



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

JAK UDĚLAT CHYTROU DOMÁCNOST POMOCÍ OPEN-SOURCE IOT?

HOW TO MAKE A SMART HOME USING OPEN-SOURCE IOT?

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

RADIM BLAHA

Ing. JAN PLUSKAL

BRNO 2019

Zadání bakalářské práce



21809

Student: **Blaha Radim**
Program: Informační technologie
Název: **Jak udělat chytrou domácnost pomocí open-source IoT?**
How to Make the Smart Home Using Open-Source IoT?
Kategorie: Vestavěné systémy

Zadání:

1. Nastudujte komerční řešení elektrifikace chytré domácnosti a po konzultaci s vedoucím vyberte podmnožinu zařízení, které bude navržená domácnost obsahovat.
2. Navrhněte infrastrukturu chytré domácnosti, zejména elektrické obvody, osvětlení, integraci se senzory a vybranými zařízeními. Taktéž prozkoumejte volně dostupné platformy pro domácí automatizaci. Zaměřte se na robustnost celého systému, aby minimalizoval všechny kritické body náchylné k poruše.
3. Na základě návrhu, implementujte "proof-of-concept" automatizačních a sensorických prvků, dle pokynů vedoucího. Prvky integrujte s využitím vhodně zvolené platformy pro domácí automatizaci.
4. Proveďte teoretické porovnání Vašeho řešení oproti řešením již existujícím z bodu 1.

Literatura:

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

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 a 2.

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

Vedoucí práce: **Pluskal Jan, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 28. října 2018

Abstrakt

Cílem této práce je analyzovat trh s komerčními řešeními chytrých domácností a s open-source platformami, poté na základě průzkumu navrhnout vlastní řešení založené právě na open-source součástech a software. Řešení obsahuje návrh zásuvkové a světelné infrastruktury ve spojení s jejich automatizací, dále návrh infrastruktury síťové. Součástí je i návrh a implementace tří komponent sestavených z dílčích součástí a jejich integrace do zvolené platformy pro domácí automatizaci. Tyto komponenty slouží k ovládání střešního okna, ovládání garážových vrat a měření teploty. Navržené řešení je v práci následně zhodnoceno a srovnáno vůči řešením komerčním.

Abstract

The aim of this thesis is analysis of commercial and open-source home automation market, results of this research were used to create custom solution based on open-source parts and software. The solution includes design of sockets and lights infrastructures and their automation, as well as design of network infrastructure. It also includes design and implementation of three components assembled from subcomponents and their integration into a selected home automation platform. These components are used for window control, garage door control and temperature measurement. Designed solution is further evaluated and compared to commercial solutions.

Klíčová slova

IoT, internet věcí, chytrá domácnost, ESP8266, WEMOS, Home Assistant

Keywords

IoT, Internet of Things, smart home, ESP8266, WEMOS, Home Assistant

Citace

BLAHA, Radim. *Jak udělat chytrou domácnost pomocí open-source IoT?*. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jan Pluskal

Jak udělat chytrou domácnost pomocí open-source IoT?

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Pluskala. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Radim Blaha
15. května 2019

Poděkování

Rád bych poděkoval Ing. Janu Pluskalovi za vedení práce, jeho odbornou pomoc a cenné rady, které mi po dobu psaní této práce poskytoval.

Obsah

1	Úvod	5
2	Průzkum technologií a řešení pro chytré domácnosti	6
2.1	Open-source platformy pro správu IoT zařízení	6
2.1.1	ThingSpeak	7
2.1.2	ThingsBoard ve verzi zdarma	8
2.1.3	Domoticz	9
2.1.4	openHAB	10
2.1.5	Home Assistant	11
2.1.6	Srovnání zkoumaných platforem	12
2.2	WEMOS	13
2.3	Technologie používané v komerci	13
2.3.1	Z-Wave	13
2.3.2	EnOcean	14
2.4	Průzkum elektrifikace v komerční sféře	15
2.4.1	Drátové řešení	15
2.4.2	Bezdrátové řešení	16
2.4.3	Kombinace více technologií	17
2.4.4	Shrnutí zkoumaných řešení	17
3	Návrh pomocí open-source technologií	18
3.1	Software pro navrhované řešení	18
3.2	Infrastruktura	19
3.2.1	Světelné okruhy	19
3.2.2	Zásuvkové okruhy	21
3.2.3	Síťová infrastruktura pro IoT prvky	23
3.3	Vlastní IoT zařízení	24
3.3.1	Automatizace střešního okna	24
3.3.2	Ovládání garážových vrat	25
3.3.3	Měření teploty	26
3.4	Napájení a bezpečnost	27
3.5	Shrnutí navrženého řešení	28
4	Implementace a testování	29
4.1	Společná část implementace vlastních komponent	29
4.2	Automatizace střešního okna	29
4.3	Ovládání garážových vrat	34
4.4	Měření teploty	36

4.5	Integrace komponent do platformy	37
5	Zhodnocení vlastního řešení a srovnání s komerčními řešeními	39
5.1	Celková finanční náročnost	39
5.2	Srovnání návrhu s komerčními řešeními	41
5.3	Možná rozšíření	43
5.4	Závěrečné zamýšlení autora	43
6	Závěr	44
	Zkratky	45
	Literatura	46
	Přílohy	48
	Seznam příloh	49
A	Obsah CD	50
B	Zdrojové kódy ve formátu YAML	51
C	Zdroje obrázků	57

Seznam obrázků

2.1	Ukázka Android aplikace <i>ThingView</i>	8
2.2	Ukázka platformy <i>ThingsBoard</i>	9
2.3	Ukázka Android aplikace <i>Domoticz Home Automation Lite</i>	10
2.4	Ukázka Android aplikace <i>openHAB</i>	11
2.5	Ukázka Android aplikace <i>HomeAssist</i>	12
2.6	Schéma infrastruktury systému <i>Control4</i>	16
3.1	Spínač <i>Xiaomi</i>	20
3.2	Spínač <i>Sonoff Touch</i>	20
3.3	Půdorys domu se zakreslenými světelnými okruhy.	21
3.4	Chytrá zásuvka <i>BlitzWolf BW-SHP6</i>	22
3.5	Přiblížený půdorys domu se zakreslenými zásuvkovými okruhy.	23
3.6	Schéma ovládání elektrických oken.	25
3.7	Schéma ovládání garážových vrat.	26
3.8	Schéma měřiče teploty.	27
4.1	Krabice se součástmi komponenty pro ovládání střešního okna.	30
4.2	Kompletní komponenta pro ovládání střešního okna.	31
4.3	Detail instalační krabice komponenty pro ovládání střešního okna.	32
4.4	Schéma zapojení relé nakreslené v programu <i>ProfiCAD</i>	32
4.5	Schéma zapojení komponenty ovládání okna.	33
4.6	Zapojení při jedné z forem testování.	33
4.7	Zkonstruované zařízení pro ovládání garážových vrat.	34
4.8	Schéma zapojení komponenty ovládání garážových vrat.	35
4.9	Ukázka z testování komponenty.	35
4.10	Zkonstruované zařízení pro měření teploty.	36
4.11	Upravené teplotní čidlo <i>DS18B20</i>	36
4.12	Schéma zapojení komponenty pro měření teploty.	37
4.13	Vlastní zařízení integrovaná do platformy.	38

Seznam tabulek

2.1	Standard ISO/IEC 14543-3-1X (<i>EnOcean</i>).	15
3.1	Součástky pro komponentu automatizace střešního okna a popis jejich významu v rámci komponenty.	25
3.2	Součástky pro komponentu ovládání garážových vrat a popis jejich významu v rámci komponenty.	26
3.3	Součástky pro komponentu měření teploty a popis jejich významu v rámci komponenty.	27
5.1	Ceny za jednotlivé součásti komponenty pro ovládání střešního okna.	40
5.2	Ceny za jednotlivé součásti komponenty pro ovládání garážových vrat.	40
5.3	Ceny za jednotlivé součásti pro komponentu měření teploty.	41
5.4	Ceny za ostatní komponenty.	41

Kapitola 1

Úvod

Ke spoustě komerčních řešení existují alternativy v podobě open-source zařízení a software, není tomu jinak ani v případě internetu věcí (IoT). Vzhledem k finančním náročnostem komerčních projektů je snahou této práce přinést alternativní řešení ve smyslu open-source filozofie. Práce se proto zabývá průzkumem open-source platforem a zařízení využitelných pro správu chytrých domácností a průzkumem existujících komerčních řešení. Na základě nabytých vědomostí o existujících open-source zařízeních a software je ukázán možný návrh chytré domácnosti včetně elektrické a síťové infrastruktury a několika vlastních komponent řízených deskou *WEMOS*. Tyto komponenty jsou následně implementovány a integrovány do zvolené open-source platformy. Aby vynikl rozdíl mezi použitím komerčního řešení a navrženého open-source, tak je součástí i porovnání těchto řešení z pohledu finanční náročnosti, případně náročnosti na implementaci.

Dle analýzy z roku 2018 bylo ve zmíněném roce připojeno 7 miliard IoT zařízení [13]. Při předpokládaném nárůstu počtu takovýchto zařízení o zhruba 17% ročně, bude v roce 2025 takovýchto zařízení již 21,5 miliardy [13]. Na základě těchto informací můžeme tedy usuzovat zvýšený zájem o IoT zařízení a s nimi spojené chytré domácnosti v budoucnosti. Právě tento předpoklad je základem pro vznik celé práce.

Práce je rozdělena dle obsahu do několika kapitol, kde kapitola *Průzkum existujících řešení a používaných technologií v IoT* obsahuje popis vybraných SW open-source platforem, pro řízení chytrých domácností, a jejich základní srovnání. Součástí je průřez jednotlivými technologiemi používanými v komerčních řešeních chytrých domácností. Poté jsou na příkladech vybraných komerčních systémů popsány různé řešení elektrifikace IoT zařízení.

V kapitole *Návrh pomocí open-source technologií* je popisováno vlastní navržené řešení včetně elektrické a síťové infrastruktury. Čtenář se zde může dočíst o schopnostech, které navrhovaná domácnost má, a jaká zařízení se použijí. Dále je zde podrobnější návrh tří vlastních komponent včetně blokových schémat a řeší se zde i způsob ovládání pomocí zvolené platformy.

Kapitola *Implementace a testování* ukazuje konstrukci jednotlivých zařízení, a to včetně jejich podrobných schémat. Dále je zde popsáno testování jednotlivých zařízení.

Obsah kapitoly *Zhodnocení vlastního řešení a srovnání s komerčními řešeními* pojednává o finanční náročnosti jednotlivých vlastních komponent a provádí odhad jejich ceny. Obdobně pojednává o ceně celé navržené chytré domácnosti. Text poté provádí srovnání vlastního řešení chytré domácnosti s řešeními komerčními.

