



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**NÁVRH CHYTRÉ DOMÁCNOSTI ZA POMOCI
OPEN-SOURCE IOT**

SMART HOME DESIGN USING AN OPEN SOURCE IOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. MARTIN TOMOVIČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN PLUSKAL

BRNO 2019

Zadání diplomové práce



21337

Student: **Tomovič Martin, Bc.**
Program: Informační technologie Obor: Počítačové sítě a komunikace
Název: **Návrh chytré domácnosti za pomoci open-source IoT**
Smart Home Design Using an Open Source IoT
Kategorie: Vestavěné systémy
Zadání:

1. Nastudujte komerční řešení chytré domácnosti a po konzultaci s vedoucím vyberte podmnožinu zařízení, která bude navržená domácnost obsahovat. Seznamte se s možnostmi otevřených platforem pro domácí automatizaci jako je např. Home Assistant, Domoticz, openHUB.
2. Proveďte specifikaci chování hypotetického uživatele systému, zejména jeho zvyklosti (denní rituály), které mohou být automatizací usnadněny. Navrhněte scénáře, které musí domácí automatizace vykonávat.
3. Navrhněte HW infrastrukturu chytré domácnosti a její integraci ve zvoleném automatizačním systému dle pokynů vedoucího.
4. Navrhněte způsob komunikace zajišťující robustnost a bezpečnost systému. Řešení musí být odolné vůči poruše libovolné komponenty v jeden čas tak, aby nebyl ovlivněn chod celého systému. Řešení musí být energeticky efektivní a vzdáleně přístupné.
5. Implementujte řešení na základě návrhů a jednotlivá dílčí rozhodnutí konzultujte s vedoucím.
6. Zhodnoťte výhody Vašeho řešení oproti alespoň jednomu komerčnímu řešení z bodu 1.

Literatura:

- Blum, J. (2013). *Exploring Arduino: tools and techniques for engineering wizardry*. John Wiley & Sons.
- Gill, K., Yang, S. H., Yao, F., & Lu, X. (2009). A zigbee-based home automation system. *IEEE Transactions on Consumer Electronics*, 55(2).
- Gomez, C., & Paradells, J. (2010). Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6).
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7), 1645-1660.

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Pluskal Jan, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 22. května 2019
Datum schválení: 30. října 2018

Abstrakt

Táto práca obsahuje popis technológií z oblasti internetu vecí. Konkrétne je zameraná na problematiku inteligentných domácností a popisuje aktuálne dostupné možnosti. Cieľom práce je navrhnúť inteligentnú domácnosť využívajúcu open-source technológie. Dôraz je kladený na bezpečnosť a vysokú dostupnosť. Riešenie je odolné voči poruche ľubovoľného komponentu, resp. výpadok ľubovoľného komponentu neovplyvní chod celého systému. Práca umožňuje zorientovať sa v aktuálnych možnostiach pre open-source inteligentnú domácnosť.

Abstract

Diploma thesis contains of a decription of Internet of things and its technologies. Primarily, it is focused on the issue of smart homes and presents currently available options. The ultimate objective is to design smart home using open-source technologies. The emphasis is placed on security and high accessibility. The solution is fault-tolerant for any component, f.e. the failure of any component does not affect the system running. Diploma thesis allows orientation in the current options for open-source smart home.

Kľúčové slová

inteligentná domácnosť, automatizácia domácnosti, internet vecí, Home Assistant, MQTT

Keywords

smart home, home automation, internet of things, Home Assistant, MQTT

Citácia

TOMOVIČ, Martin. *Návrh chytré domácnosti za pomoci open-source IoT*. Brno, 2019. Diplomová práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Návrh chytré domácnosti za pomoci open-source IoT

Prehlásenie

Čestne vyhlasujem, že diplomovú prácu som vypracoval sám s použitím zdrojov uvedených v zozname literatúry a pod vedením pána Ing. Jana Pluskala.

.....
Martin Tomovič
22. mája 2019

Podakovanie

Ďakujem svojmu vedúcemu Ing. Janovi Pluskalovi za cenné rady. Taktiež ďakujem snúbenici a rodine za podporu pri vytváraní práce.

Obsah

1	Úvod	3
2	Existujúce riešenia	4
2.1	Komunikačné protokoly	4
2.2	Platformy inteligentnej domácnosti	10
2.2.1	Open-source riešenia	13
3	Automatizácia a scenáre	18
3.1	Denné rutiny	18
3.1.1	Popis základných rutín	18
3.1.2	Návrh rutín	21
3.2	Návrh automatizácií	21
4	Návrh riešenia	22
4.1	Návrh HW infraštruktúry	22
4.1.1	Centrálny uzol	23
4.1.2	Uzly v miestnostiach	23
4.1.3	Hlasový asistent	25
4.1.4	Senzory	25
4.1.5	Aktory	26
4.2	Návrh SW infraštruktúry	27
4.2.1	Komunikačný protokol	27
4.2.2	Prostredie	28
4.2.3	Vzdialený prístup	30
5	Implementácia	32
5.1	HW infraštruktúra	32
5.1.1	Centrálny uzol	32
5.1.2	Uzly v miestnostiach	32
5.2	SW infraštruktúra	33
5.2.1	Prostredie	33
5.2.2	Správa inteligentnej domácnosti	33
5.2.3	Konfigurácia jednotlivých zariadení	35
5.2.4	Komunikácia	37
5.2.5	Vzdialený server	39
5.3	Automatizácie	40
5.3.1	Konfigurácia pravidiel	40
5.3.2	Implementované automatizácie	41

6 Záver	42
Literatúra	43
A Obsah CD	46
B Zoznam komponentov	47

Kapitola 1

Úvod

Automatizácia a robotizácia bežných úkonov sa v našej spoločnosti postupne stáva realitou. S pojmom inteligentná domácnosť sa už určite stretol takmer každý. Nové aj existujúce firmy sa snažia preraziť v tomto odvetví. V oblasti open-source má tento segment taktiež silné zázemie v podobe veľkého množstva nástrojov a komunity ľudí okolo nich. Táto práca ponúka prehľad existujúcich riešení v kapitole 2.2.

Technicky si za pojmom inteligentná domácnosť môžeme predstaviť vzájomne prepojené špecializované zariadenia, medzi ktoré patria senzory a aktory. Sensory sú zariadenia, ktoré merajú určité fyzikálne veličiny a sprostredkovávajú namerané informácie na ďalšie spracovanie. Aktory sú zariadenia, ktoré môžu vykonávať akcie, ako napr. rozsvietiť sa, začať kúriť. Prepojenie je realizované pomocou káblového spojenia, bezdrôtovo alebo kombináciou týchto prístupov.

Prepojené zariadenia môžu svoje namerané dáta alebo funkcionality sprístupniť do siete Internet, preto sa často stretávame s pojmom IoT (z angl. Internet of Things — internet vecí). Tento pojem vyjadruje sieť zariadení resp. „vecí“, ktoré môžu vzájomne interagovať a vymieňať si dáta. Problémom je, že spôsob komunikácie komponentov môže byť odlišný. Táto práca v kapitole 2.1 rozoberá jednotlivé komunikačné protokoly, ktoré sú štandardom v oblasti IoT. Zameriava sa najmä na štandardy pre siete PAN (z angl. Personal Area Network - osobné siete) a LAN (z angl. Local Area Network - lokálne siete), ktoré sa domácnosťami týkajú najviac.

Firmy pôsobiace v segmente inteligentných domácností ponúkajú úsporu energií, varovanie pri nepredvídateľných udalostiach, detekciu bezpečnostných hrozieb a iné výhody inštalácie inteligentnej domácnosti. Medzi najviac spomínané prednosti patrí pohodlie a komfort. Riešenia umožňujú ovládanie kúrenia, osvetlenia a rôznych zariadení, ako napríklad žalúzií pomocou mobilného telefónu alebo hlasových príkazov. Taktiež je možné vytvárať scenáre a pomocou nich automatizovať ovládanie zariadení na základe dát zo senzorov. Takéto scenáre sú rozobrané v kapitole 3. Sledovanie a ovládanie domácnosti je väčšinou možné aj mimo domu prostredníctvom internetu.

Cieľom práce je navrhnuť inteligentnú domácnosť so zameraním sa na robustnosť, bezpečnosť a energetickú efektívnosť. Tiež sa sústreďuje na odolnosť voči chybám, vďaka ktorej výpadok jednotlivých komponentov neovplyvní chod celého systému. Návrh riešenia sa nachádza v kapitole 4.

Návrh domácnosti je realizovaný a umožní inteligentné ovládanie domácnosti aj v praxi. Poznatky, ktoré boli zistené pri implementovaní sú spomenuté v kapitole 5.

Kapitola 2

Existujúce riešenia

Táto kapitola je zameraná na prieskum existujúcich možností a systémov na výstavbu inteligentnej domácnosti a tvorbu automatizácií. V prvej časti rozoberieme komunikačné protokoly, s ktorými sa môžeme v tejto oblasti stretnúť. Následne rozoberieme riešenia inteligentnej domácnosti, resp. nástroje na správu domácnosti, s ktorými sa bude konečný používateľ pri obsluhu stretávať najčastejšie.

2.1 Komunikačné protokoly

Namerané údaje alebo príkazy, ktoré používateľ zadá je nutné prenášať čo najviac efektívnym spôsobom. Na tento účel slúžia komunikačné protokoly, ktoré popisujú spôsob prenosu na rôznych vrstvách OSI modelu. Prvé zariadenia a senzory, ktoré sa dostávali do našich domácností disponovali tradičnými protokolmi Wi-Fi a Bluetooth a mnohí z nás si ich ešte vtedy nespájali s pojmami inteligentná domácnosť alebo internet vecí. Dnes už je táto oblasť pomerne dobre popísaná a vznikli aj protokoly špecializované práve pre tento segment. Vybrané protokoly [26] vhodné pre aplikáciu v tejto oblasti si rozoberieme v nasledujúcich sekciách. Najprv sa zameriame na protokoly popisujúce viaceré vrstvy OSI modelu, prakticky od fyzickej až po aplikačnú vrstvu (Zigbee, Z-wave). Ďalej popíšeme použiteľné protokoly linkovej vrstvy (Wi-Fi, Bluetooth) pre bezdrôtovú komunikáciu. Popíšeme aj aplikačné protokoly, ktoré sú často využívané práve v oblasti internetu vecí.

Zigbee

Technológia Zigbee je založená na štandardoch skupiny IEEE 802.15.4. Je vyvíjaná organizáciou Zigbee Alliance¹. Popisuje spôsob komunikácie na viacerých vrstvách OSI modelu (fyzická, linková, sieťová a aplikačná vrstva) a radí sa medzi technológie pre WPAN (z angl. Wireless Personal Area Networks - Bezdrôtová osobná sieť).

Zigbee je aktuálne veľmi rozšírené, najmä v priemyselnej oblasti. Medzi výhody patrí jednoduchosť implementácie (použiteľné aj na jednoduchých mikročipoch), robustnosť, možnosti poskytujúce nízku spotrebu energie, vysokú bezpečnosť a škálovateľnosť s veľkým počtom uzlov. Technológia operuje na frekvenciách ISM pásma okolo 2,4 GHz. Prenosová rýchlosť je na úrovni 250 kbps. Dosah je 10 až 100 metrov [26]. Štandard umožňuje pracovať aj v regionálnych ISM pásmach (915 Mhz a 868 Mhz). Výhodou je taktiež podpora technológie Mesh, ktorá umožňuje konektivitu aj so vzdialenejšími zariadeniami vďaka preposielaniu

¹Viac o Zigbee Alliance na <https://www.zigbee.org/> [navštívené 10.1.2019]

informácií cez zariadenia v dosahu. Štandard Zigbee 3.0 ponúka spätnú kompatibilitu so staršími štandardami [29].

Prieskumom produktov v obchodoch sme zistili, že mnoho obľúbených zariadení určených pre inteligentné domácnosti využíva túto technológiu.

Z-wave

Bezdrôtový protokol Z-wave je aktuálne vyvíjaný a špecifikovaný v spoločnosti Silicon Labs. Výrobcov, ktorí tento štandard využívajú, zastrešuje organizácia Z-Wave Alliance. Špecifikácia popisuje spôsob komunikácie na všetkých vrstvách. Nižšie vrstvy protokolu sú popísané v odporúčaní ITU-T G.9959². Komunikácia je realizovaná na ISM frekvenčných pásmach (915 Mhz a 868 Mhz).

Výhodou Z-Wave je, že komunikácia na frekvenčných pásmach pod 1 GHz nie je rušená rozšírenými technológiami Wi-Fi a Bluetooth. Nižšie frekvencie nie sú schopné dosiahnuť veľké prenosové rýchlosti. Z-Wave poskytuje nízku latenciu a dosahuje prenosovú rýchlosť maximálne 100 kbps, čo je pre prenos senzorických dát a správ medzi zariadeniami dostatočné. Dosah je 30 metrov, no vďaka Mesh sieti je možné komunikovať aj so vzdialenejšími uzlami pomocou preposielania dát cez bližšie uzly [26]. Sieť je škálovateľná a umožňuje spojenie 232 zariadení. Protokol je jednoduchý, čo umožňuje rýchlejší vývoj zariadení.

Bluetooth, BLE

Bluetooth je štandard bezdrôtovej komunikácie vyvíjaný neziskovou organizáciou Bluetooth SIG (Bluetooth Special Interest Group). Protokol je určený pre WPAN. V prvých verziách bol protokol aj súčasťou štandardu IEEE 802.15.1-2005.

Postupom času vznikalo viacero verzií tohto štandardu, ktoré sú spätne kompatibilné. Vo verzii Bluetooth 3.0 pribudla verzia protokolu s možnosťou vysokorýchlostného spojenia realizovaná na základe štandardu IEEE 802.11 (viď. 2.1). S verziou Bluetooth 4.0 vznikla technológia BLE - Bluetooth Low Energy, ktorá sa zameriava na nízke energetické nároky spojenia. Táto verzia obsahovala protokoly Classic Bluetooth (klasické), Bluetooth high speed (vysokorýchlostné) and Bluetooth Low Energy (s nízkou spotrebou energie). Technológie pracujú na frekvenciách z ISM pásma okolo 2,4GHz. Výhodou tohto štandardu je veľký počet zariadení, ktoré ho podporujú. V nových notebookoch, mobilných telefónoch, prenosných zariadeniach, ale i autách sa jeho podpora stáva samozrejmosťou. Najnovšia verzia štandardu je Bluetooth 5.

BLE je významným protokolom v oblasti IoT, avšak nie je určený na prenos veľkého množstva dát (prenosová rýchlosť 1Mbps resp. 2Mbps pre Bluetooth 5). Dosah je 50-150 metrov (Bluetooth 4.2). Štandard Bluetooth 4.2 priniesol pre BLE podporu 6LoWPAN. Ide o princíp definovaný v štandarde RFC6282, ktorý umožnil odosielanie IPv6 paketov, a tak sprístupnil komunikáciu s existujúcou IP infraštruktúrou, vďaka stanoveniu zapúzdrenia a mechanizmu kompresie hlavičky [26]. Štandard Bluetooth 5 priniesol techniky na zvýšenie dosahu na úkor rýchlosti a naopak.

Wi-Fi

Pri výpise možností bezdrôtovej komunikácie nesmieme zabudnúť na Wi-Fi. Jedná sa o viacero štandardov označovaných ako *IEEE 802.11* alebo *IEEE 802.x*. Jednotlivé štandardy

²Text odporúčania ITU-T G.9959 na <https://www.itu.int/rec/T-REC-G.9959-201501-I/en> [navštívené 20.1.2019]

sa líšia frekvenciou, šírkou pásma, metódou modulácie a ďalšími technickými požiadavkami, na základe ktorých sa líšia aj maximálne prenosové rýchlosti, dosah a iné vlastnosti. Z prieskumu obchodov sme zistili, že aktuálne najpredávanejšie sú komponenty pracujúce so štandardom *802.11ac* (*IEEE 802.11ac-2013*) a *802.11n* (celým názvom *IEEE 802.11n-2009*). Posledný spomínaný štandard dosahuje maximálnu prenosovú rýchlosť 600 Mbps (v praxi skôr nižšie rýchlosti okolo 150-200 Mbps v závislosti od počtu antén, kanálu a iných činiteľov). Dosah je približne 50 metrov [26].

Technológiu Wi-Fi využíva mnoho zariadení vrátane počítačov, mobilných telefónov, televízorov na pripojenie k internetu. V týchto zariadeniach je kladený dôraz na vysokú prenosovú rýchlosť. Mnohé komponenty v inteligentných domácnostiach nepotrebujú vysoké prenosové rýchlosti, keďže väčšinou prenášajú len senzorické dáta s veľkosťou pár bajtov. Vyššie prenosové rýchlosti sú potrebné najmä na prenos multimédií, preto je tento protokol vhodný na komunikáciu s kamerovým systémom a multimediálne centrá.

HTTP

HTTP (z angl. Hypertext Transfer Protocol - hypertextový prenosový protokol) je protokol spravovaný organizáciou W3C, aktuálne používaný HTTP/1.1 je definovaný v dokumente RFC2616³. Jedná sa o veľmi rozšírený protokol pracujúci s transportným protokolom TCP. Zariadenia sú adresované pomocou URI (z angl. Uniform Resource Identifier - jednotný identifikátor prostriedku). Má sadu metód, ktoré sú vhodné na tvorbu API (z angl. Application Programming Interface - rozhranie pre programovanie aplikácií), konkrétne RESTful API. Natívne nepodporuje žiadne QoS.

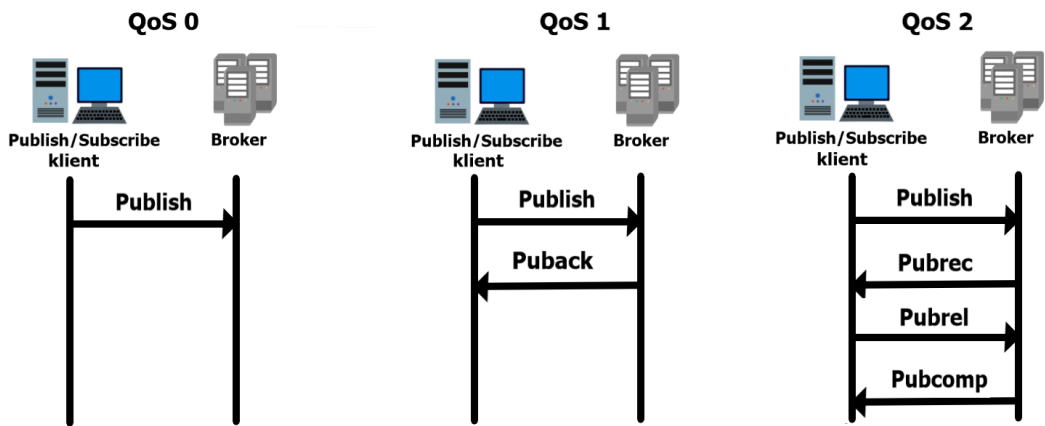
MQTT

MQTT (z angl. Message Queuing Telemetry Transport) bol navrhnutý vo firme IBM, má vlastný ISO štandard (ISO/IEC 20922:2016⁴). Nahradil kedysi používaný HTTP protokol. Jedná sa o protokol správ, určený pre *many to many* (viaceré viacerým) komunikáciu. Jeho výhodou je jeho jednoduchosť, implementácia podľa návrhového vzoru Publisher-Subscriber a narozdiel od jeho alternatívy CoAP, ktorá má dve úrovne QoS, MQTT poskytuje úrovne tri. Stupeň QoS je potvrdzovacia komunikácia medzi odosielateľom správy a jej príjemcom, ktorá definuje garanciu doručenia danej správy. V rámci MQTT existujú tri úrovne zobrazené na obrázku 2.1.

