



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**EFEKTIVNÍ ŘÍZENÍ OHŘEVU TEPLÉ UŽITKOVÉ VODY  
S VYUŽITÍM FOTOVOLTAIKY**

EFFECTIVE MANAGEMENT OF DOMESTIC HOT WATER HEATING USING PHOTOVOLTAICS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**MARTIN TALAJKA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. ZDENĚK MATERNA, Ph.D.**

BRNO 2024

## Zadání bakalářské práce



156832

Ústav: Ústav počítačové grafiky a multimédií (UPGM)  
Student: **Talajka Martin**  
Program: Informační technologie  
Název: **Efektivní řízení ohřevu teplé užitkové vody s využitím fotovoltaiky**  
Kategorie: Softwarové inženýrství  
Akademický rok: 2023/24

### Zadání:

1. Proveďte rešerši existujících řešení a přístupů používaných pro hospodárný a současně komfortní ohřev TUV pomocí energie z domácí FVE.
2. Navrhněte vlastní řešení umožňující automatický, hospodárný a komfortní ohřev TUV ve formě doplňku do systému Home Assistant.
3. Zahajte sběr dat na testovací instalaci.
4. Navržené řešení implementujte.
5. Srovnajte dosažené výsledky s předchozím stavem (spotřeba energie, komfort uživatelů).
6. Vytvořte video prezentující cíle vaší práce a její výsledky.

### Literatura:

- Gomes, Isaiás, et al. "Recent Techniques Used in Home Energy Management Systems: A Review." *Energies* 15.8 (2022): 2866.
- Jin, Xin, et al. "Foresee: A user-centric home energy management system for energy efficiency and demand response." *Applied Energy* 205 (2017): 1583-1595.
- Yildiz, Baran, et al. "Analysis of electricity consumption and thermal storage of domestic electric water heating systems to utilize excess PV generation." *Energy* 235 (2021): 121325.

Při obhajobě semestrální části projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Materna Zdeněk, Ing., Ph.D.**  
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.  
Datum zadání: 1.11.2023  
Termín pro odevzdání: 9.5.2024  
Datum schválení: 9.11.2023

## Abstrakt

Táto práca rieši problém efektívneho riadenia ohrevu vody s využitím fotovoltického systému. K vyriešeniu zvoleného problému som navrhol algoritmus a implementoval ho vo forme integrácie do systému Home Assistant, ktorý zároveň slúži ako centrálné riadiace zariadenie. A zabezpečuje integráciu medzi bojlerom a fotovoltickým systémom. Vďaka tejto integrácii je možný automatický a komfortný ohrev vody z prebytkovej energie fotovoltického systému, poprípade batérií alebo distribučnej siete. Zároveň je možné sledovanie stavu systému a vzdialená správa zariadení. Užívateľ napríklad môže sledovať teplotu nahriatej vody, stav celkového systému, upraviť teplotu výhrevu alebo si nastaviť ohrev vody na určitý čas, pomocou aplikácie, alebo webového rozhrania. Výsledky práce ukazujú, že navrhnutý systém je jednoduchý na používanie a je schopný efektívne riadiť ohrev vody s využitím fotovoltického systému a tým aj šetriť energiu a plyn.

## Abstract

This work solves the problem of efficient control of water heating using photovoltaic system. To solve the chosen problem, I designed an algorithm and implemented it in the form of integration into the Home Assistant system, which also serves as a central control device. And it provides the integration between the boiler and the PV system. Thanks to this integration, it is possible to automatically and comfortably heat water from the surplus energy of the photovoltaic system, or alternatively from batteries or the distribution network. At the same time, system status monitoring and remote device management are possible. For example, the user can monitor the temperature of the heated water, the status of the overall system, adjust the heating temperature or set the water heating for a specific time, using an app or a web interface. The results of the work show that the proposed system is easy to use and is able to efficiently manage the water heating using the PV system, thus saving energy and gas

## Kľúčové slová

Fotovoltika, Venus OS, Ohrev vody, Bojler, Home Assistant, Doplnok, IoT, Inteligentná domácnosť, ESP32, MQTT

## Keywords

Photovoltaics, Venus OS, Water heating, Boiler, Home Assistant, Component, IoT, Smart home, ESP32, MQTT

## Citácia

TALAJKA, Martin. *Efektivní řízení ohřevu teplé užitkové vody s využitím fotovoltaiky*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Zdeněk Materna, Ph.D.

# Efektivní řízení ohřevu teplé užitkové vody s využitím fotovoltaiky

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Zdenka Materny, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....  
Martin Talajka  
5. mája 2024

## Podakovanie

Rád by som poďakoval môjmu vedúcemu, pánovi Ing. Zdeňkovi Maternovi, Ph.D. za jeho ochotu, konzultácie a rady, ktoré mi poskytol počas riešenia tejto práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Prehľad problematiky</b>	<b>4</b>
2.1	Inteligentné zariadenia a Internet vecí . . . . .	4
2.2	Home Assistant . . . . .	7
2.3	MQTT . . . . .	8
2.4	API . . . . .	9
2.5	Energia a termodynamika . . . . .	10
2.6	Štatistika . . . . .	12
<b>3</b>	<b>Prieskum existujúcich možností</b>	<b>13</b>
3.1	Integrácia bojleru . . . . .	13
3.2	Fotovoltaický systém . . . . .	14
3.3	Home Assistant integrácia . . . . .	15
<b>4</b>	<b>Návrh riešenia</b>	<b>16</b>
4.1	Návrh integrácie . . . . .	16
4.2	Prevedenie integrácie . . . . .	17
4.3	Návrh logiky . . . . .	18
<b>5</b>	<b>Implementácia</b>	<b>22</b>
5.1	Použité zariadenia a technológie . . . . .	22
5.2	Implementácia logiky . . . . .	25
5.3	Testovanie integrácie . . . . .	38
<b>6</b>	<b>Testovanie</b>	<b>40</b>
6.1	Nasadenie systému . . . . .	40
6.2	Komfort užívateľov . . . . .	43
6.3	Možné zmeny a rozšírenia . . . . .	43
<b>7</b>	<b>Záver</b>	<b>45</b>
	<b>Literatúra</b>	<b>46</b>

# Zoznam obrázkov

2.1	IoT komunikácia zariadenie-zariadenie (prevzaté z [18]). . . . .	5
2.2	IoT komunikácia zariadenie-cloud (prevzaté z [18]). . . . .	6
2.3	IoT komunikácia pomocou Back-Endu (prevzaté z [18]). . . . .	6
2.4	MQTT komunikácia medzi zariadeniami. . . . .	8
4.1	Schéma komunikácie medzi zariadeniami. . . . .	18
4.2	Diagram návrhu logiky. . . . .	19
5.1	Schéma zapojenia fotovoltického systému a bojleru. . . . .	22
5.2	Rozhranie Venus OS a zariadenia fotovoltického systému. . . . .	23
5.3	Bojler s mikrokontrolérom ESP32. . . . .	24
5.4	Architektúra jadra Home Assistant (prevzaté z [5]). . . . .	25
5.5	Architektúra integrácií Home Assistanta (prevzaté z [6]). . . . .	26
5.6	Diagram zapnutia/vypnutia komponentu. . . . .	27
5.7	Diagram inicializácie komponentu. . . . .	29
5.8	Diagram kontroly pred spustením logiky. . . . .	29
5.9	Diagram spustenia/zastavenia ohrevu vody v bojleri (počas dňa). . . . .	31
5.10	Diagram kontroly pred nočným ohrevom. . . . .	32
5.11	Diagram plánovania spustenia nočného ohrevu. . . . .	33
5.12	Diagram logiky spustenia nočného ohrevu. . . . .	34
5.13	Diagram spracovania zmeny stavu distribučnej siete. . . . .	35
5.14	Diagram spracovania zmeny stavu MQTT pripojenia. . . . .	37
5.15	Web stránka využitá na testovanie. . . . .	39
6.1	Graf teploty vody. . . . .	41
6.2	Graf stavu nabitia batérií. . . . .	41
6.3	Graf FV generácie a spotreby v domácnosti. . . . .	42

# Kapitola 1

## Úvod

V posledných rokoch pribúda stále viac domácností, ktoré využívajú fotovoltické panely. Hlavnými dôvodmi býva úspora alebo minimalizácia závislosti na dodávkach energií z tradičných zdrojov. Na maximalizovanie využitia energie, ktorú môže fotovoltika poskytnúť, sa využívajú hlavne batérie. Ktoré prebytočnú energiu ukladajú na neskoršie využitie. Je, ale možné, túto energiu uložiť aj v iných formách, ako je napríklad teplo.

V tomto prípade sa využíva bojler, ktorý nahrieva vodu na určitú teplotu, pomocou dostupných prebytkov. Aby tento systém fungoval správne, musí byť zabezpečená nejaká forma zistenia, kedy je dostatok zdrojov. To môže fungovať na strane fotovoltického systému, kde býva k tomuto účelu vyhradená zásuvka a samotný systém zásuvku zapína a vypína podľa dostupných zdrojov. Tento typ systému má niekoľko problémov, medzi ktoré patrí napríklad konštantné vypínanie a zapínanie zásuvky, keď sa systém snaží zistiť či dokáže poskytnúť dostatok zdrojov. Nemožnosť pokrytia kolísania generácie fotovoltických panelov z batérii alebo naplánovať ohrev na určitý čas, čo môže znížiť komfort užívateľov, ak sa jedná o jediný zdroj teplej vody. Je, ale možnosť využiť obyčajný bojler, ktorý, nedisponuje inteligentnými funkciami a tento cenový rozdiel vyvážiť na väčšom objeme. Alternatívou je využiť inteligentný bojler. Takéto bojlerové zariadenia zväčša nie sú lacnou záležitosťou a ich funkcie a možnosti sa veľmi líšia. Od najlacnejších, ktoré dokážu detegovať lacnejší prúd, disponujú iba mobilnou aplikáciou na správu, až po tie najdrahšie, ktoré umožňujú pripojenie a ovládanie pomocou Wi-Fi alebo pridanie do uzatvorenej aplikácie, ktorú poskytuje firma priamo k bojleru. Tu sa naskytá možnosť využiť lacnejší typ inteligentného bojlera, ktorý bude komunikovať s centrálnou jednotkou pomocou mikrokontroléru. A centrálna jednotka sa postará o celkové riadenie systému, tým dokážeme odstrániť nedostatky, ktoré sa vyskytujú pri použití obyčajného bojlera.

Cieľom tejto práce je navrhnúť a implementovať systém, ktorý umožní automatické ovládanie elektrického bojlera, na základe prebytkov z fotovoltických panelov, batérií a distribučnej sústavy. Jeho integráciu a jednoduchú správu pomocou centrálného systému na riadenie inteligentnej domácnosti. Vďaka tejto integrácii je možné dosiahnuť zvýšenia celkovej efektivity využitia solárnej energie v domácnosti. A v neposlednom rade aj komfort pre koncových užívateľov.

V kapitole 2 sa venujem teoretickým základom, ktoré sú potrebné pri návrhu a implementácii riešenia. Kapitola 3 rozoberá prieskum existujúcich riešení, ktoré mi poslúžili ako zdroj inšpirácie. Následne kapitola 4 obsahuje návrh riešenia a popis možnej implementácie. Zatiaľ, čo kapitola 5 popisuje využité zariadenia a technológie, implementáciu a testovanie integrácie. A kapitola 6 obsahuje samotné nasadenie systému s jeho testovaním a hodnotením od koncových užívateľov a možné budúce vylepšenia.

## Kapitola 2

# Prehľad problematiky

Táto kapitola sa zameriava na dôležité pojmy z teórie, ktoré sú podstatné k správne mu porozumeniu tejto práce a súvisia s tvorbou vlastného komponentu do systému Home Assistant. Na pochopenie tejto problematiky je nevyhnutné mať základné znalosti o IoT a inteligentných zariadeniach (kapitola 2.1), o fungovaní systému Home Assistant a jeho komponentoch (kapitola 2.2), o spôsoboch komunikácie medzi takýmito zariadeniami, ako je MQTT protokol (kapitola 2.3) a API (kapitola 2.4). Taktiež je dôležité mať prehľad o vzorcoch, použitých k výpočtu potrebnej energie na vyhriatie vody a spôsobu spracovania dát pomocou percentilu, spomenutých v kapitole 2.5 a 2.6.

### 2.1 Inteligentné zariadenia a Internet vecí

Aj, keď sa na prvý pohľad môže zdať, že tieto dva názvy sú podobné a môžeme ich zameniť, nie je tomu tak. Je veľa definícií, ktoré popisujú tieto dva koncepty rôzne, ale zhodujú sa v hlavných kľúčových funkciách, ktoré by malo, takéto zariadenie podporovať.

Inteligentné zariadenie je elektronické zariadenie, schopné vykonávať autonómne výpočty s možnosťou pripojiť sa k iným zariadeniam pomocou drôtového alebo bezdrôtového pripojenia [14]. Medzi takéto zariadenia, ale môžu patriť aj prístroje, ktoré neposkytujú možnosť pripojenia a komunikácie s inými zariadeniami mali by, ale poskytovať istú formu autonómneho správania za splnenia určitých podmienok. Napríklad bojler, ktorý sleduje lacnú tarifu v elektrickej sieti a podľa toho dokáže spustiť nahrievanie vody.

Kdežto Internet vecí (z anglického názvu „Internet of Things“ - *IoT*) môžeme definovať ako inteligentné zariadenia navzájom prepojené pomocou siete. Takto prepojené zariadenia navzájom vytvárajú distribuovanú sieť zariadení komunikujúcich s ľuďmi a inými prístrojmi [19]. V dnešnej dobe sa tento systém využíva vo všetkých odvetviach bežného života. Od uľahčenia alebo zefektívnenia práce a výrobných procesov vo firmách, kde zariadenia pripojené do IoT medzi sebou komunikujú a zdieľajú informácie o dostupných prostriedkoch, zataženosti liniek a podobne. Až po domácnosti, ktoré môžu disponovať inteligentnou elektrickou inštaláciou, zariadeniami, senzormi, vozidlami a rôznymi inými inteligentnými prostriedkami. Ktoré následne môžu autonómne spracovávať dostupné informácie, napríklad pri zapnutí televízora zhasnú svetlá alebo sa zatiahnu rolety. A v prípade pripojenia týchto zariadení do centrálného systému, ako je Home Assistant (viac v kapitole 2.2) je možné sledovať stav a ovládať domácnosť vzdialene pomocou aplikácie a docieľiť tak ešte väčší komfort pre užívateľov.



Aby zariadenia mohli úspešne naviazať spojenie a dokázali medzi sebou komunikovať musíme ich pripojiť do siete. K tomu sa využívajú rôzne komunikačné protokoly a modely. Preto ďalším dôležitým aspektom IoT, je pochopenie toho, ako sa tieto zariadenia pripájajú a komunikujú medzi sebou. K tomuto účelu je možné využiť dokument RFC 7452 [18], ktorý popisuje štyri základné modely komunikácie. Ďalej sa venujem iba trom modelom, ktoré boli využité v tejto práci.

### Komunikácia zariadenie-zariadenie (anglicky „Device-to-Device“<sup>1</sup>)

Tento model reprezentuje dve alebo viac zariadení, ktoré sa pripájajú a komunikujú priamo medzi sebou. Zvyčajne tieto zariadenia využívajú protokoly ako Bluetooth<sup>2</sup>, Z-Wave<sup>3</sup>, Zig-Bee<sup>4</sup> alebo MQTT<sup>5</sup>, ktorému sa bližšie venuje kapitola 2.3. Tento komunikačný model sa bežne využíva v systémoch, ako sú domáce automatizácie, ktoré komunikujú so zariadeniami s relatívne nízkymi požiadavkami na rýchlosť prenosu a objemom dát [12].

Vďaka tomu je možné veľmi rýchlo a efektívne sledovať a spracovávať stavy rôznych zariadení, ako je napríklad solárny systém. Tento systém publikuje aktuálny výkon generácie, iné zariadenie sa prihlási k odberu tejto informácie, spracuje ju a na základe výsledku dokáže zapnúť bojler.

Veľkou výhodou tohto riešenia môže byť nízka réžia a bezpečnosť, keďže komunikácia prebieha iba po lokálnej sieti alebo len medzi zariadeniami (príkladom môže byť Bluetooth<sup>2</sup>).



Obr. 2.1: IoT komunikácia zariadenie-zariadenie (prevzaté z [18]).

### Komunikácia zariadenie-cloud (z anglického názvu „Device-to-Cloud“)

Model, pri ktorom zariadenie priamo komunikuje s internetovou službou poskytovateľa, ktorá je určená na výmenu správ. Tento spôsob komunikácie často využíva existujúce technológie ako káblové alebo Wi-Fi pripojenie. Vytvorí sa priamy spoj medzi zariadením a internetovou sieťou, ktorý sa následne pripojí k cloudovej službe [12].

Nevýhodou tohto typu komunikácie je nutnosť pripojenia k internetu a následné zdieľanie dát s treťou stranou, ktorá poskytuje cloudové služby. Ďalšou významnou nevýhodou môže byť nedostatočné zabezpečenie, ktoré je kritické, pri takomto prístupe. V prípade

<sup>1</sup>Stretnúť sa môžeme aj s M2M alebo Machine-to-Machine názvom.

<sup>2</sup><https://www.bluetooth.com/>

<sup>3</sup><https://www.z-wave.com/>

<sup>4</sup><https://csa-iot.org/all-solutions/zigbee/>

<sup>5</sup><https://mqtt.org/>

nedostatočnej ochrany by mohol útočník získať prístup k citlivým dátam alebo pri použití inteligentného zámku, prístup do domu, poprípade k iným zariadeniam v domácnosti.

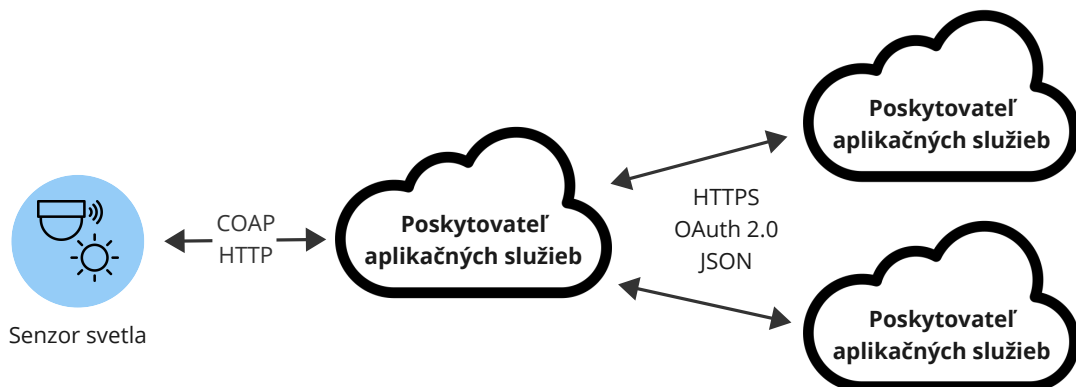


Obr. 2.2: IoT komunikácia zariadenie-cloud (prevzaté z [18]).