Kapitola 2

Průzkum technologií a řešení pro chytré domácnosti

Kapitola pojednává o používaných platformách pro ovládání chytré domácnosti a zobrazování dat ze senzorů. U těchto platforem jsou popsány způsoby instalace, podporované protokoly, aplikace, které s danou platformou interagují a další. Na konci sekce čtenář nalezne shrnutí a závěrečné srovnání platforem. Zde popisovanými platformami jsou:

- *ThingSpeak*
- *ThingsBoard*
- *Domoticz*
- *openHAB*
- *Home Assistant*

V textu kapitoly jsou popsány některé technologie používané v komerci. Dále zde jsou vybráni zástupci několika komerčních systémů pro domácí automatizaci a tito zástupci jsou kategorizováni dle způsobu napájení zařízení. Jako zástupce plně drátového řešení byl vybrán systém *Control4*. Jako zástupce bezdrátového řešení byl vybrán systém *Fibaro* a zástupcem kombinace drátového a bezdrátového řešení je systém *Loxone*.

Mimo průzkum prováděný v této kapitole existuje také velmi citovaný článek [9] na téma IoT s názvem *Internet of Things (IoT): A vision, architectural elements, and future directions*. V tomto článku lze nalézt mimo jiné vizi o tom, jakým směrem se bude ubírat svět internetu věcí.

2.1 Open-source platformy pro správu IoT zařízení

Dle průzkumu prováděného v rámci této práce existuje spousta platforem pro komunikaci s IoT zařízeními. Některé slouží převážně pro vizualizaci dat, jiné slouží k ovládání, některé zase umožňují interagovat se zařízeními na základě změny stavu telefonu. Spousta z platforem má ale ve větší či menší míře implementovány kombinace předešlých vlastností. Zde jsou vybrány a popsány některé z takových open-source platforem. Platformy jsou porovnávány dle způsobu instalace, podporovaných protokolů, dle použití na jednotlivých zařízeních (telefon, web, desktop) a dle jejich funkcionalit.

Vzhledem k očekávanému použití, jsou především na obrázcích v této sekci zobrazovány snímky obrazovky z mobilních verzí aplikací dané platformy. Konečné srovnání na konci této sekce se zaměřuje na funkcionality poskytované při použití na chytrém telefonu.

2.1.1 ThingSpeak

ThingSpeak je platforma a zároveň API pro sběr, analyzování a vizualizaci dat získaných ze senzorů. Díky jejímu vývojáři, kterým je *MathWorks*, umožňuje využívat programovací jazyk *MATLAB* ke zpracování dat ze senzorů.

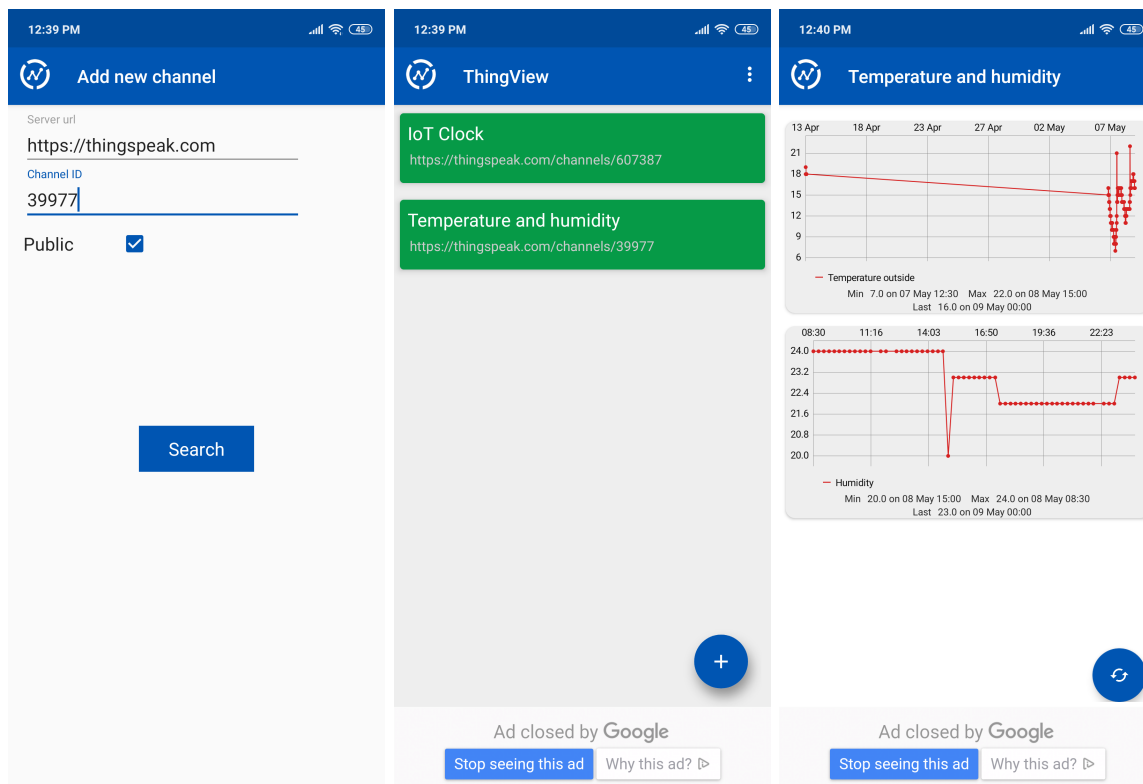
Platforma umožňuje mimo analýzy a vizualizace dat dále publikovat tweety na sociálních sítích, například publikovat informaci o změně stavu nějakého senzoru. Dále umožňuje provádět akce naplánované na určitý čas nebo ovládat zařízení, když nastane určitá podmínka, například, když je poloha uživatelova telefonu poblíž domu, může rozsvítit světla [16]. Obdobně může reagovat, když jiné zařízení zjistí změnu stavu, například přítomnost osoby v místnosti [16].

Vybrané vlastnosti

- **Instalace** Bez instalace, dostupné z cloudu.
- **Podporované protokoly HTTP**.
- **Aplikace** Webová aplikace v responzivním designu pro správu a vizualizaci dat. Případně pro Android existuje aplikace *ThingView — ThingSpeak viewer*¹ (viz obrázek 2.1). Pro iOS pak *ThingView*² od autora se jménem Marcelo Prolo, která slouží pouze pro vizualizaci dat z připojených kanálů.
- **Správa Ovládání**, pokyny na základě událostí, vizualizace.

¹ThingView pro OS Android dostupné z: https://play.google.com/store/apps/details?id=com.cinetica_tech.thingview

²ThingView pro iOS dostupné z: <https://itunes.apple.com/us/app/thingview/id1284878579?mt=8>



Obrázek 2.1: Ukázka Android aplikace *ThingView*. Na prvním obrázku je vidět snímek obrazovky s přidáváním sledovaného kanálu. Na druhém obrázku je seznam sledovaných kanálů. Na třetím obrázku jsou v grafech vizualizované hodnoty z konkrétního kanálu.

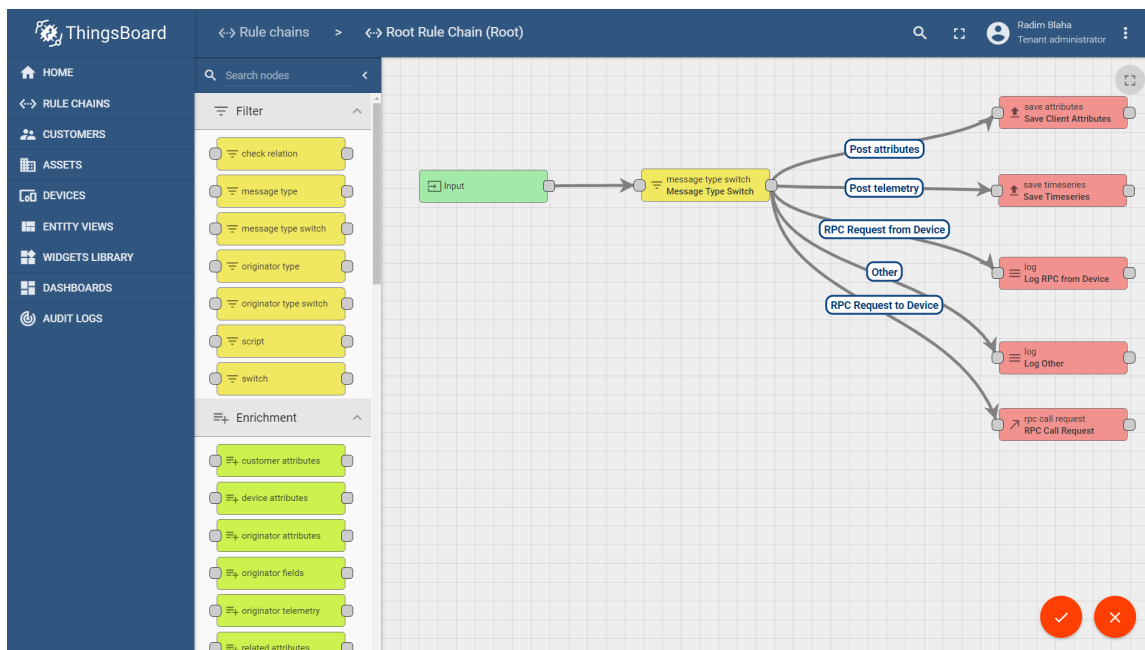
2.1.2 ThingsBoard ve verzi zdarma

Tato platforma existuje i v placené verzi, kde poskytuje širší možnosti, zde jsou popsány vlastnosti verze, která je dostupná zdarma. Aplikace umožňuje poskládat přehlednou obrazovku, složenou například z grafů naměřených hodnot nebo z grafických widgetů, které ukazují aktuální hodnoty.

ThingsBoard umožňuje vytvářet tzv. rule chains. V aplikaci se nacházejí bloky, které lze různě nastavovat a propojovat mezi sebou. Propojením bloků lze definovat vzájemnou interakci jednotlivých částí systému. Na obrázku 2.2 je zobrazena webová verze platformy, nicméně díky použitému responzivnímu designu je možné obdobným způsobem používat aplikaci i na mobilním zařízení.

Vybrané vlastnosti

- **Instalace** Docker, Windows/Linux, AWS EC2, ...
- **Podporované protokoly** HTTP, MQTT, CoAP.
- **Aplikace Web** s responzivními prvky — pohodlné ovládání i z telefonu.
- **Správa** Sběr dat, zpracování, vizualizace a ovládání zařízení (viz obrázek 2.2).
- **Ostatní** Automatická detekce selhání uzlu, chybný uzel může být okamžitě nahrazen s nulovým časem výpadku.