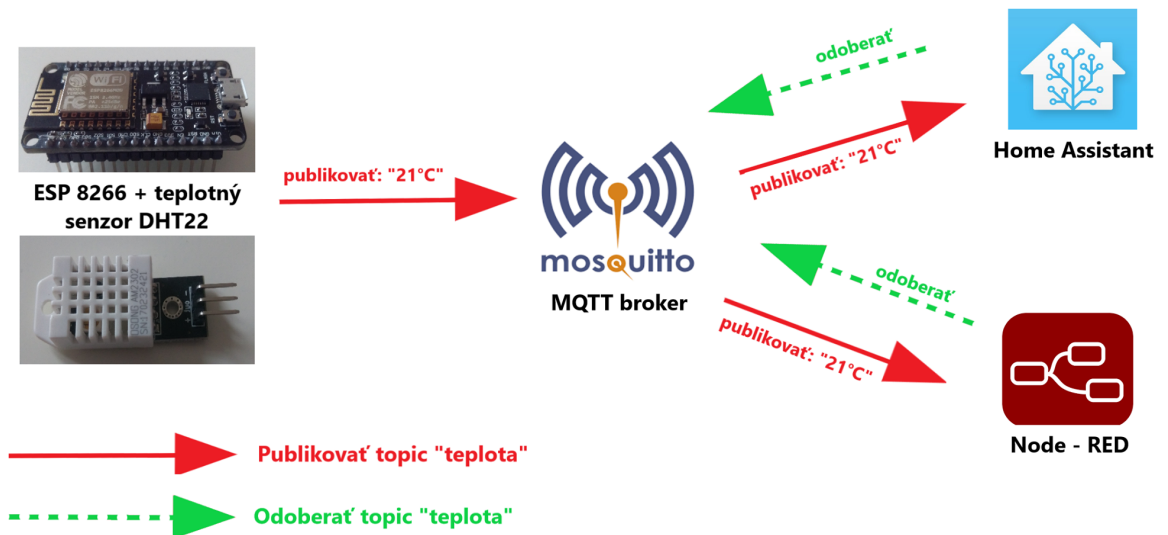
MQTT je ideálnym riešením pre IoT aplikácie, pretože umožňuje odsielanie príkazov do riadiacich výstupov, čítanie a publikovanie informácií zo snímacích uzlov a veľa ďalšieho. Nakoľko ide o komunikačný protokol, jeho primárnou úlohou je komunikácia medzi klientmi (jednotlivé zariadenia) prostredníctvom MQTT brokeru (centrálneho bodu). Tento protokol rozlišuje dve skupiny klientov: vydavateľa (angl. publisher) a odberateľa (angl. subscriber). Server spravuje štruktúru tém (angl. topic). Klienti môžu do daných tém odosielať správy (angl. publish) alebo dané témy odoberať (angl. subscribe). Vždy pri zmene správy v téme je rozoslaná PUBLISH správa klientom, ktorý odoberajú daný topic. Najčastejšie používaný port protokolu je 1883, pri TLS spojení 8883. Ako transportný protokol je využitý TCP, klienti udržiavajú s brokerom TCP spojenie, vďaka čomu je server schopný doručiť im správy v reálnom čase.

³RFC2616 - <https://www.ietf.org/rfc/rfc2616.txt> (navštívené 20.5.2019)

⁴ISO/IEC 20922:2016 - <https://www.iso.org/standard/69466.html> (navštívené 5.5.2019)



Obr. 2.1: QoS 0 - najviac raz (angl. at most once), QoS 1 - najmenej raz (angl. at least once), QoS 2 - práve jeden raz (angl. exactly once) (pôvodný autori obrázka: Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong, Hongtaek Ju)[18]



Obr. 2.2: Ukážka MQTT komunikácie

CoAP

Protokol CoAP (z angl. Constrained Application Protocol) je aplikačný protokol určený pre obmedzené siete (napr. siete s väčšou stratovosťou paketov alebo siete zamerané na nízku spotrebu energie), a teda je vhodný pre využitie v internete vecí. Tento protokol je štandardizovaný od roku 2014 ako *RFC 7252*⁵. Určený je pre takzvané M2M (z angl. machine to machine - zariadenie zariadeniu) aplikácie, je to spôsob komunikácie založený na princípe *one to one* (jeden jednému), no moderné M2M dokážu komunikovať aj nad rámec tohto princípu. Do značnej miery je inšpirovaný protokolom HTTP a poskytuje štandardné metódy, ktoré poznáme z REST prístupu (GET, POST, PUT, DELETE). Rozdiel je najmä v odľahčenej implementácii ideálnej pre mikrokontroléry s malou kapacitou pamäte RAM a ROM, zjednodušenej 4-bajtovej binárnej hlavičky paketov. Využíva prenosový protokol UDP. Prenášané správy môžu byť v ľubovoľnom formáte, napr. JSON alebo XML. Narozdiel od HTTP formát správy (z HTTP známy ako Content-type) nie je definovaný klasickým MIME (z angl. Multipurpose Internet Mail Extensions - viaceúčelové rozšírenie internetovej pošty) typom, ale CoAP zavádza vlastné označenie Content-Formats vopred definované 16-bitovým číselným kódom. Správy môžu byť posielané aj viacerým zariadeniam vďaka podpore IP broadcast a multicast. Výhodou je podpora automatického objavovania zdrojov (angl. CoAP resources discovery).

XMPP

XMPP (z angl. Extensible Messaging and Presence Protocol) je otvorený protokol pôvodne určený na IM (z angl. instant messaging - služba pre okamžité posielanie správ) na komunikáciu medzi používateľmi pomocou správ v reálnom čase s čo najmenším oneskorením. Vznikol pri vývoji aplikácie Jabber [28]. Je ľahko rozšíriteľný. Využíva sa aj oblasti internetu vecí. Využíva transportný protokol TCP, prípadne HTTP, keďže je povolený na väčšine firewallov. Obsah správy je vo formáte XML. Každé zariadenie má unikátnu adresu vo formáte „pouzivatel@domena.com“.

mDNS

Protokol multicast DNS (z angl. Domain Name System - systém názvov domén) má za úlohu v rámci počítačových sietí riešiť mená hostiteľov ku konkrétnym IP adresám v rámci malých sietí, ktoré neobsahujú lokálny názov serveru. Táto služba s nulovou konfiguráciou používa v podstate rovnaké programovacie rozhrania, formáty paketov a taktiež operačnú sémantiku ako unicast DNS. mDNS bol navrhnutý Stuartom Cheshirom, ktorého prvotná myšlienka bola, aby mDNS pracoval ako samostatný protokol, môže však pracovať v zhode so štandardnými DNS serverami. mDNS protokol je publikovaný ako RFC 6762 (dokument špecifikujúci, ako vykonávať DNS dotazy cez IP Multicast), používa pakety IP multicast User Datagram Protokol (UDP) a je implementovaný softvérovými balíkmi, ako napríklad Apple Bonjour alebo open-source Avahi. Taktiež bol implementovaný v Androide i vo Windows 10, kde však bolo jeho použitie obmedzené iba na objavovanie sieťových tlačiarň. mDNS môže pracovať v spojení s DNS Service Discovery (DNS-SD), čím rozširuje službu Multicast DNS tak, aby poskytovala jednoduché zisťovanie služieb, napr. prehliadanie siete; i so sprievodnou technikou s nulovou konfiguráciou, ktorá je samostatne vysvetlená a špecifikovaná v RFC 6763. Ak klient mDNS má za úlohu vyriešiť meno hostiteľa, pošle správu

⁵RFC 7252 - <https://tools.ietf.org/html/rfc7252> (navštívené 5.5.2019)

s IP multicast dotazom, ktorý požiada hostiteľa s týmto menom, aby sa identifikoval. Cieľové zariadenie, počítač rozosiela správu, ktorá obsahuje IP adresu tohto hostiteľa, potom môžu všetky ostatné počítače v sieti použiť tieto informácie na aktualizáciu vyrovnávacích pamätí mDNS. Akýkoľvek hostiteľ má právo sa zrieknuť svojho nároku na názov domény zaslaním paketu s odpoveďou s TTL (z angl. time to live) rovným nule.

Zhrnutie

V tabuľke 2.1 sú porovnané transportné protokoly opísané v tejto kapitole. Spomínané aplikačné protokoly okrem protokolu tu porovnané v tabuľke 2.2, jediný nespomenutý je menej rozšírený protokol XMPP. Protokol mDNS budeme využívať v súvislosti s objavovaním a prístupovaním k zariadeniam v lokálnej sieti.

	Bluetooth LE	Zigbee	Z-Wave	Wi-Fi
Frekvencia	2,4GHz	868MHz/915MHz/ 2,4GHz	868MHz/908MHz/ 2,4GHz	2,4GHz/5GHz
Modulácia	GFSK	OQPSK	GFSK	QPSK
Dosah	50-150 metrov	10-100 metrov	~30 metrov	~50 metrov
Prenosová rýchlosť	1Mbps/ 2Mbps(BT 5)	20kbps/40kbps/ 250kbps	9,6kbps/40kbps/ 100kbps	600Mbps(802.11n)/ 3.46Gbps(802.11ac)
Topológia	Mesh	Mesh	Mesh	hviezdicová
Spotreba energie	malá	malá	malá	veľká

Tabuľka 2.1: Porovnanie transportných protokolov

	HTTP	MQTT	CoAP
Architektúra	Klient/Server	Klient/Broker	Klient/Server Klient/Broker
Abstrakcia	Požiadavka/Odpoveď	Publikovať/Odoberať	Požiadavka/Odpoveď Publikovať/Odoberať
Veľkosť hlavičky	Nedefinovaná	2 B	4 B
Veľkosť správy	Veľká a nedefinovaná (závisí od web servera alebo programovacej technológie)	Malá a nedefinovaná (max. 256 MB)	Malá a nedefinovaná (bežne malá, aby sa vošla do jedného IP datagramu)
Metódy	Get, Post, Head, Put, Patch, Options, Connect, Delete	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Put, Delete
QoS/ Spôľahlivosť	Obmedzené (cez TCP)	QoS 0 - najviac raz (angl. at most once), QoS 1 - najmenej raz (angl. at least once) QoS 2 - práve jeden raz (angl. exactly once)	Potvrdivelná správa (približne ako QoS 0) alebo Nepotvrdivelná správa (približne ako QoS 1)
Štandardy	IETF a W3C	OASIS, Eclipse Foundations	IETF, Eclipse Foundation
Prenosový protokol	TCP	TCP, (MQTT-SN môže použiť UDP)	UDP, SCTP
Zabezpečenie	TLS/SSL	TLS/SSL	DTLS, IPSec
Port	80/443 (TLS/SSL)	1883/ 8883 (TLS/SSL)	5683 (UDP Port)/ 5684 (DLTS)
Kódovanie	Textové	Binárne	Binárne
Licenčný model	Free	Open-source	Open-source
Organizačná podpora	Global Web Protocol Standard	IBM, Facebook, Eurotech, Cisco, Red Hat, Software AG, Tibco, ITSO, M2Mi, Amazon, Web Services (AWS), InduSoft, Fiorano	Large Web Community Support, Cisco, Contiki, Erika, IoTivity

Tabuľka 2.2: Porovnanie vybraných aplikačných protokolov používaných v oblasti IoT [24].

2.2 Platformy inteligentnej domácnosti

V tejto sekcii sú rozoberané existujúce platformy a systémy, ktoré slúžia na konfiguráciu, ovládanie a automatizáciu inteligentnej domácnosti. Opíšeme nástroje z komerčnej sféry aj open-source alternatívy. Nie všetky spomenuté sú kompletné riešenia, najmä v oblasti open-source je prínosné kombinovať viacero nástrojov a služieb. Taktiež sa musíme problematikou zaoberať hlbšie, až na úroveň mikročipov a integrovaných obvodov, aby sme boli schopní implementovať inteligentnú domácnosť s čo najviac benefitami.

V rámci tejto práce sme skúmali taktiež existujúce komerčné riešenia, ktoré ponúkajú firmy zaoberajúce sa realizáciami inteligentných domácností. Zamerali sme sa na riešenia, ktoré sú dostupné na Slovensku [17].

Presný princíp fungovania nižšie spomínaných systémov je často obchodným tajomstvom a k presným technickým informáciám nie je možné získať prístup alebo do nich nahliadnuť. Sú určené aj pre menej technicky zdatných používateľov, ktorí požadujú spoľahlivé a cenovo výhodné riešenie. Nevýhodou týchto riešení inteligentnej domácnosti je viazanosť na určitú podmnožinu zariadení, ktoré daný výrobca ponúka. Používateľ je závislý na určitom dodávateľovi alebo dodávateľoch. Tento prístup sa nazýva *Vendor Lock-In* [25]. V súčasnosti existujú snahy o štandardizáciu určitých možností smarthome. Dokazuje to angažovanie firiem v organizáciách vyvíjajúcich štandardizované protokoly Z-wave a Zigbee. Firmy taktiež ponúkajú spôsoby, ako systém prepojiť so zariadeniami tretích strán, no často sú tieto postupy finančne náročné, komplikované a neumožňujú využívať kompletnú funkcionálnosť zariadení.

Výhodou týchto riešení je väčšinou dobrá podpora a jednoduchosť inštalácie. Systémy sú inštalované špecializovanými inštaláčnymi firmami, ktoré majú vyškolených technikov, ktorí si vedia poradiť s rôznymi problémami a možnosťami.

Loxone

Platforma Loxone je komplexným riešením, ktoré pochádza z Rakúska. Firma Loxone, ktorá bola založená v roku 2009, slúži na správu množstva zariadení a automatizáciu veľkého množstva činností vo vašej domácnosti. K dispozícii je káblová inštalácia pomocou zbernice a bezdrôtové riešenie. Platforma disponuje podrobnou dokumentáciou pre inštaláciu jednotlivých komponentov [20].

Základným komponentom takejto platformy je server - *Loxone Miniserver* pre káblovú inštaláciu a *Loxone Miniserver Go* pre bezdrôtovú. Firma deklaruje, že v týchto zariadeniach používa operačný systém Loxone OS, ktorý nie je žiadnou modifikáciou Linuxu či iného existujúceho operačného systému. Navrhli a vytvorili ho sami na mieru pre tento hardware, vďaka čomu ponúka stabilitu, výkon a maximálnu bezpečnosť [22]. Server je možné rozširovať pomocou *Extensions* komponentov, ktoré umožňujú zvýšiť počet pripojených zariadení, prípadne pripojiť zariadenia pracujúce s podporovanými komunikačnými technológiami. So serverom sú prepojené pomocou *Loxone Link* [20] zbernice. K dispozícii sú *Extensions* pre Loxone Air, DALI, DMX, Modbus, RS232, RS485, EnOcean, IR, 1-Wire. Topológia kabeľáže je *Loxone Tree* [20], ktorá umožňuje jednoduchú konfiguráciu a redukuje množstvo použitej kabeľáže. Bezdrôtová technológia sa nazýva *Loxone Air* [20] a umožňuje bezpečnú a úspornú (kvôli zariadeniam pracujúcim na batériu) komunikáciu. Pracuje na frekvenciách 865—924MHz, využíva bezlicenčné frekvenčné pásmo, v závislosti na oblasti, kde je systém v prevádzke.

Spoločnosť ponúka vlastné senzory a aktory podporujúce tieto technológie. Ich zoznam je veľmi obsiahly a obsahuje napr. meteostanicu, videovrátnika či okenné kľučky. Rovnako ponúkajú audiosystém, ktorý je plne integrovateľný do tejto platformy a je možné ho skombinovať napr. s osvetlením alebo bezpečnostným systémom.

Konfigurácia prebieha pomocou konfiguračnej aplikácie pre Windows, pričom počítač musí byť v rovnakej sieti ako server [23]. Na jednoduché ovládanie domu, bez väčších konfiguračných nastavení, slúži aplikácia, ktorá je dostupná pre iOS, Android aj Windows. Vzdialený prístup by tiež po správnej konfigurácii nemal byť problém. Niektoré softvérové služby sú platené nad rámec štandardu. Jedná sa o precízne údaje o počasí a telefonické informovanie o udalostiach [21].

Control4

Control4 je americká spoločnosť založená v roku 2003, ktorá pôsobí v 100 krajinách. Rovnomenný systém pre inteligentné domácnosti sa snaží integrovať čo najviac produktov tretích strán. CEO spoločnosti, Martin Plaehn povedal: „*Sme radi, že sú tu všetky tieto nové zariadenia. Môžeme využiť inováciu zvyšku sveta ako náš dodávateľský reťazec*“ [15]. Aktuálne podporujú 12500 produktov tretích strán [5].

Hlavným komponentom systému sú riadiace jednotky, ktoré sú k dispozícii vo viacerých variantoch, ktoré slúžia zároveň ako mediálne centrá. Spoločnosť ponúka aj kamery, audio a video prepínače, prídavné audio zosilňovače, elektronické zámky, rôzne typy senzorov a elektroinštalčné moduly, ako napr. relé, brány na integráciu s inými protokolmi (napr. DALI alebo KNX), input/output moduly. Väčšina komponentov od tejto firmy využíva na komunikáciu protokol ZigBee. Všetky komponenty inteligentnej domácnosti je možné ovládať vďaka jednému univerzálnemu nástroju. Týmto nástrojom môže byť dotykový panel, diaľkové ovládanie, konfigurovateľná klávesnica alebo aplikácia pre TV, smartfón, tablet alebo hodinky.

K dispozícii je aplikácia pre iOS, Android, Windows alebo Mac. Spoločnosť ponúka taktiež predplatenú službu *4Sight*, ktorá slúži na vzdialené ovládanie domácnosti.

iNELS

Toto kompletné riešenie vzniklo v Českej republike v roku 2007 ako produkt firmy *ELKO EP* [14]. Zameriava sa nielen na riešenia pre rodinné domy, ale aj väčšie budovy ako sú napríklad hotely. Táto firma ponúka dva typy inštalácií – bezdrôtová elektroinštalácia a zbernicová elektroinštalácia.

