



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

BEZDRÁTOVÉ OVLÁDÁNÍ ELEKTRONIKY MOBILNÍM/EM-BEDDED ZAŘÍZENÍM S VYUŽITÍM WIFI

WIRELESS CONTROL OF ELECTRONIC DEVICES USING MOBILE/EMBEDDED DEVICE USING

WIFI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR MAREK

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK,

BRNO 2020

Zadání bakalářské práce



Student: **Marek Petr**

Program: Informační technologie

Název: **Bezdrátové ovládání elektronik mobilním/embedded zařízením s využitím WiFi**
Wireless Control of Electronic Devices Using Mobile/Embedded Device Using WiFi

Kategorie: Vestavěné systémy

Zadání:

1. Prostudujte dostupná řešení a literaturu na téma bezdrátové sítě embedded zařízení s využitím WiFi, případně jiných přenosových bezdrátových protokolů, zaměřte se i na možnosti využití Raspberry Pi.
2. Vytipujte vhodnou aplikaci, například pro domácnost, diskutujte dosažitelné vlastnosti řešení takové aplikace, například ovládání světel a domácích spotřebičů s tím, že samotným ovladačem by mohl být i mobilní telefon.
3. Navrhněte způsob implementace vybrané aplikace a popište možnosti a dosažitelné vlastnosti.
4. Implementujte aplikaci a její ovládání a demonstруйте vlastnosti a možnosti na vhodném příkladu.
5. Diskutujte dosažené výsledky a další možnosti pokračování práce.

Literatura:

- Dle pokynů vedoucího

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

Vedoucí práce: **Zemčík Pavel, prof. Dr. Ing.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Abstrakt...

Abstract

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

Klíčová slova

Raspberry Pi, ESP8266, ESP32, Wi-Fi, bezdrátové ovládání, automatizace domácnosti

Keywords

Raspberry Pi, ESP8266, ESP32, Wi-Fi, remote control, home automation

Citace

MAREK, Petr. *Bezdrátové ovládání elektroniky mobilním/embedded zařízením s využitím Wi-Fi*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Dr. Ing. Pavel Zemčík,

Bezdrátové ovládání elektroniky mobilním/embedded zařízením s využitím WiFi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Petr Marek
22. dubna 2021

Obsah

1	Úvod	3
2	Automatizace domácnosti	4
2.1	Automatizace domácnosti a chytrý dům	4
2.2	Existující řešení chytrých domů	8
3	Protokoly a technologie používané v automatizované domácnosti	17
3.1	Komunikační protokol CoAP	17
3.2	MQTT protokol	19
3.3	Bezdrátový přenos dat a signálů	20
3.4	Technologie bezdrátového přenosu WiFi	21
3.5	Ostatní technologie bezdrátového přenosu	22
4	Vestavné systémy	24
4.1	Vestavné systémy, mikropočítače a mikrokontrolery	24
4.2	Jednodeskový počítač Raspberry Pi	25
4.3	Ostatní mikropočítače	25
4.4	Moduly ESP8266, ESP32 a ESP32-S2	25
4.5	Ostatní mikrokontrolery a vývojové desky	25
4.6	Možnosti bezdrátového přenosu ve vestavných systémech	26
5	Zhodnocení současného stavu a plán práce	27
5.1	Zhodnocení současného stavu v oblasti automatizace domácnosti	27
5.2	Návrh technického řešení	28
5.3	Požadované vlastnosti navrhovaného systému	29
6	Realizace a testování	31
6.1	Celkový návrh systému	31
6.2	Grafický návrh webové aplikace	34
6.3	Databáze, autentizace v systému a webhosting	35
6.4	Implementace klientské aplikace	35
6.5	Implementace serveru na Raspberry Pi a aplikace pro koncové moduly . . .	37
6.6	Testování a vyhodnocení systému	40
7	Závěr	43
	Literatura	44
A	Zprovoznění systému na různých zařízeních	50

Kapitola 1

Úvod

Nacházíme se v době rychlého vývoje techniky, který pronikl i do našich domácností. Každým dnem roste počet zařízení denní potřeby, která jsou připojená do celosvětové sítě internet. A spolu s tímto trendem roste také komfort a možnosti bydlení, které z toho plynou. Díky tak rychlému rozvoji automatizace je možné provádět úkoly, které by dříve ani nebyli možné. Internet věcí lidem umožňuje vykonávat úkony jako vypínat spotřebiče při opuštění domácnosti, nastavit efektivní režim vytápění před příjezdem domů, či například zajistit zabezpečení celé domácnosti.

Na trhu se nachází nejrůznější typy systémů domácí automatizace. V závislosti na ceně je možné pořídit jednoduchá zařízení, jako jednotlivé zásuvky či světla až po různá komplexní řešení určená k zabudování do rozvaděčů domácnosti za účelem ovládnutí celé domácnosti. Komplexní řešení mohou pokrývat řadu různých snímačů a zařízení, které spolu mohou vytvořit opravdu automatizovaný dům. Přestože jde vývoj rychle dopředu, tak u komplexnějších systémů je cena velmi vysoká a většina domácností se tak bude muset bez chytrých zařízení obejít.

A právě kvůli zajímavosti těchto systémů a jejich doposud vysoké ceně jsem se i já rozhodl v této práci zabývat možnostmi ovládnutí zařízení na dálku, potažmo možností jejich automatizace. V rámci práce bych chtěl vyvinout takový komplexní systém, který by byl použitelný v praxi a zároveň příliš nezatížil kapsu uživatele. Systém bude postaven na platformách Raspberry Pi a ESP8266, protože jsou cenově velmi dostupné a kromě toho se rychle vyvíjí, nebude tedy chybět podpora. Ke komunikaci mezi sebou budou tyto platformy využívat Wi-Fi, jelikož již samy o sobě mají potřebné prostředky a protokol CoAP, který je k podobným aplikacím určený. Systém bude sice využívat Raspberry Pi (jelikož se jedná o levné a energeticky relativně nenáročné zařízení), ale budu jej realizovat tak, aby byl multiplatformní (alespoň tam kde to bude mít smysl). Grafické rozhraní pro komunikaci s uživatelem bude mít responzivní design. Kromě toho bude systém fungovat jak v rámci sítě internet (bude jej tedy možné ovládat na dálku, například z práce), tak i v lokální síti. Nebude tedy bezprostředně závislý na internetovém připojení. Veškerý kód, který v rámci práce vytvořím bych chtěl uvolnit jako open source a nadále jej vyvíjet.

Kapitola 2

Automatizace domácnosti

Následující část je shrnutím současného stavu v oblasti chytré či automatizované domácnosti. Není encyklopedickým výkladem problematiky, ale souhrnem informací, které mají k práci bezprostřední vztah. Nejprve je zde vysvětlení co je to chytrá či automatizovaná domácnost, jaké jsou dnes možnosti jejího využití, jaké má přínosy a obvyklé funkce. Ve druhé podkapitole se věnuji některým existujícím řešením chytrých domácností, které je možné si pořídit.

2.1 Automatizace domácnosti a chytrý dům

Internet věcí a chytrá zařízení

K automatizaci domácnosti neodmyslitelně patří internet věcí, který celý koncept automatizace významně rozšiřuje. Zatímco pod pojmem automatizace je možné si představit jakoukoli automatizovanou činnost, internet věcí označuje systém propojení různých zařízení, aplikací, snímačů a akčních členů, které mezi sebou mohou navzájem komunikovat a reagovat. Mezi aplikace konceptu internetu věcí široké spektrum odvětví jako chytrý průmysl, chytrá města, chytré zemědělství, chytré zdravotnictví, nebo konečně chytré domácnosti [51].

Automatizace domácnosti a chytrý dům

Automatizace domácnosti spočívá v automatizování činností, které řídí domácnost, normálně vykonávané člověkem. Můžeme ji definovat jako mechanismus, který nahrazuje lidskou námahu (při ovládání domácnosti), natolik, nakolik je to jen možné [U]. V souvislosti s tím někdy hovoříme o inteligentní, řízené či chytré domácnosti. Jedná se o kolekci zařízení a (pod)systémů, které jsou schopny spolu komunikovat či fungovat nezávisle. Automatizovaný „dům budoucnosti“ slibují výrobci domácích zařízení prakticky již téměř od počátku minulého století [1]. Chytrá domácnost (či smart home) je také definován jako dům, vybavený výpočetní a informační technologií, které předvídá uživateli potřeby a odpovídá na ně, a přitom dbá na jeho pohodlí, bezpečnost a zábavu [13]. Často se tedy tyto dva pojmy (automatizovaná a chytrá domácnost) zaměňují. Chytrý dům však označuje spíše skupinu zařízení, schopných spolu nějakým způsobem (zejména po síti) komunikovat, zatímco automatizovaný dům je pojem, který popisuje, jakým způsobem jsou tato zařízení využívána [64].

Možnosti využití automatizace v domácnosti

Automatizace v mnohém usnadňuje život a umožňuje provádění akcí, které by jinak byli prakticky nemožné (například zabezpečení domu, efektivní řízení vytápění domácnosti a spotřeby energie). V současné době patří automatizace domácnosti mezi rychle se rozvíjející technologie [66].

Mezi typické aplikace automatizace domácnosti patří například:

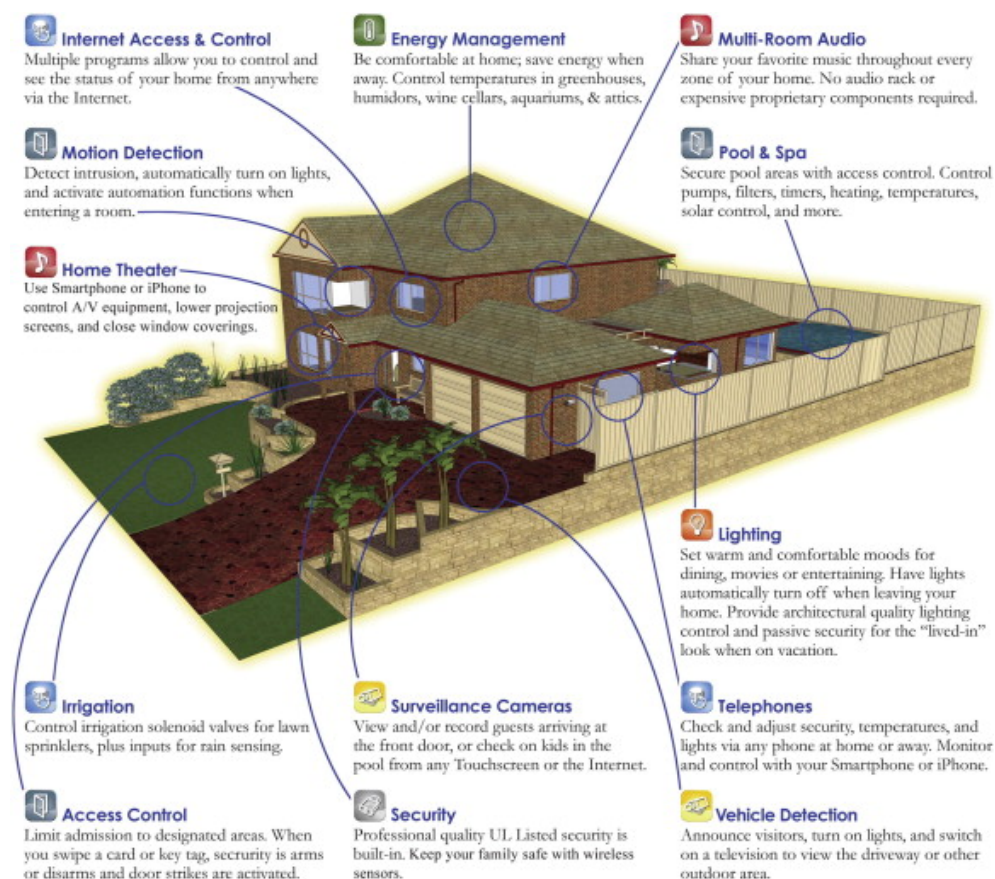
- Zabezpečovací systém
- Systém pro inteligentní vytápění a ventilaci (HVAC)
- Zábava a multimédia
- Komunikace
- Osvětlení [3]
- Ovládání spotřebičů [T]
- Samo zavlažovací systémy [U]

A samozřejmě mnoho dalšího. Zmíněné aplikace jsou nejtýpější, ke kterým se automatizace používá, ovšem na trhu vzniká čím dál více inovací, jako chytrá lednice či kávovar. Na obrázku 2.1 Můžeme vidět příklad chytrých prvků v domácnosti z nichž mnohé je dnes již možné automatizovat.

Přínosy automatizace domácnosti

Přidání inteligence do domácnosti přináší do života lidí řadu přínosů. Jde zejména o:

- Bezpečí – Chytré domy mohou používat různé senzory, které detekují nebezpečí a v souvislosti s nimi provést patřičné akce k jejich zabránění, případně minimalizaci škod. Příkladem mohou být záplavové a kouřové senzory a v neposlední řadě také zabezpečovací systém domácnosti.
- Komfort – Chytré domácnosti svými funkcemi nabízejí různé způsoby, jak jejich uživatelům zpříjemnit různé rutinní akce. Mohou se postarat o automatické nastavování žaluzií dle intenzity venkovního světla, přes dotykový displej na dálku ztlumit světlo či hlasovým pokynem uvést celý byt do jiného světelného režimu.
- Přehled o provozu – Systémy pro automatizaci domácnosti zahrnují i displeje s přehledem o stavu jednotlivých zařízení a čidel. Také je v některých systémech možné tyto informace sledovat i z chytrých telefonů, tabletů či počítačů (a to i vzdáleně). V některých komplexnějších systémech, které například zahrnují komunikaci přes mobilní síť je možné získávat přehled o provozu dokonce pomocí SMS zprávy (hodí se třeba při absenci internetového připojení) [15]
- Úspora – V chytrých domácnostech je možné použít inteligentní vytápění domu založené na údajích z teplotních čidel, denní doby, případně nastaveném režimu domácnosti. Společnost ELKO EP odhaduje, že díky bezdrátové regulaci topení je možné ušetřit až 30% nákladů na energii [17]. Úsporu rovněž zajistí automatizovaná světla, o kterých je možné mít v automatizované domácnosti vždy přehled, na dálku je zapínat/vypínat dle potřeby a rovněž je napojit na senzory, které je budou ovládat například na základě přítomnosti osob v místnosti.



Obrázek 2.1: Příklad možností automatizované domácnosti[U]

Základní klasifikace chytré domácnosti

Chytrou domácnost můžeme rozdělit dle kabeláže na:

- Drátovou
- Bezdrátovou
- Kombinovanou

Pokud má být domácnost komplexně automatizovaná, je často vhodnější mít celý systém propojený pomocí kabelů, jelikož takový systém bude spolehlivější a v případě potřeby nabízí rychlejší přenos dat (například pokud mají být součástí systému multimédia). Bezdrátové systémy se hodí zejména tam, kde není žádané zasahovat do elektroinstalace, či pokud uživatel potřebuje pouze jednodušší systém (například s ovládáním několika málo zařízení). Připravená kabeláž pro automatizaci domácnosti rovněž přináší výhodu snadnějšího řešení napájení jednotlivých chytrých zařízení, které se tak může rozvádět po bytě spolu s datovými kabely. Přitom pro propojení jednotlivých chytrých zařízení mezi sebou je možné využít různé typy kabelů (např. ethernetový) [15]. Systémy s kombinovanou kabeláží pak vycházejí z klasické kabelové instalace s možností použití některých bezdrátových prvků (např. snímačů).

Dále můžeme systémy chytré domácnosti dělit rozdělít na:

- Komplexní, dodávané specializovanými firmami
- Sestavené uživatelem dle jeho potřeb

Nevýhodou systémů od specializovaných firem je především obvykle mnohem vyšší cena. Systémy sestavené uživatelem mohou využívat nějakého open source projektu a cena je tak daná jen použitými komponentami, které mohou být nepoměrně levnější oproti těm od specializovaných firem. Samozřejmě má toto řešení řadu nevýhod, jelikož uživatel si musí celý systém navrhnout a sestavit sám a je tedy sám zodpovědný za funkčnost [73].

Také můžeme systémy rozdělit na ty, které je možné ovládat hlasem a které ne [27].

Funkce používané v chytrých domácnostech

V chytré domácnosti se často používají některé z následujících funkcí:

- Přímé ovládání spotřebičů
- Nastavení scény
- Podmínky [16, pokud neseženu lepší]

Přímé ovládání spotřebičů se provádí například dálkovým ovládáním, kde se často využívá rádiové komunikace na frekvenci 443 MHz. Příkladem takového zařízení mohou být bezdrátové zásuvky od společnosti Emos ¹. Jiný způsob přímého ovládání rovněž zahrnuje použití jiného chytrého zařízení (například chytrého telefonu), pokud ovládané zařízení umí komunikovat pomocí stejné technologie (Wi-Fi či Bluetooth). Ovládání pomocí telefonu či podobného chytrého zařízení je možné i v případě, že ovládané zařízení neumí komunikovat stejnou technologií, ale v domácnosti existuje centrální prvek (tzv. hub), který podporuje obě technologie a funguje zde jako prostředník mezi oběma zařízeními. Funkce nastavení scény obvykle jistým způsobem sdružuje několik příkazů přímého ovládání. Může se například jednat o scénu odchodu z domu, která vypne všechna světla, odpojí spotřebiče od elektrické sítě a aktivuje zabezpečovací systém. Některé chytré domácnosti například i mohou rozpoznat večerní zapnutí televize a nastavit komfortním způsobem osvětlení. Dalším principem uplatňovaným v chytré domácnosti je vlastní automatizace pomocí předem definovaných podmínek. Ty způsobí, že při určité akci systém zareaguje předem nastaveným způsobem. Například může systém díky nastaveným podmínkám udržovat teplotu v domácnosti na určité úrovni, nebo třeba zapnout venkovní osvětlení při detekci osoby za tmy.