Obrázek 2.2: Ukázka platformy *ThingsBoard*. Editor pro vytváření tzv. rule chains, pomocí kterých lze definovat chování systému.

2.1.3 Domoticz

Domoticz je odlehčený systém pro domácí automatizaci, který umožňuje konfigurovat a monitorovat různá zařízení (světla, vypínače, senzory, ...). Zprávy o stavu těchto zařízení mohou být odesílány do mobilních zařízení [2].

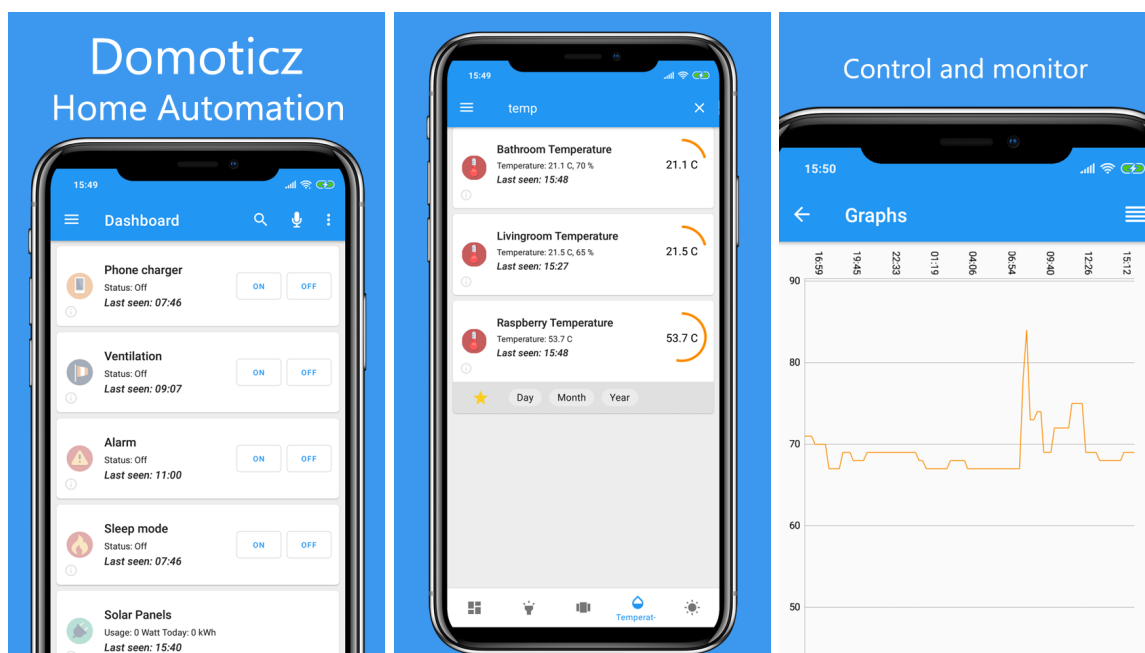
Vybrané vlastnosti

- **Instalace** Windows, Linux, MacOS, Synology NAS, FreeNAS.
- **Podporované protokoly** Možnost využití Python modulů — například MQTT discovery a další³.
- **Aplikace** Webová aplikace. Pro iOS/Android je k dispozici aplikace s názvem *Domoticz — Home Automation*⁴ (viz obrázek 2.3) nebo *ImperiHome*⁵ a jiné. Pro OS Tizen existuje *Gearoticz* [3], díky tomuto je možné využívat platformu i na chytrých hodinkách.
- **Správa** Díky široké škálovatelnosti platformy je možné platformu nastavit různými způsoby.
- **Ostatní** HA clustering — možnost jednoduchého řešení s vysokou dostupností o 2 uzlech při instalaci na Synology NAS (služba Synology High Availability).

³Seznam modulů je dostupný na: <https://www.domoticz.com/wiki/s>

⁴Domoticz pro OS Android dostupný z <https://play.google.com/store/apps/details?id=nl.hnogames.domoticz>, pro iOS dostupný z <https://itunes.apple.com/app/id1251318855>.

⁵ImperiHome pro OS Android dostupný z: <https://play.google.com/store/apps/details?id=com.imperihome.lite>, pro iOS dostupný z <https://itunes.apple.com/cz/app/imperihome-smart-home-city/id987826098?mt=8&ign-mpt=uo%3D4>.



Obrázek 2.3: Ukázka Android aplikace *Domoticz Home Automation Lite*. První snímek ukazuje stránku s nastavením komponent chytré domácnosti a zobrazením dat ze senzorů. Druhý snímek zobrazuje teploty naměřené jednotlivými senzory. Na třetím snímku je možno vidět v grafu vizualizovanou hodnotu naměřenou pomocí některého ze senzorů.⁶

2.1.4 openHAB

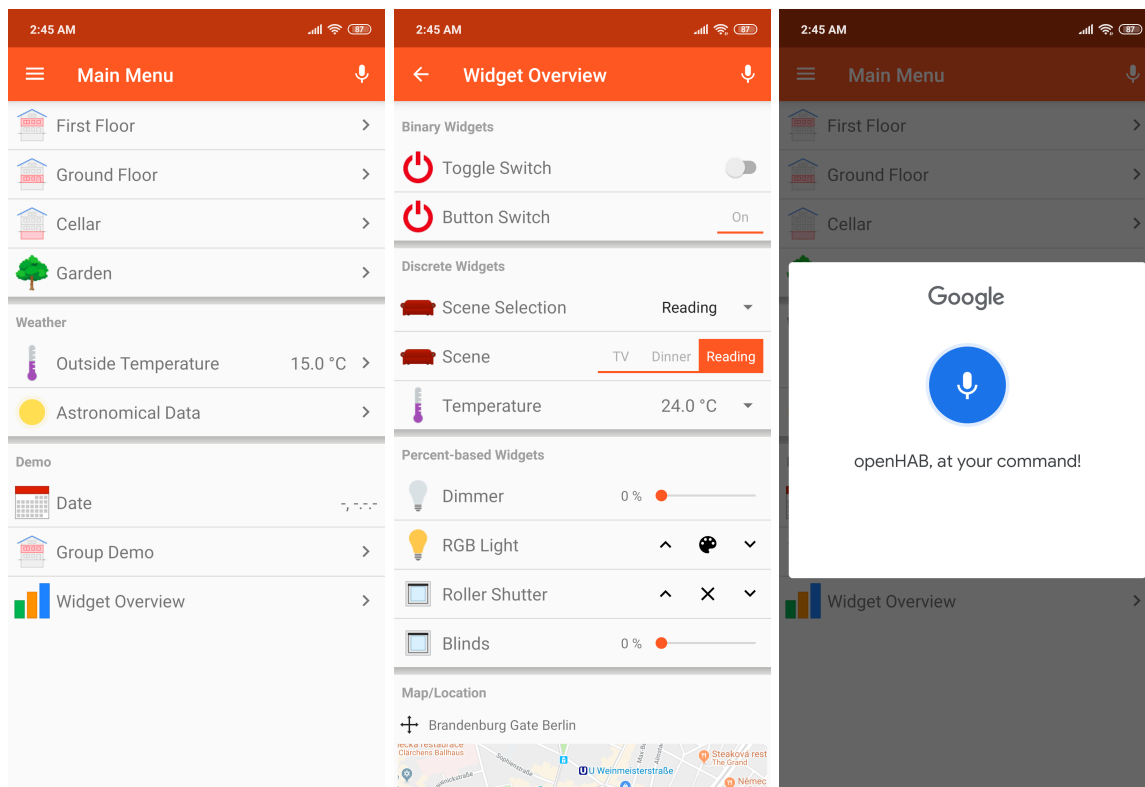
Platforma *openHAB* integruje rozdílné systémy pro domácí automatizaci, zařízení a technologie do jednoho řešení. Poskytuje jednotné uživatelské rozhraní a společný přístup, bez ohledu na počet zúčastněných podsystémů [14].

Vybrané vlastnosti

- **Instalace** Linux, Windows, MacOS, dostupný i image SD karty s názvem openHABian (nakonfigurovaný Linux s openHAB), Docker, Synology DiskStation, QNAP NAS.
- **Podporované protokoly** Pro připojení různých zařízení lze doinstalovat moduly pro komunikaci se zařízeními chytré domácnosti. Lze tak nainstalovat podporu pro MQTT, Z-Wave, Zigbee, 1-Wire, EnOcean a mnohé další⁷. Aktuálně (leden 2019) podporuje openHAB již více jak 200 rozdílných technologií a systémů a tisíce zařízení [14].
- **Aplikace** Oficiální aplikace *openHAB* pro Android (viz obrázek 2.4), iOS, Windows 10, REST API, možnost propojení s Amazon Alexa, Google Assistant, Apple Home Kit, IFTTT, Mycroft.AI.
- **Správa** Ovládání a nastavování zařízení domácnosti, zobrazování dat ze senzorů.

⁶Převzato z: <https://play.google.com/store/apps/details?id=nl.hnogames.domoticz>

⁷Seznam modulů je dostupný na: <https://www.openhab.org/addons/>



Obrázek 2.4: Ukázka Android aplikace *openHAB*. První obrázek ukazuje hlavní menu aplikace a údaje ze senzorů. Na druhém obrázku je vidět snímek části aplikace, kde je možné nastavovat jednotlivé komponenty domácnosti. Na třetím snímku je zobrazeno použití hlasového asistenta Google pro ovládání aplikace, respektive chytré domácnosti.

2.1.5 Home Assistant

Zde popisovanou zkoumanou platformou je *Home Assistant*. Tato platforma je ve své podstatě velice podobná platformě *openHAB* (viz sekce 2.1.4). Platforma je vhodná ke spuštění na *Raspberry Pi* nebo na lokálním serveru [14].

