

Návrh a implementace chytré domácnosti pomocí systému openHAB

Libor Výleta

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Libor Výleta**
Osobní číslo: **A22087**
Studijní program: **B0613A140020 Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Návrh a implementace chytré domácnosti pomocí systému openHAB**
Téma práce anglicky: **Design and Implementation of a Smart Home Using the OpenHAB System**

Zásady pro vypracování

1. Seznamte se se základy provozu platformy openHAB a potřebným vybavením pro její provozování
2. Provedte návrh konceptu chytré domácnosti pomocí systému openHAB pro vybranou konkrétní aplikaci a prostředí.
3. Provedte instalaci a nastavení systému OpenHAB na minipočítači Raspberry Pi nebo na jiném vybraném zařízení. Proveďte v tomto systému přidání vybraných zařízení.
4. Navrhněte a implementujte uživatelsky přívětivé rozhraní pro ovládání chytré domácnosti.
5. Zhodnotte možnosti Vámi navrženého systému.

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

Seznam doporučené literatury:

1. UPTON, Eben a HALFACREE, Gareth. *Raspberry Pi: uživatelská příručka*. 2. aktualizované vydání. Brno: Computer Press, 2016. ISBN 978-80-251-4819-8.
2. KYAS, Othmar. *How To Smart Home*. Wyk: Key Concept Press. 2020. ISBN 978-3-944980-19-5.
3. OpenHAB. Online. Dostupné z: <https://www.openhab.org/docs/>. [cit. 2024-10-17].
4. LINN, Robert. *Make Project Home Automation*. Online. 2018. Dostupné z: <https://www.scribd.com/document/381261984/openhab-homeautomation>. [cit. 2024-10-23].
5. Kreuzer, Kai. *Building a Cloudless Smart Home with openHAB*. FrOSCon 2018, Free and Open Source Software Conference (FrOSCon) e.V., 25–26 Aug. 2018. <https://doi.org/10.5446/45788>. [cit. 2024-10-28].

Vedoucí bakalářské práce: **doc. Ing. Martin Adámek, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **27. října 2024**
Termín odevzdání bakalářské práce: **2. června 2025**



L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 17. listopadu 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato bakalářská práce se zaměřuje na návrh a realizaci systému chytré domácnosti s využitím open-source platformy openHAB. Hlavním cílem bylo sjednotit ovládání různých chytrých zařízení, která byla původně řízena samostatnými aplikacemi, do jednoho centralizovaného prostředí. V rámci práce byla provedena instalace systému, výběr vhodného hardwaru, integrace zařízení z domácnosti a návrh uživatelského rozhraní. Součástí řešení je také automatizace pomocí pravidel a vizualizace dat včetně grafického výstupu. Výsledky potvrzují vhodnost zvoleného řešení a identifikují jeho limity, například v oblasti podpory některých zařízení bez oficiálních doplňků.

Klíčová slova: openHAB, chytrá domácnost, domácí automatizace, vizualizace, uživatelské rozhraní

ABSTRACT

This bachelor thesis focuses on the design and implementation of a smart home system using the open-source platform openHAB. The main goal was to unify the control of different smart devices, which were originally managed by separate applications, into one centralized system. The work includes the installation of the system, selection of suitable hardware, integration of home devices, and the design of a user interface. The solution also involves automation through rules and data visualization, including graphical output. The results confirm that the chosen solution is suitable and also highlight some limitations, such as the lack of official support for some devices.

Keywords: openHAB, smart home, home automation, visualization, user interface

Na tomto místě bych rád vyjádřil upřímné poděkování vedoucímu bakalářské práce doc. Ing. Martinu Adámkovi, Ph.D., za odborné vedení, cenné rady a trpělivost během celé doby zpracování práce.

Z celého srdce děkuji své rodině za neochvějnou podporu, pochopení a trpělivost, kterou mi věnovali nejen při psaní této práce, ale po celou dobu studia. Jejich podpora pro mě znamenala bezpečné zázemí, sílu překonávat překážky a motivaci nikdy se nevzdávat.

OBSAH

ÚVOD.....	13
I TEORETICKÁ ČÁST.....	14
1 CHYTRÁ DOMÁCNOST A JEJÍ PRINCIPY	15
1.1 VÝHODY CHYTRÉ DOMÁCNOSTI	15
1.2 HISTORIE A VÝVOJ CHYTRÝCH DOMÁCNOSTÍ	15
1.3 BEZPEČNOST CHYTRÉ DOMÁCNOSTI.....	16
1.3.1 Kybernetická bezpečnost chytré domácnosti	16
1.3.2 Ochrana chytré domácnosti před fyzickými hrozbami	17
1.4 TECHNOLOGIE POUŽÍVANÉ V CHYTRÉ DOMÁCNOSTI	18
1.4.1 Komunikační protokoly a standardy	18
1.4.2 Chytrá zařízení a senzory	21
2 SYSTÉM OPENHAB.....	23
2.1 HISTORIE A VÝVOJ.....	24
2.2 ARCHITEKTURA SYSTÉMU OPENHAB.....	24
2.2.1 Bindings	24
2.2.2 Things.....	25
2.2.3 Channels.....	25
2.2.4 Items	26
2.2.5 Rules.....	26
3 INSTALACE A KONFIGURACE SYSTÉMU OPENHAB.....	28
3.1 VÝBĚR PLATFORMY A HARDWARE	28
3.1.1 Minimální požadavky na instalaci openhabianu	29
3.1.2 Raspberry Raspberry Pi imager	29
3.1.3 Etcher	29
II PRAKTICKÁ ČÁST.....	30
4 INSTALACE SYSTÉMU OPENHAB – OPENHABIANU	31
4.1.1 Instalace pomocí Raspberry Pi imager.....	31
4.1.2 Instalace pomocí Etcher	31
4.2 ZÁKLADNÍ NASTAVENÍ SYSTÉMU	32
4.3 POPIS JEDNOTLIVÝCH ČÁSTÍ NA HLAVNÍ STRÁNCE.....	34
4.3.1 Hlavní obsah – číslo 1	35
4.3.2 Navigační menu (Správa) – číslo 2	36
4.3.3 Uživatelský účet – číslo 3	38
4.3.4 Rychlé odkazy – číslo 4	38
4.3.5 Logo a stav – číslo 5.....	39
5 CÍL PRÁCE	40
5.1 POPIS VYBRANÉHO PROSTŘEDÍ A APLIKACE.....	40

5.1.1	Popis prostředí	40
5.1.2	Požadavky na aplikaci	41
6	NÁVRH SYSTÉMU CHYTRÉ DOMÁCNOSTI	42
6.1	POPIS VYBRANÝCH ZAŘÍZENÍ PRO CHYTROU DOMÁCNOST	42
6.1.1	HP Color LaserJet MFP M274n	42
6.1.2	Sinclair SIH – 13BITC	42
6.1.3	Sinclair SIH – 09BIK	42
6.1.4	Hybridní měnič GoodWe GW10K-ET	42
6.1.5	Sony KD-49X8307C	43
6.1.6	Sony KD-55AG8	43
6.1.7	Myčka nádobí Bosch SMi6ZCS00E	43
6.1.8	Sonoff TH16	43
6.2	POUŽITÉ DOPLŇKY A JEJICH SPECIFIKACE	43
6.2.1	HP Printer Binding	43
6.2.2	Gree Binding	44
6.2.3	ESPHome binding for the Native API	44
6.2.4	Exec Binding	44
6.2.5	Home Connect Binding	44
6.2.6	Sony Binding	45
6.2.7	HTTP Binding	45
7	PŘIDÁNÍ A KONFIGURACE ZAŘÍZENÍ	46
7.1	PŘIDÁNÍ ZAŘÍZENÍ POMOCÍ WEBOVÉ STRÁNKY	46
7.2	PŘIDÁNÍ ZAŘÍZENÍ POMOCÍ VS CODE	46
7.3	INTEGRACE TISKÁRNÍ HP	47
7.3.1	Vytvoření souboru .things	48
7.3.2	Vytvoření souboru .items	48
7.3.3	Vytvoření sitemapy	49
7.3.4	Vytvoření .map transformace	49
7.4	INTEGRACE TEPLOTNÍHO SENZORU S ESPHOME	50
7.4.1	Seznámení s potřebnými komponenty	50
7.4.2	Instalace a konfigurace ESPHome	50
7.4.3	Zapojení komponent do mikrokontroléru ESP8266	52
7.4.4	Implementace teplotní stanice	53
7.4.5	Vytvoření ochranné krabičky	56
7.4.6	Přidání teplotního senzoru	57
7.4.7	Vytvoření sitemapy	59
7.5	INTEGRACE KLIMATIZACÍ V DOMĚ	59
7.5.1	Vytvoření souboru .things	59
7.5.2	Vytvoření souboru .items	60
7.5.3	Vytvoření sitemapy	60
7.6	IMPLEMENTACE A KONFIGURACE MĚNIČE GOODWE	61
7.6.1	Připojení pomocí SSH k Raspberry PI	61
7.6.2	Instalace potřebných nástrojů	62

7.6.3	Vytvoření Python skriptu pro získání dat z měniče	62
7.6.4	Vytvoření bash skriptu pro spuštění Python skriptu	64
7.6.5	Vytvoření souboru .things	64
7.6.6	Vytvoření souboru .items	64
7.6.7	Vytvoření souboru .rules	65
7.6.8	Vytvoření widgetu pro informace	65
7.6.9	Vytvoření sitemapy	66
7.7	IMPLEMENTACE OVLÁDÁNÍ MYČKY	68
7.7.1	Home Connect API – účet, vývojářský účet, aplikace	68
7.7.2	Home Connect API – Integrace	70
7.7.3	Integrace myčky na nádobí	71
7.7.4	Vytváření kanálů(Channels)	72
7.7.5	Vytvoření sitemapy	73
7.7.6	Vytvoření pravidla	74
7.8	INTEGRACE ZÍSKÁNÍ CENY ELEKTŘINY	74
7.8.1	Soubor pro získání ceny	74
7.8.2	Vytvoření bash skriptu pro spuštění Python skriptu	75
7.8.3	Vytvoření souboru .things	76
7.8.4	Vytvoření souboru .items	76
7.8.5	Vytvoření souborů .rules	77
7.8.6	Vytvoření sitemapy	77
7.9	INTEGRACE OVLÁDÁNÍ BOJLERU	78
7.9.1	Připojení zařízení eWeLink do služby SmartThings pro ovládání bojleru	78
7.9.2	Vytvoření tokenu pro ovládání přes API	78
7.9.3	Vytvoření souboru .things	79
7.9.4	Vytvoření souboru .items	80
7.9.5	Vytvoření sitemapy	80
7.10	INTEGRACE OVLÁDÁNÍ SONY TV	80
7.10.1	Vytvoření sitemapy	81
7.11	INTEGRACE UKLÁDÁNÍ DAT DO PAMĚTI	82
8	NASTAVENÍ A PŘIPOJENÍ OPENHAB CLOUD	83
8.1	CO JE TO OPENHAB CLOUD?	83
8.2	INTEGRACE OPENHAB CLOUD	83
8.3	VYTVOŘENÍ NOVÉHO UŽIVATEL S ROLÍ – REGULAR USER	84
8.4	PŘIPOJENÍ APLIKACE OPENHAB NA MOBILNÍM ZAŘÍZENÍ	85
8.5	VYTVOŘENÍ NOTIFIKACÍ	86
8.5.1	Notifikace – Program dokončen mycí zařízení	86
8.5.2	Notifikace – Zařízení je offline	86
9	VÝSLEDKY	88
9.1	OVLÁDÁNÍ ZAŘÍZENÍ	88
9.2	VIZUALIZACE SYSTÉMU	88

ZÁVĚR.....	89
SEZNAM POUŽITÉ LITERATURY	91
SEZNAM OBRÁZKŮ	95
SEZNAM TABULEK.....	97
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	98
SEZNAM PŘÍLOH.....	100

ÚVOD

Téma chytré domácnosti se v posledních letech stalo výrazně aktuálním. Zejména v souvislosti s rostoucí dostupností chytrých zařízení a potřebou efektivního hospodaření s energiemi. Vzhledem k vlastnímu zájmu o automatizaci a praktickému nasazení v reálném domě byl zvolen systém openHAB, který umožňuje integraci různorodých zařízení napříč protokoly bez závislosti na konkrétním výrobci. Tato bakalářská práce se zabývá návrhem a implementací řešení pro sjednocení ovládání chytrých zařízení, která byla doposud řízena samostatnými aplikacemi určenými přímo pro jednotlivá zařízení.

Cílem práce je vytvořit systém správy chytré domácnosti, který by zajistil jednotné a intuitivní ovládání jednotlivých zařízení pomocí open-source platformy openHAB. V rámci realizace jsou popsány teoretické principy chytré domácnosti, proveden výběr vhodného hardwaru a softwaru, popsány procesy instalace, konfigurace a integrace konkrétních zařízení včetně tvorby pravidel, uživatelského rozhraní a notifikací.

Přínosem práce je vytvoření plně funkčního systému, který slouží jako ukázka moderní implementace chytré domácnosti pro běžného uživatele. Výsledný systém snižuje technickou náročnost obsluhy, zvyšuje přehlednost a podporuje efektivní energetické chování domácnosti.

Pro formulaci některých částí textu byl použit generativní nástroj ChatGPT, a to výhradně pro stylistickou úpravu odborného textu. Obsah byl po vygenerování pečlivě zkontrolován a za výsledné znění autor přebírá plnou odpovědnost.

I**TEORETICKÁ ČÁST**

1 CHYTRÁ DOMÁCNOST A JEJÍ PRINCIPY

Chytrá domácnost (smart home) je definována jako systém propojených zařízení umožňujících automatizaci a vzdálené ovládání různých prvků v domácnosti. Do tohoto systému mohou být zahrnuty oblasti, jako je řízení osvětlení, vytápění, stínění, zabezpečení nebo ovládání domácích spotřebičů. Rozsah a funkcionality chytré domácnosti se odvíjejí od konkrétních požadavků uživatele. Ovládání může být realizováno prostřednictvím nástěnných panelů, mobilních aplikací či hlasových asistentů. Automatizace umožňuje například zapínání světel při detekci pohybu, regulaci teploty v závislosti na denní době nebo automatické uzamykání dveří při opuštění objektu. [1]

1.1 Výhody chytré domácnosti

Mezi hlavní výhody chytré domácnosti patří zvýšení komfortu, úspora času a energie, posílení bezpečnosti a moderní estetické zpracování. Uvedené přínosy přispívají k vytvoření pohodlnějšího a efektivnějšího životního prostředí v domácnosti. [2]

1.2 Historie a vývoj chytrých domácností

Za první chytrý dům v moderním smyslu slova je považován objekt, který si v roce 1950 postavil americký technik Emil Mathias. Dům s označením Push Button Manor („panství na tlačítko“) umožňoval na stisk tlačítka ovládat žaluzie či zavírání oken. Vybaven byl mimo jiné větrem poháněným mlýnkem na kávu, systémem pro noční kontrolu zamčených dveří či osvětleným kosmetickým zrcátkem, které reagovalo na otevření šuplíku.

Další významný pokrok byl zaznamenán v roce 1957, kdy byl v Disneylandu představen modelový chytrý dům jako součást výstavní expozice. Tato instalace zahrnovala například hands-free telefon, výškově nastavitelný chytrý dřez a většinu vybavení vyrobeného z plastu.

V roce 1966 byl vyvinut prototyp zařízení ECHO IV, které je považováno za první chytré domácí zařízení. Tento systém o hmotnosti přibližně 360 kg umožňoval monitorování teploty v domě a základní ovládání spotřebičů. Do sériové výroby však uveden nebyl. [3]

Opravdový rozmach v oblasti automatizovaných a chytrých domů pak nastal po nástupu nového milénia, kdy se tento koncept začal stávat čím dál populárnějším. Na vzestupu je i v současnosti, například jen v Evropě se počet chytrých domácností mezi lety 2014 a 2017 ztrojnásobil. V České republice drží chytré domácnosti momentálně

čtyřprocentní podíl, zájem o ně se ale zvláště mezi mladou generací zvyšuje a do pěti let by jich zde mohlo být až patnáct procent. Komfort bydlení může být výrazně zvýšen díky chytrým domácnostem ovládaným prostřednictvím smartphonu či tabletu. Tyto systémy bývají navrhovány s ohledem na úsporu energií. [4]

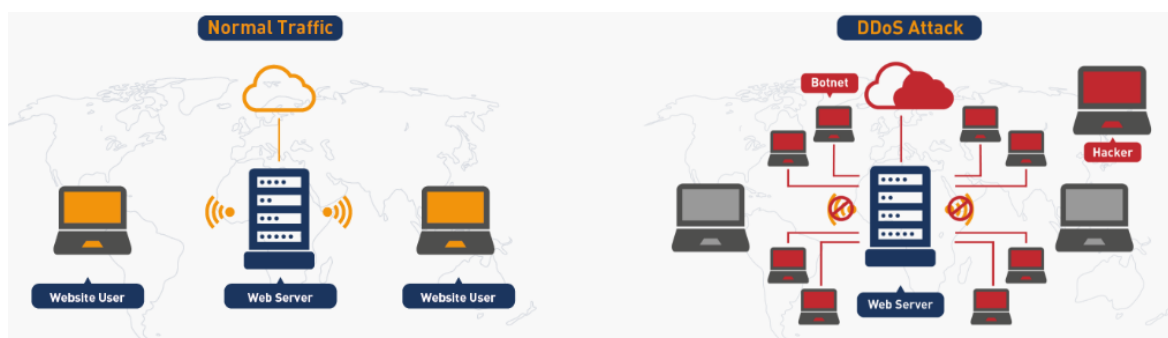
1.3 Bezpečnost chytré domácnosti

Bezpečnostní aspekty chytré domácnosti zahrnují jak kybernetická rizika spojená s připojením k internetu, tak i ochranu před fyzickými hrozbami, jako jsou výpadky elektrické energie nebo selhání jednotlivých zařízení.

1.3.1 Kybernetická bezpečnost chytré domácnosti

1. **Nezabezpečená Wi-Fi síť** – Nešifrovaná nebo nedostatečně zabezpečená Wi-Fi síť je považována za snadný cíl útoků. V případě připojení zařízení chytré domácnosti ke stejné síti jako počítače nebo mobilní telefony je třeba počítat s rizikem, že může být prostřednictvím zranitelného zařízení získán neoprávněný přístup k citlivým datům. Doporučené opatření zahrnuje vytvoření oddělené Wi-Fi sítě určené výhradně pro zařízení chytré domácnosti, použití šifrování WPA3 a silného hesla s délkou alespoň 12 znaků včetně kombinace velkých a malých písmen, číslic a speciálních znaků. [5]
2. **Slabé nebo výchozí heslo u zařízení** – Mnoho chytrých zařízení je dodáváno s výchozím heslem, které je veřejně dostupné na internetu. Použití slabého nebo opakovaného hesla významně zvyšuje riziko neoprávněného přístupu k systému. Doporučené opatření zahrnuje změnu výchozího hesla za silné a jedinečné a aktivaci dvoufázového ověření (2FA) pro maximální úroveň zabezpečení. [5]
3. **Neaktualizovaný firmware a software** – Zastaralý firmware často obsahuje bezpečnostní zranitelnosti, které mohou být zneužity útočníky. Vzhledem k tomu, že mnoho uživatelů neprovádí pravidelné aktualizace, stávají se tato zařízení snadným cílem kybernetických útoků. Doporučené opatření zahrnuje pravidelnou aktualizaci softwaru a firmwaru všech zařízení a povolení automatických aktualizací, pokud to zařízení umožňuje. [5]
4. **Botnet útoky a IoT** – Zařízení internetu věcí (IoT) představují potenciálně rozsáhlou plochu pro kybernetické útoky, jelikož mohou být zneužita k vytváření rozsáhlých botnetů (Obrázek 1). Tento stav je způsoben nejen rostoucí popularitou

těchto zařízení, ale také jejich často nedostatečným zabezpečením. IoT zařízení bývají dodávána s výchozími, snadno zneužitelnými přihlašovacími údaji a aktualizace jejich firmwaru bývají ze strany uživatelů zanedbávány. Jako příklad může být uveden botnet Mirai, který v roce 2016 infikoval tisíce IoT zařízení a následně je využil k rozsáhlým DDoS útokům. [6]



Obrázek 1 - Ukázka běžného provozu sítě a její zneužití pomocí DDoS útoku

(zdroj:[7])

5. **Zneužití cloudových služeb** – Cloudové služby mohou být cílem útoků typu DDoS, pokusů o neoprávněný přístup, zneužití cloudových API rozhraní nebo nedostatečně zabezpečených úložišť. V důsledku těchto útoků může dojít k dočasnému vyřazení služby z provozu. Nedostatečné šifrování dat může při narušení bezpečnosti vést k jejich úniku. Riziko ztráty dat však existuje i při použití silného šifrování (například v případě kompromitace uživatelského účtu). [8]

1.3.2 Ochrana chytré domácnosti před fyzickými hrozbami

1. **Výpadky elektrického proudu** – Záložní zdroj, označovaný jako UPS (nepřerušitelný zdroj napájení), slouží k zajištění napájení při výpadku dodávky elektrické energie ze sítě. Tímto způsobem lze předejít rizikům spojeným s náhlým přerušením napájení. UPS jednotky jsou vybaveny vysokokapacitními akumulátory, které se při výpadku automaticky aktivují a zajišťují plynulé napájení připojených zařízení. Přepnutí probíhá rychle, automaticky a bez negativního dopadu na provoz těchto zařízení. [9]
2. **Výpadky internetového připojení** – V případě delšího výpadku internetového připojení může být jako dočasné řešení využit mobilní hotspot z chytrého telefonu, který je dnes běžnou funkcí u zařízení s aktivními mobilními daty. Některé typy routerů rovněž umožňují automatické přepnutí na mobilní síť v případě ztráty

primárního připojení. U klíčových prvků chytré domácnosti, jako jsou chytré zámky nebo garážová vrata, je důležité zajistit dostupnost manuální alternativy (například fyzických klíčů, ručního ovládání termostatů nebo spínačů světel a zásuvek). [10]

3. **Další bezpečnostní prvky** – Senzory pohybu: Pohybová čidla jsou elektronická zařízení, která slouží k detekci pohybu v okolním prostředí. Tato čidla jsou vybavena senzory, kterými jsou sledovány změny teploty, intenzity světla nebo infračerveného záření. Po zaznamenání pohybu bývá automaticky aktivováno připojené osvětlení nebo je spuštěna jiná akce. [11]

Senzory dveří a oken: Jsou určeny k monitorování stavu otevření nebo zavření těchto prvků v domácnosti. Mohou být integrovány do systému chytré domácnosti, kde je jejich stav vyhodnocován a v případě změny může být odesláno upozornění na mobilní zařízení uživatele. Tímto způsobem je zvýšena úroveň zabezpečení objektu a zároveň je umožněno sledování přístupu do objektu v reálném čase. [12]

Detektory kouře a plynu: Zařízení jsou určena k včasné detekci vznikajícího požáru nebo úniku plynu. V případě zaznamenání nebezpečí je vyvolán akustický nebo vizuální poplach. Tato zařízení jsou určena k instalaci v domácnostech za účelem zvýšení úrovně bezpečnosti. [13]

1.4 Technologie používané v chytré domácnosti

Představen je přehled technologií, které umožňují fungování chytré domácnosti. Pozornost je zaměřena zejména na komunikační protokoly, řídicí platformy a chytrá zařízení tvořící základní stavební prvky moderní domácí automatizace.