Bezdrôtová komunikácia prebieha pomocou protokolu *iNELS RF Control (RFIO)* na frekvenciách bezlicenčného frekvenčného pásma. Zariadenia sú prepojené pomocou dvojvodičovej zbernice *iNELS BUS* s voľnou topológiou, avšak podmienkou je, že nesmie byť uzavretý fyzický kruh. Firma *ELKO EP* ponúka veľké množstvo zariadení, ktoré slúžia na ovládanie a sledovanie spotreby, vykurovania, osvetlenia, zásuviek, kamier, meteostaníc a žalúzií. Okrem iného ponúka aj *LARA Radio* a *LARA Intercom*, ktoré umožňujú centrálnu prehrávanie hudby. Pomocou zariadenia Connection server je možné komunikovať so zariadeniami tretích strán vďaka vybraným protokolom. iNELS je kompatibilný so systémom na riadenie budov *Niagara*⁶, vďaka ktorému vieme zabezpečiť komunikáciu s veľkým množstvom ďalších zariadení.

Systém je možné ovládať pomocou mobilnej aplikácie *iNELS Home Control*, fyzických ovládačov, kľúčenky alebo iNELS Touch panel (dotykový ovládací panel). Mobilná aplikácia využíva cloudové riešenie ELKO Cloud, ktoré umožňuje komunikovať so zariadeniami aj mimo lokálnej siete.

Fibaro

Fibaro je systém založený a vyvíjaný od roku 2010 v Poľsku. Systém je kompatibilný s mnohými zariadeniami tretích strán [9].

Základným komponentom je *Fibaro Home Center*. Pripájajú sa naň kamery, senzory a aktory, z ktorých zbiera informácie a analyzuje ich. Taktiež umožňuje ich ovládanie a automatizácie. *Fibaro Home Center* poskytuje Lua API a REST API pre potreby vývojárov. Komponenty vyvíjané touto firmou sú založené na bezdrôtovom Z-wave protokole [10].

⁶Viac o systéme *Niagara* na <https://www.tridium.com/> (navštívené 29.12.2018)

Firma taktiež ponúka verzie sensorov, ktoré komunikujú pomocou protokolu Apple HomeKit [12]. Medzi komponenty patrí napr. spínaná zásuvka s meraním spotreby alebo modul na ovládanie žalúzií.

Automatizácie je možné vytvárať pomocou scén. Základný postup vytvárania scén je založený na spúšťačoch a akciách. Používateľ definuje spúšťač a akciu, ktorú má vyvolať. Pokročilejšou možnosťou je vytvárať scény pomocou blokových schém. Systém umožňuje programovanie najpokročilejších scén vďaka programovaciemu jazyku Lua [11].

Na ovládanie slúži webová aplikácia prístupná v lokálnej sieti. Tiež je možné ovládať domácnosť na diaľku vďaka cloudovej službe. K dispozícii majú používatelia i aplikácie pre operačný systém Android, iOS. Firma takisto ponúka inštalačné aplikácie určené pre inštalačné firmy na spomínané operačné systémy. Na svojich webových stránkach deklarujú dôraz na zabezpečenie.

Ostatné

V tejto sekcii spomenieme riešenia, ktoré nie sú až tak komplexné ako riešenie spomenuté vyššie. Ponúkajú len určitú množinu zariadení alebo inteligentné inštalácie nie sú ich hlavným predmetom podnikania. Tieto riešenia sú často využívané v kombinácii s inými platformami.

Philips Hue je súhrnný názov pre inteligentné svietidlá od firmy *Philips*. V ponuke je množstvo svietidiel, ale aj vypínače a pohybové senzory na ich ovládanie. Pre väčšinu funkcionality je potrebné zakúpiť *Philips Hue Bridge*, ktorý prepája viaceré zariadenia a taktiež slúži na prepojenie s ďalšími systémami pre domácu automatizáciu. Zariadenia komunikujú vďaka protokolu ZigBee.

IKEA taktiež ponúka svoje riešenie. Výrobky sa predávajú ako séria *TRÅDFRI*. Ponuka sa týka najmä osvetlenia, no v ponuke je aj pohybový senzor alebo vypínateľná zásuvka. K dispozícii je aj aplikácia, ktorá funguje pri prepojení zariadení na prenosovú bránu. Na komunikáciu sa využíva protokol ZigBee.

Xiaomi Smart Home zahŕňa niekoľko zariadení ako inteligentné svietidlá, žiarovky *Xiaomi Yeelight*, kamery, senzory teploty a vlhkosti, pohybu, otvorenia dverí resp. okien a iné. Väčšina z nich využíva ako bránu na komunikáciu zariadenie *Xiaomi Gateway*. Po spárovaní je možné zariadenia sledovať a ovládať cez mobilnú aplikáciu. Komunikácia je založená na protokole ZigBee, no napr. zariadenie Xiaomi Mijia na sledovanie teploty a vlhkosti komunikuje pomocou protokolu Bluetooth LE.

2.2.1 Open-source riešenia

V tejto kapitole sú rozoberané open-source nástroje na automatizáciu domácnosti podľa [2]. Výhodou sú integrácie množstva technológií, služieb, protokolov a zariadení, ktoré vytvára komunita ľudí, ktorá okolo týchto riešení vznikla. Každý môže integrovať zariadenie, ktoré doma má a chce ho sprístupniť na ovládanie do open-source platformy. Prieskumom sme zistili, že na pochopenie niektorých zariadení je potrebné využiť dokonca metódy reverzného inžinierstva. Zdrojové kódy, ktoré pri tom vzniknú, ľudia často zdieľajú, aby mohli ich integráciu použiť aj iní používatelia. Rozvinuté platformy majú dokumentované jednotlivé moduly a časti zdrojového kódu na nižšej úrovni, kým komerčné dokumentácie sú väčšinou iba na úrovni návodov na inštaláciu konfiguráciu.

Používatelia, ktorí chcú inteligentné technológie vo svojich domácnostiach sa často obávajú bezpečnostných rizík. Možnosť vidieť a modifikovať zdrojový kód programov, ktoré sa

používajú na domácu automatizáciu môže byť pre nich vhodný spôsob ako kontrolovať, čo všetko sa s dátami o ich domácnosti deje [2].

Calaos

Automatizačný softvér Calaos vznikol v rovnomennej firme vo Francúzku. Keď spoločnosť ukončila svoju činnosť, prešiel tento softvér na GPLv3 licenciu. Okolo projektu vznikla menšia komunita. Podľa počtu prispievateľov a počtu commitov v repositároch tohto projektu je aktivita súviaca s ďalším vývojom veľmi nízka. Časti dokumentácie a väčšina komunikácie na fóre priaznivcov tohto systému sú vo francúzštine.

Projekt je tvorený súborom viacerých aplikácií. Hlavnou časťou je multiplatformový Calaos Server, ktorý je jadrom tohto riešenia. Je napísaný prevažne v programovacom jazyku C++. Ako ďalšiu kľúčovú časť projektu môžeme označiť „touchscreen“ rozhranie aplikácie, ktoré slúži na konfiguráciu a taktiež sa zobrazenie prehľadu domácnosti, ktorá je logicky rozčlenená na miestnosti. Ďalšími časťami tohto riešenia je webová a mobilná aplikácia. Softvérové riešenie zahŕňa taktiež multimedialny prehrávač a monitorovací kamerový systém, do ktorého sa dajú pridať všetky kamery poskytujúce MPEG stream. Zaujímavosťou je vlastná linuxová distribúcia nazvaná Calaos OS s predkonfigurovaným systémom.

Systém natívne využíva protokol WAGO PLC na komunikáciu so zariadeniami. Je stavaný na riadiacu jednotku od firmy Wago, ktorá umožňuje pripojenie senzorov, zariadení a pod. Odporúčaná riadiaca jednotka je Wago 750-841, prípadne Wago 750-849, ktorá s použitím prídavného modulu môže slúžiť ako KNX/IP smerovač. S použitím ďalších modulov je možné komunikovať so zariadeniami aj pomocou iných protokolov ako DALI, DMX atď. Tieto riadiace jednotky sú stavané pre stabilné riešenia pre potreby priemyslu, energetiky, čo sa odráža aj na ich cene. Oficiálne zatiaľ neexistuje podpora protokolov správ ako MQTT alebo ZeroMQ.

OpenHAB

Open-source platforma OpenHAB (z angl. Open Home Automation Bus) je vyvíjaná primárne v programovacom jazyku Java. Jej licencia je Eclipse Public License v 2.0. Táto platforma je založená na Eclipse SmartHome Framework a používa modulárny systém založený na OSGi na premostenie rôznych komponentov. Za touto platformou stojí veľká komunita ľudí na portáli Github.

Väčšina prispievajúcich vývojárov sa venuje tvorbe nových doplnkov, ktoré najčastejšie slúžia na interakciu so zariadeniami alebo službami (Bindings), no môžu byť chápané aj ako iné rozšírenia tejto platformy napr. vzhľad užívateľského rozhrania, hlasové syntetizátory. Väčšina častí platformy je konfigurovateľná pomocou konfiguračných súborov, ktoré majú vlastnú definovanú syntax. Pravidlá pre automatizáciu sa definujú pomocou jazyka založenom na Xbase. Pre zvýrazňovanie syntaxe konfiguračných súborov môžeme využiť rozšírenia do známych textových editorov. Okrem toho je OpenHAB konfigurovateľný pomocou viacerých konfiguračných nástrojov. Každý nástroj je vhodný na niečo iné. Výhodou je kvalitná dokumentácia, ktorá pomôže nielen s konfiguráciou systému. OpenHAB má viacero voliteľných používateľských rozhraní a možnosti konfigurácie pomocou nich sú pri každom iné. Štruktúra rozhrania je definované pomocou súborov sitemap, ktoré slúžia na logické usporiadanie jednotlivých ovládacích a monitorovacích prvkov. Do platformy je taktiež možné pristupovať pomocou REST API. Pre Android, iOS aj Windows je dostupná aplikácia. Aplikácie vedú spolupracovať so službou OpenHAB Cloud, ktorý zabezpečuje vzdialený prístup

pomocou tunelovania do lokálnej siete. Taktiež je k dispozícii operačný systém openHABian pre RaspberryPi s predkonfigurovanou platformou.

Home Assistant

Home Assistant je platforma určená pre domácu automatizáciu napísaná v jazyku Python a licencovaná pod licenciou Apache 2.0. Komunita pracujúca na vývoji je pomerne veľká, čo dokazuje aktivita v repositároch projektu a taktiež frekvencia vydávania nových releasov. Platforma umožňuje pracovať s takmer sto rôznymi IoT protokolmi pomocou komponentov nazývaných huby. Platforma tiež ponúka integrácie množstva aplikácií, či už pre prácu s multimédiami, kamerami alebo notifikáciami.

Konfigurácia tohto systému a aj jednotlivých komponentov je užívateľsky prívetivá. Je riešená pomocou konfiguračných súborov vo formáte YAML⁷. Tento formát je ľahko čitateľný bežným človekom. Pomocou spomenutých konfiguračných súborov si taktiež môže používateľ zdefinovať automatizácie obsahujúce spúšťače, podmienky a akcie. Editor automatizácií, ktorý je súčasťou grafického rozhrania, môže ešte viac zjednodušiť tvorbu pravidiel, no taktiež vyžaduje znalosť názvov používaných entít. Veľkou výhodou je kvalitná dokumentácia, ktorá je k dispozícii takmer pre každý komponent.

Grafické rozhranie aplikácie je tvorené pomocou widgetov, ktoré predstavujú jednotlivé komponenty. Mobilná aplikácia je k dispozícii iba pre operačný systém iOS, no responzívny front-end webovej aplikácie umožňuje plnohodnotné využívanie aj na mobilných zariadeniach. Pre vývojárov sú dostupné API (REST, WebSockets, Server sent). Tvorcovia ponúkajú i operačný systém Hassbian vytvorený pre Raspberry Pi. Umožňuje rýchlu a jednoduchú inštaláciu Home Assistant.

Jadro tejto platformy je tvorené štyrmi časťami (obr. 4.3). Hlavnou časťou je *Event Bus* (zbernica udalostí), ktorá dovoľuje komponentom spúšťať a počúvať rôzne udalosti.

Domoticz

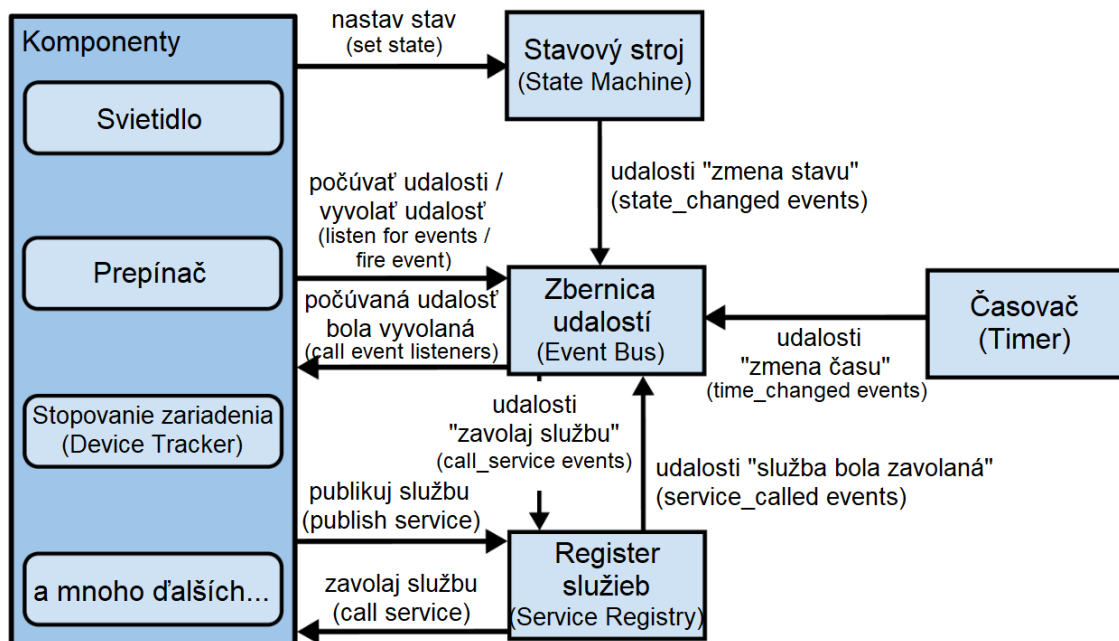
Domoticz je systém na domácu automatizáciu, ktorý umožňuje monitorovanie a ovládanie mnohých zariadení. Je vyvíjaný pod open-source licenciou GNUv3 v programovacom jazyku C++. Okolo projektu sa rozvinula komunita ľudí na čele s používateľom “Gizmo-cuz”, ktorý pracuje na projekte takmer denne. Podpora vývoja je založená na dobrovoľných príspevkoch.

Domoticz podporuje množstvo protokolov a zariadení. Podporované zariadenia je možné rozšíriť pomocou pluginov. Kompletný zoznam zariadení a taktiež dokumentáciu je možné nájsť na Domoticz wiki. Pre používateľov je tiež k dispozícii manuál a diskusné fórum. Nevýhodou je absencia dokumentácie pre vývojárov. Väčšina systému je konfigurovateľná priamo z grafického rozhrania aplikácie, prípadne je potrebné sa držať dostupných návodov. Tieto postupy nie sú unifikované a líšia sa pre každú funkčnosť alebo plugin.

Grafické rozhranie je rozdelené na nástroje, nastavenia, zoznam spínačov, scén, teplotných senzorov, sekcie počasia a nástenku, ktorá obsahuje obľúbené položky. Zariadenia je možné spájať do skupín a ovládať ich naraz. Automatizačná logika (v Domoticz nazývaná Skripty) je programovateľná pomocou programovacieho jazyka Lua alebo nástroja Blockly, ktorý slúži na užívateľsky prívetivé a vizuálne programovanie.

Systém je dostupný pre operačné systémy Windows, Linux aj macOS. Taktiež existujú postupy, ako systém spustiť na vybraných NAS serveroch. Domoticz má tiež k dispozícii

⁷YAML - Ain't Markup Language, viac na <https://yaml.org/>



Obr. 2.3: Architektúra jadra Home Assistant - *Zbernica udalostí* umožňuje spúšťať a počúvať udalosti. *Stavový stroj* sleduje stav komponentov a informuje o ich zmene na zbernicu udalostí. *Register udalostí* spracováva udalosti „zavolaj službu“ zo zbernice udalostí a volá služby v jednotlivých komponentoch. Ako príklad služby si môžeme predstaviť príkaz na zapnutie (`turn_on`), ktorý je k dispozícii v komponente svetidla (`Light`). *Časovač* odosiela na zbernicu udalostí informáciu o zmene času každú sekundu. (pôvodný autor obrázka: Paulus Schoutsen)[13]

aplikácie pre Android, iOS, Tizen (konkrétne pre hodinky Samsung Gear S2 alebo S3). Na mobilných zariadeniach je možné prepojiť Domoticz aj s viacerými aplikáciami, ktoré nie sú vyvíjané priamo komunitou okolo tohto systému. Systém má k dispozícii aj českú lokalizáciu.

OpenMotics

Tento systém na automatizáciu domu zahŕňa open-source hardvér aj softvér. Hardvér je licencovaný licenciou Creative Commons Attribution-ShareAlike 4.0 International License a softvér licenciou GPLv2. Pôvodne sa jednalo o komerčné riešenie, no jeho autori ho celé otvorili komunite ako open-source.

Celé riešenie je zamerané hlavne na hardvér. Systém pozostáva z modulov, ktoré sa najčastejšie inštalujú do rozvádzačov. Sú prispôbené uchyteniu na DIN lištu. Každý z modulov slúži na iný účel (Gateway module, Input module, Output module, Temperature module atď.). Moduly sú prepojené pomocou sériovej zbernice RS-485. Jadrom systému je Gateway modul, ktorý obsahuje mikropočítač so softvérom na automatizáciu.

Back-end systému je písaný v programovacom jazyku Python. Firmware jednotlivých modulov bol pôvodne programovaný v PicBasic Pro, no postupne je prepisovaný do C, keďže prvý spínaný jazyk nie je dostupný pod voľnou licenciou.

MisterHouse

MisterHouse je program na automatizáciu domácnosti vyvíjaný pod licenciou GPLv2 v programovacom jazyku Perl.

Umožňuje vytvárať pravidlá na interakciu zariadení a senzorov. Taktiež pravidlá založené na čase a hlasových príkazoch. Program je konfigurovateľný pomocou webového rozhrania a mobilnej aplikácie pre operačný systém Android.

