field property_data has an invalid number of bytes
PROPERTY_IS_READ_ONLY	0x0025	a writing to this property is not allowed
INVALID_DATA	0x0026	this value is impossible for this property
DATA_TOO_SMALL	0x0027	the value is below the minimum limit
DATA_TOO_BIG	0x0028	the value is above the maximum limit
WRITE_PROPERTY_FAILED	0x0029	writing is possible, but failed
READ_PROPERTY_FAILED	0x002A	reading is possible, but failed
ACCESS_DENIED	0x002B	insufficient user access

SCOM_ERROR_OBJECT_NOT_SUPPORTED	0x002C	this object id, through existant, is not supported by the current implementation of the gateway
SCOM_ERROR_MULTICAST_READ_NOT_SUPPORTED	0x002D	Read operation is not supported when used on multicast adresses.
OBJECT_PROPERTY_INVALID	0x002E	During a file transfer, the use of this property was unexpected
FILE_OR_DIR_NOT_PRESENT	0x002F	Attempt to download a file not present on the sd card
FILE_CORRUPTED	0x0030	A read error occurred during the download of a file
INVALID_SHELL_ARG	0x0081	the command line tool used received the wrong arguments

Remark: when writing Xtender parameters, take in account the parameters interdependencies that reduce the allowed value ranges. See in RCC manual : "APPENDIX 1: LIST OF CONFIGURATION INTERDEPENDENCIES".

4.4 User info objects

These objects are the information about the current state of the system. They cannot be modified and their values change during the operation of the system. Previously known as system states.

object_type = 0x0001

object_id : see the table in next section

4.4.1 Properties

Name	property_id	format	remark
Value	0x0001	DYNAMIC	variable length, see the format in following table

4.4.2 Available user info

The available user information is the same as the values that can be chosen to be displayed on the RCC. The user information is related with the inverter parameters that can be configured with the RCC. The functionalities of each parameter are described in the RCC manual. You can easily find specific parameters by using the parameter index at the end of the manual.

4.4.3 Software version encoding

The software version is of the form X.Y.Z. It is encoded in a 32 bit unsigned value:

8 bit (MSB): X	8 bit: reserved	8 bit: Y	8 bit (LSB):Z
----------------	-----------------	----------	---------------

The 32 bit value is formed by combining the SOFT ID MSB (Most Significant Bits) and SOFT ID LSB (Least Significant Bits) User Info. These two values are in FLOAT and must be converted to 16 unsigned beforehand.

4.4.4 FID encoding

The FID is a unique identifier for the device. It is encoded in a 32 bit unsigned value.

This value is formed by combining the ID FID MSB and ID FID LSB User Info. These two values are in FLOAT and must be converted to 16 unsigned beforehand.

4.5 Parameter objects

object_type = 0x0002

All parameters accessible from the remote control can also be modified with the protocol. The behaviour is the same as if a physical person changes the value with the remote control buttons. Currently, only changes at the level qsp are possible.

Values of type FLOAT are stored internally in various 16 bit fixed point formats. For this reason, the read back value after a write can be rounded slightly.

4.5.1 Properties

Name	property_id	format	Remark
value_qsp	0x0005	DYNAMIC	the value that can be entered on the remote control in level qsp or installer.
min_qsp	0x0006	DYNAMIC	Minimum that can be entered on the remote control in level qsp or installer.
max_qsp	0x0007	DYNAMIC	Maximum that can be entered on the remote control in level qsp or installer.
level_qsp	0x0008	ENUM	accessibility level of this parameter modifiable in level qsp or installer.
unsaved_value_qsp	0x000D	DYNAMIC	the value that can be entered on the remote control in level qsp or installer, but without saving the value in internal flash. See section "Cyclic write of parameters on the Xtender Inverter" for more detail.

4.5.2 Values of level properties

The property `level_qsp` of type ENUM can take the following values:

Name	value
VIEW_ONLY	0x0000
BASIC	0x0010
EXPERT	0x0020
INSTALLER	0x0030
QSP	0x0040

4.5.3 Available parameters on the Xtender Inverter

The change of parameters when the inverters are in operation should be done carefully. The modification of parameters can restart the corresponding algorithm inside the inverter. For example, the change of a delay can restart the timer attached to it.

object_id : a number starting at 1000. See the complete parameter references at the end of the RCC User manual.

4.5.4 Cyclic write of parameters on the Xtender Inverter

The Xtender inverter store the parameter values in a non volatile flash memory. Because of the endurance of this memory, the number of write on a single parameter property is only guaranteed for 1000 write operations.

To allow the cyclic write of parameters without count limit, the flash should not be accessed. There are two different ways to archive this. The first is to deactivate the write via parameter {1550} "Parameters saved in flash memory". On newer software versions, there is the possibility to modify the value without accessing the flash with the property *unsaved_value_qsp*.

Deactivating the save in flash with {1550}

The parameter {1550} "Parameters saved in flash memory" has the value "yes" by default. A write of "no" to this parameter value stop the write in the non-volatile flash memory. This operation is written in the flash memory only the first time, so consecutive writes of the value "no" to {1550} can be repeated without limit.

After parameter {1550} has been set to "no", all other parameters can be written without count limit. Because the values of all other parameters are not stored in flash, the read operation will give the values before {1550} as be changed to "no". Also, after a reset the old values will be taken.