### Back-End zdieľanie dát (z anglického názvu „Back-End Data Sharing“)

Tento prístup rozširuje model komunikácie zariadenie-cloud o export a analýzu dát z cloudu poskytovateľa. Vďaka tomu môže užívateľ pristupovať k dátam zo zariadenia alebo viacerých zariadení, ktoré sú nahrané poskytovateľovi. Často sa využíva API model (kapitola 2.4) s kombináciou využitia autorizácie [12].

Model zdieľa rovnaké nevýhody, ako pri komunikácii zariadenie-cloud, ktoré boli spomenuté vyššie. Zároveň, ale disponuje výhodami, ako je vzdialený prístup k dátam, poprípade vzdialené ovládanie zariadení. Toto môže byť využité vo firmách, kde napríklad technik vidí aktuálny stav liniek, zásoby alebo iné informácie, ktoré sú preňho dôležité. Ale i v domácnostiach, kde si užívateľ môže skontrolovať či zhasol svetlá alebo si v zime zapnúť kúrenie skôr, ako príde domov.



Obr. 2.3: IoT komunikácia pomocou Back-Endu (prevzaté z [18]).

## 2.2 Home Assistant

Home assistant, je bezplatný softvér s otvoreným zdrojovým kódom, zameraný na automatizáciu domácností. Ktorý kladie vysoký dôraz na bezpečnosť a lokálne ovládanie [3]. Funguje ako centrálny systém na ovládanie, spravovanie a sledovanie stavu inteligentných zariadení v domácnosti.

Vďaka tomu, že je jednoduchý a modulárny, je jednoduché ho prispôsobiť k individuálnym potrebám každého používateľa. K jednému z podstatných aspektov patrí jeho možnosť rozšírenia o rôzne doplnky a integrácie, ktoré sú aktívne vyvíjané a udržiavané komunitou vývojárov a používateľov. Tieto doplnky a integrácie poskytujú užívateľom veľké množstvo rozšírených funkcií, vďaka ktorým je možné pridať skoro akékoľvek zariadenie do tohto systému.

Taktiež disponuje rôznymi funkciami, ktoré pomáhajú aj koncovým užívateľom, vytvorí jednoduchú automatizáciu rôznych zariadení. Pomocou automatizácií si môže veľmi jednoducho nastaviť akcie, ktoré sa majú vykonať pri nejakej zmene v systéme. Napríklad ak sa užívateľ vráti domov, zapnú sa svetlá na chodbe. Ak užívateľ každý večer pozerá televízor so zapnutými svetlami v pozadí a zatiahnutými roletami, môže si nastaviť scénu a pri jej aktivácii sa tieto zariadenia automaticky uvedú do príslušného stavu.

Nasledujúce časti vysvetľujú hlavné koncepty Home Assistanta a sú prevzaté z [4].

### Integrácia

Integrácia je časť softvéru, ktorá umožňuje Home Assistentovi komunikáciu s inteligentnými zariadeniami v domácnosti. Vďaka tomu si môže užívateľ pripojiť a spravovať rôzne zariadenia, ktoré vlastní. Ak by si chcel pridať napríklad osvetlenie, môže vyhľadať a nainštalovať integráciu, ktorá tieto svetlá podporuje a umožní ich ovládanie. Po úspešnom pridaní a správnom nastavení sa svetlá v systéme zobrazia ako samostatné zariadenia alebo entity, so stavmi, podľa podporovaných funkcií.

### Entity

Sú základnou súčasťou Home Assistanta. Slúžia na uchovávanie údajov, sledovanie fyzických parametrov inteligentných zariadení a ich ovládanie alebo ovládanie iných entít. Väčšinou sú vytvorené ako súčasť zariadenia alebo služby, ktorá bola pridaná pomocou integrácie. Napríklad, ak si užívateľ pridá inteligentné osvetlenie, môžu sa vytvoriť entity ako jas svetla, aktuálny stav - zapnuté alebo vypnuté, poprípade možnosti farby a ďalšie, podľa podporovaných funkcií.

### Zariadenia

Zariadenia sú jednoduchým zoskupením pre jednu alebo viac entít. Môže reprezentovať fyzické zariadenie, ktoré disponuje jedným alebo viacerými senzormi. Sensory sa potom zobrazujú ako entity, ktoré sú prepojené so zariadením. Napríklad, senzor pohybu môže byť reprezentovaný ako zariadenie, ktoré poskytuje informácie o detekcii pohybu, teplote a úrovni osvetlenia, ktoré sú následne zobrazené ako samostatné entity.

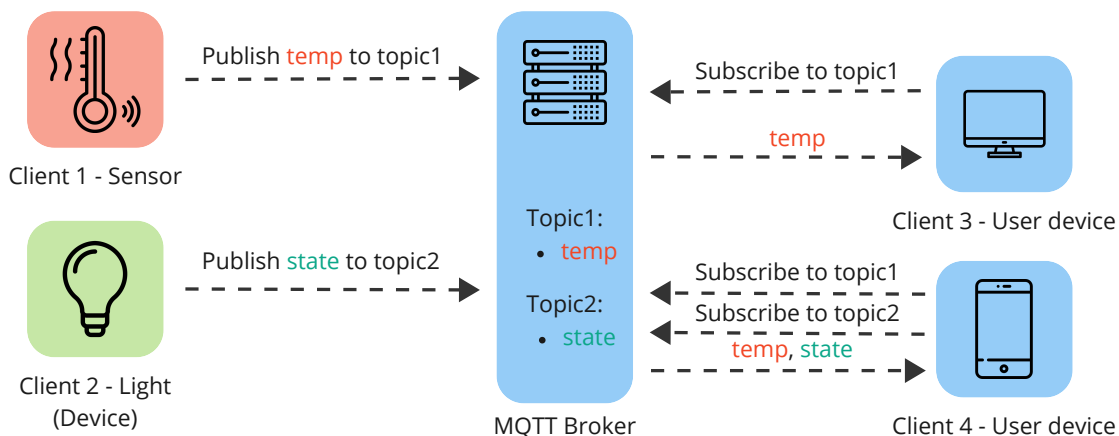
## 2.3 MQTT

MQTT (*Message Queuing Telemetry Transport*) je komunikačný protokol určený na používanie v IoT prostredí. Jeho hlavným cieľom je efektívna a spoľahlivá výmena správ medzi zariadeniami, a to s využitím minimálnych zdrojov [11]. Funguje nad Device-to-Device modelom a pracuje s publish-subscribe mechanizmom, ktorý funguje na jednoduchom posielaní a prijímaní správ s minimálnym oneskorením medzi zariadeniami. Vďaka tomu je ideálny na využitie pri zariadeniach a rôznych senzoroch, ktoré majú obmedzené zdroje alebo obmedzené pripojenie na sieť.

### Komunikácia

Spojenie medzi zariadeniami zabezpečuje MQTT sprostredkovateľ - server (anglicky „broker“), ktorý sa správa ako centrálny uzol v systéme. A zohráva dôležitú úlohu v riadení komunikačného toku medzi klientmi. Zabezpečuje spoľahlivé doručenie správ, filtráciu na základe odoberanej témy a určuje, ktorému klientovi má byť daná správa preposlaná. Tak tiež zabezpečuje autorizáciu a autentizáciu klientov, pomocou prihlasovacích údajov a TLS šifrovania [16].

K tomuto serveru sa následne pripájajú klienti, ktorí sa zvyčajne označujú ako odosielateľ (anglicky „Publisher“), ktorý dáta odosiela alebo príjemca (anglicky „Subscriber“), ktorý dáta prijíma. Nie je to, ale nutnou podmienkou, keďže klient môže dáta zároveň odosielať aj prijímať. Klient môže byť akékoľvek zariadenie, ktoré využíva sieťový protokol TCP/IP a dokáže implementovať funkcie MQTT klienta. Medzi takéto zariadenia môžu patriť jednoduché mikrokontroléry, inteligentné zariadenia, ale aj počítače, servery a iné prístroje [16].



Obr. 2.4: MQTT komunikácia medzi zariadeniami.

Model správ je založený na témach a odberoch. Téma je refazec, ktorý udáva cestu, kde sa daná správa publikuje a odoberá. Témy sú hierarchické a môžu obsahovať viacero úrovní, ktoré sú oddelené pomocou lomítka, napríklad `system/pv_generation/power`. Odber sa následne využíva na určenie tém, z ktorých chce klient prijímať správy. Po tom, čo sa klient prihlási k odberu témy, server sleduje odoberanú tému a preposiela všetky správy, ktoré sú v danej téme publikované, prihlásenému klientovi [15].

## Zástupné znaky

Ak klient potrebuje odoberať viac tém naraz alebo nevie vopred určiť celú hierarchiu témy, môže využiť zástupné znaky (anglicky „wildcards“). MQTT poskytuje dva typy týchto znakov, a to jedno-úrovňové a viac-úrovňové. Jedno-úrovňový zástupný znak je reprezentovaný symbolom plus (+) a umožňuje nahraďiť jednu úroveň témy. Po prihlásení k odberu sa porovná každá téma, ktorá obsahuje ľubovoľný reťazec na mieste zástupného znaku. Výsledný reťazec by mohol vyzeráť takto `system/+power`. Viac-úrovňový zástupný znak pokrýva viacero úrovní témy, je reprezentovaný znakom mriežky (#) a musí byť umiestnený ako posledný znak v téme. Keď sa klient prihlási k odberu témy s viac-úrovňovým zástupným znakom, bude dostávať všetky správy tém, ktoré sa začínajú vzorom, pred znakom mriežky. Takýto reťazec môže vyzeráť ako `system/#` [17].

## 2.4 API

API rozhranie (anglicky „application programming interface“) predstavuje súbor pravidiel a protokolov, ktoré umožňujú softvérovým aplikáciám vzájomne komunikovať a zdieľať údaje. Tieto rozhrania sú dôležité k zjednodušeniu a urýchleniu vývoja aplikácií a softvéru, pretože umožňujú integráciu existujúcich údajov z iných aplikácií, bez potreby vytvárania nových komponentov a prístupov. Vďaka tomu môžu tretie strany sprístupniť len nevyhnutné údaje systému a zvyšné dáta zostávajú skryté, čo pomáha pri zabezpečení systému a obmedzení veľkosti prenesených dát. Táto časť a nasledujúce časti boli prevzaté z [2].

### Funkcia

Fungovanie si zakladá na princípe požiadavka-odpoveď, kde medzi sebou komunikujú dve strany a API funguje ako spoj medzi nimi. Klient posíla požiadavky na danú API, server následne požiadavky spracuje a vráti odpoveď. Odpoveď môže byť kladná s vyžiadanými dátami alebo záporná, ak napríklad, klient nepoužije správne údaje.

### Prístup

Existuje niekoľko prístupov k návrhu API, medzi ktoré patrí napríklad REST. Tento prístup sa často považuje za jednoduchší a flexibilnejší, keďže prístup komunikuje pomocou HTTP požiadavkov a podporuje rôzne formáty správ, ako je, napríklad obyčajný text, YAML, JSON a ďalšie. Je dátovo orientovaný a definuje štyri základné metódy pre prístup, tiež známe ako *CRUD* operácie. Tieto operácie podporujú vytváranie (POST), čítanie (GET), aktualizáciu (PUT) a mazanie (DELETE) dát.

### Zabezpečenie

K dôležitej súčasť API patrí aj jeho zabezpečenie. Je to proces overenia identity používateľa, ktorý zasiela požiadavky na danú API. Existuje viacero spôsobov overenia, medzi ktoré patria: základné overenie pomocou protokolu HTTP, použitie API kľúča, JWT (*JSON Web Token*) a OAuth. Každá z týchto metód má svoje výhody a nevýhody, ako aj optimálne prípady použitia. Napriek tomu majú všetky metódy spoločný cieľ a to ochrániť citlivé údaje a zabezpečiť aby nedošlo k zneužitiu API [1].

## 2.5 Energia a termodynamika

Nasledujúce časti sa venujú energii, ktorá popisuje definíciu energie, jej jednotky, premeny, dôležité fakty. A termodynamike, ktorá popisuje vzťahy medzi veličinami týkajúcich sa tepelnej sústavy. Tieto vzorce a vety sú neskôr využité pri výpočtoch potrebného času na vyhriatie vody v bojleri a k lepšiemu porozumeniu fungovania systému. Treba ale myslieť na to, že nasledujúce vety a vzorce popisujú ideálny systém a nepočítajú so stratami.

Tieto časti boli prevzaté zo zdrojov [20], [9] a boli vynechané nedôležité informácie, ktoré sa netýkajú tejto práce.

### Energia

Energiu môžeme definovať ako *skalárnu fyzikálnu veličinu, ktorej hodnota je určená stavom fyzikálnej sústavy (objektu)*. Podľa charakteru pôsobiacich síl hovoríme o energii *mechanickej, elektrickej, chemickej, jadrovej a pod.* Ďalej je dôležité spomenúť že *prenos energie medzi objektom a jeho okolím môže byť sprostredkovaný silovým pôsobením alebo tepelnou výmenou*. To v jednoduchosti znamená, že energiu môžeme „uložiť“ vo forme teplej vody v bojleri, ktorý pomocou špirály ohrieva vodu a tým jej predáva energiu.

### Zákon zachovania energie a 1. Termodynamický zákon

Zákon o zachovaní energie nám hovorí, že

*Energia nevzniká a nezaniká, ale sa len premieňa z jednej formy energie na druhú formu energie.*

A 1. Termodynamický zákon znie

*Vnútoraná energia  $U$  systému narastie, ak mu dodá okolie teplo  $Q$  a klesne, keď systém vykoná nejakú prácu  $W$ .*

Tento zákon môžeme vyjadriť vzťahom

$$\Delta U = Q - W \tag{2.1}$$

Inak povedané, *ak sústave dodáme teplo a vykonáme na nej nejakú prácu, jej vnútoraná energia narastie. Ak sústava odovzdáva teplo alebo koná prácu, jej vnútoraná energia klesá.*

Na základe týchto zákonov máme možnosť energiu premieňať a ukladať v rôznych formách, v našom prípade vo forme elektrickej energie v batériách a vo forme teplej vody v bojleri.

### Premena slnečnej energie na elektrickú energiu

Slnečná energia je typ energie, ktorá dopadá na Zem vo forme tepelnej a svetelnej energie a je získaná zo Slnka. Svetelná energia môže byť následne premenená na elektrickú energiu pomocou fotovoltických článkov.

## Výkon a práca

Tú istú prácu môžeme vykonať za rôzny čas. Mierou toho, ako „rýchle“ koná sila prácu je fyzikálna veličina, ktorá sa nazýva výkon  $P$ . Jednotkou výkonu v SI je watt ( $[P] = W = J/s$ ). A pri stálom výkone  $P$  môžeme prácu vyjadriť ako súčin výkonu a času

$$W = Pt \quad (2.2)$$

a z tohto vzťahu nám pre prácu vyplýva bežne používaná jednotka pri spotrebe elektrickej energie – kilowatthodina (kilowatthodina =  $1kWh = 3,6 \times 10^6 J = 3,6MJ$ ).

Ak potrebujeme zistiť koľko časových jednotiek  $t$  bude treba na vykonanie práce  $P$ , je možné vzorec 2.2 upraviť nasledovne

$$t = \frac{W}{P} \quad (2.3)$$

Takto upravený vzorec, môžeme neskôr využiť pri výpočte potrebného času na vyhriatie vody v bojleri.

## Teplo a tepelná kapacita

Teplo môžeme definovať ako formu prenosu energie. A pri styku dvoch telies s rozdielnou teplotou sa kinetická energia molekúl teplejšieho telesa odovzdáva molekulám s nižšou kinetickou energiou chladnejšieho telesa, čo vnímame ako prenos tepla. Pri prenose tepla sa určuje množstvo tepelnej energie, ktoré je dodané, alebo odobraté určitému telesu. Toto množstvo tepelnej energie sa zvykne označovať  $Q$ .

Potrebné množstvo tepla, či už odovzdaného alebo prijatého, pri zmene teploty o hodnotu  $\Delta T$  môžeme vyjadriť v tvare

$$\Delta Q = cm\Delta T \quad (2.4)$$

kde  $c$  je merná tepelná kapacita látky,  $m$  je hmotnosť telesa a  $\Delta T$  je teplotný rozdiel. Jednotkou tepla je jeden joule.

Tabuľka 2.1: Merná tepelná kapacita vody

Látka	$c(J \cdot kg^{-1} \cdot K^{-1})$
voda	4186

Merná tepelná kapacita udáva množstvo tepla, ktoré je potrebné na ohriatie jedného kilogramu látky o jeden teplotný stupeň.

Následne je možné pomocou týchto informácií vypočítať potrebný čas  $t_b$  v sekundách na vyhriatie vody v bojleri o  $\Delta T$  stupňov ako

$$t_b = \frac{cm_b(t_\alpha - t_\beta)}{P_b} \quad (2.5)$$

kde  $t_b$  značí potrebný čas v sekundách,  $c$  je merná tepelná kapacita vody,  $m$  je objem bojleru v litroch (podľa rovnice 2.4,  $m$  značí hmotnosť telesa, ale v tomto prípade je možné hmotnosť zameniť za objem, keďže 1 liter vody váži približne 1 kilogram),  $t_\alpha$  značí aktuálnu teplotu vody,  $t_\beta$  značí konečnú teplotu vody a  $P_b$  je výkon špirály v bojleri.

## 2.6 Štatistika

Štatistiku môžeme definovať ako *jazyk pre zhromažďovanie údajov, manipuláciu s nimi a ich interpretáciu*. Úlohou štatistiky pri hodnotení informácií je *spracovanie hromadných pozorovaní, ich interpretácia a analýza*. Nasledujúca časť popisuje štatistickú metódu, ktorá bola využitá pri spracovávaní historických dát v systéme a k zlepšeniu odozvy systému na aktuálne stavy. Táto časť a nasledujúce časti sú prevzaté zo zdrojov [7] a [8].

### Štatistický súbor

Štatistický súbor tvorí skupina štatistických jednotiek, ktoré majú určité spoločné charakteristiky a definujú ho podľa časových a priestorových parametrov. Každý súbor má svoj rozsah a obsah. Rozsah súboru je stanovený počtom jednotiek, ktoré sú súčasťou daného súboru a označujeme ho  $n$ . Obsah súboru je určený štatistickými znakmi.

Základný súbor tvorí skupina všetkých príslušných štatistických jednotiek, na ktorých sa sleduje hodnota štatistického znaku. A výberový súbor je zložený z výberu štatistických jednotiek, ktoré reprezentujú významnú podmnožinu základného súboru.

### Štatistické triedenie

Prvý krok pri spracovaní štatistických údajov je roztriedenie daných dát. Aby sme dostali *prvotnú tabuľku* s údajmi, musíme zapísať číselné hodnoty v poradí v akom sú získavané. Následne tieto hodnoty roztriedime podľa ich veľkosti, vzostupne, teda od najmenšieho prvku po najväčší a zapíšeme ich toľko krát, koľko krát sa v prvotnej tabuľke vyskytujú, tým dostaneme *variačný rad*.