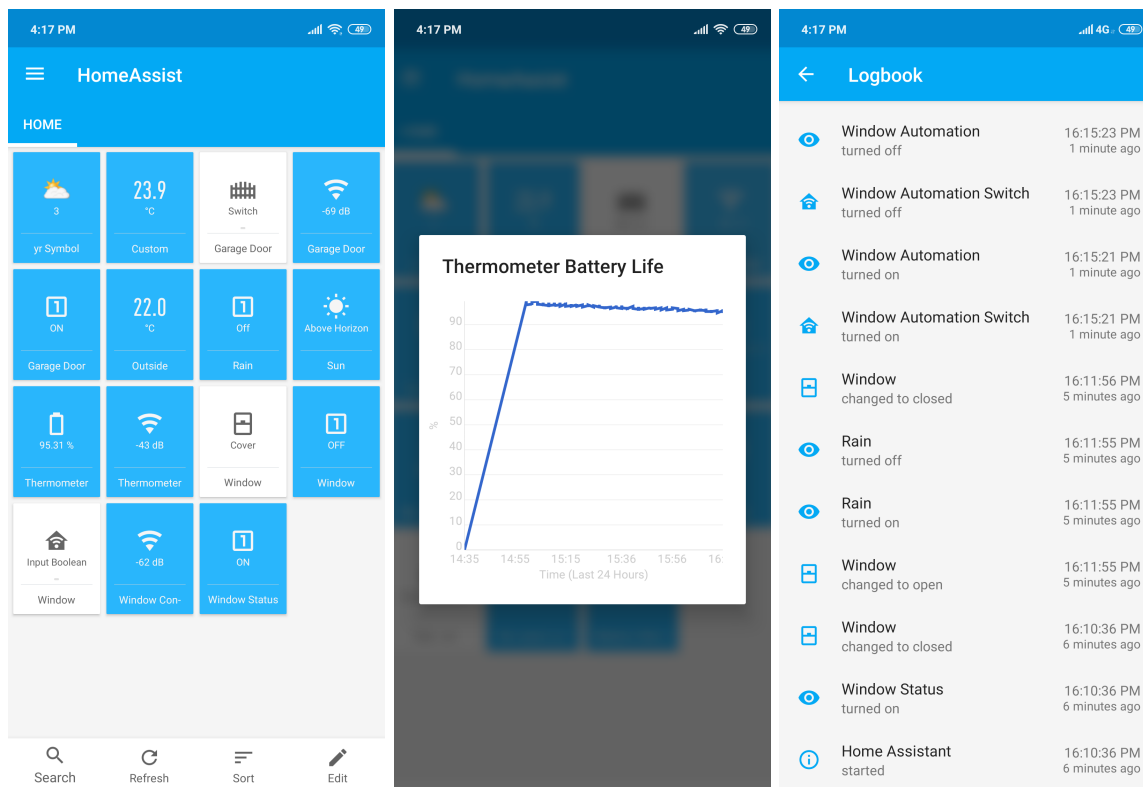
Vybrané vlastnosti

- **Instalace** Manuální instalace na Linux, Docker, dostupný image pro Raspberry Pi nebo image pro virtuální stroje.
- **Podporované protokoly** Home Assistant je platforma, která umožňuje přidávat funkce pomocí doinstalování komponent. Mezi komponentami dostupnými na stránkách platformy Home Assistant⁸ jsou protokoly a technologie jako MQTT, Z-Wave, IFTTT, EnOcean a další. Samozřejmostí je i podpora Amazon Alexa nebo Google Assistant.
- **Aplikace** Responzivní webová aplikace, jednoduchý přístup k ní lze zajistit například přidáním záložky prohlížeče na plochu v telefonu. Dále pro Android existuje napří-

⁸Seznam komponent je dostupný na: <https://www.home-assistant.io/components/>

klad aplikace *HomeAssist* od vývojáře axzae⁹ (viz obrázek 2.5), pro iOS je to *Home Assistant Companion* od autora se jménem Robert Trencheny¹⁰.

- **Správa** Ovládání a nastavování zařízení domácnosti, zobrazování dat ze senzorů.



Obrázek 2.5: Ukázka Android aplikace *HomeAssist*. Na prvním obrázku je zobrazen snímek obrazovky zařízení a senzorů a je možné je na této stránce nastavovat. Na druhém snímku je vidět graf vývoje vybíjení baterie. Třetí snímek ukazuje historii změn stavů komponent domácnosti.

2.1.6 Srovnání zkoumaných platforem

Vybrané platformy byla snaha seřadit dle množství jejich funkcionalit, které poskytují při použití na mobilních zařízeních. Každá z platforem poskytuje mírně odlišné funkcionality.

ThingView Mobilní aplikace *ThingView* platformy *ThinkSpeak* umožňuje pouze připojování k různým kanálům a vizualizaci dat, nicméně je možné využít responzivní verzi webu *ThingSpeak*, která obsahuje veškeré nabízené funkcionality. Pro zjednodušení přístupu na web lze přidat záložku na plochu v telefonu.

⁹HomeAssist pro OS Android dostupný z: <https://play.google.com/store/apps/details?id=com.axzae.homeassistant>

¹⁰Home Assistant Companion pro iOS dostupný z: <https://itunes.apple.com/us/app/home-assistant-open-source-home-automation/id1099568401?mt=8>

ThingsBoard Pro platformu *ThingsBoard* mobilní aplikace nejsou, nicméně web i této platformy je napsán za použití responzivních prvků, takže se dá aplikace pohodlně využívat i na telefonu. Klíčovou funkcionalitou této platformy je definování chování systému pomocí nastavování a propojování bloků.

Domoticz *Domoticz* je systémem, pro který existuje spousta modifikací jeho vzhledu a dále i několik klientských platform. Moduly jsou napsány v Pythonu, ale jejich instalace je mnohem méně intuitivní než u následujících dvou platform. Vše je sepsáno na nepříliš přehledných Wiki stránkách.

openHAB Pro platformu *openHAB* existuje stejnojmenná oficiální aplikace. Do platformy je možné nainstalovat velké množství modulů. Instalace a jejich nastavování je v dokumentaci poměrně dobře popsáno.

Home Assistant Další zmíněná platforma *Home Assistant* je v mnohém podobná platformám *Domoticz* a *openHAB*, alespoň co se týče rozšiřování pomocí modulů. Instalace modulů se provádí přímo skrze webovou platformu a tento způsob instalace a nastavování se zdá být nejvíce pohodlný. *Home Assistant* umožňuje instalaci velkého množství modulů, a to i takových, za pomoci kterých je možné sledovat například cenu Bitcoinu. Aplikace pro platformu se dá využít buď webová, dostupná z oficiálních stránek nebo některá ze zmíněných mobilních aplikací.

2.2 WEMOS

Po open-source *Arduino* deskách, které jsou založeny na mikrokontrolérech *Atmel* [1], se v IoT světě rozmohl trend zařízení založených na mikročipu *ESP8266*. Zařízení obsahující tento čip jsou využitelná pro spoustu aplikací, protože obsahuje mimo jiné Wi-Fi přijímač/vysílač a 16 GPIO (general-purpose input/output) pinů [18]. Existuje spousta desek založených na tomto čipu, které zpřístupňují piny pro snadné připojení periférií. Jedná se například o desky *ESP-12* [15], *NodeMcu* nebo *WEMOS*.

Právě desku *WEMOS*, konkrétně *WEMOS D1 mini*, lze využít pro vytváření vlastních IoT zařízení. To díky své velikosti, počtu pinů, shieldů, které na ni existují a její ceně. Tato konkrétní deska zpřístupňuje 11 vstupně/výstupních pinů, má jeden analogový vstup a 4 MB čip s pamětí typu flash [17]. Rozměry desky potom jsou 34,2 mm na 25,6 mm, operační napětí je 3,3 V [17]. Desku je ale možné napájet pomocí napětí 5 V, například přes microUSB konektor, přes který ji lze mimo to i programovat.

2.3 Technologie používané v komerci

Tato sekce má za úkol informovat o technologiích používaných v komerčních řešeních popsaných v sekci 2.4 této kapitoly. Konkrétně se jedná o komunikační protokol *Z-Wave* a je zde vyzdvihnuta technologie nazývaná *EnOcean*. Tato technologie je zajímavá především kvůli jejímu způsobu napájení a komunikace.

2.3.1 Z-Wave

Z-Wave je bezdrátový komunikační protokol pro komunikaci mezi zařízeními v rámci inteligentních domácností, v Evropě operující na frekvenci 868 MHz [8, 19]. Protokol byl původně

vytvořený již v roce 2001, ale až v roce 2016 byly jisté části tohoto protokolu poskytnuty veřejnosti jako open-source [19]. Jedná se o části zabezpečení (Security 2), přenosu přes IP (Z/IP) a Z-Wave middleware [19].

Z-Wave využívá topologii typu mesh, takže zařízení Z-Wave jsou schopna se připojit na jiná zařízení podporující tuto technologii a tím je celý systém odolný vůči výpadku jednoho prvku a dále je takto zajištěno i připojení zařízení, která by byla za normálních okolností příliš daleko od přístupového bodu.

Obrovskou výhodou tohoto protokolu je, že veškerá zařízení, která komunikují skrze tento protokol jsou certifikovaná, což zajišťuje standardizaci protokolu a kompatibilitu napříč zařízeními různých značek a výrobců [19]. Certifikace zařízením jsou udělovány organizací s názvem *Z-Wave Alliance*, která byla ustavena v roce 2005 jako sdružení společností, které vytváří produkty a služby založené na technologii Z-Wave [19].

Dle oficiálních stránek aliance je v současnosti (konec roku 2018) členem konsorcia 700 společností a existuje již více než 2400 produktů podporujících tuto technologii [22]. Součástí konsorcia jsou mimo jiné i velké společnosti jako například Honeywell, Huawei, Sharp Corporation a další¹¹.

Zařízení Z-Wave je možné připojit například do open-source platformy *Home Assistant* pomocí některého ze zařízení nazývaných Z-Wave USB Stick. Seznam podporovaných zařízení tohoto typu je k nalezení na stránkách platformy¹². O platformě *Home Assistant* podrobněji pojednává sekce 2.1.5.

2.3.2 EnOcean

Jednou z technologií, která stojí za zmínku je technologie *EnOcean*. Tato technologie umožňuje provozovat zařízení bez potřeby externího napájení anebo výměny baterií a vystačí si s čerpáním energie z alternativních zdrojů, využívá totiž techniku energy harvesting.

Energy harvesting jsou techniky pro získávání energie například ze slunečního záření, rozdílů teplot nebo z pohybové energie [20]. Bezdrátový senzor pro energy harvesting obvykle sestává z antény, vysílače, energy harvesting jednotky, power management jednotky, senzorové/procesorové jednotky a někdy i baterie [10]. Přestože energie takto získaná může být poměrně malá, je možné ji využít v nízkopříkonových zařízeních pro odeslání malého množství dat. V současné době je bezdrátová technologie *EnOcean* mezinárodním standardem ISO/IEC 14543-3-1X [4].

EnOcean protokol Protokol *EnOcean* využívá standardů ISO/IEC 14543-3-1X a IEEE 802.15.4 [6], kde druhý zmíněný je základem i pro protokol *Zigbee* [7, 8] nebo *WirelessHART*. Pro přenos 1 B dat ze senzoru, což je velikost dostačující pro přenos naměřených hodnot, požaduje pouze 7 B režijních dat [6].