To use the inverter with cyclic write operations you must:

- ensure that all inverters have a firmware version $\geq 1.4.6$
- set the parameter {1550} to "no" on all targeted inverter
- avoid to write cyclically on other devices like BSP, RCC, ...
- ensure that no "reset default/factory settings", "apply configuration file (masterfile)" or modification with the remote control change {1550} to "yes"

It is a good practice to cyclically write "no" to {1550}.

A write of "yes" to the parameter {1550} reactivate the write in flash. It will be written in the flash every time and should not be used more that 1000 time.

Use of the *unsaved_value_qsp* property

This possibility is only available since version XT $\geq 1.6.12$.

The property *unsaved_value_qsp*, described in section "Parameter objects, Properties", allows to modify the value used by the device software in RAM, without changing the value saved in flash and loaded at start up. The property *value_qsp* can be used at the same time, for example to configure the default value after a turn off or a watchdog reset.

4.5.5 Cyclic write of parameters on VarioTrack and VarioString

The VarioTrack and Variostring behave the same way as the Xtender Inverter. The parameter {10058} for the VarioTrack and {14069} for the VarioString allows to deactivate the write in non volatile memory.

The property *unsaved_value_qsp* is available since version VT $\geq 1.6.14$ and VS $\geq 1.6.12$.

4.5.6 Cyclic write of parameters on other devices

On other devices, such as BSP, Xcom-232i, Xcom-CAN, there no parameter to deactivate the writing to the non volatile flash memory.

On BSP $\geq 1.6.12$, xcom-CAN $\geq 1.6.12$ and Xcom-232i $\geq 1.6.12$, the property *unsaved_value* can be used for cyclical write.

4.5.7 Hours encoding

the hours encoding is in minute since 00:00 in INT32. For example 13:41 is $13*60+41 = 821$.

4.5.8 Days of the week encoding

The days of the week selection (parameters {1205}, for example) is coded as a bit field in a INT32. A day selected as it bit set to 1.

bit	BIT31-7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
day of the week	undefined	SU	SA	FR	TH	WE	TU	MO

4.5.9 Month of the year encoding

The month of the year selection (parameters {1479}, for example) is coded as a bit field in a INT32. A month selected as it bit set to 1. January is BIT0 and December BIT11. The BIT31 to 12 are undefined.

4.5.10 Date and time encoding

The time of the Real Time Clock of the system is coded as a INT32. The value is the number of second since 1.1.1970 00:00:00. The parameters {5002} (Date) take and return the value that contains the complete day and hour information.

4.5.11 Signal encoding

The Signal (parameters {1468}, for example) is coded as a INT32. To send a signal, you must write the value 1 to the parameter value.

4.6 Message objects

The Message objects are supported for Xcom-232 with version $\geq 1.5.0$.

The messages sent by the devices on the communication bus are stored by Xcom-232i in its non-volatile flash memory. They can be read on the Xcom-232i (address 501) later.

4.6.1 Description of the reading function:

Reading a message with index 0 will return the last saved message in the flash memory of the Xcom-232i. In the response frame from the SCOM the first data indicates the number of remaining messages before attaining the very first message saved in the flash memory (this behaviour is identical with the history of RCC). A pointer is saved in the Xcom-232i when reading the index 0 (SCOM_MSG_IDX).

Reading a message with an index superior to 0 will return the message saved in the SCOM_MSG_IDX index.

Reading a message with index 0 will erase the flag informing that there are new messages.

If a new message is received after the last reading of index 0, the notification flag is reactivated. Then the PC user must make a new index 0 reading in order to update the pointer (SCOM_MSG_IDX).

4.6.2 Notification of new messages:

A notification flag indicating new messages is sent in every response frame from the Xcom.

4.6.3 Sorting the messages

In order to determine whether the PC user has received all messages, he has to create a unique identifier including the address of the source, the time and date.

4.6.4 Request frame :

The server sends a request in the following format :

flags	service_id	object_type	object_id	property_id
1 byte 0x00	1 byte 0x01	2 bytes	4 bytes	2 bytes

flags : is_response =0, error=0

service_id : 0x01 for READ_PROPERTY

object_type : 0x0003 (MESSAGE)

object_id : Index desired message

property_id : 0x00

4.6.5 Response frame :

The XCOM-232i responds with the following format :

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02 or 0x03	1 byte 0x01	2 bytes	4 bytes	2 bytes	N bytes

flags : is_response = 1, error= 0 ou 1
service_id : Same value as the query
object_type : Same value as the query
object_id : Same value as the query
property_id : Same value as the query
property_data : If error=0 → the message asked (see 4.6.6)
 If error=1 → 2 bytes for the error code

4.6.6 Content of property_data

Name	Size	Format	Remark
message_total_number	4 bytes	INT32	The total number of message in the XCOM-232i
message_type	2 byte	ENUM	The number defining the meaning of the message. See the appendix part messages.
source_address	4 byte	INT32	Source address of the message. See 3.5 Addressing the devices.
timestamp	4 byte	INT32	The time at which the message occurred in seconds since January 1, 1970.
value	4 byte	DYNAMIC	An optional value of the message. Not yet used currently.

4.7 Custom datalog field object

object_type = 0x0005

The custom datalog field is a string present in each Xcom-232i on the system. It has a maximum length of 250 bytes and is stored in volatile memory. The most recent string of all the Xcom-232i in a system is stored at the end of the datalog file when it is written to the SD card. This appends every day at 00:00:30 or when the user requests it via parameter {5059}, for example.

During the save process of the custom datalog field on the SD card, the use of the SCOM service results in an error "0x0013 SCOM_ERROR_GATEWAY_BUSY".

After a reset, this value is an empty string. Only the service WRITE_PROPERTY on property_id = 0x1 is allowed.