Zde stojí za zmínku webová služba IFTTT² (If This Then That), která přidává nastavování podmínek opravdu široký rozměr. Umožňuje propojit i spolu nijak nesouvisející služby či zařízení, které spolu běžně nekomunikují [1].

Komponenty chytré domácnosti

Na trhu dnes existuje nepřeberné množství různých systémů. Tyto systémy se mezi sebou liší složitostí, cenou, dosahem apod. Chytrá domácnost může obsahovat některé z následujících komponent:

- Vstupní prvky (různá čidla, tlačítka, dotykové displeje. . .)

¹<https://www.czc.cz/emos-dalkove-ovladane-zasuvky-bila/261962/produkt>

²<https://ifttt.com/>

- Výstupní prvky (Světla, spotřebiče a různá zařízení)
- Virtuální (hlasový) asistent
- Centrální jednotka
- Aplikace pro řízení domácnosti z chytrých zařízení (telefonu, tabletu, počítače...)

Systém nemusí obsahovat všechny zmíněné komponenty. Záleží na komplexnosti a složitosti daného systému. Některé systémy mohou jako centrální jednotku využívat virtuálního (hlasového) asistenta, příkladem může být Google Assistant či Amazon Alexa.

2.2 Existující řešení chytrých domů

Dnes je na trhu nepřeberné množství systémů, lišících se v ceně, komplexnosti, způsobem komunikace a podobně. Není možné na jednotlivé systémy pohlížet stejně, protože každý systém má své vlastnosti, a některé systémy je jen stěží možné srovnávat jako alternativy (např. stěží budeme srovnávat jako alternativu hotové řešení řady chytrých zařízení Home connect a systém Apple HomeKit, který propojuje různá chytrá zařízení). Některé systémy mezi sebou dokáží dokonce komunikovat a spolupracovat, například Apple HomeKit, obecně hlasoví asistenti či dříve zmíněná služba IFTTT, pomocí které je možné propojit některé systémy. Jednou z nejčastějších aplikací automatizace, kterou různé systémy nabízejí je ovládání světel a zásuvek. Mezi další aplikace patří ovládání hlavic radiátorů, chytré termostaty, ovládání ventilátorů, stínící techniky, alarm a podobně. Níže uvádím seznam některých konkrétních systémů:

- Loxone
- Jablotron
- Apple HomeKit
- Sonoff
- Fibaro
- Homeconnect
- Kangtai
- A mnoho dalších

Kromě komerčně prodávaných systémů je k dispozici rovněž open source řešení, mezi známější patří například:

- Home Assistant
- ESPHome
- A mnoho dalších

Jednotlivé systémy se mezi sebou různým způsobem liší a není možné na ně pohlížet stejným způsobem. Každý systém má svůj způsob přidávání nových zařízení, podporu, rozšiřitelnost, architekturu apod. V jistém smyslu speciálním typem systémů chytré domácnosti patří systém, využívající hlasového asistenta. Takový systém obvykle bývá snadno rozšiřitelný. Hlasovým asistentům se věnuji na závěr této podkapitoly.

Loxone

Loxone je společnost, zaměřující se na automatizaci budov, v rozsahu od malých bytů, přes hotely až po rozsáhlé budovy a výrobní haly. Zaměřují se na širokou škálu aplikací, v oblasti automatizace domácnosti jde zejména o:

- Bezpečnost (Pohybové senzory, dveřní a okenní senzory)
- Přístup do budovy (Přístup kódem zadávaným na klávesnici, NFC přívěškem či iButtonem, kamera)
- Řízení filtrace bazénu
- Větrání (Automatické řízení ventilace, například na základě přítomnosti osoby, vlhkosti, teplotě...)
- Regulace teploty (Loxone je možné připojit k jakémukoli zdroji teploty i chlazení)
- Úspora energie (Ovládání budovy k úsporám energie – například automatické stínění jako ochrana přetopení ze slunečního tepla)
- Osvětlení (ovládání bodového světla, LED pásků či Loxone závěsných světel)
- Multimédia (Ovládání audia, TV...)
- Stínění (Ovládání stínící techniky pomáhá při vytápění a chlazení v domě)
- A díky rozšířením také mnoho dalšího [32]

System loxone se skládá z několika různých prvků:

- Miniserver
- Rozšíření
- Příslušenství (Loxone Tree zařízení)
- Loxone Tree a Loxone Link kabeláž
- Aplikace Loxone App a Loxone Config

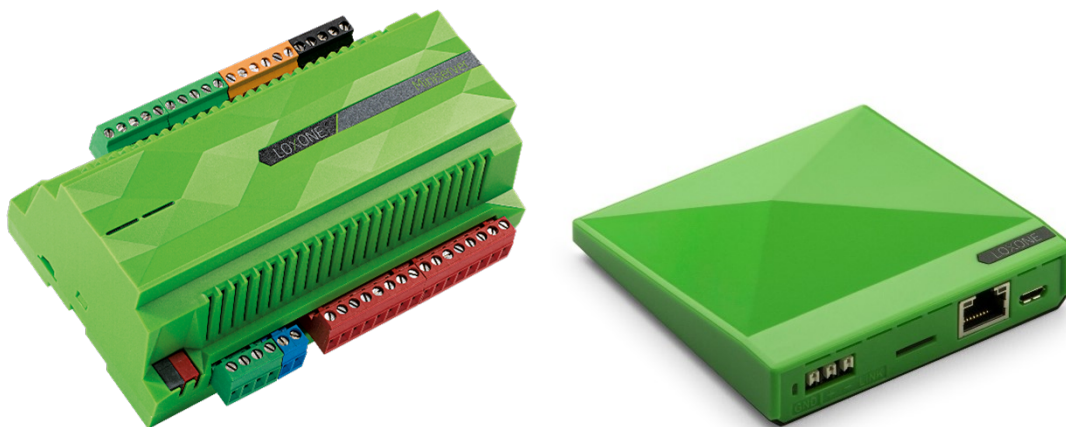
Loxone prvky ke své činnosti potřebují tzv. miniserver. Ten v systému funguje jako centrální řídicí jednotka, která se stará o automatizaci domácnosti. Loxone nabízí celkem 3 různé verze miniserveru:

- Miniserver Gen. 1
- Miniserver Gen. 2
- Miniserver Go

Miniserver 1. i 2. generace mají oba 8 digitálních a 4 analogové vstupy a 8 digitálních výstupů (relé spínající max 250VAC/30VDC). Miniserver 1. generace má ještě navíc 4 analogové výstupy [27]. Obě generace miniserveru slouží pro kabelovou komunikaci a jsou určeny k instalaci na DIN lištu. Také jsou obě generace vybaveny rozhraním Loxone Link (pro

kabelové připojení až 30 tzv. rozšíření) a LAN port (Fast ethernet). Pouze první generace obsahuje KNX rozhraní, naopak pouze druhá generace a verze Go obsahují již integrované rozhraní Loxone Tree (K první generaci je pro komunikaci po Loxone Tree sběrnici dodán rozšiřující modul) [28].

Pokud si uživatel přeje využívat bezdrátové komunikace mezi prvky systému Loxone (zejména Pro bezdrátové ovládání pak Loxone nabízí 3. verzi miniserveru – Miniserver Go. Ten komunikuje s bezdrátovými perifériemi (rozšířeními a příslušenstvím) rádiovou komunikací na frekvenci 868MHz pro SRD pásmo pro Evropu (na 4 kanálech), případně 915MHz pro ITU region 2 (10 kanálů), s maximálním výkonem 3.16 mW [26]. Obsahuje také LAN port (Fast ethernet) a rozhraní Loxone Link. K této verzi miniserveru je možné bezdrátově připojit až 128 periférií [31].



Obrázek 2.2: Loxone Miniserver gen. 1 a Miniserver Go. Převzato z Loxone web

Všechny verze Miniserveru v sobě obsahují Loxone OS s integrovaným webovým serverem, jsou konfigurovatelné z programu Loxone Config a ovladatelné přes mobilní aplikaci (Loxone App) [33]. Všechny miniservery obsahují slot pro SD kartu (s firmwarem).

Loxone Extensions (rozšíření) slouží pro rozšíření funkcí Miniserveru. K Miniserveru se připojují pomocí sběrnice Loxone Link (kterou obsahují všechny verze Miniserveru). Tato sběrnice může být až 500 m dlouhá. Díky rozšířením může uživatel zakoupit systém pouze s těmi technologiemi, které chce opravdu využívat a nemusí tak platit za zbytečné vlastnosti systému. Příkladem rozšíření mohou být Tree Extension (pro připojení až 100 Tree zařízení; zejména pro doplnění Miniserveru 1. generace, který neobsahuje rozhraní pro komunikaci přes tree sběrnici) [36], Air Base Extension (Pro doplnění Miniserverů 1. a 2. gen – k bezdrátové komunikaci) [37], Dimmer Extension (pro stmívání světel) [38] a mnoho dalších.

Loxone nabízí pro automatizaci domácnosti více než 400 produktů [33]. Loxone pro propojení prvků v systému vyvinulo tzv. Loxone Tree technologii. Jedná se o sběrnici, na kterou je možné připojit až 50 prvků, a Loxone uvádí, že díky tomu je možné ušetřit až 80% kabeláže [DOLOŽIT!]. Podobně jako Loxone Link, i Loxone Tree může sahát až 500 m daleko.

V oblasti inteligentního vytápění nabízí Loxone souhrn technologií pro vytápění, chlazení, rekuperaci, a automatizovanou stínící techniku, což přináší do regulace vytápění vysokou efektivitu [34].

Z hlediska regulace teploty nabízí tzv. „zónové“ vytápění. Jedná se o inteligentní topení, které na rozdíl od klasického inteligentního vytápění (zahrnující obvykle nějaký bezdrátový



Obrázek 2.3: Loxone App. Převzato z Loxone web

termostat, wifi termostatické hlavice apod.) umožňuje inteligentněji řídit teplotu – tím že uživatel zvolí, ve které místnosti (případně i ve který čas) má být jaká teplota. Uživatel chytrého domu s tímto systémem si tak může navolit například větší teplo v koupelně oproti například místnosti kde spí. Tento systém tak umožňuje mít větší kontrolu nad vytápěnými místnostmi, potažmo vyšší efektivitu.

Inteligentní vytápění Loxone podporuje režim učení, systém se tedy na základě předchozích zkušeností spustí vytápění tak, aby byla v dané místnosti požadovaná teplota ve správný čas. Uživatel si tak může nastavit například to, aby měl v 7:00 vyhřátou koupelnu na 23 °C.

Loxone vytápění má dle oficiálních stránek [25] následující výhodné vlastnosti:

- Inteligentní řízení teploty – využití již zmíněného režimu učení k dosažení požadované teploty v žádaný čas. Loxone rovněž při regulaci zohledňuje venkovní teplotu.
- Úspora nákladů – Loxone dokáže inteligentně rozhodovat o nejefektivnějším řešení. Například energeticky náročnou klimatizaci může nahradit energeticky výhodnějším stíněním
- Režim nepřítomnosti – Systém od loxone podporuje úsporný režim pro chvíle, kdy uživatel není doma
- Ochrana budovy – Loxone dokáže reagovat na různá nebezpečí, například v případě vzniku požáru vypnout ventilaci i rekuperaci
- Loxone aplikace a statistiky – Loxone nabízí zdarma aplikaci na zařízení s androidem přes které uživatel může sledovat i nastavovat teplotu v domě vzdáleně
- Notifikace – V případě problému s některou technologií Loxone upozorní uživatele

- Státní svátky – Na základě znalosti státních svátků může Loxone adekvátně upravovat svoji činnost
- Údržba – Loxone uživatele upozorňuje na termín pravidelné údržby

Z hlediska automatizace domácnosti v porovnání s dříve uvedenými systémy je rovněž důležitá přítomnost ovládané chytré zásuvky. Ta s Miniserverem komunikuje technologií Loxone Air. Má v sobě teplotní čidlo a rovněž elektroměr s vyhodnocením výkonu a spotřeby //ještě zmínit loxone touch a tlačítko na stul

Jablotron

Jablotron je česká firma, která se od svého založení zaměřuje především na zabezpečovací systémy [40]. Kromě nich se také zabývá zabezpečením a monitoringem vozidel, topením a ventilací, monitoringem dechu a rovněž ovládáním a automatizací domácnosti [39].

Jablotron nabízí několik různých systémů. Dva nejnovější jsou Jablotron 100 a Jablotron 100+ [40]. Primárním úkolem obou systémů je zabezpečení budov, ovšem je možné je využít i v oblasti automatizace (zejména díky programovatelným výstupům). Samotné zabezpečení je možné využít v rámci automatizace (Například automatické zapnutí světel při odkódování alarmu) [43].

Na své systémy poskytuje Jablotron při splnění podmínek až 7letou záruku [42].

Pro odjištění/zajištění systému se vždy musí provést nejprve autorizace uživatele. Systém totiž uchovává informaci o oprávnění jednotlivých uživatelů. Každému z uživatelů je možné pro účely autorizace přiřadit jeden kód (4,6 nebo 8místný) a až dva RFID čipy [44].

Typy automatizace, které je možné v těchto systémech použít jsou:

- Zapínání a vypínání
- Akce v kalendáři
- Automatické akce [47]

Mezi akce, které lze automatizovat v systému Jablotron patří zejména ovládání světel, ovládání žaluzií, chytrá termoregulace (řízení vytápění a klimatizace) [48] či ovládání jiných zařízení pomocí programovatelných výstupů.

Systém od Jablotronu je možné rozdělit na tyto různé části:

- Ústředna
- Různé vstupní či výstupní prvky
- Aplikace MyJablotron
- Program J-Link

Ústředna v systémech Jablotron slouží jako centrální prvek, který shromažďuje informace ze snímačů a patřičně na ně reaguje. Komunikace mezi prvky systému a ústřednou může probíhat podobně jako u systému Loxone buď pomocí kabelů, nebo bezdrátově. Zařízení, která komunikují pomocí kabelu se zde nazývají sběrnice [43].

Mezi produkty firmy Jablotron pro automatizaci domácnosti můžeme najít například:

- Záplavový detektor

- Snímač teploty
- Magnetický detektor (detekce otevření dvěří/okna)
- Termoelektrická hlavice
- Relé na DIN lištu
- A další [41]

Systém Jablotron 100+ je možné ovládat celkem 4 způsoby a to:

- Přístupovým modulem
- Mobilní aplikací pro chytré telefony (MyJABLOTRON)
- Webovou aplikací (rovněž MyJABLOTRON)
- Či klíčenkou [46]

Přístupový modul slouží pro rychlé odjištění/zajištění objektu, případně k dalším funkcím automatizace. Jablotron nabízí celkem 3 typy těchto modulů:

- Čtečka RFID karet
- Klávesnice se čtečkou RFID karet
- Klávesnice s displejem a čtečkou RFID karet

Ke každému z modulů je možné připojit až 20 segmentů. Ty obsahují popisek a dvě prosvětlená tlačítka. Jejich funkcí může být buďto zajištění/odjištění, signalizace stavu (například signalizace otevření garážových vrat) nebo ovládání zařízení v rámci automatizace (například žaluzií) [44]. Barvy prosvětlení odpovídají semaforu, kde červená odpovídá stavům jako zajištěno/zapnuto, žlutá zajištěno částečně a zelená znamená odjištěno/vypnuto. Jak již bylo zmíněno, systém od Jablotronu lze ovládat rovněž mobilní aplikací MyJablotron. Je k dispozici jak na Google Play (pro zařízení s androidem), tak i na App Store (pro iOS zařízení). Kromě toho existuje i její webová verze. Jablotron tak nabízí rychlý přehled o tom co se děje v domácnosti. Ovládání domácnosti přes aplikaci funguje podobným způsobem jako přístupový modul – pomocí tlačítek s barvami semaforu.

Klíčenka k ovládání systému je dostupná ve dvou verzích – jednosměrný a obousměrný ovladač. Ten druhý má výhodu v tom, že provedení akce je potvrzeno kontrolkou na ovladači. V případě chyby tak ví, že je například mimo dosah ústředny a akce se neprovedla [43].

K nastavení uživatelských parametrů v systému (jako oprávnění) slouží program J-Link. V něm je možné definovat uživatele i s jejich přístupovými oprávněními, provádět diagnostiku systému, kontrolu programovatelných výstupů a vytvářet či upravovat kalendář akcí (pro ovládání automatizovaných funkcí) [47].