Veľa funkcionality, nových zariadení a protokolov je dostupnej vďaka návodom v dokumentácii, no tu už je potrebná určitá zručnosť a v mnohých prípadoch aj znalosť programovacieho jazyka Perl. Dokumentácia je obširná a miestami neaktuálna. Taktiež je roztrúsená po viacerých portáloch.

Zhrnutie

	Home Assistant	OpenHAB	Domoticz	MisterHouse
Programovací jazyk	Python3	Java	C++	Perl
Frontend	HTML, iOS	HTML, Android, iOS, Windows	HTML	NA
API	Áno	Áno	Áno	Nie
Úložisko dát	SQLite	Prezistentné služby	SQLite	NA
Automatizácia	Pravidlá, skripty	Pravidlá, skripty	Skripty	Perl kód
Platforma	Linux, Windows, Max OS X	akékoľvek zariadenie s JVM	Linux, Windows, Max OS X	Linux, Windows, Max OS X
Dokumentácia	Dobrá	Dobrá	Slabá	Slabá

Tabuľka 2.3: Porovnanie open-source nástrojov na správu inteligentnej domácnosti [3].

Kapitola 3

Automatizácia a scenáre

Táto kapitola sa bude venovať štúdiu denných rituálov, ktoré nám môže automatizácia domácnosti zjednodušiť. Existuje široká škála vecí, ktoré môžeme automatizovať alebo ovládať hlasom. V prvej sekcii si rozoberieme základné denné rutiny, ktoré ponúkajú hlasový asistenti ako Google Home či Amazon Alexa.

3.1 Denné rutiny

Google Assistant má predpripravených nasledujúcich 6 rutín, ktoré sa dajú vyvolať hlasovým príkazom. Rutiny je možné editovať prípadne pridávať vlastné, čo môže byť prínosné ak je asistent prepojený s inteligentnou infraštruktúrou domácnosti. Zoznam základných rutín[4]:

- Good Morning (Dobré ráno)
- Leaving Home (Odchod z domu)
- Commuting to Work (Cesta do práce)
- Commuting Home (Cesta domov)
- I'm Home (Som doma)
- Bedtime (Spánok)

3.1.1 Popis základných rutín

Dobré ráno

Rutina určená na vykonanie úkonov ráno po prebudení.

Prispôsobenie osvetlenia a zapnutie spotrebičov: Automatizácia zabezpečí rozsvietenie svetiel, napríklad na schodisku alebo v kúpeľni. Vykoná sa zapnutie zariadení alebo spotrebičov pomocou inteligentných zásuviek (hriankovač, kávovar a pod.).

Nastavenie termostatu: Táto funkcia je veľmi užitočná či už pre úsporu elektrickej energie alebo pre komfort používateľa. Počas spánku môže byť teplota v miestnosti nižšia a ráno ju môže systém automaticky nastaviť na príjemnú úroveň. Automatizovaným otváraním a zatváraním hlavíc radiátorov zabezpečí ranné zvýšenie teploty.

Predpoveď počasia: Hlasový asistenti môžu používateľovi oznámiť aké bude počas dňa počasie, ten sa následne vhodne obliešť alebo sa pripraviť napr. na zhoršenú premávku, keď

má celý deň snežiť. Asistenti vedia vyčítať aktuálne počasie aj z niektorých inteligentných meteorologických staníc v domácnosti.

Správy: Hlasový asistenti sú schopní na požiadanie používateľa prečítať aktuality a titulky z prednastavených portálov.

Môj deň: Používateľ môže použiť príkaz „Ok, Google, tell me about my day.“ hocikedy počas dňa. Túto funkciu využijeme najmä ráno. Používateľ takto získa prehľad o svojom dni: predpoveď počasia, udalosti z kalendára (prvé stretnutie, prehliadka u lekára a iné).

Hudba: Na hlasových asistentoch je samozrejmosťou prehrávanie hudby z obľúbených streamovacích služieb.

Odchod z domu

Pri odchode do školy alebo práce môže byť táto rutina využitá na prepnutie celého domu do režimu úspory energie.

Odomknutie dverí: Ak domácnosť disponuje inteligentným zámkom dverí, môže odomknutie dverí uľahčiť používateľovi odchod z domu. Nemusí hľadať kľúče a otvárať manuálne zámky.

Zhasnutie svetiel: Či už hovoríme o Google Assistant alebo Alexe, obe platformy umožňujú pomocou inteligentných žiaroviek vypínanie svetiel hlasom. Taktiež by bolo vhodné pri odchode z domu posledného obyvateľa automaticky zhasnúť všetky svietidlá.

Otvorenie garáže: Garážové brány často disponujú bezdrôtovými ovládačmi, alebo dokonca technológiami kompatibilnými s platformami na domácu automatizáciu. Otvorenie garážových dverí môže byť takisto veľmi osočné.

Prispôsobenie spotrebičov: Automatické spustenie rôznych spotrebičov vrátane vysávača, môže byť veľmi prínosné. Vysávač nebude nikoho vyrušovať, keď bude dom prázdny. Taktiež je možné automaticky spustiť niektoré typy pračiek a umývačiek, prípadne realizovať spustenie cez spínanú zásuvku.

Nastavenie termostatu: Keď sa v domácnosti nikto nenachádza, taktiež môžeme znížiť teplotu v miestnostiach a tým dosiahnuť úsporu energií.

Cesta do práce

Nakoľko používateľ môže už sedieť v aute byť na ceste do práce, pomocou smartfónu môže vykonať pokyny v rámci rutiny „Cesta do práce“.

Zamknutie dverí: Rutinná funkcia „Odchod z domu“ bola určená na otváranie dverí. Teraz rutina „Cesta do práce“ môže zabezpečiť, aby boli dvere zamknuté a aby bol bezpečnostný systém aktivovaný. Pri nečakanom pohybe v dome, rozbití okna a pod. môže byť vygenerované varovná hláška, zapnutá siréna. Používateľ môže byť informovaný o zabudnutých otvorených oknách.

Informácie o premávke: Inteligentný asistenti môžu používateľovi dať informácie o premávke alebo nehodách, resp. vyhľadať rýchlejšiu cestu do cieľa.

Hudba, audioknihy alebo podcasty: Zariadenie je schopné na požiadanie zahrať najnovšiu verziu playlistu, najnovšiu epizódu obľúbeného podcastu alebo audioknihu podľa výberu.

Prečítanie emailov: Obe platformy, Google Assistant a Alexa, umožňujú čítanie textov či emailov.

Cesta domov

Takisto ako rutina „Cesta do práce“ aj táto rutina nie je veľmi o dome ako takom, ale o interakcii so smartfónom.

Informácie o premávke: Po nastavení domácich a pracovných adries vo vašom systéme, Alexa alebo Google Assistant môžu upozorniť na rôzne problémy v premávke, ktoré by mohli ovplyvniť príchod používateľa domov.

Príprava domu: Umožní automatizovať zapnutie vybraných svetiel (napríklad pred domom alebo v garáži), pripraviť teplotu v dome pre maximálny komfort, vypnúť bezpečnostný systém alebo otvoriť garáž a to všetko na základe požiadaviek používateľa.

„Som na ceste“: Je možné prepojenie so všetkými hlasovými zariadeniami, cez ktoré môže byť iný člen domácnosti notifikovaný, že používateľ je na ceste domov.

Pripomienky: Zariadenie s hlasovým asistentom vie poskytnúť a pripomenúť rôzne informácie, ako napríklad nákupný zoznam.

Som doma

Táto rutina sa zaoberá momentom, keď používateľ dorazí domov. Aj v tomto prípade je možné používateľovi jednoducho spríjemniť deň a pripraviť všetko potrebné.

Zapnutie svetiel: Zapnutie svetiel sa môže stať veľmi užitočnou funkciou pre používateľa, najmä ak v zimných mesiacoch prichádza za tmy, ale pracuje dlho do noci. Zapínanie svetiel v konkrétnych miestnostiach si môže nastaviť podľa seba a svojho pohybu pri príchode domov, podľa toho, do ktorej miestnosti vchádza najskôr, napríklad najprv chodba a potom kuchyňa.

Hudba: Ako už sme spomenuli hlasový asistent môže spustiť obľúbenú hudbu. Taktiež je možné nastavenie zapnutia televízie, pomocou spínanej zásuvky alebo priamo, ak televízia podporuje integráciu so systémom domácej automatizácie. Pomocou technológií ako napr. Chromecast je možné hlasovým povelením spustiť obľúbený seriál alebo tvorcu na videoporáli *YouTube*.

Pripomenky: Pripomienky založené na polohe sú ďalšou výhodou nastavovania rutiny „Som doma“, asistent nám pripomenie práce, ktoré sme chceli doma vykonať.

Spánok

Poslednou rutinou, ktorá ukončuje deň je „Spánok“. S nastavením tejto rutiny používateľ môže automatizovať mnohé úkony, čím ušetrí čas, ale i energie.

Nastavenie termostatu: Nastavenie teploty v miestnosti na teplotu príjemnú pre spánok.

Zhasnutie svetiel: Je možné automaticky zhasnúť všetky svietidlá v dome, keď ide spať posledný obyvateľ domácnosti.

Nočný chod spotrebičov: Pomocou inteligentného asistenta je možné nastaviť napríklad umývačku riadu, či práčku na vykonanie svojich úloh v noci.

Alarm: Je možné aktivovať bezpečnostný systém, ktorý môže detekovať bezpečnostné hrozby v čase spánku a v prípade ohrozenia nás o tom notifikovať.

Polievanie: Systém riadenia zavlažovania môže pomôcť pri šetrení vody. Rastliny ako aj trávu, treba polievať v noci, aby sa cez horúci deň nespálila studenou vodou. Počas noci sa tak isto voda hneď neodparí, ale má čas na vsiaknutie do pôdy.

3.1.2 Návrh rutín

Idem variť

Ak členovia domácnosti radi a často varia, stačí spustiť rutinu príkazom „“OK Google, I am going to cook”

Nastavenie termostatu: Zníženie teploty v kuchyni zabezpečí, že prebytočné teplo vzniknuté pri varení nespôbí nepríjemné dusné prostredie.

Prisposobenie hlasitosti: Keď sa Google Home nachádza ďalej od miesta varenia je vhodné prednastaviť ideálnu hlasitosť.

Zapnutie svetiel: Automaticky sa rozsvietia svietidlá v blízkosti kuchynskej linky.

Náhodný recept: Hlasový asistent povie používateľovi náhodný recept, prechádza si ho s sním po jednotlivých ingredienciách a krokoch.

Hudba: Spustenie obľúbených piesní, playlistov alebo rádia.

Ukončenie: Bude potrebné vytvoriť zvlášť rutinu na ukončenie aby sa nastavenia domácnosti vrátili do pôvodného stavu. Prípadne môžeme vytvoriť automatizáciu mimo rutín asistenta, v inom nástroji.

Cvičenie

Táto rutina zabezpečí prípravu prostredia vhodného na cvičenie.

Nastavenie termostatu: Zníženie teploty, keďže sa používatelia zahrejú pohybom.

Hudba: Spustenie obľúbených piesní, playlistov alebo rádia.

Ukončenie: Bude potrebné vytvoriť zvlášť rutinu na ukončenie aby sa termostat vrátil do pôvodného stavu. Google assistant zatiaľ nepodporuje oneskorenie v rutinách, ak táto funkcia pribudne bolo by dobré vrátenie do pôvodného stavu napr. po 30 minútach.

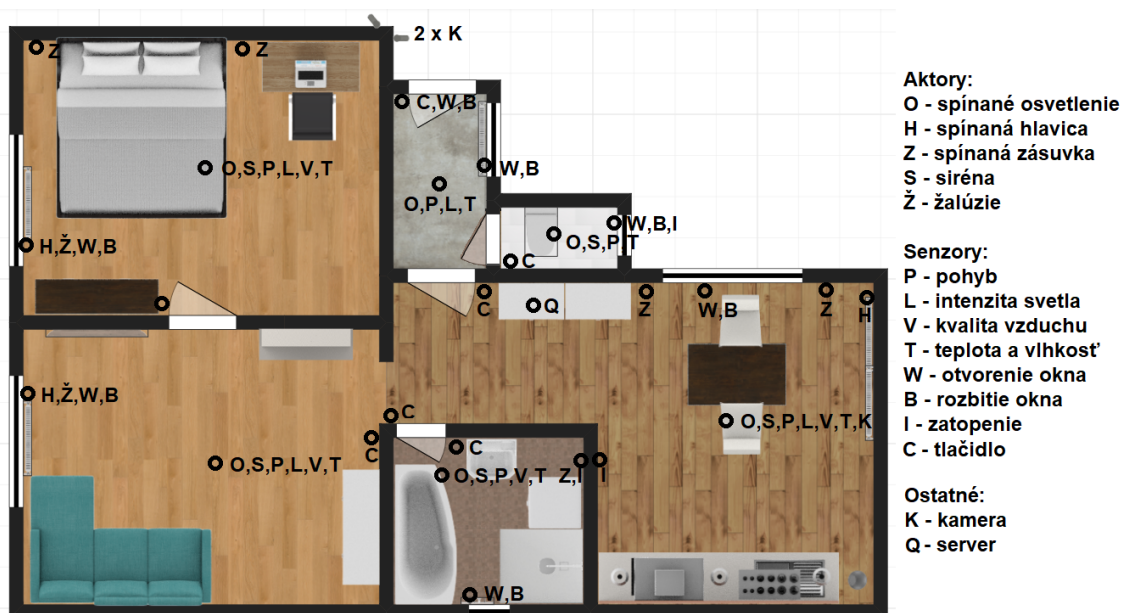
3.2 Návrh automatizácií

V rámci práce budú vypracované základné pravidlá automatizácie, ktoré budú zjednodušovať bežné úkony aj bez potreby hlasového asistenta. Bude pripravené automatické ovládanie osvetlenia a kúrenia.

Kapitola 4

Návrh riešenia

V tejto kapitole je prezentovaný návrh riešenia pre vybranú ukázkovú inteligentnú domácnosť. Jedná sa o dvoj-izbový byt v rodinnom dome. Práca sa zameriava na riešenie v interiéri, v exteriéri budú realizované iba kamery na monitorovanie. Pôdorys domu je znázornený na obrázku 4.1.



Obr. 4.1: Pôdorys ukázkovej domácnosti s predbežným návrhom umiestnenia senzorov a aktorov. V pravej časti obrázku sa nachádza legenda.

4.1 Návrh HW infraštruktúry

V nasledujúcej časti popíšeme použitie jednotlivých hardvérových komponentov v našom systéme. Komponenty sme vybrali kvôli dobrej komunite, ktorá sa okolo nich vytvorila, čoho následkom je, že vzniklo množstvo open-source nástrojov, ktoré sú pre komponenty určené. Pri výbere komponentov zohrala úlohu aj dostupnosť, cena, kvalita dokumentácie. Počet vstupných a výstupných rozhraní vrátane podporovaných zberníc je taktiež dôležitý z dôvodu rozšíriteľnosti o nové zariadenia do budúcnosti.

4.1.1 Centrálny uzol

Centrálny uzol - lokálny server bude v riešení použitý ako MQTT broker a zároveň na ňom bude spustená vybraná open-source platforma na automatizáciu. Taktiež na ňom bude realizovaný zber dát zo senzorov do databázy. Ako operačný systém bude použitá open-source linuxová distribúcia. V prípade, že bude nutné využiť rôzne podporné programy a skripty, napr. na spracovávanie obrazu z kamier, taktiež budú spustené na serveri. Trvalé napájanie servera bude zabezpečené záložným zdrojom.

Jednotlivé softvérové súčasti budú fungovať ako vzájomne komunikujúce mikroslužby, budú spustené v Docker kontajneroch, čo umožní lepšiu škálovateľnosť a logické prepojenie. Z dôvodu vysokej dostupnosti bude server realizovaný ako klaster viacerých nezávislých počítačov. V prípade výpadku bude logika navrhnutá tak, aby spracovanie príkazov a dát prevzali funkčné počítače z klasteru.

Raspberry Pi

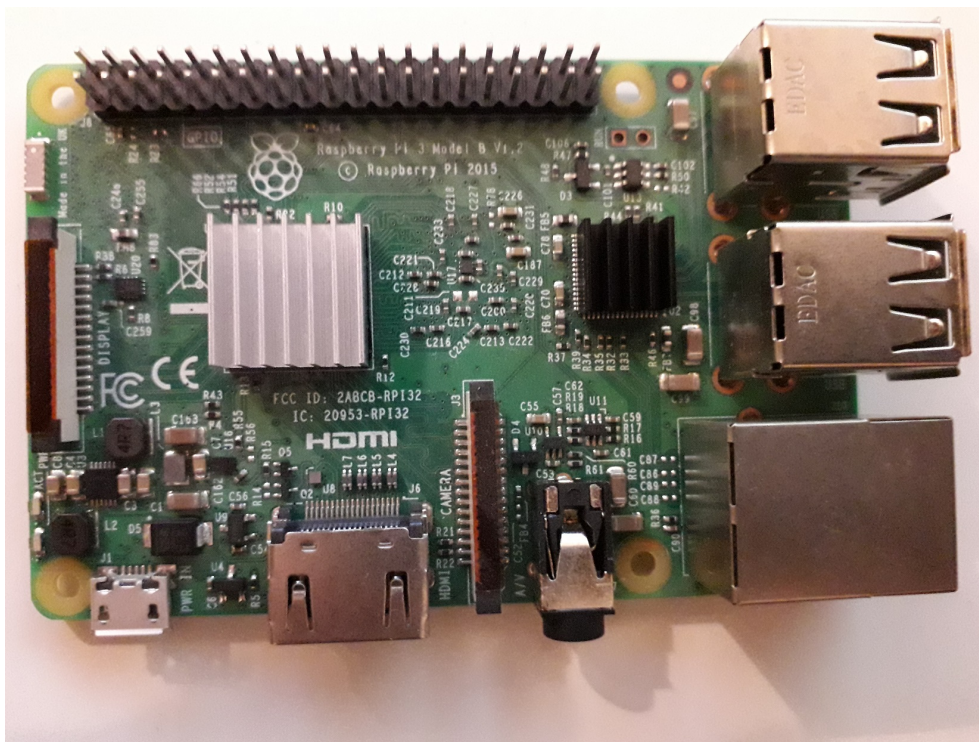
V riešení využijeme jednodoskové počítače Raspberry Pi 3 Model B. Tieto počítače boli vybrané z dôvodu cenovej dostupnosti, nízkej spotreby a taktiež veľkej komunity ľudí, ktorá sa okolo Raspberry Pi vytvorila. Komunita a dostupnosť materiálov zjednoduší riešenie prípadných problémov. Základné parametre:

- Quad-core 1.2 GHz Broadcom BCM2837 64 bit CPU
- 1GB RAM
- slot na micro SD
- napájacie napätie 5 V, operačná teplota -0°C - 50°C
- Wi-Fi 802.11 b/g/n
- Bluetooth 4.1
- 10/100 Ethernet
- 27 × GPIO (3,3 V logika)
- 4 × USB 2.0, UART, I2C, 2 × SPI
- video výstupy HDMI 1.3, composite video, DSI, vstup pre kamery CSI-2

Keďže chceme vytvoriť klaster z týchto zariadení, potrebujeme ich zaobstarať viacero kusov. Jednotlivé počítače budú prepojené pomocou sieťového prepínača. Nevýhodou je použitie micro SD karty ako systémového úložiska, preto sa musíme pri výbere karty zamerať na rýchlosť, ale aj počet cyklov čítania a zápisu. Alternatívnym riešením môže byť pripojenie externého pamäťového média pomocou USB portu a beh systému na ňom.