4.7.1 Format of the string

The 3 characters ; , and # are reserved characters. The characters ; and , are used as a separator within the string.

The string should begin with a string identifier, for example #STU for the company Studer. The # character is used to identify the name of the string.

String example:

```
"#STU:;Service datalog text;;STUDER INNOTEK SA,Rue des Casernes 57,1950 Sion,Switzerland;;Tel.: +41 (0)27 205 60 80,Fax.: +41 (0)27 205 60 88,info(at)studer-innotec.com;;Long:46.227649,Lat:7.380954".
```

4.7.2 Importing into Data Analysis Tool

The imported data is displayed on the second sheet of the Data Analysis Tool.

The string identifier is not shown in the display of the Data Analysis Tool.

Each ; or , of the string corresponds to a line break in the Data Analysis Tool.

4.7.3 Property

Name	property_id	format	Remark
value	0x0001	STRING	A string of maximum 250 bytes that will be stored in the datalog file. This value is write only.

4.8 File transfer object

This object allows to download the files stored on the SD card of the Xcom-232i. The `object_type` defines the accessed directory. File names are stripped of their directory name. The current version only supports the datalog directory.

4.8.1 Request frame

The server sends a request in the following format:

flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x00	1 byte 0x01	2 bytes	4 bytes	2 bytes	N bytes

flags: is_response = 0, error = 0

service_id: 0x01 for READ_PROPERTY
0x02 for WRITE_PROPERTY

object_type: 0x0101 (Datalog Transfer, content of CSVFILES\LOG)

object_id: 0x00000001 → Directory list
0x00000002 → File access

property_id: 0x0000 → Invalid Action
0x0021 → SD_Start
0x0022 → SD_Datablock
0x0023 → SD_Ack_Continue
0x0024 → SD_Nack_Retry
0x0025 → SD_Abort
0x0026 → SD_Finish

property_data: If object_id = Directory list → No data
If property_id = Start → filename "LGYMMDD.CSV"¹ in Big Endian
If property_id = Ack and Continue → No data
If property_id = Nack and Retry → No data
If property_id = Abort → No data

¹ YY = Year, MM = Month, DD = Day

4.8.2 Response frame

The XCOM-232i responds with the following format:

flags	service_id	object_type	object_id	property_id	property_data
1 byte 0x02	1 byte 0x01	2 bytes	4 bytes	2 bytes	N bytes

flags: is_response = 1, error = 0

service_id: 0x01 for READ_PROPERTY
0x02 for WRITE_PROPERTY

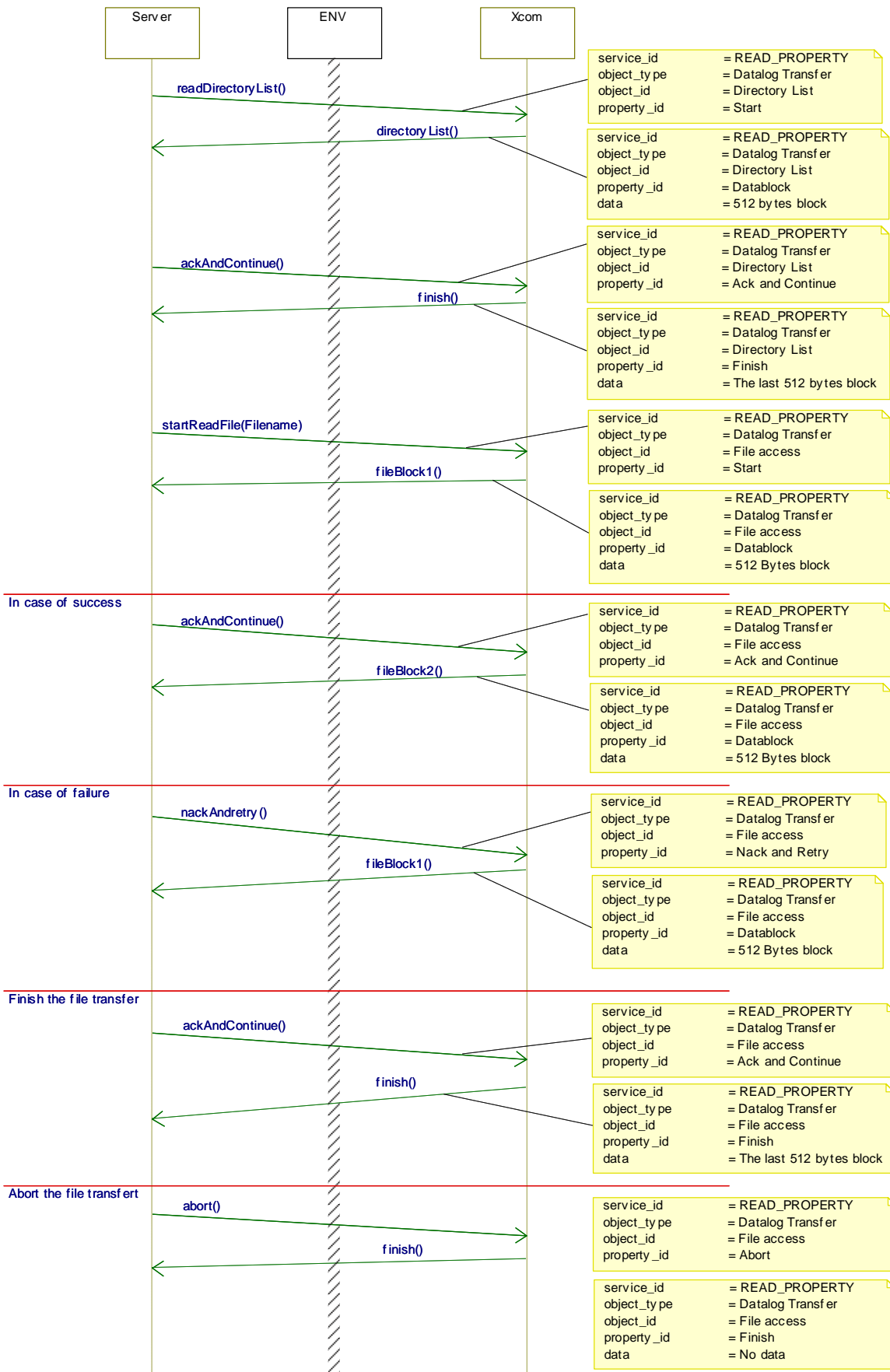
object_type: 0x0101 (Datalog Transfer, content of CSVFILES\LOG)