Apple HomeKit

Apple HomeKit je systém, který umožňuje uživateli bezdrátově ovládat nejrůznější chytrá zařízení v domácnosti. Na rozdíl od systémů jako je Loxone je HomeKit orientován spíše na bezdrátovou komunikaci. V základu je systém založen na komunikaci pomocí Wi-Fi a Bluetooth. Pro podporu dalších přenosových technologií je potřeba do systému přidat tzv. bridge, který potom slouží jako prostředník mezi zařízeními. Zařízení v systému mezi sebou komunikují pomocí aplikačního protokolu HAP. Pokud dané koncové zařízení tento protokol nezná, tak s ním komunikuje bridge zařízení pomocí protokolu, kterému rozumí, takže bridge opět funguje jako prostředník [76].

Systém Apple HomeKit je možné ovládat pomocí chytrého telefonu iPhone. V tomto případě je však možné ovládání pouze v lokální síti. Druhou variantou je ovládání přes internet, ale to již vyžaduje použití nějakého bridge, tedy centrálního prvku (např. Apple TV či iPad). Pro ovládání systému pomocí telefonu s operačním systémem Android je potřeba mít v systému nějakého hlasového asistenta, který je se systémem Apple HomeKit kompatibilní (např. Amazon Alexa či Google Assistant) [10].

Mezi zařízení, která je možné v systému Apple HomeKit používat patří například nejrůznější tlačítka, světla, kamery, zásuvky a mnoho dalšího, viz. seznam na stránkách společnosti Apple³. Systém Apple HomeKit představuje otevřený systém, do kterého postupně přibývají nové prvky, přičemž se nejedná jen o zařízení od společnosti Apple. Obchodníci s těmito zařízeními pak uvádějí u konkrétních zařízení, zda jsou s Apple HomeKit kompatibilní, pokud tak tomu je [71].

Po přidání zařízení do systému (v aplikaci Domácnost) je možné zařízení ovládat jednak okamžitým nastavováním hodnoty [37], ale také je možné využít automatizace, která umožňuje zakomponovat některou z následujících situací:

- Někdo přijde
- Někdo odejde
- Nastane určitá denní doba
- Změna stavu příslušenství
- Čidlo něco detekovalo [38]

Na zařízení s operačními systémy iOS, iPadOS, watchOS či macOS je možné je ovládat pomocí aplikace Domácnost⁴. Případně je také možné použít aplikaci Zkratky či ovládání pomocí hlasového asistentovi Siri [39]. Nakonec je samozřejmě také možné jednotlivá chytrá zařízení ovládat pomocí aplikace od jejich dodavatele [37].

Sonoff

Sonoff je bla bla. . .

Home connect

Poněkud jiný přístup k chytré domácnosti přináší technologie Home connect. Tou jsou vybavena některá zařízení, například od společnosti Bosh⁵. Předchozí zmíněné systémy slouží

³<https://www.apple.com/ios/home/accessories/>

⁴<https://support.apple.com/cs-cz/guide/iphone/iph22d98bbca/ios>

⁵<https://www.bosch-home.com/cz/novinky/home-connect>

k ovládání a automatizaci spíše jednoduchých prvků domácnosti, jako jsou světla, zásuvky apod. Naproti tomu technologie Home connect je přítomna v některých spotřebičích, jako je například pračka, lednice či kávovar. Aplikace, která s Home connect přichází je dostupná například na operační systém Android, ale také pro hlasové asistenty Amazon Alexa či Google Assistant. Díky aplikaci je možné získávat některé informace o spotřebičích (např. zda nejsou dveře lednice pootevřené) a také zařízení ovládat (např. na dálku zapnout troubu a předehřát ji tak) [26].

Virtuální hlasoví asistenti a centrální prvky chytré domácnosti

Virtuální osobní asistent (VPA) je osobní asistent, který zajišťuje interakci mezi uživatelem chytré domácnosti a zařízeními v ní. Jako jiné označení se rovněž používá inteligentní, digitální osobní, či mobilní asistent. Je-li ovládaný hlasem, pak se někdy označuje jako hlasový asistent [6]. Dále v textu této kapitoly je vždy asistentem míněn právě hlasový asistent, nebude-li specifikováno jinak. Jedná se o software, jehož úlohou je asistovat uživateli při nejrůznějších příležitostech, mezi jinými i při ovládání domácnosti. Hlasoví asistenti běží na některém zařízení s reproduktorem a mikrofony nebo mobilním zařízením. Dnes jich existuje na trhu velké množství (zejména pro chytré telefony). Mezi nejznámější představitele hlasových asistentů v současné době patří:

- Apple Siri
- Amazon Alexa
- Microsoft Cortana
- Google Assistant [2]
- Samsung S voice
- Facebook M
- Nuance dragon [6]

Podobně jako systém Apple HomeKit mají obecně i hlasoví asistenti (resp. zařízení, která je obsahují) funkci centrálních jednotek, které spojují jednotlivá zařízení. Při výběru chytrých zařízení pro automatizaci domácnosti je tedy vhodné se ujistit, zda podporuje daného hlasového asistenta⁶.

V současné době žádný z výše uvedených hlasových systémů nepodporuje češtinu, nicméně Assistant od společnosti Google by ji v dohledné době mohl podporovat [49][14]. Amazon Alexa pak sice nerozumí česky, ale již dokáže číst některé knihy v češtině. Pro českého uživatele jsou tak pouze 2 možnosti – buďto používat asistenta v angličtině a spokojit se s případnou absencí některých funkcí, které nejsou v Česku podporované (například ovládání domácnosti), nebo použít některého českého virtuálního asistenta, ovšem s omezenou funkcionalitou oproti jejich vyvinutějším protějškům. Mezi české hlasové asistenty patří například:

- Emma
- Intelli

⁶<https://www.alza.cz/jak-postavit-chytrou-domacnost>

Každý z virtuálních asistentů má své vlastní specifikace, ovšem jsou některé úlohy, které je možné považovat za typické:

- Číst a psát SMS a emailové zprávy, uskutečňovat hovory
- Nastavovat časové a kalendářové akce (časovače, upomínky. . .)
- Odpovídat na některé základní informativní otázky (počasí, čas, převody jednotek. . .)
- Ovládat média jako televizi či připojené reproduktory (pouštět filmy, hudbu)
- Vyprávět vtipy a příběhy
- Konečně ovládat prvky chytré domácnosti [2]

Používání virtuálních osobních asistentů nejen, že umožňuje přistupovat k různým úkolům inovativním a interaktivním způsobem, ale v mnoha případech i zjednodušuje jinak relativně zdlouhavou činnost. Dobrým příkladem je například manuální nastavení budíku (bez použití VPA). Na mobilním telefonu (Nexus 5) je dle [6] potřeba vykonat následující akce:

1. Kliknout na tlačítko pro návrat na domovskou obrazovku (pokud se tam uživatel nenachází)
2. Kliknout na ikonu hodin
3. V otevřené aplikaci najít ikonu budíku a kliknout na ni
4. Kliknout na tlačítko „+“ pro přidání budíku
5. Nastavit hodinu, překliknout na volbu minuty a nastavit minuty
6. Potvrdit kliknutím na tlačítko „OK“

Při použití hlasového asistenta je celá úloha značně zredukována pouze na aktivování asistenta a vyslovení požadovaného úkolu.

Bez použití VPA je dokonce řada úkolů nerealizovatelná. Například připomenout či udělat něco v okamžiku, kdy se uživatel vrátí domů, což je funkce, kterou někteří virtuální asistenti podporují [16]. Interaktivitu zajišťují virtuální asistenti i při automatizaci domácnosti, pro její řízení není potřeba otevírat k tomu určené (a mnohdy jednoúčelové) aplikace, ale stačí vyslovit žádost, třeba i s jistou vzdáleností od zařízení s hlasovým asistentem a ten se již o vše postará. Kromě toho v sobě virtuální asistenti mohou mít i funkce sloužící přímo pro automatizaci domácnosti. Například nastavení podmínek, scénářů atd. Virtuální asistenti díky svým funkcím a vlastnostem k chytré domácnosti neodmyslitelně patří, ovšem je nutné si uvědomit, že zde nejsou nutností. Spíše často fungují jako prostředník mezi uživatelem a chytrými zařízeními, který usnadňuje řízení domácnosti. Často pak bývají zabudováni do chytrého zařízení, plní funkci centrálního prvku (hubu), jako například v případě zařízení Echo od společnosti Amazon.

Emma je český hlasový asistent, kterého vytvořil David Beck pomocí aplikace Zkratky (na systému iOS). Jak již bylo zmíněno, nativním virtuálním asistentem pro iPhone je Siri, ta však neumí česky, a to se rozhodl David Beck změnit [20]. Nejedná se o samostatnou aplikaci, ale o zkratku v aplikaci Zkratky na systému iOS. Tato aplikace zkratky umožňuje uživatelům sloučit různé akce do jedné zkratky. Celkově pro zkratku Emma nastavil 7 tisíc akcí a další se chystá přidávat. Systém v současné době již podporuje češtinu, částečně slovenštinu a polštinu. Plánuje také přidat maďarštinu, řečtinu a rumunštinu [21].

Kapitola 3

Protokoly a technologie používané v automatizované domácnosti

Následující část je shrnutím protokolů a technologií, používaných systémy automatizace domácnosti. Není encyklopedickým výkladem problematiky, ale souhrnem informací, které mají k práci bezprostřední vztah. První část se věnuje protokolům CoAP a MQTT, jelikož se jedná o nejpoužívanější protokoly pro IoT. Následuje obecný úvod k technologii bezdrátového přenosu a nakonec uvádím krátké seznámení s technologiemi Wi-Fi, Bluetooth a ZigBee. Samotné technologii Wi-Fi věnuji samostatnou podkapitolu, jelikož ji později využívám v realizační části práce.

3.1 Komunikační protokol CoAP

CoAP protokol je protokol, určený pro komunikaci mezi uzly (zařízeními) s omezenými zdroji v omezené síti (tedy např. s vysokou ztrátovostí paketů). Je definován v RFC 7252⁷. Tento protokol je speciálně navržený zejména pro tzv. machine-to-machine komunikaci v rámci internetu věcí. CoAP protokol se svojí strukturou podobá aplikačnímu protokolu HTTP, ačkoli každý z těchto protokolů má jiné zaměření. Mezi velkou výhodou CoAP protokolu patří jeho malá režije a jednoduchost [23].

Používané principy

Jak už bylo zmíněno, tento protokol je velmi podobný protokolu HTTP. Podobně jako při komunikaci přes HTTP, i v CoAP existují 2 role, které zařízení může zastávat - klient a server. Klient posílá požadavek na server a ten vrací odpověď. Na rozdíl od HTTP je však při použití CoAP běžné, že zařízení vystupuje v obou rolích (vzhledem k povaze M2M interakce). V jednu chvíli tedy může zařízení jako klient posílat požadavky jinému zařízení a následně samo v roli serveru obsluhovat klienty. Dalším rozdílem oproti HTTP je rovněž to, že se jedná o asynchronní protokol (používá UDP protokol). Z toho důvodu je nutné nějakým způsobem zaručit doručení dat (pokud je to pro danou aplikaci nutné). CoAP protokol proto definuje 4 typy zpráv:

- Confirmable (potvrditelná)
- Non-confirmable (nepotvrditelná)

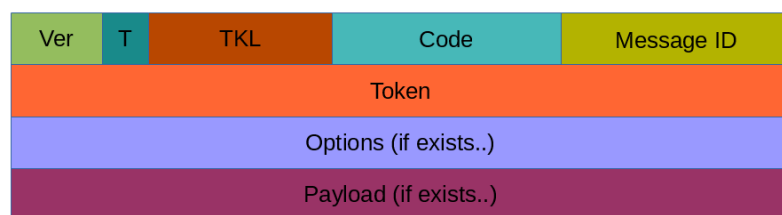
⁷<https://tools.ietf.org/html/rfc7252>

- Acknowledgment (potvrzení)
- Reset [69]

Jelikož je CoAP protokol navržen pro ztrátovou síť, počítá se s jistými ztrátami jednotlivých zpráv. Protokol tedy rozlišuje mezi potvrditelnou a nepotvrditelnou zprávou. Pokud příjemce obdrží potvrditelnou zprávu, musí odpovědět zprávou typu acknowledgment (potvrzení). V případě že vyprší časový limit a příjemce neobdrží tuto odpověď, znovu odešle původní zprávu. Toto provádí, dokud není dosaženo maximálního počtu opakovaných přenosů [59]. Tento počet je definován dokumentem RFC 7252 v rámci přenosových parametrů a je roven číslu 4, ale je možné jej (spolu s ostatními parametry) upravit pro danou aplikaci. Odesílatel vyhodnotí případnou přijatou zprávu jako odpověď podle identifikátoru zprávy, který se posílá spolu se zprávou. Ten musí být v odpovědi stejný jako v původním požadavku. Pokud zařízení posílá nepotvrditelnou zprávu, neočekává se, že příjemce pošle zpět potvrzení. Zpráva typu reset se používá, pokud chce příjemce odesílateli sdělit, že potvrditelnou zprávu nedokáže zpracovat, resp. ani vrátit vhodnou chybovou odpověď [69].

Formát CoAP zprávy

Formát CoAP zprávy je možné vidět na obr. 3.1.



Obrázek 3.1: Formát CoAP zprávy⁸

CoAP zpráva je zakódovaná pomocí binárních dat. Začíná povinnou 4-bitovou hlavičkou, obsahující informaci o:

- Verzi CoAP protokolu (Ver) - 2 bity
- Typ zprávy - 2 bity

Verze protokolu musí být nastavena na 1 (zbylé možnosti jsou určeny pro budoucí verze protokolu). Typ zprávy je myšlen z hlediska potvrditelnosti, jak bylo definováno výše. Jednotlivým hodnotám odpovídají následující typy:

- 0 - confirmable (potvrditelná)
- 1 - non-confirmable (nepotvrditelná)
- 2 - acknowledgment (potvrzení)
- 3 - reset

⁸<https://www.survivingwithandroid.com/wp-content/uploads/2018/11/coap-message-format.png>

Po typu zprávy je ve zprávě specifikována délka přítomného tokenu (4 bity). Na základě této délky může mít token 0 - 8 bajtů (délka 9 - 15 je rezervována a nesmí být nastavená). Následuje kód zprávy, kterým může být buď metoda požadavku (CoAP je totiž podobně jako HTTP postaven na REST [29] architektuře), nebo stavový kód odpovědi, který má opět podobný účel, jako stavové kódy u HTTP protokolu. Tento kód má délku 8 bitů. Poslední součástí hlavičky CoAP zprávy je identifikátor zprávy, který má délku 16 bitů. Používá se k identifikaci duplikátních zpráv a rovněž ke spojení potvrditelné zprávy s potvrzením (resp. reset zprávou). Po hlavičce následuje token, který má délku 0 - 8 bajtů, jak bylo vysvětleno dříve. Tento token se používá ke spojení požadavku a odpovědi. Při komunikaci totiž může odesílatel poslat požadavek na server, na který sice může server ihned odpovědět potvrzením (zde tedy posílá stejný identifikátor zprávy, jaký měl požadavek), ale daný požadavek vyřeší až později. A tehdy posílá server klientovi stejný token, jako obdržel v původní zprávě od klienta, ačkoli identifikátor zprávy již bude jiný. To je tedy rozdíl mezi identifikátorem a tokenem. Po tokenu následují tzv. options, tedy volby, které mají podobnou funkci, jako některé hlavičky v HTTP protokolu. Tyto volby mohou mít různou délku a rovněž jejich počet může být různý (ani nemusejí být přítomny). Poslední částí CoAP zprávy tvoří nepovinný datový obsah (payload).

3.2 MQTT protokol

MQTT (Message Queuing Telemetry Transport protokol) je aplikační protokol, určený podobně jako CoAP pro přenos dat v nestabilní síti. Na rozdíl od CoAP se však nejedná o protokol typu klient – server, ale používá tzv. publish–subscribe mechanismus [25]. Jednotlivé uzly v síti mohou tedy mít jednu ze tří funkcí:

- Publisher
- Subscriber
- MQTT broker

Publish–subscribe mechanismus

Mechanismus publish–subscribe je založen na tom, že v systému existují zařízení, která sdílí nějaké údaje (publish). V síti je přítomen MQTT broker, což je zařízení, které má funkci centrálního prvku, přes který se odehrává komunikace. Pokud chce nějaké zařízení naslouchat novým informacím od jiného zařízení, musí se (u brokeru) přihlásit k odběru jeho zpráv. Stane se tak z něj subscriber daných zpráv. Identifikace zařízení (resp. zpráv), kterým bude naslouchat se pak provádí přes tzv. topic (téma). To specifikuje cestu k danému zdroji a má hierarchickou strukturu. Může mít podobu například "domov/kuchyně/světlo". Jelikož jsou témata v MQTT zprávě zakódována v UTF-8, mohou používat dokonce diakritiku. Pokud naopak nějaké zařízení potřebuje naslouchat na změny od nějakého zařízení, přihlásí se u brokeru k odběru daných zpráv, které opět specifikuje cestou, na které bude naslouchat. Pokud následně zařízení typu publisher získá nějaký údaj, který sdílí (např. hodnotu měřené veličiny), odešle zprávu MQTT brokeru. Ten zprávu přepoše všem zařízením, které jsou přihlášeny k odběru daného tématu. Jedno zařízení může samozřejmě vystupovat pro určitá témata jako publisher a pro jiná subscriber [62].