### Medián

*Medián chápeme ako prostrednú hodnotu štatistického súboru. Medián je číslo, ktoré leží uprostred variačného radu, označujeme  $Me = \tilde{x}$ . Ak  $n$  je rozsah štatistického súboru, potom platí*

$$\text{pre } n \text{ párne: } \tilde{x} = \frac{x_{(\frac{n}{2})} + x_{(\frac{n+2}{2})}}{2} \text{ a pre } n \text{ nepárne: } \tilde{x} = x_{(\frac{n+1}{2})} \quad (2.6)$$

### Kvantily

O kvantiloch môžeme vo všeobecnosti hovoriť ako o zovšeobecnení mediánu. Keďže sú to hodnoty, ktoré delia variačný rad na dve časti. Najbežnejšie sa používa 50%-ný kvantil, ktorému sa tiež hovorí *medián*. Označuje sa ako  $x_{0,50}$  alebo  $\tilde{x}$  a rozdeľuje variačný rad na dve rovnaké časti. Medzi zvyšné bežne používané kvantily patria *kvartily*, *decily* a *percentily*.

### Percentil

Percentil je teda len variácia kvantilu, ktorý udáva že aspoň  $\alpha \cdot 100\%$  dát nadobúda hodnotu menšiu rovnú, ako  $\alpha$ -kvantil. Vďaka tomu je možné z štatistického súboru „odfiltrovať“ hodnoty, ktoré neodzrkadľujú aktuálnu spotrebu energie v domácnosti (viac v kapitole 5.2). Inak povedané, výsledná hodnota nebude ovplyvnená nárazovým zaťažením elektrickej siete, čo sa napríklad v prípade priemernej hodnoty môže stať, ak dôjde k veľmi vysokému nárastu.



## Kapitola 3

# Prieskum existujúcich možností

V tejto kapitole sa venujem existujúcim riešeniam týkajúcich sa bojlerov, fotovoltických systémov a doplnkov do Home Assistanta. Kapitola popisuje ich funkčnosť, možnú integráciu do IoT a riešenia, ktoré mi boli inšpiráciou pri vytváraní vlastnej integrácie do systému Home Assistant.

### 3.1 Integrácia bojleru

V dnešnej dobe máme veľmi rozmanité možnosti, čo sa týka výberu bojleru. Od tých úplne základných, ktoré poskytujú len jednoduché nahrievanie vody na nastavenú teplotu, až po „inteligentné“ bojler, ktoré môžu disponovať rôznymi funkciami, ako je sledovanie HDO, učiace mechanizmy na optimalizovanie spotreby, pripojenie pomocou Bluetooth alebo Wi-Fi a mnohé ďalšie. Ale žiadne z týchto zariadení neposkytuje priamo pripojenie do centrálného systému alebo integráciu s iným systémom, čo môže byť problém, ak chceme toto zariadenie integrovať do inteligentnej domácnosti. A v prípade tejto práce, jeho ovládanie na základe dostupnej energie z fotovoltického systému.

Je niekoľko možných riešení tohto problému. V prípade obyčajných bojlerov, je možné dokúpiť centrálny termostat, ovládací modul alebo využiť riešenie pomocou inteligentnej zásuvky[10]. Pomocou týchto riešení je možné daný bojler ovládať, alebo pridať do centrálného systému.

V prípade inteligentných bojlerov je možné využiť, napríklad mikrokontrolér a s jeho pomocou zabezpečiť komunikáciu medzi bojlerom a centrálnym systémom. Vďaka tomu je možné bojler ovládať vzdialene, integrovať ho s ostatnými zariadeniami a systémami alebo využiť rôzne automatizácie.

Ak je zámerom ohrev vody v bojleri pomocou prebytkovej energie, je možné využiť aj niektoré z komerčných riešení, medzi ktoré patrí napríklad WattRouter, ktorý dokáže presmerovať prebytky z fotovoltických panelov do zariadenia, v tomto prípade do elektrického bojleru a tým zabezpečiť ohrev vody.

## 3.2 Fotovoltický systém

Solárne systémy sa stávajú čoraz bežnejšou súčasťou života a čím ďalej tým viac domácností, firiem a koncových užívateľov si dáva takéto systémy inštalovať. To nám potvrdzuje aj tlačová správa od SAPI, v ktorej sa píše, že *celkový nárast inštalovanej kapacity bol až o 400% vyšší v porovnaní s rokom 2022. A najväčší podiel na medziročnom prírastku mali malé fotovoltické rezidenčné zdroje, ktoré predstavovali 92% z celkového počtu inštalácií a 52% celkového nárastu inštalovanej kapacity* [13].

Takéto systémy delíme na 3 typy:

- On-grid fotovoltický systém
- Hybridný fotovoltický systém
- Off-grid fotovoltický systém

Tieto typy sa ďalej rozdeľujú na jednofázové a viacfázové, podľa toho koľko fáz fotovoltický systém dokáže podporovať. A každý z týchto typov má svoje výhody a nevýhody a má svoje určenie. Ďalej sa zaoberám iba jednofázovým hybridným fotovoltickým systémom, ktorý je využitý pri vypracovaní tejto práce.

Tento typ systému obsahuje samotné fotovoltické panely, menič a batérie. Poprípade zásuvku (vidno na pravom obrázku 5.2, v strede), ktorá je vyhradená iba na prebytky z fotovoltického systému. Funguje nezávisle od elektrickej rozvodnej siete, ktorá pokrýva zvyšné dve fázy, poprípade nedostatok energie z fotovoltických panelov. Prebytky sú ukladané do batérií, z ktorých môže byť neskôr pokrytý nedostatočný výkon generácie. Prebytky, ale môžu byť uložené aj v inej forme, ako je elektrická energia a tou je teplo (viz definícia 2.5). A to je využité v tejto práci, kde je elektrický bojler nahrievaný pomocou prebytkovej energie a nedostatok (poprípade výkyvy v generácií alebo nárazová spotreba, ktorá je väčšia ako generácia) je kompenzovaný z batérií. Vďaka tomu je možné plynulejšie nahrievanie vody, keďže bojler nie je konštantne vypínaný a zapínaný ak generácia na chvíľu klesne pod výkon špirály, ako je tomu pri „prebytkovej zásuvke“.

### Integrácia do IoT

Samotný fotovoltický systém nemusí byť nijak inteligentný a nemusí mať možnosť integrácie do centrálného systému. Aby takáto integrácia bola možná, je nutné využiť zariadenie, ktoré dokáže poskytnúť prístup k stavom systému. Takéto zariadenie môže byť napríklad Cerbo GX od spoločnosti Victron Energy, ktoré disponuje všetkými potrebnými funkciami v jednom zariadení. Alebo je možné využiť open source softvér Venus OS<sup>1</sup>, ktorý je možné nainštalovať na Raspberry Pi alebo iný mikropočítač. Vďaka Venus OS, je možné komunikovať so systémom a sledovať jeho stav pomocou MQTT (kapitola 2.3) a tým aj jednoducho integrovať systém do Home Assistanta. Tento spôsob využívam na sledovanie aktuálnej spotreby v domácnosti, aktuálnej generácie, stavu batérií a energie prijatej/odoslanej z/do rozvodnej siete (ostatné stavy systému sú zobrazené v Home Assistantovi).

Pri použití zariadenia Cerbo GX, poskytuje Victron Energy prístup k portálu, kde je možné sledovať výkon fotovoltických panelov, aktuálnu spotrebu a ďalšie informácie o systéme. A prístup k VRM API (kapitola 2.4), cez ktorú je možné zistiť predpoveď generácie, ktorá bola využitá pri implementácii logiky, ktorej sa venuje kapitola 5.

<sup>1</sup><https://github.com/victronenergy/venus>

### 3.3 Home Assistant integrácia

Na komunitnom fóre Home Assistanta je možné nájsť veľké množstvo príspevkov<sup>234</sup>, kde komunita vytvára rôzne automatizácie, ktoré pracujú s bojlerom, jeho integráciou, nahrievaním, zapínaním a vypínaním podľa nejakých parametrov a podobne. Väčšina z týchto príspevkov pracuje s obyčajnými bojlermi, do ktorých zaviedli teplotné senzory a bojler ovládajú pomocou inteligentnej zásuvky. Takéto riešenia sú použiteľné, ale vyžadujú veľa rôznych nastavení a úprav, aby konfigurácia fungovala. Taktiež sú implementované ako jednoduché automatizácie, ktoré disponujú iba základnými možnosťami a je v nich zložité alebo nemožné implementovať nejaké pokročilejšie funkcie.

Takáto implementácia môže byť vo forme automatizácie, v ktorej si užívateľ môže nastaviť rôzne podmienky, na ktoré bude Home Assistant automaticky reagovať, napríklad ak entita *generácia* dosiahne hodnotu 3000, zapni entitu *bojler*. Alebo skript, ktorý je nutné spustiť ručne, poprípade automaticky pomocou automatizácie. Vďaka skriptu je možné vykonávať komplexnejšie akcie z viacerých automatizácií. Taktiež je možné využiť blueprints, ktoré fungujú ako šablóna pre automatizácie a skripty. Pomocou nich si užívateľ nemusí nastavovať celú logiku, len si pridá vybraný blueprint a nastaví potrebné entity.

Implementácia pomocou automatizácie alebo skriptu môže byť jednoduchá a je možné ju vytvoriť aj s minimálnymi programátorskými skúsenosťami, ale nemusí byť možné v nej implementovať nejaké zložitejšie, alebo pokročilejšie funkcie. A to ma inšpirovalo k vytvoreniu samostatnej integrácie, ktorá pomocou jednoduchej inštalácie, umožní automatické a komfortné ovládanie bojleru. Po pridaní integrácie užívateľ vyplní potrebné informácie a entity a komponent sa sám postará o vytvorenie zvyšných entít, naviazanie komunikácie a nastavenie ostatných parametrov, aby mohol fungovať plne automaticky. Takáto implementácia umožňuje vytvoriť logiku presne na mieru mojim potrebám a neskôr z toho spraviť univerzálny nástroj, ktorý nebude fungovať iba s konkrétnym bojlerom a fotovoltickým systémom.

---

<sup>2</sup><https://community.home-assistant.io/t/smart-immersion-heating-for-domestic-hot-water/536281>

<sup>3</sup><https://community.home-assistant.io/t/how-ha-is-helping-heat-my-water-for-free/446782>

<sup>4</sup><https://community.home-assistant.io/t/victron-venus-os-with-mqtt-sensors-switches-and-numbers/527931>

# Kapitola 4

## Návrh riešenia

Táto kapitola sa venuje všetkým aspektom, ktoré som musel pri vytváraní integrácie zvážiť, aby bolo možné vytvoriť funkčný návrh. Medzi ktoré patrí návrh integrácie (kapitola 4.1), ktorý sa venuje možnostiam a funkciám, ktoré by mala integrácia obsahovať. Prevedeniu integrácie v kapitole 4.2, ktorá opisuje, ako zabezpečiť univerzálnosť, ovládanie a komunikáciu. A samotnej logike (kapitola 4.3), v ktorej je vysvetlené ako by malo fungovať ohrievanie vody v bojleri, nočný ohrev a spracovanie neočakávaných udalostí.

### 4.1 Návrh integrácie

Vytvoriť návrh na univerzálnu integráciu, pomocou ktorej bude možný komfortný ohrev vody v bojleri, môže byť komplexná záležitosť, keďže je nutné brať ohľad na veľké množstvo faktorov, ktoré hrajú dôležitú rolu.

Medzi najdôležitejšie faktory patria:

- Čo od toho očakávajú členovia v domácnosti a, ako si predstavujú, že by to malo fungovať?
- Aké funkcie a možnosti využívajú ľudia, ktorí implementovali podobné riešenia?
- Ako zabezpečiť univerzálnosť riešenia?
- Ako zabezpečiť jednoduchú inštaláciu a ovládanie?
- Ako zabezpečiť komunikáciu medzi zariadeniami (typy, protokoly)?
- Ako spracovať neočakávané udalosti (výpadok komunikácie, nedostupnosť distribučnej siete)?

Pomocou týchto otázok bolo možné vymyslieť a implementovať funkčný návrh, ktorý zodpovedá daným požiadavkám.

## Očakávania od integrácie

Ako prvé som zhodnotil požiadavky a očakávania od členov v domácnosti a pridal k nim funkcie z už existujúcich riešení, keďže sa tieto požiadavky prekrývali. Medzi tie najhlavnejšie patrili:

- Voda by mala mať ráno „nejakú“ minimálnu teplotu, kvôli rannému sprchovaniu, umývaniu a podobne (nočný ohrev).
- Všetky informácie o systéme a ovládanie by malo byť dostupné z jednej aplikácie.
- Batérie by mali mať prednosť, pred nahriatím vody.
- Batérie by sa mali udržiavať na určitom percente nabitia, nemali by pod toto percento klesnúť.
- Bojler by sa mal vypnúť pri vysokej záťaži siete, aby sa zbytočne nevybíjali batérie.
- Pri vysokom odbere energie z distribučnej siete vypnúť bojler.
- Zakomponovanie predikcií generácie fotovoltických panelov.

## 4.2 Prevedenie integrácie

### Univerzálnosť

Dosiahnuť univerzálnosť integrácie, nie je jednoduché, keďže každý systém môže vyzeráť inak, poskytovať iné informácie alebo sa inak správať. Preto som sa rozhodol využiť pri inštalácii výberu z dvoch možností. Užívateľ si môže vybrať medzi automatickou alebo manuálnou inštaláciou. Možnosť výberu je ako pre fotovoltický systém tak aj pre bojler.

Automatická inštalácia zabezpečí, že si komponent sám vytvorí všetky potrebné entity, naviaže komunikáciu a zabezpečí celkové fungovanie systému. Táto možnosť funguje ak užívateľ využíva rovnaké zariadenia ako naša inštalácia, na ktorú bola práca primárne dizajnovaná (spomenuté v kapitole 5.1).

Ak využíva iné zariadenia, môže si vybrať manuálnu možnosť, v ktorej si užívateľ sám vyberie potrebné entity, ktoré sú vyžadované k správne fungovaniu komponentu. Táto inštalácia je komplexnejšia o veľké množstvo nastavení od automatickej inštalácie, ale zabezpečí univerzálnejšie fungovanie.

### Ovládanie

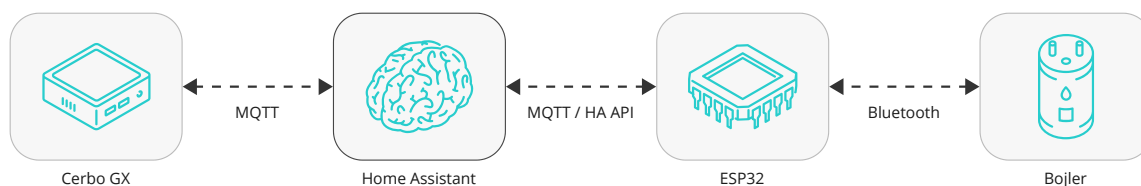
Ak je všetko správne nastavené a užívateľ si vybral správne entity, komponent po inštalácii vyzerá rovnako, s rovnakými možnosťami pre automatickú aj manuálnu inštaláciu. Mal by disponovať stavovou entitou, ktorá zobrazuje aktuálny stav systému. Možnosťou zmeny jeho stavu (manuálny, automatický alebo vypnutý stav), nastavenie teploty výhrevu, zapnutie, alebo vypnutie nočného nahrievania s možnosťou zmeny tejto teploty a nastavenie času, v akú hodinu má byť táto teplota dosiahnutá. Popríklad zobrazenie predpovede generácie fotovoltických panelov na dnešný a zajtrajší deň (v prípade využitia VRM API).

Ak bola inštalácia prevedená cez automatickú možnosť, sú navyše vytvorené všetky entity, ktoré sa týkajú fotovoltického systému. V opačnom prípade si tieto entity užívateľ vytvára/pridáva sám ešte pred samotnou inštaláciou integrácie.

## Komunikácia medzi zariadeniami

Túto časť je tiež možné „pomyselne“ rozdeliť na dva typy – automatickú a manuálnu. Keďže komunikácia sa bude odvíjať od typu inštalácie. Automatická počíta s využitím zariadenia Cerbo GX (poprípade iného zariadenia, na ktorom beží Venus OS) a komunikácia medzi fotovoltickým systémom a Home Assistantom prebieha pomocou MQTT protokolu (kapitola 2.3). Medzi bojlerom a Home Assistantom zabezpečuje komunikáciu *ESPHome component for Dražice OKHE smart water heater*<sup>1</sup>. V opačnom prípade si užívateľ môže zvoliť iný druh komunikácie a komponent priamo interaguje s entitami.

Pri automatickom type inštalácie (pre bojler aj fotovoltický systém), bude komunikácia vyzeráť takto:



Obr. 4.1: Schéma komunikácie medzi zariadeniami.

## 4.3 Návrh logiky

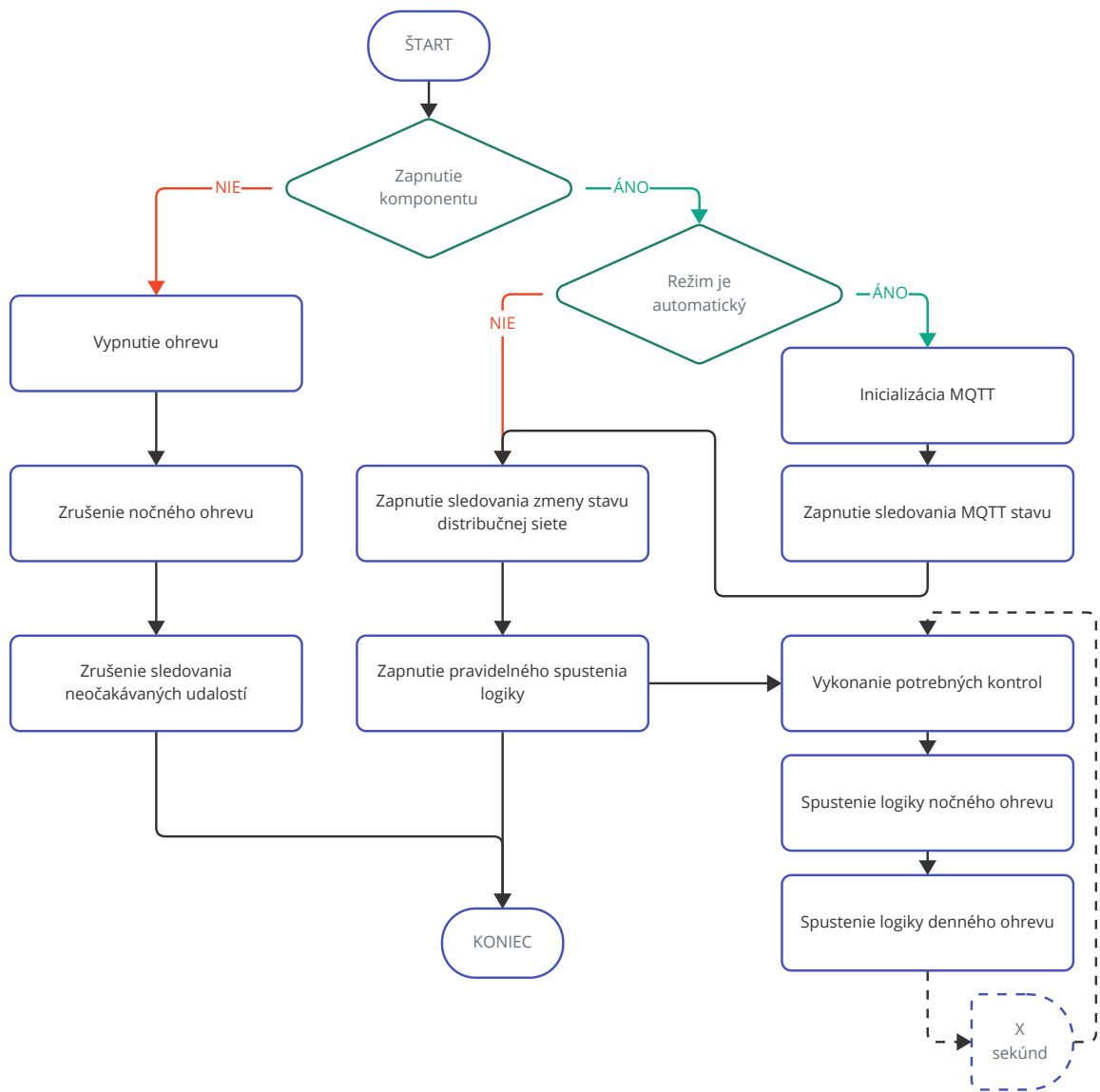
Logika sa skladá z niekoľkých dôležitých častí, medzi ktoré patrí plánovanie nočného ohrevu, spracovanie neočakávaných udalostí (napríklad nedostupnosť MQTT, výpadok distribučnej siete) a samotný ohrev bojleru pomocou prebytkovej energie z fotovoltických panelov (poprípade pokrytie nedostatku z batérií).