object_id: 0x00000001 → Directory list
0x00000002 → File access

property_id: 0x0000 → Invalid Action
0x0021 → SD_Start
0x0022 → SD_Datablock
0x0023 → SD_Ack_Continue
0x0024 → SD_Nack_Retry
0x0025 → SD_Abort
0x0026 → SD_Finish

property_data: If object_id = Directory list → Each filename with "CR" for separation
If property_id = Datablock → 512 Bytes / block
If property_id = Finish → No data
If error = 1 → Error number

4.8.3 Transfer sequence



5. Examples of frames

The byte stream is represented in hexadecimal and the encoding is little endian as specified in chapter 3.2.

In the different examples, the "frame_flags" byte contains the value 0x34, 00110100b in binary code. It means :

- BIT2 : SD card present.
- BIT4 : there is a new datalog file on the SD card.
- BIT5 : the datalogger is supported.

5.1 C library

A portable C library that implements the protocol is included with this specification on www.studer-innotec.com. See the documentation provided with the library for more detail.

5.2 Command line tool

To help the implementation of the protocol we supply also the command line tool `scom.exe`. It is included with the protocol specification available on www.studer-innotec.com.

5.3 Read the value of a user info

Generated by the command:

```
>scm.exe --port=COM3 --verbose=3 read_property src_addr=1 dst_addr=101
object_type=1 object_id=3000 property_id=1 format=FLOAT
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	101 (first inverter)	= N = 10	computed	10 bytes	computed
AA	00	01 00 00 00	65 00 00 00	0A 00	6F 71	10 bytes	C5 90

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	0 byte
is_response=false error=false	read_property	user_info	3000 (battery voltage)	value	-
00	01	01 00	B8 0B 00 00	01 00	-

Total number of bytes: 14+10+2 = 26 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00110100	101 (first inverter)	1	= N = 14	computed	14 bytes	computed
AA	34	65 00 00 00	01 00 00 00	0E 00	A7 45	14 bytes	0D CB

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	4 bytes
is_response = true error = false	read_property	user_info	3000 (battery voltage)	value	12.3594
02	01	01 00	B8 0B 00 00	01 00	00 C0 45 41

Total number of bytes: 14+14+2 = 30 bytes

5.4 Read the qsp_value of a parameter

Generated by the command:

```
> scom.exe --port=COM3 --verbose=3 read_property src_addr=1 dst_addr=101
object_type=2 object_id=1138 property_id=5 format=FLOAT
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	101 (first inverter)	= N = 10	computed	10 bytes	computed
AA	00	01 00 00 00	65 00 00 00	0A 00	6F 71	10 bytes	7D D9

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	0 bytes
is_response = false error = false	read_property	parameter	1138 (Battery charge current)	value_qsp	-
00	01	02 00	72 04 00 00	05 00	-

Total number of bytes: 14+10+2 = 26 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00110100	101 (first inverter)	1	= N = 14	computed	14 bytes	computed
AA	34	65 00 00 00	01 00 00 00	0E 00	A7 45	14 bytes	31 0B

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	4 bytes
is_response = true error = false	read property	parameter	1138 (battery charge current)	value_qsp	60
02	01	02 00	72 04 00 00	05 00	00 00 70 42

Total number of bytes: 14+14+2 = 30 bytes

5.5 Write the qsp_value of a parameter

Set the battery charge current at 12.0 A. Generated by the command:

```
>scm.exe --port=COM3 --verbose=3 write_property src_addr=1 dst_addr=101
object_type=2 object_id=1138 property_id=5 format=FLOAT value=12.0
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	101 (first inverter)	= N = 14	computed	14 bytes	computed
AA	00	01 00 00 00	65 00 00 00	0E 00	73 79	14 bytes	FF 9B

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	4 bytes
is_response = false error = false	write_property	parameter	1138 (Battery charge current)	value_qsp	12.0
00	02	02 00	72 04 00 00	05 00	00 00 40 41

Total number of bytes: 14+14+2 = 30 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00110100	101 (first inverter)	1	= N = 10	computed	10 bytes	computed
AA	34	65 00 00 00	01 00 00 00	0A 00	A3 3D	10 bytes	80 F6

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	0 byte
is_response = true error = false	write_property	parameter	1138 (battery charge current)	value_qsp	-
02	02	02 00	72 04 00 00	05 00	-

Total number of bytes: 14+10+2 = 26 bytes

5.6 Read messages

Reading of message 0. Generated by the command:

```
>scm --port=COM3 read_property src_addr=1 dst_addr=501 object_type=3
object_id=0 property_id=0 format=BYTE_STREAM
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	501 (Xcom- 232i gateway)	= N = 10	computed	10 bytes	computed
AA	00	01 00 00 00	F5 01 00 00	0A 00	00 D6	10 bytes	03 17

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	0 byte
is_response = false error = false	read_property	message	0 (message 0)	0	-
00	01	03 00	00 00 00 00	00 00	-

Total number of bytes: 14+10+2 = 26 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00110100	501 (Xcom- 232i gateway)	1	= N = 28	computed	28 bytes	computed
AA	34	F5 01 00 00	01 00 00 00	1C 00	46 0A	28 bytes	77 60

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	18 bytes
is_response = true error = false	read_property	message	0 (message 0)	0	-
02	01	03 00	00 00 00 00	00 00	D0 03 00 00 10 00 69 00 00 00 82 AD 9E 59 00 00 00 00

Total number of bytes: 14+28+2 = 44 bytes

« Property_data » decoding :

Décodage de la partie property_data :

Name	Size	Value	Format	Remark
message_total_number	4 bytes	D0 03 00 00 = 976d	INT32	The total number of message in the XCOM-232i
message_type	2 bytes	10 00 = 16d : Warning (016): Fan error detected	ENUM	The number defining the meaning of the message.
source_address	4 bytes	69 00 00 00 = 105d : Xtender 5	INT32	Source address of the message. See 3.5 Addressing the devices.
timestamp	4 bytes	82 AD 9E 59 = 1503571330 : 24.08.2017 at 10:42:10	INT32	The time at which the message occurred in seconds since January 1, 1970.
value	4 bytes	00 00 00 00	DYNAMIC	An optional value of the message. Not yet used currently.

5.7 File transfer object

5.7.1 Reading a directory

Generated by the command:

```
>scom --port=COM3 read_property src_addr=1 dst_addr=501 object_type=257
object_id=1 property_id=33 format=BYTE_STREAM
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	501 (Xcom- 232i gateway)	= N = 10	computed	10 bytes	computed
AA	00	01 00 00 00	F5 01 00 00	0A 00	00 D6	10 bytes	24 56

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	0 byte
is_response = false error = false	read_property	file tranfer	directory list	0x0021 start	-
00	01	01 01	01 00 00 00	21 00	-

Total number of bytes: 14+10+2 = 26 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00100111	501 (Xcom- 232i gateway)	1	= N = 36	computed	36 bytes	computed
AA	27	F5 01 00 00	01 00 00 00	24 00	41 8B	36 bytes	EE C9

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	26 bytes
is_response = true error = false	read_property	file tranfer	directory list	0x0026 finish	LG171010.CSV LG171011.CSV
02	01	01 01	01 00 00 00	26 00	4C 47 31 37 31 30 31 30 2E 43 53 56 0A 4C 47 31 37 31 30 31 31 2E 43 53 56 0A

Total number of bytes: 14+36+2 = 52 bytes

5.7.2 Reading a datalog file

Generated by the command:

```
>scom --port=COM3 read_property src_addr=1 dst_addr=501 object_type=257
object_id=2 property_id=33 format=BYTE_STREAM
```

Request

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	0	1	501 (Xcom- 232i gateway)	= N = 10	computed	10 bytes	computed
AA	00	01 00 00 00	F5 01 00 00	0A 00	00 D6	23 bytes	24 56

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	13 bytes
is_response = false error = false	read_property	file transfer	file access	0x0021 start	LG171010.CSV
00	01	01 01	02 00 00 00	21 00	4C 47 31 37 31 30 31 30 2E 43 53 56 0A

Total number of bytes: 14+23+2 = 39 bytes

Response

start_- byte	frame_- flags	src_addr	dst_addr	data_length	header_- checksum	frame_- data	data_- checksum
1 byte	1 byte	4 bytes	4 bytes	2 bytes	2 bytes	N bytes	2 bytes
always AA	00100111	501 (Xcom- 232i gateway)	1	= N = 36	computed	36 bytes	computed
AA	27	F5 01 00 00	01 00 00 00	24 00	41 8B	36 bytes	EE C9

service_flags	service_id	object_type	object_id	property_id	property_data
1 byte	1 byte	2 bytes	4 bytes	2 bytes	26 bytes
is_response = true error = false	read_property	file transfer	file access	0	LG171010.CSV LG171011.CSV
02	01	01 01	02 00 00 00	26 00	4C 47 31 37 31 30 31 30 2E 43 53 56 0A 4C 47 31 37 31 30 31 31 2E 43 53 56 0A

Total number of bytes: 14+36+2 = 52 bytes

Příloha č. 6: Navržený algoritmus řízení

```
DBOpen( "Connection1" )  
DBOpen( "Connection1.Recordset1" )  
id=$hour + Tpredpoved
```

```
SQLDotaz = "id, Zareni, Teplota FROM DATATABLE where ID = " + id  
DBExecute ("Connection1.Recordset1", "Source", SQLDotaz)  
DBExecute ("Connection1.Recordset1", "Requery" )
```

```
SetupCOMPort(1,set)  
baudrate=38400  
startbit=1  
bytesize=8  
paritybyte=1  
parity= even  
stopbits=1
```