Quality of Service

Jelikož je protokol určen pro funkci i v nestabilní síti, definuje 3 stupně tzv. QoS (Quality of Service):

- 0 - At most once
- 1 - At least once
- 2 - Exactly once [21]

Stupeň 0 říká, že odesílatel zprávu odešle a zahodí, nekontroluje tedy, zda zprávu přijali příjemci. V podstatě se jedná o obdobu nepotvrditelné zprávy u CoAP protokolu. Stupeň 2 rozhoduje, že odesílatel po odeslání zprávy čeká na potvrzení od příjemce že zprávu dostal. V případě že vyprší stanovený časový limit a toto potvrzení neobdrží, tak zprávu odesílá znovu. Poslední stupeň se používá, pokud chce mít odesílatel jistotu, že příjemce dostal (resp. zpracoval) zprávu přesně jednou. K tomuto účelu se používá čtveřice zpráv. Nejprve odesílatel posílá sdílená data. Následně čeká na potvrzení. Pokud od příjemce nepříjde potvrzení, posílá zprávu znovu se značkou, definující, že se jedná o duplicitní zprávu. Po přijetí potvrzení posílá příjemci další zprávu jako potvrzení, že obdržel potvrzení. Nakonec příjemce posílá zpět zprávu pro potvrzení, že byl přenos dat úspěšně dokončen [72].

3.3 Bezdrátový přenos dat a signálů

Pro přenos dat či řídicích signálů je vždy potřeba zvolit vhodné médium, přes které se budou tyto informace přenášet. V některých situacích není pro přenos vhodné (a někdy dokonce ani možné) používat kabely (ať už metalické nebo optické). V těchto případech je potřeba přenášet informace bezdrátově, tj. za využití jiných médií, jako je vzduch. Podobně jako je nutné u kabelového spojení využít vhodný způsob komunikace (například zvolit vhodnou sběrnici a nastavit ji správné parametry) je potřeba se způsobem komunikace zabývat rovněž u bezdrátového přenosu. Zde je nutné zejména zvolit vhodnou technologii (jako je Wifi, Bluetooth či ZigBee) a její parametry [A].

Výhody bezdrátového přenosu

Bezdrátová komunikace má oproti kabelové řadu výhod. Zejména se jedná o následující:

- Jednodušší připojení – zařízení není potřeba připojovat kabelem, a dokonce nemusí být ani vybaveno konektorem pro toto připojení (pozn. pro dálkový přenos prostřednictvím světla je však stále potřeba mít nějaký přijímající port). Z toho rovněž plyne, že není potřeba měnit strukturu sítě kvůli změnám v místnosti a rovněž není potřeba myslet na konkrétní strukturu sítě ještě před budováním.
- Větší spolehlivost – Častým zdrojem problémů s kabelovým připojením jsou chyby na straně kabelů – jejich poškození. Použitím bezdrátových technologií se lze vyhnout tomuto typu chyb.
- Snadná rozšiřitelnost sítě – U kabelového připojení je potřeba řešit způsob rozšíření sítě a v případě, že stávající struktura sítě rozšíření nepodporuje, tak je potřeba ji celou pozměnit. Bezdrátové sítě tento problém eliminují.

- Nižší cena – Použitím bezdrátových technologií se značně sníží pořizovací cena sítě – není potřeba kupovat drahou kabeláž. Rovněž instalace kabelů do starých budov může být velmi nákladná a problémová.

Nevýhody bezdrátového přenosu

Kromě množství výhod, které bezdrátová komunikace představuje jsou zde rovněž některé nevýhody tohoto typu komunikace:

- Rušení signálu – zařízení, využívající bezdrátové technologie může způsobovat rušení ostatních zařízení a rovněž opačně – dané zařízení může být rušeno od ostatních zařízení, pracujících na podobném principu
- Bezpečnost – bezdrátová komunikace často vysílá (a přijímá) signály do relativně rozsáhlého otevřeného prostoru, tudíž jsou takto vysílaná data často daleko méně chráněná než u kabelového přenosu (kde je k získávání dat potřeba mít fyzické připojení k síti, ve které se data přenáší) [A] [Q, str. 5-6] [R, str. 406]. Je tedy nutné zabezpečit přenos dat.

3.4 Technologie bezdrátového přenosu WiFi

Wifi je technologie, využívající standardů z rodiny IEEE 802.11. První verze tohoto standardu byla organizací IEEE schválena v roce 1977 [A, s.6]. Od té doby vyšlo mnoho dalších verzí standardů. Jednotlivé verze se od sebe mohou odlišovat různými parametry, například frekvenčním pásmem, šířkou pásma jednotlivých kanálů, maximální rychlostí přenosu atd. Organizace Wi-Fi Alliance rozlišuje některé standardy IEEE 802.11 číslem generace WiFi, nejnovější je prozatím zatím 6. generace (založená na standardu 802.11ax).

IEEE 802.11 PHY Standards							
Release date	Standard	Frequency Band	Bandwidth	Transmission Scheme	Max Modulation	MIMO	Max Data Rate
1997	802.11	2.4 GHz	20 MHz	DSSS, FHSS	QPSK	N/A	2 Mb/s
1999	802.11b	2.4 GHz	20 MHz	DSSS	QPSK	N/A	11 Mb/s
1999	802.11a	5 GHz	20 MHz	OFDM	64QAM	N/A	54 Mb/s
2003	802.11g	2.4 GHz	20 MHz	DSSS, OFDM	64QAM	N/A	54 Mb/s
2009	802.11n	2.4 GHz 5 GHz	20 MHz 40 MHz	OFDM	64QAM	4x4	600 Mb/s
2013	802.11ac	5 GHz	20 MHz 40 MHz 80 MHz 160 MHz	OFDM	256QAM	8x8	6.93 Gb/s
2018	802.11ad	60 GHz	2160 MHz	SC, OFDM	256QAM	Beamforming	6.93 Gb/s



Obrázek 3.2: Některé důležité verze standardu IEEE 802.11 a jejich parametry. Převzato z <https://www.grandmetric.com/2018/05/29/wi-fi-standards-evolution/>

Wifi funguje na principu vysílání a přijímání rádiových vln. Organizace IEEE rozhodla využít pro technologii Wi-Fi frekvence z pásma ISM [B, str.2]. Wifi standardně využívá

frekvencí 2,4Ghz a 5Ghz. Nejprve byla zařízení Wi-Fi schopná pracovat pouze v jednom z těchto dvou frekvenčních pásem, ale 4. generace (IEEE 802.11n) přidává možnost práce v obou zmíněných pásmech. Moderní zařízení s wifi si tak mohou vybrat (a dokonce během své činnosti měnit) frekvenci, na které budou spolu komunikovat. Obě pásma mají svá pro i proti. Mezi výhody pásma 2.4Ghz patří zejména větší pokrytí signálu a rovněž větší kompatibilita (platí spíše pro starší zařízení). Na druhou stranu pásmo 5Ghz nabízí podstatně vyšší přenosové rychlosti a dále větší množství komunikačních kanálů [D].

Režim sítě

Wifi nachází uplatnění v (bezdrátových) lokálních sítích. V nich pak rozlišujeme 3 režimy na základě toho, jak se Wifi zařízení v síti mezi sebou navzájem spojují (jakou plní roli):

- Režim infrastruktury
- Ad hoc režim
- Smíšený režim

V režimu infrastruktury je v síti přítomen minimálně jeden centrální prvek (tzv. přístupový bod), který zprostředkovává komunikaci mezi jednotlivými prvky (klienty) sítě, případně poskytuje připojení do jiné sítě přes distribuční systém (DS). V tomto režimu sítě je výhoda, že je snadné připojit do stávající infrastruktury nový prvek.

Ad hoc je režim bezdrátové sítě, ve které není přítomen žádný centrální prvek (přístupový bod) se kterým by prvky sítě komunikovali, ani zde není žádné spojení se pevnou sítí přes distribuční systém. Jedná se tedy o decentralizovanou síť. Jednotlivé prvky tedy mezi sebou navzájem komunikují přímo (toto spojení se někdy označuje jako tzv. peer-to-peer). V tomto režimu má síť rovněž SSID identifikátor, kterým je možné síť identifikovat. [A][B]

3.5 Ostatní technologie bezdrátového přenosu

Technologie Bluetooth

Bluetooth je standard, definovaný v IEEE 802.15.1. Vytvořila jej firma Ericsson v roce 1994 a od té doby vyšlo několik nových verzí [A]. Podobně jako WiFi pracuje v ISM pásmu 2,4 GHz. Na rozdíl od Wi-Fi však není definován pouze na prvních dvou vrstvách ISO/OSI, ale definuje protokoly na všech sedmi vrstvách tohoto modelu. Na nejnižší úrovni, kde definuje způsob přenosu jednotlivých bitů využívá metodu FHSS, která zajišťuje, že při přenosu bitů vysílač přeskakuje mezi několika frekvencemi [AD].

Zařízení, které jej využívají, umožňuje vytvořit tzv. PAN (osobní síť). V těchto sítích má každé zařízení přiřazeno unikátní 48bitovou adresu BD_ADDR (BlueTooth Device Address) – jedná se o obdobu MAC adresy u ethernetu. Tu používá pro komunikaci s ostatními zařízeními. Jedno zařízení může být v roli master (řídící), slave (podřízená) nebo obojího [AB, str.4]. K jedné řídící stanici se připojuje jedno a více podřízených zařízení (používá se pouze adhoc komunikace mezi master a slave stanicí). Zde hovoříme o tzv. piconetu (pikosíti). Maximální počet zařízení v jedné pikosíti je 8 (jedna řídící stanice a až 7 podřízených). Stanice náležící do jedné pikosítě může zároveň patřit do jiné pikosítě. Jedná se tedy o rozšíření sítě mezi zařízeními. Takto vytvořenou síť nazýváme tzv. scatternet (rozprostřená síť). V každé rozprostřené síti má každá pikosíť unikátní identifikátor – je jím

BD_ADDR její řídicí stanice. Díky rozlišení jednotlivých pikosítí pak může každá tato síť využívat jiné skokové sekvence (frekvenčních kanálů na kterých se vysílají/přijímají data) [AC, str. 20]. Výhodou Bluetooth technologie je její nízká spotřeba energie (zejména od verze 4) [17].

Technologie ZigBee

ZigBee je bezdrátová technologie, založená na standardu IEEE 802.15.4. Je určena pro vytváření sítě PAN (osobní síť) a pracuje v pásmu ISM 868 MHz, 902-928 MHz a 2,4 GHz [A].

Zařízení v ZigBee síti

ZigBee standard specifikuje 2 typy zařízení – FFL (Full Function Device) a RFD (Reduced Function Device). FFL zařízení je obvykle schopné mnoha funkcí a je stále aktivní, zatímco RFD se nachází většinu času v režimu spánku, ze kterého se občas probudí, například aby odeslalo hodnoty neměřené na nějakém senzoru.

V síti pak každé ze zařízení plní některou ze 3 funkcí:

- Koordinátor
- Koncové zařízení
- Směrovač

Topologie ZigBee sítě

Na základě definovaných zařízení pak existují 3 možné topologie ZigBee sítě:

- Hvězda
- Strom
- Mesh síť [AG, str.5]

ZigBee Model

ZigBee podobně jako Bluetooth definuje komunikaci na všech úrovních modelu ISO/OSI, nekopíruje však přesně jednotlivé vrstvy. První 3 vrstvy modelů ISO/OSI a ZigBee si odpovídají, ale vrstvy L4-L7 jsou spojené do vrstev APS (Application Support) a ZDO (ZigBee Device Object). [AH, str. 42]

Thread, WeMo, ZigBee and Z-Wave (<https://www.tomsguide.com/us/smart-home-wireless-network-primer,news-21085.html>)

Kapitola 4

Vestavné systémy

Následující část je shrnutím současného stavu v oblasti vestavných systémů a Python knihoven pro vývoj GUI aplikací. Není encyklopedickým výkladem problematiky, ale souhrnem informací, které mají k práci bezprostřední vztah. Nejprve je zde úvod do vestavných systému, následně je pojednáno o platformě Raspberry Pi, modulech ESP8266 a ESP32 a nakonec o možnostech programování GUI pomocí Python knihoven.

4.1 Vestavné systémy, mikropočítače a mikrokontrolery

Vestavný systém můžeme definovat jako software spolu s počítačem, zabudovaným do nějakého zařízení takovým způsobem, že jej uživatel nevidí jako počítač [E, str.3]. Tento počítač je většinou jednoúčelový, určený pro předem navržené použití. Tím se liší od univerzálních počítačů, které mohou poskytovat různé funkce a jejichž uplatnění se může měnit (například osobní počítač) [F, str. 3].

[[Definovat mikropočítač, jednodeskový počítač, mikrokontrolér, mikroprocesor... (možná kapitola něco jako základní pojmy u vestavných systémů??)]

[[Zkusit něco vytvořit z nějakého takového souhrnu: <https://www.guru99.com/embedded-systems-tutorial.html>]] **[[<https://jayconsystems.com/blog/microprocessor-vs-microcontroller-vs-microcomputer>]]** **[[Historie smazána, vrátit ji? (Ale upravit!)]]**

Architektura řídicího zařízení

Při návrhu systému se využívá jedna ze dvou architektur:

- Von Neumannova architektura
- Harvardská architektura

Hlavním rozdílem je způsob práce s pamětí. Ve Von Neumannově architektuře je paměť pro program i data spojena do jedné paměti, v Harvardské architektuře je pak rozdělena. Mikroprocesor (CPU) je programovatelné elektronické výpočetní zařízení, určené pro všestranné použití. Jedná se o čip, obsahující 3 základní součásti:

- Aritmeticko-logickou jednotku
- Řídicí jednotku
- Registry [I, str. 18–19]

Mikroprocesor sám o sobě je z hlediska vestavěných systémů relativně jednoduché zařízení, které pro funkci systému potřebuje připojit některé další součásti, jako jsou paměti (RAM a ROM), čítače, časovač a podobně. Návrhář tedy musí tyto součásti přidat externě, aby zařízení fungovalo správně. Systémy, zahrnující mikroprocesory jsou obvykle založeny na Von Neumannově architektuře [J].

Mikrokontrolér je zařízení, které na rozdíl od mikroprocesoru má již všechny součásti, potřebné pro svoji činnost v sobě. Obvykle využívá Harvardské architektury [J].

4.2 Jednodeskový počítač Raspberry Pi

Zařízení Raspberry Pi je levný univerzální počítač malých rozměrů. Poskytuje široké možnosti v oblasti multimédií a 3D grafiky, předpokládá se, že bude časem využíván i jako herní platforma.

Název Raspberry Pi vytvořila komise dozorčí rady. Slovo Raspberry je vzato jako název ovoce (malina), jak už je u počítačových systémů zvykem nazývat podle ovoce. Slovo “Pi” označuje zkráceně “Python” - programovací jazyk, který měl být původně jediným programovacím jazykem dostupným na platformě Raspberry Pi [AA].

Historie

Raspberry Pi vzniklo v r. 2006 za přispění studijního ředitele pro informatiku na Cambridgeské univerzitě za účelem lokálních potřeb. Měl to být nástroj, který by poskytl prvotní impuls studentů k nějakému z univerzitních kurzů.

4.3 Ostatní mikropočítače

4.4 Moduly ESP8266, ESP32 a ESP32-S2

[[<https://maker.pro/esp8266/tutorial/a-comparison-of-the-new-esp32-s2-to-the-esp32>]] Jedná se o levný mikročip, disponující Wi-Fi stackem, schopný provozu RTOS (realtime operačního systému). Je založen na 32bitovém procesoru s architekturou RISC [AE]. ESP32 je nástupce ESP8266. Kromě komunikace přes Wi-Fi umožňuje rovněž komunikaci pomocí Bluetooth, díky hybridnímu Wi-Fi/Bluetooth čipu [AF].

[[Další moduly, viz: <https://www.cnx-software.com/2020/01/17/realtek-rtl8720dn-dual-band-wifi-bluetooth-5-0-iot-module/>]]

4.5 Ostatní mikrokontrolery a vývojové desky

[[Popsat arduino (jakožto nejznámější), STM32 a dále nějaké wifi/bluetooth, např.: <https://www.cnx-software.com/2020/01/17/realtek-rtl8720dn-dual-band-wifi-bluetooth-5-0-iot-module/>]] [[Tady ty nejsou s ESP: <https://www.microchip.com/design-centers/wireless-connectivity/embedded-wi-fi>]]

⁹Převzato z <https://www.cnx-software.com/2016/03/25/esp8266-and-esp32-differences-in-one-single-table/>

Specifications	ESP8266	ESP32
MCU	Xtensa® Single-Core 32-bit L106	Xtensa® Dual-Core 32-bit LX6 600 DMIPS
802.11 b/g/n Wi-Fi	Yes, HT20	Yes, HT40
Bluetooth	None	Bluetooth 4.2 and below
Typical Frequency	80 MHz	160 MHz
SRAM	160 kBytes	512 kBytes
Flash	SPI Flash , up to 16 MBytes	SPI Flash , up to 16 MBytes
GPIO	17	36
Hardware / Software PWM	None / 8 Channels	1 / 16 Channels
SPI / I2C / I2S / UART	2/1/2/2	4/2/2/2
ADC	10-bit	12-bit
CAN	None	1
Ethernet MAC Interface	None	1
Touch Sensor	None	Yes
Temperature Sensor	None	Yes
Working Temperature	- 40°C – 125°C	- 40°C – 125°C

Obrázek 4.1: Srovnání specifikace modulů ESP8266 a ESP32⁹

4.6 Možnosti bezdrátového přenosu ve vestavných systémech

[[Wifi, bluetooth, 433mht, 2,4GH, zigbee]]

[[Přidat podkapitolu o Firebase?]]