1.4.1 Komunikační protokoly a standardy

1. Drátové technologie:

- a) KNX – Různá přenosová média, například kroucená dvoulinka (KNX TP), Ethernet (KNX IP), bezdrátový přenos (KNX RF) nebo přenos po silovém vedení (KNX PL), jsou podporována standardem KNX. Tím je zajištěna vzájemná kompatibilita zařízení od různých výrobců, která mohou být flexibilně kombinována do jednotného systému. Každé zařízení je vybaveno vlastním mikroprocesorem a pro provoz není vyžadována centrální řídicí jednotka. [14]

- b) BACnet – je komunikační protokol určený pro automatizaci budov, který podporuje různá přenosová média, jako je Ethernet (BACnet/IP), RS-485 (BACnet MS/TP), RS-232 (BACnet PTP), ARCNET a LonTalk. Díky standardu ISO 16484-5 je zajištěna interoperabilita mezi zařízeními různých výrobců. Tím je umožněna jejich snadná integrace do jednotného systému. Protokol BACnet je využíván zejména v systémech HVAC, řízení osvětlení, požární ochraně a dalších oblastech automatizace budov. [15]
- c) Modbus – je komunikační protokol založený na architektuře master/slave nebo klient/server, navržený pro snadnou, spolehlivou a rychlou komunikaci mezi automatizačními zařízeními a zařízeními pro procesní instrumentaci. Mezi hlavní přednosti patří standardizovaná výměna dat v souladu s normami, vysoká přenosová rychlost a nezávislost na konkrétní technologii nebo výrobci. Podporována je komunikace prostřednictvím sériových rozhraní (RS-232 a RS-485) a Ethernetu, a to s využitím režimů přenosu Modbus TCP, Modbus RTU a Modbus ASCII. Díky této flexibilitě může být zajištěna integrace různých zařízení do jednotného systému bez ohledu na výrobce. [16]

2. Bezdrátové technologie:

- a) Bluetooth – Tato technologie je určena pro bezdrátovou komunikaci na krátké vzdálenost. Přímé propojení dvou zařízení bez nutnosti využití podpůrné síťové infrastruktury, jako je bezdrátový router nebo přístupový bod. V současnosti je nejčastěji využívána k připojení zařízení, jako jsou bezdrátová sluchátka, klávesnice, myši a reproduktory, k osobním počítačům nebo mobilním telefonům. [17]
- b) Zigbee – Jedná se o technologii navrženou pro komunikaci jednodušších zařízení, která pracují při nižších přenosových rychlostech (do 250 kbps). Výhodou tohoto protokolu je nízká energetická náročnost při přenosu malého množství dat, proto je vhodný pro zařízení napájená z baterie. [18]
- c) Z-Wave – Protokol Z-Wave je určen pro automatizované řízení zařízení v chytré domácnosti a patří mezi významné technologie v oblasti internetu věcí (IoT). Jeho hlavní výhodou je schopnost vzájemné kompatibility zařízení různých výrobců při zachování jednoduchého ovládání a stabilního provozu. [19]
- d) Thread – Tato technologie byla speciálně vyvinuta pro aplikace chytré domácnosti s cílem výrazně zlepšit propojení mezi jednotlivými produkty. Zařízení určená pro

Apple Home mohou být díky Threadu propojena nejen pomocí Wi-Fi nebo Bluetooth, ale také prostřednictvím této technologie. Síť Thread je koncipována jako jiný typ sítě, v níž mohou být chytré rolety, světla, termostaty a senzory vzájemně propojeny přímo mezi sebou bez nutnosti spojení prostřednictvím centrálního uzlu, jakým je například most. Samostatný komunikační prvek v síti Thread není vyžadován. Pokud dojde k selhání jednoho zařízení, jsou data jednoduše předávána dalšímu zařízení v této síti. [20]

- e) Wi-Fi – Jedná se o bezdrátovou síťovou technologii, která je využívána pro zařízení, jako jsou počítače (notebooky a stolní počítače), mobilní zařízení (chytré telefony a nositelná elektronika) a další vybavení (tiskárny, videokamery), připojit se k internetu. Prostřednictvím této technologie mohou jednotlivá zařízení vzájemně komunikovat a vytvářet síť pro přenos dat. [21]

Standardy Wi-Fi jsou definovány organizací IEEE (Institute of Electrical and Electronics Engineers) v rámci protokolu 802.11. Těmito standardy je určováno, jakým způsobem jsou bezdrátová zařízení vzájemně propojena a jak se připojují k internetu. Vývojem těchto standardů bylo dosaženo výrazného zlepšení v oblasti rychlosti, dosahu a spolehlivosti bezdrátového připojení. [22]

Tabulka 1 - Standardy Wi-Fi

Standard Wi-Fi	Frekvenční pásmo	Maximální rychlost přenosu	Šířka kanálu
802.11 b	2,4 GHz	11 Mb/s	20 MHz
802.11 a	5 GHz	54 Mb/s	20 MHz
802.11 g	2,4 GHz	54 Mb/s	20 MHz
802.11 n	2,4 GHz, 5GHz	600 Mb/s	20, 40 MHz
802.11 ac	5 GHz	3,5 Gb/s	20, 40, 80, 160 MHz
802.11 ax	2,4 GHz, 5 GHz	9,6 Gb/s	20, 40, 80, 160 MHz
802.11 be	2,4GHz, 5GHz, 6GHZ	46 Gb/s	20, 40, 80, 160 MHz

(zdroj:[22])

3. Hybridní technologie

- a) Matter – Jedná se o nový standard vytvořený sdružením Connectivity Standards Alliance (CSA), jehož cílem je sjednocení různých protokolů (včetně Zigbee, Thread a dalších) s cílem usnadnit interoperabilitu mezi zařízeními v prostředí internetu věcí. [23]

1.4.2 Chytrá zařízení a senzory

1. Osvětlení a energetika:

- a) Chytrá žárovka – Je součástí inteligentního osvětlovacího systému. Regulace svítivosti je zajišťována prostřednictvím dálkového ovladače nebo mobilní aplikace. Běžná žárovka je v tomto případě doplněna o elektroniku, která umožňuje měnit barevný tón, odstín a intenzitu světla. [24]
- b) Termostaty – Zařízení sloužící k dálkovému řízení teploty v domácnosti. Prostřednictvím mobilního telefonu, tabletu nebo počítače může být teplota nastavena odkudkoli a kdykoli. [25]
- c) Zásuvky – Snadné zapojení do stávající elektrické sítě a ovládání připojeného spotřebiče je umožněno podle uživatelských požadavků. Elektrický proud je zapínán a vypínán automaticky dle nastaveného režimu. Tyto zásuvky zpravidla využívají připojení k domácí Wi-Fi síti nebo technologii Bluetooth a ovládají se prostřednictvím mobilních zařízení. [26]

2. Zabezpečení a monitoring:

- a) Chytré zámky – Jsou navrženy pro ovládání prostřednictvím chytrého telefonu nebo jiného zařízení připojeného k internetu. Tyto zámky bývají vybaveny technologií Bluetooth nebo Wi-Fi. Vzdálené ovládání je tímto umožněno. U některých modelů je k dispozici čtečkou otisků prstů nebo numerickým klávesovým panelem. [27]
- b) Kamery – Slouží ke zvýšení úrovně zabezpečení domu nebo pozemku. Streamování obrazu v reálném čase je umožněno, zpravidla prostřednictvím chytrého telefonu a detekovaný pohyb může být ohlášen formou notifikace. Nahrávání může být spuštěno manuálně nebo automaticky v kombinaci s pohybovým senzorem. Chytré kamery se využívají nejen k ochraně majetku, ale také k monitorování dětí, domácích mazlíčků či aktivních spotřebičů. [28]

- c) Detektory kouře – Zařízení sloužící k včasné detekci kouře vznikajícího v důsledku doutnání nebo hoření. Včasné varování před vznikajícím požárem je jimi umožněno a zároveň je zajištěno zvýšení bezpečnosti objektu. [29]

3. Automatizace a řízení

- a) Chytré rolety – Jsou určeny k zajištění stínění, tepelné izolace a zvýšení bezpečnosti domácnosti. Úplné zatemnění interiéru i během dne je jimi umožněno a při plném zatažení fungují jako ochranný prvek proti vnějším vlivům. V zimním období je jimi minimalizována tepelná ztráta, v létě je zajištěna ochrana před přehříváním místností. Zároveň může být ztížen od pokusů o neoprávněný vstup. [30]
- b) Detektory pohybu a přítomnosti – Jsou elektronická zařízení určená k detekci pohybu v okolním prostředí. Senzory pro změnu teploty, intenzitu světla nebo infračerveného záření jsou v těchto čidlech integrovány. Po zaznamenání pohybu bývá automaticky aktivováno připojené osvětlení nebo jiná akce. [11]
- c) Senzory úniku vody – Jsou určeny k detekci vody v místech, kde se její výskyt neočekává. V případě zaznamenání úniku jsou uživateli zasílána varování prostřednictvím zvukového signálu nebo notifikace do mobilního zařízení. Tato zařízení mohou být integrována do systému chytré domácnosti nebo fungovat samostatně. Díky včasné detekci je možné předcházet škodám způsobeným vodou a je umožněna rychlá reakce na vzniklou situaci. [31]

4. Multimédia a zábava

- a) Chytré reproduktory – Reproduktory tohoto typu jsou připojeny k internetu. Přijímání hlasových povelů je těmito zařízeními umožněno. Na základě zadaných pokynů je zajištěno nejen zpracování a odpověď na dotaz, ale také ovládání propojených zařízení v domácnosti. [32]
- b) Domácí kino – Je určeno k vytvoření filmového zážitku srovnatelného s kinosálem. Základem systému je prostorové ozvučení tvořené předními, zadními a centrálním reproduktorem. Pro dosažení plného zvukového efektu je doporučeno doplnění o aktivní subwoofer. [33]

2 SYSTÉM OPENHAB

OpenHAB (Open Home Automation Bus) představuje open-source platformu pro automatizaci domácnosti, která umožňuje integraci různých zařízení a technologií do jednotného systému bez ohledu na jejich výrobce nebo použitý komunikační protokol. Díky široké kompatibilitě je podporováno více než 300 integrací, včetně technologií jako Zigbee, Z-Wave nebo MQTT.

Využívány jsou koncepty Things (zařízení), Items (položky) a Rules (pravidla), které umožňují vytváření flexibilních automatizačních scénářů přizpůsobených specifickým potřebám uživatele.

Jednou z klíčových výhod systému OpenHAB je nezávislost na cloudových službách. Tím je zajištěna vyšší míra soukromí a lepší kontrola nad chytrou domácností. Platformu lze provozovat na různých zařízeních, včetně Raspberry Pi, Windows, Linux a Dockeru.

Z technického hlediska je OpenHAB vyvíjen v jazyce Java a využívá modularitu prostřednictvím platformy OSGi. Apache Karaf slouží jako kontejner pro správu služeb, Eclipse Equinox poskytuje běhové prostředí a Jetty je integrován jako HTTP server. [34]

- a) OSGi je definován jako modulární přístup a specifikace, pomocí nichž mohou být vytvářeny robustní, vysoce oddělené a dynamické aplikace v jazyce Java. Tato specifikace je zpřístupněna prostřednictvím několika implementací. Mezi open-source varianty patří například Apache Felix, Eclipse Equinox a Knopflerfish. [35]
- b) Apache Karaf – Jedná se o modulární aplikační kontejner postavený na OSGi, který je navržen pro dynamické nasazování a správu služeb. Podporovány jsou různé OSGi frameworky, například Apache Felix nebo Eclipse Equinox. K dispozici jsou také robustní administrativní nástroje, včetně konzole a možnosti vzdáleného přístupu prostřednictvím SSH. Snadná konfigurace a dynamické načítání modulů jsou tímto kontejnerem umožněny, což zajišťuje vhodnost jak pro podniková řešení, tak pro mikroslužby. Správa aplikací a jejich závislostí je usnadněna podporou systému Karaf Features pro zavádění (nasazování) modulů a komponent. Možnost provozu v lokálním prostředí i v cloudu přispívá k vyšší flexibilitě nasazení. [36]
- c) Eclipse Equinox – Eclipse Equinox je označován jako implementace základní specifikace OSGi frameworku, v jejímž rámci může být poskytována sada modulů pro různé volitelné OSGi služby a infrastruktura potřebná pro provoz systémů

založených na tomto standardu. Je začleněn do projektu Eclipse a je využíván jako výchozí modulární běhové prostředí (runtime) pro aplikace vyvíjené v jazyce Java. [37]

- d) Jetty – Tento webový server a kontejner pro webové aplikace je napsán v jazyce Java a je charakterizován vysokou škálovatelností a paměťovou efektivitou. Podporovány jsou protokoly jako HTTP/1, HTTP/2, HTTP/3 a WebSocket. Prostřednictvím tohoto serveru je umožněna efektivní komunikace mezi klienty a servery. Jetty je poskytován jako open-source projekt pod licencemi EPL2 a Apache 2.0, které zajišťují jeho volné komerční využití a distribuci. Díky modulární architektuře je tento server vhodný pro použití v embedded systémech, cloudových službách i tradičních webových aplikacích. [38]

2.1 Historie a vývoj

Projekt openHAB byl založen Kaiem Kreuzerem v roce 2010 a od té doby je rozvíjen s přispěním mnoha spolupracujících vývojářů. Na konci roku 2013 bylo rozhodnuto o částečném pokračování vývoje formou oficiálního projektu Eclipse pod názvem Eclipse SmartHome. Oba projekty (openHAB a Eclipse SmartHome) jsou nadále vyvíjeny paralelně. Architektura a funkcionality openHABu byly z velké části začleněny do Eclipse SmartHome. Oficiální web openHABu je nadále využíván jako hlavní zdroj pro vývojáře doplňků. [39]

Systém openHAB zvítězil v soutěži IoT Challenge 2013 a získal ocenění JavaOne Duke's Choice Award 2013. Dále byl nominován na JAX Innovation Award 2014 a ocenění People's Choice v rámci Postscapes IoT Awards 2014/2015 bylo systému openHAB uděleno. [40; 41; 42]

2.2 Architektura systému openHAB

V této části je popsána architektura systému openHAB, která je postavena na principu oddělení jednotlivých vrstev reprezentujících fyzická zařízení, jejich funkcionality a logickou automatizaci.

2.2.1 Bindings

Doplňky (Bindings) jsou popisovány jako softwarové balíčky, jejichž prostřednictvím může být umožněna komunikace mezi chytrými zařízeními a samotným systémem.

Propojení konkrétního zařízení s jeho definicí ve formě zařízení (Thing) a zajištění překladu příkazů mezi openHABem a daným zařízením je realizováno prostřednictvím těchto doplňků. Jsou zpřístupněny v sekci Doplňky systému (Add-ons) a jejich počet je průběžně rozšiřován. Každý doplněk bývá doplněn podrobným návodem k použití, včetně popisu podporovaných zařízení (Things), jejich kanálů (Channels) a ukázek propojení s položkami (Items) i s uživatelským rozhraním systému. [43]

2.2.2 Things

Zařízení (Things) jsou definována jako entity, které mohou být fyzicky nebo logicky přidány do systému. Prostřednictvím nich mohou být zajišťovány určité funkce. Nejedná se vždy výhradně o fyzická zařízení. Jako zařízení může být reprezentována také webová služba nebo jiný spravovatelný zdroj informací a funkcionalit. Význam těchto entit je zejména při nastavení a konfiguraci systému. Během samotného provozu je častěji pracováno s položkami (Items).

Každé zařízení (Thing) je tvořeno jedním nebo více kanály (Channels), kterými jsou reprezentovány konkrétní funkce nebo hodnoty daného zařízení. Samostatný kanál pro nastavení barevné teploty a jiný pro změnu barvy může být v případě chytré žárovky vymezen. Různé funkce jednoho fyzického zařízení jsou odlišeny odpovídajícími kanály. V případě datových zdrojů je zařízením například modul s informacemi o počasí. Údaje o teplotě, tlaku a vlhkosti jsou těmito kanály poskytovány.

Kanály zařízení jsou následně propojeny s položkami (Items). Je vytvářen most mezi fyzickou (nebo logickou) vrstvou systému a její virtuální reprezentací. Po vytvoření tohoto propojení mohou být události přijímány daným zařízením ze strany příslušné položky a zároveň mohou být aktualizace hodnot do položky odesílány kanálem, s nímž je zařízení propojeno. [44]

2.2.3 Channels

Kanály (Channels) jsou definovány jako logické propojení mezi zařízením (Thing) a položkou (Item). Z definice konkrétního zařízení je odvozován každý kanál, je určeno, jakým způsobem může být zajištěna komunikace mezi tímto zařízením a příslušnou položkou.

Kanály jsou vytvářeny při definici zařízení a prostřednictvím nich jsou reprezentovány jednotlivé funkce nebo měřené hodnoty daného zařízení. Při nastavování systému vývojář

u každé položky rozhoduje, ke kterému kanálu daného zařízení ji připojí. Prostřednictvím tohoto propojení jsou informace v systému openHAB přenášeny mezi fyzickým zařízením a jeho virtuální reprezentací, tedy mezi Thing a Item. [43]

2.2.4 Items

Abstraktní vrstva v systému openHAB je tvořena položkami (Items), které slouží k modelování funkcí a vlastností jednotlivých zařízení v rámci domácí automatizace. Zařízení (Things) jsou v systému openHAB chápána jako konkrétní fyzické komponenty nebo služby, ale položky jsou považovány za jejich virtuální reprezentaci. Jsou klíčovým prvkem při návrhu uživatelských rozhraní a při vytváření automatizačních pravidel (Tabulka 1).

Položky lze dále seskupovat do skupinových položek (Group Items) pro efektivní řízení více prvků současně. Typickým příkladem je seskupení všech světel v jedné místnosti, která jsou poté ovládána jedním společným příkazem. [45]

Tabulka 2 - Příklady prvků Items

Jméno	Popisek
Switch	Reprezentuje dvoustavové zařízení, jako je světlo(zapnuto/vypnuto)
Dimmer	Nesoucí hodnotu v procentech pro stmívače
Location	Ukládá GPS souřadnice
Number	Ukládá hodnoty v číselném formátu
String	Ukládá textové řetězce
DateTime	Ukládá datum a čas
Contact	Sleduje stav kontaktů, např. otevření nebo zavření dveří či oken

(zdroj: [45])

2.2.5 Rules

Automatizace procesů v rámci systému openHAB je zajišťována pravidly. Každé pravidlo je tvořeno skriptem, prostřednictvím něhož jsou po spuštění vykonávány různé úlohy, například zapnutí světel prostřednictvím změny stavu položek (Items), provádění matematických výpočtů či spuštění časovačů.

Každé pravidlo se skládá ze tří hlavních částí:

1. Spouštěče(Triggers): Události, které aktivují pravidlo, například změna stavu položky nebo určitý časový okamžik.
2. Podmínky (Conditions): Volitelné kontroly, které musí být splněny, aby se akce pravidla provedly.
3. Akce (Actions): Úkoly, které se vykonají, když je pravidlo spuštěno a podmínky jsou splněny, jako je odeslání příkazu položce nebo spuštění skriptu.

Pravidla v openHABu jsou založena na událostech, na které systém reaguje provedením definovaných akcí. Tímto způsobem je umožněno dosažení plně automatizovaného chování domácnosti bez nutnosti zásahu uživatele.

Pro začínající uživatele byl zaveden vizuální nástroj Blockly. Pravidla mohou být tvořena prostřednictvím grafických bloků bez potřeby programován. [46; 47]

3 INSTALACE A KONFIGURACE SYSTÉMU OPENHAB

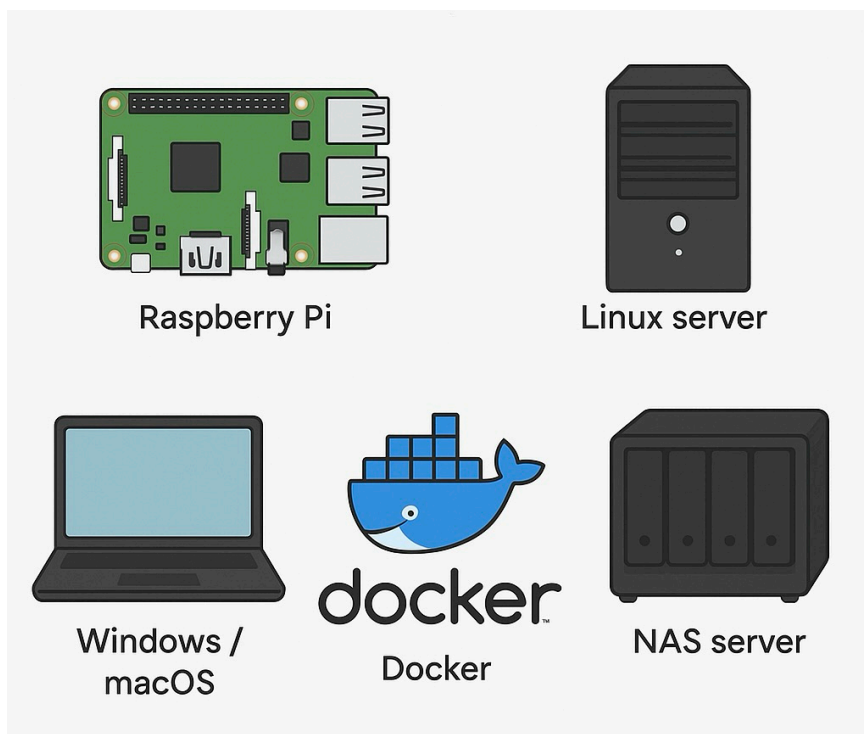
V této části je popsán postup výběru vhodné platformy a hardwaru pro provoz systému openHAB.

3.1 Výběr platformy a hardware

Před samotnou instalací systému openHAB by měla být zvolena vhodná platforma a hardwarové řešení, na kterém má být celý systém provozován. Tento výběr by měl být proveden s ohledem na požadavky na výpočetní výkon, energetickou efektivitu, dostupnost komponent, udržitelnost a celkovou spolehlivost provozu systému v dlouhodobém horizontu.

Systém openHAB byl kompletně napsán v jazyce Java, a proto je závislý pouze na Java Virtual Machine (JVM), která je dostupná pro většinu platform. OpenHAB může být provozován na různých verzích operačních systémů macOS a Windows i na mnoha variantách Linuxu (například Ubuntu či Raspbian). [48]

Možnosti platform pro běh openHABu jsou znázorněny na obrázku (Obrázek 2).



Obrázek 2 - Možnosti platform pro běh systému openHAB

(zdroj: [49])

3.1.1 Minimální požadavky na instalaci openhabianu

Raspberry Pi – openhabian

Pro zajištění optimálního výkonu systému openHAB je doporučováno použití modelu Raspberry Pi 4 nebo 5 s kapacitou 2 GB nebo 4 GB RAM. Starší modely Raspberry Pi nebo varianty s nižší kapacitou RAM mohou být využívány pro menší instalace systému, avšak pro rozsáhlejší a náročnější nasazení je vhodnější volba novějších modelů. Vzhledem k požadavku na 64bitový procesor ve verzi openHAB 5 byla ukončena podpora pro modely starší než Raspberry Pi 3. Pokud má být použito zařízení s 1 GB RAM, doporučuje se použít 32bitový režim, protože běh 64bitového režimu na tomto zařízení může způsobovat problémy z důvodu nedostatečné paměti. [50]

3.1.2 Raspberry Pi imager

Raspberry Pi Imager představuje rychlý a jednoduchý způsob, jakým může být operační systém Raspberry Pi OS a další operační systémy nainstalovány na microSD kartu, která je poté připravena k použití s jednodeskovým počítačem Raspberry Pi. [51]

3.1.3 Etcher

Etcher je považován za výkonný nástroj pro zapisování obrazů operačního systému. Je postaven na webových technologiích. Cílem je, aby byl zajištěn bezpečný proces zápisu na SD kartu nebo USB zařízení. Systém chrání před nechtěným přepsáním pevných disků, ověřuje správnost zápisu každého bajtu dat a nabízí další funkce. [52]

II

PRAKTICKÁ ČÁST

4 INSTALACE SYSTÉMU OPENHAB – OPENHABIANU

Pro účely této bakalářské práce bylo zvoleno zařízení Raspberry Pi, které představuje nízkonákladový a všestranný jednodeskový počítač nabízející plnohodnotné počítačové prostředí v kompaktním hardwarovém provedení. Díky své dostupnosti, nízké spotřebě energie a dostatečnému výkonu je Raspberry Pi považováno za ideální řešení pro provoz systému openHAB v domácím prostředí.

Pro instalaci systému openHABian, který je připravenou distribucí určenou právě pro Raspberry Pi, jsou k dispozici dvě hlavní možnosti instalace, které budou v následujících částech podrobně popsány.

4.1.1 Instalace pomocí Raspberry Pi imager

Prvním krokem instalace systému openHABian je stažení a instalace nástroje Raspberry Pi Imager, který slouží k zápisu obrazu systému na paměťovou kartu. Po spuštění aplikace je nejprve zvoleno cílové zařízení, pro které bude systém instalován. Před samotným výběrem systému musí být do počítače vložena SD karta, neboť na ni bude následně nahrán instalační obraz.

V sekci pro výběr operačního systému je vybrána položka „Other specific-purpose OS“, následně kategorie „Home assistants and home automation“, a poté možnost „openHAB“. V závislosti na použitém modelu Raspberry Pi musí být zvolena odpovídající architektura. Tedy 32bitovou nebo 64bitovou verzi systému. Poté je vybráno úložiště, tj. konkrétní SD karta, na kterou bude systém nainstalován.

Po úspěšném zápisu obrazu je SD karta vložena do zařízení Raspberry Pi, které je následně připojeno k síti pomocí ethernetového kabelu a připojeno ke zdroji napájení. Instalace systému openHABian obvykle probíhá automaticky a trvá přibližně 10 až 30 minut, v závislosti na výkonu zařízení a použité paměťové kartě.

Po dokončení instalace lze přistoupit do uživatelského rozhraní openHABu prostřednictvím webového prohlížeče na adrese <http://openhabian:8080>. V případě potřeby je možné namísto názvu openhabian použít přímo IP adresu zařízení v lokální síti.

4.1.2 Instalace pomocí Etcher

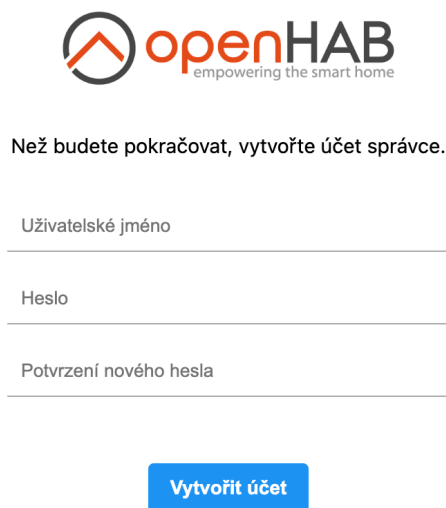
Jako alternativní způsob instalace systému openHABian je využíván nástroj Etcher, který slouží k zápisu obrazů operačních systémů na paměťové karty. Nejprve byl nainstalován

software Etcher a z oficiálního webu openHAB byl stažen odpovídající obraz systému openHABian určený k zavedení ze SD karty.

Po spuštění aplikace Etcher byla v kroku „Select image“ vybrána stažená verze obrazu systému openHABian. Následně bylo jako cílové úložiště zvoleno zařízení typu SD karta, na které byl obraz systému zapsán. Poté byl zahájen samotný proces zápisu. Po jeho úspěšném dokončení byla SD karta vyjmuta z počítače, vložena do zařízení Raspberry Pi a postup dále pokračoval stejným způsobem jako v předchozím případě, tedy připojením k síti, zapojením napájení a automatickým dokončením instalace systému openHABian.