```
OpenCOMPort1(1, Open)  
Napeti_baterie="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x01\x00\xB8\x0B\x00\x00\x01\x00\xC5\x90'  
InputCOMPort(1,write, Status)  
Napeti_bat=readCOMPort(1,read)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Proud_baterie="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x01\x00\xBD\x0B\x00\x00\x01\x00\xCA\xAE'  
InputCOMPort(1,write, Status)  
Proud_bat=readCOMPort(1,read)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Vstupni_proudAC="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xC4\x0B\x00\x00\x01\x00\xD1\xD8'  
InputCOMPort(1,write, Status)  
ProudAC_vst=readCOMPort(1,read)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Vykon_stridac="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x01\x01\x00\xCF\x0B\x00\x00\x01\x00\xDC\x1A'  
InputCOMPort(1,write, Status)  
Pstrid=readCOMPort(1,read)  
CloseCOMPort(1,close)
```

```
IF &hour>="6" AND &minute>="0" AND &hour<="19" AND &minute<="59"  
SUN=TRUE  
MOON=FALSE  
ELSEIF &hour>="20" AND &minute>="0" AND &hour<="5"AND &minute<="59"  
SUN=FALSE  
MOON=TRUE
```

END IF

SELECT CASE prepinač

Case 1

SELECT CASE Tpredpoved

Case 0

IF Tpredpoved=' '

Error_message='Vyplň políčko pro předpověď'

END IF

IF Pstrid>=Spotreba AND Napeti_bat >=7.2 AND Napeti_bat < 12

OpenCOMPort1(1, Open)

Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

OpenCOMPort1(1, Open)

Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

OpenCOMPort1(1, Open)

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

OpenCOMPort1(1, Open)

If ProudAC_vst >= 2 AND <3

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

Elseif ProudAC_vst >= 3 AND <4

OpenCOMPort1(1, Open)

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

Elseif ProudAC_vst >= 4 AND <5

OpenCOMPort1(1, Open)

Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'

InputCOMPort(1,write, Status)

CloseCOMPort(1,close)

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If

BatteryState = "Nabijeni "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= TRUE
SipkySB= TRUE
SipkyAC=FALSE
Dum= TRUE

ElseIF Pstrid>=Spotreba AND Napeti_bat >= 12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

BatteryState = "Vybíjení "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= TRUE
SipkySB= FALSE
SipkyAC=FALSE
Dum= TRUE

ElseIF Pstrid<Spotreba AND Napeti_bat >= 12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
AC_GRID="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= FALSE
SipkyAC=TRUE
Dum= TRUE
```

```
ElseIF Pstrid<Spotreba AND Napeti_bat <= 7.2
OpenCOMPort1(1, Open)
Inverter_ON="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
AC_GRID="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
End If
```

```
BatteryState = "Nabijeni "
```

```
SipkyBS=FALSE
```

```
SipkySD= TRUE
```

```
SipkyPS= FALSE
```

```
SipkySB= TRUE
```

```
SipkyAC=TRUE
```

```
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND Napeti_bat >=12
```

```
OpenCOMPort1(1, Open)
```

```
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= FALSE
SipkyAC=FALSE
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Ppred<=10 AND Napeti_bat <=7.2
```

```
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If

BatteryState = "Nabijeni "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE

CASE ELSE
Select Case Zareni
Case 0 to 99
Ppred=10

Case 100 to 199
Ppred=350

Case 200 to 299
Ppred=750

Case 300 to 399
Ppred=1125

Case 400 to 499
Ppred=1500

Case 500 to 599
Ppred=1875
```



```
Case 600 to 699  
Ppred=2250
```

```
Case 700 to 799  
Ppred=2625
```

```
Case 800 to 899  
Ppred=3000  
Case 900 to 999  
Ppred=3375  
Case else  
Ppred=3750  
End select
```

```
IF Ppred>=Spotreba AND Napeti_bat >= 12  
OpenCOMPort1(1, Open)  
Inverter_ON="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = " Vybíjení "  
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= FALSE  
SipkySB= TRUE  
SipkyAC=FALSE  
Dum= TRUE  
END IF
```

```
ElseIF Ppred<Spotreba AND Napeti_bat >= 7.2 AND Napeti_bat <= 12  
Inverter_ON="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost="\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
If ProudAC_vst >= 2 AND <3  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

Elseif

```
If ProudAC_vst >= 6 AND <7  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)  
End If
```

```
BatteryState = " Nabíjení "  
SipkyBS=FALSE  
SipkySD= TRUE
```

```
SipkyPS= FALSE  
SipkySB= TRUE  
SipkyAC=TRUE  
Dum= TRUE
```

```
ElseIF Ppred<Spotreba AND Napeti_bat <= 7.2
```

```
OpenCOMPort1(1, Open)
```

```
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
If ProudAC_vst >= 2 AND <3
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If
```

```
BatteryState = " Nabíjení "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND Napeti_bat >=12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
BatteryState = "Vybijeni "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= FALSE
SipkyAC=FALSE
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND Napeti_bat <=7.2
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\
x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If
```

```
BatteryState = "Nabijeni "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE
ENDIF
END SELECT
```

Case 2

```
IF hod1=' ' OR hod2=' ' OR hod3=' ' OR hod4=' ' OR hod5=' ' OR hod6=' ' OR hod7=' '
OR hod8=' ' OR hod9=' ' OR hod10=' ' OR hod11=' ' OR hod12=' ' OR hod13=' ' OR
hod14=' ' OR hod15=' ' OR hod16=' ' OR hod17=' ' OR hod18=' ' OR hod19=' ' OR
hod20=' ' OR hod21=' ' OR hod22=' ' OR hod23=' ' OR hod00=' ' OR
Error_message='Vyplň profil spotřeby!'
END IF
```