Kapitola 5

Zhodnocení současného stavu a plán práce

V této kapitole se věnuji zhodnocení již existujících řešení automatizace domácnosti. Následně uvádím můj návrh řešení na základě nastudovaných řešení a vhodného rozsahu práce. Nakonec v bodech stanovuji cíle vyplývající z návrhu řešení, které se v práci snažím splnit.

5.1 Zhodnocení současného stavu v oblasti automatizace domácnosti

Na trhu se v současné době nachází velké množství systémů. Z těch, které jsem popsal v části o existujících řešení je nejrozvinutějším systémem ten od společnosti Loxone. Zabírá opravdu širokou škálu možností a jen stěží by se hledala aplikace, pro kterou by nebyl vhodný. Kromě komplexnosti u něj oceňuji rovněž českou jazykovou lokalizaci. V češtině je k dispozici jak aplikace na ovládání (Loxone App), tak rovněž program pro konfiguraci systému (Loxone Config). Čím mě Loxone mile překvapilo je, že jsem si jejich aplikaci Loxone App mohl vyzkoušet v demoverzi i bez zakoupených komponent.

Jako nevýhodu Loxone vidím příliš vysokou cenu. Uživatel, který si chce nainstalovat pár chytrých zařízení bude zřejmě překvapen cenou. Například při pořízení 3 chytrých zásuvek a miniserveru (který je k ovládání zásuvek potřebný) zaplatí přibližně 15 000 Kč. Přitom adekvátní řešení od jiných firem, jako Sonoff bude stát necelé 3000 Kč, což je velký rozdíl – a při rozšiřování domácnosti o další prvky tento rozdíl znatelně roste. Na druhou stranu, pokud uživatel staví nový dům, může řešení od Loxone stát srovnatelnou cenu, jako konkurenční „neinteligentní“ instalace. Jako další nevýhodu vidím to, že celkově je instalace systému orientovaná spíše pro profesionální montáž pro pracovníky s příslušnou klasifikací (většina produktů je určena k zabudování do rozvaděče, příp. ke komunikaci s moduly v něm). Celkově je však řešení od Loxone na hodně vysoké úrovni.

Řešení od firmy Jablotron je jistě zajímavé jejich dvoutlačítkovými (rozšiřitelnými) segmenty. Zdá se mi však nepraktické spojovat přístupovou klávesnici do domu s prvky automatizace domácnosti. Působí to poněkud omezeně. Navíc rozhraní pro ovládání domácnosti a alarmu v aplikaci MyJablotron se snaží napodobovat onu klávesnici, což příliš k přehlednosti nepřispívá. Na druhou stranu pro uživatele, jehož hlavní požadavek je zabezpečení objektu a pouze doplňková automatizace domácnosti (jako rozsvícení světel při odjždění

domu) budou systémy od firmy Jablotron ideální.

Systém HomeKit je zajímavý v tom, že zde není potřeba žádný „speciální“ centrální prvek – pokud již uživatel vlastní například iPhone (či jiný produkt, který zastoupí funkci centrálního prvku). Nevýhodou je to, že aby byl systém ovladatelný globálně, tak je potřeba přeci jen mít v domácnosti nějaký prvek, co bude domácnost řídit. A pokud uživatel již nějaký nevlastní, tak se stává další investicí. Co je na systému HomeKit pozitivní je jeho nízká cena – ve srovnání se systémy od společnosti Loxone či Jablotron. Systém HomeKit se stále rozrůstá a má velkou podporu v rozmanitosti produktů. A na rozdíl od předchozích zmíněných systémů je více orientovaný na běžné uživatele. Jednou z nevýhod je zde to, že je systém orientovaný zejména na bezdrátovou komunikaci, která samozřejmě někdy může být méně spolehlivá. Tím spíše že mnoho produktů komunikuje pouze pomocí Bluetooth, takže si uživatelé musejí hlídat dosah zařízení.

Systém HomeConnect vnáší do automatizace domácnosti zajímavý koncept. Zatímco některé systémy umožňují automatizovat domácnosti například chytrými zásuvkami či spínači, HomeConnect ve spolupráci s jinými společnostmi vyvíjí přímo spotřebiče s prvky chytré domácnosti, čímž tyto spotřebiče obsahují mnohem více „inteligence“, na rozdíl od pouhého „zapínání/vypínání“. Nicméně nevýhodou je zde příliš malý sortiment produktů, a tudíž jednoúčelová aplikace navíc, kterou stejně musejí uživatelé doplnit o další aplikace, chtějí-li například rovněž ovládat zásuvky, světla či rolety. Kromě toho je rovněž cena produktů dost vysoká.

...

5.2 Návrh technického řešení

Na základě výzkumu dostupných řešení a jejich zhodnocení jsem se rozhodl vyvinout systém, který bude mít některé spíše základní funkce automatizace. Především zde bude řešeno dálkové ovládání jednoho zařízení druhým, jelikož je to zadání mé bakalářské práce. V systému tedy bude figurovat nějaký ovladač a dále ovládané prvky.

V práci tudíž bude nutné zvolit vhodné vestavné zařízení, které bude sloužit jako ovládací část systému a také zařízení, která budou přijímat povely. Rozhodl jsem se, že ovládaný prvek zde nebude žádné konkrétní zařízení (jako zásuvka, spínač, či lampička) ale spíše obecný modul se vstupně výstupními porty, přes které bude možné ovládat jiná, už konkrétnější zařízení. Pro plnohodnotnou funkci systému bude tento modul nutné opatřit dalšími přídatnými součástkami. Půjde zejména o relé pro možnost ovládání zapnutí a vypnutí zařízení připojeného k tomuto modulu (tímto způsobem bude možné například ovládat LED pásek, či vytvořit bezdrátovou zásuvku). K modulu budou připojeny rovněž tranzistory pro možnost použití pulzně šířkové modulace na výstupu - takový výstup pak bude sloužit pro stmívání světla, zejména LED pásku či bodových LED světla na 12 V. Ovládat tak bude možné rovněž servo motory, které se standardně řídí signálem pulzně šířkové modulace.

Aby bylo ovládání systému co nejjednodušší, bude zde možnost ovládání na dotykovém displeji. Ovládání systému jen z jednoho místa však není v oblasti automatizace úplně nejlepším a efektivním řešením (například uživatel sice nemusí dojít k vypínači světla, aby zhasl, ale stejně musí dojít někde k ovládacímu zařízení systému). Rozhodl jsem se tedy, že bude možnost ovládání i z telefonu, počítače a dalších zařízení (požadavky na zařízení budou definovány později v části 6.1 Celkový návrh systému) **[[Zmínit v implementaci**

požadavky]]. Aby byl plněji využit potenciál displejů (ať už připojeného k ovládacímu zařízení, či ostatním zařízením, ze kterých bude možné moduly ovládat), rozhodl jsem se, že do systému bude možné zapojit i některé snímače.

Půjde o tyto snímače:

- teploty
- vlhkosti vzduchu
- tlaku vzduchu
- intenzity světla
- pohybu
- sepnutí (kontaktu/tlačítka)

Jde o typické představitele analogových i digitálních senzorů. Pro účely práce budou použity jen tyto, nicméně řešení chci pojmut jako open source (a s ohledem na tento fakt se budu snažit o co největší rozšiřitelnost systému), a další senzory bude možné přidat v budoucnu. Všechny senzory budou mít v systému pouze informativní charakter pro uživatele, tzn. bude si moci například zobrazit teplotu v konkrétní místnosti, zda se v ní někdo nachází, či zda byl sepnut nějaký kontakt (např. k detekci otevření okna). V základu (tedy v rámci této práce) nebude možné přímo systém pomocí senzorů řídit, ale opět nic nebrání tomu, aby byla tato funkce implementována v budoucnu. Nebude zde tedy nějaká automatizace například rozsvícení světel na základě detekce pohybu. U modulů s výhodou využiji, že budou mít více vstupů/výstupů a jeden modul tak může sdružovat více různých funkcí (mít připojeno několik snímačů a výstupních zařízení, místo jednoho).

Mým řešením bych chtěl doplnit existující open source řešení o takové, které bude svými vlastnostmi určeno spíše pro nadšence v oblasti elektroniky a bude obsahovat česky lokalizované texty. Bude poskytovat jednoduchý systém pro ovládání mnoha zařízení pomocí jednoho vestavného zařízení s připojeným displejem. Svou prací bych si chtěl rovněž vyzkoušet celý návrh a realizaci systému automatizace domácnosti, který bych následně mohl sám využívat. Vylepšením oproti některým již existujícím systémům bude možnost sdružení několika přijímajících zařízení do jednoho (jak jsem zmínil v předchozím odstavci). Také bych chtěl klást důraz zejména na nízkou cenu, která by u pár ovládaných zařízení (bez ceny těchto koncových zařízení) neměla přesáhnout 2000 Kč.

Při práci budu využívat již existující moduly a mikropočítač (který bude sloužit jako „mozek“ systému), které však naprogramuji a vhodným způsobem doplním o některé elektronické součástky (jako zmíněné relé, či tranzistor). Práce se však nebude zabývat konstrukčním návrhem prvků systému, ani návrhem DPS pro hotový systém. Pro případné propojení součástek a zařízení systému budou použita nepájivá pole.

5.3 Požadované vlastnosti navrhovaného systému

Na základě předchozích úvah jsem se rozhodl, že vytvořím systém, který bude splňovat následující vlastnosti:

- Bude zvolen vhodný mikropočítač, který bude sloužit jako ovládací část systému

- Pro ovládací část bude zvolen dotykový displej patřičných rozměrů, aby byl systém přehledný a mohl sloužit pro ovládání zařízení
- Displej by měl být rovněž vybrán s ohledem na možnost zobrazení přehledu o stavu zařízení v systému (například. zobrazení, které zařízení jsou zapnutá, či jaké hodnoty se nacházejí na PWM výstupu)
- K implementaci bude zvolen vhodný programovací jazyk
- Bude podporována funkce přímého ovládání výstupů ovládaných modulů
- Také zde bude funkce zobrazení dat ze senzorů
- Systém bude obsahovat českou jazykovou lokalizaci
- V systému budou fungovat uživatelské účty, přičemž z jednoho účtu bude možné ovládat jen jednu domácnost
- Bude zvolena vhodná technologie bezdrátového přenosu tak, aby byl systém co nej-jednodušší na implementaci a případné rozšiřování
- Projekt bude uvolněn jako open source, čímž bude cena systému jako takového pro potenciální uživatele minimální (daná pouze cenou použitých součástek a zařízení)
- Součástky a zařízení budou v systému voleny s ohledem na co nejnižší cenu.

Kapitola 6

Realizace a testování

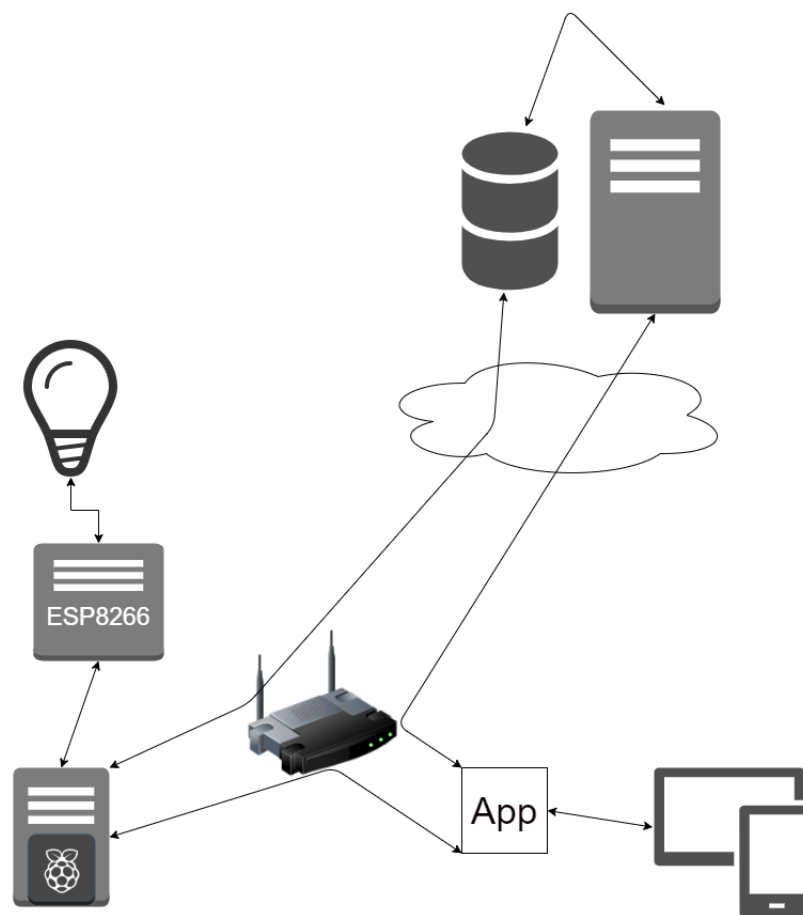
V této kapitole se věnuji vlastní realizaci řešení a následnému testování a vyhodnocení. V první podkapitole uvádím celkový návrh systému - tedy z jakých aplikací a zařízení se bude skládat, jak mezi sebou budou komunikovat apod. Následně se věnuji návrhu grafického uživatelského rozhraní pro aplikaci, ke kterým uživatel bude přistupovat. Dále vysvětluji jakým způsobem se v systému ukládají data a jak jsem vyřešil autentizaci uživatelů v systému. V dalších kapitolách již rozebírám implementaci konkrétních aplikací. Nejprve klientské aplikace, která přímo komunikuje jen s databází a potom v samostatné podkapitole popisují spolu aplikaci pro Raspberry Pi a moduly ESP8266 (jelikož tyto dvě aplikace spolu přímo komunikují a jsou na sobě závislé). Nakonec uvádím jak jsem systém testoval a k jakým výsledkům jsem došel.

[[Někde vymezit pojmy...klient, klientská aplikace, (koncový) modul...]]

6.1 Celkový návrh systému

Systém bude tvořen třemi aplikacemi a několika zařízeními (každé zde bude mít svou roli). Na obrázku 6.1 je možné vidět celkovou koncepci systému.

Pro uživatele zde budou klíčovými různá zařízení s displejem (notebook, mobilní telefon, Raspberry Pi s displejem...), přes který bude systém konfigurovat, ovládat a přes který bude informován o naměřených hodnotách na snímačích. Prostřednictvím grafické aplikace na těchto zařízeních bude vlastně upravovat databázi. Bude se jednat o webovou aplikaci, která bude generována webovým serverem (ať již přístupným na internetu přes webhosting, tak i lokálně z Raspberry Pi). Jak je naznačeno na obrázku, databáze bude přístupná v síti internetu, ovšem s nastavenými přístupovými pravidly, aby bylo minimalizováno riziko napadení systému. Mikropočítač Raspberry Pi bude v systému mít roli nejen lokálního webového serveru, ale také řídicí jednotky. Ta bude naslouchat změnám v databázi a pokud k nějakým dojde, tak patřičným způsobem zareaguje (např. odešle pokyn k nastavení výstupu na modul ESP8266). Mezi webovou aplikací (spuštěnou např. v prohlížeči mobilního telefonu) a aplikací na Raspberry Pi nebude probíhat žádná přímá komunikace (pouze skrze změny v databázi). Dalším zařízením v systému budou moduly ESP8266 (resp. některé vývojové desky, obsahující tento modul). Ty mohou mít ke svým vstupům připojeny snímače a k výstupům pomocné součástky (jako relé či tranzistory) a k nim dále již výstupní ovládaná zařízení, např. světla.



Obrázek 6.1: Architektura systému pro automatizaci domácnosti

Zařízení

Jak tedy vyplývá z předchozího popisu koncepce systému, budou v systému tato zařízení:

- Centrální (řídící) prvek (s připojeným displejem)
- Koncové moduly
- Snímače
- Ovládaná zařízení
- Ostatní zařízení, přes která bude možné systém ovládat (klienti)

Centrální jednotka bude mít na starosti celý systém řídit. Pokud například uživatel zadá pokyn ke změně hodnoty na některém výstupu modulu, bude to právě Centrální jednotka, kdo tuto žádost zpracuje a pošle ji dále jako příkaz na konkrétní koncový modul. Jelikož k centrální jednotce bude potřeba připojit (dotykový) displej, musí mít dostatečný výkon pro vykreslování na něj a zpracování dat. Bylo tedy potřeba zvolit vhodný mikropočítač (protože mikrokontrolery mívají obecně daleko menší výkon) opatřený portem pro přenos obrazového signálu. Jedním z nejpoužívanějších mikropočítačů je Raspberry Pi. Je relativně levné (ve verzi 3 se dá sehnat do jednoho tisíce Kč, verzi Zero W je dokonce možné na českém

trhu sehnat za přibližně 300 Kč) a tedy i snadno dostupné. Nadto je použití Raspberry Pi doporučeno v zadání mé práce. Pro účely práce budu používat právě zmíněnou verzi 3.