4.1.2 Uzly v miestnostiach

Jednotlivé uzly v miestnostiach budú realizované pomocou mikrokontrolérov. Budú pri riešení využité ako MQTT klienti a so serverom budú komunikovať pomocou Wi-Fi. Jeden klient bude spracovávať dáta z viacerých senzorov, prípadne ovládať viacero aktorov. Pre



Obr. 4.2: Raspberry Pi 3 Model B.

ulahčenie programovania, správy napájania, testovania a podobne, budú využité vývojové dosky.

Dôležitým parametrom je počet GPIO (vstupných a výstupných) pinov. Ak bude potrebné rozšíriť počet pinov, využijeme integrovaný obvod MCP23017, ktorý pridáva 16 ďalších digitálnych I/O pinov, pripája sa pomocou I²C zbernice. Taktiež je dôležitá čo najnižšia spotreba mikrokontroléra, hlavne pri riešeníach napájaných z batérie. Pri výbere mikrokontroléra zohľadníme tiež cenovú dostupnosť resp. najmä dobrý pomer ceny a parametrov.

ESP32

ESP32 je čip od firmy *Espressif*, ktorý je vhodný na použitie v domácej automatizácii a inteligentných budovách. Vyznačuje sa nízkou spotrebou, najmä pri využití podtaktovania procesora a viacerých možností spánku. Disponuje tiež koprocesorom s nízkou spotrebou (150 μ A v móde so zapnutým koprocesorom), ktorý umožňuje monitorovať stavy senzorov, periférií a časovačov [8]. Podrobnejšie informácie možno nájsť na stránke výrobcu¹. Základné parametre [8]:

- Tensilica Xtensa® 32-bit LX6 microprocessor, dual-core 160 MHz (max. 240 MHz), 600 MIPS
- 520 KB SRAM, 448 KB ROM
- operačné napätie 2.3 V - 3.6 V, operačná teplota -40°C - 125°C
- Wi-Fi 802.11 b/g/n (HT40 - schopný pracovať s 40 Mhz širokým pásmom)

¹Webová stránka *Espressif* na <https://www.espressif.com/> (navštívené 20.1.2019)

- Bluetooth 4.2
- $34 \times$ GPIO (3,3 V logika)
- $4 \times$ SPI, $2 \times$ I²C, $2 \times$ I²S, $3 \times$ UART
- $2 \times$ DAC , 18-kanálový 12-bitový ADC (digitálno-analógový a analógovo-digitálny prevodník)
- $10 \times$ kapacitný dotykový senzor
- OTA (bezdrôtové nahrávanie firmvéru)

ESP8266

Starší čip firmy *Espressif* ESP8266 je taktiež vhodný na použitie v domácej automatizácii a inteligentných budovách [7]. Tento čip sa vyznačuje nižšou cenou ako vyššie spomínaný ESP32, no napriek tomu môže byť pre množstvo projektov dostačujúci. Využíva sa aj v niektorých komerčných produktoch ako napr. *Sonoff POW R2*. Podrobnejšie informácie možno nájsť na stránke výrobcu¹. Základné parametre:

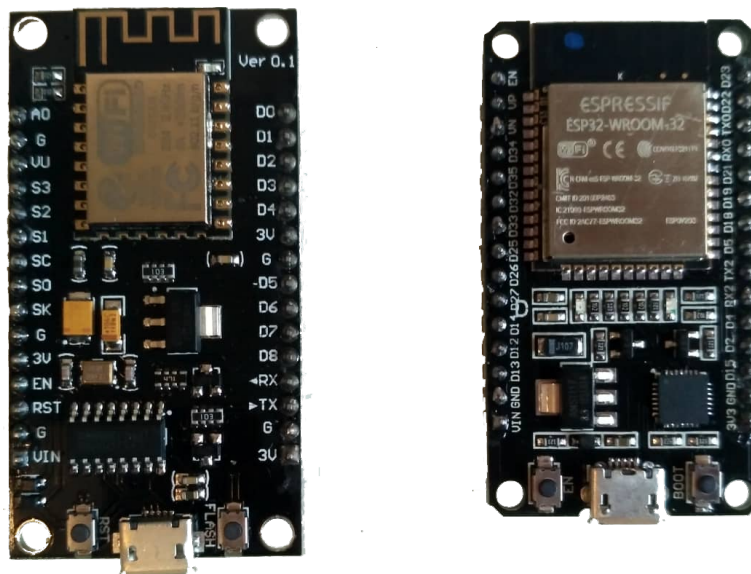
- Tenslica Xtensa® 32-bit L106 microprocessor, single-core 80 MHz (max. 160 MHz)
- 520 KB SRAM, 448 KB ROM
- operačné napätie 2.3 V - 3.6 V, operačná teplota -40°C - 125°C
- Wi-Fi 802.11 b/g/n (HT20 - schopný pracovať s 20 Mhz širokým pásmom)
- $17 \times$ GPIO (3,3 V logika)
- $2 \times$ SPI, $1 \times$ I²C, $2 \times$ I²S, $2 \times$ UART
- 10-bitový ADC (analógovo-digitálny prevodník)
- OTA (bezdrôtové nahrávanie firmvéru)

4.1.3 Hlasový asistent

V riešení bude využitý inteligentný reproduktor Google Home Mini s hlasovým asistentom. Bude slúžiť na hlasové ovládanie niektorých aktorov. Aktuálne tento asistent nepodporuje slovenský jazyk, avšak bolo avizované, že Google Assistant bude podporovať český jazyk [27]. Mohol by sa teda tiež dostať do zariadenia Google Home Mini.

4.1.4 Senzory

Senzory budú slúžiť na zaistenie bezpečnosti, zdravia a komfortu obyvateľov domácnosti. Pohybový senzor, senzory rozbitia a otvorenia okien a dverí budú slúžiť na detekciu hrozieb spojených s neželaným vniknutím. Senzor zatopenia a senzory kvality ovzdušia budú monitorovať nepredvídateľné udalosti. V domácnosti sa nachádzajú plynové spotrebiče (sporák, kotol) a taktiež krb na pevné palivo, preto bude nutné sledovať prítomnosť dymu a plynu v ovzduší. Z pohľadu zdravia budú senzory na kvalitu ovzdušia merať prítomnosť plynov,



Obr. 4.3: Vývojové dosky s čipmi ESP8266 (vľavo) a ESP32 (vpravo)

napr. oxidu uhličitého, ktorý môžu ovplyvňovať výkonnosť, spôsobovať únavu. Monitorovanie vlhkosti taktiež môže pomôcť udržiavať pre zdravie optimálnu hodnotu. Pri zaznamenaní nízkej vlhkosti, bude možné aktivovať spínanú zásuvku so zvlhčovačom vzduchu. Sensory teploty budú slúžiť na ovládanie vykurovania.

Tlačidlá

Návrh elektroinštalácie počíta s tlačidlovými spínačmi namiesto vypínačov. Tieto spínače možno pripojiť ako mechanické tlačidlá na vývojové moduly, detekovať ich stlačenie a následne spracovať túto informáciu.

4.1.5 Aktory

V tejto časti spomenieme podrobnejšie niektoré typy aktorov, ktoré budú v riešení využité. Okrem nižšie spomenutých, môžeme medzi aktory zaradiť aj sirény, ktoré budú informovať o bezpečnostných hrozbách a nepredvídateľných udalostiach. Návrh počíta aj s prípravou na inteligentné ovládanie žalúzií.

Relé

Spínanie zásuviek a osvetlenia bude realizované pomocou relé modulu, ktorý umožňuje spínanie vyšších napätí a prúdov pomocou dátových output pinov. Na spínanie osvetlenia je použitý modul s maximálnym zaťažením 250 V, 10 A na jedno relé. Svetelné okruhy sú istené pomocou 6 A ističov. Zásuvkové obvody sú tradične istené 16 A ističmi, preto je nutné použiť iný modul, aby sme sa vyhli poškodeniu modulu pri väčšom odbere prúdu. Alternatívou môže byť použitie 10 A ističa, no tu už môžeme byť obmedzovaní, ak by sme chceli na spínanú zásuvku zapojiť spotrebiče s väčším odberom ako napr. vykurovacie telesá alebo mobilnú klimatizáciu. Na spínanie zásuviek teda bude použitý modul s relé, ktoré umožňujú maximálne zaťaženie 250 V 30 A.

Elektrotermické hlavice

Elektrotermická hlavica je zariadenie umožňujúce otvárať a zatvárať prívod teplej vody do radiátora alebo podlahového kúrenia. Hlavice disponujú štandardizovaným pripojením M30 x 1,5, ktoré nájdeme takmer na všetkých ventiloch v radiátoroch alebo rozdeľovačoch. Pri kúpe si treba dať pozor na výber správneho typu. Prvý typ označovaný NC (z angl. normally closed - normálne zatvorený) značí, že hlavica bez prívodu prúdu uzatvára ventil. Typ NO (z angl. normally open - normálne otvorený) bez prívodu prúdu necháva ventil otvorený. Typu treba prispôbiť programovú logiku. Ak chceme, aby hlavica zmenila stav, musíme jej dodať elektrický prúd. Štandardne sa hlavice predávajú stavané na napätie 24 V a 230 V. Nie je ich možné ovládať priamo z dátového pinu modulu, preto na ovládanie bude nutné použiť relé.

4.2 Návrh SW infraštruktúry

V tejto sekcii sa budeme venovať zvoleným softvérovým nástrojom, ktoré použijeme pri implementácii nášho riešenia. Najprv rozoberieme komunikačné protokoly, ktoré budú využité. Potom popíšeme nástroje, ktoré budú poskytovať infraštruktúru pre beh služieb poskytujúcich správu a automatizáciu inteligentnej domácnosti. Budeme využívať niektorú z open-source platforiem spomenutých v sekcii 2.2.1.

4.2.1 Komunikačný protokol

Sekcia obsahuje popis protokolov využitých v práci a odôvodnení, prečo sme dané protokoly zvolili. Najznámejšie protokoly využívané v oblasti internetu vecí a najmä inteligentnej domácnosti sme opísali v kapitole 2.1.

Linková vrstva

V našom riešení sme sa rozhodli majoritne používať protokol Wi-Fi. Zvolili sme ho najmä kvôli dobrej dostupnosti podporovaného hardvéru. Konkrétne budeme využívať najmä zariadenia podporujúce protokol 802.11n. Keďže sa v našom návrhu zaoberáme rodinným domom, ktorý nie je v husto zaľudnenej oblasti, nemusíme sa obávať slabého signálu kvôli rušeniu. Šírka pásma by mala byť dostačujúca. Pri husto zaľudnených oblastiach môže vzniknúť problém, kedy veľký počet dostupných Wi-Fi vysielačov pracujúcich na rovnakom frekvenčnom rozsahu spôsobí nízky dosah a kvalitu signálu. Protokol sa vyznačuje vyššou spotrebou energie, takže nie je vhodný na zariadenia napájané z batérie. Aby sme mohli k nášmu riešeniu pripojiť zariadenia pracujúce na batériu, bude v našom systéme pripravená Z-wave/Zigbee brána. Ide o najrozšírenejšie protokoly v oblasti IoT používané na kratšie vzdialenosti. Za spomenutie stojí taktiež Bluetooth Low Energy, ktorý sa tiež vyznačuje nízkou spotrebou energie. RaspberryPi Model 3B, prípadne mikropočítač ESP32, ktorý dokáže komunikovať so zariadeniami pracujúcimi na tomto protokole bez nutnosti prídavného zariadenia, ak to bude potrebné, bude možné využiť aj tento spôsob komunikácie.

Aplikačná vrstva

V riešení bude primárne použitý aplikačný protokol MQTT, bude nutné pripraviť MQTT broker na centrálnom uzle a vybrať vhodné nástroje, ktoré tento protokol podporujú. Keďže sa jedná o veľmi rozšírený protokol v oblasti open-source nebude to problém. Ak niektoré

komponenty podporujú integráciu s vybranou platformou pomocou iného protokolu, bude im umožnené sa s ňou spojiť aj inou formou. Príkladom môže byť spojenie s Google Home alebo inými komponentami, ktoré majú dostupné HTTP API. Bude potrebné umožniť prvkom pracujúcim s niektorým z protokolov na automatické objavovanie zariadení, aby sa mohli s platformou spojiť automaticky. Tento prístup na nazýva Plug and Play (pripoj a hraj).

4.2.2 Prostredie

V tejto sekcii popíšeme nástroje, ktoré nám umožnia pripraviť infraštruktúru tak, aby bola dostupná aj v prípade výpadku niektorého z komponentov. Jednotlivé časti systému budú fungovať ako nezávislé mikroslužby izolované v kontajneroch, čo uľahčí riešenie problémov za behu bez znefunkčnenia celého systému.

Docker

Docker je open-source projekt, ktorý bol vyvinutý za účelom uľahčovania vytvárania, zavádzania a spúšťania aplikácií pomocou tzv. kontajnerov v prostredí Linux a Windows. Tieto kontajnery umožňujú vývojárovi zostaviť aplikáciu so všetkými potrebnými časťami, ako napríklad knižnice a iné závislosti, a odosielať všetko ako jeden balík. Vďaka kontajnerom sa autor môže uistiť, že aplikácia bude fungovať a bežať na akomkoľvek počítači so systémom Linux (alebo Windows) bez ohľadu na prispôsobené nastavenia počítača, ktorý sa odlišuje od toho, na ktorom bol písaný a testovaný kód. Jedny z najväčších výhod používania Dockeru sú flexibilita, menšia veľkosť, čo umožňuje menšie náklady na prevádzku.

Svojím spôsobom Docker je niečo ako virtuálny stroj. Narodiel od neho však nevytvára virtuálny operačný systém, ale umožňuje aplikáciám využívať rovnaké jadro Linuxu ako na systéme, na ktorom sú spustené a vyžaduje, aby aplikácie boli dodané s príslušenstvom, ktoré ešte nie je spustené na hostiteľskom počítači. Týmto spôsobom je docielené zreteľné zvýšenie výkonu a zníženie veľkosti.

Ako bolo už spomenuté, Docker je open-source projekt, čo znamená, že ktokoľvek sa môže podieľať na prispievaní do Dockera a jeho rozširovaní tak, aby vyhovoval potrebám používateľov, v prípade ak potrebujú ďalšie funkcie, ktoré základný Docker neposkytuje.

V prvom rade, Docker je nástroj, ktorý bol navrhnutý tak, aby bol prínosom nielen pre vývojárov, ale aj pre administrátorov systému, čo z neho robí súčasť reťazca nástrojov DevOps (angl. developers and operations). Pre vývojárov umožňuje písanie kódu bez obavy o konečný systém, na ktorom to bude bežať. Taktiež im to umožňuje byť v predstihu pomocou jedného z mnohých už navrhnutých programov, aby bežali v Docker kontajneri ako súčasť ich aplikácie.

Docker síce prináša značnú známku bezpečnosti pre aplikácie, ktoré bežia v zdieľanom prostredí, avšak kontajnery ako také, nie sú vhodné na prijímanie bezpečnostných opatrení. Dash Walsh, vedúci počítačovej bezpečnosti, tiež známy pre svoju prácu na SELinux, dáva do pozornosti dôležitosť zabezpečenia Dockeru i kontajnerov.

Docker Swarm

Docker Swarm je pomenovanie označujúce plánovací a klastrovací (z angl. cluster - zoskupujúci) nástroj pre Docker kontajnery. Spolu so Swarm môžu IT administrátori a vývojári založiť a spravovať skupinu uzlov – samostatných počítačov ako jeden celok, jeden virtuálny systém.

Súčasná verzia Dockera taktiež zahŕňa Swarm režim slúžiaci na natívne riadenie klastra Docker Engines. Docker Swarm je obohatený o niekoľko funkcií, ktoré prinášajú rôzne pozitíva a výhody, nimi napríklad sú: Správa klastrov (skupín) integrovaná s Docker Engine: Swarm Docker Engine je možné vytvoriť použitím nástroja Docker Engine CLI (z angl. Command-line interface - rozhranie príkazového riadku). Do toho Swarmu je možné nasaďiť aplikačné služby. Nie je potrebné mať ďalší doplnkový orchestračný softvér na správu či vytvorenie tohoto Swarmu.

Decentralizovaný návrh: Docker Engine dokáže zvládnuť hocikakú špecializáciu za behu namiesto toho, aby musel zvládať rozdiely medzi úlohami uzlov. Použitím Docker Engine, je možné nasadenie oboch druhov uzlov, manažérov (angl. manager) a pracovníkov (angl. worker). V konečnom dôsledku to znamená, že je vývojárovi sprístupnené vytvoriť celý Swarm len z jedného obrazu disku (angl. single disk image).

Deklaratívny model služby: Docker Engine požíva deklaratívny prístup, aby definoval požadovaný stav rôznych služieb v aplikačnom zásobníku (angl. application stack). Mierka: Pre každú službu je možné deklarovať počet úloh, ktoré majú byť spustené. Keď sa počet pohybuje hore alebo dole, Swarm správca je pripravený sa automaticky prispôbiť pridaním alebo odstránením úloh, aby požadovaný stav bol zachovaný.

Zosúladenie požadovaného stavu: Swarm manažér stále monitoruje stav klastra (angl. cluster) a zosúladuje rozdiely medzi reálnym stavom a stavom požadovaným od používateľa. Multi-host networking: Manažér Swarmu automaticky priradí adresy ku kontajnerom v sieti prekrývaných údajov, keď iniciuje alebo aktualizuje aplikáciu. Objavovanie služby: Uzly spravujúce Swarm priradujú každej službe v rámci Swarmu jedinečný DNS názov a zariadenia, ktoré bežia v kontajeroch. Je možné sa dotazovať o každý kontajner bežiaci vo Swarme cez DNS server vložený do Swarmu.