```
IF Tpredpoved=' '
Error_message='Vyplň políčko pro předpověď'
END IF
```

```
IF hod1>Spotreba OR hod2>Spotreba OR hod3>Spotreba OR hod4>Spotreba OR
hod5>Spotreba OR hod6>Spotreba OR hod7>Spotreba OR hod8>Spotreba OR
hod9>Spotreba OR hod10>Spotreba OR hod11>Spotreba OR hod12>Spotreba OR
hod13>Spotreba OR hod14>Spotreba OR hod15>Spotreba OR hod16>Spotreba OR
hod17>Spotreba OR hod18>Spotreba OR hod19>Spotreba OR hod20>Spotreba OR
hod21>Spotreba OR hod22>Spotreba OR hod23>Spotreba OR hod24>Spotreba OR
znamenko=1
END IF
```

```
IF hod1<Spotreba OR hod2<Spotreba OR hod3<Spotreba OR hod4<Spotreba OR
hod5<Spotreba OR hod6<Spotreba OR hod7<Spotreba OR hod8<Spotreba OR
hod9<Spotreba OR hod10<Spotreba OR hod11<Spotreba OR hod12<Spotreba OR
hod13<Spotreba OR hod14<Spotreba OR hod15<Spotreba OR hod16<Spotreba OR
hod17<Spotreba OR hod18<Spotreba OR hod19<Spotreba OR hod20<Spotreba OR
hod21<Spotreba OR hod22<Spotreba OR hod23<Spotreba OR hod24<Spotreba OR
znamenko=2
END IF
```

```
SELECT CASE Tpredpoved ;policko pro zadani hodin predpovedi
Case 0 ;rizeni dle reálného casu
```

```
IF Pstridac >= Spotreba AND znamenko = 1 AND Napeti_bat >= 7.2 AND Napeti_bat < 12
```

```
OpenCOMPort1(1, Open)
```

```
Inverter_ON = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
OpenCOMPort1(1, Open)
```

```
If ProudAC_vst >= 2 AND < 3
```

```
Nabijeci_proud = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
Elseif ProudAC_vst >= 3 AND < 4
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
Elseif ProudAC_vst >= 4 AND < 5
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
Elseif ProudAC_vst >= 5 AND < 6
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud = '\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
```

```
InputCOMPort(1, write, Status)
```

```
CloseCOMPort(1, close)
```

```
Elseif ProudAC_vst >= 6 AND < 7
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If

BatteryState = "Nabijeni "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= TRUE
SipkySB= TRUE
SipkyAC=FALSE
Dum= TRUE

ELSEIF Pstridac>=Spotreba AND znamenko=1 AND Napeti_bat >=12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

BatteryState = "Vybíjení "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= TRUE
SipkySB= FALSE
SipkyAC=FALSE
Dum= TRUE

ELSEIF Pstridac<=Spotreba AND znamenko=2 AND Napeti_bat >=7.2 AND
Napeti_bat < 12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```



```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\
x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If
```

```
BatteryState = "Nabíjení "
SipkyBS=FALSE
```

```
SipkySD= TRUE  
SipkyPS= FALSE  
SipkySB= TRUE  
SipkyAC=TRUE  
Dum= TRUE
```

```
ELSEIF Pstridac<=Spotreba AND znamenko=2 AND Napeti_bat >= 12  
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\  
x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "   
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= FALSE  
SipkySB= FALSE  
SipkyAC=TRUE  
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND znamenko=2 AND Napeti_bat >=12  
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "   
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= FALSE
```

```
SipkySB= FALSE  
SipkyAC=FALSE  
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND znamenko=2 AND Napeti_bat <=7.2
```

```
OpenCOMPort1(1, Open)
```

```
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
If ProudAC_vst >= 2 AND <3
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
```

```
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If

BatteryState = "Nabíjení "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE
END IF

CASE ELSE
Select Case Zareni
Case 0 to 99
Ppred=10

Case 100 to 199
Ppred=350

Case 200 to 299
Ppred=750

Case 300 to 399
Ppred=1125

Case 400 to 499
Ppred=1500

Case 500 to 599
Ppred=1875

Case 600 to 699
Ppred=2250

Case 700 to 799
Ppred=2625
```

```
Case 800 to 899  
Ppred=3000  
Case 900 to 999  
Ppred=3375  
Case else  
Ppred=3750  
End select
```

```
IF Ppred>=Spotreba AND znamenko=1 AND Napeti_bat >= 12
```

```
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "
```

```
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= TRUE  
SipkySB= FALSE  
SipkyAC=FALSE  
Dum= TRUE  
END IF
```

```
ElseIF Ppred<Spotreba AND Napeti_bat >= 7.2 AND Napeti_bat <= 12 AND  
znamenko=2
```

```
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\  
x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif
If ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If
```

```
BatteryState = "Nabíjení "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= TRUE
SipkySB= TRUE
SipkyAC=FALSE
Dum= TRUE
```

```
ElseIF Ppred<Spotreba AND Napeti_bat <= 7.2 AND znamenko=2
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\
x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\
x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

OpenCOMPort1(1, Open)
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 3 AND <4
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\
x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)  
End If
```

```
BatteryState = "Nabíjení "  
SipkyBS=FALSE  
SipkySD= TRUE  
SipkyPS= FALSE  
SipkySB= TRUE  
SipkyAC=TRUE  
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND Napeti_bat >=12 AND znamenko=2  
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "  
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= FALSE  
SipkySB= FALSE  
SipkyAC=FALSE  
Dum= TRUE
```

```
ElseIF Ppred>=0 AND Pred<=10 AND Napeti_bat <=7.2 AND znamenko=2  
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```