Koncový modul (dále již jen modul) bude zařízení, které bude mít vstupy a výstupy. Modul bude měnit hodnoty na svých výstupech dle instrukcí, přicházejících z centrální jednotky. K těmto výstupům pak budou dále připojeny příslušné součástky (jako relé, či tranzistor) a k nim dále již reálná zařízení, ve kterých budou nějakým způsobem spínat kontakt. Rovněž bude v pravidelných intervalech číst hodnoty na svých vstupech (ke kterým budou připojeny snímače). Jako koncový modul je potřeba vybrat zařízení s nízkou cenou, jelikož těchto zařízení v systému bude obecně více. Kromě toho bude modul přijímat pouze jednoduché příkazy od Raspberry Pi, může se tedy klidně jednat o nějaký mikrokontroler s omezeným výpočetním výkonem. Je však potřeba, aby byl tento mikrokontroler schopný komunikovat po síti (tento požadavek vyplývá ze zadání mé bakalářské práce). Buďto je možné vybrat nějaký mikrokontroler a k němu připojit modul, který jej o funkci komunikace po síti doplní (například vývojovou desku Arduino a nějaký WiFi modul), nebo zvolit takový, který již v sobě obsahuje prostředky nutné k bezdrátové komunikaci. Já jsem se rozhodl pro druhou možnost. Na trhu existuje spousta takových mikrokontrolerů, mezi nejznámější (a zároveň relativně levný) patří ESP8266 a novější ESP32. Jelikož je pak ESP8266 levnější (a výpočetní výkon pro danou aplikaci dostatečný), tak použiji právě tento modul.

Jako snímače jsem pro demonstraci vybral následující:

- BMP280 (snímač tlaku a teploty)
- SHT21 (snímač teploty a vlhkosti)
- BH1750 (snímač intenzity světla)
- HC - SR501 (PIR pohybové čidlo)
- tlačítko

Ovládanými zařízeními zde myslím ta, která budou připojena na některý z výstupů modulu. Půjde tedy například o LED pásek, či zařízení, u kterého se bude spínat nějaký kontakt.

Klientská zařízení budou libovolná zařízení s displejem, přes která bude uživatel moci ovládat systém. Jedinými požadavky na tato zařízení jsou zde vzhledem k implementaci (viz dále) následující:

- Přístup k internetu
- Dostatečně velký displej (ideálně 5" a více)

Aplikace

Dále se systém bude skládat z následujících tří aplikací:

- Aplikace s grafickým rozhraním pro uživatele k ovládání systému
- Server, který bude od od výše zmíněné grafické aplikace získávat instrukce a vykonávat je
- Aplikace pro moduly (v následujícím textu jako klientská aplikace)

Po úvaze jsem dospěl k názoru, že nejvhodnější bude, když bude první zmíněná aplikace implementována jako jednostránková webová aplikace. A to z několika důvodů:

- Webové aplikace jsou obecně vzato multiplatformní (pokud mají alespoň trochu responzivní design), čehož vhodně využiji v rámci možnosti ovládat domácnost z různých zařízení (a navíc na dálku)
- K přístupu k aplikaci tak bude stačit mít zařízení s přístupem k internetu (a obrazovkou pochopitelně)
- Webové technologie patří mezi rychle se rozvíjející, což přispívá ke snadnému budoucímu rozvoji systému
- Jazyk Typescript (který budu používat) je jeden z vůbec nepoužívanějších a neznámějších programovacích jazyků, pokud se tedy najde programátor, který by chtěl v budoucnu systém rozšířit, je velká pravděpodobnost se nebude muset učit nic nového.

Tato aplikace tedy bude běžet v internetovém prohlížeči. Mezi zařízení, která budou k aplikaci přistupovat bude i Raspberry Pi (díky připojenému dotykovému displeji).

Druhá aplikace (server) pak bude běžet na Raspberry Pi jako server zpracující požadavky od klientů (sem patří i případ displeje Raspberry Pi). Na základě těchto požadavků bude server dále komunikovat s jednotlivými koncovými moduly. Nebude-li explicitně řečeno jinak, je V dalším textu pod pojmem server vždy myšlena právě tato aplikace, běžící na Raspberry Pi.

Jako alternativu těchto prvních dvou aplikací bylo možné vytvořit jen jednu desktopovou aplikaci, běžící na Raspberry Pi. Přes tu by se systém ovládal pouze z připojeného displeje, nicméně uživatelé by tak přišli o možnost ovládat systém i vzdáleně, resp. i z dalších zařízení, což by bylo neefektivní a mnohdy nepříjemné. Zvolil jsem tedy řešení dvou aplikací.

Poslední uvedená aplikace poběží na jednotlivých koncových modulech. Jejím úkolem bude přijímat pokyny od serveru a na jejich základě ovládat své výstupy. Rovněž bude aplikace v pravidelných intervalech serveru posílat naměřené hodnoty od připojených senzorů.

Kromě těchto třech aplikací bude součástí systému (i když nepřímou) rovněž vzdálený webový server a databáze. Nebudou sice nějakým způsobem u uživatele (že by si je uživatel musel nějak pořizovat), nicméně jedná se rovněž o nepostradatelnou součást systému. Pro webhosting a databázi jsem se rozhodl použít služby Firebase od společnosti Google. Více se této volbě věnuji v části 6.2.

6.2 Grafický návrh webové aplikace

Aplikaci jsem rozdělil na několik samostatných částí (oken), mezi kterými může uživatel přecházet. Jelikož se jedná o jedno stránkovou aplikaci, nejde v případě jednotlivých částí o klasické stránky (v tom smyslu, že by každá měla vlastní html dokument, který ji generuje). V následujícím textu se však pro jednoduchost budu odkazovat ke každému takovému samostatnému oknu, jako k jedné stránce.

Přihlašovací stránka a registrace

Na úvod při spuštění aplikace se zobrazí stránka, na které se uživatel bude moci přihlásit (pomocí emailu a hesla). Rovněž zde bude možnost přejít k registraci nového účtu.

Menu

V aplikaci se přihlášenému uživateli bude zobrazovat ikona menu. Po kliknutí na tuto ikonu se zobrazí vysouvací menu, kterým se bude moci přepínat mezi jednotlivými stránkami (jako domovská stránka či nastavení). V menu také bude možnost přepnutí na celoobrazovkový režim a odhlášení. Opětovným kliknutím na ikonu menu se menu uživateli skryje.

Domovská stránka

Za domovskou stránku považuji tu, na kterou se uživatel dostane ihned po přihlášení do systému. Tato stránka bude sloužit i jako výchozí pro ovládání jednotlivých zařízení. Bude zde také přehled hodnot na snímačích. Tuto stránku bude mít uživatel zobrazenou většinu času na displeji Raspberry Pi, pokud jej bude chtít používat jako takový rychlý přehled o stavech ovládaných zařízení (a snímačích). Jak vidíme na obrázku XYZ bude tato stránka rozvržena na jednotlivé místnosti. Pro každou zde bude pod sebou vyhrazené místo a v něm pro každou místnost se stejným rozvržením - název místnosti, seznam hodnot na senzorech a ovládaná zařízení (ze kterých bude možné vyčíst jejich stav i ovládat je). Pokud bude chtít uživatel ovládat nějaký výstup (zařízení), klikne na něj. V případě, že je výstup digitálního typu (tedy s hodnotami vypnuto/zapnuto) okamžitě se změní stav zařízení. V případě výstupů, řízených pomocí PWM se zobrazí nad zařízeními posuvník (nastavený na aktuální hodnotu), kterým bude možné okamžitě měnit hodnotu na výstupu modulu (a tedy stav zařízení).

Nastavení

V aplikaci bude také stránka s nastavením. Bude zde možnost přidávat nová zařízení a konfigurovat ta stávající. Stránka je navržena jako hierarchie do sebe vnořených seznamů, ve kterých uživatel může zvolit místnost, koncový modul a snímač/zařízení. Pod seznamy je detail, ve kterém může uživatel u aktuálně zvoleného prvku (místnosti/modulu/snímače/-zařízení) konfigurovat některé vlastnosti, jako je název, typ vstupu/výstupu apod.

6.3 Databáze, autentizace v systému a webhosting

V systému je potřeba nějakým způsobem ukládat data a to ze dvou důvodů. Prvním je, že některé části systému (zejména server) nemusejí být vždy v provozu a po opětovném uvedení do provozu je potřeba systém uvést do stavu před vypnutím. Druhým důvodem je, že se v s **[[Popsat proč jsem zvolil Firebase]]**

6.4 Implementace klientské aplikace

[[Zmínit že spárování uživ. účtu s RPi bude tím, že se přihlásí uživatel na lokálním serveru]] **[[Zmínit alternativy hierarchické vs postupné nastavení a proč jsem zvolil hierarchii]]** **[[Zmínit přidávání modulu (dialogové okno)]]**

Pro implementaci klientské aplikace jsem se rozhodl použít programovací jazyk TypeScript, jak jsem již odůvodnil v podkapitole 6.1. Oproti Javascript (který bude výstupem) má tu výhodu, že umožňuje psát staticky typovaný kód, což napomůže k eliminaci chyb a rovněž některá vývojová prostředí pro Typescript poskytují funkci našeptávání.

Vlastní komponenty

Aplikace využívá možnosti definování vlastních komponent (tzv. custom components). Za tímto účelem jsem vytvořil třídu `AbstractComponent`. Tato třída není určena k inicializaci objektů, ale k vytváření nových tříd zděděním této třídy. Každá třída, která z ní dědí musí přepsat vlastnost `tagName`, která definuje název značky, pod jakým se komponenta bude zobrazovat v html dokumentu. Je zde pravidlem, že se v názvu musí vyskytovat alespoň jedna pomlčka. Kromě této vlastnosti existuje několik metod, které je možné v potomcích implementovat:

- `addListeners()` - slouží pro inicializaci posluchačů událostí
- `connectedCallback()` - je volaná po připojení komponenty do DOM stromu
- `disconnectedCallback()` - je volaná po odstranění komponenty z DOM stromu
- `attributeChangedCallback()` - je volaná při změně některého atributu komponenty
- `disconnectComponent()` - slouží pro odstranění komponenty z DOM stromu

Ve skutečnosti opomenutá implementace některé z těchto metod aktuálně vede k vyvolání výjimky, ale toto chování je možné vypnout v třídě `Config`.

Dále se ve funkci nachází několik funkcí pro přidávání jiných komponent k dané komponentě a metoda `initializeFromProps()`, která se volá z konstruktoru pro nastavení komponenty (a případně je možné ji kdykoliv volat znovu odjinud - v případě že se nějak změní žádané vlastnosti a chceme komponentu reinitializovat). Zmíněná metoda přijímá jako parametr rozhraní `IComponentProperties`, které obsahuje několik vlastností, kterými je možné komponentu nastavit (např. komponenty, které se připojí jako dceřinné prvky v HTML dokumentu) a dále všechny CSS vlastnosti (je tak možné již při vytváření komponenty nadefinovat její výchozí vzhled). Samozřejmě CSS vlastnosti jsou uloženy v patřičném CSS souboru, ale díky rozhraní `IComponentProperties` je možné udržet CSS kód bez zbytečných detailů a může tak obsahovat pouze obecné styly. ??V dalším textu používám frázi komponenta a instance třídy (dědicí z `AbstractComponent`) jako synonymum??.

Životní cyklus aplikace ???

Vstupní třída aplikace je `AutoHomeApp`. Zde se nainicializují Firebase funkce, nadefinují všechny vlastní komponenty a nakonec se vytvoří instance třídy `PageCreator`. Ta je, jak vyplývá z jejího názvu zodpovědná za celkové sestavení HTML dokumentu. Ústřední metoda této funkce se nazývá `renderPage()`. V konstruktoru třídy `PageCreator` se zmíněná metoda zaregistruje v instanci třídy `URLManager`, na kterou se posílá požadavek na změnu stránky (zejm. při kliknutí na položku menu) a ta vždy následně volá zaregistrovanou metodu `renderPage()`.

V `renderPage()` se získá (z instance třídy `AppRouter`) informace, která stránka se má aktuálně vytvořit. Není-li definovaná, tak se pošle požadavek (instanci třídy `URLManager`) na přesměrování na domovskou stránku. V opačném případě se vytvoří a přidá do správce stránek (instance třídy `PageManager`) požadovaná stránka (pokud se tam již nenachází) a následně se aktivuje.

Stránky, které se přidávají do správce stránek jsou vlastní komponenty, které dědí třídu `BasePage`. Jejich vlastností je, že jsou pozicovány absolutně a jejich šířka je rovna šířce okna

prohlížeče. Veškerá logika, která není implementovaná přímo v jednotlivých komponentách je řešena právě v jednotlivých potomcích třídy `BasePage`.

Celkem se v aplikaci vyskytují 3 stránky:

- Přihlášení
- Registrace nového uživatele
- Domovská stránka
- Nastavení

implementace jednotlivých stránek

Na stránkách s přihlášením a registrací není vcelku nic zajímavého, nachází se tam jen formulář pro přihlášení, resp. registraci. Domovská stránka (komponenta `HomePage`) je sestavena z komponent `RoomCard` (karet místností). V konstruktoru domovské stránky se registruje posluchač události změny databáze (konkrétně změny v místnostech). Při vyvolání této události se vyhodnotí, zda se změnilo pořadí místností (v tomto případě se znovu sestaví celé rozložení domovské stránky). Dále se (nezávisle na změně pořadí místností) aktualizují jednotlivé karty místností voláním metody `updateCard()` na těchto komponentách.

Nastavení (komponenta `SettingsPage`) se skládá z několika komponent `ListFrame` (editovatelný seznam), `TabLayout` (rozložení se záložkami) a jednoho `DetailFrame` (editovatelného detailu). Z `TabLayout` komponent je vytvořena hierarchie seznamů pro výběr položky k editaci. V aplikaci je `TabLayout` pro místnosti, moduly a pak společný pro snímače a zařízení. Z praktického hlediska má využití jen pro dvojici snímače a zařízení, ale použil jsem ho i pro místnosti a moduly, protože snadno (jako název záložky) zobrazuje o co v dané úrovni jde. Detail slouží pro editaci posledně zvolené položky v některém ze seznamu.

6.5 Implementace serveru na Raspberry Pi a aplikace pro koncové moduly

Podobně jako webový klient je i server napsaný v programovacím jazyce Typescript. Aby bylo možné na Raspberry Pi spustit Javascript (který je výstupem Typescriptu) mimo internetový prohlížeč, nainstaloval jsem na něj Javascriptové běhové prostředí NodeJS.

Aplikaci pro koncové moduly jsem implementoval v jazyce Wiring a C++. Při spuštění modulu se provede základní vstupní nastavení aplikace. Zejména se nastaví sériová komunikace, modul se připojí k Wi-Fi, spustí se server, naslouchající v CoAP multicastové skupině, CoAP serveru se přidají callback funkce a spustí se CoAP server.

Komunikační protokol

Pro implementaci komunikace mezi zařízeními bylo potřeba vybrat vhodný aplikační protokol. Mezi možnosti, které jsem zvažoval patří HTTP, MQTT a CoAP. Vzhledem k tomu, že bude na jedné straně komunikace figurovat relativně nevýkonný modul (ESP8266) a navíc modul může být vzdálený od přístupového bodu (a připojení může kolísat), rozhodl jsem se nepoužít pro tento účel poněkud těžkopádný protokol HTTP. Zbylé 2 protokoly vyhovují malou strukturou přenášených zpráv, nicméně nakonec jsem zvolil CoAP. Vedli mě k tomu 2 důvody:

- U MQTT protokolu se předpokládá, že centrální prvek bude fungovat jen jako směrovač zpráv a celková zodpovědnost za interpretaci dat bude na jednotlivých koncových zařízeních (což je v mém systému nežádané, jak jsem naznačil dříve)
- Protokol CoAP se v jistém směru chová, jako odlehčená verze protokolu HTTP, který má pro komunikaci některé výhodné mechanismy

Pro server na Raspberry Pi jsem využil dostupnou knihovnu `coap`¹⁰ a pro moduly knihovnu `CoAP simple`¹¹.

Komunikace mezi Raspberry Pi a ESP8266

Před kompilací a uložením kódu pro ESP8266 do paměti je potřeba upravit konfigurační část kódu, ve které je potřeba zapsat správný SSID a heslo k Wi-Fi síti, ve které má systém fungovat. Nastavování údajů o síti se sice může zdát jako zbytečný krok (systém by se o výměnu těchto informací mohl postarat sám), nicméně v rámci bakalářské práce nebylo řešeno šifrování přenášených dat, které by zde bylo nutnou podmínkou a další komunikace (po úspěšném připojení modulu do sítě) bude komunikace probíhat výhradně v rámci lokální sítě. Předpokládám tedy jisté zabezpečení na této úrovni. Navíc zabezpečení (pomocí SSL/TLS) nepodporují i některé komerčně prodávané systémy, např. dálkově ovládané zásuvky od společnosti Kangtai. To jsem si ověřil pomocí aplikace na android `Packet Capture`¹². Při nasazení na produkci by bylo samozřejmě vhodné i tento aspekt bezpečnosti dořešit. V případě potřeby je v konfigurační části možné změnit typ vývojové desky, který je standardně nastaven na NodeMCU.