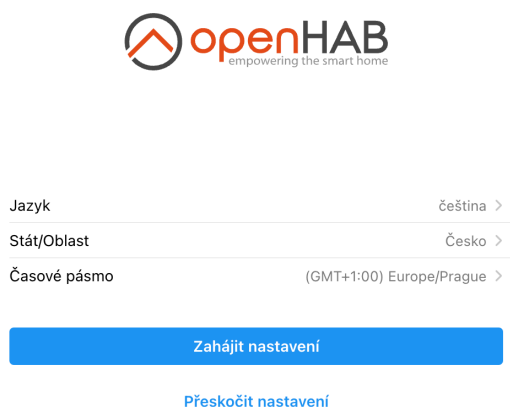
4.2 Základní nastavení systému

Po otevření stránky <http://localhost:8080> nebo http://<IP_adresa>:8080 byl zobrazen úvodní formulář pro vytvoření administrátorského účtu (Obrázek 3). V tomto kroku bylo zvoleno vhodné uživatelské jméno, které je následně možné využít i při propojení s openHAB Cloud. Heslo bylo nastaveno tak, aby bylo dostatečně silné a nefigurovalo v databázích kompromitovaných údajů. Pokračovalo se potvrzením vytvoření účtu kliknutím na tlačítko „Vytvořit účet“.



The screenshot shows the openHAB logo at the top. Below it, the text reads: "Než budete pokračovat, vytvořte účet správce." (Before you continue, create an administrator account). The form contains three input fields: "Uživatelské jméno" (Username), "Heslo" (Password), and "Potvrzení nového hesla" (Confirm new password). At the bottom of the form is a blue button labeled "Vytvořit účet" (Create account).

Obrázek 3 - Vytvoření účtu správce
(zdroj: vlastní)



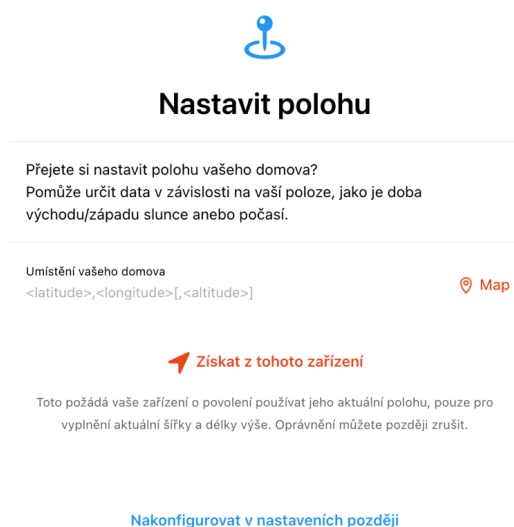
The screenshot shows the openHAB logo at the top. Below it, there are three dropdown menus: "Jazyk" (Language) set to "čeština", "Stát/Oblast" (State/Region) set to "Česko", and "Časové pásmo" (Time zone) set to "(GMT+1:00) Europe/Prague". At the bottom of these settings is a blue button labeled "Zahájit nastavení" (Start settings). Below this button is a link labeled "Přeskočit nastavení" (Skip settings).

Obrázek 4 - Nastavení jazyka, státu a časového pásma
(zdroj: vlastní)

Dále bylo přistoupeno k nastavení geografických údajů, jako je jazyk, stát a časové pásmo, které jsou důležité pro správnou lokalizaci systému (Obrázek 4). Toto nastavení

může být přeskočeno, nicméně jeho dodatečné doplnění je doporučeno pro správnou funkci lokalizovaných služeb a automatizačních funkcí systému.

V následujícím kroku bylo vyzváno k zadání přibližné polohy domácnosti, případně umístění zařízení Raspberry Pi (Obrázek 5). Toto nastavení slouží například pro správné získávání údajů o východu a západu slunce, které mohou být dále využity v pravidlech automatizace.



Obrázek 5 - Nastavení polohy

(zdroj: vlastní)



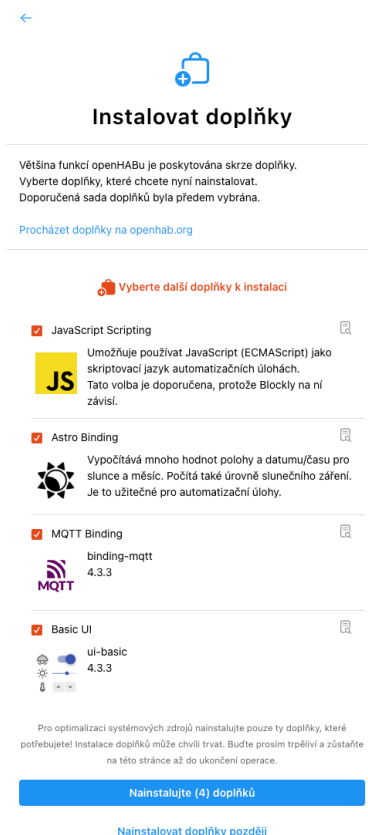
Obrázek 6 - Menu pro doplňky pro ukládání dat

(zdroj: vlastní)

Následně byl systémem nabídnut výběr doplňků pro perzistenci (Obrázek 6), tedy pro ukládání stavů jednotlivých položek (Items). Tyto doplňky jsou klíčové pro uchovávání a opětovné načítání hodnot v systému. Tato funkčnost je důležitá nejen pro analýzu historických dat, ale také pro správné fungování pravidel, vizualizací a obnovu systému po restartu.

V dalším kroku byly vybrány funkční doplňky, které rozšiřují schopnosti systému openHAB. Výchozí sada doporučených doplňků (Obrázek 7) byla předem označena, nicméně uživatel má možnost přidat další doplňky podle svých preferencí. Tyto moduly slouží zejména k integraci s konkrétními zařízeními, technologiemi a cloudovými službami.

Všechny dostupné doplňky je možné spravovat i později v sekci Obchod doplňků, která zahrnuje jak oficiální rozšíření, tak i komunitní řešení.



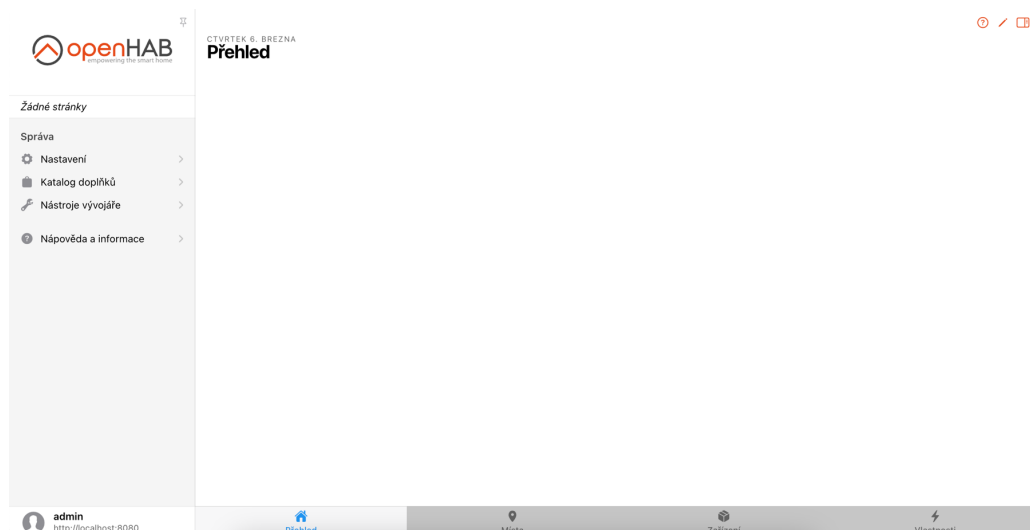
Obrázek 7 - Instalovat doplňky/Add-ons

(zdroj: vlastní)

Po dokončení instalačního průvodce byl uživatel automaticky přesměrován na hlavní stránku systému (Obrázek 8), kde může pokračovat v integraci chytré domácnosti. Přístup do uživatelského rozhraní je možný prostřednictvím adresy <http://localhost:8080/overview>, případně s využitím IP adresy zařízení v lokální síti.

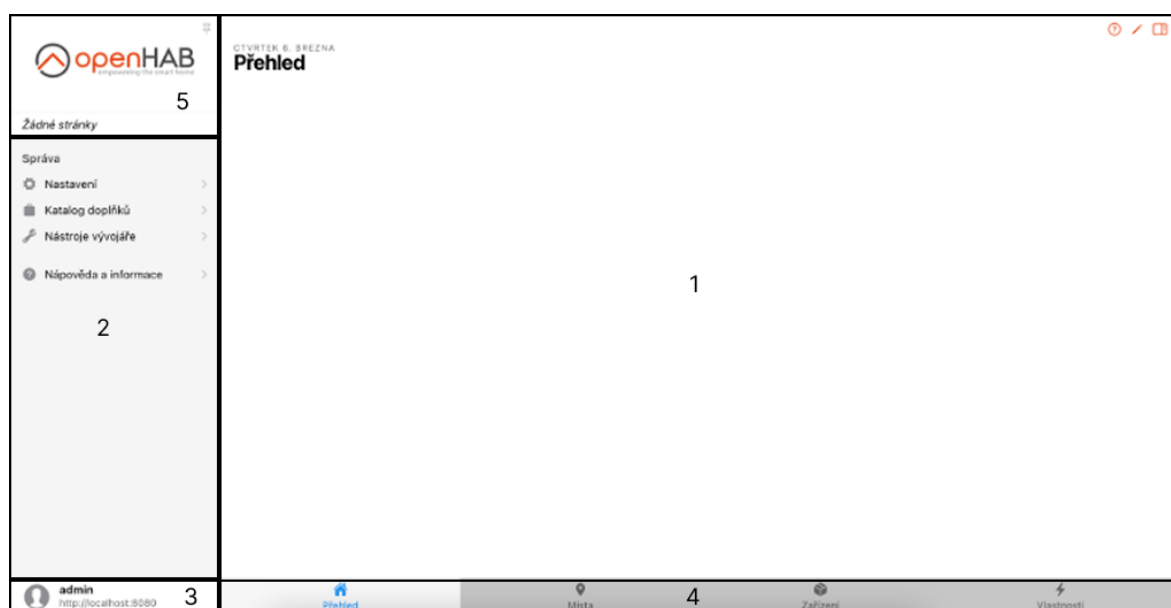
4.3 Popis jednotlivých částí na hlavní stránce

V této kapitole je popsána struktura základních prvků uživatelského rozhraní hlavní stránky systému openHAB. Jednotlivé sekce jsou pro větší přehlednost očíslovány (Obrázek 9) a označeny na přiloženém obrázku, který slouží jako vizuální pomůcka pro orientaci v prostředí aplikace.



Obrázek 8 - Hlavní stránka systému openHAB

(zdroj: vlastní)



Obrázek 9 – Rozdělení hlavní stránky

(zdroj: vlastní)

4.3.1 Hlavní obsah – číslo 1

Zobrazován je hlavní přehledový panel systému openHAB, kde jsou soustředěny klíčové informace o aktuálním stavu chytré domácnosti. Poskytovány jsou souhrnné údaje o jednotlivých zařízeních, jejich stavech, případných událostech a výstrahách. Panel slouží jako výchozí rozhraní pro sledování provozního stavu celé instalace.

Obsah této části lze upravovat prostřednictvím nabídky Nastavení → Stránky → Přehled, kde je možné definovat, které položky a vizualizace budou na hlavní stránce zobrazeny.

4.3.2 Navigační menu (Správa) – číslo 2

V této sekci uživatelského rozhraní jsou umístěny hlavní ovládací prvky systému. Zajištěn je přístup k částem Nastavení, Katalog doplňků, Nástroje pro vývojáře a Nápověda a informace. Menu slouží jako výchozí bod pro správu, konfiguraci systému, rozšiřování jeho funkcionality a údržbu.

1. Nastavení

a) Konfigurace

Zahrnuje správu zařízení, modelování domácnosti a definování uživatelského rozhraní.

- Things – Slouží ke správě fyzických zařízení (např. senzory, spínače, termostaty) a jejich propojení se systémem
- Model – Umožňuje vytvořit sémantický model domácnosti, který napomáhá lepší organizaci a řízení zařízení
- Items – Virtuální prvky systému (např. vypínače, teploměry), které jsou propojeny s fyzickými zařízeními a slouží k reprezentaci jejich stavů a ovládání v rámci uživatelského rozhraní
- Pages – Vlastní rozhraní pro ovládání domácnosti (např. přehledy, ovládací panely) může být navrženo podle individuálních požadavků uživatele.
- Transformations – Nastavení převodních pravidel, která překládají data do srozumitelnější podoby (např. z „Cool“ na „chlazení“)

b) Automatizace

Tato část slouží k nastavování automatizací v systému.

- Pravidla (Rules) - Definují automatické reakce systému na určité podněty (např. rozsvícení světla při detekci pohybu)
- Scény (Scenes) - Umožňují definovat a uložit konkrétní stavy zařízení pro snadnou aktivaci (např. „Noční režim“)

- Skripty (Scripts) - Nabízejí možnost vytváření vlastních skriptů pro pokročilou automatizaci
- Plánování (Schedule) - Slouží k časovému spouštění pravidel (např. v určitou hodinu nebo při východu slunce)

c) Systémová nastavení

Sekce Systémová nastavení v systému openHAB slouží ke správě základních funkcí a parametrů celého systému. Umožňuje například nastavení zabezpečení API, konfiguraci hlasového ovládání a správu zařízení, která čekají na připojení prostřednictvím tzv. Inboxu. Dále je možné přizpůsobit regionální nastavení a jednotky měření podle preferencí uživatele. Součástí této sekce je také správa doplňků třetích stran a přístup ke komunitnímu obchodu s rozšířeními, které umožňují další rozvoj a přizpůsobení systému. Tato část tedy poskytuje nástroje pro celkovou správu systému a jeho přizpůsobení konkrétním potřebám domácnosti.

2. Katalog doplňků

Katalog doplňků slouží k rozšiřování jeho funkcionality a integraci různých zařízení a služeb. Je rozdělen do několika tematických kategorií. Každá z nich nabízí specifické rozšíření, které lze dle potřeby nainstalovat.

- a) Doplňky (Bindings) - Moduly pro propojení s různými zařízeními a službami dostupné pro ovládání a monitorování v systému
- b) Automatizace (Automation) - Doplňky určené pro tvorbu pravidel, skriptů a šablon pro automatizaci činností v chytré domácnosti
- c) Transformace (Transformations) - Rozšíření pro převod dat do čitelnějších a srozumitelnějších formátů, využívána např. pro lokalizaci textů
- d) Persistence - Moduly pro ukládání historických dat, která lze později analyzovat nebo obnovit při restartu systému
- e) User Interfaces (Uživatelské rozhraní) - Zpřístupněny widgety pro rozhraní a alternativní uživatelské rozhraní k přizpůsobení vzhledu systému
- f) Systém Integrations (Systémová integrace) - Nástroje pro propojení openHABu s dalšími systémy a technologickými platformami
- g) Voice & Speech (Hlas a řeč) - Doplňky pro hlasové ovládání

3. Nástroje vývojáře

V této sekci jsou dostupné nástroje pro přizpůsobení a rozšíření systému. Jsou využívány zejména pokročilejšími uživateli při vytváření vlastních rozhraní nebo při práci s API.

- a) Widgets – Doplnky pro přidání a úpravu widgetů v uživatelském rozhraní
- b) Block Libraries – Knihovny bloků pro rozšíření Blockly editoru a tvorbu automatizačních skriptů
- c) API Explorer – Nástroj pro prozkoumání REST API systému a testování komunikace s jednotlivými zařízeními nebo službami

4. Nápověda a informace

Tato část poskytuje základní informace o aktuálním stavu systému a nabízí uživateli přístup k důležitým zdrojům a nastavením. Zobrazovány jsou zde údaje o verzi systému, včetně čísla verze a data sestavení. K dispozici jsou také odkazy na oficiální dokumentaci, uživatelské příručky, vývojářské návody a komunitní fórum, které mohou pomoci při řešení problémů nebo rozšiřování systému. V neposlední řadě je v této sekci možné upravit vzhled uživatelského rozhraní specificky pro dané zařízení, a to například přepnutím mezi světlým a tmavým motivem.

4.3.3 Uživatelský účet – číslo 3

Po kliknutí na uživatelský profil v levém dolním rohu hlavní stránky je uživatel přesměrován na stránku s nastavením účtu. Zobrazovány jsou základní informace o aktuálně přihlášeném uživateli, jako je uživatelské jméno a případně další údaje. Pod těmito informacemi se nachází možnost změny hesla prostřednictvím odkazu „Změnit heslo“. Následuje seznam aktivních relací, který poskytuje přehled o tom, kdy a odkud byl uživatel přihlášen. V této sekci je rovněž dostupná funkce pro vytvoření API tokenu, který lze využít například pro autorizaci při propojení systému openHAB s vývojovým prostředím Visual Studio Code.

4.3.4 Rychlé odkazy – číslo 4

Tato část rozhraní poskytuje uživateli rychlý přístup k nejdůležitějším funkcím systému. Je umístěna přímo na hlavní stránce a slouží k přepínání mezi klíčovými sekcemi, jako jsou přehled systému, správa zařízení nebo jejich seskupení podle místností. Tímto způsobem je umožněna efektivní správa chytré domácnosti bez nutnosti procházení více úrovní nabídky.

1. Přehled – Zobrazuje vybrané prvky systému, které mají být zobrazeny na úvodní přehledové stránce. Obsah této sekce se definuje v nabídce Nastavení → Pages → Overview. Výchozí název „Overview“ může být upraven na libovolný název podle preferencí uživatele.
2. Místa – Umožňuje organizovat a spravovat zařízení podle jejich fyzického umístění v domácnosti. Například lze vytvořit jednotlivé místnosti, do kterých jsou zařízení přiřazena pro jednodušší správu a ovládání.
3. Zařízení – Obsahuje přehled všech připojených zařízení, jako jsou senzory, světla, termostaty a další. V této sekci lze zařízení spravovat, sledovat jejich stav a přímo je ovládat.
4. Vlastnosti – Umožňuje se správa specifických parametrů jednotlivých zařízení. Hodnoty, jako je teplota, jas nebo jiné provozní charakteristiky, mohou být upravovány. Tímto způsobem je možné jemně doladit chování zařízení podle individuálních potřeb.

4.3.5 Logo a stav – číslo 5

Logo systému openHAB je v této části rozhraní zobrazeno jako odkaz pro návrat na hlavní stránku. Po jeho aktivaci je uživatel přesměrován zpět na úvodní přehledové rozhraní systému.

5 CÍL PRÁCE

Popisuje se prostředí rodinného domu, kam byl systém nasazen, a současně byly vymezeny požadavky na jeho funkcionalitu. Mezi klíčové požadavky patřila bezdrátová komunikace, přehledné ovládání, vizualizace dat a odesílání notifikací.

5.1 Popis vybraného prostředí a aplikace

Kapitola je věnována popisu prostředí, ve kterém byl systém nasazen, a stanovení požadavků na aplikaci. Byly specifikovány podmínky realizace v rodinném domě a určeny funkce, které měly být systémem zajištěny, včetně bezdrátové komunikace, vizualizace dat a odesílání notifikací.

5.1.1 Popis prostředí

Systém chytré domácnosti byl nasazen v rodinném domě (Obrázek 10), který byl postaven před více než patnácti lety. Z tohoto důvodu není realizace automatizace prostřednictvím drátových technologií technicky ani ekonomicky proveditelná. Jako hlavní řešení proto byla zvolena bezdrátová komunikace, která umožňuje efektivní integraci jednotlivých zařízení a jejich centralizované řízení bez nutnosti zásahů do stavební struktury objektu.



Obrázek 10 - Ukázka prostředí chytré domácnosti

(zdroj: Generováno pomocí AI nástroje[49])

5.1.2 Požadavky na aplikaci

Zvolený systém openHAB poskytuje vlastní mobilní aplikaci, ve které je možné ovládat jednotlivá zařízení prostřednictvím přehledně uspořádaných uživatelských rozhraní ve formě tzv. stránek (Pages) nebo Sitemaps. Prvky chytré domácnosti mohou být kromě toho ovládány také prostřednictvím webového rozhraní, které navíc umožňuje detailní analýzu a vizualizaci dat, například ve formě grafů zobrazujících teploty v jednotlivých místnostech nebo spotřebu elektrické energie.

Součástí systému je i podpora notifikací, které jsou odesílány v případě změn stavu zařízení nebo při detekci neobvyklých událostí. Díky tomu je uživatel včas informován o důležitých stavech systému. Cílem celého řešení je vytvoření přehledného a intuitivního prostředí, které usnadní efektivní správu domácí automatizace.

6 NÁVRH SYSTÉMU CHYTRÉ DOMÁCNOSTI

Návrh systému chytré domácnosti vychází z připojených zařízení v domácnosti. Postupně jsou popsána zařízení určená k integraci a uvedeny informace o příslušných doplňcích (Bindings), které zajišťují jejich propojení se systémem

6.1 Popis vybraných zařízení pro chytrou domácnost

V této kapitole jsou představena vybraná zařízení chytré domácnosti, která mají být implementována do systému.

6.1.1 HP Color LaserJet MFP M274n

HP Color LaserJet Pro MFP M274n je multifunkční laserová tiskárna určená pro domácí i kancelářské využití. Umožňuje tisk rychlostí až 18 stran za minutu při maximálním rozlišení 600 dpi.[53]

6.1.2 Sinclair SIH – 13BITC

Sinclair SIH-13BIT je klimatizační jednotka s chladicím výkonem 3,5 kW a topným výkonem 3,8 kW. Je vybavena funkcemi, jako je Wi-Fi modul, sedm úrovní rychlosti ventilátoru a možnost pohybu lamel ve vodorovném i svislém směru. Mezi pokročilé funkce patří režim I FEEL pro automatické přizpůsobení teploty a funkce temperování na 8 °C pro udržení minimální teploty.[54]

6.1.3 Sinclair SIH – 09BIK

Sinclair SIH-09BIK je klimatizační jednotka s chladicím výkonem 2,7 kW a topným výkonem 2,8 kW. Zařízení je vybaveno funkcemi, jako je Wi-Fi modul, režim I FEEL pro automatické přizpůsobení teploty a generátor studené plazmy pro zajištění vyšší kvality vzduchu.[55]

6.1.4 Hybridní měnič GoodWe GW10K-ET

Solární měnič GoodWe GW10K-ET představuje třífázový hybridní střídač s maximálním výkonem 10 kW. Zařízení je vybaveno dvěma sledovači maximálního výkonu (MPPT) a umožňuje připojení vysokonapěťových baterií. K dispozici jsou dva samostatné výstupy (první je určen pro napájení domu a druhý pro zálohované spotřebiče), které mají zůstat v provozu i při výpadku distribuční sítě. Měnič podporuje režim bez přetoků, monitorování přes Wi-Fi a dosahuje účinnosti až 98,2 %. [56]

6.1.5 Sony KD-49X8307C

Sony KD-49X8307C je televizor s úhlopříčkou 49 palců a technologií 4K UHD. Vysoká kvalita obrazu je zajištěna podporou vysokého dynamického rozsahu (HDR). Model je vybaven operačním systémem Android TV, který umožňuje přístup k široké škále aplikací a služeb přímo z televizního rozhraní. Připojení k internetu a streamování obsahu je umožněno prostřednictvím integrovaného Wi-Fi modulu. [57]

6.1.6 Sony KD-55AG8

Sony KD-55AG8 je OLED televizor s úhlopříčkou 55 palců a rozlišením 4K. Vynikající kvalita obrazu je zajištěna technologií OLED a podporou vysokého dynamického rozsahu (HDR). Zařízení je vybaveno funkcí Pixel Contrast Booster pro zobrazení živých barev a vysokého kontrastu. Díky operačnímu systému Android TV a integrovanému Wi-Fi připojení je umožněn přístup k široké škále aplikací a online obsahu. [58]

6.1.7 Myčka nádobí Bosch SMI6ZCS00E

Bosch SMI6ZCS00E ze série 6 je vestavná myčka nádobí s nízkou hlučností 44 dB. Díky technologii PerfectDry jsou dosaženy vynikající výsledky sušení i u obtížně schnoucího nádobí. Zařízení je vybaveno funkcí Home Connect, která umožňuje ovládání prostřednictvím Wi-Fi připojení a poskytuje možnost stahování dalších programů. [59]

6.1.8 Sonoff TH16

Wi-Fi modul Sonoff TH16 je určen pro měření teploty a vlhkosti a pro dálkové ovládání externích spotřebičů přes internet. Využívá cloudové služby pro snadné připojení k Wi-Fi síti bez nutnosti složité konfigurace. Ovládání je zajištěno prostřednictvím mobilní aplikace dostupné pro systémy Android a iOS. Napájení je řešeno přímo ze sítě 230 V a aktivní výstup umožňuje spínání spotřebičů do zátěže až 16 A. [60]

6.2 Použité doplňky a jejich specifikace

Tato kapitola se zabývá doplňky pro integraci různých zařízení. Jsou zde specifikovány doplňky, které budou použity k propojení s konkrétními komponentami systému.