Samotný komponent má 3 režimy – vypnutý, manuálny a automatický režim. Pomocou týchto režimov je možné komponent vypnúť alebo zmeniť jeho fungovanie.

Pri vypnutí komponentu by sa mal vypnúť ohrev, zrušiť naplánovaný nočný ohrev a sledovanie neočakávaných udalostí. Ak bola inštalácia prevedená pomocou automatickej možnosti, entity z fotovoltického systému budú aj naďalej aktualizované (je, teda možné sledovať stav fotovoltického systému, aj bez toho, aby komponent bol spustený).

Medzi automatickým a manuálnym režimom je len minimálny rozdiel, tento rozdiel je v nočnom ohreve vody. Vtedy automatický režim plánuje nočný ohrev podľa predikcie generácie a berie ohľad na to či bude počas dňa dostatok energie na nahriatie vody na určitú teplotu a na nabitie batérií. Tento režim je určený do domácností, ktoré majú viac zdrojov teplej vody, napríklad plynový kotol a elektrický bojler. Keďže sa môže stať, že sa nočný ohrev nenaplánuje, a teda ráno nebude teplá voda (v prípade využitia iba elektrického bojleru). Naopak pri manuálnom režime sa nočný ohrev plánuje vždy a neberie ohľad na predpoveď generácie. Tým bude zabezpečené, že voda bude vyhriata aspoň na minimálnu nastavenú teplotu. Automatický režim funguje iba v prípade využitia VRM API, bez údajov k API je k dispozícii iba manuálny režim.

<sup>1</sup><https://github.com/LubosD/esphome-smartboiler>



Obr. 4.2: Diagram návrhu logiky.

## Ohrev vody v bojleri

Na základe kritérií z kapitoly 4.1 bolo možné vypracovať logiku, ktorá optimalizuje proces ohrevu vody v bojleri. Základnou myšlienkou logiky je prednosť nabíjania batérií, ktoré sú udržiavané na požadovanom (užívateľom nastavenom) rozsahu nabitia. A využitie, čo najviac energie z fotovoltických panelov.

Tým, že sa budú batérie udržiavať v požadovanom rozsahu, môže byť pomocou nich pokrytá nedostatočná generácia. To znamená, že sa ohrev vody spustí až po tom, čo batérie dosiahnu hornú hranicu požadovaného rozsahu a zastaví pri klesnutí pod dolnú hranicu rozsahu. Vďaka tomu je možné spustiť ohrev skôr, ako fotovoltické panely dosiahnu minimálne generáciu rovnú výkonu bojleru. Takýto výkon, ale panely vôbec nemusia dosiahnuť, ak nie je pekné počasie to, ale neznamená, že nebudú produkovať dostatok energie, aby vznikli prebytky. Príkladom môže byť ak panely budú generovať 1300 W, spotreba v domácnosti je približne 200 W a výkon špirály v bojleri je 1500 W. Pri takomto scenári by množstvo prebytkov dosiahlo 1100 W, ale bojler sa nezapne.

Ďalším dôležitým aspektom je sledovanie zariadenia siete (inak povedané spotreby v domácnosti). Táto spotreba vrátane spotreby bojleru by ideálne nemala byť vyššia, ako je generácia fotovoltických panelov, čo, ale nie je možné dosiahnuť vždy (napríklad v skorších hodinách, pri zlom počasí a podobne), mala by byť, ale dostatočná, aby pokryla aspoň nejakú časť tejto spotreby. Taktiež by sa bojler nemal zbytočne vypnúť a následne zapnúť, ak spotreba na chvíľu narastie a znovu klesne (v prípade našej domácnosti sú nárazové zvýšenia spotreby spôsobené ponorným čerpadlom v studni).

A k poslednej časti patrí aktívne sledovanie prijatej energie z distribučnej siete, ak by táto hodnota stúpila nad nastavený limit, bojler by mal pozastaviť ohrev. Keďže dochádza k odberu energie z distribučnej siete namiesto toho, aby túto spotrebu pokrýval fotovoltický systém. Táto podmienka funguje skôr ako „bezpečnostný“ prvok v systéme, keď menič nestihne zareagovať alebo je prekročený jeho maximálny výkon.

## Nočný ohrev vody v bojleri

Ako už bolo spomenuté v úvode tejto kapitoly, nočný ohrev pracuje v dvoch režimoch, a to automatickom a manuálnom. Pri automatickom režime by sa nočný ohrev mal naplánovať iba v prípade dostatku energie na nahriatie vody počas dňa a nabitia batérií na požadovanú úroveň. Keďže predikcie nemusia byť úplne správne, užívateľ má možnosť si nastaviť variačnú hodnotu teploty. Následne sa táto hodnota odčíta od nastavenej teploty výhrevu a na základe tejto novej hodnoty sa bude počítať potrebná energia na výhrev vody počas dňa. Manuálny režim by mal nočný ohrev plánovať vždy, aby bola voda vyhriata aspoň na minimálnu teplotu.

Aby sa voda stihla ohriať do určitej hodiny (času nastaveného užívateľom), je potrebné ohrev naplánovať dostatočne vopred. To by malo zabezpečiť niekoľko plánovaní, kde každé z nich pokryje určitú časť logiky. Prvá časť sa spustí pri prvom spustení komponentu, vypočíta potrebný čas na ohrev vody z 1 stupňa na užívateľom požadovanú nočnú teplotu. A naplánuje spustenie druhej časti podľa vypočítaného času. Druhá časť sa spustí automaticky v naplánovaný čas a zameriava sa na presnejší výpočet potrebného času na základe minimálnej teploty vody v bojleri. Na tento čas naplánuje tretiu časť, ktorá bude spúšťať ohrev vody na základe aktuálnej teploty v bojleri.

Príkladom môže byť: Užívateľ si nastaví nočný ohrev na 40°C, ranný čas na 6:00 a spustí komponent, prvá časť vypočíta, že k ohrevu vody z 1°C na 40°C potrebuje 4 hodiny, naplánuje spustenie druhej časti v 2:00. O druhej ráno sa spustí druhá časť, ktorá zistí, že



minimálna teplota v bojleri je 30°C následne vypočíta, že ohrev z 30°C na 40°C bude trvať hodinu a naplánuje finálnu časť na 5:00. O piatej ráno sa spustí posledná časť, skontroluje či teplota vody nie je väčšia, ako požadovaná teplota a spustí ohrev vody, ktorý bude následne o 6:00 ukončený.

## **Spracovanie neočakávaných udalostí**

Aby bol komponent spoľahlivý a schopný adekvátne reagovať na neočakávané udalosti, je dôležité navrhnúť logiku, ktorá bude pripravená tieto udalosti správne spracovať. Medzi najdôležitejšie udalosti patrí výpadok MQTT komunikácie (napríklad pri výpadku Wi-Fi, reštarte zariadenia, strate paketov, atď.), výpadok distribučnej siete a výpadok komunikácie s bojlerom.

V prípade výpadku MQTT komunikácie by mal komponent informovať užívateľa o tejto skutočnosti, a ak sa komunikácia neobnoví do určitého času, mal by vypnúť bojler a zobrazíť chybovú hlášku. Nie je nutné vypínať bojler hneď pri výpadku, keďže sa môže jednať iba o krátky výpadok, napríklad kvôli nestabilnej internetovej sieti. Ak dôjde k znovu obnoveniu komunikácie, komponent sa môže vrátiť do rovnakého stavu v akom bol pred výpadkom a ďalej fungovať.

Ak dôjde k výpadku distribučnej siete, komponent by mal neodkladne vypnúť ohrev vody v bojleri a zobrazíť chybovú hlášku. Túto udalosť považujem za kritickú, keďže pri takejto situácii môže dôjsť k vybitiu batérií, ktoré sú v tejto situácii potrebné k napájaniu domácnosti. Preto sa komponent vypne a po obnovení distribučnej siete sa neuvedie do predchádzajúceho stavu. Až užívateľ si môže komponent znovu zapnúť, keď to uvaží za vhodné. Tu je treba myslieť aj na to, že zariadenia na ktorých beží Home Assistant a komponent na komunikáciu s bojlerom, musia byť pripojené na fázy, ktorá je podporovaná fotovoltickým systémom alebo musia mať záložný zdroj. Inak sa nedokáže táto udalosť spracovať.

Pri výpadku komunikácie s bojlerom, nie je možné situáciu vyriešiť ideálne, keďže sa bojler nebude dať ovládať. Komponent môže iba zobrazíť chybovú hlášku a po znovu nadviazaní komunikácie pokračovať v činnosti.

## Kapitola 5

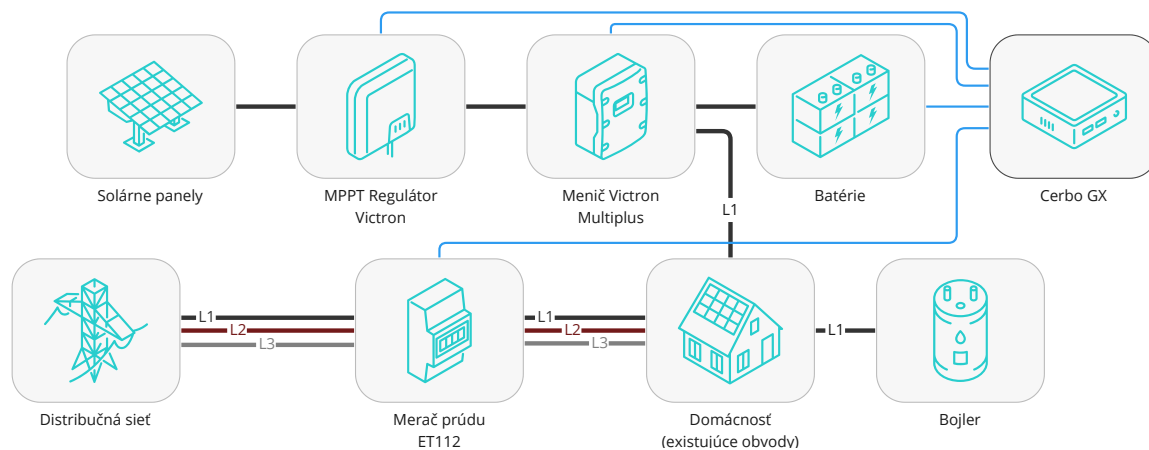
# Implementácia

Kapitola opisuje proces implementácie integrácie a jej fungovanie v rámci systému Home Assistant. V kapitole 5.1 sú popísané zariadenia a technológie, ktoré boli pri implementácii využité. Následne sa kapitola 5.2 venuje samotnej implementácii návrhu, ktorá opisuje rozšírenie vlastností entít, logiku inicializácie komponentu, hlavnú logiku komponentu, nočný ohrev vody a spracovanie neočakávaných udalostí. A v záverečnej časti kapitoly (5.3) je popísaný priebeh testovania integrácie, ktorý overuje jej funkcionality.

### 5.1 Použité zariadenia a technológie

Táto kapitola popisuje všetky technológie a zariadenia, ktoré boli využité pri vývoji a pri následnom nasadení systému.

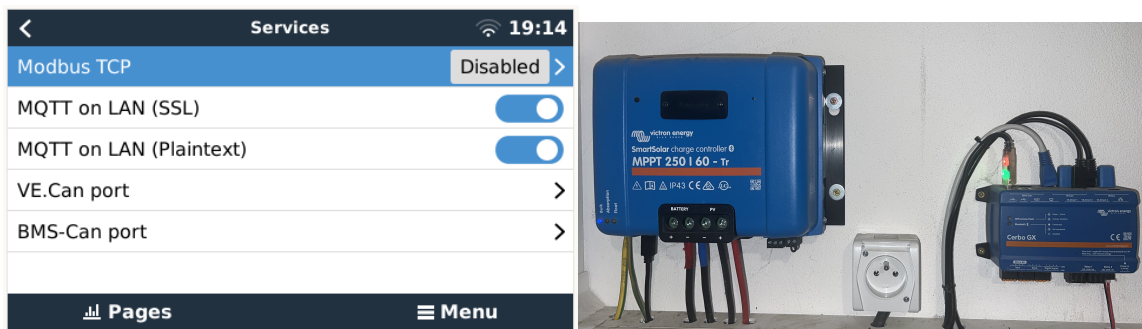
Pred implementáciou systému, bol k ohrevu využívaný iba plynový kotol a neskôr bol dokúpený elektrický bojler, ktorý bol pripojený na prebytkovú zásuvku (spomenuté v kapitole 3.2). Vďaka tomu bolo možné bojler nahrievať z prebytkov, ale nebolo možné konštantne sledovať jeho stav, keďže zásuvka bola zapnutá iba v prípade dostatku prebytkov na zapnutie bojleru. Taktiež nebolo možné pokrývať nedostatok generácie z batérií alebo distribučnej siete. Aktuálne zapojenie systému je možné vidieť na obrázku 5.1.



Obr. 5.1: Schéma zapojenia fotovoltaického systému a bojleru.

## Zariadenia

Ku komunikácii s fotovoltaickým systémom je využité zariadenie Cerbo GX s operačným systémom Venus OS s verziou v2.87. K tomu, aby mohlo dôjsť k úspešnému naviazaniu komunikácie medzi Home Assistantom a týmto zariadením, je nutné v systéme povoliť komunikáciu pomocou MQTT (kapitola 2.3) a nastaviť statickú IP adresu. Keďže Cerbo GX funguje ako MQTT broker, je neskôr toto zariadenie (pomocou jeho IP adresy) využité, ako MQTT server, na ktorý sa pripája Home Assistant na získanie potrebných informácií a publikáciu konfigurácie entít.



Obr. 5.2: Rozhranie Venus OS a zariadenia fotovoltaického systému.

Druhým komunikačným zariadením je ESP32-WROOM-32U, tento konkrétny typ mikrokontroléru som vybral kvôli jeho možnosti pripojiť externú Wi-Fi anténu. Tá zabezpečuje stabilnejšie pripojenie k sieti, keďže sa zariadenie nachádza v technickej miestnosti so slabým pokrytím signálu. Toto zariadenie funguje, ako komunikačný spoj medzi bojlerom a Home Assistantom. A je na ňom nainštalovaný systém ESPHome<sup>1</sup> pod ktorým beží komponent *ESPHome component for Dražice OKHE smart water heater*<sup>2</sup>, ktorý zabezpečuje vzdialené ovládanie bojleru.

Na ohrev vody je použitý bojler OKHE 160-SMART od značky Dražice, ktorý disponuje objemom o veľkosti 149 litrov a výhrevným telesom o príkone 1500 W.

Cerbo GX (obrázok 5.2, zariadenia úplne v pravo), mikrokontrolér ESP32 aj bojler sú pripojené do obvodu (fáza) dotovaného fotovoltaickým systémom. Pomocou tohto zapojenia je možné bojler nahrievať z prebytkov (poprípade batérií) a zamedziť výpadku komunikácie pri výpadku distribučnej siete.

<sup>1</sup>System na ovládanie mikrokontrolérov pomocou konfiguračných súborov a na ich vzdialené ovládanie (<https://esphome.io/>).

<sup>2</sup><https://github.com/LubosD/esphome-smartboiler>



Obr. 5.3: Bojler s mikrokontrolérom ESP32.

## Home Assistant

Základný komponent potrebný pri vývoji integrácie a k následnému nasadeniu tohto systému do bežného života, je bežiaci Home Assistant (kapitola 2.2). Tento systém musí byť neustále spustený, pretože na ňom bežia samotné integrácie. Systém je možné nainštalovať a spustiť ako osobitný operačný systém (HA OS) alebo ako docker aplikáciu, ktorá beží pod existujúcim operačným systémom. V mojom prípade som sa rozhodol využiť druhú možnosť, z toho dôvodu, že mám doma server s Linuxovým operačným systémom, na ktorom bolo možné jednoducho Home Assistant nasadiť. Tento server je tiež pripojený do obvodu (fáza) dotovaného fotovoltickým systémom, aby sa zamedzilo výpadku pri nedostupnosti distribučnej siete a integrácia sa mohla správne zastaviť a popri prípade aj vypnúť ohrev vody v bojleri.

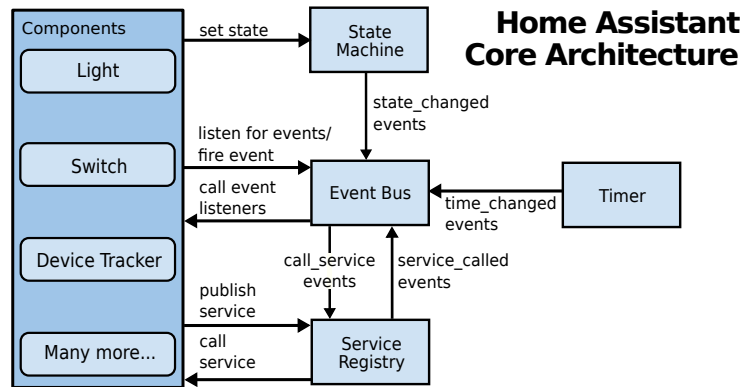
Pri vývoji som využil vývojové prostredie<sup>3</sup>, poskytnuté priamo vývojármi Home Assistanta, ktoré umožnilo efektívnejší a jednoduchší vývoj a testovanie integrácie (ďalej sa testovaniu venuje kapitola 5.3). Pomocou týchto dvoch na sebe nezávislých inštancií bežiacieho systému, som mohol súbežne vyvíjať integráciu a monitorovať dlhodobý stav reálneho systému, bez toho, aby som musel prerušiť jeho prevádzku.