Vlastnosti ISO/IEC 14543.3.1x Vybrané vlastnosti mezinárodního standardu, obsah tabulky 2.1 vychází z webu *EnOcean* [5].

¹¹Seznam aktuálních členů Z-Wave Alliance je dostupný z: https://z-wavealliance.org/z-wave_alliance_member_companies/

¹²Zařízení Z-Wave USB Stick podporované platformou *Home Assistant* dostupné z: <https://www.home-assistant.io/docs/z-wave/controllers/>

Vysoká spolehlivost	v Evropě běží na frekvenci 868 MHz
	přenos více zpráv s kontrolním součtem
	krátké zprávy s dobou přenosu zhruba 1 ms snižují pravděpodobnost kolizí s ostatními EnOcean senzory
	velký dosah: až 30 metrů v budovách a 300 metrů v otevřeném prostoru
	jednosměrná i obousměrná komunikace
	pokročilé zabezpečení dat pomocí rolling code a 128 bitové AES šifrování
Nízká spotřeba	přenos dat rychlostí 125 kbit/s
	malá velikost režijních dat
	ASK a FSK modulace
Interoperabilita	založen na mezinárodních standardech
	jedinečné ID přenosu (32 bit)

Tabulka 2.1: Standard ISO/IEC 14543-3-1X (*EnOcean*). Tabulka zobrazuje vlastnosti tohoto standardu.

Zajímavé statistiky Dle stránek *EnOcean* lze z pouhých $50 \mu\text{Ws}$ uvnitř budovy přenést signál až na 30 metrů dlouhou vzdálenost [5]. Další zajímavou informací je, že spolehlivost přenosu je lepší, než 99,99 % pro 100 bezdrátových senzorů, kde každý odesílá data jednou za minutu [5].

Zařízení *EnOcean* je možné připojit například do open-source platformy *Home Assistant*, pomocí USB brány. Postup na připojení do platformy je uveden na jejích stránkách¹³. O platformě *Home Assistant* podrobněji pojednává sekce 2.1.5.

2.4 Průzkum elektrifikace v komerční sféře

Při průzkumu řešení elektrifikace na poli komerčních chytrých domácností byli vybráni tři zástupci. Každý ze zástupců nabízí odlišný způsob řešení. Jeden uváděný systém je plně drátový a ovládání je prováděno z centrálního bodu. Další systém je systémem využívajícím ke komunikaci *Z-Wave*, tedy je bezdrátový. Třetí systém je kombinací více technologií, drátových i bezdrátových. Vybrané systémy jsou následně v závěru této sekce vzájemně srovnány.

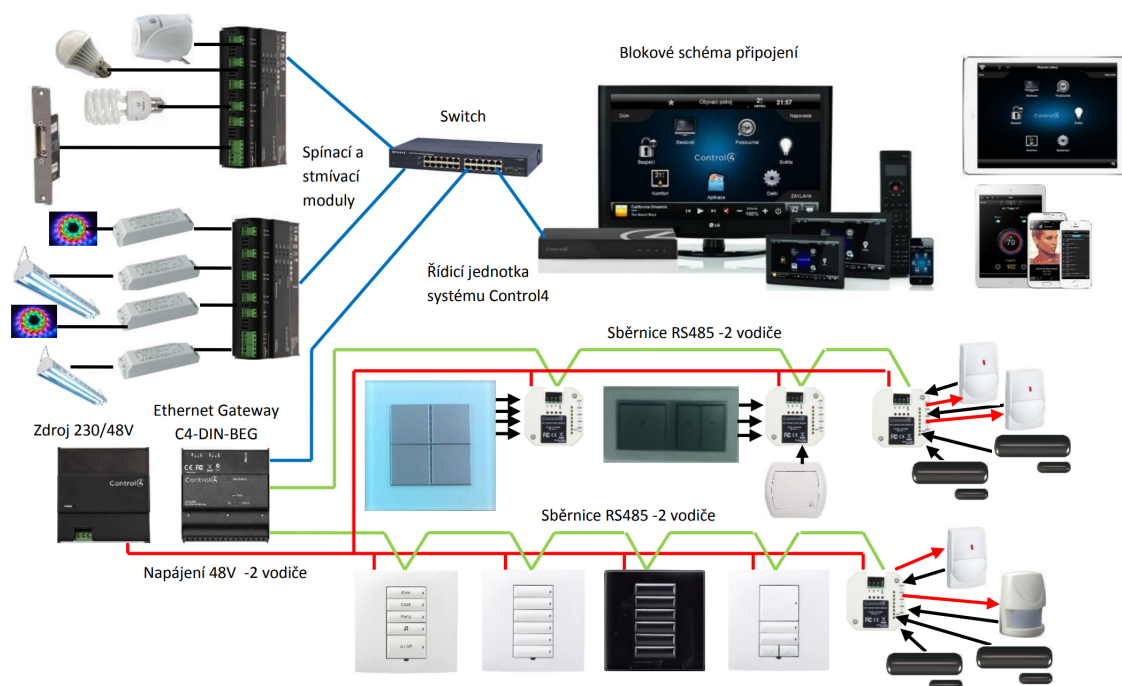
2.4.1 Drátové řešení

Při průzkumu drátového řešení byla uvažována zařízení *Control4*. *Control4* pracuje s elektroinstalačními moduly, skrze které se ovládají zařízení připojená do systému. Existují varianty elektroinstalačních modulů vhodné pro ovládání různých zařízení. Moduly jsou využitelné například pro stmívání světel, spínání spotřebičů pomocí relé modulů, ovládání žaluzií a jiné.

Z obrázku 2.6 je vidět, že systém využívá pro komunikaci dva datové vodiče a standardizovaný sériový komunikační protokol RS-485. Prvky systému jsou dále napájeny z centrálního zdroje, který konvertuje síťové napětí 230 V na 48 V stejnosměrného napětí. Za

¹³Postup pro připojení *EnOcean* zařízení do platformy *Home Assistant*: <https://www.home-assistant.io/components/enocan/>

účelem napájení je tedy nutné vést 2 vodiče. Celkově to potom pro komunikaci a napájení znamená 4 vodiče, které je nutné vést ke každému ovládacímu prvku.



Obrázek 2.6: Schéma infrastruktury systému Control4. Na příkladu infrastruktury je možno vidět, že se využívá sériové komunikace RS-485. Dále je zde vidět, že je využito čtyř vodičů, dva pro napájení a dva pro data.¹⁴

2.4.2 Bezdrátové řešení

Pro reprezentaci systému bezdrátového řešení domácnosti byl vybrán systém *Fibaro*. Tento systém uvažuje instalaci bez nutnosti rekonstrukcí nebo rozsáhlých stavebních úprav, proto je vhodný do již existujících domů.

Zařízení *Fibaro* využívají pro komunikaci technologii *Z-Wave* (viz sekce 2.3.1), takže je zapotřebí k systému zakoupit ještě některé ze zařízení *Fibaro Home Center*, které slouží jako řídicí jednotka, která je na jedné straně připojená do datové sítě a na straně druhé komunikuje skrze *Z-Wave* s moduly systému.

Dle průzkumu zařízení poskytovaných touto společností jsou zařízení napájena buď bateriemi, přímo z rozvodné sítě, případně je možné si vybrat. V případě detektoru pohybu je napájeno z baterií, v případě bezdrátové zásuvky se odebírá proud pro provoz chytrých funkcí zásuvky přímo z rozvodné sítě. Pokud je do systému zapojen detektor zaplavení, tak zde je možné napájet buď z baterií, nebo ze sítě.

U bateriově napájených zařízení je zajímavá i výdrž, kterou výrobce udává u detektoru pohybu a detektoru zaplavení na 2 roky. Termostatická hlavice pro ovládání topení by poté měla vydržet celou topnou sezónu. Díky použití *Z-Wave* je zařízení možno připojit i do systémů jiných výrobců [21], kteří jsou součástí *Z-Wave Alliance* [19].

¹⁴Převzato z: <https://www.control4.cz/wp-content/uploads/modul-4-vstupu-pod-vypinac-2014-11-29-cz.pdf>

2.4.3 Kombinace více technologií

Jako zástupce systému, který kombinuje více technologií byl vybrán systém *Loxone*. Do systému je možné nainstalovat například zařízení komunikující pomocí RS232, RS485, 1-Wire a další. Systém má mimo jiné podporu i pro *EnOcean*. Tato technologie byla popsána v sekci 2.3.2. Podpora právě pro takové technologie je v systému zajišťována pomocí rozšiřujících modulů (tzv. extensions).

Loxone Tree Společnost *Loxone* vyvinula vlastní technologii pro komunikaci s perifériemi v chytrém domě, tu nazývá *Loxone Tree*. Pro vytváření *Loxone Tree* se využívá tzv. *Tree Extension*, což je zařízení pro připojení periférií. Toto zařízení je napájeno 24 V stejnosměrného napětí a připojuje se do systému pomocí *Loxone Link*. Pomocí tohoto zařízení mohou být vytvořeny dvě větve *Loxon Tree*, kde každá z větví pojme až 50 zařízení, tzn. 100 zařízení na jeden Extension. *Loxon Tree* tedy umožňuje připojovat několik zařízení na jeden kabel a tím zjednodušuje instalaci.

Loxone Air Do systému *Loxone* je možné připojovat zařízení i bezdrátově za pomoci *Loxone Air*. *Loxone Air* je bezdrátová technologie pro komunikaci vyvinutá firmou *Loxone*. Používá topologii typu mesh, implementuje IPsec, umožňuje obousměrnou komunikaci a technologie, dále dle stránek výrobce minimalizuje spotřebu zařízení využívajících *Loxone Air* a díky tomu je možné zařízení napájet z baterie i roky [12].

2.4.4 Shrnutí zkoumaných řešení

Tito odlišní zástupci ukazují, jakými způsoby je možné chytrou domácnost řešit. Systém *Control4* využívá vodiče jak pro napájení, tak pro datovou komunikaci. Systém *Fibaro* využívá bezdrátové technologie a napájení jednotlivých zařízení provádí buď z baterií nebo z elektrické sítě. Řešení *Loxone* používá drátovou i bezdrátovou datovou komunikaci a způsob napájení záleží na použitém zařízení, napájí se po kabelu ze zdroje, z elektrické sítě nebo z baterie.

Kapitola 3

Návrh pomocí open-source technologií

Tato kapitola se zaměřuje na poskytnutí komplexního návrhu chytré domácnosti, a to včetně návrhu infrastruktur. IoT zařízení/komponenty, které obsahuje jsou buď komponenty založené na volně dostupných prvcích, anebo zařízení různých výrobců, které lze spravovat pomocí open-source SW. Oba typy prvků lze integrovat do zde zvolené platformy a přes tuto platformu ovládat a monitorovat. V případě pojmu komponenta/zařízení chytré domácnosti práce hovoří o dílčím funkčním celku, tedy o jednom celistvém IoT zařízení.