6.2.1 HP Printer Binding

Tento doplněk podporuje tiskárny značky HP, které jsou dostupné prostřednictvím IP adresy v místní síti. Pro správné propojení a ovládání tiskárny je každá její funkce (např. stav,

tisková úloha) zpřístupněna prostřednictvím příslušných kanálů. Ty jsou následně propojeny s položkami. [61]

6.2.2 Gree Binding

Tento doplněk podporuje integraci klimatizačních jednotek využívajících protokol GREE (např. značky GREE, Sinclair a další). Pokud je klimatizace připojena k síti a má aktivní Wi-Fi modul, může být automaticky detekována prostřednictvím webového rozhraní systému openHAB. Odesláním broadcastové IP zprávy jsou všechna odpovídající zařízení přidána do sekce Inbox. V případě manuálního přidání je nutné zadat IP adresu zařízení ručně. Pro správné ovládání klimatizace je každý funkční prvek (např. teplota, režim) propojen s příslušným kanálem, který je následně spojen s položkou pro pohodlné ovládání a monitorování prostřednictvím systému openHAB. [62]

6.2.3 ESPHome binding for the Native API

ESPHome je open-source platforma určená pro snadnou integraci zařízení ESP32 a ESP8266 do chytré domácnosti. Umožňuje konfigurovat senzory, spínače a další komponenty bez nutnosti programování v klasickém smyslu. Chování zařízení se definuje prostřednictvím konfiguračního souboru ve formátu YAML, který je následně nahrán do daného mikrokontroléru. [63]

6.2.4 Exec Binding

Doplněk Exec binding umožňuje spouštění externích příkazů terminálu nebo skriptů. Je vhodný zejména pro integraci zařízení či služeb, které nejsou přímo podporovány systémem, nebo pro provádění specifických úloh na úrovni operačního systému v rámci domácí automatizace. [64]

6.2.5 Home Connect Binding

Doplněk Home Connect Binding pro openHAB umožňuje integraci chytrých spotřebičů, které podporují platformu Home Connect. Do této kategorie spadají zařízení, jako jsou myčky, lednice, pračky či trouby od výrobců, jako je Bosch, Siemens a další. Uvedený doplněk umožňuje tato zařízení ovládat a monitorovat přímo v prostředí systému openHAB. Tím je výrazně usnadněna jejich integrace do domácí automatizace. [65]

6.2.6 Sony Binding

Doplněk Sony Binding slouží k integraci zařízení značky Sony, která jsou ovladatelná prostřednictvím IP protokolu. Patří sem například televizory, AV receivery, Blu-ray přehrávače, soundbary a bezdrátové reproduktory. Prostřednictvím tohoto doplňku je umožněno ovládání a monitorování zařízení přímo v systému openHAB. Verze dostupná na komunitním fóru byla upravena pro kompatibilitu s openHAB 4.x a obsahuje drobné opravy. Spárování s televizorem je realizováno prostřednictvím PIN kódu a jsou podporovány funkce, jako je zapnutí či vypnutí zařízení, přepínání vstupů a ovládání hlasitosti. [66]

6.2.7 HTTP Binding

Doplněk HTTP Binding systému openHAB je určen pro komunikaci s libovolnými webovými API prostřednictvím protokolů HTTP a HTTPS. Umožňuje odesílání příkazů (např. GET, POST, PUT, DELETE) a čtení dat ze vzdálených služeb ve formátech, jako jsou JSON, XML nebo prostý text. [67]

7 PŘIDÁNÍ A KONFIGURACE ZAŘÍZENÍ

Zařízení mohou být do systému openHAB přidávána dvěma způsoby – buď prostřednictvím webového rozhraní, nebo pomocí vývojového prostředí Visual Studio Code, které umožňuje přímou práci s konfiguračními soubory.

7.1 Přidání zařízení pomocí webové stránky

Zařízení mohou být do systému openHAB přidávána prostřednictvím uživatelského rozhraní webové stránky. Po otevření webového rozhraní v internetovém prohlížeči se přejde do sekce Things, kde lze vyhledat a přidat nové zařízení. Konfigurace probíhá pomocí grafického průvodce, který vede uživatele krok za krokem. Díky intuitivnímu a snadno dostupnému rozhraní je možné zařízení přidávat rychle, bez nutnosti manuální editace konfiguračních souborů.

7.2 Přidání zařízení pomocí VS Code

Zařízení mohou být do systému přidávána také prostřednictvím vývojového prostředí Visual Studio Code. Tento přístup představuje pokročilejší, ale zároveň flexibilnější způsob konfigurace. Nejdříve je třeba nainstalovat rozšíření openHAB VS Code Extension, které umožňuje přímou práci s konfiguračními soubory. Zařízení jsou následně přidávána ruční editací souborů jako *.things, *.items a *.rules. Tento způsob přináší vyšší míru kontroly nad systémem, avšak vyžaduje základní znalost syntaxe konfiguračních souborů.

Pro úspěšné propojení Visual Studio Code se systémem openHAB je nutné vytvořit API token (Obrázek 11), který slouží k autorizaci komunikace mezi editorem a serverem. Tento token je vygenerován v sekci Uživatelský účet kliknutím na možnost „Vytvořit nový API token“ (Create new API token). Po jeho vygenerování je token vložen do rozšíření ve Visual Studio Code a tím je navázáno zabezpečené spojení s instancí openHAB.

Jako další krok při přidávání zařízení prostřednictvím editoru Visual Studio Code byla provedena konfigurace dříve nainstalovaného rozšíření. V rámci nastavení tohoto rozšíření byl otevřen konfigurační soubor settings.json, do něhož byly vloženy potřebné parametry pro připojení k instanci openHAB běžící na zařízení Raspberry Pi.

```
{  
    "openhab. connection.host": "IP_ADRESA_RPi",  
    "openhab.connection.basicAuth.username": "email@uzivatele.com",  
    "openhab.connection.basicAuth.password": "HesloUzivatele",
```

```
"openhab.connection.authToken": "VytvorenyToken",  
"editor.unicodeHighlight.nonBasicASCII": false,  
}
```



Než budete pokračovat, vytvořte účet správce.

Vytvořit účet

Obrázek 11 - Vytvoření API tokenu

(zdroj: vlastní)

Pro přístup k souborovému systému zařízení Raspberry Pi, na kterém běží instance, byla využita aplikace Finder. V horní nabídce bylo zvoleno „Otevřít“, následně možnost „Připojit k serveru...“, a do textového pole byla zadána IP adresa zařízení ve formátu `smb://<IP_adresa_RPi>`. Po úspěšném připojení bylo možné procházet adresářovou strukturu a byla vybrána složka `openHAB-conf`, ve které se nacházejí veškeré konfigurační soubory systému.

Tato složka je strukturována dle jednotlivých typů konfigurací – obsahuje například podsložky `things`, `items`, `rules`, `sitemaps`, `persistence` a další. Každá z těchto složek je určena pro specifickou oblast správy systému, jako je integrace zařízení, automatizační pravidla nebo vizualizace.

Ve Visual Studio Code byla následně využita možnost „Otevřít složku“ (Open Folder), kde byla otevřena právě složka `openHAB-conf`. Díky tomu bylo možné pohodlně upravovat jednotlivé konfigurační soubory přímo z vývojového prostředí.

7.3 Integrace tiskárny HP

Pro účely chytré domácnosti byla integrována tiskárna značky HP, aby bylo možné sledovat její stav a vybrané provozní parametry prostřednictvím systému `openHAB`.

7.3.1 Vytvoření souboru .things

Ve složce „Things“ byl vytvořen nový soubor s názvem Printer.things, do kterého byl vložen následující kód.

```
Thing    hpprinter:printer:djprinter    "Printer"    @    "Office"    [
  ipAddress="ipAdresa_tiskarny", usageInterval="30", statusInterval="4"
]
```

Tato definice zařízení (Thing) představuje konkrétní zařízení, které je připojeno k systému openHAB. tomto případě se jedná o zařízení typu hpprinter:printer, které reprezentuje tiskárnu značky HP. Identifikátor hpprinter označuje použitý doplněk (Binding) pro komunikaci s tímto zařízením. Název Printer slouží k identifikaci zařízení v systému a Office označuje místnost, ve které je tiskárna umístěna. V hranatých závorkách jsou uvedeny parametry zařízení, jako je IP adresa tiskárny v místní síti, usageInterval, který je nastaven pro sledování využití tiskárny (v tomto případě každých 30 minut), a statusInterval, který sleduje její stav (každé 4 minuty).

7.3.2 Vytvoření souboru .items

V rámci adresáře items byl vytvořen soubor Printer.items, do něhož byly zapsány definice položek určených ke sledování stavových a provozních parametrů tiskárny. Obsahuje deklarace typu String a Number, jejichž prostřednictvím lze získávat informace o aktuálním stavu zařízení, úrovních inkoustu a počtu vtištěných stránek.

```
String          PrinterStatus          "Status"          {
  channel="hpprinter:printer:djprinter:status#status" }
Number:Dimensionless  PrinterBlackLevel  "Black    Level"  {
  channel="hpprinter:printer:djprinter:ink#blackLevel" }
Number:Dimensionless  PrinterCyanLevel   "Cyan     Level"  {
  channel="hpprinter:printer:djprinter:ink#cyanLevel"  }
Number:Dimensionless  PrinterMagentaLevel "Magenta  Level"  {
  channel="hpprinter:printer:djprinter:ink#magentaLevel" }
Number:Dimensionless  PrinterYellowLevel "Yellow   Level"  {
  channel="hpprinter:printer:djprinter:ink#yellowLevel" }

Number          PrinterTotalPages        "Total    Pages"  {
  channel="hpprinter:printer:djprinter:usage#totalCount" }
Number          PrinterTotalColourPages  "Total    Colour  Pages"  {
  channel="hpprinter:printer:djprinter:usage#totalColorCount" }
Number          PrinterTotalMonochromePages "Total    Monochrome  Pages"  {
  channel="hpprinter:printer:djprinter:usage#totalMonochromeCount" }
```

Pomocí těchto položek jsou prostřednictvím kanálů monitorovány klíčové vlastnosti tiskárny. Stav zařízení je reprezentován položkou PrinterStatus. Jednotlivé úrovně inkoustu

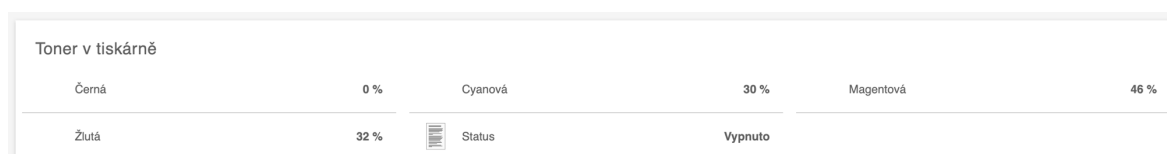
(černá, azurová, purpurová a žlutá) jsou sledovány samostatně. Statistické údaje o provozu tiskárny zahrnují celkový počet výtisků i jejich rozdělení na barevné a černobílé.

7.3.3 Vytvoření sitemapy

Pro zobrazení informací o stavu tiskárny bylo navrženo vlastní uživatelské rozhraní (sitemap), ve kterém byly jednotlivé položky (Items) propojeny s aktuálními stavy zařízení (Obrázek 12). Toto rozhraní bylo vytvořeno prostřednictvím webového editoru, alternativně je však možné jeho vytvoření i ve vývojovém prostředí Visual Studio Code.

```
sitemap Control_Print label="Tiskárna" {
  Frame label="Toner v tiskárně" {
    Text item=PrinterBlackLevel label="Černá [%0f %]"
    Text item=PrinterCyanLevel label="Cyanová [%0f %]"
    Text item=PrinterMagentaLevel label="Magentová [%0f %]"
    Text item=PrinterYellowLevel label="Žlutá [%0f %]"
    Text item=PrinterStatus label="Status [MAP(cz.map):%s]"
  }
}
```

V rámci struktury rozhraní byl definován vizuální blok (Frame), určený pro zobrazení informací o úrovních tonerů tiskárny. Zobrazeny byly hodnoty inkoustu pro jednotlivé barvy (černá, cyanová, magentová, žlutá) a aktuální stav zařízení (Obrázek 12). Pro lokalizaci stavových hodnot byla využita transformace MAP, která umožňuje překlad výstupních textů do českého jazyka.



Obrázek 12 - Sitemapa pro informace o tiskárně

(zdroj: vlastní)

7.3.4 Vytvoření .map transformace

Transformace MAP byla využita pro překlad výstupních stavů tiskárny z anglického jazyka do češtiny. Za tímto účelem byl vytvořen soubor cz.map, který obsahuje potřebná přiřazení hodnot pro lokalizaci. Tento soubor bude dále rozšiřován o další překladové položky podle potřeby systému.

OPEN=Otevřeno

CLOSED=Zavřeno
NULL=Neznámý
shuttingDown=Vypnuto
ready=Připravena
inPowerSave=Režim spánku
processing = zpracování
trayEmptyOrOpen = zásobník prázdný nebo otevřený
Ready=Připraveno

7.4 Integrace teplotního senzoru s ESPHome

Integrace teplotního senzoru byla realizována pomocí mikrokontroléru ESP8266 a platformy ESPHome. Postup zahrnoval vytvoření konfiguračního souboru, nahrání do zařízení a propojení se systémem openHAB prostřednictvím nativního rozhraní.

7.4.1 Seznámení s potřebnými komponenty

Pro realizaci vlastní termostanice v prostředí ESPHome byl zvolen mikrokontrolér ESP8266.

ESP8266 je založený na procesoru L106, obsahuje 32 kB RAM pro instrukce, 80 kB RAM pro data, umožňuje připojit až 16 MB externí sériové FLASH, nabízí rozhraní SPI, analogově-digitální převodník, 17 GPIO pinů a integrované rozhraní pro Wi-Fi. [63]

DS18B20 je teplotní senzor s 1-Wire rozhraním, které umožňuje komunikaci pomocí jediného datového pinu. Měří teploty od -55 °C do +125 °C s přesností $\pm 0,5$ °C v rozsahu od -10 °C do +85 °C. [68]

OLED Displej 0,96" I²C SSD1306 je malý, vysoce kontrastní displej vhodný pro projekty s Arduino nebo jinými embedded systémy. Tento displej je připraven pro napájení 5 V a je vybaven integrovaným regulátorem, který umožňuje použití s deskami napájenými 3,3 V. To jej činí kompatibilním s většinou vývojových desek. Pro snadné připojení používá I²C rozhraní, které vyžaduje pouze dva piny pro komunikaci (SDA a SCL). [69]

7.4.2 Instalace a konfigurace ESPHome

Nejprve bylo nutné nainstalovat nástroj ESPHome do počítače, ve kterém byl připravován konfigurační soubor a následně nahrán do mikrokontroléru. Instalace probíhá zadáním jednoduchého příkazu do terminálu. Pro její úspěšné provedení je nutné mít předem nainstalováno prostředí Pythonu včetně správce balíčků pip.

Python je dostupný ke stažení z oficiálních stránek <https://www.python.org>. Novější verze Pythonu již obsahují pip jako součást instalace, v opačném případě lze správce balíčků doinstalovat pomocí následujících příkazů.

1. Ověření verze Pythonu

```
python --version
python3 --version
```

(podle toho, která verze Pythonu je nainstalována)

2. Ověření verze pip(balíčkový správce pro Python)

```
pip --version
pip3 --version
```

3. Instalace pip, pokud ji nemáme nainstalovanou

```
python get-pip.py
python3 get-pip.py
```

Po instalaci všech závislostí mohl být nainstalován nástroj ESPHome pomocí příkazu:

```
pip install esphome
```

Následně byl inicializován nový projekt. V terminálu se přešlo do adresáře, ve kterém měl být konfigurační soubor uložen, a byl spuštěn průvodce.

```
esphome wizard nazev_prace.yaml
```

Po spuštění příkazu se otevřelo úvodní menu, v jehož rámci byl uživatel vyzván k zadání názvu zařízení, který slouží k jeho identifikaci v systému. Například pro zařízení v obývacím pokoji mohl být zvolen název „livingroom“.

V dalším kroku bylo vyžadováno zvolit platformu, tedy typ mikrokontroléru (např. ESP32, ESP8266, BK72XX, RTL87XX nebo RP2040). Název musel být zadán ve formátu psaném velkými písmeny. Poté byl vybrán konkrétní model vývojové desky, např. „nodemcu“ pro zařízení NodeMCU. Výběr desky musí odpovídat typu použitého hardwaru.

Následovalo nastavení připojení k Wi-Fi síti zadáním SSID a hesla. Během konfigurace byla rovněž nabídnuta možnost nastavení OTA hesla (Over-the-Air), které umožňuje bezdrátovou aktualizaci firmwaru bez nutnosti fyzického propojení zařízení s počítačem.

Výsledná konfigurace mohla vypadat následovně:

```
esphome:
  name: livingroom
esp8266:
```

```
board: nodemcu
# Enable logging
logger:
# Enable Home Assistant API
api:
  password: ""
ota:
  - platform: esphome
    password: ""
wifi:
  ssid: "nazev_wifi"
  password: "psw_wifi"
  # Enable fallback hotspot (captive portal) in case wifi connection
  fails
  ap:
    ssid: "Livingroom Fallback Hotspot"
    password: "tVRDDA7sfDjP"
captive_portal:
```

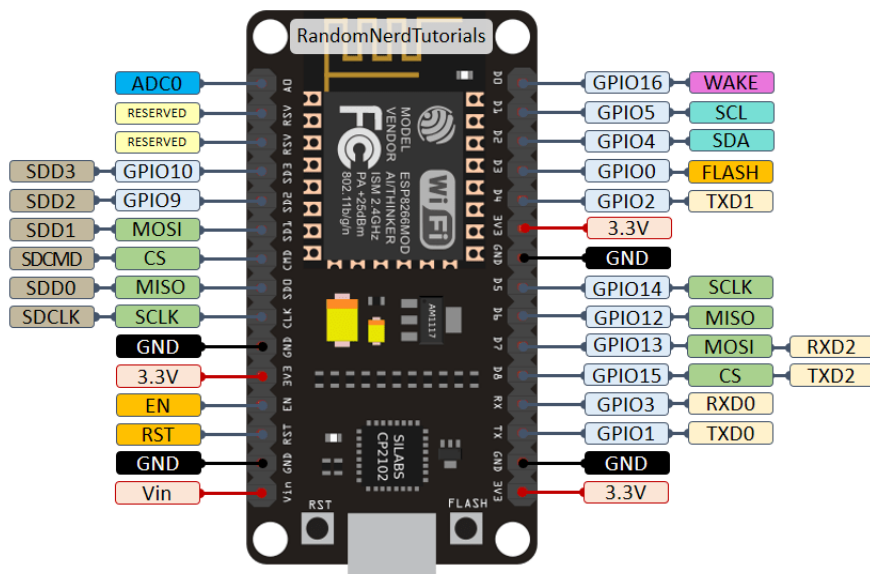
Tento soubor představuje předdefinovanou konfiguraci zařízení ESP8266 (NodeMCU) pro použití v systému ESPHome. Obsahuje nastavení připojení k Wi-Fi, možnosti vzdálené aktualizace (OTA) a také integraci s Home Assistant API. Vzhledem k tomu, že rozhraní Home Assistant API nebude v daném řešení využíváno, může být příslušný řádek ze souboru odstraněn.

7.4.3 Zapojení komponent do mikrokontroléru ESP8266

Následně byl mikrokontrolér ESP8266 připojen k zařízení, ve kterém byl připravován zdrojový kód, a zároveň byly zapojeny jednotlivé komponenty využívané při realizaci projektu.

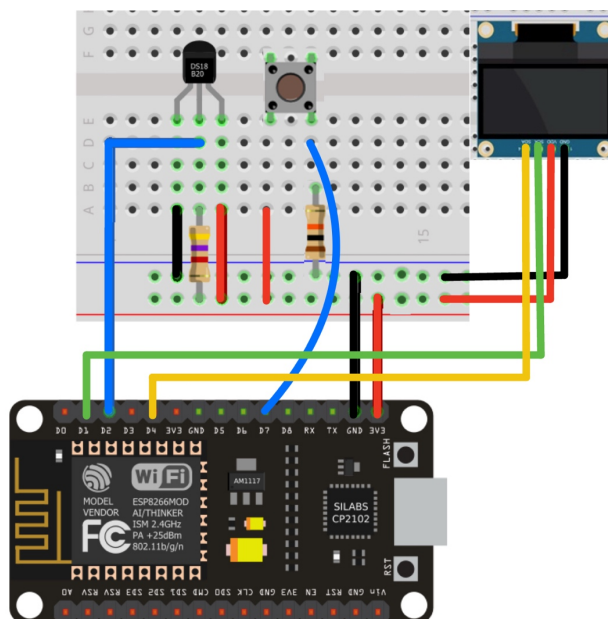
Důležitým krokem při připojování teplotního senzoru a OLED displeje bylo důkladné prostudování pinového rozložení (pinoutu) daného mikrokontroléru (Obrázek 13).

Veškeré komponenty byly zapojeny podle předem navrženého schématu zapojení (Obrázek 14), které znázorňuje propojení mikrokontroléru s jednotlivými prvky teplotní stanice. V rámci tohoto schématu je zobrazen způsob, jakým jsou jednotlivá čidla, zobrazovací jednotka a další periferní zařízení připojena k mikrokontroléru. Díky tomuto přehlednému znázornění bylo možné zajistit správné a funkční propojení všech součástí teplotní stanice.



Obrázek 13 - ESP8266 – Pinout

(zdroj: [70])



Obrázek 14 - Schéma zapojení teplotní stanice

(zdroj: vlastní)

7.4.4 Implementace teplotní stanice

Jakmile bylo vše zapojeno podle schématu, bylo možné přejít k programování zdrojového kódu dle definovaných požadavků. Základní konfigurace již byla popsána, a proto jsou v této části uvedeny pouze doplňující úseky kódu týkající se použitých komponent, jako je teplotní senzor, OLED displej a vlastní definovaný font.

```
font:
  - file: "Roboto-Black.ttf"
    id: font1
    size: 16
```

Tato část definuje font, který je následně použit pro zobrazování textu na OLED displeji s informacemi o teplotě. Parametr file určuje soubor s písmem, v tomto případě jde o font typu TTF – konkrétně Roboto-Black. Tento soubor musí být umístěn ve stejné složce jako projektový soubor .yaml. Identifikátor id: font1 slouží k označení konkrétního fontu v rámci celého projektu a je využíván i v dalších částech kódu pro výstup textu tímto písmem. Parametr size určuje velikost písma, jeho velikost byla zvolena 16.

```
i2c:
  sda: D4
  scl: D1
  scan: false
display:
  - platform: ssd1306_i2c
    model: "SSD1306 128x64"
    address: 0x3C
    id: my_display
    brightness: 1.0
    contrast: 1.0
    update_interval: 1s
    flip_x: true
    flip_y: true
    lambda: |-
      if (id(display_on)) {
        it.printf(0, 0, id(font1), "Teplota: %.1f°C",
id(teplotaobyvak).state);
      }
globals:
  - id: display_on
    type: bool
    restore_value: no
    initial_value: "false"
```

Konfigurace I²C sběrnice slouží k určení připojeného zobrazovacího zařízení. Pin SDA (D4) je využíván pro datový přenos, pin SCL (D1) přenáší synchronizační hodinový signál. Displeji je přiřazena pevná adresa 0x3C. Parametry brightness: 1.0 a contrast: 1.0 určují maximální hodnoty jasu a kontrastu. Parametr update_interval: 1s zajišťuje obnovu obsahu

obrazovky každou sekundu. Dále byly nastaveny parametry `flip_x: true` a `flip_y: true`, které zajišťují zrcadlení obrazu v obou osách.

Lambda funkce slouží k vykreslení aktuální teploty získané ze senzoru teplotaobvyvak. Stavová proměnná `display_on` určuje, zda je displej zapnutý. Výchozí hodnota `false` znamená, že displej je po spuštění systému vypnutý. Tato konfigurace umožňuje flexibilní zobrazování teploty s možností řízení chování displeje dle potřeb.

```
binary_sensor:
  - platform: gpio
    pin:
      number: D7
      mode: INPUT_PULLUP
    name: "Rozsvítit displej"
    on_press:
      - lambda: |-
          id(display_on) = true;
          id(my_display).set_brightness(1.0);
      - script.execute: display_off_script

script:
  - id: display_off_script
    mode: restart
    then:
      - delay: 10s
      - lambda: |-
          id(display_on) = false;
          id(my_display).set_brightness(0.0);
```

Součástí konfigurace je také definice tlačítka připojeného na pin D7, které pracuje v režimu `INPUT_PULLUP`. Tento režim znamená, že pin má výchozí logickou hodnotu `HIGH` a po stisknutí tlačítka přejde do stavu `LOW`. Po aktivaci tlačítka je proměnná `display_on` nastavena na hodnotu `true`, displej je zapnut a jas nastaven na maximální úroveň. Zároveň je spuštěn skript `display_off_script`, který po uplynutí 10 sekund displej automaticky vypne a opět sníží jas na minimum.

```
one_wire:
  platform: gpio
  pin: D2
```

```
sensor:  
  - platform: dallas_temp  
    name: "Teplota obyvak"  
    id: teplotaobyvak  
    update_interval: 30s
```

Teplotní senzor Dallas DS18B20 byl připojen na pin D2 s využitím protokolu One – Wire. Senzor je definován pod názvem „Teplota obyvak“ s identifikátorem teplotaobyvak. Hodnota teploty je aktualizována každých 30 sekund, jak určuje parametr update_interval.

Po dokončení přípravy konfiguračního souboru bylo možné připojit mikrokontrolér k počítači prostřednictvím USB portu a zahájit proces nahrávání kódu.

```
Esphome run nazev_prace.yaml run
```

Po spuštění nahrávání byla detekována více než jedna možnost připojení k zařízení. Byly nabídnuty dvě varianty – nahrání přes sériový port (fyzické připojení přes USB) nebo bezdrátové nahrání pomocí OTA (Over-the-Air) přes síťové připojení. Pro účely tohoto projektu byla zvolena první možnost, tedy fyzické připojení přes USB. Po potvrzení volby došlo ke spuštění samotného nahrávání programu do zařízení.