Vďaka Home Assistantovi je možné využiť už existujúce integrácie, bez nutnosti vlastnej implementácie alebo vytvárania ďalších Docker aplikácií, ktoré by boli inak k tomuto účelu potrebné. Medzi integrácie, ktoré využívam patrí ESPHome<sup>4</sup> a MQTT<sup>5</sup>. Následne pomocou Home Assistant jadra, ktoré poskytuje prístup k týmto integráciám, dokážem zistiť ich entity, stavy entít, popri prípade naslúchať udalostiam (napríklad pri výpadku distribučnej siete), ale taktiež môžem ich stav zmeniť.

<sup>3</sup>[https://developers.home-assistant.io/docs/development\\_environment](https://developers.home-assistant.io/docs/development_environment)

<sup>4</sup><https://www.home-assistant.io/integrations/esphome/>

<sup>5</sup><https://www.home-assistant.io/integrations/mqtt/>



Obr. 5.4: Architektúra jadra Home Assistant (prevzaté z [5]).

Obrázok popisuje 4 hlavné časti Home Assistant jadra:

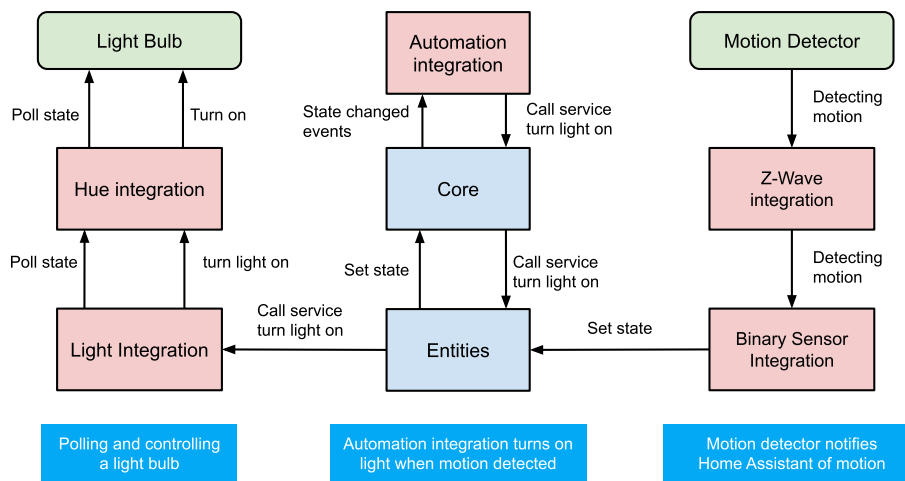
- **Event Bus:** zbernica, ktorá umožňuje spúšťanie a počúvanie udalostí.
- **State Machine:** sleduje stav vecí a pri zmene vyvolá udalosť `state_changed`.
- **Service Registry:** počúva na zbernici príchod `call_service` udalosti a umožňuje ostatným kódom registrovať služby.
- **Timer:** posielala udalosť `time_changed` na zbernicu `event bus` každú sekundu.

## 5.2 Implementácia logiky

Implementácia je vytvorená pomocou integrácie, ktorá rozširuje funkcie základného jadra Home Assistanta. Tento prístup umožňuje integrácii naslúchať udalostiam, spúšťať akcie, poskytovať špecifické služby a udržiavať stavy konkrétnych entít. Zloženie tejto integrácie pozostáva z komponentu, ktorý definuje samotnú logiku implementácie a platforiem, ktoré umožňujú interakciu s ostatnými integráciami. Komponent je implementovaný pomocou programovacieho jazyka Python.

Logika je rozvrhnutá do logických celkov, ktoré sú rozdelené medzi súbory. Každý z týchto súborov sa stará o spracovanie špecifických častí logiky komponentu. Medzi najhlavnejšie patrí `config_flow.py` na spracovanie inštalácie, `__init__.py`, ktorý obsahuje inicializáciu komponentu a inicializáciu MQTT komunikácie (diagram 5.7). A súbor `manager.py` (diagramy 5.8 až 5.12), ktorý definuje správanie a riadenie celého systému.

Okrem týchto súborov, obsahuje komponent aj štandardizované entity, ktoré sú rozdelené do súborov podľa príslušných vlastností. K nim patrí `number.py`, `select.py`, `sensor.py`, `switch.py` a `time.py`. Každá z týchto entít poskytuje základnú funkcionálnosť, ktorá je následne rozšírená o ďalšie vlastnosti a funkcionality potrebné pre správne fungovanie a spracovanie logiky komponentu. Tento prístup umožňuje jednoduchú rozšíriteľnosť kódu a zvyšuje jeho prehľadnosť a modularitu.



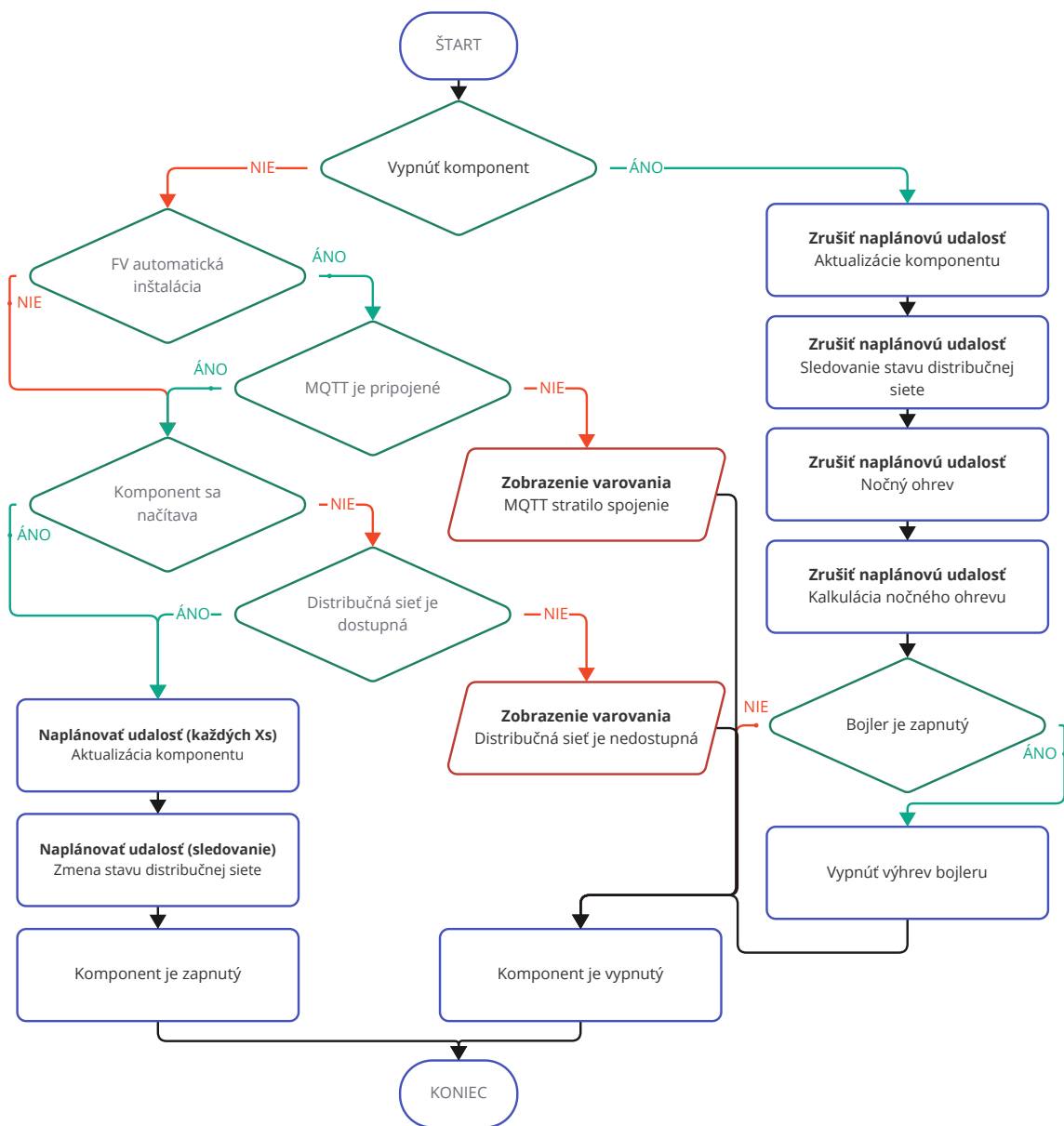
Obr. 5.5: Architektúra integrácií Home Assistanta (prevzaté z [6]).

## Rozšírenia vlastností entít

Každá z entít poskytuje základnú funkčnosť, ako je napríklad posuvník na nastavenie teploty, prepínač na zapnutie nočného ohrevu, vstup na vyplnenie požadovaného času nočného ohrevu a podobne. Ale niektoré z týchto entít sú rozšírené o ďalšie funkčnality, potrebné k správne spracovaniu stavov.

Entity rozšírené o ďalšiu funkčnality:

- **Switch** slúži na zapnutie popripade vypnutie nočného ohrevu a disponuje navyše funkčnality, ktorá sa postará, pri vypnutí nočného ohrevu užívateľom, o zrušenie naplánovaných logických častí nočného ohrevu, vypnutie bojleru a uvedenie premenných spojených s nočným ohrevom do pôvodného stavu.
- **Senzor**, ktorý obsahuje dáta predpovede generácie FV systému na dnešný a zajtrajší deň, automaticky získava nové dáta pomocou žiadostí na VRM API každú hodinu. A má naplánovanú pravidelnú aktualizáciu dát minútu po polnoci, aby sa hodnoty dnešnej a zajtrajšej predpovede správne aktualizovali a zobrazovali tak stále aktuálne hodnoty.
- **Number** poskytuje posuvník na nastavenie teploty ohrevu počas dňa a teploty nočného ohrevu. Aby bolo možné správne ovládať teplotu v bojleri, entita pri zmene teploty automaticky túto hodnotu nastaví aj na bojleri.
- **Select** slúži na zmenu režimu komponentu (vypnutý, automatický, manuálny). Keďže pomocou tejto entity je možné ovládať celý komponent, obsahuje aj niekoľko ďalších krokov, ktoré sa postarajú o správne vypnutie, popripade zapnutie komponentu. V prípade uvedenia komponentu do vypnutého režimu sa zrušia všetky naplánované udalosti ako pravidelné aktualizácie komponentu, sledovanie stavu distribučnej siete a naplánovaný nočný ohrev. A vypne sa ohrev vody v bojleri. V opačnom prípade sa skontroluje MQTT pripojenie, stav distribučnej siete, naplánuje sa pravidelná aktualizácia komponentu a zapne sa sledovanie stavu distribučnej siete. Túto logiku popisuje diagram 5.6, ktorý sa nachádza na ďalšej strane.



Obr. 5.6: Diagram zapnutia/vypnutia komponentu.

## Inicializácia komponentu

Po inštalácii integrácie alebo znovu spustení systému (napríklad reštartovanie Home Asistanta) sa `__init__.py` postará o správne vytvorenie premenných, inicializáciu entít a naviazanie MQTT komunikácie.

Ak bola inštalácia fotovoltického systému (ďalej iba FV systém) prevedená pomocou automatického režimu, navyše ešte publikuje auto-discovery konfiguráciu senzorov na MQTT broker. A prihlási sa k odberu zmien vo všetkých témach, ktoré FV systém publikuje. Prihlásenie k odberu daných tém a k publikácii konfigurácie, zabezpečuje MQTT integrácia (spomenutá v 5.1).

Publikovaná téma k odberu vyzerá nasledovne:

```
R/<venus_topic>/system/+Settings/CGwacs/BatteryLife/State/+Power
```

Obsahuje niekoľko zástupných znakov (kapitola 2.3), ktoré nahrádzajú časť hierarchie témy, ktorá môže byť pre každého užívateľa iná. A `<venus_topic>`, ktorý je zároveň aj MAC adresa zariadenia s Venus OS si užívateľ zadá pri inštalácii. Pre každý senzor je publikovaná jedna jedinečná téma.

Publikácia konfigurácie funguje podobne, ako pri odbere, každý senzor má vlastnú tému a táto téma je v tvare `homeassistant/sensor/<domain>/<sensor_id>/config`, kde doména je jedinečná doména integrácie a id senzoru je jedinečný názov pre každý senzor. Každá téma navyše obsahuje konfiguračné dáta, medzi ktoré patrí meno senzoru, trieda<sup>6</sup>, téma, kde sú publikované dáta od FV systému, jednotka merania, vzor hodnoty, jedinečné id a zariadenie, pod ktoré daný senzor patrí. Tým je zabezpečené automatické pridanie a aktualizácia všetkých entít z FV systému. Následne je pri prvom príchode dát na MQTT broker, nastavená fáza, ktorá je podporovaná FV systémom (toto platí iba v prípade automatickej FV inštalácie, v opačnom prípade si fázu nastavuje užívateľ pri inštalácii).

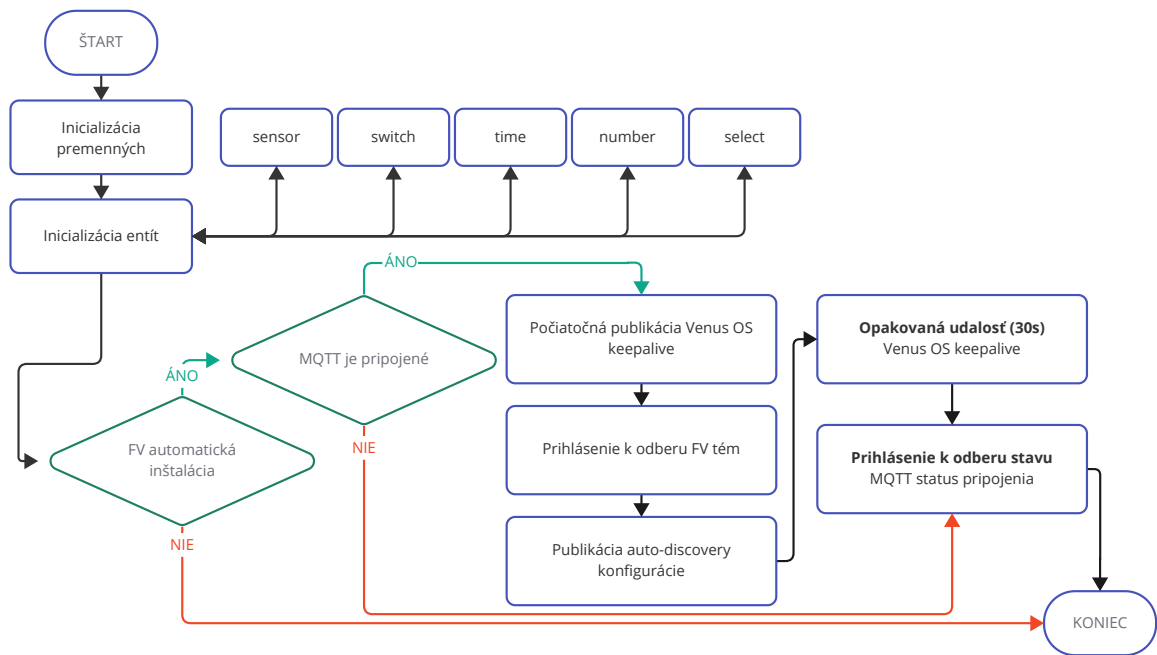
Keďže Venus OS po určitej dobe neaktivity prestane publikovať nové stavy systému, naplánuje sa opakovaná udalosť, ktorá každých 30 sekúnd publikuje správu na MQTT broker (na diagrame 5.7 označené ako *Opakovaná udalosť*). Táto správa je publikovaná do témy `R/<venus_topic>/keepalive` s dátami, ktoré obsahujú všetky témy, ktoré majú byť publikované FV systémom. Jedna z tém na ukážku, je v tvare `system/+Ac/Grid/+Power`, kde časti témy, sú nahradené zástupným charakterom (kapitola 2.3), rovnako, ako pri publikovaní tém k odberu.

Aby komponent dokázal zaznamenať a správne spracovať výpadok komunikácie MQTT, prihlási sa k odberu stavu MQTT komunikácie, na diagrame 5.7, znázornené ako posledný proces. Viac v kapitole 5.2 o spracovaní neočakávaných udalostí.

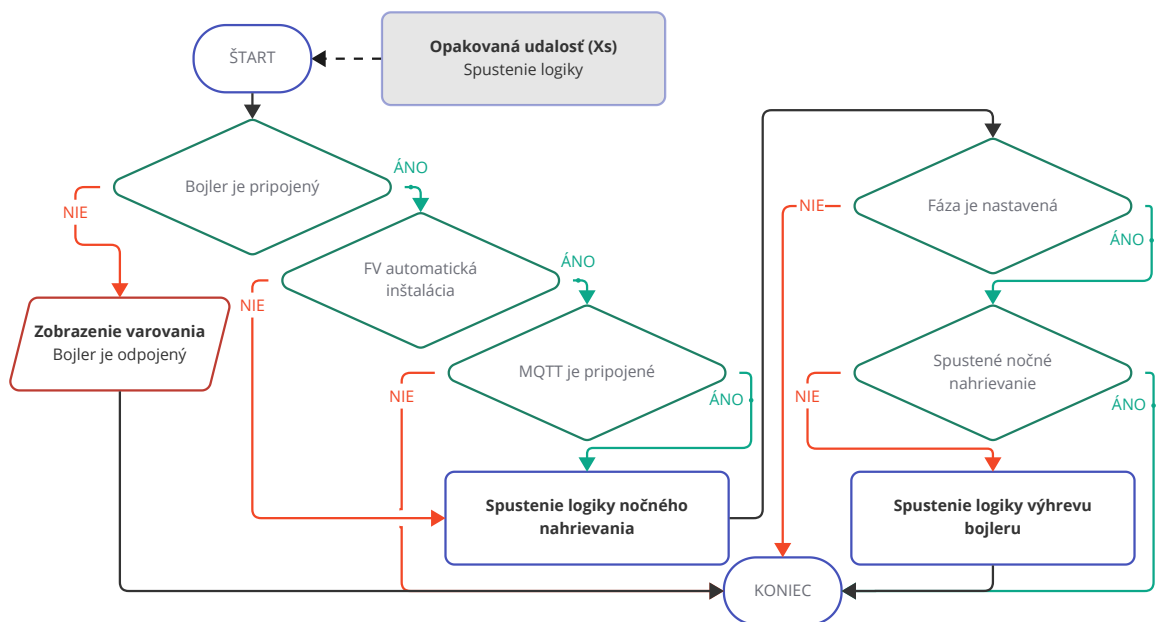
---

<sup>6</sup><https://developers.home-assistant.io/docs/core/entity/sensor/#available-state-classes>





Obr. 5.7: Diagram inicializácie komponentu.



Obr. 5.8: Diagram kontroly pred spustením logiky.

## Hlavná logika komponentu

Po úspešnej inicializácii a spustení komponentu je naplánovaná udalosť, ktorá v pravidelných intervaloch spustí logiku v súbore `manager.py`. Čas, ako často sa komponent aktualizuje je nastavený užívateľom pri inštalácii. Pred samotným spustením logiky na ohrev vody v bojleri a nočného ohrevu vody, je nutné vykonať niekoľko kontrol, aby bolo možné, tieto procesy vykonať úspešne. Tieto kontroly popisuje predchádzajúci diagram na obrázku 5.8.