Na počátku komunikace mezi Raspberry Pi a jednotlivými moduly stojí přidání nového modulu do databáze (typicky v nastavení systému klientské aplikace). Server neustále naslouchá změnám v databázi a pokud zaregistruje přidání modulu, tak se pokusí kontaktovat nový modul CoAP zprávou. V tuto chvíli ještě není známá IP adresa potenciálního modulu, bylo tedy potřeba zvolit vhodný mechanismus, jakým způsobem modul kontaktovat. Server pošle zmíněnou zprávu tzv. "All CoAP Nodes" multicastové skupině (IP adresa 224.0.1.187). Použitá knihovna `CoAP simple` (kterou využívají moduly) nepodporuje přijímání multicastových CoAP zpráv, bylo tedy nutné nastavit UDP server, který bude na dané multicastové skupině naslouchat (spuštěný server se přihlásí do dané multicastové skupiny pomocí IGMP zprávy, díky čemuž následně modul může komunikovat v této skupině jako by se jednalo o jeho vlastní IP adresu). Nové, dosud nepřidané moduly v této multicastové skupině naslouchají a v případě příchozí zprávy odpoví opět CoAP zprávou, v rámci které posílají informaci o tom, o jaký typ vývojové desky se jedná (toho následně využívá webový klient, když uživateli nabízí možné V/V piny v nastavení systému na základě typu vývojové desky, kterou uživatel konfiguruje). Pokud server získá odpověď, aktualizuje záznam o modulu v databázi (vloží IP adresu modulu a typ vývojové desky). Následně pošle modulu jeho ID záznamu z databáze. Klient si až v tuto chvíli uloží IP adresu serveru (zpráva je totiž již posílána jako unicast na konkrétní modul). Ten od tohoto okamžiku přestane naslouchat multicastovým CoAP zprávám.

K již přidáním modulům lze přes klientskou aplikaci přidávat snímače a výstupní zařízení. Do databáze se vloží příslušný záznam a v reakci na tento nový záznam se daný

¹⁰<https://www.npmjs.com/package/coap>

¹¹<https://www.arduino.cc/reference/en/libraries/coap-simple-library/>

¹²<https://play.google.com/store/apps/details?id=app.greyshirts.sslcapture>

snímač/ výstupní zařízení objeví na domovské stránce ve webové aplikaci. Server zaznamená změnu v databázi a jako reakci pošle zprávu příslušnému modulu. V případě nového snímače odešle modulu žádost o sledování hodnoty na daném vstupu. Od této chvíle modul kontroluje v pravidelných intervalech daný vstup a v případě dostatečné změny pošle informaci na server, který změny uloží do databáze. Pro jednotlivé snímače je v kódu definovaný minimální rozdíl hodnot, která se odešle oproti poslední odeslané a to z toho důvodu, aby nebyla síť zbytečně zahlcovaná a pak také aby server nebyl tak zaneprázdněn přijímáním zbytečných zpráv. Některé snímače totiž mění naměřené hodnoty i v řádu několika desetinných míst, což například u teploty v místnosti nemá smysl sledovat. Kromě toho se tyto informace posílají jako nepotvrditelné zprávy, jelikož jich může z modulů přicházet relativně velké množství a ve většině případů není nijak kritické, pokud server nějakou zprávu občas nezachytí. Při návrhu jsem váhal, zda nebude vhodnější mechanismus takový, že se v pravidelných intervalech bude server sám dotazovat modulů na hodnoty na jejich vstupech. Tím by se šetřili zdroje modulů. Nicméně za cenu velkého zahlcování sítě a zbytečného zaneprázdnění serveru. Také naměřené hodnoty by nebyli k dispozici "okamžitě", protože by bylo nevhodné posílat dotaz například každých 500 ms. Navíc použité moduly nemají zas tak žalostný výkon, rozhodl jsem se tedy pro popsání řešení. Dále pokud se přidává nové výstupní zařízení, odešle se na modul zpráva o nastavení daného pinu jako výstup a hodnotu, kterou má modul na výstup nastavit. Při přidání nového výstupního zařízení je tato hodnota 0 (tedy vypnuto). Pokud následně uživatel ovládá hodnotu na daném výstupu (z domovské stránky klientské aplikace), odešle se stejná zpráva, jako při přidávání zařízení s tím rozdílem, že tentokrát se přenáší jiná hodnota. Ovládání výstupů je v systému poněkud citlivější záležitost (uživatel chce mít jistotu, že například světla, která vypnul jsou skutečně vypnutá) a proto se tyto zprávy posílají jako potvrditelná CoAP zpráva. Server tedy po odeslání zprávy čeká na potvrzení a pokud jej v daném časovém limitu neobdrží (pokud nepřekročí maximální počet pokusů), tak se pokusí požadavek poslat znovu. TODO-po inicializaci se nastavují všechny vstupy a výstupy - popsat

IP adresy jednotlivých modulů jsou uchovávány v databázi. Je však potřeba řešit případ, kdy se změní IP adresa modulu. Např. pokud se ...

Formát přenášených zpráv mezi Raspberry Pi a ESP8266

Jak již bylo zmíněno dříve, ke komunikaci mezi Raspberry Pi a ESP8266 jsem se rozhodl použít CoAP protokol. CoAP zpráva /*OLD: V systému bude několik typů zpráv. Zprávy přenášené po síti budou mít V/V pin, či sběrnici specifikovanou jako cestu ke zdroji a v samotném datovém obsahu (payload) přenášenou hodnotu. Příklad získání hodnoty senzoru na pinu A0 může Raspberry Pi získat následujícím GET požadavkem:

```
coap://192.168.5.1/A0
```

V případě posílání se pak pošle POST žádost s příslušným datovým obsahem, což bude číslo od 0 do 1024. To jsou totiž hodnoty, ve kterých je ESP8266 modul schopný převádět digitální hodnoty na analogové (jak bylo zmíněno již dříve).

Přidání nového modulu do systému Jelikož systém není uzavřený, ale je možné jej rozšiřovat o nové moduly (nová ovládaná zařízení), musel jsem vytvořit mechanismus, kterým se do systému budou přidávat nové moduly..... Postup při přidávání modulu do systému je následující:

- Uživatel upraví v kódu pro ESP8266 SSID a heslo k WiFi síti, ve které bude modul operovat.
- Program zkompiluje a načte do ESP8266.
- Spustí modul. Ten se přihlásí k multicastové skupině 224.0.1.187 (All CoAP Nodes skupina)
- V konfiguraci systému (v

klientské aplikaci) klikne na tlačítko pro přidání modulu (pro požadovanou místnost) - Tím se ve všech aktuálně spuštěných klientských aplikacích, které jsou připojené k internetu zobrazí dialogové okno s informací o čekání na přidání nového modulu do systému. Tímto oknem se zablokuje jakékoliv akce v klientské aplikaci dokud se nepřidá nový modul. Dialogové okno bude obsahovat tlačítko na zrušení akce. - Raspberry Pi v reakci na žádost o přidání modulu vyšle dotaz na multicastovou skupinu, přes který nově přidávanému modulu sdělí svou IP*/

/*

Cesta v CoAP dotazu by měla obsahovat cestu ke zdroji, ke kterému chceme přistupovat. Jelikož však obecně není nutné zdroje (myšleno GPIO) ESP8266 nějak dále dělit, rozhodl jsem se, že tuto cestu využiji spíše na rozpoznání typu požadavku na CoAP server. K tomu mě vedl i způsob implementace CoAP protokolu v použité knihovně Simple Coap, ve které je před spuštěním serveru nutné nadefinovat cesty, na kterých server naslouchá. Pokud bych chtěl cestu používat k identifikaci zdroje, pak by bylo nutné takto zaregistrovat funkci pro každou

...a tak místo abych přenášel požadovanou akci (na daném vstupu/výstupu), tak tuto akci specifikuji přímo cestou.

*/

6.6 Testování a vyhodnocení systému

Provedené testy

*/Proč jsem testoval... V rámci testování jsem provedl tyto testy:

- Test "multiplatformnosti" klientské aplikace
- Test multiplatformnosti serveru
- Test responzivity
- Test podporovaných prohlížečů
- Test reaktivity
- Test intuitivnosti uživatelského prostředí
- Test stability při potížích s připojením k internetu
- Test autonomie?

Test Test "multiplatformnosti" klientské aplikace probíhal takto...

Systém jako takový byl vytvořen s ohledem na běh (serveru) na Raspberry Pi. Implementoval jsem jej však tak aby běželo v prostředí Node.js, které je možné spustit na mnoha různých zařízeních a bylo vhodné otestovat funkčnost systému. Jelikož jsem funkce serveru navrhoval s ohledem na monost běhu i na méně výkonných zařízeních (které však umožňují běh Node.js), rozhodl jsem se otestovat běh systému na dalších zařízeních. Konkrétně šlo o notebook HP (s technickými parametry 4GB RAM, Intel Core i5-7200U) a chytrý telefon (s OS Android verze 8). Test multiplatformnosti serveru pak probíhal takto..... Chytré telefony nemají nativní podporu pro Node.js, musel jsem to tedy obejít. Na telefon jsem nainstaloval aplikaci Termux (která funguje jako emulátor terminálu pro Android a Linuxové prostředí)

Test responzitivity...
Test podporovaných prohlížečů....
Test reaktivity (neboli odezvy systému na podněty v reálném čase) probíhal takto...
Test intuitivnosti uživatelského prostředí probíhal takto...
Test stability při potížích s připojením k internetu probíhal takto...
Test autonomie (funkce systému bez lidského zásahu) probíhal takto...

Testy které nebyli provedeny

Kromě testů, které jsem už provedl by bylo vhodné provést ještě dále zmíněné. V případě absence prvních dvou nedojde k žádným potížím (jen systém možná nebude fungovat na jiných zařízeních). Problémy, které by se otestovali zbylými dvěma testy by však mohli být kritické. Alespoň u čtvrtého testu ale věřím, že nejsou obavy na místě, přeci jen je konkurence velká a systém by neměl být zahlcený. V rámci mé práce tyto testy vykonány nebyli, zejména z finančních a časových důvodů. Testy které by bylo vhodné provést:

- Funkčnost systému na alternativních mikropočítačích
- Funkčnost systému na jiných verzích Raspberry Pi
- Stabilita systému v průběhu několika let chodu
- Test spolehlivosti systému při velkém množství uživatelů
- Test limitů Firebase služby free plan

Raspberry Pi je jedním z nejpoužívanějších a nejznámějších mikroprocesorů, nicméně na trhu se nacházejí další, u kterých by bylo vhodné otestovat, zda bude server i webový klient (na připojeném displeji) plně funkční. Zejména mikropočítače uvedené v kapitole 4 (tam jsem totiž popisoval rovněž nejpopulárnější mikropočítače)

Funkčnost systému na jiných verzích Raspberry Pi je potřeba otestovat zejména z toho důvodu, že se jedná o poměrně rychle se rozvíjející platformu a někteří potenciální uživatelé tak mohou mít starší model s nižším výkonem. Zvláště zajímavý by byl test na verzi Zero W, jelikož tuto verzi Raspberry Pi je možné pořídit za přibližně 300 Kč. V případě bezproblémového chodu by byl projekt velmi zajímavou alternativou k některým velmi drahým systémům jako je **[XYZ]**. Je možné, že by například nebyla tato verze Raspberry Pi dostatečně výkonná na ovládání systému z připojeného displeje, ale zvládala by na běh aplikace serveru, pak by bylo možné systém prostě ovládat z klientů (například chytrých telefonů). Případně kdyby tato verze nezvládala bezproblémový běh serveru s poskytováním statické stránky, tak by bylo možné tuto část serveru upravit (odstranit) a pak by se klienti připojovali pouze na vzdálený server a úloha Raspberry Pi by se zredukovala pouze na kontrolu (resp. naslouchání) změn v databázi a komunikaci s koncovými moduly, což by snad mělo opravdu bez jakýchkoli potíží fungovat. Pak by se samozřejmě muselo vyřešit spárování Raspberry Pi s uživatelským účtem, protože aktuálně to funguje právě na principu, který vyžaduje, aby se uživatel (alespoň poprvé) přihlásil na poskytované statické stránce z Raspberry Pi.

Test stability systému by pak spočíval v tom, že by systém pravidelně po dobu několika let využívalo jisté množství lidí (například 100 domácností) a pozorovalo, zda se systém v průběhu času chová stále stejně, nezpomaluje se a podobně.

Test spolehlivosti systému při velkém množství uživatelů by bylo provést hlavně z toho důvodu, že v systému funguje jedna veřejná databáze. Vyro by vhodné otestovat, jak se systém bude chovat při velké zátěži databáze (ve chvíli, kdy bude k databázi aktivně současně přistupovat mnoho uživatelů).

Test limitů Firebase služby free plan.....

Kapitola 7

Závěr

Nápady na pokračování práce:

- Úspornější řešení - jestli má arduino nižší spotřebu, mohlo by číst senzory a vzbouzet ESP pouze kvůli komunikaci po síti...zjistit jak to mají tyto dva čipy se spotřebou...!
- ovládání hlasem
- přidání podmínek
- přidání módů (scén) - v podstatě by se jednalo o sdružení více akcí ovládání do jedné
- vývoj by mohl směřovat i na odlehčení serveru tak, že bude pouze kontrolovat databázi a komunikovat s moduly (jak bylo polemizováno v testování) => a mohla by se vytvořit verze pro ESP8266 v režimu Centrální jednotky => velice levný systém...Bylo by však pravděpodobně potřeba přepsat server...do C? Pythonu...?

Moje odkazy [58] [80] [78] [52] [67] [50] [44] [18] [19] [20] [57] [77] [54] [22] [75] [47] [24] [40] [70] [55] [74] [53] [32] [41] [65] [35] [36] [31] [43] [42] [48] [63] [45] [33] [30] [46] [34] [16] [9] [15] [68] [14] [56] [11] [12] [61] [6] [79] [60] [7] [8] [2] [3] [4] [5] [13] [28]

Literatura

- [1] Welcome to IFTTT guide. *IFTTT* [online]. Dostupné z: https://ifttt.com/explore/welcome_to_ifttt.
- [2] Naše chytrá řešení. *Jablotron* [online]. Dostupné z: <https://www.jablotron.com/cz/produkty/>.
- [3] O Jablotronu. *Jablotron* [online]. Dostupné z: <https://www.jablotron.com/cz/o-jablotronu/>.
- [4] Katalog produktů. *Jablotron* [online]. Dostupné z: <https://www.jablotron.com/cz/katalog-produktu/>.
- [5] Váš dům vás bude poslouchat. *Jablotron* [online]. Dostupné z: <https://www.jablotron.com/cz/produkty/chytre-ovladani/ja-100-pg/>.
- [6] Funkce chytrého domu pro Vaši domácnost. *Loxone.com* [online]. Dostupné z: <https://www.loxone.com/cscz/wp-content/uploads/sites/7/2017/10/kontrolni-seznam-pro-stavitele-domu.pdf>.
- [7] Miniserver Go. *Loxone.com* [online]. Dostupné z: <https://www.loxone.com/cscz/kb/miniserver-go/>.
- [8] Inteligentní řízení vnitřního klima. *Loxone.com* [online]. Dostupné z: <https://www.loxone.com/cscz/produkty/topeni-klimatizace/>.
- [9] What is the Internet of Things (IoT)? *Redhat.com* [online]. [cit. 2021-01-12]. Dostupné z: <https://www.redhat.com/en/topics/internet-of-things/what-is-iot>.
- [10] Jak začít s chytrou domácností. *VOCOLinc* [online]. Dostupné z: <https://www.vocolinc.cz/blog/jak-zacit-s-chytrou-domacnosti>.
- [11] Naše vize, mise a příběh. *Loxone.com* [online]. Září 2016. Dostupné z: <https://www.loxone.com/cscz/o-nas/mise/>.
- [12] Chytrá regulace teploty: chytré topení a klimatizace. *Loxone.com* [online]. Září 2016. Dostupné z: <https://www.loxone.com/cscz/produkty/topeni-klimatizace/>.
- [13] Nejen chytrou domácnost. Můžete mít i chytrou firmu! *Jablotron* [online]. Duben 2019. Dostupné z: <https://www.jablotron.com/cz/o-jablotronu/blog/zabezpecena-chytra-firma/nejen-chytrou-domacnost-muzete-mit-i-chytrou-firmu/>.
- [14] Použití zkratk pomocí Siri, aplikace Zkratky nebo Návrhů Siri. *Apple* [online]. Prosinec 2020. Dostupné z: <https://support.apple.com/cs-cz/HT209055>.