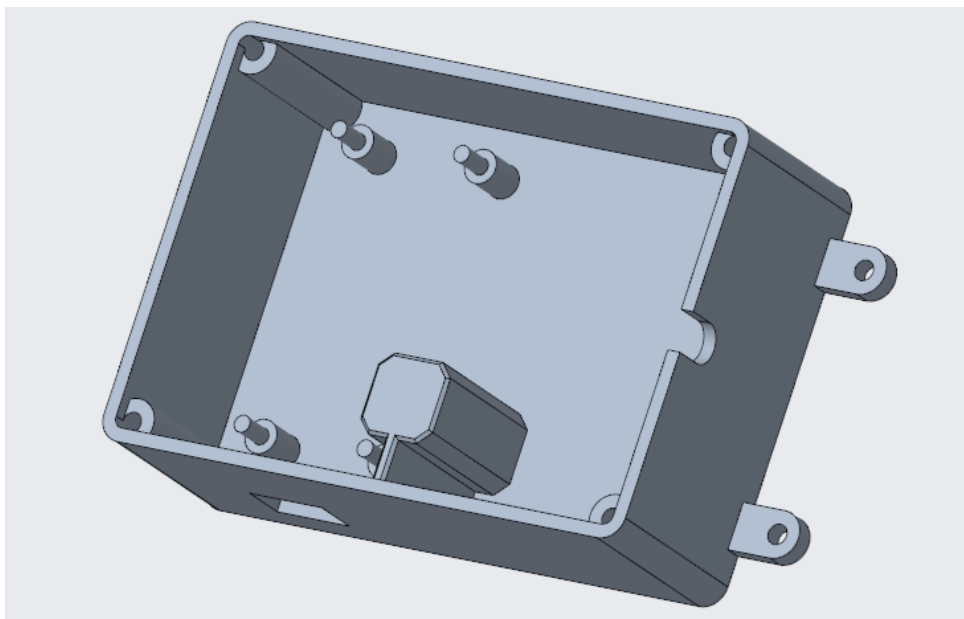
Po úspěšném nahrání programu byla naměřená teplota pravidelně vypisovat do terminálu. Následně bylo možné přistoupit k integraci dat do systému openHAB.

7.4.5 Vytvoření ochranné krabičky

Pro teplotní stanici byla navržena a zhotovena ochranná krabička (Obrázek 15 a 16), do které budou umístěny veškeré součástky. Krabička slouží k ochraně elektronických komponent před prachem a mechanickým poškozením, a zároveň plní estetickou funkci zakrytí všech prvků. Vizuální podoba zařízení byla přizpůsobena pro montáž na stěnu v interiéru.

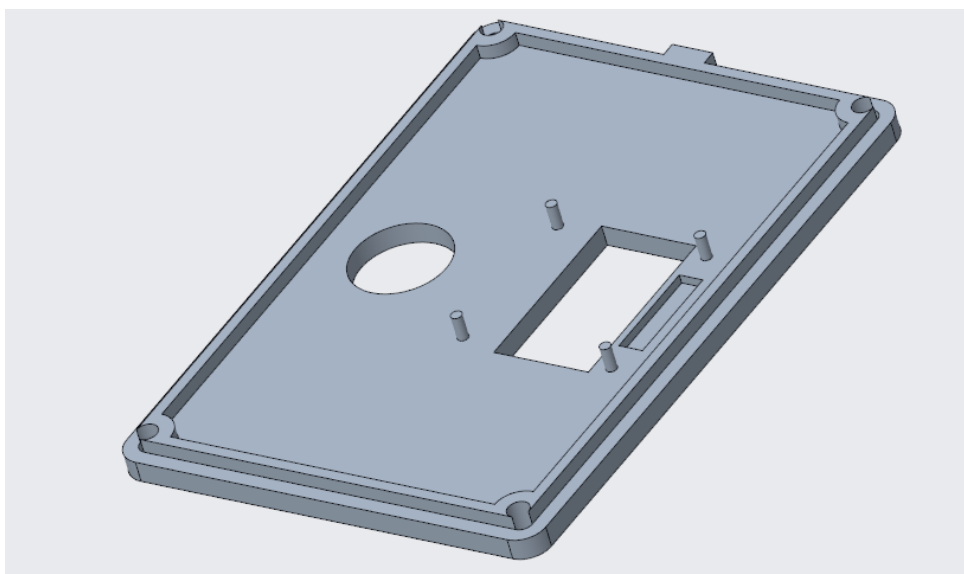
Design krabičky byl navržen s ohledem na potřebu dostatečného prostoru pro umístění všech komponent, jako jsou mikrokontrolér ESP8266, OLED displej, tlačítko a teplotní senzor DS18B20. Dále byl integrován výřez pro přívod napájení a pro pohodlné ovládání tlačítka.

Celá krabička byla připravena k výrobě pomocí 3D tisku. Po vytištění byla konstrukce připevněna na stěnu, aby bylo zajištěno pohodlné ovládání a sledování teploty prostřednictvím displeje.



Obrázek 15 - 3D model spodní části krabičky pro teplotní stanici

(zdroj: vlastní)



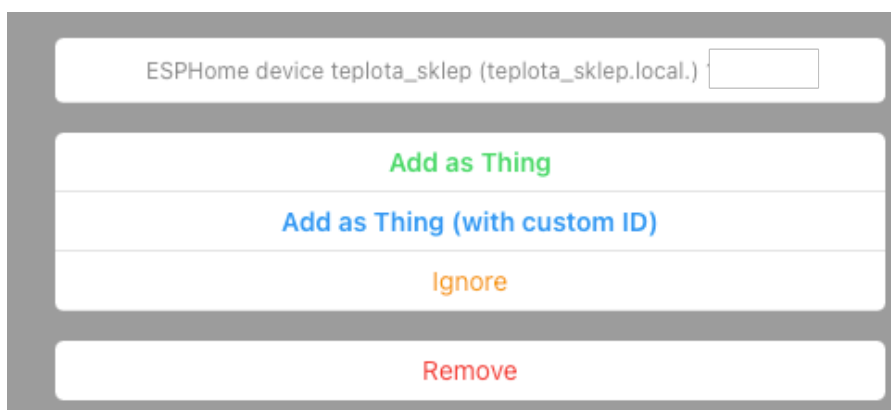
Obrázek 16 - 3D model vrchní části krabičky pro teplotní stanici

(zdroj: vlastní)

7.4.6 Přidání teplotního senzoru

Pro integraci teplotního senzoru do systému byl zvolen doplněk (binding) s názvem ESPHome binding for the Native API. Tento doplněk umožňuje přímou komunikaci mezi mikrokontrolérem a systémem openHAB bez nutnosti zprostředkovatelů, jako je například MQTT.

Nejprve bylo v uživatelském rozhraní otevřeno menu Inbox, kde byl zobrazen název detekovaného zařízení společně s jeho IP adresou v lokální síti. Po kliknutí na zařízení a výběru možnosti „Add as Thing“ bylo zařízení úspěšně přidáno do systému jako nový objekt typu Thing (Obrázek 17). Následně bylo možné přistoupit k definování vazeb mezi jeho kanály a položkami.



Obrázek 17 - Přidání zařízení ESPHome do systému openHAB

(zdroj: vlastní)

V sekci Kanály (Channels) bylo vytvořeno propojení mezi kanálem teplotního senzoru a novou položkou. Tento proces zahrnoval kliknutí na kanál s označením „Teplota sklep“, výběr možnosti „Add Link to Item...“ a následné vytvoření nové položky přes „Create a new Item“. Po zadání názvu nové položky a jejím propojení pomocí tlačítka „Link“ bylo propojení dokončeno (Obrázek 18).

Teplota sklep esphome:device:REMOVEWHENADDINGteplota_sklep:teplota_sklep (Number:Temperature)	
Item	
☐ Use an existing Item ☒ Create a new Item	
Name	ESPHome_device_teplota_sklep_teplota_skleplocal_100113_Teplota_sklep
Label	Teplota sklep
Type	Number
Dimension	Temperature (°C)
Unit	°C
State Description Pattern	%1f °C
Category	temperature
Semantic Class	Measurement
Semantic Property	Temperature

Obrázek 18 - Přidání položky teplota sklep

(zdroj: vlastní)

Tímto způsobem byla vytvořena položka (Item) pro načítání a zobrazování hodnot teploty v uživatelském rozhraní systému openHAB.

7.4.7 Vytvoření sitemapy

K teplotní stanici byla kromě zobrazování aktuální hodnoty teploty přidána také možnost jejího grafického zobrazení ve formě časového průběhu (Obrázek 19). Před samotným vykreslením grafu však bylo nutné nejprve přidat příslušnou položku (Item) do souboru pro ukládání dat. Tento postup je podrobně popsán v samostatné kapitole věnované persistenci dat.

```
Frame label="Teplotní stanice" {  
    Text icon=temperature item=livingroomteplota_Teplota_obyvak  
    label="Teplota[%.1f °C]" {  
        Chart item=livingroomteplota_Teplota_obyvak label="Graf"  
        period=P1D refresh=600  
    }  
}
```



Obrázek 19 - Sitemapa pro zobrazování teploty

(zdroj: vlastní)

7.5 Integrace klimatizací v domě

V této kapitole je popsán proces připojení klimatizačních jednotek do systému. Integrace byla rozdělena do několika kroků zahrnujících definici zařízení, vytvoření položek (Items) a návrh uživatelského rozhraní ve formě sitemap.

7.5.1 Vytvoření souboru .things

V adresáři Things byl vytvořen konfigurační soubor KlimaP1.things, do kterého byla zapsána následující definice.

```
Thing gree:airconditioner:KlimaP1 [ ipAddress="ip_adresa", refresh=2 ]
```

Tento zápis definuje zařízení typu gree:airconditioner, které reprezentuje klimatizaci značky GREE. Doplněk (Binding) Gree zajišťuje komunikaci se zařízením prostřednictvím IP adresy v lokální síti. Identifikátor KlimaLibor slouží k odlišení této jednotky od ostatních. Parametr refresh=2 určuje interval aktualizace hodnot v intervalu dvou sekund.

Doplňěk (Binding) Gree je primárně určen pro klimatizace značky GREE, avšak podporuje také vybrané modely jiných výrobců, například Sinclair, pokud využívají kompatibilní komunikační protokol. Tento scénář odpovídá i implementaci popsané v této práci.

7.5.2 Vytvoření souboru .items

V adresáři items byl následně vytvořen soubor KlimaPl.items, který obsahuje definice položek propojených s jednotlivými kanály zařízení.

```
Switch Air_Conditioner_Napajeni1 {
    channel="gree:airconditioner: KlimaPl:powerchannel" }

Number Air_Conditioner_Unit_Model {
    channel="gree:airconditioner: KlimaPl:modechannel" }

Switch Air_Conditioner_Turbo_Model {
    channel="gree:airconditioner: KlimaPl:turbochannel" }

Number Air_Conditioner_Target_Temperature1 "Temperature [%1f °C]"
{channel="gree:airconditioner: KlimaPl:tempchannel" }

Number Air_Conditioner_Wind_Speed1 {
    channel="gree:airconditioner: KlimaPl:windspeedchannel" }

Number Air_Conditioner_Current_Temperature1 {
    channel="gree:airconditioner:KlimaPl:currentTemperature
    (Number:Temperature)" }
```

Tento zápis slouží k definování položek, které zprostředkovávají ovládání a sledování jednotlivých parametrů klimatizace. Vytvořeny byly položky pro řízení napájení, provozního režimu, aktivace turbo režimu, cílové teploty, rychlosti ventilátoru a aktuální teploty v místnosti. Každá z těchto položek je propojena s příslušným kanálem.

7.5.3 Vytvoření sitemapy

Pro vizualizaci stavu a ovládání klimatizace bylo vytvořeno vlastní uživatelské rozhraní (sitemap) s názvem Control_AirConditioner (Obrázek 20). V něm byl definován rámec (Frame), který slouží k zobrazení a ovládání parametrů klimatizace KlimaLibor.

```
sitemap Control_AirConditioner label="Klimatizace" {
    Frame label="Klimatizace - Libor" {
        Switch icon=switch item=Air_Conditioner_Napajeni1
        label="Napájení"

        Switch icon=energy item=Air_Conditioner_Unit_Model label="Mód
        klimatizace" mappings=[1="Chlazení", 4="Topení"]

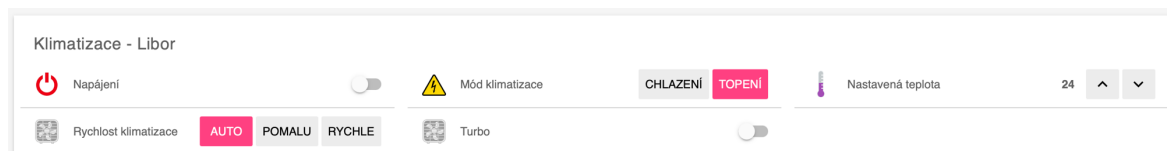
        Setpoint icon=temperature
        item=Air_Conditioner_Target_Temperature1 label="Nastavená teplota"
        maxValue=30 minValue=16 step=1
```

```

        Switch icon=fan item=Air_Conditioner_Wind_Speed1
label="Rychlost klimatizace" mappings=[0="Auto", 1="Pomalou",
5="Rychle"]

        Switch icon=fan item=Air_Conditioner_Turbo_Model label="Turbo"
    }
}

```



Obrázek 20 - Sitemapa pro ovládání klimatizace

(zdroj: vlastní)

Rámec (Frame) v uživatelském rozhraní obsahuje prvky pro řízení napájení, výběr provozního režimu (např. chlazení nebo topení), nastavení cílové teploty, regulaci rychlosti ventilátoru a aktivaci turbo režimu. U některých položek, jako je režim klimatizace nebo rychlost ventilátoru, byly použity možnosti výběru definované pomocí mapování (mappings). Díky tomu bylo dosaženo přehledného a uživatelsky přívětivého rozhraní. Pro nastavení teploty byl využit prvek Setpoint, který umožňuje změnu hodnoty v definovaném rozsahu s pevně daným krokem.

Další klimatizace v domě byly integrovány obdobným způsobem. Změněny byly pouze názvy položek, aby nedocházelo ke konfliktům způsobeným duplicitními identifikátory.

7.6 Implementace a konfigurace měniče Goodwe

Vzhledem k tomu, že doplněk (Binding) poskytovaný platformou SEMSPortal nepodporuje dostatek informací potřebných pro domácí využití, bylo nutné navrhnout alternativní způsob získávání dat z měniče. Vhodným řešením je využití jazyka Python a knihovny Goodwe, která umožňuje číst data přímo přes místní síť.

7.6.1 Připojení pomocí SSH k Raspberry PI

Pro vzdálený přístup k zařízení Raspberry Pi byl využit protokol SSH. Připojení bylo zahájeno otevřením terminálu a zadáním následujícího příkazu:

```
ssh openhabian@ip_adresa
```

Po odeslání příkazu byl uživatel vyzván k zadání hesla. Po úspěšném přihlášení bylo možné provádět další konfiguraci systému a přípravu prostředí pro komunikaci s měničem.

7.6.2 Instalace potřebných nástrojů

V první fázi bylo nutné ověřit, že je v systému nainstalován doplněk (Binding) `exec` a transformace `JSONPath`. Oba komponenty lze získat prostřednictvím Obchodu doplňků v kategorii „Transformations“.

Následně bylo ověřeno, zda je v systému dostupný správce balíčků `pip3` pro Python 3. Pokud nebyl nainstalován, bylo použito následujícího příkazu:

```
sudo apt install python3-pip
```

Pro komunikaci s měničem značky GoodWe byla následně nainstalována knihovna `goodwe`:

```
sudo pip3 install goodwe --break-system-packages
```

Po úspěšném nastavení prostředí bylo možné pokračovat dvěma způsoby: buď přímo přes terminál pomocí SSH připojení, nebo prostřednictvím vývojového prostředí Visual Studio Code, které bylo připojeno k zařízení Raspberry Pi. Oba přístupy jsou z hlediska implementace téměř identické, a proto budou popsány společně.

7.6.3 Vytvoření Python skriptu pro získání dat z měniče

V SSH terminálu se nejprve přejde do složky `scripts` pomocí následujícího příkazu:

```
cd /etc/openhab/scripts
```

Následně je vytvořen soubor s názvem `goodwejson.py` příkazem:

```
sudo nano goodwejson.py
```

Alternativně bylo možné vytvořit stejný soubor také ve vývojovém prostředí Visual Studio Code. Do souboru byl vložen následující kód, ve kterém je nutné nahradit řetězec `ip_adresa_menice` skutečnou IP adresou zařízení.

```
import asyncio
import goodwe

async def get_runtime_data():
    ip_address = 'ip_adresa_menice'
    sensors = [
        "ppv",                # Výkon fotovoltaiky (W)
        "pbattery1",          # Výkon baterie (W) - = nabíjení, + = vybíjení
        "battery_mode",       # 1 = pohotovostní režim, 2 = vybíjení, 3 = nabíjení
        "battery_soc",        # Stav nabití baterie (%)
        "active_power",       # Výkon do/ze sítě (W): - = nákup, + = prodej
```

```

    "grid_in_out",          # 1 = prodej (export do sítě), 2 = nákup
                             (import ze sítě) 0 = Idle (žádný tok)
    "house_consumption", # Spotřeba domácnosti (W)
    "e_day",              # Dnešní výroba FV elektrárny (kWh)
    "e_total",            # Celková výroba FV elektrárny (kWh)
    "meter_e_total_exp", # Celková prodaná (exportovaná) energie (kWh)
    "meter_e_total_imp" # Celková nakoupená (importovaná) energie (kWh)
]

inverter = await goodwe.connect(ip_address)
runtime_data = await inverter.read_runtime_data()
print(f'{{{')

for sensor in inverter.sensors():
    if sensor.id_ in runtime_data:
        if sensor.id_ in sensors:
            print(f'"{sensor.id_}": {runtime_data[sensor.id_]},')
print(f'"end": 0')
print(f'}}}')

asyncio.run(get_runtime_data())

```

Po uložení souboru byl skript připraven ke spuštění. Funkčnost byla ověřena zadáním následujícího příkazu v terminálu.

```
python3 goodwejson.py
```

Po úspěšném spuštění byly vypsané aktuální hodnoty senzorů načtené z měniče prostřednictvím místní sítě (Obrázek 21).

```

openhavian@openhavian:/etc/openhab/scripts $ python3 goodwejson.py
{
  "ppv": 3122,
  "active_power": -72,
  "grid_in_out": 0,
  "pbattery1": 1861,
  "battery_mode": 2,
  "e_total": 32314.7,
  "e_day": 14.7,
  "house_consumption": 5055,
  "meter_e_total_exp": 7251.294,
  "meter_e_total_imp": 13060.618,
  "battery_soc": 76,
  "end": 0
}

```

Obrázek 21 - Výpis informací z měniče

(zdroj: vlastní)

7.6.4 Vytvoření bash skriptu pro spuštění Python skriptu

Ve stejné složce (/etc/openhab/scripts) byl vytvořen nový soubor s názvem goodwe.sh pomocí příkazu.

```
sudo nano goodwe.sh
```

Obsah tohoto souboru je následující:

```
#!/bin/bash
```

```
RESULT=$(python3 /etc/openhab/scripts/goodwejson.py)
```

```
echo "$RESULT" | tr '\n' ' '
```

Následně bylo potřeba upravit oprávnění k souboru, aby mohl být skript spouštěn. Tato operace byla provedena pomocí následujícího příkazu (nutné provést přes SSH):

```
sudo chmod 744 goodwe.sh
```

Dále bylo nutné přidat příkaz pro spuštění skriptu do souboru exec.whitelist ve složce /etc/openhab/misc. Tímto zápisem je systému povoleno spustit skript pomocí rozhraní doplňku (Binding) exec.

```
bash /etc/openhab/scripts/goodwe.sh
```

7.6.5 Vytvoření souboru .things

Byl vytvořen konfigurační soubor exec.things, který definuje příkaz ke spuštění skriptu a nastavuje frekvenci jeho provádění. Podle doporučení uživatelů na fórech openHAB by interval neměl být kratší než 30 sekund, aby nedocházelo k zahlcení komunikace s portálem SEMS.

```
sudo nano /etc/openhab/things/exec.things
```

Obsah definice:

```
exec:command:goodwe_json[command="bash/etc/openhab/scripts/goodwe.sh,
interval=30,timeout=10,autorun=false]
```

7.6.6 Vytvoření souboru .items

V rámci systému byly vytvořeny položky (Items), které budou přijímat a zobrazovat hodnoty z výstupu JSON dat načítaných ze zařízení. Každá položka odpovídá konkrétní hodnotě reprezentující stav systému.

```
String ZP_JSON_Out "[%s]" {channel="exec:command:goodwe_json:output"}
Number:Power      zpPv      "zpPv [%0f W]"      <energy>
```

Number:Power	zpBattery	"zpBattery [%.0f W]"	<energy>
Number	zpBatteryStatus	"zpBatteryStatus [%d]"	
Number	zpSoc	"zpSoc [%d %%]"	<battery>
Number:Power	zpActivePower	"zpActivePower [%.0f W]"	<energy>
Number	zpGridStatus	"zpGridStatus [%d]"	
Number:Power	zpConsumption	"zpConsumption [%.0f W]"	<energy>
Number:Energy	zpEday	"zpEday [%.1f kWh]"	<energy>
Number:Energy	zpEtotal	"zpEtotal [%.0f kWh]"	<energy>
Number:Energy	zpEtotalExp	"zpEtotalExp [%.0f kWh]"	<energy>
Number:Energy	zpEtotalImp	"zpEtotalImp [%.0f kWh]"	<energy>

7.6.7 Vytvoření souboru .rules

V souboru goode.rules (Příloha č.1) byla definována pravidla, která zajišťují zpracování přijatých JSON dat a jejich aktualizaci v jednotlivých položkách.

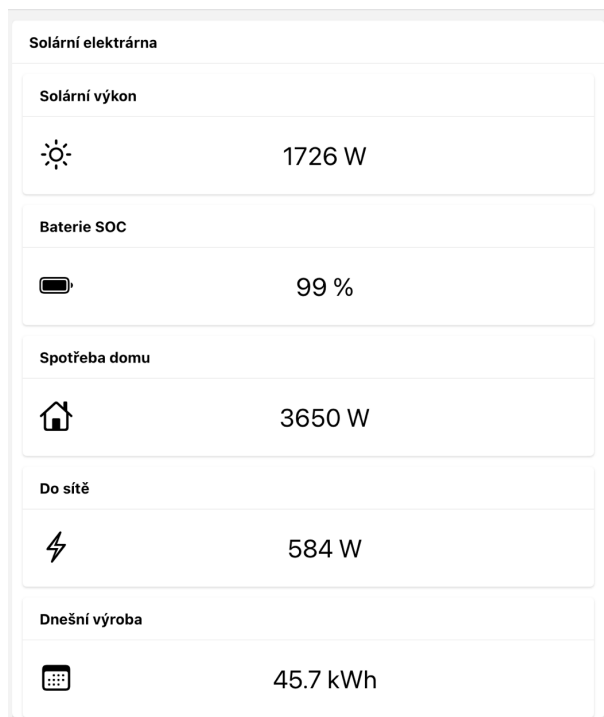
Toto pravidlo je navrženo tak, aby při každé změně položky ZP_JSON_Out došlo k transformaci přijatých dat pomocí JSONPATH. Vybrané hodnoty, jako například výkon fotovoltaických panelů, stav baterie, SOC, spotřeba nebo export/import energie, jsou extrahovány a aktualizovány v odpovídajících položkách pomocí metody postUpdate. Tím je zajištěna automatická aktualizace hodnot v reálném čase bez nutnosti zásahu uživatele.

Dále bylo vytvořeno dodatečné pravidlo pro interpretaci některých číselných stavů jako například identifikaci, zda se baterie nabíjí nebo vybíjí (Příloha č.3), či zda domácnost odebírá nebo dodává elektřinu do sítě. Tyto stavy jsou v JSON výstupu reprezentovány číselnými hodnotami, které je třeba převést na srozumitelnou podobu.

Pravidlo pro zobrazení informací o odběru nebo dodávce energie bylo vytvořeno stejným způsobem, jaký je uveden v příloze č. 3. Liší se pouze použitými položkami (Items) a textovými popisky, které slouží pro výpis do uživatelského rozhraní.

7.6.8 Vytvořené widgetu pro informace

Pro rychlý přehled o stavu solární elektrárny byl vytvořen vlastní widget (Příloha č. 2), který zobrazuje klíčové informace (Obrázek 22). Widget slouží jako základní vizualizace systémových. Jeho účelem je poskytnout uživateli okamžitý přístup ke stavu zařízení bez nutnosti otevírat detailní pohledy.



Obrázek 22 - Widget solární elektrárna

(zdroj: vlastní)

7.6.9 Vytvoření sitemapy

Tato uživatelská rozhraní (sitemapy) byla navržena za účelem zajištění přehledného a intuitivního zobrazení informací o fotovoltaickém systému (Obrázek 23, 24 a 25). Uživatelům je tímto způsobem umožněno sledovat klíčové provozní parametry, jako je aktuální výkon fotovoltaických panelů, kapacita baterie a tok elektrické energie do nebo ze sítě.

Zobrazovány jsou rovněž doplňující informace, například výroba za aktuální den, celková vyrobená energie nebo výkon baterie.