Ako prvé sa vykoná kontrola konektivity s bojlerom, typ FV inštalácie (manuálny alebo automatický) a poprípadе aj stav MQTT pripojenia. Po úspešnom absolvovaní týchto testov, program spustí logiku nočného nahrievania (podrobnejšie opísané v 5.2 časti o nočnom ohreve vody). Následne pokračuje preverením fázy a stavu nočného ohrevu. V prípade, že údaj o fázy chýba, je logika pozastavená do príchodu dát z MQTT, podľa ktorých je možné fázu nastaviť. Ak údaj o fázy nechýba a nočný ohrev nie je spustený, program pokračuje na spustenie samotnej logiky ohrevu vody v bojleri.

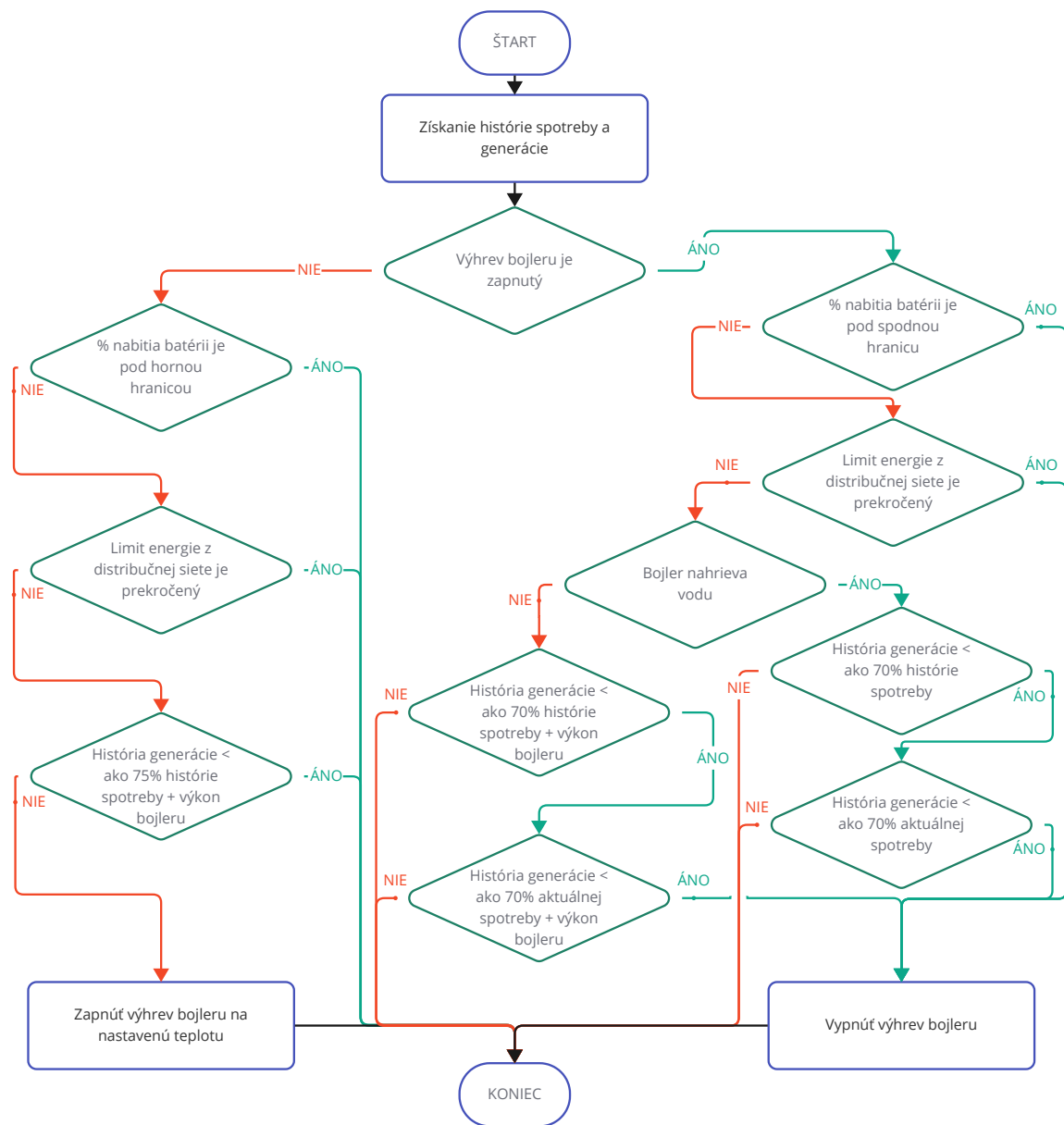
Pred zahájením rozhodovania o spustení alebo zastavení ohrevu vody bojleru, je nutné získať hodnoty potrebné k výpočtom a porovnávaniam. Tieto hodnoty zahŕňajú nastavenia definované užívateľom, ako je výkon bojleru, prahové hodnoty batérií a distribučnej siete a nastavená teplota výhrevu. Okrem toho je nutné zohľadniť aj aktuálny stav nabitia batérií, spotrebu v domácnosti a stav bojleru. Následne je možné vďaka Home Assistantovi, ktorý zaznamenáva a poskytuje históriu dát zo senzorov, stanoviť percentilnú hodnotu (viz podkapitola 2.6) spotreby energie v domácnosti a generácie FV systému za určitý čas.

Konkrétne, 50. percentil (alebo medián) spotreby v domácnosti, je stanovený na základe údajov z posledných 30 sekúnd. To znamená, že polovica z nameraných hodnôt je menšia alebo rovná ako táto hodnota. Vzhľadom k tomu, že tieto hodnoty sú z malého štatistického súboru, môžeme túto hodnotu považovať za približnú spotrebu v domácnosti, ktorá nie je výrazne ovplyvnená krátkodobým alebo prudkým zaťažením siete. Keďže generácia FV panelov nie je stabilná a je veľmi ovplyvnená počasím a inými faktormi, je táto hodnota stanovená ako 70. percentil určený z dát za posledných 10 minút. Tieto hodnoty sú nastavené na základe pozorovania správania reálneho systému a jeho reakcií na reálne podmienky.

Následne sa systém riadi logikou popísanou v diagrame na obrázku 5.9, kde sa rozhoduje na základe dostupných informácií o spustení prípadne vypnutí nahrievania vody v bojleri.

Logika odzrkadľuje očakávania a návrh implementácie. Kde berie v ohľad batérie, ktoré udržiava nabitie v nastavenom rozmedzí, kontroluje neprekročenie stanoveného limitu odberu energie z distribučnej siete a zapína alebo vypína ohrev podľa dostupnej generácie FV panelov.

Pri kontrole spustenia výhrevu by FV panely mali pokrývať aspoň 75% energie spotrebovanej v domácnosti vrátane výkonu bojleru. Táto hodnota je o 5% vyššia, ako pri kontrole vypnutia, z toho dôvodu, aby nedochádzalo k zbytočnému zapínaniu a vypínaniu výhrevu, ak by výkon generácie v krátkej dobe poklesol. Po spustení ohrevu, je teda nutné, aby FV panely pokrývali aspoň 70% spotreby energie a o zvyšok sa postarajú batérie. Tu je treba brať ohľad aj na stav bojleru, keďže môže dosiahnuť nastavenú teplotu a sám vypne výhrev. Pri takejto situácii ho systém považuje stále za zapnutý, ale bojler nenahrieva vodu, čiže ani nečerpá energiu. Je, teda nutné k spotrebe v domácnosti pripočítať výkon bojleru, aby logika správne fungovala. Ak generácia FV panelov nepostačuje na pokrytie spotreby v domácnosti, kontroluje sa ešte s aktuálnou spotrebou, keďže mohlo dôjsť k dlhšiemu zaťaženiu siete, ktoré percentil zachytil, ale aktuálna spotreba môže byť už nižšia.

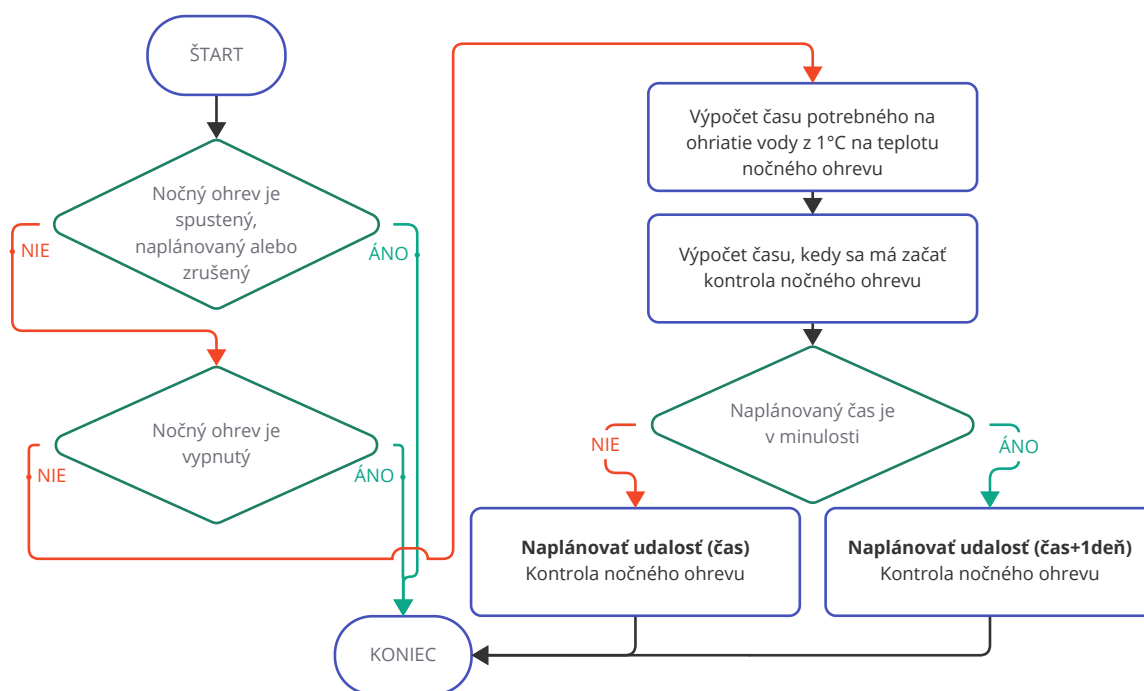


Obr. 5.9: Diagram spustenia/zastavenia ohrevu vody v bojleri (počas dňa).

## Nočný ohrev vody

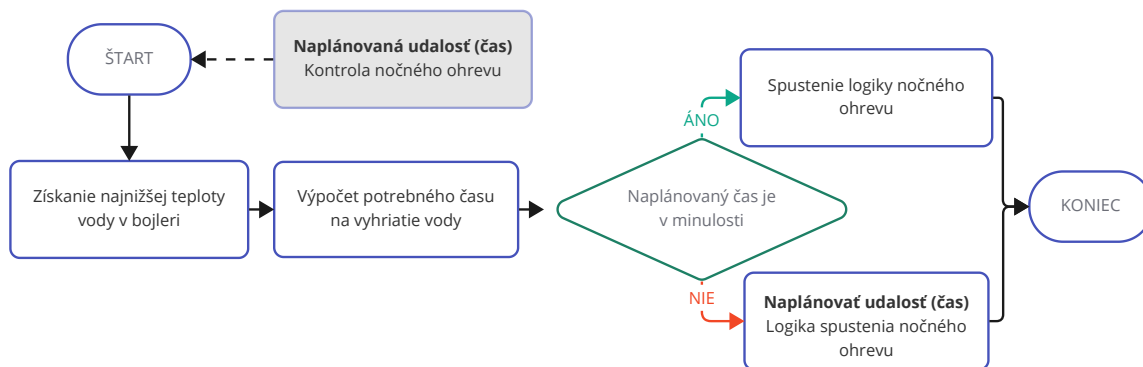
Aby sa nočný ohrev vody nespustilo príliš skoro a tým nebol ovplyvnený výpočet potrebného času, je logika rozdelená do štyroch logických segmentov. Pričom každý z týchto segmentov sa stará o konkrétnu časť postupu logického celku.

V nasledujúcich diagramoch je spomenutá kontrola plánovania času do minulosti. Pri tejto kontrole sa porovnáva aktuálny čas s plánovaným časom a je vyhodnotená, ako pravdivá v prípade, že naplánovaný čas je menší (odohráva sa skôr), ako je aktuálny čas alebo je rozdiel týchto časov menší ako 5 minút. Päť minútový rozdiel funguje ako časová rezerva, v prípade, že sa časy líšia v sekundách, poprípade milisekundách, kontrola sa vyhodnotí, ako nepravdivá, ale, skôr ako sa dokončí plánovanie, môže byť tento čas prekročený.



Obr. 5.10: Diagram kontroly pred nočným ohrevom.

Prvá časť znázornená na diagrame 5.10, sa zameriava na predbežnú kontrolu a plánovanie nadchádzajúceho kroku s dostatočným predstihom. Keďže táto kontrola je spúšťaná pri každom spustení hlavnej logiky (diagram 5.8), na začiatku je vykonaná kontrola či je naplánovaný niektorý z ďalších segmentov logiky nočného ohrevu poprípade, či je nočný ohrev spustený alebo zrušený. Následne sa skontroluje či má užívateľ nočný ohrev povolený. A v kladom prípade je možné pomocou vzorca na výpočet potrebného času na vyhriatie vody v bojleri (viz rovnica 2.5), vypočítať, koľko bude trvať vyhriatie vody z 1°C na nastavenú teplotu nočného ohrevu. Posledným krokom tejto časti je naplánovanie spustenia ďalšieho segmentu logiky. Ten je naplánovaný na vypočítaný čas, poprípade na ďalší deň, ak bol tento čas už prekročený. Tým je zabezpečené ďalšie pokračovanie logiky dostatočne včas na ohrev vody.



Obr. 5.11: Diagram plánovania spustenia nočného ohrevu.

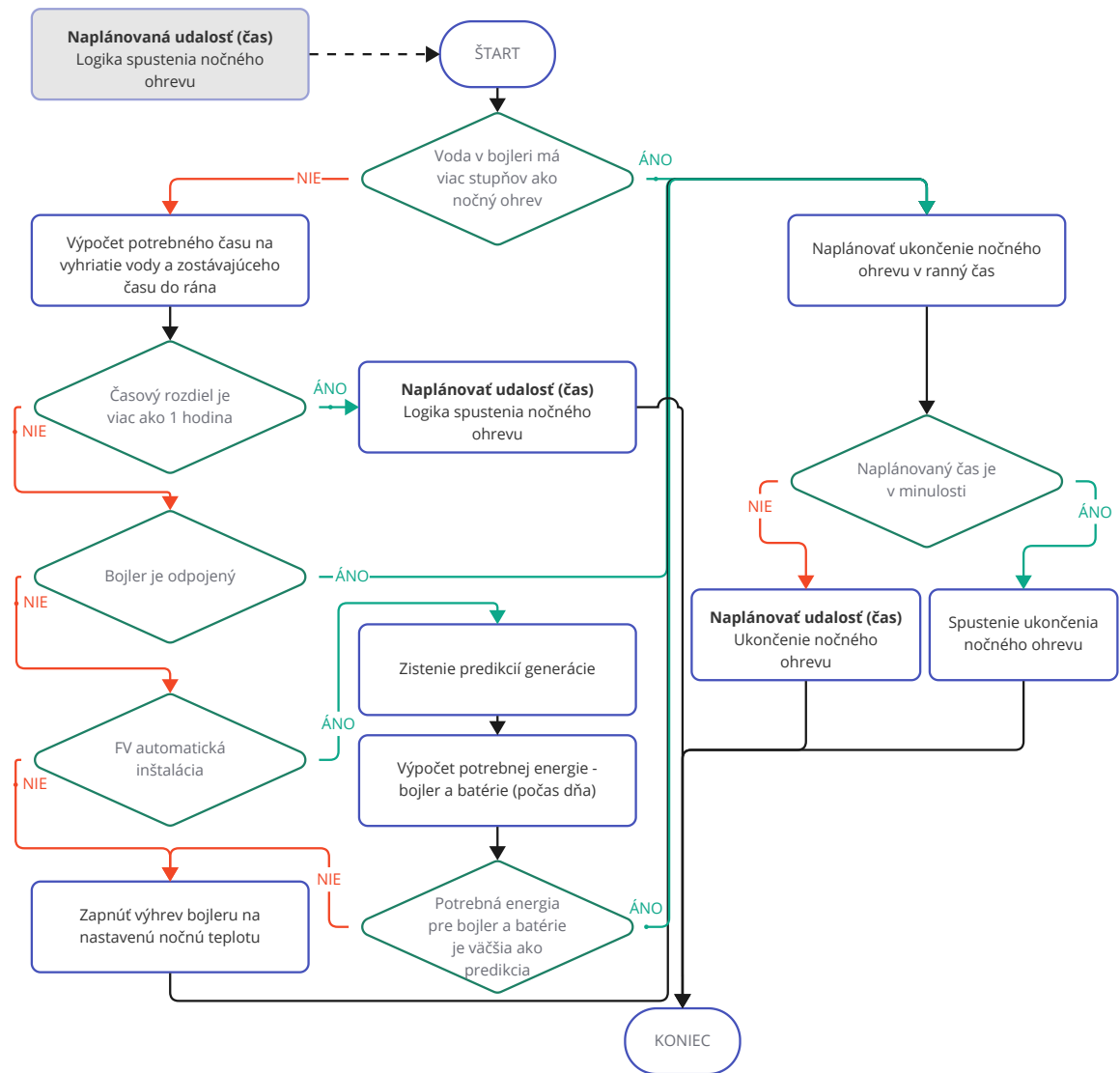
Druhý segment, znázornený na diagrame 5.11, je spustený na základe plánovania z prvej časti. Zameriava sa na presnejší výpočet potrebného času ohrevu (rovnica 2.5), pomocou minimálnej teploty vody v bojleri. Táto teplota je získaná z aktuálnej teploty a z minimálnej teploty, ktorú voda dosiahla deň predtým v 5 hodinovom rozmedzí do ukončenia nočného ohrevu (ak by bol koniec ohrevu naplánovaný na 5:00, berie sa rozmedzie od polnoci do piatej ráno). Následne je naplánovaná tretia časť logiky, poprípade je rovno spustená, ak bol naplánovaný čas prekročený.

Diagram na obrázku 5.12, popisuje logiku tretieho segmentu nočného ohrevu. Táto časť logiky je zodpovedná za samotné spustenie nočného ohrevu vody v bojleri. Je spustená pomocou predchádzajúceho plánovania a keďže sa jedná o finálnu časť nočného ohrevu (nasleduje už iba ukončenie), je možné považovať teplotu vody v bojleri za konečnú a je, teda možné túto teplotu využiť na výpočty.

Spustenie ohrievania je podmienené aktuálnou teplotou vody v bojleri. Ak teplota neklesla pod nastavenú teplotu nočného ohrevu, ohrievanie je zrušené a naplánuje sa jeho ukončenie. V opačnom prípade, sú od tejto teploty odčítané  $3^{\circ}\text{C}$ , ktoré označujú akúsi rezervu pri výpočtoch. Pomocou tejto hodnoty je vypočítaný potrebný čas ohrevu na základe rovnice 2.5 a zostávajúci čas do rána (nastaveného ranného času). Ak je rozdiel týchto časov väčší ako jedna hodina, pre-plánuje sa spustenie ohrevu o daný rozdiel a logika sa znovu spustí neskôr.