Dílčí část návrhu se zaměřuje na to, jak by měla být řešena infrastruktura pro osvětlení a zásuvky v chytré domácnosti. Dále je zde teoreticky navrhována i síťová infrastruktura včetně serveru a na něm spouštěných služeb.

Z důvodu potřeby několika specifických zařízení je v návrhu uvažováno i s vytvořením tří vlastních komponent s deskou *WEMOS*, jako řídicím členem. Počítá se s automatizací střešního okna ovládaného mj. na základě dat získaných z teplotního a vlhkostního senzoru, ovládáním současných garážových vrat a s navržením teplotních senzorů pro instalaci na několika místech v domácnosti.

3.1 Software pro navrhované řešení

Pro funkčnost navrhovaného hardware je nutné navrhnout i software, který bude celý systém pohánět. Je třeba zvolit platformu pro správu jednotlivých komponent a zaměřit se i na programovou implementaci jednotlivých komponent.

Serverová část

Pro správu IoT zařízení byla zvolena platforma *Home Assistant*, respektive její vylepšení v podobě *Hass.io*. Důvodem volby této platformy jsou široké možnosti použití v porovnání s ostatními zkoumanými platformami.

Instalace platformy *Hass.io* proběhne na serveru s deskou *Alix*. Tato deska je zmíněna v sekci 3.2.3. Platforma bude instalovaná nad operačním systémem *Debian* a spouštěna v rámci *Docker* kontejnerů, dle postupu uvedeného na stránkách platformy¹.

¹Instalace Hass.io: <https://www.home-assistant.io/hassio/installation/>

Klientská část

Pro správu chytré domácnosti se dá využít výchozí webová aplikace platformy, nebo je možné využít aplikaci pro chytrý telefon. Některé aplikace, které je možné využít pro připojení k serveru, byly vypsány v sekci 2.1.5 (například Android aplikace *HomeAssist*).

Z důvodu maximálního využití funkcionalit poskytovaných platformou bude ale ke správě IoT zařízení z klientského zařízení využito právě výchozí webové aplikace. Ke zobrazení webu s platformou se poté využije libovolný moderní prohlížeč.

Programování IoT zařízení

Pro programování IoT zařízení bude využito volně dostupných knihoven a dále open-source frameworku *esphomelib*². Tento framework a zároveň knihovna slouží pro jednodušší programování zařízení založených na ESP8266/ESP32. Umožňuje například snadné připojení k Wi-Fi a nastavení OTA aktualizací. Dále například zjednodušení programování komunikace se senzory DS18B20 nebo DHT22, zesnadnění programování ovládání trojbarevných LED pásků a spoustu dalších zjednodušení pro programování IoT zařízení.

Takto naprogramovaná zařízení se dají připojit do platformy *Home Assistant*, kde se dá zařízení vyhledat a přidat do systému skrze *MQTT Discovery*. Případně se pro komunikaci s platformou MQTT nemusí využívat a místo něj mohou zařízení komunikovat s platformou pomocí *ESPHome API*.

Při implementaci vlastních zařízení bude využít doplněk pro *Hass.io*, kde se dá kód pro zařízení psát na vyšší míře abstrakce, než je C/C++, konkrétně v *yaml* formátu. Příklady pro jednotlivé komponenty a postup instalace doplňku je dostupný na stránkách frameworku². Skrze tento doplněk budou také prováděny OTA aktualizace IoT komponent.

3.2 Infrastruktura

Pro kompletní návrh je potřebné vyřešit problematiku elektrických rozvodů, a to jak světelných, tak zásuvkových. Zároveň je třeba navrhnout napojení na IoT prvky, které budou ovládat elektrické okruhy. Návrh se zamýšlí i nad více možnými způsoby ovládání těchto okruhů. Dále je zde proveden i návrh síťové infrastruktury, která zajišťuje konektivitu pro IoT zařízení.

3.2.1 Světelné okruhy

V původním návrhu bylo uvažováno, že by se jako svítidla použily LED pásky a centrální zdroj o 12 V. Tímto řešením ale vzniká kritický bod v podobě zdroje. Pokud by zdroj selhal, uživatel nebude schopen rozsvítit jediné světlo v domácnosti. Z toho důvodu je lepší konvertovat napětí až u jednotlivých LED pásků, tzn. za vypínačem.

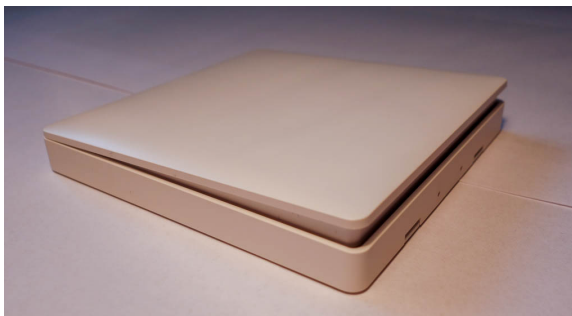
Dalším důvodem k nepoužití centrálního 12 V zdroje je absence zařízení na trhu. Konkrétně vestavných IoT spínačů, které by pracovaly na napětí 12 V, zároveň taková napětí spínaly a zároveň měly možnost napojení do sítě pomocí WiFi.

Alternativním spínačem by mohl být i bezdrátový spínač od společnosti Xiaomi, ukázaný na obrázku 3.1. Tento spínač využívá pro komunikaci technologii *ZigBee*, založenou na IEEE 802.15.4, která pracuje v pásmech 868 MHz, 915 MHz a 2,4 GHz [8] a je možné ho připojit přímo do platformy *Home Assistant* například pomocí modulu *zigbee2mqtt*³ nebo

²Odkaz na web frameworku *esphomelib*: <https://esphomelib.com/index.html>

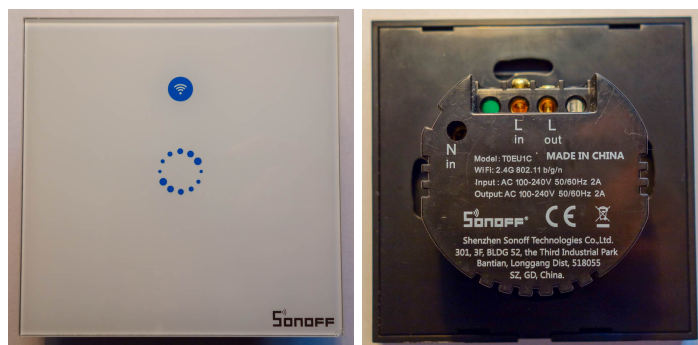
³Odkaz na *zigbee2mqtt*: <https://github.com/danielwelch/hassio-zigbee2mqtt>

skrze *Xiaomi Gateway*. Nevýhodou tohoto řešení je možná nefunkčnost světel při výpadku datové sítě nebo vybití baterie ve spínači.



Obrázek 3.1: Spínač *Xiaomi*. Bezdrátový spínač s baterií pracující na principu *ZigBee*.

Aktuální návrh ovládání osvětlení klade důraz především na nezávislost na datové síti. Uvažuje se se zapojením světel, kde k instalační krabici spínače jsou přivedeny minimálně vodiče L a N. Potom je možné využít spínač *Sonoff* ukázaný na obrázku 3.2. Tento spínač má tu výhodu, že je napájen přímo síťovým napětím v rozsahu 100–240 VAC a stejné napětí je po sepnutí na výstupu. Výhodou tohoto řešení je, že ačkoliv má spínač WiFi a lze ho připojit do chytré domácnosti, tak dokáže fungovat i sám o sobě. Spínání spínače je tedy závislé pouze na něm samotném a při výpadku datové sítě bude domácí osvětlení nadále fungovat.



(a) Přední část.

(b) Zadní část.

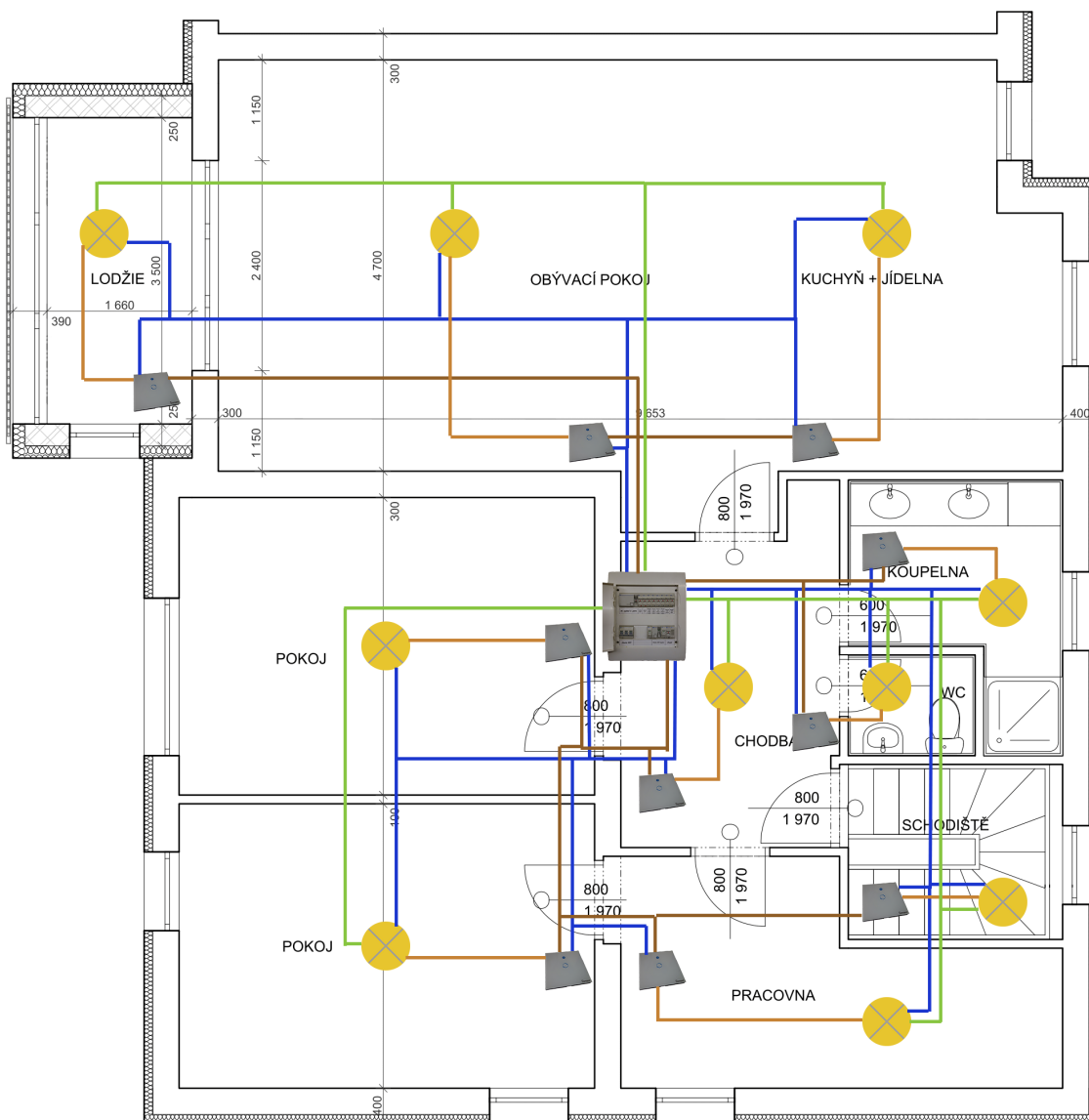
Obrázek 3.2: Spínač *Sonoff Touch*. Dotykový spínač napětí 230 VAC, který lze používat i bez připojení k datové síti.