U parametrů týkajících se baterie je aplikován znaménkový systém, kde kladná hodnota označuje vybíjení a záporná hodnota nabíjení. Pro zvýšení srozumitelnosti byl k číselným hodnotám přidán i textový popis aktuálního stavu. Stejný princip je uplatněn také u informace o směru toku energie mezi domácnostmi a distribuční sítí – záporné hodnoty označují odběr energie, kladné představují dodávku do sítě.

```
Frame label="Solární panely" {
    Text item=zpPv label="Výkon panelů [%0f W]"
    Text item=zpEday label="Dnešní výroba [%1f kWh]"
    Text item=zpEttotal label="Celková výroba [%0f kWh]"
}
```

```

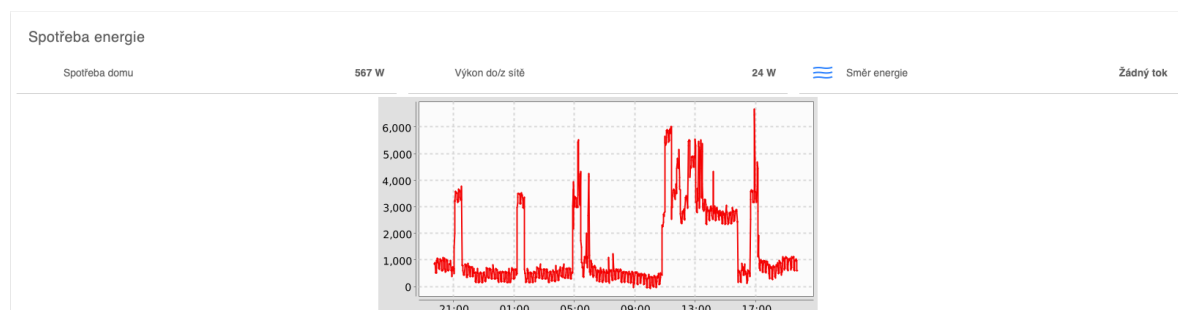
        Chart item=zpPv label="Graf výkonu panelů" period=D refresh=600
    }
    Frame label="Spotřeba energie" {
        Text icon=power item=zpConsumption label="Spotřeba domu [%0f
W]"
        Text icon=power item=zpActivePower label="Výkon do/z sítě [%0f
W]"
        Text icon=flow item= zpGridStatusText label="Směr energie [%s]"
        Chart item=zpConsumption label="Graf spotřeby domu" period=D
refresh=600
    }
    Frame label="Baterie" {
        Text item=zpBattery label="Výkon baterie [%0f W]"
        Text item=zpSoc label="Stav baterie [%1f %]"
        Text icon=flow item=zpBatteryStatusText label="Režim baterie"
        Chart item=zpBattery label="Graf výkonu baterie" period=D
refresh=600
    }

```



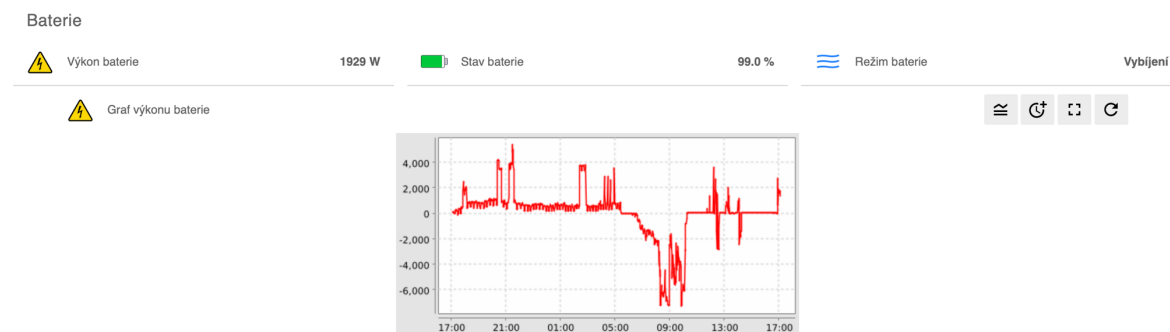
Obrázek 23 - Sitemapa pro Solární panely

(zdroj: vlastní)



Obrázek 24 - Sitemapa pro spotřebu energie

(zdroj: vlastní)



Obrázek 25 - Sitemapa pro baterii

(zdroj: vlastní)

7.7 Implementace ovládání myčky

V této kapitole je popsán proces připojení myčky do systému prostřednictvím doplňku (Binding) Home Connect. Integrace byla provedena přes webové rozhraní systému, kde bylo využito tlačítko „+“ pro přidání nového zařízení a následně byl vybrán příslušný doplněk (Home Connect Binding).

7.7.1 Home Connect API – účet, vývojářský účet, aplikace

Pro úspěšném nainstalování doplňku Home Connect bylo nejprve nutné vytvořit komunikační most (Bridge), který zajišťuje výměnu dat mezi oběma systémy.

1) Vytvoření účtu Home Connect

V první fázi bylo nutné vytvořit běžný uživatelský účet na portálu Home Connect. Tento účet bylo možné založit buď prostřednictvím oficiálních webových stránek, nebo pomocí mobilní aplikace.

2) Založení vývojářského účtu

Dalším krokem bylo vytvoření vývojářského účtu v rámci platformy Home Connect Developer Program (Obrázek 26). Po registraci byl uživatel vyzván k zadání základních údajů, jako jsou jméno, příjmení, uživatelské jméno a e-mailová adresa. Je nezbytné, aby zde byl použit stejný e-mail, který je spojen s dříve vytvořeným účtem Home Connect, jinak nebude autorizace úspěšná. Ostatní údaje, jako typ účtu nebo země, jsou volitelné a závisí na preferencích uživatele.

The image shows a web form for creating a Home Connect developer account. It contains several input fields and dropdown menus, each with a label and a required asterisk. The fields are: First Name*, Last Name*, Username*, Email*, Default Home Connect Account for Testing, Country*, and Account Type*. There are also two informational icons with text: 'Spaces and punctuation is not allowed except for periods, hyphens, apostrophes, and underscores.' and 'The email address is not made public. It will only be used if you need to be contacted about your account or for opted-in notifications.' and 'The specified Home Connect user account can be used with your applications during the development phase.'

First Name*

Last Name*

Username*

Spaces and punctuation is not allowed except for periods, hyphens, apostrophes, and underscores.

Email*

The email address is not made public. It will only be used if you need to be contacted about your account or for opted-in notifications.

Default Home Connect Account for Testing

The specified Home Connect user account can be used with your applications during the development phase.

Country*

- None -

Account Type*

- None -

Obrázek 26 - Vytvoření Home Connect developer account

(zdroj:[71])

3) Registrace aplikace v sekci „Applications“

Po úspěšném přihlášení do vývojářského portálu bylo přistoupeno k vytvoření nové aplikace v sekci Applications (Obrázek 27). Tato aplikace funguje jako rozhraní, které propojuje openHAB s Home Connect API a umožňuje přístup k datům a ovládání kompatibilních spotřebičů.

- Název aplikace (Application ID) by měl být zvolen tak, aby jednoznačně vystihoval její účel.
- V části OAuth Flow byla zvolena možnost Authorization Code Grant Flow.
- Do pole Home Connect User Account for Testing byla zadána e-mailová adresa propojená s uživatelským účtem Home Connect.
- Nakonec byla zadána přesměrovací URL adresa:

`http://openhabian:8080/homeconnect`

Po uložení konfigurace byla aplikace vytvořena.

Po dokončení registrace byly automaticky vygenerovány Client ID a Client Secret, které budou využity v dalších krocích při konfiguraci připojení v systému openHAB.

Application ID *

The application ID is the internal name of the application.

OAuth Flow *

- Select -

Please select the OAuth Flow you want to use with your application. This can not be changed afterwards.

Home Connect User Account for Testing

The specified Home Connect user account can be used with this application during the development phase. Leave blank to use the Home Connect user account defined in [your profile](#).

Redirect URI *

Please provide a valid redirect URI for the OAuth process, e.g. <https://example.com>.

If you want to use localhost or IP as redirect URI, please resubmit the form after passing the security challenge

☐ Add additional redirect URIs

One Time Token

☐ Enable One Time Token Mode

A refresh token can be only used once. In case of a second usage, all created access and refresh tokens are revoked

☐ Sync to China

If enabled, application data will be synced to the Chinese server. This is needed so that you can use your application in China.

Obrázek 27 - Vytvoření aplikace pro připojení k systému openHAB

(zdroj:[71])

7.7.2 Home Connect API – Integrace

V sekci zařízení (Things) bylo zahájeno přidání nového zařízení kliknutím na ikonu „+“ a výběrem doplňku (Binding) Home Connect. Následně byla z nabídky zvolena možnost Home Connect API (Bridge) v rámci, které byly zadány dříve získané přihlašovací údaje: Client ID a Client Secret (Obrázek 28).

Thing ID	HCAPI
	Note: cannot be changed after the creation
Thing UID	homeconnect:api_bridge:HCAPI
Label	Home Connect API
Location	e.g. Kitchen

Home Connect API

This bridge represents the gateway to the Home Connect API.

Show advanced ☐

Client Id
your_clientID

Required Application client id

Client Secret
.....

Required Application client secret

Create Thing

Obrázek 28 - Konfigurace mostu (bridge) pro Home Connect API

(zdroj: vlastní)

Po úspěšném vytvoření mostu bylo přistoupeno k autorizaci prostřednictvím přechodu na adresu:

`http://IP_Adresa_RPI:8080/homeconnect`

Na této stránce bylo třeba kliknout na tlačítko Authorize Bridge. Následně byl uživatel přesměrován na přihlašovací stránku účtu Home Connect, kde proběhlo ověření identity. Po úspěšném přihlášení byla aplikaci udělena oprávnění k ovládání zařízení pomocí tlačítka Approve.

Po dokončení autorizace byla na stránce zobrazena zpráva potvrzující, že most (Bridge) byl úspěšně autorizován a připraven pro další použití v systému openHAB.

7.7.3 Integrace myčky na nádobí

Po úspěšném nastavení a autorizaci komunikačního mostu (Bridge) byl v rámci doplňku (Binding) Home Connect vybrán typ zařízení Dishwasher (myčka nádobí).

Thing ID	a6b9a6b9e9 Note: cannot be changed after the creation
Thing UID	homeconnect:dishwasher:a6b9a6b9e9
Label	Dishwasher
Location	e.g. Kitchen

Parent Bridge
This type of Thing needs to be associated to a working Bridge to function properly.

Bridge

Dishwasher
Home Connect connected dishwasher (e.g. Bosch or Siemens).

Required Unique identifier representing a specific home appliance.

[Create Thing](#)

Obrázek 29 - Konfigurace myčky nádobí

(zdroj: vlastní)

V rámci konfigurace zařízení (Obrázek 29) byl zadán jeho název a informace o jeho umístění. V sekci Bridge byl následně zvolen dříve nakonfigurovaný most, který zajišťuje spojení s Home Connect API.

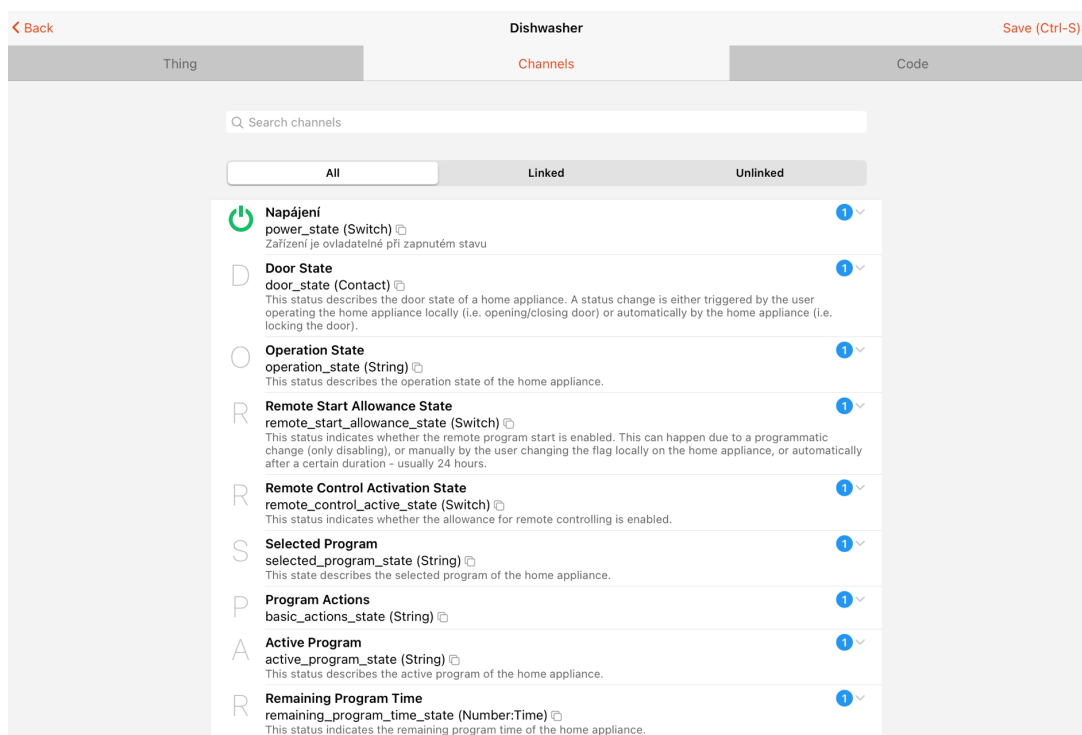
Pro zadání identifikátoru zařízení (haId) existuje několik možností. Nejpřímějším způsobem je využití oficiální mobilní aplikace Home Connect, kde je haId reprezentováno

sériovým číslem zařízení. Alternativně je možné využít API dotaz přes webové rozhraní mostu – na stránce http://IP_Adresa_RPI:8080/homeconnect se přejde do sekce Appliances (Zařízení), kde je dostupné tlačítko Appliance Details. Po jeho stisknutí se zobrazí podrobné informace o zařízení, včetně požadovaného haId.

Po zadání všech údajů bylo zařízení vytvořeno kliknutím na tlačítko Create Thing. Tímto krokem bylo zařízení úspěšně připojeno do systému openHAB. V dalším kroku lze přistoupit k připojení jednotlivých kanálů (Channels), které zařízení nabízí pro sledování stavu a ovládání funkcí myčky.

7.7.4 Vytváření kanálů(Channels)

Kanály (Channels) zařízení byly nakonfigurovány prostřednictvím webového rozhraní (Obrázek 30). Alternativně je možné vytvořit celé zařízení (Thing) včetně všech potřebných kanálů pomocí vývojového prostředí Visual Studio Code.



Obrázek 30 - Nastavení kanálů pro ovládání myčky na nádobí

(zdroj: vlastní)

Pro účely ovládání myčky nádobí byly vybrány následující funkční kanály:

- zapnutí / vypnutí myčky
- aktivace / deaktivace vzdáleného startu (nezbytné pro spuštění přes aplikaci)

- c) výběr a spuštění programu
- d) sledování stavu dveří (otevřeno / zavřeno)
- e) zbývající čas do ukončení programu

Konkrétní postup vytvoření propojení položky s kanálem je popsán na příkladu napájení. V seznamu kanálů zařízení bylo kliknuto na položku Napájení a zvolena možnost Add Link to Item. Kanál zůstal zachován, avšak v části Items byla vybrána volba Create a new Item. Po potvrzení volby tlačítkem Link byla vytvořena nová položka, která umožňuje ovládat napájení myčky.

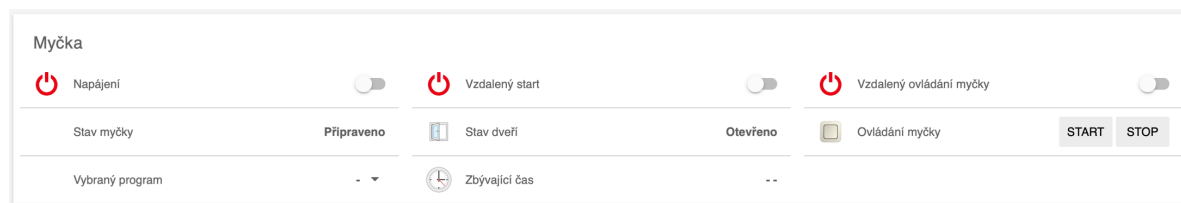
7.7.5 Vytvoření sitemapy

Pro potřeby ovládání a monitorování stavu myčky bylo navrženo uživatelské rozhraní (sitemap), které zobrazuje klíčové provozní informace a zároveň umožňuje ovládání základních funkcí zařízení (Obrázek 31).

V rámci této struktury byly definovány jednotlivé položky (Items), které slouží ke spínání napájení, aktivaci vzdáleného ovládání, výběru a spuštění programu, zobrazení stavu dveří, aktuálního režimu provozu a zbývajících času programu.

Rozhraní využívá mapování (MAP) pro převod vnitřních hodnot do srozumitelných českých popisků.

```
Frame label="Myčka" {
    Switch icon=switch item=Dishwasher_Napajeni label="Napájení"
    Switch icon=switch
item=Dishwasher_Remote_Start-Allowance_State label="Vzdálený start"
    Switch icon=switch
item=Dishwasher_Remote_Control_Activation_State label="Vzdálený
ovládání myčky"
    Text icon=play-pause item=Dishwasher_Operation_State
label="Stav myčky[MAP(cz.map):%s]"
    Text icon=contact item=Dishwasher_Door_State label="Stav dveří
[MAP(cz.map):%s]"
    Switch icon=wallswitch item=Dishwasher_Program_Actions
label="Ovládání myčky" mappings=["start"="Start", "stop"="Stop"]
    Selection item=Dishwasher_Selected_Program label="Vybraný
program[%s] refreshInterval=1 "
    Text icon=time item=Dishwasher_Remaining_Program_Time
label="Zbývající čas"
}
```



Obrázek 31 - Sitemapa pro ovládání myčky nádobí

(zdroj: vlastní)

7.7.6 Vytvoření pravidla

Bylo vytvořeno pravidlo, které zabraňuje spuštění myčky v případě, že jsou otevřené dveře. Pokud je při pokusu o zapnutí zjištěno, že dveře nejsou zavřené, je zapnutí automaticky zrušeno, uživatel je upozorněn a informace je zaznamenána do logu.

```
rule " Zakaz spusteni - dveře otevřené"
when
    Item Dishwasher_Napajeni received command ON
then
    if (Dishwasher_Door_State.state == OPEN) {
        // Zrušíme zapnutí
        Dishwasher_Napajeni.sendCommand(OFF)
        sendNotification("email@email.com", "Myčka nebyla zapnuta,
        protože jsou otevřené dveře.")
        logInfo("Myčka", "Zabráněno zapnutí myčky - dveře jsou
        otevřené.")
    } else {
        logInfo("Myčka", "Myčka byla zapnuta.")
    }
end
```

7.8 Integrace získání ceny elektřiny

V této části byla do systému openHAB integrována funkcionality pro automatizované získávání aktuálních cen elektřiny, přičemž data jsou pravidelně stahována prostřednictvím rozhraní API a dále zpracovávána pro potřeby optimalizace řízení spotřeby v domácnosti.

7.8.1 Soubor pro získání ceny

Pro získání aktuální ceny elektřiny bylo využito rozhraní Electree API, které poskytuje informace o SPOTové ceně elektřiny v daný okamžik. Poskytovatel Electree byl zvolen především z důvodu, že umožňuje provádět API dotazy na základě EAN čísla odběrného místa, díky tomu je zajištěna pravost získaných údajů.

Získávání cenových dat v reálném čase je klíčové pro efektivní řízení spotřeby elektrické energie v domácnosti, zejména v kombinaci se solárními panely, bateriemi a měničem. Na základě těchto dat lze například rozhodovat, kdy odebírat energii ze sítě a kdy ji spíše ukládat nebo spotřebovávat z vlastních zdrojů.

Níže uvedený kód odesílá požadavek na rozhraní Electree API a vrací odpověď s cenou elektřiny pro aktuální hodinu. Aby bylo možné využít jednodušší implementaci, tento skript navíc zahrnuje i časový filtr, který umožňuje výběr konkrétní cenové hodnoty podle aktuálního času.

```
import requests
from datetime import datetime
import pytz
EAN = "EAN"
# URL pro Electree API
url = f"https://api.electree.cz/{EAN}/prices"
response = requests.get(url)
if response.status_code != 200:
    print("CHYBA")
    exit(0)
data = response.json()
# Aktuální čas zaokrouhlený na celou hodinu ve formátu JSON
now = datetime.now(pytz.timezone("Europe/Prague")).replace(minute=0,
    second=0, microsecond=0)
formatted_time = now.strftime("%Y-%m-%d, %H:%M:%S")
# Cena pro aktualni hodinu
for entry in data:
    if entry["timeLocalStart"] == formatted_time:
        cena_czk_mwh = entry["priceCZK"]
        cena_czk_kwh = cena_czk_mwh / 1000
        print(f"{cena_czk_kwh:.2f}")
        break
    else:
        print("Cena pro aktuální čas nebyla nalezena.")
```

7.8.2 Vytvoření bash skriptu pro spuštění Python skriptu

Podobně jako při integraci měniče solární elektrárny byl vytvořen Bash skript `electree.sh`, který slouží ke spuštění Python skriptu `electree_price.py` uloženého ve složce

/etc/openhab/scripts. Skript provede načtení aktuální ceny elektřiny a převede výstup do jednořádkového formátu.

```
#!/bin/bash
RESULT=$(python3 /etc/openhab/scripts/electree_price.py)
echo "$RESULT" | tr '\n' ' '
```

Pro umožnění spouštění skriptu bylo nutné upravit oprávnění pomocí následujícího příkazu.

```
sudo chmod 744 electree.sh
```

7.8.3 Vytvoření souboru .things

Následně byla vytvořena definice nové věci (Thing) typu `exec:command`. V definici je uvedeno, jaký příkaz má být spuštěn. Interval spouštění je nastaven na 0 (neperiodický) a volba `autorun` je deaktivována. Spuštění skriptu je řízeno samostatným pravidlem, které zajišťuje, že se požadavek odešle pouze ve specifikovaném čase nebo při splnění určitých podmínek. Tento přístup se liší od integrace měniče, kde byl skript spouštěn periodicky.

```
Thing exec:command:electree_price "Electree Price" [
  command="bash /etc/openhab/scripts/electree.sh",
  interval=0,
  timeout=20,
  autorun=false
]
```

7.8.4 Vytvoření souboru .items

Byly vytvořeny následující položky (Items) pro práci se skriptem a pro zpracování a zobrazení výstupních dat.

```
String ElectreePrice "Cena elektřiny [%s Kč/kWh]" {
  channel="exec:command:electree_price:output" }

Switch ElectreeTrigger "Spustit načtení ceny" {
  channel="exec:command:electree_price:run" }
```

Položka `ElectreePrice` slouží k načtení výstupu skriptu a zobrazení aktuální ceny elektřiny ve formátu textu. Spuštění skriptu je řízeno pomocí přepínače `ElectreeTrigger`, který je napojen na kanál `run` definovaný v rámci věci `exec:command:electree_price`.

Tímto způsobem lze provádět manuální nebo pravidlové načítání cen elektřiny z rozhraní `Electree API` dle potřeby.

7.8.5 Vytvoření souborů .rules

Pro zajištění pravidelného načítání ceny elektřiny bylo vytvořeno pravidlo, které se aktivuje pomocí časového spínače (cron). Spuštění je naplánováno tak, aby proběhlo vždy v první minutě každé celé hodiny.

Po aktivaci pravidla je do položky ElectreeTrigger odeslán příkaz ON a dojde ke spuštění skriptu, který načítá aktuální cenu elektřiny prostřednictvím rozhraní Electree API.

```
rule "Spustit Electree skript každou hodinu"
when
    Time cron "0 1 * * * ?"
then
    logInfo("electree", "Spouštím načtení ceny z Electree API")
    ElectreeTrigger.sendCommand(ON)
End
```

Položka ElectreeTrigger byla definována i v souboru .items jako ovládací prvek, který umožňuje manuální spuštění skriptu přímo uživatelem. Tento přístup je užitečný zejména v případech, kdy by došlo k výpadku připojení, problému s API nebo jiným okolnostem, které by si vyžádaly opakované načtení dat mimo pravidelný časový interval.

7.8.6 Vytvoření sitemapy

Pro přehledné zobrazení aktuální ceny elektřiny byl do uživatelského rozhraní (sitemap) přidán jednoduchý vizuální blok (Frame), který obsahuje jednu textovou položku a ikonu. (Obrázek 32).

Zobrazená hodnota pochází z položky ElectreePrice, která je aktualizována na základě výstupu skriptu spouštěného každou hodinu.

```
Frame label="Cena elektřiny" {
    Text icon=Price item=ElectreePrice label="Cena elektřiny"
}
```



Obrázek 32 - Sitemapa pro zobrazování ceny elektřiny

(zdroj: vlastní)

7.9 Integrace ovládání bojleru

Pro integraci ovládání bojleru bylo využito rozhraní SmartThings API od společnosti Samsung. Ovládání bojleru je zajišťováno zařízením Sonoff TH16R2, které bylo doposud spravováno prostřednictvím aplikace výrobce eWeLink. Vzhledem k tomu, že zařízení Sonoff nemají oficiální doplněk pro systém openHAB, bylo nutné zvolit alternativní řešení právě pomocí služby SmartThing.