Vyvažovanie záťaže: Porty pre služby môžu byť vystavené externému vyrovnávačovi záťaže. Swarm nechá používateľa upresniť spôsob distribúcie servisných kontajnerov medzi uzlami. Štandardné zabezpečenie: Každý uzol vo Swarme vynúti vzájomnú autentifikáciu a šifrovanie TLS, aby zabezpečil komunikáciu medzi sebou a ostatnými uzlami. Aktualizácie: Po spustení je možné postupne aktualizovať služby na uzloch. Swarm manažér umožňuje kontrolu oneskorenia medzi nasadením služieb do rôznych sád uzlov. V prípade poruchy je možné vrátiť úlohu a na pôvodnú verziu.

Gluster FS

GlusterFS je zdarma dostupný, škálovateľný súborový systém, ktorý pomocou bežného hardvéru môže vytvoriť veľké a distribuované úložné riešenia pre streamovanie médií, analýzu údajov a ďalšie úlohy náročné na údaje a šírku pásma. GlusterFS štandardne používa protokol TCP a agreguje rôzne úložné severy cez Ethernet alebo Infiniband RDMA prepája do jedného veľkého paralelného sieťového súborového systému. Ide o voľne a zadarmo šíriteľný softvér, ktorého niektoré časti sú licencované Heunder the GNU General Public License (GPL) v3, zatiaľ čo iné sú duálne licencované buď GPL v2 alebo licenciou Lesser General Public License (LGPL) v3.

Základnou stavebnou jednotkou Glustera je tehla (angl. brick), ktorých môže byť niekoľko na jednom serveri. Tieto tehly sa na serveroch spájajú do zväzkov na základe konfigurácie, a potom sú prístupné ako klasický súborový systém cez NFS alebo GlusterFS protokolu. V súčasnosti existujú tri druhy zväzkov, ktoré je možné medzi sebou kombinovať[1]:

- **Distribovaný zväzok** – rozkladá údaje na úrovni jednotlivých súborov medzi rôzne tehly, z čoho vyplýva, že každý súbor bude práve na jednej z tehliel. Dôležité je zabezpečenie proti strate údajov, a to napríklad mať tehlu na spoľahlivom úložisku.
- **Replikovaný zväzok** – pri vytváraní takéhoto zväzku sa dá špecifikovať miera redundancie a každý súbor je uložený na daný počet rôznych tehliel.
- **Prekladaný zväzok** – ide o obmenu distribuovaného zväzku, avšak k rozkladaniu údajov prichádza na jednotlivých častiach súborov (podľa toho, koľko tehliel sa nachádza v zväzku).

Monitoring

Na lokálnom serveri je vhodné pripraviť služby pre monitoring, ktoré umožnia detekovať chyby. Na správu kontajnerov Docker je dobré použiť napríklad nástroj Portainer, ktorý dokáže monitorovať a upravovať všetky kontajnery spustené v klastri Docker Swarm. Možnosť upravovať nastavenia a sprístupnenie konzoly v daných kontajneroch v kombinácii so vzdialeným prístupom umožní chyby rovno riešiť aj v prípade neprítomnosti.

4.2.3 Vzdialený prístup

Správa inteligentnej domácnosti, ktorú nám umožňujú platformy uvedené v kapitole je vždy dostupná z lokálnej siete. Niektoré platformy ponúkajú vlastné riešenia v cloude, no vzdialený prístup je možné realizovať aj inými spôsobmi. Pri vzdialenom prístupe do ovládacieho panela domácnosti je potrebné myslieť na skutočnosť, že infraštruktúra v domácnosti už nebude izolovaná od hrozieb a rizík, ktoré vznikajú pripojením do internetu.

Verejná IP adresa

Najjednoduchšou možnosťou je použiť na správu vlastnú verejnú IP adresu. V prípade, že je od providera k dispozícii dynamická verejná IP adresa je možné využiť služby poskytujúce DDNS. DDNS resp. dynamické DNS umožňuje v reálnom čase aktualizovať DNS záznamy domény a teda ihneď reagovať na zmenu pridelenej verejnej dynamickej IP adresy. Nevýhodou je, že nie každý provider túto službu ponúka, prípadne sú poplatky za verejnú IP vysoké, taktiež sú tu riziká ako DDoS útoky, ktoré môžu zatažovať router alebo zariadenia s verejnými portami a spôsobovať výpadky. V prípade výpadku internetu nebude možná správa domácnosti, ale všetky automatizácie a funkcie budú v lokálnej sieti naďalej funkčné.

Vzdialený server

Ďalším riešením je vlastný vzdialený server. Ponuka serverov, najmä VPS, teda virtuálnych privátnych serverov, je široká. Jednou z možností je prevádzkovať potrebné aplikácie inteligentnej domácnosti priamo na vzdialenom serveri, no veľkou nevýhodou je, že v prípade výpadku internetu sa stane inteligentná domácnosť nepoužiteľná. Preto sa ďalej budeme zaoberať riešením, ktoré nebude nijako obmedzovať chod domácnosti a jediným cieľom bude možnosť vzdialenej správy.

Pre vzdialený prístup do lokálnej siete sa dajú využiť jednoduché no zároveň bezpečné techniky nenáročné na výpočtový výkon, taktiež počet dotazov nebude veľký, keďže aplikáciu budú využívať len členovia domácnosti. Štandardne majú servery pridelenú verejnú

IP adresu. Ceny poskytovateľov serverov sú často nižšie ako ceny providerov internetu za verejnú IP. Server môžeme súčasne využívať aj na iné aplikácie. Poskytovatelia serverov ponúkajú k svojim serverom domény, pravidelné zálohovanie a iné balíky služieb, ktoré môžu byť výhodou oproti riešeniu s vlastnou verejnou IP adresou.

Vytvorenie takzvaného sieťového tunela resp. tunelovanie. Je to proces prenášania dát s využitím zaobalenia nákladného protokolu (angl. payload protocol) do prenosového protokolu (angl. delivery protocol). Najčastejšie v prípade, keď je nákladný protokol určený na prenos po sieti iného typu, a teda je bez tunelovania nemožné dostať náklad (pakety) na požadované miesto. Môže ísť napríklad o nastavenie firewallu providera. Táto technika umožňuje prenos dát určených pre súkromnú sieť prostredníctvom verejnej siete spôsobom, že prenosové uzly verejnej siete nevedia, že ide o komunikáciu súkromného charakteru. Verejná sieť v tomto prípade spravidla býva internet a pod súkromnou sieťou si môžeme predstaviť lokálnu sieť (LAN), ktorá chce komunikovať so vzdialeným počítačom. [6]

Port forwarding alebo presmerovanie portov je chápané ako aplikácia NAT (z angl. network address translation - preklad sieťových adries), ktorej úlohou je smerovanie z jednej adresy a čísla portu na inú adresu a port, prostredníctvom smerovača. Smerovač zmení cieľovú IP adresu a číslo portu komunikácie, tak aby boli pakety doručené správne zaříadeniu v inej sieti a mohla ich spracovať služba bežiaci na stanovenom porte.

V spojení s tunelovaním umožňuje presmerovanie komunikácie z určeného portu na vzdialenom serveri na port na lokálnom serveri alebo naopak. Konečnému používateľovi sa bude zdať, že služby sú spustené priamo na vzdialenom serveri, keďže budú vystavené priamo na portoch servera. Výhodou je, že spojenie pomocou prenosového protokolu môže nadviazať lokálny server. Potrebuje k tomu iba IP adresu vzdialeného servera a prípadné prístupové údaje alebo svoj verejný kľúč uložený na vzdialenom serveri. Táto technika sa nazýva „Remote port forwarding“. Následne je možné pomocou proxy servera zabezpečiť pohodlný prístup na webové rozhrania služieb cez pridelené subdomény.

Kapitola 5

Implementácia

V tejto kapitole nájdete popis riešenia, ktoré bolo implementované podľa návrhu v kapitole 4. Riešenie je založené na open-source riešeniach, ktoré budú podrobne rozobrané. Jedná sa najmä o nástroj na správu aplikácií v kontajneroch Docker a riešenie pre inteligentnú domácnosť Home Assistant. Na konfiguráciu jednotlivých zariadení je využitý nástroj Esp Home.

5.1 HW infraštruktúra

V tejto sekcii popíšeme hardvérové komponenty, ktoré sme pri riešení využili. Uvedieme spôsoby zapojenia, ktoré budú zabezpečovať vysokú dostupnosť riešenia. Taktiež spomenieme nameraný odber jednotlivých prvkov.

5.1.1 Centrálny uzol

V rámci riešenia bol vytvorený centrálny uzol - server tvorený tromi jednodoskovými počítačmi Raspberry Pi 3B. Toto je minimálny počet počítačov aby pri výpadku niektorého z nich zostalo zachované takzvané kvórum, ktoré je nutné na zachovanie prevádzky všetkých kontajnerov v Docker Swarm klastri. Raspberry Pi má integrovaný jeden Ethernet port, každý počítač bol rozšírený jednou sieťovou kartou do USB, čím sa zabezpečí konektivita ak by nastal výpadok jedného z pripojení. Každé Ethernet rozhranie je prepojené do separátneho smerovača, tým predídeme aj výpadkom spôsobeným nefunkčnosťou smerovača. Toto zapojenie sme aj prakticky otestovali, keďže jeden zo smerovačov bol vadný a často sa samovoľne reštartoval. Na počítače boli taktiež nasadené chladiče na procesor a sieťový čip.

Napájanie je zabezpečené tromi adaptérmí (5 V/3 A), ktoré sú zároveň s adaptérmí smerovačov zapojené do UPS (z angl. Uninterruptible Power Supply/Source - zdroj neprerušovaného napájania). Odber zostavy je priemerne 20 W aj s riadiacou elektronikou UPS, čo podľa údajov na krabici zabezpečí napájanie na približne 2 hodiny. Odber jedného Raspberry pi bol nameraný priemerne 2,5 W.

5.1.2 Uzly v miestnostiach

Uzly v miestnostiach sú mikrokontroléry ESP8266 alebo ESP32, ku ktorým sú pripojené senzory a aktory z celej miestnosti. Vo väčšine miestností je nutná detekcia prítomnosti

osôb pomocou BLE, preto je majoritne využívané ESP32. V priechodzej chodbe je použité ESP8266.

Napájanie je zabezpečené zo štandardnej elektrickej siete s pomocou transformátorových modulov (5 V; 700 mA; 3,5 W). Niektoré ESP32 sú osadené na vývojových doskách so slotom na batériu typu 18650. K ostatným doskám s mikrontrolérmi boli pridané moduly so slotom na batériu, ktoré v prípade výpadku napájania budú moduly napájať z batérie typu 18650. Batérie, ktoré v implementácii používame majú reálnu kapacitu približne 3500mAh. Príkion ESP32 bol približne 0,5W (100 mA 5 V) bez použitia techník uspávania (ESP32 podporuje viacero úrovní spánku). Mikrokontroléru ESP8266 bol nameraný približne príkon 0,4W (100 mA 5V). Príkion bol meraný až po boote v pohotovostnom stave s nahraným firmvérom ESPHome.

5.2 SW infraštruktúra

V tejto sekcii rozoberieme softvérové nástroje, ktoré boli súčasťou implementácie a spôsob akým boli použité.

5.2.1 Prostredie

Na 3 zariadeniach Raspberry Pi bol nasadený Docker a vytvorený Docker Swarm klaster. Perzistenciu dát zabezpečuje nástroj Gluster. Vytvára replikované úložisko na všetkých troch zariadeniach, čo zabezpečí rovnaké dáta v prípade výpadku niektorého z počítačov klastra.

Dostupnosť uzlov v miestnostiach zabezpečuje protokol mDNS. Dostupnosť služieb na jednej doméne zabezpečuje vlastný DNS server, ktorý poskytuje pre služby viaceré A záznamy.

MQTT broker

Na serveri je v Docker Swarm klastri spustený open-source (licencia EPL/EDL) MQTT broker Eclipse Mosquitto. Je nenáročný a vhodný pre použitie na všetkých zariadeniach počnúc počítačmi s nízkym výkonom až po servery. Tento protokol poskytuje nenáročný spôsob, ako prenášať správy pomocou publish modelu alebo subscribe modelu. Týmto je zradený medzi favoritov na zasielanie správ IoT, ako sú nízkoenergetické senzory alebo mobilné zariadenia (telefóny, vstavané počítače, mikrokontroléry). Projekt Mosquitto je súčasťou nadácie Eclipse Foundation a iot.eclipse.org.

5.2.2 Správa inteligentnej domácnosti

Na správu inteligentnej domácnosti bol vybraný nástroj Home assistant. Zvolili sme ho kvôli dobrej komunite, taktiež nám vyhovovala konfigurácia pomocou YAML súborov a písanie skriptov v jazyky Python.

Komunita vytvorila aj riešenie s názvom *Hass.io*, ktoré taktiež využíva Docker a umožňuje automatizovanú správu kontajnera s Home assistant. Okrem toho môže spúšťať ďalšie doplnky (angl. addons) pre tento softvér. Ponuka doplnkov je naozaj široká, doplnky dokážu zjednodušiť množstvo činností. Prvotnou snahou bolo spustenie *Hass.io* v Docker Swarm klastri, toto sa nám bohužiaľ nepodarilo. *Hass.io* je spustený ako kontajner, no potrebuje pre svoj beh privilegovaný mód, ktorý v čase písania práce nie je pri službách v Docker Swarm klastri podporovaný (na Github Docker je k tomu vytvorená „issue“, ktorá je stále



Obr. 5.1: Vizualizácia spustených služieb na počítačoch spojených v Swarm klastrí.

v stave otvorená). Docker napriek tomu umožňuje vytvoriť službu, ktorá by *Hass.io* spustila na uzle, na ktorom sama beží ako klasický kontajner s privilegovaným režimom. *Hass.io* funguje podobne a následne spúšťa Home assistant a doplku ako kontajnery. Prvou nevýhodou je, že sú všetky kontajneri spustené na jednom uzle. Pri výpadku a opätovnom spustení uzla sa spustené doplnky obnovili a došlo ku konfliktu s doplnkami, ktoré sa pri výpadku spustili na inom uzle. V budúcnosti by bolo najvhodnejšie upraviť *Hass.io* tak, aby bolo schopné spúšťať doplnky ako služby v Docker Swarm klastri. Jedná sa o netriviálnu úlohu.

Home assistant sme nakoniec spustili ako službu v Docker Swarm, bez použitia *Hass.io*. Keďže sme chceli aby vedel objavovať rôzne zariadenia v lokálnej sieti museli sme ho spustiť s príznakom, ktorý zabezpečil, že kontajner sa choval akoby bol spustený priamo na uzle. Štandardne sa z kontajnerov na hostovský uzol zrkadlia len niektoré porty, no nájsť kompletný zoznam portov, ktoré treba zrkadliť sa nám nepodarilo. Docker taktiež neumožňuje načúvať na už použitých portov, preto vznikali konflikty napríklad s funkciou mDNS na hostovskom uzle. Takéto spustenie sa ale v prostredí Docker Swarm vylučuje s pripojením do „overlay“ siete, ktorá zabezpečí prístup k ostatným kontajnerom pomocou mien kontajnerov vďaka Docker DNS naprieč celým Swarm klastrom. K ostatným službám (napr. MQTT broker) bolo teda potrebné pristupovať pomocou ich adries v lokálnej sieti.

Home assistant sa chová ako ďalší MQTT klient. Prihlasuje sa na odber tém a môže byť aj publisher. Témy pri konfigurácii aktov označené ako `statetopic` sú odoberané a témy označené ako `commandtopic` sa využívajú na publikovanie správ.

Bolo nakonfigurované spojenie Home assistant s hlasovým asistentom *Google Assistant*. Home asistent umožňuje pohodlné spojenie pomocou platenej služby *Home Assistant Cloud* alebo možnosť manuálneho nakonfigurovania, ktorá zahŕňa vytvorenie vlastnej akcie v *Google console*. Súčasťou domácnosti je inteligentný reproduktor *Google home mini* a aktory by mali byť ovládateľné pomocou neho. Po pridaní nami vopred vytvorenej akcie v aplikácii *Google Home* je možné ovládať zariadenia v dome hlasovými povelmi, napr. „OK Google, turn Kitchen primar light on“. Príkazy sú odosielané cez servery Google a nie priamo cez LAN sieť, preto je podmienkou mať funkčný vzdialený prístup zabezpečený SSL certifikátom, v konzole akcií je nutné nadefinovať endpointy podľa Home Assistant dokumentácie.

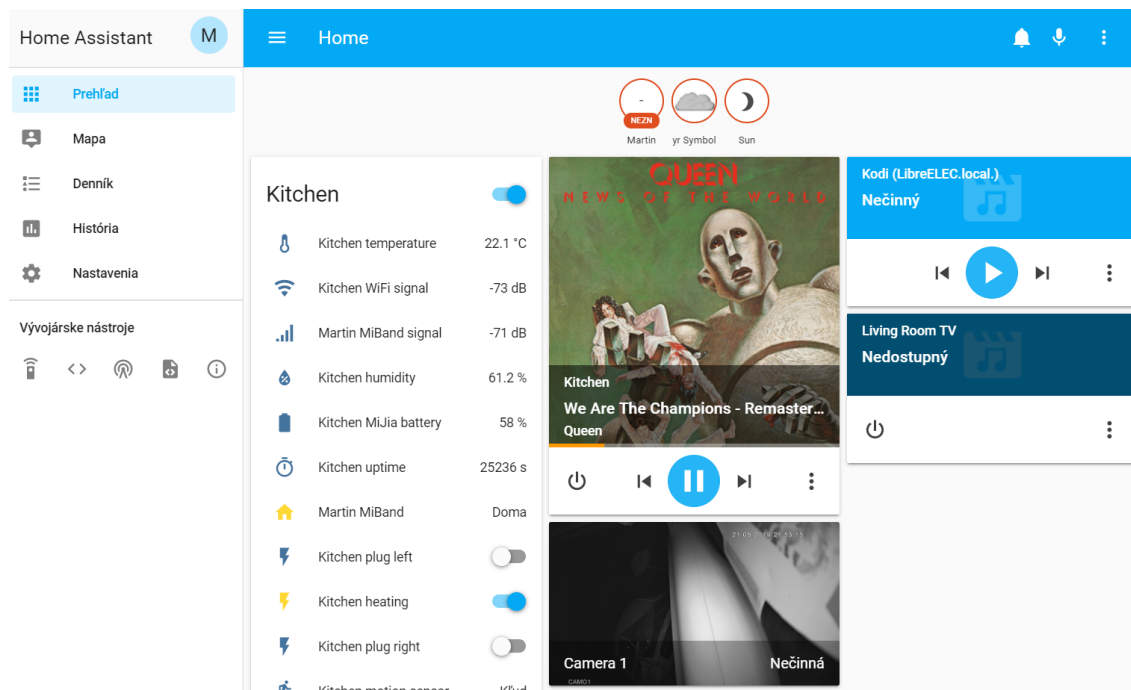
Využívali sme štandardné grafické rozhranie, no prostredie Lovelace UI je možné prispôbovať pomocou konfiguračných súborov.