```
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
If ProudAC_vst >= 2 AND <3  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7  
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'  
InputCOMPort(1,write, Status)
```

```
CloseCOMPort(1,close)
End If
```

```
BatteryState = "Nabíjení "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE
ENDIF
END SELECT
```

Case 3

```
IF hod1=' ' OR hod2=' ' OR hod3=' ' OR hod4=' ' OR hod5=' ' OR hod6=' ' OR hod7=' '
OR hod8=' ' OR hod9=' ' OR hod10=' ' OR hod11=' ' OR hod12=' ' OR hod13=' ' OR
hod14=' ' OR hod15=' ' OR hod16=' ' OR hod17=' ' OR hod18=' ' OR hod19=' ' OR
hod20=' ' OR hod21=' ' OR hod22=' ' OR hod23=' ' OR hod00=' ' OR
Error_message='Vypln profil spotřeby!'
END IF
```

```
IF hod1>Spotreba OR hod2>Spotreba OR hod3>Spotreba OR hod4>Spotreba OR
hod5>Spotreba OR hod6>Spotreba OR hod7>Spotreba OR hod8>Spotreba OR
hod9>Spotreba OR hod10>Spotreba OR hod11>Spotreba OR hod12>Spotreba OR
hod13>Spotreba OR hod14>Spotreba OR hod15>Spotreba OR hod16>Spotreba OR
hod17>Spotreba OR hod18>Spotreba OR hod19>Spotreba OR hod20>Spotreba OR
hod21>Spotreba OR hod22>Spotreba OR hod23>Spotreba OR hod00>Spotreba OR
znamenko=1
END IF
```

```
IF hod1<Spotreba OR hod2<Spotreba OR hod3<Spotreba OR hod4<Spotreba OR
hod5<Spotreba OR hod6<Spotreba OR hod7<Spotreba OR hod8<Spotreba OR
hod9<Spotreba OR hod10<Spotreba OR hod11<Spotreba OR hod12<Spotreba OR
hod13<Spotreba OR hod14<Spotreba OR hod15<Spotreba OR hod16<Spotreba OR
hod17<Spotreba OR hod18<Spotreba OR hod19<Spotreba OR hod20<Spotreba OR
hod21<Spotreba OR hod22<Spotreba OR hod23<Spotreba OR hod00<Spotreba OR
znamenko=2
END IF
```

```
IF Pstridac>=Spotreba AND znamenko=1 AND Napeti_bat >=7.2 AND Napeti_bat <
12
```

```
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
If ProudAC_vst >= 2 AND <3
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x00\x40\x9F\xE4'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 3 AND <4
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 4 AND <5
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 5 AND <6
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
Elseif ProudAC_vst >= 6 AND <7
```

```
OpenCOMPort1(1, Open)
```

```
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
```

```
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
End If
```

```
BatteryState = "Nabíjení "
```

```
SipkyBS=FALSE
```

```
SipkySD= TRUE
```

```
SipkyPS= TRUE
```

```
SipkySB= TRUE
```

```
SipkyAC=FALSE  
Dum= TRUE
```

```
ELSEIF Pstridac>=Spotreba AND znamenko=1 AND Napeti_bat >=12  
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "
```

```
SipkyBS=TRUE  
SipkySD= TRUE  
SipkyPS= TRUE  
SipkySB= FALSE  
SipkyAC=FALSE  
Dum= TRUE
```

```
ELSEIF Pstridac<=Spotreba AND znamenko=2 AND Napeti_bat >=7.2 AND  
Napeti_bat < 12
```

```
OpenCOMPort1(1, Open)  
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\  
x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0  
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)  
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0B\x00\x70\x73\x00\  
x02\x02\x00\x65\x04\x00\x00\x05\x00\x01\x72\x06'  
InputCOMPort(1,write, Status)  
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
```

```
If ProudAC_vst >= 2 AND <3
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x9F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 3 AND <4
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x40\x40\xDF\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 4 AND <5
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\x80\x40\x1F\xE4'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 5 AND <6
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xA0\x40\x3F\x24'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)

Elseif ProudAC_vst >= 6 AND <7
OpenCOMPort1(1, Open)
Nabijeci_proud='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0E\x00\x73\x79\x00\x02\x02\x00\x53\x04\x00\x00\x05\x00\x00\x00\xC0\x40\x5F\x64'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
End If

BatteryState = "Nabíjení "
SipkyBS=FALSE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= TRUE
SipkyAC=TRUE
Dum= TRUE

ELSEIF Pstridac<=Spotreba AND znamenko=2 AND Napeti_bat >= 12
OpenCOMPort1(1, Open)
Inverter_ON='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\x02\x00\xCF\x0B\x18\x00\x01\x00\xAC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
Smart_boost='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x0
2\x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1C'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
OpenCOMPort1(1, Open)
AC_GRID='\xAA\x00\x01\x00\x00\x00\x65\x00\x00\x00\x0A\x00\x6F\x71\x00\x02\
x02\x00\xCF\x0B\x00\x00\x01\x00\xRC\x1B'
InputCOMPort(1,write, Status)
CloseCOMPort(1,close)
```

```
BatteryState = "Vybití "
SipkyBS=TRUE
SipkySD= TRUE
SipkyPS= FALSE
SipkySB= FALSE
SipkyAC=TRUE
Dum= TRUE
END IF
```

```
END SELECT
```