- [15] *Hello Emma* [online]. 2020. Dostupné z: <https://helloemma.cz/>.
- [16] ADMOSPHERE, N. S funkcí hlasových asistentů se už setkala polovina internetové populace, jen 13 % je aktivně využívá. *Technologie pro měření médií SimMetry - Nielsen Admosphere* [online]. Březen 2020. Dostupné z: <https://www.nielsen-admosphere.cz/press/s-funkci-hlasovych-asistentu-se-uz-setkala-polovina-internetove-populace-jen-13-je-aktivne-vyuziva/>.
- [17] AL SARAWI, S., ANBAR, M., ALIEYAN, K. a ALZUBAIDI, M. Internet of Things (IoT) communication protocols: Review. In: *2017 8th International Conference on Information Technology (ICIT)*. 2017, s. 685–690. DOI: 10.1109/ICITECH.2017.8079928.
- [18] ALBANO, S. WiFi frequency bands: 2.4 GHz and 5 GHz. *Minim: the residential managed WiFi and IoT security platform* [online]. Duben 2019. 2019-04-01. Dostupné z: <https://www.minim.co/blog/wifi-frequency-bands-2.4-ghz-and-5-ghz>.
- [19] APOORVE. What is the difference between microprocessor and microcontroller? *Circuit Digest* [online]. Červen 2015. 2015-06-02. Dostupné z: <https://circuitdigest.com/article/what-is-the-difference-between-microprocessor-and-microcontroller>.
- [20] ARDUINO. What is Arduino? *Arduino* [online]. 2007. 2018-02-05 [cit. 2020-09-25]. Dostupné z: <https://www.arduino.cc/en/Guide/Introduction>. Path: About.
- [21] ATMOKO, R., RIANINI, R. a HASIN, M. IoT real time data acquisition using MQTT protocol. *Journal of Physics: Conference Series*. Květen 2017, sv. 853, s. 012003. DOI: 10.1088/1742-6596/853/1/012003.
- [22] AVISON, J. *The World of Physics*. 2. vyd. Cheltenham, Spojené království: Nelson Thornes Ltd, červen 1989. ISBN 9780174382454.
- [23] AZZOLA, F. CoAP Protocol: Step-by-Step Guide. *DZone* [online]. Listopad 2018 [cit. 2020-12-12]. Dostupné z: <https://dzone.com/articles/coap-protocol-step-by-step-guide>.
- [24] BAGAD, V. S. *Microwave Engineering*. 1. vyd. Pune, Indie: Technical Publications, 2009. ISBN 9788184313604.
- [25] BECKHOFF. TwinCAT podporuje protokoly pro výměnu dat v IoT. *Automa – časopis pro automatizační techniku*. 1. vyd. 2018, sv. 10, s. 20–21. ISSN 1210-9592.
- [26] CHAN, J. BOSCH HOME CONNECT TAKES SMART APPLIANCES MAINSTREAM. *Reviewed* [online]. Červenec 2020. 2020-07-30 [cit. 2020-11-10]. Dostupné z: <https://www.reviewed.com/refrigerators/features/bosch-home-connect-smart-appliance-review>.
- [27] CHARLTON, A. Yes, you can build and run a smart home without a voice assistant. *GearBrain* [online]. Prosinec 2019. Dostupné z: <https://www.gearbrain.com/smart-home-without-voice-control-2641525934.html>.
- [28] CLOVER, J. HomeKit: Everything You Need to Know. *MacRumors* [online]. Červenec 2020. Dostupné z: <https://www.macrumors.com/guide/homekit/>.

- [29] DASHKOVA, E. a GURTOV, A. Survey on Congestion Control Mechanisms for Wireless Sensor Networks. In: ANDREEV, S., BALANDIN, S. a KOUCHERYAVY, Y., ed. *Internet of Things, Smart Spaces, and Next Generation Networking*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, s. 75–85. ISBN 978-3-642-32686-8.
- [30] DOLEJŠ, J. Google Assistant v praxi: Jak funguje mozek budoucích telefonů s Androidem? (video). *Svět Androida* [online]. Říjen 2016. Dostupné z: <https://www.svetandroida.cz/google-assistant-v-praxi/>.
- [31] EADY, F. *Hands-On ZigBee: Implementing 802.15.4 with Microcontrollers*. 1. vyd. UK: Elsevier, Newnes, březen 2007. Embedded technology. ISBN 9780123708878.
- [32] EL BENDARY, M. A. M. *Wireless Personal Communications: Simulation and Complexity*. 1. vyd. Singapur: Springer, duben 2018. Signals and Communication Technology. ISBN 9789811071300.
- [33] ELEMENT14, F. Does the IoT really need the Internet? *Farnell* [online]. Leden 2018. Dostupné z: <https://uk.farnell.com/does-the-iot-really-need-the-internet#>.
- [34] ELKO EP, s. Topte s rozumem aneb jak ještě více ušetřit za topení: Vedlejší název článku. *A-Z ELEKTRO* [online]. Praha: GIVERSDON s.r.o. 09/10 2015, září/říjen 2015, s. 46–47. Chytré elektroinstalace. ISSN 1805-1073. Dostupné z: <http://www.azcasopis.cz/casopis/2015/9-10/>.
- [35] ESPRESSIF. ESP8266. *ESPRESSIF SYSTEMS: Wi-Fi & Bluetooth MCUs and AIoT Solutions* [online]. Dostupné z: <https://www.espressif.com/en/products/socs/esp8266>. Path: Products; SoCs; ESP8266.
- [36] ESPRESSIF. ESP32. *ESPRESSIF SYSTEMS: Wi-Fi & Bluetooth MCUs and AIoT Solutions* [online]. Dostupné z: <https://www.espressif.com/en/products/socs/esp32>. Path: Products; SoCs; ESP32.
- [37] FILIP, J. Začínáme stavět chytrou domácnost na HomeKit – část 2. *Letem světem Applem* [online]. Leden 2020. Dostupné z: <https://www.letemsvetemapplem.eu/2020/01/16/zaciname-stavet-chytrou-domacnost-na-homekit-cast-2/>.
- [38] FILIP, J. Začínáme stavět chytrou domácnost na HomeKit – část 3. *Letem světem Applem* [online]. Únor 2020. Dostupné z: <https://www.letemsvetemapplem.eu/2020/02/03/zaciname-stavet-chytrou-domacnost-na-homekit-cast-3/>.
- [39] FILIP, J. Začínáme stavět chytrou domácnost na HomeKit – část 1. *Letem světem Applem* [online]. Leden 2020. Dostupné z: <https://www.letemsvetemapplem.eu/2020/01/09/zaciname-stavet-chytrou-domacnost-na-homekit-cast-1/>.
- [40] FULAY, P. a LEE, J.-K. *Electronic, magnetic, and optical materials*. 2. vyd. Boca Raton, Florida, USA: CRC Press, Taylor & Francis Group, 2017. Advanced materials and technological series. ISBN 9781498701693.
- [41] GANGULI, M. *Getting started with Bluetooth*. 1. vyd. Cincinnati, Ohio, USA: Premier Press, září 2002. Networking. ISBN 9781931841832.

- [42] GERHART, J. *Home automation and wiring*. London: McGraw-Hill Publishing, 1999. ISBN 9780070246744.
- [43] GISLASON, D. *Zigbee Wireless Networking*. 1. vyd. UK: Elsevier, Newnes, říjen 2008. ISBN 9780750685979.
- [44] GODSE, A. P. a GODSE, D. A. *Microprocessor and Interfaces*. 1. vyd. Pune, Indie: Technical Publications, leden 2008. ISBN 9788184311259.
- [45] HAMERNIK, P., MUDRONCIK, D. a PAVOL, T. Classification of Functions in Smart Home. *International Journal of Information and Education Technology* [online]. Duben 2012, sv. 2, s. 149–155. DOI: 10.7763/ijiet.2012.v2.98. Dostupné z: https://www.researchgate.net/publication/275156028_Classification_of_Functions_in_Smart_Home.
- [46] HAMERNIK, P. a MUDRONCIK, D. Smart Home Support of Projecting. In: *International Conference on Advanced Computer Theory and Engineering, 4th (ICACTE 2011)*. ASME Press, Leden 2011. DOI: 10.1115/1.859933.paper42. ISBN 9780791859933. Dostupné z: <https://doi.org/10.1115/1.859933.paper42>.
- [47] HERMAN, J. Why Everything Wireless Is 2.4 GHz. *Wired* [online]. Červenec 2010. 2010-07-09. Dostupné z: <https://www.wired.com/2010/09/wireless-explainer/>.
- [48] HOY, M. B. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*. 2018, sv. 37, s. 81 – 88.
- [49] HRMA, J. Google Asistent a čeština: přípravy jsou zřejmě v plném proudu (oživeno). *SMARTmania* [online]. Prosinec 2020 [cit. 2021-02-15]. Dostupné z: <https://smartmania.cz/google-asistent-a-cestina-pripravy-jsou-zrejme-v-plnem-proudu/>.
- [50] ITU. Terms and definitions. In: *Radio Regulations* [online]. ITU, 2020, sv. 1, s. 8,63 [cit. 2020-11-10]. ISBN 9789261302917. Dostupné z: <https://www.itu.int/pub/R-REG-RR-2020>.
- [51] JOSE, J. *Internet of Things*. 1. vyd. Indie: Khanna Book Publishing Co. (P) Ltd., leden 2010. ISBN 9789386173591.
- [52] KAMAL, R. *Embedded Systems: Architecture, Programming and Design*. 2. vyd. New Delhi, Indie: Tata McGraw-Hill Publishing Company Limited, 2008. ISBN 9780073404561.
- [53] KAMMER, D., MCNUTT, G. a SENESE, B. *Bluetooth Application Developer's Guide: The Short Range Interconnect Solution*. Rockland, Massachusetts, USA: Syngress Publishing, Inc, prosinec 2001. ISBN 9781928994428.
- [54] KEIRL, A. a CHRISTIE, C. *Clinical Optics and Refraction: A Guide for Optometrists, Contact Lens Opticians and Dispensing Opticians*. Velká Británie: Elsevier Limited, září 2007. ISBN 9780750688895.
- [55] KIZZA, J. M. Security in Wireless Networks and Devices. In: KIZZA, J. M., ed. *Guide to Computer Network Security*. Cham, Švýcarsko: Springer International Publishing, 2017, s. 397–427. DOI: 10.1007/978-3-319-55606-2_18. ISBN 9783319556062. Dostupné z: https://doi.org/10.1007/978-3-319-55606-2_18.

- [56] KOS, A. Hlasová asistentka Alexa má nový skill. Naučila se přehrávat české audioknihy. *Mobilmania.cz* [online]. Březen 2019. Dostupné z: <https://www.mobilmania.cz/clanky/hlasova-asistentka-alex-ma-novy-skill-naucila-se-prehravat-ceske-audioknihy/sc-3-a-1344493/default.aspx>.
- [57] KURNIAWAN, A. *Arduino Programming with .NET and Sketch*. 1. vyd. Berkeley, CA, USA: Apress, 2017. ISBN 9781484226599.
- [58] LABIOD, H., AFIFI, H. a DESANTIS, C. *Wi-Fitm, Bluetoothm, Zigbeetm and WiMaxtm*. Springer-Verlag New York Inc., říjen 2006. ISBN 1402053967.
- [59] LEE, J. J., KIM, K. T. a YOUN, H. Y. Enhancement of congestion control of Constrained Application Protocol/Congestion Control/Advanced for Internet of Things environment. *International Journal of Distributed Sensor Networks*. Listopad 2016, sv. 12. DOI: 10.1177/1550147716676274.
- [60] LOXONE. Komplexní řešení inteligentního ovládání. *Create Automation* [online]. Loxone. 2020, č. 2, s. 30–31. Dostupné z: https://www.loxone.com/cscz/wp-content/uploads/sites/7/2020/05/Loxone_Magazine_2020-CZ-web.pdf.
- [61] LOXONE. *Loxone.com* [online]. Květen 2020. Dostupné z: https://www.loxone.com/wp-content/uploads/datasheets/Datasheet_MiniserverGo.pdf.
- [62] MALÝ, M. Protokol MQTT: komunikační standard pro IoT. *Root.cz - informace nejen ze světa Linuxu* [online]. Červen 2016 [cit. 2021-12-14]. Dostupné z: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>.
- [63] MCTEAR, M., CALLEJAS, Z. a GRIOL, D. *The conversational interface: Talking to smart devices*. Cham, Switzerland: Springer International Publishing, 2016. ISBN 978-3319329659.
- [64] MIDWINTER, C. J. Smart Home vs. Home Automation – What’s the difference? *149 Walnut* [online]. Srpen 2018 [cit. 1. dubna 2021]. Dostupné z: <https://149walnut.com/2018-07-smart-home-vs-home-automation-whats-the-difference/>.
- [65] O’BRIEN, H. Why 2.4GHz? Chasing wireless history. *Indiegogo* [online]. Listopad 2012. 2013-05-20. Dostupné z: <https://www.indiegogo.com/projects/why-2-4ghz-chasing-wireless-history>.
- [66] RICHTER, F. Smart Home Technology Poised for Blockbuster Growth. *Statista* [online]. Říjen 2018 [cit. 2. dubna 2021]. Dostupné z: <https://www.statista.com/chart/15736/smart-home-market-forecast/>.
- [67] SANCHEZ, J. a CANTON, M. P. *Embedded Systems Circuits and Programming*. 6000 Broken Sound Parkway NW, suite 300, Boca Raton, FL 33487-2742: CRC Press, květen 2012. ISBN 978-1439879047.
- [68] SEDLÁK, J. David Beck (Hello Emma): Nechápu, proč Apple dělá s češtinou pro Siri takové drahoty. *Lupa.cz* [online]. Květen 2020. ISSN 1213-0702. Dostupné z: <https://www.lupa.cz/clanky/david-beck-hello-emma-apple-iphone-siri/>.

- [69] SHELBY, Z., HARTKE, K. a BORMANN, C. *The Constrained Application Protocol (CoAP)* [Internet Requests for Comments]. RFC 7252. RFC Editor, June 2014. Dostupné z: <http://www.rfc-editor.org/rfc/rfc7252.txt>.
- [70] SINGAL, T. L. *Wireless communication*. New Delhi, Indie: Tata McGraw Hill Education Private Limited, 2010. ISBN 9780070681781.
- [71] SVOBODA, T. Chytrá domácnost s HomeKitem není dokonalá. Co jí chybí? *Applíště* [online]. 2017. Dostupné z: <https://www.appliste.cz/homekit-dokonalost/>.
- [72] TEAM, T. H. Quality of Service 0,1 & 2 - MQTT Essentials: Part 6. *HiveMQ* [online]. Únor 2015 [cit. 2021-12-14]. Dostupné z: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>.
- [73] TOYAMA. A quick overview on DIY Home Automation vs Professionally Installed Home Automation to help you decide. *Toyama* [online]. Březen 2021 [cit. 2. dubna 2021]. Dostupné z: <https://www.toyamaworld.com/post/home-automation-diy-or-professionally-installed>.
- [74] UPTON, E., HALFACREE, G. a GONER, J. *Raspberry Pi: uživatelská příručka. 2., aktualizované vydání*. Přeložil Jakub GONER. Brno: Computer Press, 2016. ISBN 9788025148198.
- [75] VUJOVIĆ, V. a MAKSIMOVIĆ, M. Raspberry Pi as a Sensor Web node for home automation. *Computers & electrical engineering* [online]. 2015, sv. 44, s. 153 – 171. DOI: 10.1016/j.compeleceng.2015.01.019. ISSN 0045-7906. Dostupné z: <http://www.sciencedirect.com/science/article/pii/S0045790615000257>.
- [76] VÁCLAVOVIČ, J. Chytrá domácnost a HomeKit. *Medium* [online]. Leden 2019. Dostupné z: <https://medium.com/@jvaclavovic/chytr%C3%A1-dom%C3%A1cnost-a-homekit-e16803428604>.
- [77] What is Infrared? *Cool Cosmos* [online]. [cit. 2020-01-24]. Dostupné z: http://coolcosmos.ipac.caltech.edu/cosmic_classroom/cosmic_reference/whatisir.html.
- [78] WILMSHURST, T. *Designing Embedded Systems with PIC Microcontrollers: Principles and Applications*. Elsevier Books, listopad 2009. ISBN 1856177505.
- [79] WOOD, L. We Proudly Present: The Miniserver Go! The Loxone Smart Home For Retrofitters! *Loxone.com* [online]. Září 2014. Dostupné z: <https://www.loxone.com/en/miniserver-go/>.
- [80] ZANDL, P. *Bezdrátové sítě WiFi*. Brno: Computer Press, 2003. ISBN 8072266322.

Příloha A

Zprovoznění systému na různých zařízeních

Je rozdíl jestli chce uživatel zprovoznit pouze server (a ESP), nebo i vlastní web. app...

Zprovoznění webového klienta na vlastní doméně

todo: založení firebase čtu, databáze...

[[Přesto že se jedná o open source, je potřeba na vzdálený server nahrát svou verzi klientské aplikace, nebo použít tu moji na XYZ.cz]]

Zprovoznění serveru na Raspberry Pi a alternativních mikro-počítačích

Systém je možné zprovoznit na mikropočítačích Raspberry Pi, alespoň od verze 3.

Nejprve je nutné stáhnout zdroje z githubu... následně pokud uživatel nemá na zařízení nainstalované nodejs či npm: `sudo apt update` `sudo apt install nodejs -assume-yes` `sudo apt install npm -assume-yes` Pokud by verze npm neodpovídala aktuální verzi, je potřeba ji aktualizovat: `sudo npm cache clean -f`
`sudo npm install -g n`
`sudo n stable`
`PATH="$PATH"`

Dále v adresáři `src/server` je potřeba nainstalovat všechny dependencies... `npm install`
.....

Podobný postup by měl být na alternativních mikropočítačích (jako je **[[XYZ]]**), ačkoli to nebylo testováno (jak vyplývá z kapitoly 6.6)

Zprovoznění na systémech s OS Windows a Linux???

...

Zprovoznění na systémech s OS Android

...

Příloha B

Struktura databáze a používané hodnoty