Při použití LED pásek je tedy v tomto řešení nutné napětí konvertovat až za vypínačem, ale je zajištěna jak nezávislost na datové síti, tak na funkčnosti centrálního zdroje (potřebu centrálního zdroje toto řešení eliminuje).

Schéma infrastruktury osvětlení

Na obrázku 3.3 je provedený návrh osvětlení. Z rozvodné skříňky vedou ke spínači *Sonoff* vodiče L a N, kde vodič L je znázorněn hnědou barvou a vodič N modrou. Od spínače ke světlu je fázový vodič L znázorněn mírně odlišnou barvou pro větší přehlednost návrhu.

Přímo ke světlu vede ještě vodič PE znázorněný zelenou barvou. Všechny vodiče pro světelné okruhy jsou v měděném provedení, průřezu 1,5 mm². Veškeré osvětlení je poté jištěno jedním 10 A jističem.



Obrázek 3.3: Půdorys domu se zakreslenými světelnými okruhy. Na obrázku je několik spínačů *Sonoff Touch* připojených tak, aby spínaly světla. Hnědá barva značí vodič L, modrá vodič N a zelená PE. Všechna světla jsou jištěna jedním 10 A jističem.⁴

3.2.2 Zásuvkové okruhy

V rámci návrhu domácnosti jsou řešeny i zásuvkové okruhy, a to včetně jejich spínání po datové síti. Ke spínání se využije zařízení *Sonoff Pow*, které obsahuje relé pro spínání

⁴Jedná se o upravený obrázek půdorysu podlaží domu od Ing. Lukáše Doležala, ze kterého byly vymazány některé míry a byl do něj zanesen návrh infrastruktury.

elektrických okruhů. Toto zařízení navíc umožňuje měřit elektrickou energii. Díky tomu je možné získávat naměřenou spotřebu připojených zařízení a vytvářet statistiky.

Pro spínání jednotlivých zásuvek je možné využít chytrou zásuvku *BlitzWolf* (obrázek 3.4). Pomocí této zásuvky je možné rovněž spínat a měřit spotřebu a dá se využít ke spínání spotřebičů s příkonem až 2300 W.



(a) Přední část.

(b) Zadní část.

Obrázek 3.4: Chytrá zásuvka *BlitzWolf BW-SHP6*. Zásuvka je využitelná ke spínání spotřebičů s příkonem až 2300 W.

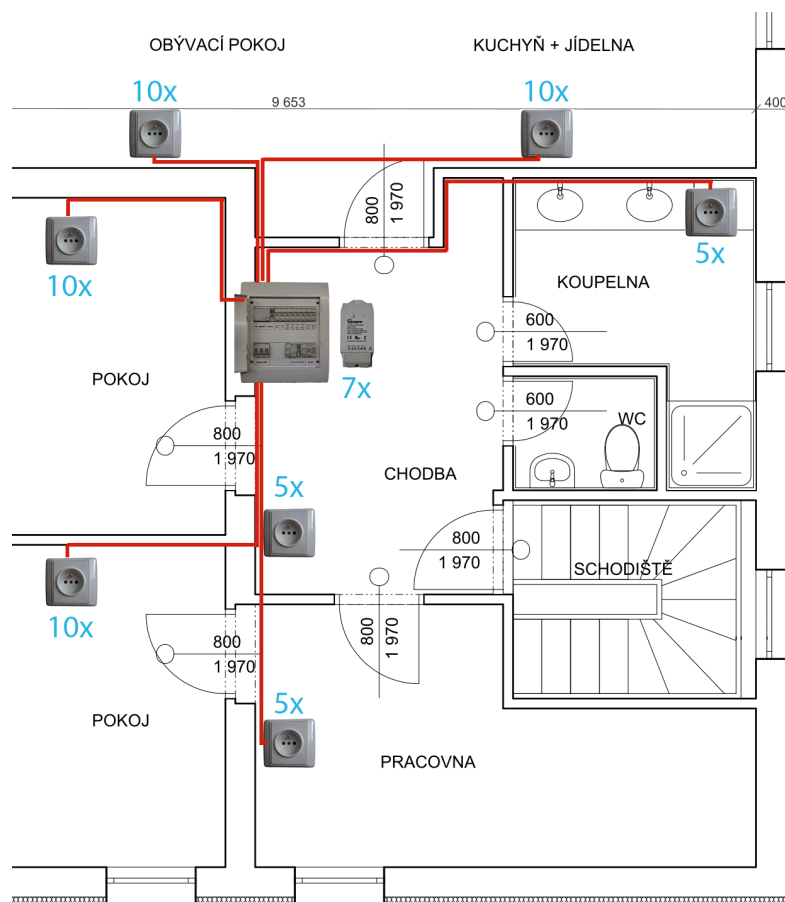
Schéma infrastruktury zásuvkových okruhů

Na obrázku přibližného půdorysu (obrázek 3.5) je znázorněna zásuvková infrastruktura. V tomto případě je trojice vodičů L, N a PE znázorněna pomocí jedné červené čáry. Každá ze zakreslených zásuvek dále náleží do samostatného okruhu odděleného jističem, přičemž je u každé z nich vyjádřen počet takovýchto zásuvek v daném okruhu.

Pro měření a spínání bude za každým zásuvkovým jističem umístěno zařízení *Sonoff Pow*. V tomto zařízení bude přehrán výchozí SW. Z důvodu větší volnosti v nastavování zařízení bude jako alternativní firmware zvolen open-source firmware *Sonoff-Tasmota*⁵. Vzhledem k aplikaci by měl tento firmware při připojení zařízení na napětí udržovat stejný stav, jaký byl před odpojením. To z důvodu standardního chování jističe, který mu je předřazen.

Stejně tak jako je *Sonoff Pow* dimenzovaný na proud 16 A, tak i pro jištění jednotlivých okruhů se využije 16 A jistič. Pro vedení se použijí vodiče průřezu 2,5 mm² v měděném provedení.

⁵ *Sonoff-Tasmota*: <https://github.com/arendst/Sonoff-Tasmota>



Obrázek 3.5: Půdorys domu se zakreslenými zásuvkovými okruhy. Na obrázku je zakresleno několik zásuvkových okruhů, kde počet zásuvek v okruhu je vyjádřen číslem vedle zásuvky. Každému zásuvkovému okruhu je předřazen spínač *Sonoff Pow* a jemu 16 A jistič.⁶

3.2.3 Síťová infrastruktura pro IoT prvky

Zde navrhovaná síťová infrastruktura počítá s tím, že veškeré komponenty domácnosti budou fungovat i při výpadku datové sítě a funkčnost každého zařízení bude závislá pouze na napájení a sama na sobě. Tímto se eliminuje potřeba sofistikované síťové infrastruktury a návrh sítě může být i méně finančně náročný.

Internetová konektivita bude zajištěna pomocí jedné přípojky. Služby pro potřebu sítě pak bude spravovat jedno centrální zařízení (server). Tímto zařízením bude nízko příkonový server *Alix*⁷. Desky tohoto druhu jsou dostatečně výkonné a mají udávanou maximální spotřebu 12 W při napětí 12 V. Ze zařízení se využijí dva sloty na Ethernet, kde rozhraní jednoho z nich se bude starat o připojení k vnější síti a druhé rozhraní se bude starat o vnitřní síť. V tomto zařízení bude dále jeden disk s možností zálohy.

Na vnitřní síťové rozhraní serveru bude připojen switch, ze kterého povedou kabely do Ethernetových zásuvek po domě. Na vhodných místech budou rozmístěny nejméně dvě WiFi

⁶Jedná se o upravený obrázek půdorysu podlaží domu od Ing. Lukáše Doležala, ze kterého byly vymazány některé míry a byl do něj zanesen návrh infrastruktury.

⁷Desky *Alix*: https://www.discomp.cz/bezdratove-prvky-pc-engines-alix-alix-apu-desky_c14506488.html?action=setcur&curid=0

zařízení v režimu přístupového bodu (AP), připojené do těchto zásuvek, tak aby se dosáhlo pokrytí celé domácnosti oběma těmito zařízeními. Oba AP pak budou mít stejné SSID a také heslo. Tím bude klientským zařízením zajištěna WiFi konektivita i při výpadku jednoho z AP.

Na serveru bude nainstalován operační systém *Debian* v 64 bitové verzi. Mezi službami běžícími na serveru bude NAT překladač a firewall s mj. povolenými porty pro přístup k webové aplikaci ovládací platformy ze sítě internet. Pro potřeby vnější sítě bude na serveru nakonfigurována služba DNS a dále DHCP pro snadné připojení zařízení do sítě.

Ačkoliv komponenty chytré domácnosti budou fungovat samy o sobě, na centrálním bodě v podobě serveru může dojít k výpadku. V případě, že by došlo k poškození dat na disku nebo fyzickému poškození disku, muselo by se vše konfigurovat od začátku. Aby se předešlo této situaci, tak návrh počítá se zálohou snímku disku na cloudové úložiště (např. *Google Drive*).

3.3 Vlastní IoT zařízení

Pro potřeby specifických aplikací chytré domácnosti, jakou je otevření/zavření střešního okna na základě změny klimatických podmínek nebo ovládání garážových vrat, jsou tato zařízení v této části návrhu probírána podrobněji. Návrh vlastních zařízení dále popisuje komponentu pro měření teploty. Všechna tři zmíněná zařízení jsou navrhována s deskou *WEMOS*, jako řídicím členem.