Pred spustením ohrevu je nutné skontrolovať v akom režime pracuje komponent. Ak sa jedná o manuálny režim, nočný ohrev sa zapne a je naplánované jeho ukončenie. Pri automatickom režime je nutné vykonať ďalšie kroky.

Vypočíta sa minimálna teplota vody, ktorá má byť dosiahnutá počas dňa. Táto teplota je určená odčítaním variačnej teploty od nastavenej dennej teploty výhrevu. Následne je vypočítaný teplotný rozdiel medzi minimálnou teplotou a nastavenou teplotou nočného ohrevu. Hodnota značí o koľko stupňov musí byť voda cez deň vyhriata. Použitím rovnice 2.4 a prevedením hodnoty na kWh použitím konverzie  $1\text{kWh} = 3,6\text{MJ}$ , je možné odhadnúť koľko energie bude potrebné na dosiahnutie tejto teploty.



Obr. 5.12: Diagram logiky spustenia nočného ohrevu.

Aby bolo možné určiť či generácia z FV panelov počas dňa bude postačujúca, je treba určiť ešte potrebnú energiu na nabitie batérií počas dňa. To je možné docieľiť, pomocou vzorca:

$$energy = \frac{battery_{cap} \cdot (battery_{dsoc} - battery_{soc})}{10000} \quad (5.1)$$

Kde *energy* je potrebná energia na dosiahnutie percentuálneho nabitia batérií v kWh, *battery<sub>cap</sub>* je maximálna uskladnená energia v batériách, *battery<sub>dsoc</sub>* je požadované percento nabitia a *battery<sub>soc</sub>* je aktuálne percento nabitia.

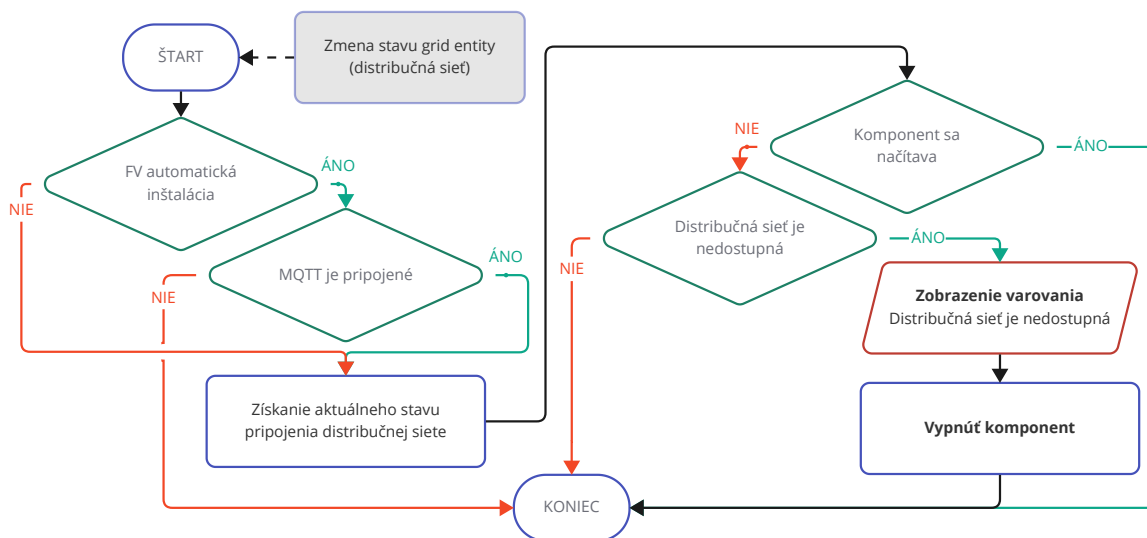
V prípade, že celková potrebná energia na nahriatie vody a nabitie batérií počas dňa, je menšia ako predpovedaná generácia FV panelov, spustí sa nočný ohrev vody a naplánuje sa jeho ukončenie.

Ukončenie ohrevu je naplánované pri každej úspešnej alebo neúspešnej časti logiky nočného ohrevu. Plánuje sa na nastavený ranný čas a zároveň funguje ako blokátor, aby sa zabránilo ďalšiemu naplánovaniu nočného ohrevu. V momente aktivácie vypína nahrievanie vody v bojleri a resetuje stavové premenné do pôvodného stavu, čím signalizuje ukončenie alebo vymazanie zrušenia nočného ohrevu. A umožňuje tak plánovanie ďalšieho cyklu a zahájenie dennej logiky komponentu.

### Spracovanie neočakávaných udalostí

Ak dôjde k zmene stavu v MQTT pripojení alebo v entite, ktorá zobrazuje stav distribučnej siete. Je táto zmena automaticky zaznamenaná a je vyvolaná príslušná logika, ktorá túto zmenu spracuje.

Ako už bolo spomenuté v návrhu logiky, výpadok komunikácie s bojlerom nie je možné ideálne vyriešiť, preto tento stav nie je sledovaný. O spracovanie tohto stavu sa postará pravidelná aktualizácia komponentu, popísaná v časti Hlavná logika komponentu.



Obr. 5.13: Diagram spracovania zmeny stavu distribučnej siete.

Diagram na obrázku 5.13, popisuje správanie komponentu, pri zmene stavu entity, ktorá zobrazuje stav distribučnej siete.

Pri spracovaní tohto stavu je dôležité zistiť či bola entita pridaná pomocou automatickej inštalácie cez MQTT konfiguráciu. V takomto prípade sa môže jednať iba o výpadok komunikácie s MQTT, o ktorý sa postará logika popísaná nižšie. Ak sa nejedná o výpadok komunikácie je ešte možné, že sa komponent nestihol inicializovať a treba vylúčiť aj túto možnosť.

Ak došlo naozaj k výpadku distribučnej siete, zobrazí sa chybová správa, vypne sa bojler a komponent sa uvedie do vypnutého stavu (bližšie popísané v 5.2, časť rozšírenia vlastností entít – `Select`). Tento stav je konečný a komponent môže znovu zapnúť až užívateľ, z dôvodu, že môže dôjsť napríklad k ďalšiemu výpadku siete a batérie by sa v medzi čase zbytočne vybíjali na ohrev vody.

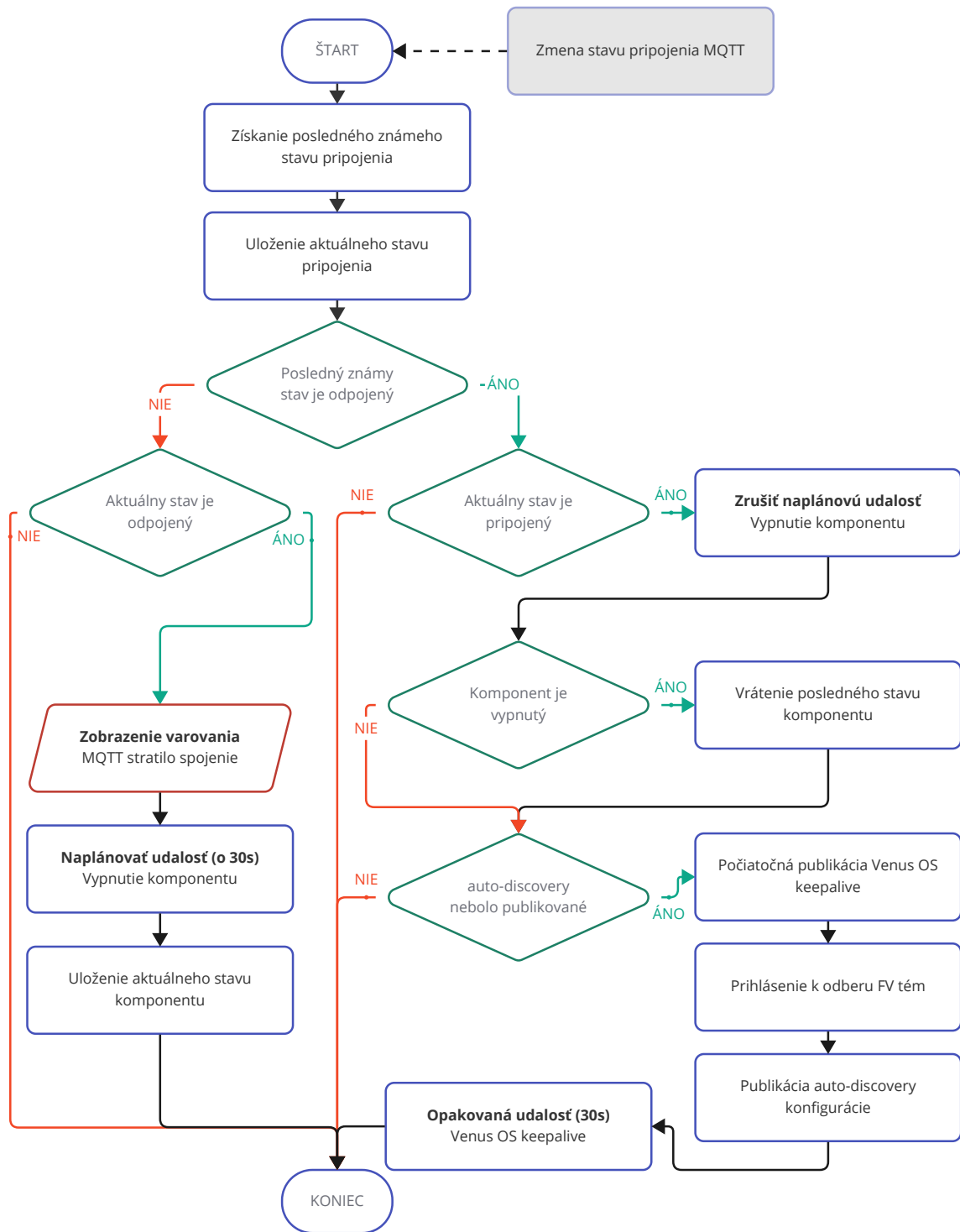
Diagram na obrázku 5.14 popisuje správanie komponentu, pri zmene stavu v MQTT pripojení.

Keďže stav sa môže zmeniť na pripojený alebo odpojený, je nutné identifikovať o akú zmenu ide. Toto je možné vďaka premennej, ktorá udržiava posledný známy stav pripojenia a novému stavu, ktorý je ohlásený pri vyvolaní logiky.

Ak došlo k výpadku komunikácie, komponent naplánuje svoje vypnutie o 30 sekúnd, uloží svoj aktuálny stav a zobrazí varovanie o nedostupnom MQTT pripojení. Ak do uplynutia tohto času nedôjde k obnoveniu komunikácie, komponent sa uvedie do vypnutého stavu a zobrazí chybovú správu. Ak dôjde k obnoveniu komunikácie v priebehu 30 sekúnd, naplánované vypnutie je ukončené a komponent pokračuje v prevádzke bez prerušenia.

V prípade, že dôjde k obnoveniu komunikácie po uplynutí časovej rezervy, teda komponent sa už vypol. Opätovne sa zapne do stavu, v ktorom bol pred vypnutím a pokračuje v prevádzke. Ak ešte nedošlo k publikácií konfigurácie entít FV systému, publikuje konfiguráciu, prihlási sa k odberu daných tém a naplánuje opakovanú udalosť, ktorá publikuje keepalive správu.





Obr. 5.14: Diagram spracovania zmeny stavu MQTT pripojenia.

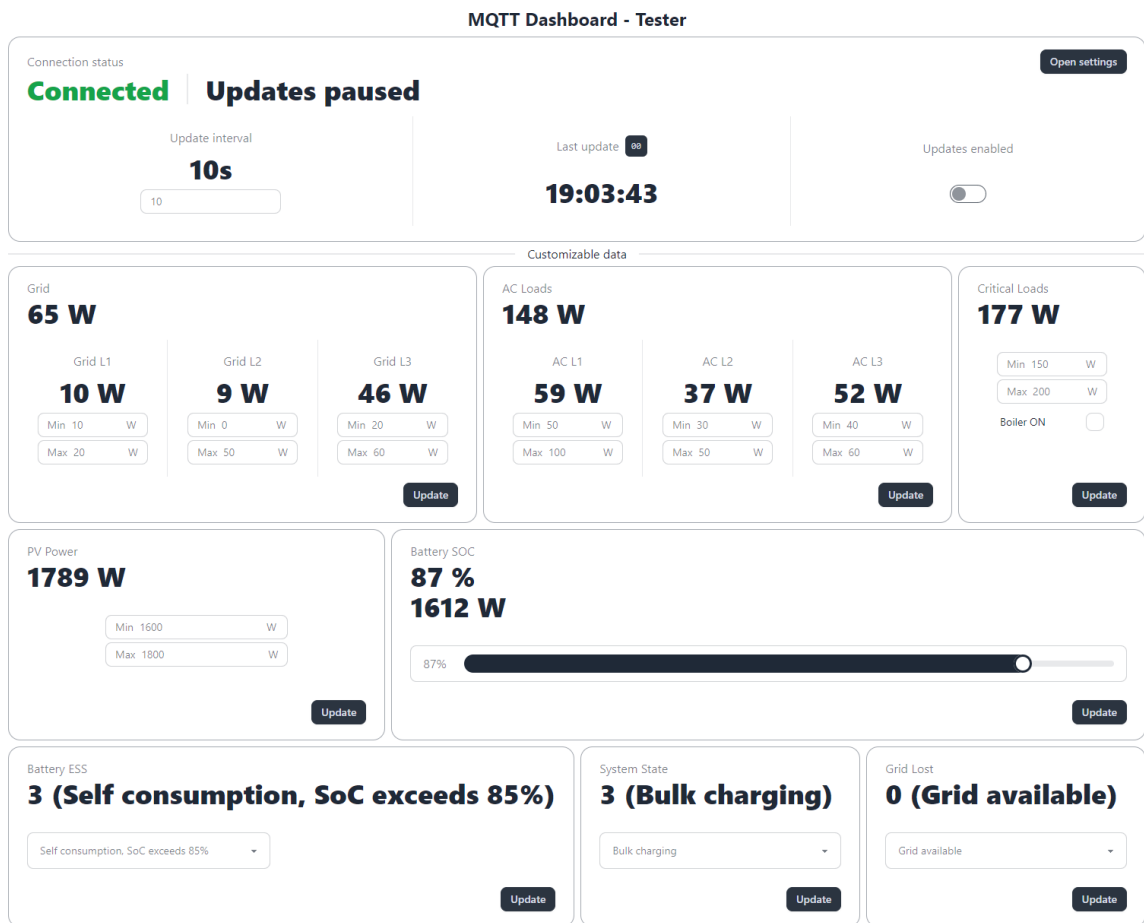
### 5.3 Testovanie integrácie

K zabezpečeniu správneho fungovania integrácie v rámci systému Home Assistant, bolo nutné integráciu počas vývoja aktívne testovať.

Testovanie bolo realizované pomocou vývojového prostredia Home Assistanta, ktoré umožňovalo testovať kód priamo v editore s využitím ladiacich nástrojov. Vďaka tomu bolo možné pozastaviť logiku v reálnom čase a identifikovať, poprípade vyriešiť problémy. Tento prístup poskytol aj možnosť sledovať interné stavy Home Assistanta a postup vykonávania logiky, ktorý bol prínosný pri sledovaní návratových hodnôt funkcií a objasnenie nie vždy dobre zdokumentovaných Home Assistant funkcií.

Ďalšou dôležitou súčasťou testovania, bolo overenie funkcionality MQTT a správne spracovanie hodnôt, ktoré poskytuje. K tomuto účelu som si napísal jednoduchú web stránku (obrázok 5.15), ktorá umožňovala pripojenie na MQTT broker a manuálne nastavenie hodnôt, ktoré sú publikované z Venus OS. Následne web stránka tieto hodnoty spracovala, upravila ich do rovnakého formátu (témy), ako Venus OS a v pravidelných intervaloch ich publikovala na broker. Vďaka tomu bolo možné nasimulovať výpadok distribučnej siete, MQTT komunikácie alebo odsledovať správanie integrácie pri rôznych zmenách ako vyťaženie siete, pokles generácie a podobne, bez nutnosti dlhodobého sledovania a čakania na zmenu v reálnom systéme.

Keďže takéto simulovanie dát neodpovedá presne realite, bolo nutné integráciu tiež dlhodobo sledovať spustenú v reálnom systéme. To som docielil pomocou ladiacich výpisov, ktoré ukladali chod systému a dáta využité pri výpočtoch a rozhodovaniach do logu. A pomocných entít, ktoré zobrazovali podrobnejší stav systému integrácie. Vďaka tomu bolo možné spätne analyzovať správanie integrácie pri reálnom nasadení a odhaliť tak prípadné problémy a vyladiť logiku.



Obr. 5.15: Web stránka využitá na testovanie.

## Kapitola 6

# Testovanie

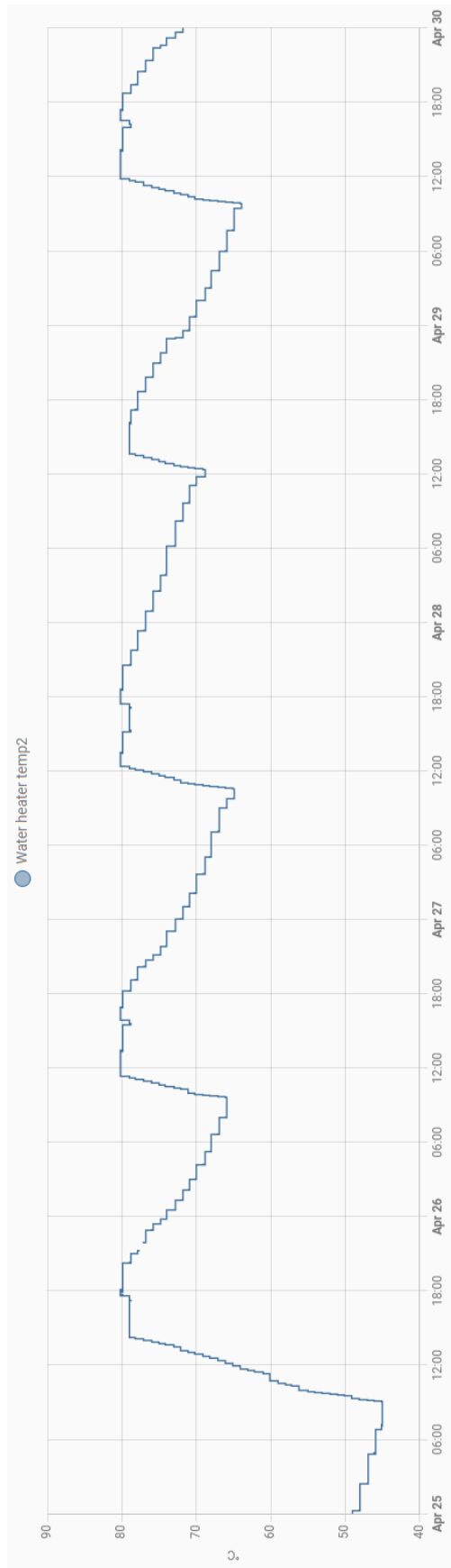
Táto kapitola opisuje reálne nasadenie systému v domácnosti a dosiahnuté výsledky, ktorým sa venuje časť 6.1. Následný komfort užívateľov a ich zhodnotenie fungovania systému (časť 6.2) a možné vylepšenia a zmeny v systéme (6.3), ktoré plánujem implementovať v budúcnosti.