5.2.3 Konfigurácia jednotlivých zariadení

Do vybraných mikrokontrolérov pre uzly v miestnostiach bolo treba pripraviť firmvér, ktorý umožní komunikáciu s Home Assistant, bude schopný pripojiť sa na MQTT broker a odoberať resp. publikovať správy.

Jednou možnosťou by bolo vytvorenie vlastného firmvéru. Vybrané mikrokontroléry je možné programovať pomocou mnohých programovacích jazykov ako napr. C/C++, Micro Python, Lua, .NET C#. Najčastejšie používaný je jazyk C/C++. Toto tvrdenie vychádza z toho, že pre jazyk C/C++ je dostupný veľký počet knižníc, C/C++ kód umožňuje kompiláciu do strojových inštrukcií, všetky nižšie spomenuté riešenia sú programované v tomto jazyku. Programovanie je možné v open-source prostredí Arduino IDE. Pri implementácii sme sa stretli aj s open-source projektom Platform IO, je to vývojové prostredie, ktoré slúži na písanie, zostavovanie aj debuggovanie kódu. Jeho výhodou je podpora veľkého množstva vývojových dosiek.

Ďalšou možnosťou je zamerať sa na open-source riešenia, ktoré už sú k dispozícii. Ak by bolo potrebné nejakú špeciálnu funkcionálnu máme možnosť si ju do riešenia dorobiť.



Obr. 5.2: Ukážka grafického rozhrania nástroja Home Assistant.

Pri prieskume možností sme zistili, že existuje viacero takýchto riešení a preto sme viaceré otestovali.

ESPurna

ESPurna je open-source firmvér pre ESP8266. Je to projekt vývojára s menom Xose Pérez, aj väčšinu kódu vytvoril práve on. Zdrojový kód je prehľadný. ESPurna podporuje mDNS mená zariadení a pripojenie k nim pomocou protokolu Telnet. V súvislosti s objavovaním zariadení v sieti podporuje taktiež protokoly NetBIOS, LLNMR, SSDP a MQTT discovery nástroja Home Assistant. Konfigurácia senzorov a aktorov prebieha pred zostavením firmvéru pomocou definícií priamo v zdrojových súboroch (pravidlá je ale možné pridávať aj po nahraní firmvéru), čo môže byť nevýhoda pre menej technicky zdatných používateľov. Pre zostavenie je odporúčané *PlatformIO* alebo *Arduino IDE*. Výhodou je prehľadné a responzívne grafické rozhranie, ktoré je spustené na webserveri priamo v ESP8266. K zariadeniam je možné pristupovať aj pomocou zabezpečenej REST API. Konfiguráciu je možné zálohovať a firmvér nahrávať s OTA (z angl. Over-the-Air - bezdrôtovo). Firmvér pre ESP32 v čase písania práce nebol dostupný. Riešenie sme sa rozhodli nevyužívať ale určite by našlo uplatnenie najmä v samostatne fungujúcich ESP8266 zariadeniach, ku ktorým sa bude ľahko pristupovať pomocou protokolov na objavovanie a elegantného grafického rozhrania.

ESPEasy

Na rozdiel od ostatných riešení toto riešenie umožňuje konfiguráciu, pridávanie zariadení, pravidiel automatizácie v čase, keď už je firmvér nahratý v zariadení. Stačí nahráť firmvér pomocou pripraveného postupu do mikrokontroléra, pripojiť sa na Wi-Fi sieť, ktorú mikrokontrolér vytvorí a nakonfigurovať všetko od vlastnej Wi-Fi, MQTT pripojenia až

po jednotlivé zariadenia a pravidlá. Toto nám umožní webové grafické rozhranie spustené na webovom severi priamo v mikrokontroléri. Webový server taktiež sprístupňuje HTTP API. Výhodou je funkcia *ESPEasy P2P Networking*, ktorá umožňuje pristupovať k senzorum a aktorom pripojených na iné zariadenia s ESPEasy v sieti. Toto umožňuje vytvoriť automatizovanú domácnosť bez nutnosti centrálného uzla - servera (napr. s MQTT brokerom). Pravidlá majú vlastnú syntax, no všetko potrebné nájdeme v dokumentácii[19]. Riešenie je aktívne vyvíjané skupinou *Lets control it* a je určené primárne pre mikrokontrolér ESP8266. V čase vytvárania práce bola dostupná iba experimentálna verzia pre ESP32. Práve pre túto skutočnosť sme sa rozhodli riešenie nakoniec nevyužiť.

ESPHome

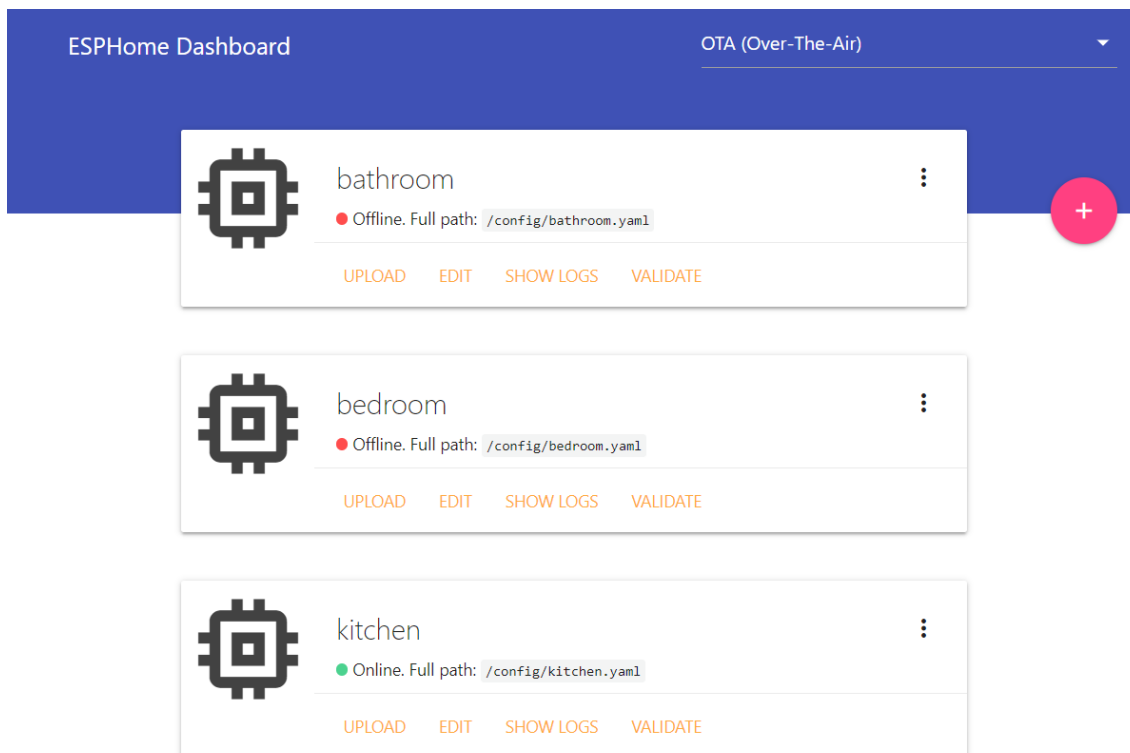
Pri implementácii používame ESPHome z dôvodu, že je najvhodnejší pre potreby v našom riešení. Podporuje mDNS a zariadenia sa dali spoľahlivo adresovať pomocou ich `.local` doménových mien. Využívame natívnu ESPHome API, ktorá umožňuje automatické objavovanie zariadení v aplikácii Homeassistant. Celkovo je tento nástroj vhodný na spoluprácu s Homeassistant, keďže konfigurácia zariadení má veľmi podobnú syntax ako konfigurácia Homeassistant. Pri každej zmene konfigurácie je nutné firmvér zostaviť a nahráť do zariadenia. Riešenie ako celok umožňuje prehľadnú správu konfigurácií zariadení v našom dome a automatizované zostavovanie a nahrávanie zdrojových kódov do mikrokontrolérov či už pomocou CLI (z angl. Command-line interface - rozhranie príkazového riadku) alebo grafického rozhrania. Firmvér je možné nahrávať s OTA (z angl. Over-the-Air - bezdrôtovo) Taktiež pri automatizáciách využívame `lambda` funkcie na písanie vlastných funkcií v C/C++. V budúcnosti bude možné písať vlastné súčasti v C++, no keďže zoznam predpripravených súčastí, najmä na komunikáciu s množstvom senzorov a aktorov, túto možnosť nepoužívame.

5.2.4 Komunikácia

Jednou z úloh, ktoré je potrebné riešiť je komunikácia jednotlivých častí riešenia. Zamerali sme sa na možnosti komunikácie pomocou MQTT, ktoré je v oblasti IoT veľmi rozšírené. Taktiež popíšeme komunikáciu medzi nástrojmi ESPHome a Home Assistant.

Cieľom riešenia je taktiež dosiahnuť interoperabilitu s existujúcimi komerčnými aj open-source riešeniami. Existujúce riešenia často využívajú na komunikáciu protokoly spomenuté v kapitole 2.1. V našom riešení na komunikáciu na aplikačnej úrovni využívame majoritne protokol MQTT, preto zariadenia podporujúce tento protokol stačí správne nakonfigurovať. Aby sme boli schopní spracovať ostatné protokoly je vhodné použiť existujúce open-source brány. Brány umožňujú konverziu jednotlivých protokolov na iné protokoly. Logickou voľbou je využiť brány ktoré umožnia zariadenia sledovať a ovládať pomocou protokolu MQTT. Experimentovali sme s nasledujúcimi open-source riešeniami, podporujúcimi MQTT:

OpenMQTTGateway je open-source riešenie (licencia GPL-3.0) určené pre mikrokontroléry ESP32 a ESP8266. Umožňuje obojsmernú konverziu komunikácie pre zariadenia pracujúce v pásme 433/315Mhz, RFM69 modulov, zariadeniach ovládaných pomocou signálov v infračervenom pásme - tzv. IR zariadenia, vytvára bránu pre 2G (GSM/GPRS), teda umožňuje odosielať a prijímať SMS správy. Tiež môže slúžiť ako brána pre IoT senzory a aktory pracujúce s protokolom LoRa. Jednosmerne dokáže preposielať správy z Mi Flora a BLE zariadení. Na jednotlivé funkcionality je potrebné zabezpečiť aj odpovedajúce periférie.



Obr. 5.3: Grafické rozhanie ESPHome na úpravu a nahrávanie firmvéru do zariadení.

Toto riešenie sme nevyužili z dôvodu chýb pri preklade firmvéru. Chyby bolo možné opraviť, keďže sa s nimi stretli viacerí používatelia. Taktiež by sme nevyužili veľké množstvo podporovaných protokolov. Toto riešenie má ale určite zmysel napr. pri integrácii LoRa zariadení, pre ktoré sme nenašli iné stabilné riešenie s dobrou komunitou.

ESPHome je vyššie spomenutý konfiguračný nástroj, ktorý okrem správy senzorov a aktorov poskytuje vďaka dodatočným perifériám aj obojsmernú konverziu komunikácie pre infračervené a 433/315 Mhz pásmo.

Podporuje preposielanie správ z BLE zariadení, konkrétne Xiaomi MiJia a MiFlora. Xiaomi MiJia, ktoré v riešení využívame. Bluetooth LE využívame aj v spojení s náramkami *Xiaomi MiBand 3*, ktoré majú všetci členovia domácnosti. Senzorovú platformu **esp32ble-tracker** využívame na detekciu prítomnosti osôb v miestnosti podľa ich náramkov.

Zigbee2mqtt je open-source program (licencia GPL-3.0) písaný v prostredí Node.js. Využíva cenovo dostupné USB zariadenie s čipom CC2531 a je určené pre mikropočítače ako Raspberry Pi a podobne.

Tento program sme spustili separátne na mikropočítači Raspberry Pi Zero. Do USB zariadenia s CC2531 bolo potrebné nahráť firmvér, ktorý je k dispozícii pod MIT licenciou. Na nahrávanie nie je potrebné zaobstaráť si *CC Debugger*, je dobré použiť alternatívnu metódu nahrávania firmvéru pomocou Raspberry Pi, ktoré budeme tak či tak potrebovať na samotný beh aplikácie. Výhodou je podpora MQTT auto discovery, ktorú sme použili na automatické objavenie zariadení v riešení HomeAssistant. Program sme otestovali v spojení so spínanou zásuvkou *IKEA Trådfri*. Obširný zoznam podporovaných zariadení ako aj podrobné postupy sú uvedené v prehľadnej dokumentácii[16].

Komunikácia medzi nástrojmi ESPHome a Home Assistant je zabezpečená pomocou natívnej API ESPHome, ktorá je z používateľského hľadiska lepšou alternatívou k MQTT. Zariadenia sú medzi nástrojmi automaticky synchronizované a komunikácia je optimalizovaná (podľa dokumentácie je napr. správa o stave binárneho senzora redukovaná na 1/10 oproti MQTT) a s nízkou latenciou. Taktiež odpadáva potreba MQTT brokera, čo odstraňuje možné miesto poruchy systému. Ďalšou výhodou je používanie služieb nástroja Home Assistant priamo v automatizáciách ESPHome. Toto ešte viac posilňuje koncept automatizácií priamo v uzloch v miestnostiach, ktorého benefitom je samostatné fungovanie aj v prípade výpadku centrálného uzla. MQTT napriek tomu môžeme využívať napr. na spojenie s bránou Zigbee2mqtt alebo s nástrojom Nodered.

5.2.5 Vzdialený server

V rámci riešenia bol nakonfigurovaný server od firmy *Scaleway*. Bol zvolený server typu „BareMetal“, to znamená, že používateľ má pridelený vlastný fyzický server a prostriedky nie sú zdieľané s ostatnými používateľmi. Pre potreby práce je dostatočný základný server C1 s architektúrou `armv7l` (podobne ako Raspberry Pi) a špecifikáciou:

- 4 x ARMv7 procesor (Marvell Armada 370/XP)
- 2 GB RAM
- 50 GB SSD
- 200 Mbps bandwidth (šírka pásma)

Mesačné náklady za prenájom servera sú v čase písania práce 2,99 EUR bez DPH. Server je fyzicky umiestnený v Paríži, no sú k dispozícii aj servery v Amsterdame. Taktiež bola zakúpená doména a pridané DNS záznamy ukazujúce na tento server.

Na serveri beží v prostredí Docker niekoľko kontajnerov, z ktorých každý má svoju funkciu:

SSHD server má na starosti udržiavanie SSH tunelu s lokálnym serverom.

Nginx Proxy Manger je nástroj, ktorý využíva *Nginx* ako proxy server a umožňuje presmerovanie požiadaviek na webové služby bežiacie na serveri na základe pridelených subdomén. Keďže beží ako kontajner, služby je možné definovať ich názvami a o preklad sa postará inerné DNS nástroja Docker. Prístup k webovým službám je možné obmedziť pomocou Basic access authentication (jednoduché overenie prístupu). Jeho súčasťou je tiež jednoduchá správa *Let's Encrypt* certifikátov. Súčasťou nástroja je prehľadné grafické rozhranie založené na šablóne *Tabler*.

MariaDB databáza je určená na ukladanie dát z nástroja *Nginx Proxy Manger*. Je prístupná iba v internej sieti prostredia Docker.

Portainer slúži na monitoring komponentov prostredia Docker.

Netdata využívame na monitoring servera, konkrétne sledovanie vyťaženia siete a prostriedkov (procesor, RAM, disk atď.) s nízkou granularitou (1 s). Vyťaženie môže byť sledované z pohľadu celého servera aj jednotlivých aplikácií. V prípade potreby môže byť integrovaný s ďalšími nástrojmi na monitoring ako Prometheus, Graphite, OpenTSDB, Kafka, Grafana a podobne.

5.3 Automatizácie

V tejto sekcii popíšeme používaný spôsob akým je možné implementovať pravidlá automatizácií v kombinácii softvérových riešení, ktoré sme zvolili. Spôsobov na písanie automatizácií je viac. Navyše bol na centrálny uzol nasadený nástroj *NodeRED*, ktorý umožňuje tvorbu automatizácii vizuálnym programovaním (presnejšie angl. flow-based programming). Popíšeme tiež postup akým fungujú automatizácie na ovládanie kúrenia a osvetlenia.

5.3.1 Konfigurácia pravidiel

V riešení sú vytvorené pravidlá automatizácie v nástroji ESPHome separátne pre každé zariadenie. Výhodou takýchto automatizácií je, že fungujú aj keď zariadenie stratí spojenie s centrálnym uzlom. Taktiež je možné konkrétne zariadenie prihlásiť na odber určitej témy z MQTT brokera, prijaté informácie spracovávať a na základe nich vytvárať pravidlá. Pravidlá sa píše do konfiguračných YAML súborov obdobne ako v nástroji Home Assistant. Pravidlá je možné vytvárať aj priamo v Home Assistante, ale uplatnenie to nájde iba v prípade interakcie senzorov a aktorov, ktoré sú na rôznych uzloch, inak by používateľ zbytočne riskoval nefunkčnosť v prípade výpadku hlavného uzlu. Aj keď je riešenie zamerané na vysokú dostupnosť, môžu nastať prípady, kedy výpadku hlavného uzlu nezabráname.

ESPHome automatizácie umožňujú spracovávať spúšťače (angl. trigger) z rôznych entít, vrátane senzorov, ktoré sú pomenované samopopisnými názvami inšpirovanými udalosťami z Javacriptu. Zoznam spúšťačov v tvare „entita.spúšťač“:

- `api.services`
- `sensor.on_value` / `sensor.on_raw_value` / `sensor.on_value_range`
- `binary_sensor.on_press` / `binary_sensor.on_release` / `binary_sensor.on_state`
- `binary_sensor.on_click` / `binary_sensor.on_double_click` / `binary_sensor.on_multi_click`
- `esphome.on_boot` / `esphome.on_shutdown` / `esphome.on_loop`
- `time.on_time`
- `mqtt.on_message` / `mqtt.on_json_message`
- `pn532.on_tag`
- `interval.interval`
- `switch.on_turn_on` / `switch.on_turn_off`

Každý spúšťač môže spúšťať množinu akcií na rôznych aktoroch. Väčšine entít môže byť používateľom pridelený jednoznačný identifikátor, ktorým ho identifikujeme pri vyvolaní akcie. Zoznam akcií, akcie viazané na určité entity v tvare „entita.akcia“:

- `delay`
- `lambda`

- `if / while / wait_util`
- `component.update`
- `script.execute / script.stop`
- `logger.log`
- `homeassistant.service`
- `mqtt.publish / mqtt.publish_json`
- `switch.toggle / switch.turn_off / switch.turn_on`
- `light.toggle / light.turn_off / light.turn_on`
- `cover.open / cover.close / cover.stop`
- `fan.toggle / fan.turn_off / fan.turn_on`
- `output.turn_off / output.turn_on / output.set_level`
- `deep_sleep.enter / deep_sleep.prevent`
- `sensor.template.publish / binary_sensor.template.publish /
cover.template.publish / switch.template.publish /
text_sensor.template.publish`
- `stepper.set_target / stepper.report_position`
- `servo.write`

5.3.2 Implementované automatizácie

Boli implementované automatizácie na ovládanie kúrenia a osvetlenia.