7.9.1 Připojení zařízení eWeLink do služby SmartThings pro ovládání bojleru

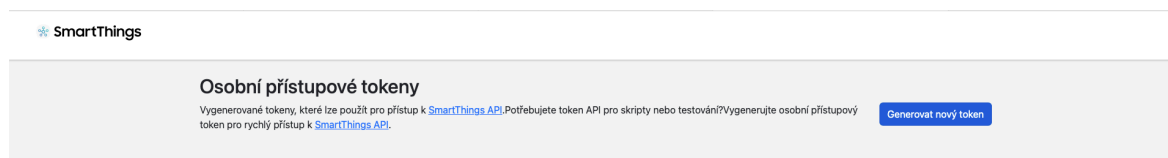
Podmínkou tohoto kroku je mít aktivní ovládání zařízení v aplikaci eWeLink, nainstalovanou aplikaci SmartThings a vytvořený účet Samsung. Po úspěšném přidání zařízení může být aplikace SmartThings následně odinstalována.

Na úvodní obrazovce aplikace SmartThings bylo nejprve zvoleno tlačítko „+“, poté možnost „Přidat zařízení“ a následně sekce „Partnerská zařízení“. Ze zobrazeného seznamu byla vybrána integrační možnost „eWeLink“. Tím byl zahájen proces propojení účtu a zařízení mezi oběma platformami.

Po výběru možnosti „eWeLink“ byl uživatel vyzván k určení typu zařízení, kde byla zvolena kategorie „Přepínač/stmívač“. Poté následovalo vybrání možnosti „eWeLink Smart Home“ a přesměrování na přihlašovací rozhraní služby eWeLink. Po úspěšném přihlášení bylo ze seznamu zvolených zařízení vybráno to, které mělo být přidáno do systému SmartThings.

7.9.2 Vytvoření tokenu pro ovládání přes API

Po přihlášení na stránce pro vytvoření tokenu <https://account.smartthings.com/tokens>, kde byla zvolena možnost „Generovat nový token“ (Obrázek 33). Následně bylo zobrazeno výběrové okno, ve kterém byla vybrána oprávnění pro zařízení.



Obrázek 33 - Generování osobního tokenu

(zdroj: vlastní)

7.9.3 Vytvoření souboru .things

Pro ovládání bojleru a sledování jeho aktuální teploty byly vytvořeny dva samostatné soubory typu Thing. V prvním souboru (smarththings_relay) bylo integrováno ovládání pro zapnutí a vypnutí zařízení Sonoff TH16R2. Využit byl doplněk HTTP Binding, který umožňuje odesílání požadavků prostřednictvím rozhraní REST API služby SmartThings. Ovládací příkazy byly definovány ve formátu JSON a komunikace probíhá metodou POST na specifikovanou adresu zařízení.

```
Thing http:url:smarththings_relay "SmartThings Switch" [

    baseUrl="https://api.smarththings.com/v1/devices/IDZarizeni/command",
    headers="Authorization=Bearer VYGENEROVANY_TOKEN",
    commandMethod="POST",
    contentType="application/json",
    ignoreSSLErrors=true
] {
    Channels:
        Type switch : power "Boiler Power" [

onValue="{\"commands\": [{\"component\": \"main\", \"capability\": \"switch\", \"command\": \"on\"}]}",

offValue="{\"commands\": [{\"component\": \"main\", \"capability\": \"switch\", \"command\": \"off\"}]}"

        ]}
}
```

Druhý soubor slouží ke čtení aktuální teploty z připojeného senzoru. Pro tento účel byla použita metoda GET na koncový bod /status, který vrací strukturovaná data ve formátu JSON. Pomocí transformace JSONPATH je z odpovědi vybrána konkrétní hodnota teploty, která je následně zobrazena v systému openHAB jako číselná hodnota v rámci kanálu typu number.

```
Thing http:url:smarththings_temp "SmartThings Teplota" [

    baseUrl="https://api.smarththings.com/v1/devices/IDZarizeni/status",
    headers="Authorization= VYGENEROVANY_TOKEN",
    refresh=60,
    ignoreSSLErrors=true
] {
    Channels:
        Type number : temperature "Sonoff TH16 Teplota" [
```

```
stateTransformation="JSONPATH:$.components.main.temperatureMeasurement.temperature.value"
  ] }
```

7.9.4 Vytvoření souboru .items

Byly vytvořeny následující položky (Items) pro ovládání zařízení a získávání aktuální teploty. První položka typu Number s názvem Sonoff_TH16_Temperature slouží ke zobrazení teploty získané z rozhraní služby SmartThings. Hodnota je formátována na jedno desetinné místo a doplněna jednotkou stupňů Celsia. Druhá položka typu Switch s názvem SmartThings_Relay je určena pro zapínání a vypínání bojleru prostřednictvím kanálu propojeného s ovládacím API. Obě položky jsou propojeny s odpovídajícími kanály definovanými v souborech typu Thing.

```
Number    Sonoff_TH16_Temperature    "Teplota bojleru [%1f °C]"    {
    channel="http:url:smartthings_temp:temperature" }
Switch          SmartThings_Relay          "Switch"          {
    channel="http:url:smartthings_relay:power" }
```

7.9.5 Vytvoření sitemapy

Bylo vytvořeno jednoduché uživatelské rozhraní (sitemapa), ve kterém je umístěn přepínač pro ovládání napájení bojleru a textová položka zobrazující aktuální hodnotu teploty získanou z připojeného senzoru (Obrázek 34).

```
Frame label="Bojler" {
    Switch icon=switch item=SmartThings_Relay label="Bojler"
    Text icon=temperature item=Sonoff_TH16_Temperature label="Teplota
    bojleru [%1f °C]"
}
```

Bojler



Bojler



Teplota bojleru

52.5 °C

Obrázek 34 - Sitemapa pro ovládání a získání informací o bojleru

7.10 Integrace ovládání Sony TV

Pro ovládání televizorů byl využit stejný doplněk (Binding) systému openHAB, a proto je v této kapitole popsána pouze jedna z integrací jako reprezentativní příklad.

V první fázi bylo zařízení spuštěno pomocí fyzického ovladače. V nastavení televizoru byly následně aktivovány možnosti vzdáleného spuštění a dálkového ovládání. Po integraci bylo důležité zaregistrovat systém openHAB jako důvěryhodné zařízení.

The screenshot shows the configuration page for a Sony TV (KD-55AG8) in the openHAB interface. The page is divided into sections: Information, Configuration, and Discovered information. The Information section includes fields for Thing UID (sony.scalar:0e617246d988), Label (KD-55AG8), and Location (e.g. Kitchen). The Configuration section includes fields for Scalar Web URL (http://192.168.1.45985/sony/webapi/ssdp/ddd.xml), Device MAC Address, IRCC URL (http://192.168.1.45985/sony/webapi/ssdp/ddd.xml), Access Code (4878), and Discovered MAC Address (READONLY). The Discovered information section includes fields for Discovered Commands Map File (READONLY) and Discovered Model Name (READONLY).

Obrázek 35 - Konfigurace Sony TV

(zdroj: vlastní)

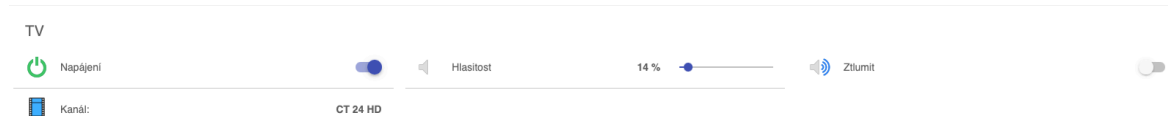
Při samotné integraci byl zvolen doplněk Sony Device, v jehož rámci byla zadána IP adresa daného televizoru (Obrázek 35). Na obrazovce televizoru se následně zobrazil PIN kód, který bylo nutné zadat v uživatelském rozhraní openHAB pro navázání zabezpečeného spojení mezi oběma zařízeními.

Po úspěšném spárování bylo přistoupeno k propojení jednotlivých kanálů (Channels), a to obdobným způsobem, jaký je podrobně popsán v kapitole věnované integraci myčky nádobí (viz kapitola 6.7). Kanály byly napojeny na jednotlivé položky (Items), které umožňují například zapnutí a vypnutí televizoru nebo změnu vstupu.

7.10.1 Vytvoření sitemapy

Pro ovládání televizoru bylo vytvořeno jednoduché uživatelské rozhraní, které umožňuje zapnutí a vypnutí zařízení, úpravu hlasitosti, ztlumení zvuku a zobrazení aktuálně přehrávaného kanálu (Obrázek 36). Jednotlivé funkce jsou realizovány prostřednictvím přepínačů, posuvníku a textového prvku v rámci vizuálního rámce s označením „TV“.

```
Frame label="TV" {
    Switch icon=switch item=KD55AG8_Power_Status label="Napájení"
    Slider      icon=soundvolume      item=KD55AG8_Volume_speaker
    label="Hlasitost [%d %%]"
    Switch      icon=soundvolume      item=KD55AG8_Mute_speaker
    label="Ztlumit"
    Text icon=video item=KD55AG8_Playing_Title label="Kanál:"}
```



Obrázek 36 - Sitemapa pro ovládání TV

(zdroj: vlastní)

7.11 Integrace ukládání dat do paměti

OpenHAB umí uchovávat data v čase – tomuto mechanismu se říká persistence (perzistence). Uložená data lze později opět načíst, například pro obnovení systému po spuštění nebo pro vytváření grafů, které se zobrazují v uživatelském rozhraní. [72]

Ve výchozím nastavení je systém openHAB dodáván s perzistentní službou RRD4J, která uchovává stav každé položky při každé změně a minimálně jednou za minutu. Tato služba navíc při startu systému obnovuje poslední známou hodnotu.

Balíček RRD4J Persistence byl nainstalován z obchodu doplňků, pokud již nebyl součástí výchozí instalace. Ve složce `etc/openhab/persistence` byl následně vytvořen konfigurační soubor `rrd4j.persist`, ve kterém byla nastavena pravidla pro ukládání dat.

```
Strategies {
    everyMinute : "0 * * * * ?"
    default = everyUpdate
}
Items {
    ESPHome_device_bedroom_bedroomlocal_100191_teploataPokoj : strategy
= everyUpdate, everyMinute
    livingroomteplota_Teploata_obyvak : strategy = everyUpdate,
everyMinute
    zpPv : strategy = everyUpdate, everyMinute
    zpBattery : strategy = everyUpdate, everyMinute
    zpConsumption : strategy = everyUpdate, everyMinute
}
```

V souboru `.persist` bylo definováno, že vybrané položky budou ukládány do paměti při každé změně (`everyUpdate`) a zároveň pravidelně každou minutu (`everyMinute`). Tím je zajištěno zachycení všech změn i průběžných hodnot, které mohou být dále využity v grafech a analýzách. Seznam sledovaných položek lze v případě potřeby rozšířit.

8 NASTAVENÍ A PŘIPOJENÍ OPENHAB CLOUD

8.1 Co je to openHAB Cloud?

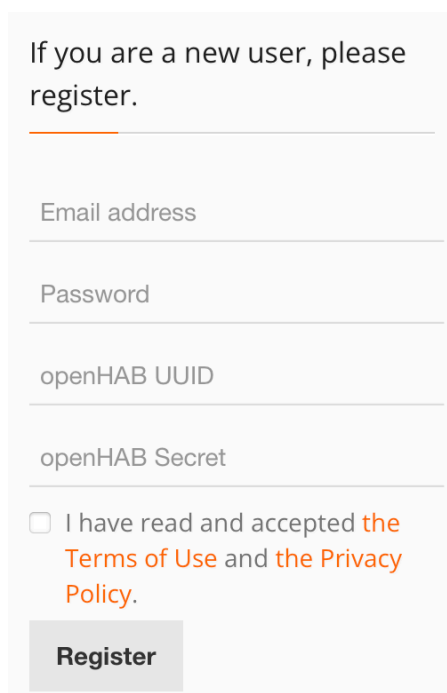
Pomocí doplňku OpenHAB Cloud Connector je umožněno propojení lokální instalace systému openHAB se vzdálenou instancí OpenHAB Cloud, například službou myopenHAB.org, která je provozována nadací openHAB. [73]

8.2 Integrace openHAB Cloud

Prvním krokem integrace bylo nainstalování služby openHAB Cloud prostřednictvím obchodu doplňků (Add-on Store). V sekci System Integrations byl vyhledán odpovídající doplněk a následně stažen. Přítomnost této služby je klíčová pro zajištění vzdáleného přístupu k chytré domácnosti a pro možnost odesílání notifikací.

Po úspěšné instalaci byla otevřena webová stránka <https://myopenhab.org>, kde byl vytvořen nový uživatelský účet. Pro dokončení registrace bylo nutné zadat identifikátory systému – openHAB UUID a Secret, které je možné získat pomocí příkazů z terminálu nebo jejich načtením ze souborů ve složce /var/lib/openhab (Obrázek 37).

```
cat /var/lib/openhab/uuid  
cat /var/lib/openhab/openhabcloud/secret
```



If you are a new user, please register.

Email address

Password

openHAB UUID

openHAB Secret

☐ I have read and accepted the Terms of Use and the Privacy Policy.

Register

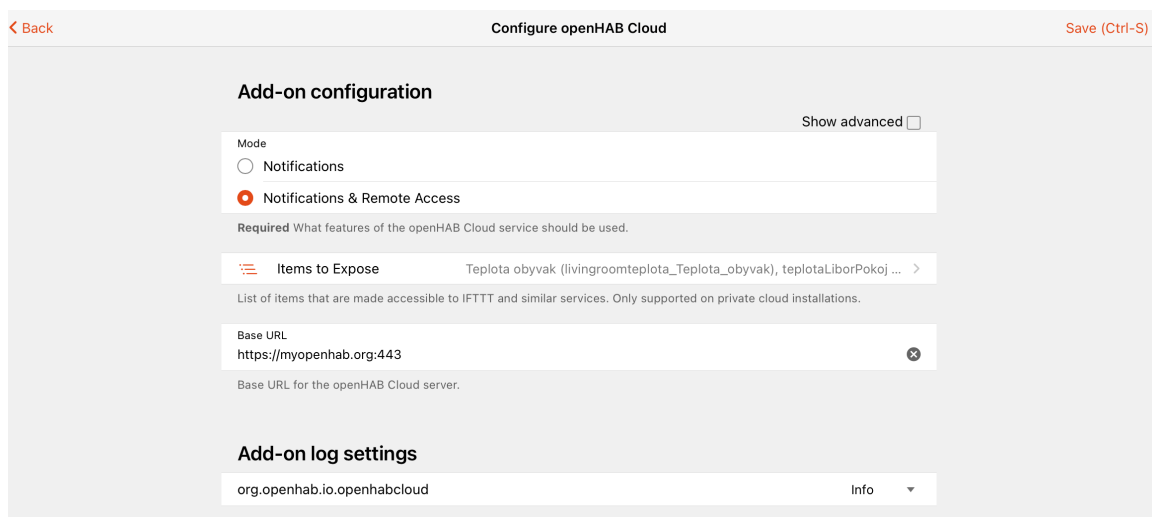
Obrázek 37 - Registrační formulář pro openHAB cloud

(zdroj: vlastní)

Po vytvoření účtu na portálu myopenHAB.org je uživatel přesměrován na hlavní stránku služby. Zde je třeba kliknout na odkaz „Click here to access your openHAB’s dashboard“ a dojde k otevření uživatelského rozhraní připojené instance openHAB.

Po přihlášení pomocí dříve vytvořeného účtu je systém připraven k plnohodnotnému vzdálenému ovládání. Tímto způsobem je zajištěn bezpečný přístup k chytré domácnosti odkudkoliv, a to bez nutnosti nastavovat veřejnou IP adresu nebo port forwarding na domácím routeru.

Dále byl nakonfigurován doplněk pro službu openHAB Cloud (Obrázek 38). Byla zvolena možnost „Notifications & Remote Access“ a zadána adresa základní URL pro vzdálené ovládání.

The screenshot shows the 'Configure openHAB Cloud' web interface. At the top, there are 'Back' and 'Save (Ctrl-S)' links. The main section is titled 'Add-on configuration' and includes a 'Show advanced' checkbox. Under 'Mode', 'Notifications & Remote Access' is selected with a radio button. A 'Required' note states: 'What features of the openHAB Cloud service should be used.' Below this is the 'Items to Expose' section, showing 'Teplota obyvák (livingroomteplota_Teplota_obyvak), teplotaLiborPokoj ...' with a right arrow. A note below says: 'List of items that are made accessible to IFTTT and similar services. Only supported on private cloud installations.' The 'Base URL' field contains 'https://myopenhab.org:443' with a help icon. A note below reads: 'Base URL for the openHAB Cloud server.' The 'Add-on log settings' section shows 'org.openhab.io.openhabcloud' with an 'Info' dropdown menu.

Obrázek 38 - Konfigurace doplňku openHAB Cloud

(zdroj: vlastní)

8.3 Vytvoření nového uživatele s rolí – Regular user

The screenshot shows a user creation form. It has four input fields: 'Email address', 'Password', and 'Verify password'. Below these is a dropdown menu with 'Regular user' selected. At the bottom left is a button labeled 'Add user'.

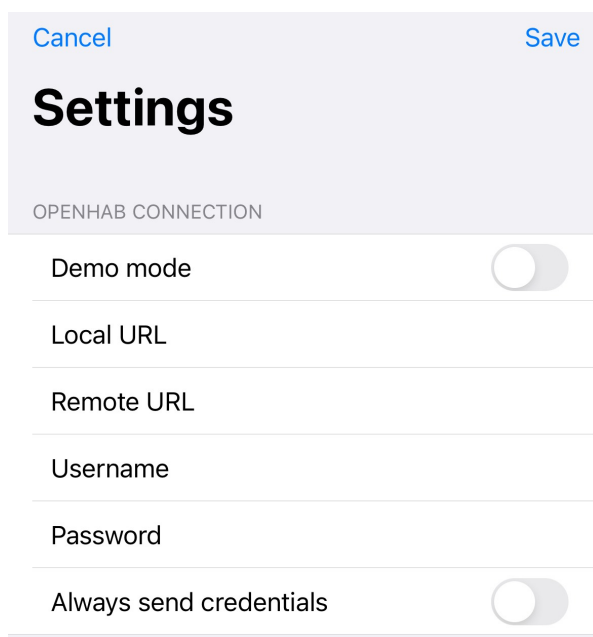
Obrázek 39 - Vytvoření nového uživatele

(zdroj: vlastní)

Pro vytvoření nového uživatele bylo v prostředí myopenHAB.org přejeto kurzorem myši na e-mailovou adresu účtu v pravém horním rohu obrazovky a z nabídky byla vybrána možnost Users. Následně bylo kliknuto na tlačítko Add new user. Otevřel se formulář pro zadání potřebných údajů. Po jeho vyplnění dle předloženého vzoru byl nový uživatel úspěšně vytvořen (Obrázek 39).

8.4 Připojení aplikace openHAB na mobilním zařízení

Aplikace openHAB je dostupná pro mobilní zařízení s operačním systémem iOS i Android a lze ji bezplatně stáhnout z oficiálního obchodu s aplikacemi (App Store nebo Google Play). Po její instalaci následuje přihlášení k vlastní instanci systému openHAB, které je nutné provést v sekci nastavení. Zde byly zadány požadované údaje: do pole „Local URL“ byla vložena adresa zařízení v lokální síti ve formátu `https://IP_Adresa:8080`. Do pole „Remote URL“ byla zadána vzdálená adresa služby openHAB Cloud, která má tvar `https://myopenhab.org:443`.



Obrázek 40 - Sekce nastavení v mobilní aplikaci

(zdroj: vlastní)

Následně byly vyplněny přihlašovací údaje, tedy uživatelské jméno nebo e-mail a heslo k účtu vytvořenému na portálu myopenHAB.org. Po uložení nastavení prostřednictvím tlačítka „SAVE“ došlo k navázání spojení se službou openHAB Cloud a bylo tak umožněno vzdálené ovládání systému i mimo lokální síť (Obrázek 40).

Aplikace zároveň nabízí možnost vyzkoušet její funkce v tzv. „Demo“ režimu, ve kterém není třeba žádné připojení k vlastní instanci systému. Tento režim je vhodný například pro první seznámení s uživatelským rozhraním a základními funkcemi aplikace.

8.5 Vytvoření notifikací

8.5.1 Notifikace – Program dokončen mycí zařízení

Pro zvýšení komfortu uživatele bylo vytvořeno pravidlo, které slouží k odeslání oznámení o dokončení programu myčky nádobí. Pravidlo se aktivuje ve chvíli, kdy se hodnota položky Dishwasher_Remaining_Program_Time změní na UNDEF. Tato hodnota indikuje, že program byl ukončen. Notifikace je následně odeslána na e-mailovou adresu prostřednictvím služby openHAB Cloud.

```
rule "Myčka dokončila program"
when
    Item Dishwasher_Remaining_Program_Time changed to UNDEF
then
    sendNotification("email@email.com", "Myčka dokončila program.")
    logInfo("Myčka", "Program dokončen - stav přešel na UNDEF.")
end
```

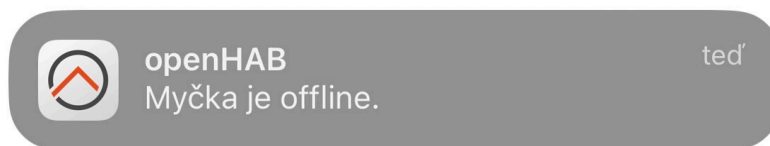
Tato funkcionálita umožňuje automaticky informovat uživatele o dokončení cyklu bez nutnosti fyzické kontroly zařízení.

8.5.2 Notifikace – Zařízení je offline

Pro zajištění včasného informování uživatele o případné ztrátě spojení se zařízením bylo vytvořeno pravidlo, které sleduje změnu stavu zařízení myčky nádobí z ONLINE na OFFLINE. V případě výskytu takové události je odeslána notifikace na zvolenou e-mailovou adresu prostřednictvím služby openHAB Cloud.

```
rule "Notifikace: Myčka je offline"
when
    Thing "homeconnect:dishwasher:MyčkaNadobi" changed from ONLINE to OFFLINE
then
    sendNotification("email@email.com", "Myčka je offline")
end
```

Tímto způsobem je uživatel upozorněn na výpadek spojení a může včas provést kontrolu zařízení nebo sítě. Tento mechanismus je univerzálně použitelný i pro další zařízení připojená k systému openHAB.



Obrázek 41 – Notifikace – zařízení offline

(zdroj: vlastní)

9 VÝSLEDKY

9.1 Ovládání zařízení

V rámci implementace byly do systému openHAB úspěšně integrovány různé prvky chytré domácnosti. Jednalo se především o zařízení, která byla původně ovládána prostřednictvím vlastních mobilních aplikací. Cílem bylo tato řešení sjednotit do jednoho centrálního systému. Ovládání bylo realizováno buď pomocí oficiálních doplňků dostupných v obchodech s doplňky nebo prostřednictvím přímé komunikace přes rozhraní API.

Pro potřeby integrace solární elektrárny a získávání aktuální ceny elektřiny byl využit programovací jazyk Python. Ten nabízí knihovny umožňující komunikaci s měničem GoodWe a zpracování požadavků vůči externím API službám.

U zařízení pro ohřev vody (bojler) byl zvolen alternativní přístup pomocí SmartThings API. Ten umožňuje základní ovládání, avšak jeho funkčnost je omezena platností autorizačního tokenu, která je limitována na 24 hodin. Z tohoto důvodu je nutné token pravidelně obnovovat, případně zařízení nahradit jiným, které bude kompatibilní s odlišným typem integrace. Na základě zjištěných skutečností se plánuje výměna zařízení s cílem usnadnit integraci a snížit závislost na externí platformě.

Ovládání ostatních zařízení, která byla integrována prostřednictvím oficiálních doplňků, probíhá bezproblémově a stabilně.

9.2 Vizualizace systému

Uživatel má k dispozici ovládání systému jak prostřednictvím webového rozhraní, tak i pomocí mobilní aplikace. Přístupová oprávnění jsou rozdělena dle úrovně uživatele a určují jeho možnosti v rámci systému. Uživatelská rozhraní byla navržena na základě přehlednosti a logického členění. Zařízení jsou seskupena podle místností, typů (např. klimatizace, fotovoltaika) a účelů. Pro rychlý přehled klíčových údajů o provozu solární elektrárny byl vytvořen vlastní widget, který je zobrazen na hlavní stránce v obou variantách rozhraní.

Pro rychlou orientaci byl na teplotní stanici použit OLED displej, který zobrazuje aktuální teplotu přímo na místě bez nutnosti přístupu k webovému rozhraní.

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a implementovat systém chytré domácnosti pomocí open-source platformy openHAB. Byla provedena analýza dostupných technologií a zařízení, navržen celkový koncept systému a následně realizována konkrétní integrace vybraných prvků domácí automatizace. Ovládání různých zařízení bylo úspěšně sjednoceno do jednoho centralizovaného prostředí. Hlavní cíl práce tím byl naplněn.

Během implementace byly identifikovány technické limity u některých zařízení. Například při integraci bojleru prostřednictvím služby SmartThings je nutné pravidelně obnovovat autorizační token, neboť jeho platnost je omezena na 24 hodin. Zařízení značky Sonoff v současnosti postrádají oficiální doplněk pro openHAB, a tím je komplikováno jejich plnohodnotné zapojení do systému.

Rovněž bylo zjištěno, že dostupný doplněk pro platformu SemsPortal, určený k integraci solárních měničů značky GoodWe, poskytuje pouze omezený rozsah informací. Uživatel má k dispozici například aktuální výkon, denní a měsíční výrobu či odhadovaný výnos, avšak chybí důležité ukazatele jako kapacita baterie, aktuální spotřeba domácnosti či tok energie mezi sítí a domácností, které jsou klíčové pro efektivní energetické řízení. Z těchto důvodů by bylo vhodné vytvořit doplněk s širší funkcionalitou, který by tyto hodnoty zpřístupnil přímo v systému.