### 6.1 Nasadenie systému

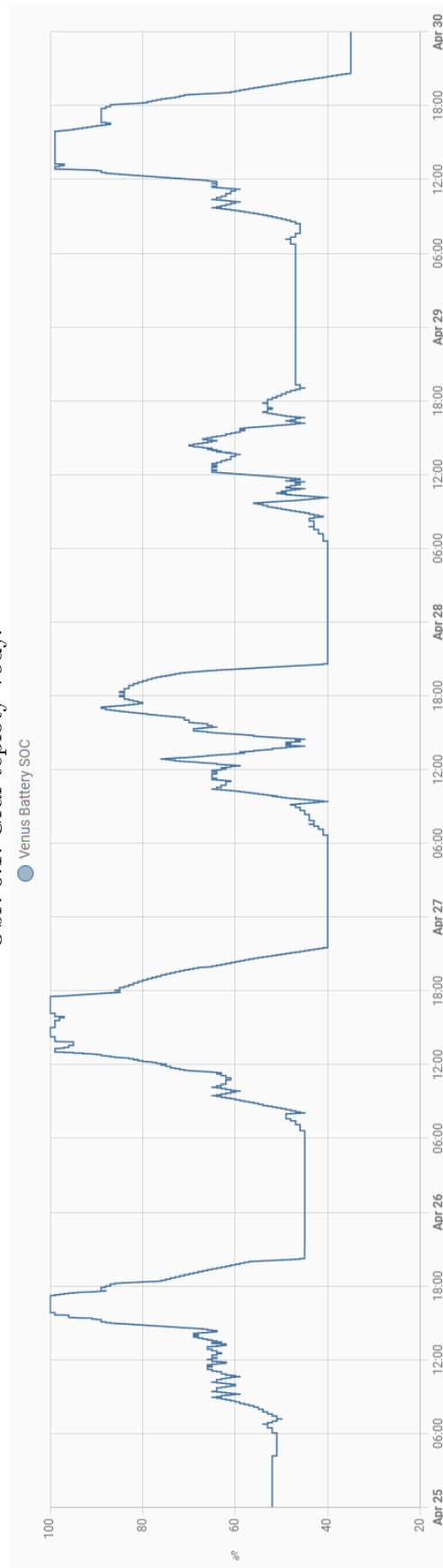
Systém je spustený v našej domácnosti, ktorá sa skladá z piatich členov (použitie zariadenia boli spomenuté v kapitole 5.1). A počas obdobia nasadenia systému neboli zaznamenané žiadne väčšie problémy, ktoré by sa týkali nedostatku teplej vody, vybitia batérií alebo nutného zásahu do systému. Taktiež som nainštaloval Home Assistant aplikáciu otcovi do telefónu, aby mohol systém sledovať, ovládať a mohol mi následne zdeliť jeho pocity týkajúce sa integrácie, viac v kapitole 6.2.

V grafe na obrázku 6.1 je možné vidieť, že za obdobie piatich dní, teplota vody neklesla pod nastavenú teplotu (v tomto prípade  $40^{\circ}\text{C}$ ) a počas dňa sa dokázala vyhriať na maximálnu možnú teplotu, ktorú bojler dovoľuje. A zároveň graf 6.2 zobrazuje stav nabitia batérií, ktoré boli udržiavané v požadovanom rozsahu 60% až 65%. Po nahriatí vody na maximálnu teplotu sa batérie stále nabíjali a neskôr večer boli vybité, kvôli povoleným prítokom elektrickej energie do distribučnej siete.

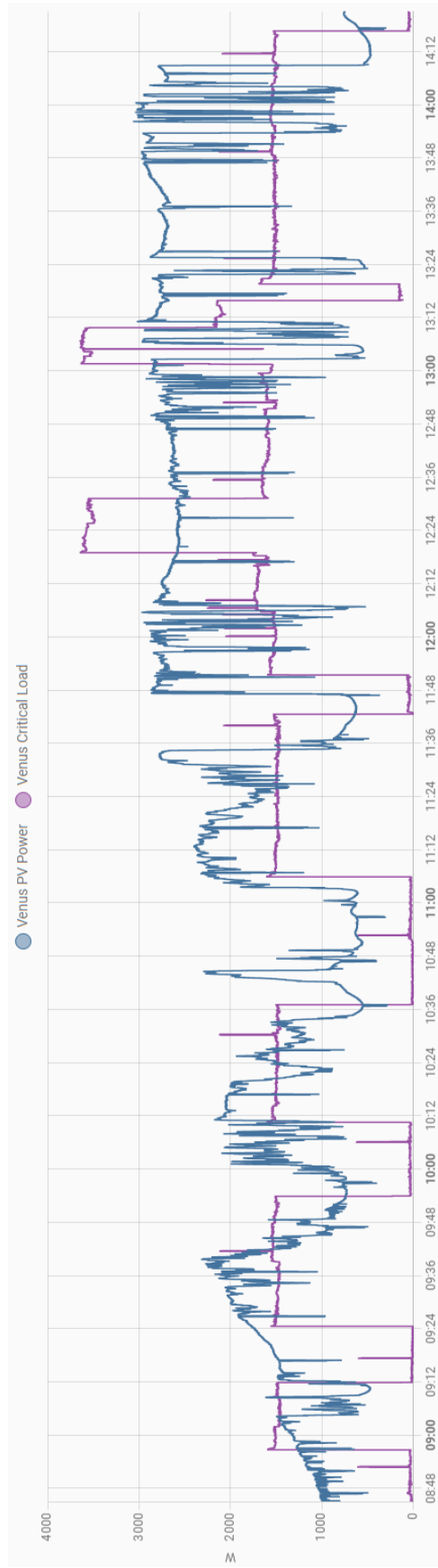
Graf 6.3 zobrazuje časť dňa fungovania systému, počas ktorého bola generácia fotovoltických panelov značne kolísavá. Modrá čiara označuje generáciu fotovoltických panelov a fialová aktuálnu spotrebu v domácnosti, ktorú fotovoltický systém dotuje. Pri spustenom ohreve spotreba stúpne na približne 1500 W a približne o 600 W (poprípade na 2100 W) nárast spôsobuje ponorné čerpadlo. V grafe je vidno, že sa ohrev zapol pred 9 hodinou ráno po tom, čo batérie dosiahli požadované horné percento nabitia a ohrev je postupne vypínaný a zapínaný, aby sa batérie udržiavali nabité. To, ako skoro sa má ohrev zapnúť poprípade, ako často sa bude vypínať/zapínať, je možné ovplyvniť nastavením väčšieho rozsahu nabitia batérií. Neskôr pri dosiahnutí väčšej generácie energie sa batérie stíhajú nabíjať aj pri zapnutom ohreve a ohrev je vypnutý iba v prípade nadmerného odberu energie, ktorú FV systém nedokáže pokryť. To je možné vidieť približne v 13:10, keď spotreba dosiahla približne 3600 W a systém nedokázal spotrebu pokryť a ohrev bol vypnutý. Spotreba vtedy klesla na približne 2100 W a následne na 150 W, kedy sa ohrev znovu spustil. Ďalej je možné vidieť, že systém zbytočne nevypína ohrev pri krátkom stúpnutí spotreby (napríklad 10:30) alebo pri dlhšom stúpnutí, ak to FV systém stále dokáže pokryť (12:24).



Obr. 6.1: Graf teploty vody.



Obr. 6.2: Graf stavu nabíjání baterií.



Obr. 6.3: Graf FV generácie a spotreby v domácnosti.

## Úspora energie a plynu

O úspore ako takej, je ťažké niečo napísať, keďže systém je určený primárne na letné obdobie, keď je počas dňa dostatok slnečného žiarenia a dostatok prebytkovej energie.

Čo sa týka úspory energie, tá sa môže prejavíť až pri dlhodobej prevádzke systému, keďže v našom FV systéme sú povolené prietoky energie do distribučnej siete, nie je jednoduché určiť úsporu. Tá sa následne prejaví až vo výslednom ročnom vyúčtovaní energie. Ale z aktuálnych grafov sa dá usúdiť, že počas dňa je dostatok energie na pokrytie ohrevu vody a nabitie batérií, čiže viac energie zostáva v domácnosti, čo bol jeden zo zámerov.

Pri plyne je situácia podobná, keďže plyn sa používa na ohrev úžitkovej vody, vykurovanie a varenie. Tak sa úspora prejaví až cez letné obdobie. Ale keďže za posledných pár dní je plynový kotol vypnutý a plyn sa teda využíva len na varenie, bolo možné odsledovať približnú spotrebu, ktorá bola 0,1 až 0,3 kubického metra plynu denne. A podľa informácií môjho otca, býva spotreba plynu pri zapnutom ohreve vody a kúrení okolo 1 kubického metra denne. To znamená, že pri využití elektrického bojlera ako primárneho zdroja teplej úžitkovej vody, dochádza k dennej úspore plynu o objeme približne 0,7 kubického metra.

## 6.2 Komfort užívateľov

Dôležitým faktorom pri zhodnotení systému, je nie len vyhodnotenie jeho fungovania, ale aj to, ako sa užívateľom s ním pracuje a či nejak ovplyvnil ich kvalitu života.

Ako už bolo spomenuté v kapitole 6.1, pri aktívnom využívaní systému nedošlo k vybitiu batérií alebo k nedostatku teplej vody. Čiže nedošlo k negatívnemu ovplyvneniu komfortu, čo potvrdili aj členovia v domácnosti. Medzi ďalšie pozitívne hodnotenia patrí vyššia teplota vody, ktorá mohla byť vďaka elektrickému bojleru dosiahnutá, keďže voda sa ohrieva v prípade dostatku prebytkov na maximálnu teplotu, ktorú bojler povoľuje. A ovládanie, ktoré otec zhodnotil ako jednoduché, čo mu vyhovovalo a bol spokojný, že vidí celý systém v jednej aplikácii a dokáže si tak skontrolovať teplotu vody, generáciu FV panelov a podobne. A zároveň dokáže systém z tejto aplikácie ovládať.

K negatívnej časti bolo pripisované iba anglické rozhranie systému, ktoré robilo rodičom zo začiatku problém.

## 6.3 Možné zmeny a rozšírenia

Žiadny systém nie je dokonalý a vždy ponúka priestor na zlepšenie. A to sa týka aj tohto systému, kde sa naskytujú možnosti, ako rozšíriť funkcie alebo zlepšiť užívateľskú prívietivosť. Táto kapitola popisuje možné zmeny a rozšírenia, ktoré by bolo dobré implementovať, poprípade plánujem v budúcnosti upraviť.

Prvým krokom, ktorý by zlepšil užívateľskú prívietivosť by bolo možné vydanie systému ako „oficiálnu“ HACS<sup>1</sup> integráciu a pridanie prekladu. To by zabezpečilo jednoduchšie pridanie integrácie do Home Assistant systému a zjednodušilo ovládanie, pre užívateľov, ktorí neovládajú anglický jazyk.

K plánovaným zmenám patrí zmena spracovania predpovede generácie pomocou verejne dostupných služieb a lepšia integrácia ostatných bojlerov, keďže momentálna implementácia sa zameriava na fungovanie hlavne s bojlermi od značky Dražice a systémom od Victron Energy. Vďaka tomu bude možné zabezpečiť univerzálnejšie použitie integrácie.

---

<sup>1</sup>Home Assistant Community Store: <https://hacs.xyz/>

Taktiež by bolo dobré pri manuálnom režime udržiavať minimálnu teplotu vody počas celého dňa, nie len zabezpečiť teplú vodu na ráno, pomocou nočného ohrevu.

A k zložitejším rozšíreniam by som si predstavoval lepšie spracovanie prebytkov pri automatickom režime, formou premenlivého výkonu bojleru, kde by sa výkon špirály upravoval podľa dostupných prebytkov, nebolo by, teda potrebné pokrývať nedostatok generácie z batérií. A v prípade využitia dvoch zdrojov teplej vody (elektrický bojler a plynový kotol), ovládanie oboch zariadení, kde by sa plynový kotol zapínal iba v prípade nedostatku prebytkov.



# Kapitola 7

## Záver

Cieľom tejto práce bolo vytvorenie riešenia, ktoré umožní automatický a komfortný ohrev teplej úžitkovej vody z prebytkov fotovoltaickej elektrárne vo forme integrácie do systému Home Assistant. Riešenie sa mi podarilo úspešne implementovať a nasadiť ho do domácnosti, kde sa ukázalo ako funkčné.

Integráciu aktívne využívame v domácnosti, kde zabezpečuje ohrev vody v bojleri, ktorý je momentálne náš jediný zdroj teplej vody. Počas používania tohto systému nenastala situácia, že by sme zostali bez teplej vody z čoho súdim, že implementácia funguje správne. Zároveň dochádza k úspore v spotrebe plynu, keďže sa nevyužíva plynový kotol a k menším prietokom prebytkovej energie z fotovoltaickej elektrárne naspäť do distribučnej siete. Vďaka tomu môžeme byť menej závislí na komerčných zdrojoch a prispieť tak aj ku ekológii.

Pri implementácii tejto práce som sa podrobnejšie naučil, ako fungujú inteligentné zariadenia, komunikácia medzi nimi, ako ich integrovať do centrálného systému a všeobecne, ako si zautomatizovať určité časti domácnosti. Myslím si, že v dnešnej dobe je tento smer zamerania veľmi dôležitý, a preto by som na práci rád pokračoval aj v budúcnosti. Kde by som rád túto prácu rozšíril o ďalšie funkcie, poprípade urobil zmeny v aktuálnej verzii s novými vedomosťami, ktoré som počas implementácie nadobudol. Medzi najdôležitejšie funkcie, ktoré by som rád urobil, patrí zabezpečenie univerzálnejšieho použitia, pomocou verejne dostupných služieb ako predpoveď generácie a zabezpečenie fungovania s rôznymi značkami bojlerov.

Integráciu som poskytol online<sup>1</sup> na GitHube, keďže verím, že táto práca bude v budúcnosti prospešná, nie len pri ovládaní a nahrievaní bojleru za pomoci fotovoltaickej elektrárne, ale aj ako inšpirácia pri vytváraní ďalších riešení, ktoré povedú k lepšiemu využitiu fotovoltaických elektrární.

---

<sup>1</sup><https://github.com/FNewel/PV-Water-Heating-Manager>

# Literatúra

- [1] *API authentication* [online]. [cit. 2024-04-14]. Dostupné z: <https://www.postman.com/api-platform/api-authentication/>.
- [2] GOODWIN, M. *What is an API* [online]. Apríl 2024 [cit. 2024-04-14]. Dostupné z: <https://www.ibm.com/topics/api>.
- [3] ASSISTANT, H. *Home Assistant* [online]. [cit. 2024-04-14]. Dostupné z: <https://www.home-assistant.io/>.
- [4] ASSISTANT, H. *Home Assistant concepts and terminology* [online]. [cit. 2024-04-14]. Dostupné z: <https://www.home-assistant.io/getting-started/concepts-terminology/>.
- [5] ASSISTANT, H. *Home Assistant Core Architecture* [online]. 2021 [cit. 2024-04-20]. Dostupné z: <https://developers.home-assistant.io/docs/architecture/core>.
- [6] ASSISTANT, H. *Home Assistant Integration Architecture* [online]. 2024 [cit. 2024-04-20]. Dostupné z: [https://developers.home-assistant.io/docs/architecture\\_components/](https://developers.home-assistant.io/docs/architecture_components/).
- [7] KNEŽO, D., ANDREJIOVÁ, M. a IŽARÍKOVÁ, G. *ZÁKLADNÉ ŠTATISTICKÉ METÓDY* [online]. November 2011 [cit. 2024-04-14]. Dostupné z: [https://www.sjf.tuke.sk/kamai/literatura/stat\\_metody.pdf](https://www.sjf.tuke.sk/kamai/literatura/stat_metody.pdf).
- [8] KOL., P. J. a. *Aplikovaná statistika*. 1. Vydanie. VŠCHT Praha, 2005. 10–11 s. ISBN 978-80-7080-569-5.
- [9] KÚDELČÍK, J. a HOCKICKO, P. *Základy fyziky*. 1. Vydanie. Žilinská univerzita v Žiline / EDIS - vydavateľstvo ŽU, 2011. 71–73, 176–177, 186–187 s. ISBN 978-80-554-0341-0.
- [10] MITURA, J. *Z hloupého bojleru chytrý pomocí chytré zásuvky*. Brno, CZ, 2021. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/24021/>.
- [11] *MQTT* [online]. [cit. 2024-04-14]. Dostupné z: <https://mqtt.org/>.
- [12] ROSE, K., ELDRIDGE, S. a CHAPIN, L. The internet of things: An overview. *The internet society (ISOC)*. Reston, VA. 2015, zv. 80, č. 15, s. 13–18.
- [13] SAPI. *Rozvoj fotovoltiky v roku 2023 prekonal očakávania* [online]. 2024 [cit. 2024-04-20]. Dostupné z: <https://www.sapi.sk/clanok/sapi-rozvoj-fotovoltiky-v-roku-2023-prekonal-ocakavania>.

- [14] SILVERIO FERNÁNDEZ, M., RENUKAPPA, S. a SURESH, S. What is a smart device?-a conceptualisation within the paradigm of the internet of things. *Visualization in Engineering*. Springer. 2018, zv. 6, č. 1, s. 8.
- [15] TEAM, H. *Introducing the MQTT Protocol – MQTT Essentials: Part 1* [online]. Január 2015. Revizované 2024-02-14 [cit. 2024-04-14]. Dostupné z: <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>.
- [16] TEAM, H. *MQTT Client, MQTT Broker, and MQTT Server Connection Establishment Explained – MQTT Essentials: Part 3* [online]. Júl 2019. Revizované 2023-06-06 [cit. 2024-04-14]. Dostupné z: <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>.
- [17] TEAM, H. *MQTT Topics, Wildcards, & Best Practices – MQTT Essentials: Part 5* [online]. August 2019. Revizované 2024-02-20 [cit. 2024-04-14]. Dostupné z: <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>.
- [18] TSCHOFENIG, H., ARKKO, J., THALER, D. a MCPHERSON, D. R. *Architectural Considerations in Smart Object Networking* [RFC 7452]. RFC Editor, marec 2015 [cit. 2024-04-14]. DOI: 10.17487/RFC7452. Dostupné z: <https://datatracker.ietf.org/doc/html/rfc7452>.
- [19] XIA, F., YANG, L. T., WANG, L., VINEL, A. et al. Internet of things. *International journal of communication systems*. 2012, zv. 25, č. 9, s. 1101.
- [20] *Zákon zachovania energie* [online]. [cit. 2024-04-14]. Dostupné z: <http://uef.fei.stuba.sk/moodleopen/mod/glossary/showentry.php?courseid=117&eid=1512&displayformat=dictionary>.