Na základe teploty zo senzorov DHT22 alebo Xiaomi Mijia sa zopne relé ovládajúce elektrotermickú hlavicu pri dosiahnutí stanovenej najnižšej nožnej teploty a pri dosiahnutí hornej hranice teploty sa relé vypne - radiátor prestane kúriť.

Osvetlenie je spínané buď pohybovým senzorom alebo tlačidlovými vypínačmi. Pri krátkom stlačení tlačidla bude svietidlo rozsvietené dovtedy, kým z pohybového senzora nepríde informácia o tom, že nie detekovaný žiadny pohyb, zároveň sa berie do úvahy počet ľudí detekovaných v miestnosti na základe detekcie prítomnosti pomocou BLE náramkov Mi Band. Ak je detekovaná prítomnosť nejakej osoby v miestnosti svetlá zostanú aj napriek nedetekovanému pohybu aktívne, pretože ak osoba vykonáva činnosť, pri ktorej nevykonáva žiadny väčší pohyb, napr. práca za počítačom, môže sa stať, že sensor nezaznamená pohyb. Pri dlhšom stlačení (viac ako 1 sekunda) svietidlá budú svietiť až do ďalšieho stlačenia tlačidla, senzory na ovládanie osvetlenia nebudú mať vplyv. Týmto sa zabezpečí možnosť svietiť aj vtedy ak je v dome osoba, ktorá nemá BLE náramok. Detekcia prítomnosti osôb je realizovaná pomocou komponentu `esp32_ble_tracker`. V budúcnosti bude potrebné optimalizovať detekciu pomocou merania intenzity signálu (RSSI), keďže binárny senzor, ktorý poskytuje tento komponent môže v prípade menších miestností detekovať aj osoby z vedľajších miestností.

Kapitola 6

Záver

V práci boli rozobrané možnosti komerčných a open-source riešení inteligentnej domácnosti a automatizácie. Bola navrhnutá realizácia inteligentnej domácnosti a komponenty v nej použité. Taktiež boli popísané a porovnané najrozšírenejšie protokoly v oblasti internetu vecí (IoT). Bol vybraný protokol MQTT ako hlavný protokol na komunikáciu medzi zariadeniami v domácnosti a nástroj Home Assistant ako nástroj na správu inteligentnej domácnosti.

Počas experimentovania bolo zistené, že na konfiguráciu jednotlivých uzlov v miestnostiach bude najvhodnejšie využiť nástroj ESPHome, ktorý dokáže spolupracovať s nástrojom Home Assistant pomocou vlastnej optimalizovanej API. Do nástroja na správu bol integrovaný hlasový asistent Google Home Mini a vyskúšané pridávanie zariadení do rutín, ktoré sa dajú vyvolať hlasovým povelenom.

Súčasťou práce bolo vytvorenie a nastavenie klastra Docker Swarm, ktoré zabezpečí funkčnosť aj po vypadnutí jedného z počítačov v klastru. Riešenie je tiež odolné voči výpadku elektrickej energie vďaka záložnému napájaniu z batérií.

Navrhnuté riešenie bolo vyskúšané v praxi. K jednotlivým uzlom v miestnostiach boli pripojené vybrané senzory a aktory. Boli nakonfigurované automatizácie na automatické ovládanie osvetlenia a kúrenia.

Aby bolo možné domácnosť sledovať a ovládať aj vzdialene bol nastavený vzdialený server, ktorý pomocou tunelovania sprostredkováva spojenie zo službami na lokálnom serveri.

V budúcnosti by bolo vhodné zamerať sa na zníženie energetickej náročnosti vďaka funkciám spánku na vybraných mikrokontroléroch. Umožnilo by to dlhší chod na batérie v prípade výpadku elektrickej energie, prípadne využitie aj na miestach kde nie je dostupné trvalé napájanie. Taktiež budú používaním pribúdať nové senzory, aktory a vznikajú nové automatizácie, najmä bude nutné vytvoriť pravidlá, ktoré budú notifikovať užívateľov v prípade nepredvídateľných udalostí. Veľkým prínosom by bolo aktívne spolupracovať s komunitou a rozvíjať existujúce open-source nástroje.

V porovnaní s komerčnými riešeniami má navrhnuté riešenie niekoľko výhod. Najväčšiu výhodou je voľnosť pri výbere zariadení a funkcionality, ktorú si môže používateľ sám doprogramovať. Komerčné riešenia sú často obmedzené na vlastné senzory a aktory. Keďže komunita okolo open-source nástrojov je tvorená rozmanitými ľuďmi, nástroje sa rýchlejšie prispôbia aktuálnym novinkám v oblasti IoT. Od komerčných riešení odrádzajú taktiež aféry so zasahovaním do súkromia. Výhodou komerčných riešení je určite servisná podpora a určitá garancia dostupnosti služieb.

Literatúra

- [1] Baier, J.: *Velká řešení otevřeně: souborový systém GlusterFS*. Root.cz, Máj 2017, [Online; navštívené 12.5.2019].
URL <https://www.root.cz/clanky/velka-reseni-otevrene-souborovy-system-glusterfs/>
- [2] Baker, J.: *6 open source home automation tools*. dec 2019, [Online; navštíveno 20.1.2019].
URL <https://opensource.com/tools/home-automation>
- [3] CAO, J.; XU, L.; Raef, A.; aj.: An OS for Internet of everything: Early experience from a smart home prototype. *ZTE COMMUNICATIONS*, ročník 15, č. 4, 2017: str. 1.
- [4] Constellation Energy Resources: *6 Essential Routines for Smart-Home Assistants*. Apríl 2018, [Online; navštíveno 20.1.2019].
URL <https://blog.constellation.com/2018/04/27/essential-routines-smart-home-assistants/>
- [5] Control4: *Control4 – Investor presentation*. November 2018, [Online; navštíveno 5.1.2019].
URL <https://investor.control4.com/static-files/784b582c-d29b-4368-b5d1-3277b1a93921>
- [6] Daniel Adam, L. M. P. S., Mária Gemeranová: *Tunneling*. [Online; navštívené 1.5.2019].
URL http://www.marsik.sk/laci/pages/oldweb/tunneling_projekt.pdf
- [7] Espressif: *ESP8266EX Datasheet*. 2018, [Online; navštíveno 20.1.2019].
URL https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- [8] Espressif: *ESP32 Series Datasheet*. 2019, [Online; navštíveno 20.1.2019].
URL https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [9] Fibaro: *Fibaro – Compatibility list*. 2019, [Online; navštíveno 20.1.2019].
URL <https://compatibility.fibaro.com/>
- [10] Fibaro: *Fibaro Home Center 2 – Technical specification*. 2019, [Online; navštíveno 20.1.2019].
URL <https://www.fibaro.com/en/products/home-center-2/specifications/>

- [11] Fibaro: *Fibaro Knowledge Base - What is scenes?* 2019, [Online; navštíveno 20.1.2019].
URL <https://manuals.fibaro.com/knowledge-base-browse/scenes-usage/>
- [12] Fibaro: *Fibaro Manuals*. 2019, [Online; navštíveno 20.1.2019].
URL <https://manuals.fibaro.com/>
- [13] Home Assistant: *Home Assistant Developer documentation – Architecture*. [Online; navštíveno 25.07.2016].
URL https://developers.home-assistant.io/docs/en/architecture_index.html
- [14] iNELS: *iNELS sběrníková elektroinstalace*. [Online; navštíveno 26.12.2018].
URL https://www.elkoep.cz/media/files/download/item/files-149/11_sec_iNELS_BUS_technical_2018_CZ_view_version.pdf
- [15] Jacobson, J.: *Control4 CEO on Home Automation Standards, DIY, Pro Install*. Január 2014, [Online; navštíveno 5.1.2019].
URL https://www.cepro.com/article/control4_ceo_on_home_automation_standards_diy_pro_install/
- [16] Kanters, K.: *Zigbee2mqtt documentation*. [Online; navštívené 12.5.2019].
URL <http://www.zigbee2mqtt.io/>
- [17] Kapusta, M.: *Inteligentná domácnosť: Automatizácia, bezpečnosť a inteligentné osvetlenie aj u vás doma*. Jún 2016, [Online; navštíveno 26.12.2018].
URL <https://www.mojandroid.sk/inteligentna-domacnost-prehľad/>
- [18] Lee, S.; Kim, H.; Hong, D.; aj.: Correlation analysis of MQTT loss and delay according to QoS level. In *The International Conference on Information Networking 2013 (ICOIN)*, Jan 2013, ISSN 1550-445X, s. 714–717, doi:10.1109/ICOIN.2013.6496715.
- [19] Let's control it: *ESPEasy*. [Online; navštívené 12.5.2019].
URL <https://www.letscontrolit.com/wiki/index.php/ESPEasy>
- [20] Loxone: *Loxone – Knowledge Base*. [Online; navštíveno 26.12.2018].
URL <https://www.loxone.com/enen/kb-cat/all/>
- [21] Loxone: *Loxone software a on-line služby*. 2019, [Online; navštíveno 20.1.2019].
URL <https://shop.loxone.com/cscz/software.html>
- [22] Loxone: *Software pro chytrý dům*. 2019, [Online; navštíveno 20.1.2019].
URL <https://www.loxone.com/cscz/produkty/software/>
- [23] Loxone: *Zprovoznění Miniserveru*. 2019, [Online; navštíveno 20.1.2019].
URL <https://www.loxone.com/cscz/kb/miniserver-zprovozneni/>
- [24] Naik, N.: Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In *2017 IEEE International Systems Engineering Symposium (ISSE)*, Oct 2017, s. 1–7, doi:10.1109/SysEng.2017.8088251.

- [25] Newman, J.: *How Smart Home Lock-In Imprisons You, And Why That Might Change*. Marec 2018, [Online; navštíveno 26.12.2018].
URL <https://www.fastcompany.com/40545455/dont-let-your-smart-home-lock-you-in>
- [26] RS Components: *11 Internet of Things (IoT) Protocols You Need to Know About*. Apríl 2015, [Online; navštíveno 10.1.2019].
URL <https://www.rs-online.com/designspark/eleven-internet-of-things-iot-protocols-you-need-to-know-about>
- [27] Trlica, D.: *Asistent Google v češtině! Oficiální podpora přijde do Česka již brzy*. Máj 2018, [Online; navštíveno 20.1.2019].
URL <https://www.svetandroida.cz/asistent-google-v-cestine-cesko-podpora/>
- [28] XMPP: *An Overview of XMPP*. [Online; navštívené 5.5.2019].
URL <https://xmpp.org/about/technology-overview.html>
- [29] Zigbee: *Zigbee is the only complete IoT solution, from the mesh network to the universal language that allows smart objects to work together*. [Online; navštíveno 10.1.2019].
URL <https://www.zigbee.org/zigbee-for-developers/zigbee-3-0/>




Príloha A

Obsah CD

- `homeassistant/` - konfigurácia Homeassistant
- `esphome/` - konfiguračné súbory uzlov v miestnostiach
- `autossh/` - konfigurácia doplnku Autossh
- `dnsmasq/` - konfigurácia Dnsmasq
- `local-docker-compose/` - súbory na nasadenie lokálneho servera
- `remote-docker-compose/` - súbory na nasadenie vzdialeného servera
- `latex/` - zdrojové súbory práce v \LaTeX -e


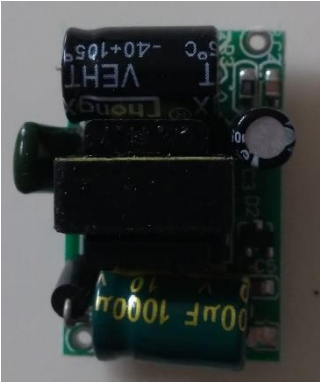

Príloha B


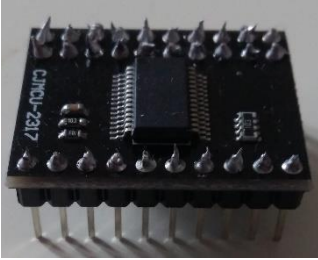
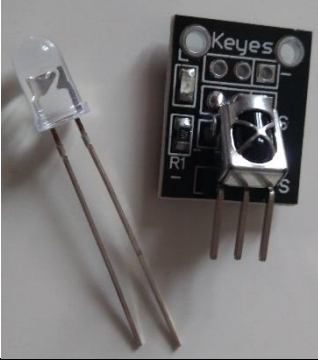

Zoznam komponentov




Názov	Obrázok	Použitie	Jednotková cena	Počet kusov	Celková cena
Senzor otvorenia okna / dverí		--	0,69€	7 ks	4,83€
<u>Besder 6024PB-IP20H1</u>		Kamerový systém	24,49 €	2 ks	48,98 €
ESP8266		Uzol v miestnosti	1,98€	1ks	1,98€




<p>3,3V to 5V logic converter</p>		<p>Obojsmerný prevod z 3,3V na 5V logickú úroveň</p>	<p>1,34€/bal.</p>	<p>10 ks (1 bal.)</p>	<p>1,34€</p>
<p>ESP32 battery shield /ESP32 + battery...</p>		<p>Uzol v miestnosti</p>	<p>6,78€</p>	<p>5ks</p>	<p>33,90 €</p>
<p>Pohybový senzor</p>		<p>Snímanie pohybu v miestnosti</p>	<p>0,75€</p>	<p>6 ks</p>	<p>4,50€</p>
<p>DHT22</p>		<p>Seznor vlhkosti a teploty, použitý chodbe, WC, kupaľni a spálni</p>	<p>2,21€</p>	<p>4ks</p>	<p>8,84€</p>




<p>CH.I.P. – 9 dollar computer</p>		<p>Zigbee2mqtt brána, použité</p>	<p>8,06€</p>	<p>1 ks</p>	<p>8,06€</p>
<p>Adaptér</p>		<p>Použité na napájanie Raspberry Pi a CH.I.P.</p>	<p>2,20€</p>	<p>4 ks</p>	<p>8,80€</p>
<p>Batéria</p>		<p>Záložné napájanie ESP8266/ESP32</p>	<p>1,75€</p>	<p>5 ks</p>	<p>8,75€</p>
<p>Teplomer s vlhkomerom Xiaomi Mijia</p>		<p>Seznor vlhkosti a teploty, použitý v obývačke a kuchyni</p>	<p>12,03 €</p>	<p>2ks</p>	<p>24,06 €</p>





<p>Sieťová karta USB to Ethernet</p>		<p>Záložná sieťová karta pre Raspberry Pi</p>	<p>1,55€</p>	<p>3 ks</p>	<p>4,65€</p>
<p>230V AC to 5V DC</p>		<p>Napájanie ESP8266/ESP32</p>	<p>0,70€</p>	<p>6 ks</p>	<p>4,20€</p>
<p>Dvojité tlačidlo LEGRAND NILOE</p>		<p>Ovládanie osvetlenia</p>	<p>3,66€</p>	<p>3 ks</p>	<p>10,98 €</p>


<p>IKEA TRÅDFRI</p>		<p>Spínaná zásuvka s protkolom Zigbee</p>	<p>9,99€</p>	<p>1 ks</p>	<p>9,99€</p>
<p>IO expander MCP2307</p>		<p>V prípade potreby rozšírenie portov, počtu IO pinov ESP</p>	<p>0,86€</p>	<p>2 ks</p>	<p>1,72€</p>
<p>IR prijímač/vysielač</p>		<p>IR brána</p>	<p>0,53€</p>	<p>1 ks</p>	<p>0,53€</p>
<p>Tlačidlo LEGRAND NILOE</p>		<p>Ovládanie osvetlenia</p>	<p>1,45€</p>	<p>6 ks</p>	<p>8,70€</p>

<p>Ističe</p>		<p>Ističe navyše štandardnej elektorinštalácie</p>	<p>1,79€</p>	<p>5 ks</p>	<p>8,95€</p>
<p>MQ2</p>		<p>Senzor CO</p>	<p>0,91€</p>	<p>4 ks</p>	<p>3,64€</p>
<p>CC2531</p>		<p>Zigbee2mqtt brána</p>	<p>4,51€</p>	<p>1 ks</p>	<p>4,51€</p>

<p>4-kanálové relé 250V 30A</p>		<p>Spínanie zásuviek</p>	<p>11,09 €</p>	<p>2 ks</p>	<p>22,18 €</p>
<p>4-kanálové relé 250V 10A</p>		<p>Spínanie osvetlenia a elektrotermických hlavíc</p>	<p>2,00€</p>	<p>6 ks</p>	<p>12€</p>
<p>Raspeberry Pi 3B</p>		<p>Centrálny uzol – lokálny server (zapojené v klastri)</p>	<p>35€</p>	<p>3 ks</p>	<p>105€</p>

<p>Kabeláž 3x2,5</p>		<p>Cca 30 m navyše oproti štandardnej elektroinštalácii (spínanie zásuviek)</p>	<p>0,75€</p>	<p>30 m</p>	<p>22,50 €</p>
<p>Kabeláž 3x1,5</p>		<p>Cca 50 m navyše oproti štandardnej elektroinštalácii (spínanie osvetlenia a elektrotermick ých hlavíc, tlačidlá)</p>	<p>0,48€</p>	<p>50 m</p>	<p>24€</p>
<p>LAN káble rôznych kategórií</p>		<p>Cca 30 m navyše oproti štandardnej elektroinštalácii</p>	<p>0,55€</p>	<p>30 m</p>	<p>16,50 €</p>
<p>POE adaptér, 4- kanálový</p>		<p>Kamerový systém</p>	<p>4,54€</p>	<p>1 ks</p>	<p>4,54€</p>

<p>Elektroinštaláčna krabica</p>		<p>Uloženie súčiastok</p>	<p>1,46€</p>	<p>6 ks</p>	<p>8,76€</p>
<p>SD karta 64GB</p>		<p>Pamäť pre Raspberry Pi 3B</p>	<p>12,67 €</p>	<p>3 ks</p>	<p>38,01</p>
<p>Elektrotermické hlavice</p>		<p>Spustenie radiátorov</p>	<p>15€</p>	<p>3 ks</p>	<p>45€</p>
<p>Google Home Mini</p>		<p>Hlasový asistent</p>	<p>50€</p>	<p>1 ks</p>	<p>50€</p>

<p>Doplňkový materiál</p>		<p>Rezistory, prepojovacie káble, tranzistory, diódy, a iné</p>	<p>--</p>	<p>--</p>	<p>10€</p>
<p>Suma spolu</p>	<p>613,40€</p>				