V budoucnu by bylo vhodné nahradit zařízení značky Sonoff jiným, které umožní plně funkční a bezproblémové ovládání bojleru bez nutnosti pravidelného obnovování tokenu. Zároveň by bylo možné zvážit přechod z Raspberry Pi na výkonnější platformu, například NAS server, který by umožnil nejen provoz systému openHAB, ale i využití pro další účely, jako je ukládání fotografií, filmů a dalších uživatelských dat.

Z hlediska výzkumného přínosu by se také nabízelo porovnání s jinými open-source systémy, jako je Home Assistant.

SEZNAM POUŽITÉ LITERATURY

- [1] *Carneo*. Online. Dostupné z: <https://www.carneo.cz/blog/chytra-domacnost-neboli-smart-home-setri-cas-a-usnadnuje-zivot/?srltid=AfmBOopQpMuE7AvjomY261xYsqCD2AsT2fgtWdTKSJzcOHEHvAvDXezX>. [cit. 2025-02-24].
- [2] *Tipa*. Online. Dostupné z: <https://www.tipa.eu/cz/czchytra-domacnost-vyhody-noveho-komfortniho-zivota/t-1457/>. [cit. 2025-02-24].
- [3] *Feedit.cz*. Online. Dostupné z: <https://feedit.cz/2018/09/21/vyvoj-chytrych-domacnosti-od-kutilova-domu-na-tlacitko-po-umelou-inteligenci-a-virtualni-realitu/>. [cit. 2025-02-24].
- [4] *InHaus*. Online. Dostupné z: <https://inhaus.cz/clanek/863/vyvoj-chytrych-domacnosti-od-kutilova-domu-na-tlacitko-po-umelou-inteligenci-a-virtualni-realitu/>. [cit. 2025-02-24].
- [5] *My Smart Home*. Online. Dostupné z: https://www.mysmarthome.cz/blog/3-nejcastejsi-bezpecnostni-hrozby-chytre-domacnosti-a-jak-se-jim-branit/?utm_source=chatgpt.com. [cit. 2025-02-24].
- [6] *Eset*. Online. Dostupné z: <https://www.eset.com/cz/botnet/>. [cit. 2025-02-24].
- [7] LOM, Michal a PŘIBYL, Ondřej. *Tzbinfo*. Online. Tzbinfo. 2017. Dostupné z: <https://elektro.tzb-info.cz/inteligentni-budovy/15569-rizika-chytrych-zarizeni-a-jejich-zabezpeceni>. [cit. 2025-02-24].
- [8] *ESET*. Online. Dostupné z: <https://www.eset.com/cz/cloud/>. [cit. 2025-02-24].
- [9] *Alza.cz*. Online. Dostupné z: <https://www.alza.cz/neprerusitelne-zdroje-napajeni-ups-pro-domacnosti-a-firmy>. [cit. 2025-02-24].
- [10] Online. Dostupné z: <https://www.alza.cz/bezpecna-chytra-domacnost>. [cit. 2025-02-24].
- [11] *Aulix*. Online. Dostupné z: <https://aulix.cz/cs/blog/page/jak-funguji-pohybove-cidla-b1807>. [cit. 2025-02-24].
- [12] *My Smart Home*. Online. Dostupné z: <https://www.mysmarthome.cz/kategorie/senzory-oken-dveri/>. [cit. 2025-02-24].
- [13] *Hasičský záchranný sbor České republiky*. Online. Dostupné z: <https://hzscr.gov.cz/clanek/detektory-koure-a-nebezpecnych-plynu.aspx>. [cit. 2025-02-24].
- [14] *Wago*. Online. Dostupné z: <https://www.wago.com/cz/knx>. [cit. 2025-02-24].
- [15] *Bacnet*. Online. Dostupné z: <https://bacnet.org/about-bacnet-standard/>. [cit. 2025-02-25].
- [16] *Wago*. Online. Dostupné z: <https://www.wago.com/cz/modbus>. [cit. 2025-02-25].
- [17] *Intel*. Online. Intel. Dostupné z: <https://www.intel.com/content/www/us/en/products/docs/wireless/what-is-bluetooth.html>. [cit. 2025-05-18].
- [18] *Tesla smart*. Online. Tesla smart. 2023. Dostupné z: <https://www.teslasmart.com/cz/technologie-zigbee-co-to-je-jake-ma-vyhody-a-jak-s-ni-zacit>. [cit. 2025-05-18].

- [19] *Alza*. Online. Alza. 2024. Dostupné z: <https://www.alza.cz/slovník/z-wave-art17515.htm>. [cit. 2025-05-18].
- [20] *EListy*. Online. EListy. 2024. Dostupné z: <https://elisty.cz/technika/matter-a-thread/>. [cit. 2025-05-18].
- [21] *Cisco*. Online. Cisco. 2025. Dostupné z: <https://www.cisco.com/c/en/us/products/wireless/what-is-wifi.html>. [cit. 2025-05-18].
- [22] *Dell Technologies*. Online. Dell Technologies. Dostupné z: <https://www.dell.com/support/contents/cs-cz/article/product-support/self-support-knowledgebase/networking-wifi-and-bluetooth/wi-fi-network-standards-overview>. [cit. 2025-06-01].
- [23] *My Smart Home*. Online. Dostupné z: <https://www.mysmarthome.cz/blog/komunikacni-protokoly-pro-chytrou-domacnost/>. [cit. 2025-02-25].
- [24] *LEDshopik*. Online. Dostupné z: <https://www.ledshopik.cz/co-je-to-chytra-zarovka-x31233>. [cit. 2025-02-25].
- [25] *Bola*. Online. Dostupné z: <https://www.bola.cz/poradna/chytre-termostaty>. [cit. 2025-02-25].
- [26] *Eon*. Online. Dostupné z: <https://www.eon.cz/radce/chytra-domacnost/chytra-mesta/co-jsou-a-co-umi-chytre-zasuvky/>. [cit. 2025-02-25].
- [27] *E-SHOP SVĚT ZÁMKŮ*. Online. Dostupné z: <https://eshop.svetzamku.cz/chytre-zamky/>. [cit. 2025-02-26].
- [28] *Smart tuya*. Online. Dostupné z: <https://www.smarttuya.cz/https/www-smarttuya-cz/chytre-kamery-c5-7-2-hm>. [cit. 2025-02-26].
- [29] *Casedo*. Online. Dostupné z: <https://www.detektor-shop.cz/cs/6-pozarni-hlasice>. [cit. 2025-02-26].
- [30] *Somfy*. Online. Dostupné z: <https://www.somfy.cz/produkty/venkovni-stineni/venkovni-rolety>. [cit. 2025-02-26].
- [31] *My Smart Home*. Online. Dostupné z: <https://www.mysmarthome.cz/kategorie/detekce-uniku-vody/>. [cit. 2025-02-26].
- [32] *Somfy*. Online. Dostupné z: <https://www.somfy.cz/objevte-somfy/so-open-se-somfy/hlasovi-asistenti>. [cit. 2025-02-26].
- [33] *Avismart*. Online. Dostupné z: <https://www.avitsmart.cz/tipy-triky/domaci-kino/>. [cit. 2025-02-26].
- [34] *OpenHAB*. Online. Dostupné z: <https://www.openhab.org/docs/>. [cit. 2024-10-17].
- [35] *OSGi Working Group*. Online. Dostupné z: <https://www.osgi.org/resources/where-to-start/>. [cit. 2025-02-11].
- [36] *Karaf Apache*. Online. Dostupné z: <https://karaf.apache.org/manual/latest/overview.html>. [cit. 2025-02-26].
- [37] *ECLIPSE FOUNDATION*. Online. Dostupné z: <https://projects.eclipse.org/projects/eclipse.equinox>. [cit. 2025-02-26].
- [38] *Jetty*. Online. Dostupné z: <https://jetty.org>. [cit. 2025-02-26].
- [39] *Heise*. Online. Dostupné z: <https://www.heise.de/news/Heimautomatisierung-mit-openHAB-neue-Bindings-und-Ausblick-in-die-Zukunft-2462924.html>. [cit. 2025-02-26].

- [40] *IoT awards*. Online. Dostupné z: <https://web.archive.org/web/20160309203723/http://postscapes.com/internet-of-things-award/2014/iot-open-source-project>. [cit. 2025-02-26].
- [41] *Java.net*. Online. Dostupné z: <https://web.archive.org/web/20140113160305/https://www.java.net//dukeschoice>. [cit. 2025-02-26].
- [42] *Wikipedia*. Online. Dostupné z: <https://en.wikipedia.org/wiki/OpenHAB>. [cit. 2025-02-26].
- [43] *OpenHAB - Docs*. Online. Dostupné z: <https://www.openhab.org/docs/>. [cit. 2025-02-26].
- [44] *Openhab - Things*. Online. Dostupné z: <https://www.openhab.org/docs/concepts/things.html>. [cit. 2025-02-26].
- [45] *OpenHAB - items*. Online. Dostupné z: <https://www.openhab.org/docs/concepts/items.html>. [cit. 2025-02-27].
- [46] *OpenHAB - Rules*. Online. Dostupné z: https://www.openhab.org/docs/tutorial/rules_introduction.html. [cit. 2025-02-27].
- [47] *OpenHAB - Textual Rules*. Online. Dostupné z: <https://www.openhab.org/docs/configuration/rules-dsl.html>. [cit. 2025-02-27].
- [48] *OpenHAB - Installation Overview*. Online. Dostupné z: <https://www.openhab.org/docs/installation/>. [cit. 2025-03-03].
- [49] *ChatGPT*. Online. Dostupné z: <https://chatgpt.com>. [cit. 2025-04-06].
- [50] *OpenHAB - openhabian*. Online. Dostupné z: <https://www.openhab.org/docs/installation/openhabian.html#installation-on-raspberry-pi-systems>. [cit. 2025-03-04].
- [51] *Raspberry Pi*. Online. Raspberry Pi. Dostupné z: <https://www.raspberrypi.com/software/>. [cit. 2025-06-01].
- [52] *Etcher*. Online. Dostupné z: <https://etcher-docs.balena.io>. [cit. 2025-03-04].
- [53] *HPmarket.cz*. Online. Dostupné z: <https://www.hpmarket.cz/productOpt.asp?konfId=M6D61A>. [cit. 2025-04-07].
- [54] *Sinclair*. Online. Dostupné z: https://www.sinclair-solutions.com/cs/produkty/nastenne-klimatizace-168/terrel-serie/9040-sih-soh-13bit.html?_gl=1*d09t2l*_up*MQ..*_gs*MQ..&gclid=CjwKCAjw--K_BhB5EiwAuWYoyPyhxQOeIQ442LKR8P3p1MTuj8RsEHAOcBilPB_V8uSZsz5bzhytBoCkNEQAvD_BwE&gbraid=0AAAAADvTPbFgnzKS33jAo4fTCFlqtP_Qk. [cit. 2025-04-11].
- [55] *Sinclair*. Online. Dostupné z: <https://www.sinclair-solutions.com/cs/produkty/multi-variable-serie-r32/nastenne-jednotky-keyon/9134-sih-09bik-053011000004300.html>. [cit. 2025-04-11].
- [56] *Q - Elektrik*. Online. Dostupné z: https://elektro.q-elektrik.cz/stridac-pro-fotovoltaku-goodwe-gw10k-et/?srsltid=AfmBOoqPbojbtHsnZKuQ8PHdN6Eqk6_IDLc3sZFGFg8U674PbuK9dh48. [cit. 2025-04-11].
- [57] *Sony CZ*. Online. Dostupné z: <https://www.sony.cz/electronics/support/televisions-projectors-lcd-tvs-android/-kd-49x8307c/specifications>. [cit. 2025-04-11].
- [58] *Sony CZ*. Online. Dostupné z: <https://www.sony.cz/electronics/support/televisions-projectors-oled-tvs-android/-kd-55ag8/specifications>. [cit. 2025-04-16].

- [59] *Heureka*. Online. Dostupné z: <https://mycky-nadobi.heureka.cz/bosch-smi6zcs00e/#specifikace/>. [cit. 2025-04-16].
- [60] *WiFi modul s teplotním senzorem Sonoff TH16*. Online. Dostupné z: https://www.pruel.cz/fotky30445/fotov/_ps_1414SonoffTH-manual.pdf. [cit. 2025-05-21].
- [61] *HP Printer Binding*. Online. Dostupné z: <https://www.openhab.org/addons/bindings/hpprinter/#hp-printer-binding>. [cit. 2025-03-11].
- [62] *GREE - Bindings*. Online. Dostupné z: <https://www.openhab.org/addons/bindings/gree/#gree-binding>. [cit. 2025-03-11].
- [63] MALÝ, Martin. *ESP32 prakticky*. 2024. Praha: Edici CZ.NIC. ISBN 978-80-88168-79-9.
- [64] *OpenHAB - Exec*. Online. Dostupné z: <https://www.openhab.org/addons/bindings/exec/>. [cit. 2025-03-17].
- [65] *Home Connect Binding*. Online. Dostupné z: <https://www.openhab.org/addons/bindings/homeconnect/#home-connect-binding>. [cit. 2025-03-17].
- [66] *OpenHAB community*. Online. 2021. Dostupné z: <https://community.openhab.org/t/sony-binding-3-2-0-3-9-0/128262>. [cit. 2025-04-23].
- [67] *OpenHAB - Http*. Online. 2025. Dostupné z: <https://www.openhab.org/addons/bindings/http/>. [cit. 2025-05-21].
- [68] *Analog.com*. Online. Dostupné z: <https://www.analog.com/en/products/ds18b20.html>. [cit. 2025-03-18].
- [69] *SOLDERED. Display OLED I2C 0.96" SSD1306 datasheet*. Online. Dostupné také z: https://www.mouser.com/datasheet/2/1398/Soldered_333099-3395096.pdf?srltid=AfmBOooEsau9j499MWphiHOWmp-yOUB0whtNHTEB3CRYgP8guFX6EVj3.
- [70] *Random Nerd Tutorials*. Online. In: . Dostupné z: <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/>. [cit. 2025-03-19].
- [71] *Home Connect*. Online. Dostupné z: <https://developer.home-connect.com/user/register>. [cit. 2025-04-15].
- [72] *OpenHAB persistence*. Online. 2025. Dostupné z: <https://www.openhab.org/docs/configuration/persistence.html#persistence>. [cit. 2025-04-23].
- [73] *OpenHAB Cloud*. Online. Dostupné z: <https://www.openhab.org/addons/integrations/openhabcloud/>. [cit. 2025-04-22].

SEZNAM OBRÁZKŮ

Obrázek 1 - Ukázka běžného provozu sítě a její zneužití pomocí DDoS útoku.....	17
Obrázek 2 - Možnosti platformy pro běh systému openHAB	28
Obrázek 3 - Vytvoření účtu správce	32
Obrázek 4 - Nastavení jazyka, státu a časového pásma.....	32
Obrázek 5 - Nastavení polohy	33
Obrázek 6 - Menu pro doplňky pro ukládání dat.....	33
Obrázek 7 - Instalovat doplňky/Add-ons.....	34
Obrázek 8 - Hlavní stránka systému openHAB.....	35
Obrázek 9 – Rozdělení hlavní stránky	35
Obrázek 10 - Ukázka prostředí chytré domácnosti.....	40
Obrázek 11 - Vytvoření API tokenu	47
Obrázek 12 - Sitemapa pro informace o tiskárně	49
Obrázek 13 - ESP8266 – Pinout	53
Obrázek 14 - Schéma zapojení teplotní stanice	53
Obrázek 15 - 3D model spodní části krabičky pro teplotní stanici.....	57
Obrázek 16 - 3D model vrchní části krabičky pro teplotní stanici	57
Obrázek 17 - Přidání zařízení ESPHome do systému openHAB	58
Obrázek 18 - Přidání položky teplota sklep.....	58
Obrázek 19 - Sitemapa pro zobrazování teploty.....	59
Obrázek 20 - Sitemapa pro ovládání klimatizace	61
Obrázek 21 - Výpis informací z měniče	63
Obrázek 22 - Widget solární elektrárna.....	66
Obrázek 23 - Sitemapa pro Solární panely	67
Obrázek 24 - Sitemapa pro spotřebu energie.....	67
Obrázek 25 - Sitemapa pro baterii	68
Obrázek 26 - Vytvoření Home Connect developer account.....	69
Obrázek 27 - Vytvoření aplikace pro připojení k systému openHAB.....	70
Obrázek 28 - Konfigurace mostu (bridge) pro Home Connect API.....	70
Obrázek 29 - Konfigurace myčky nádobí.....	71
Obrázek 30 - Nastavení kanálů pro ovládání myčky na nádobí	72
Obrázek 31 - Sitemapa pro ovládání myčky nádobí.....	74
Obrázek 32 - Sitemapa pro zobrazování ceny elektřiny	77
Obrázek 33 - Generování osobního tokenu	78
Obrázek 34 - Sitemapa pro ovládání a získání informací o bojleru.....	80

Obrázek 35 - Konfigurace Sony TV	81
Obrázek 36 - Sitemapa pro ovládání TV	82
Obrázek 37 - Registrační formulář pro openHAB cloud	83
Obrázek 38 - Konfigurace doplňku openHAB Cloud.....	84
Obrázek 39 - Vytvoření nového uživatele	84
Obrázek 40 - Sekce nastavení v mobilní aplikaci	85
Obrázek 41 – Notifikace – zařízení offline	87

SEZNAM TABULEK

Tabulka 1 - Standardy Wi-Fi	20
Tabulka 2 - Příklady prvků Items	26

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Application Programming Interface (Rozhraní pro programování aplikací)
ARCNET	Attached Resource Computer Network (Počítačová síť s připojenými prostředky)
ASCII	American Standard Code for Information Interchange (Americký standardní kód pro výměnu informací)
AV	Audio and Video (Audio a video)
BACnet	Building Automation and Control Network (Síť pro automatizaci a řízení budov)
CPU	Central Processing Unit (Centrální procesorová jednotka)
CSA	Connectivity Standards Alliance (Aliance pro standardy konektivity)
DDoS	Distributed Denial of Service (Distribuované odepření služby)
DPI	Dots per inch (Body na palec, rozlišení tisku)
EPL2	Eclipse Public License 2.0 (Veřejná licence Eclipse verze 2.0)
ESP	Espressif Platform (Platforma Espressif, např. ESP8266, ESP32)
GPIO	General Purpose Input Output (Univerzální vstupně-výstupní rozhraní)
GPS	Global Positioning System (Globální poziční systém)
HDR	High Dynamic Range (Vysoký dynamický rozsah)
HTTP	Hypertext Transfer Protocol (Protokol pro přenos hypertextu)
HVAC	Heating, Ventilation and Air Conditioning (Topení, ventilace a klimatizace)
IoT	Internet of Things (Internet věcí)
IP	Internet Protocol (Internetový protokol)
ISO	International Organization for Standardization (Mezinárodní organizace pro normalizaci)
JVM	Java Virtual Machine (Java virtuální stroj)
KNX	Konnex (Standard pro automatizaci budov)
LAN	Local Area Network (Místní síť)

MAP	Map Transformation (Překládová funkce v OpenHAB)
MPTT	Maximum Power Point Tracking (Sledování bodu maximálního výkonu)
MQTT	Message Queuing Telemetry Transport (Protokol pro zasílání zpráv v síti IoT)
MS/TP	Master-Slave/Token Passing (Protokol BACnet)
OTA	Over The Air (Bezdrátová aktualizace softwaru)
OSGi	Open Services Gateway initiative (Modulární architektura pro jazyk Java)
RAM	Random Access Memory (Operační paměť)
REST	Representational State Transfer (Architektura pro webové služby)
RPi	Raspberry Pi (Jednodeskový počítač)
RS-232	Recommended Standard 232 (Sériové rozhraní)
RS-485	Recommended Standard 485 (Sériové rozhraní)
SD	Secure Digital (Paměťová karta)
SSH	Secure Shell (Bezpečný protokol pro vzdálené připojení)
TCP	Transmission Control Protocol (Přenosový řídicí protokol)
Thread	Thread Network Protocol (Síťový protokol pro IoT)
TTF	TrueType Font (Typ písma)
UDP	User Datagram Protocol (Uživatelský datagramový protokol)
UPS	Uninterruptible Power Supply (Nepřerušitelný zdroj napájení)
URL	Uniform Resource Locator (Jednotný identifikátor adresy)
USB	Universal Serial Bus (Univerzální sériová sběrnice)
VPN	Virtual Private Network (Virtuální privátní síť)
Wi-Fi	Wireless Fidelity (Bezdrátová síťová technologie)
Zigbee	Zigbee Alliance Protocol (Bezdrátový mesh protokol pro IoT)
Z-Wave	Z-Wave Protocol (Bezdrátový protokol pro chytrou domácnost)
Wi-Fi	Wireless Fidelity (Bezdrátová věrnost)

SEZNAM PŘÍLOH

Příloha P I: Pravidlo pro zpracování JSON výstupu ze solární elektrárny

Příloha P II: Uživatelský widget pro informace o fotovoltaické elektrárně

Příloha P III: Pravidlo pro textové zobrazení stavu baterie

Příloha P IV: Konfigurační a zdrojové soubory projektu

PŘÍLOHA P I: PRAVIDLO PRO ZPRACOVÁNÍ JSON VÝSTUPU ZE SOLÁRNÍ ELEKTRÁRNY

```
rule "ZP JSON transform"
when
    Item ZP_JSON_Out changed
then
    val Pv = transform("JSONPATH", "$.ppv",
        ZP_JSON_Out.state.toString)
    val Battery = transform("JSONPATH", "$.pbattery1",
        ZP_JSON_Out.state.toString)
    val BatteryStatus = transform("JSONPATH", "$.battery_mode",
        ZP_JSON_Out.state.toString)
    val Soc = transform("JSONPATH", "$.battery_soc",
        ZP_JSON_Out.state.toString)
    val ActivePower = transform("JSONPATH", "$.active_power",
        ZP_JSON_Out.state.toString)
    val GridStatus = transform("JSONPATH", "$.grid_in_out",
        ZP_JSON_Out.state.toString)
    val Consumption = transform("JSONPATH", "$.house_consumption",
        ZP_JSON_Out.state.toString)
    val Eday = transform("JSONPATH", "$.e_day",
        ZP_JSON_Out.state.toString)
    val Etotal = transform("JSONPATH", "$.e_total",
        ZP_JSON_Out.state.toString)
    val EtotalExp = transform("JSONPATH", "$.meter_e_total_exp",
        ZP_JSON_Out.state.toString)
    val EtotalImp = transform("JSONPATH", "$.meter_e_total_imp",
        ZP_JSON_Out.state.toString)
    zpPv.postUpdate(Pv)
    zpBattery.postUpdate(Battery)
    zpBatteryStatus.postUpdate(BatteryStatus)
    zpSoc.postUpdate(Soc)
    zpActivePower.postUpdate(ActivePower)
    zpGridStatus.postUpdate(GridStatus)
    zpConsumption.postUpdate(Consumption)
    zpEday.postUpdate(Eday)
    zpEtotal.postUpdate(Etotal)
    zpEtotalExp.postUpdate(EtotalExp)
    zpEtotalImp.postUpdate(EtotalImp)
end
```

PŘÍLOHA P II: UŽIVATELSKÝ WIDGET PRO INFORMACE O FOTOVOLTAICKÉ ELEKTRÁRNĚ

```
uid: SMA_widget
tags: []
props:
parameters: []
parameterGroups: []
timestamp: Apr 23, 2025, 1:27:35 PM
component: f7-card
config:
title: Solární elektrárna
slots:
default:
  - component: oh-label-card
config:
title: Solární výkon
item: zpPv
icon: f7:sun_max
  - component: oh-label-card
config:
title: Baterie SOC
item: zpSoc
icon: f7:battery_100
  - component: oh-label-card
config:
title: Spotřeba domu
item: zpConsumption
icon: f7:house
  - component: oh-label-card
config:
title: Do sítě
item: zpActivePower
icon: f7:bolt
  - component: oh-label-card
config:
title: Dnešní výroba
item: zpEday
icon: f7:calendar
```

PŘÍLOHA P III: PRAVIDLO PRO TEXTOVÉ ZOBRAZENÍ STAVU BATERIE

```
rule "Update Battery Status"
when
    Item zpBatteryStatus changed
then
    if (zpBatteryStatus.state == NULL || zpBatteryStatus.state == UNDEF)
    {
        logWarn("BatteryRule", "zpBatteryStatus je NULL nebo UNDEF, pravidlo
        se neprovede.")
        zpBatteryStatusText.postUpdate("Neznámý stav")
        return
    }
    val status = zpBatteryStatus.state as Number
    switch(status.intValue) {
        case 1: {
            zpBatteryStatusText.postUpdate("Pohotovostní režim")
            logInfo("BatteryRule", "Baterie je v pohotovostním režimu")
        }
        case 2: {
            zpBatteryStatusText.postUpdate("Vybití")
            logInfo("BatteryRule", "Baterie se vybíjí")
        }
        case 3: {
            zpBatteryStatusText.postUpdate("Nabíjení")
            logInfo("BatteryRule", "Baterie se nabíjí")
        }
        default: {
            zpBatteryStatusText.postUpdate("Neznámý stav")
            logWarn("BatteryRule", "Neznámá hodnota zpBatteryStatus: " + status)
        }
    }
}
end
```

PŘÍLOHA P IV: KONFIGURAČNÍ A ZDROJOVÉ SOUBORY PROJEKTU

Zdrojové kódy a potřebné soubory byly nahrány do systému IS/STAG.