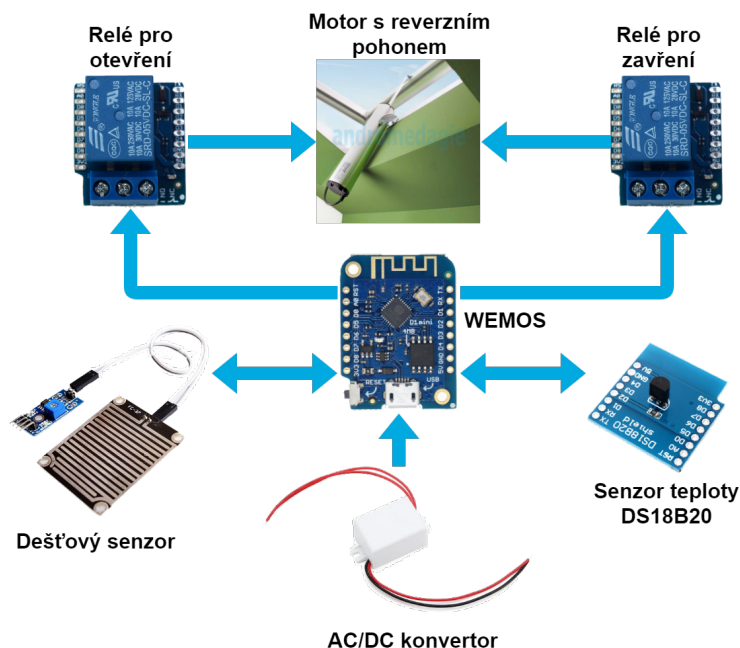
3.3.1 Automatizace střešního okna

Tato komponenta je navrhována z důvodu potřeby větrání v půdních prostorách, kam je horší fyzický přístup. Ovládání okna je uživateli umožněno skrze aplikaci v jeho chytrém telefonu. Uživatel má možnost okno otevřít nebo zavřít a vidět, v jakém stavu se okno aktuálně nachází. Druhou možností ovládání okna je plně automatický režim, ve kterém je okno otevíráno a zavíráno na základě změny venkovních klimatických podmínek. Informace o klimatických podmínkách a jejich změně je získávána pomocí venkovních senzorů teploty a deště.

Návrh

Vzhledem k nutnosti přivedení napětí 230 VAC kvůli napájení motoru, je pro napájení řídicího členu počítáno s převedením napětí z 230 VAC na 5 VDC pro řídicí člen. Jak je znázorněno na obrázku 3.6, o převod napětí z 230 V střídavého napětí na 5 V stejnosměrného napětí se stará AC/DC konvertor. Pro samotné otevírání okna se využije motor od společnosti *FAST*, konkrétně MFAST300H0B00.

Jako teplotní senzor bude použito čidlo DS18B20 ve vodotěsném obalu. Pro zjišťování přítomnosti srážek bude využit dešťový senzor a dvě relé pro spínání motoru pro pohyb tam a zpět. Jako řídicí člen je uvažována deska *WEMOS*. Souhrn veškerých součástek, modulů a popis jejich funkce v rámci komponenty se nachází v tabulce 3.1.



Obrázek 3.6: Blokové schéma ovládní elektrických oken. Deska WEMOS napájená z elektrické sítě, dva senzory pro snímání klimatických podmínek a dvě relé pro ovládní motoru, který manipuluje s oknem.

Součástka	Popis činnosti součástky
WEMOS	Řídící člen.
Motor FAST	Ovládní střešního okna.
AC/DC konvertor	Napájení desky WEMOS a teplotního a dešťového senzoru.
Relé 1	Pro sepnutí motoru.
Relé 2	Pro zpětný chod motoru.
DS18B20	Měření teploty.
Dešťový senzor	Zjišťování, zda prší/sněží.

Tabulka 3.1: Součástky pro komponentu automatizace střešního okna a popis jejich významu v rámci komponenty.

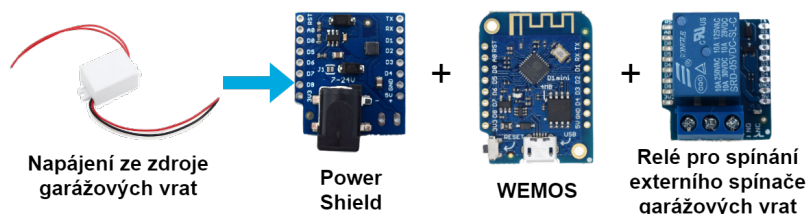
Funkce řídicího členu V tomto případě bude deska WEMOS sloužit pro komunikaci se serverem a pro ovládní dvou relé, které manipuluje s motorem. Bude měřit venkovní teplotu pomocí čidla DS18B20 a zjišťovat srážky pomocí dešťového senzoru. Na základě vyhodnocení získaných dat bude zavírat nebo otevírat okno.

3.3.2 Ovládní garážových vrat

Myšlenka ovládní vrat pomocí chytrého telefonu vychází z faktu existence pouze jednoho dálkového ovladače dodávaného ke garážovým vratům. To představuje problém, když potřebuje jeden ovladač sdílet více uživatelů. Navrhované řešení proto počítá s ovládním chytrými telefony, za využití stávajícího systému elektrického pohonu, rozšířeného o IoT prvek.

Návrh

Z obrázku 3.7 je patrné, že pro ovládání současných garážových vrat je v návrhu počítáno s deskou *WEMOS*. Ta bude napájena pomocí součástky nazývané Power Shield, která se jednoduše připojí na piny desky *WEMOS*. Důvodem k použití této součásti je existence 24 VDC napětí, se kterým pracuje stávající ovládací systém. Pro spínání bezpotenciálového kontaktu bude využito relé. Souhrn modulů a popis jejich funkce v rámci komponenty se nachází v tabulce 3.2.



Obrázek 3.7: Blokové schéma ovládání garážových vrat. Deska *WEMOS* napájena ze zdroje garážových vrat a relé pro spínání kontaktu externího spínače garážových vrat.

Součástka	Popis činnosti součástky
<i>WEMOS</i>	Řídící člen.
Power Shield	Napájení desky <i>WEMOS</i> .
Relé	Pro spínání bezpotenciálového kontaktu systému ovládání garážových vrat.

Tabulka 3.2: Součástky pro komponentu ovládání garážových vrat a popis jejich významu v rámci komponenty.

Funkce řídicího členu Deska *WEMOS* bude spínat externí kontakt systému pro manipulaci s garážovými vraty na základě dat přijatých přes síť. Při přijetí správných dat sepne relé po dobu 0,5 s, což způsobí změnu stavu garážových vrat. Nový stav (otevírání/zavírání/zastavení) garážových vrat potom závisí na předchozím stavu.

3.3.3 Měření teploty

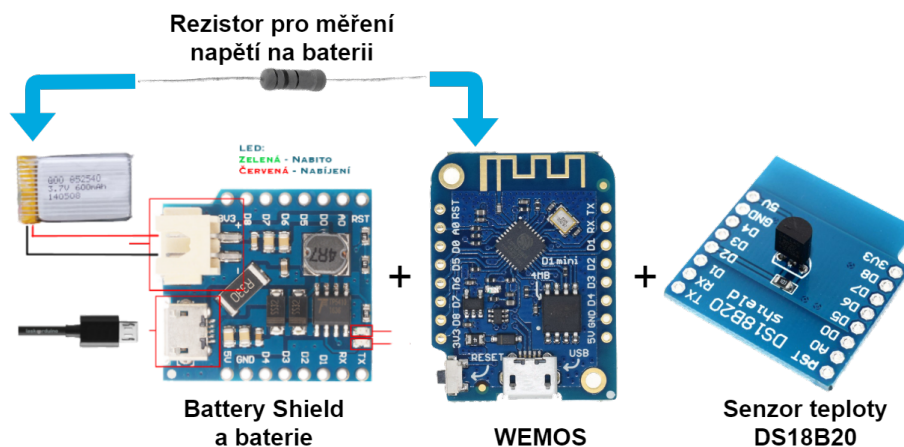
Další navrhovanou komponentou systému jsou senzory teploty, které budou měřit teplotu na několika místech domu. Obdobně jako v případě měření spotřeby, bude možné z naměřených hodnot vytvářet statistiky pro uživatele. Hodnoty měřené senzory bude potom uživatel moci v reálném čase vidět na displeji svého chytrého zařízení. Smysl existence senzorů v několika místnostech je například i pro možné budoucí napojení na regulaci vytápění, tato část ale není součástí vytvářeného návrhu.

Návrh

Na obrázku 3.8 lze vidět, že jako řídicí člen je opět počítáno s deskou *WEMOS*. Pro měření teploty se použije senzor DS18B20, který se dá zakoupit jako shield pro *WEMOS*. Z důvodu možnosti libovolného umístění senzorů a zároveň instalace do existujícího domu je výhodné mít senzory napájené z baterií. Pro napájení řídicího členu je tedy v návrhu počítáno

s napájecím štítem, který je možné připojit na desku *WEMOS*. K tomuto štítu se připojuje zmiňovaná baterie.

Nicméně, vzhledem k použitému způsobu napájení je vhodné, aby byl uživatel pohodlně informován o stavu baterie. Za tímto účelem je v návrhu počítáno i s rezistorem připojeným na analogový vstup (A/D převodník) desky *WEMOS*. Tento rezistor je na druhém konci připojen na kladný pól baterie. Souhrn veškerých součástí, modulů a popis jejich funkce v rámci komponenty je v tabulce 3.3.



Obrázek 3.8: Blokové schéma měřiče teploty. Deska *WEMOS* napájená z baterie, senzor teploty a rezistor, který slouží pro měření napětí na baterii, připojený na A/D převodník desky *WEMOS*.

Součástka	Popis činnosti součástky
<i>WEMOS</i>	Řídící člen.
Battery Shield	Pro napájení desky <i>WEMOS</i> a teplotního čidla.
Baterie	Zdroj pro napájení.
DS18B20	Měření teploty.
Rezistor	Pro měření napětí na baterii.

Tabulka 3.3: Součástky pro komponentu měření teploty a popis jejich významu v rámci komponenty.

Funkce řídicího členu V tomto případě bude deska *WEMOS* komunikovat se serverem, kterému bude odesílat data z teplotního čidla. Deska bude dále měřit napětí na baterii a naměřenou hodnotu bude také odesílat serveru.

3.4 Napájení a bezpečnost

Napájení komponent je v návrhu řešeno vícero způsoby. Tam kde to dává smysl tak je využito baterií, tam kde je dostupná elektrická síť, tak je využito jí. Protože veškerá zařízení budou pro komunikaci využívat *ESP* čip a standard *IEEE 802.11* (Wi-Fi), není třeba řešit přívod datových vodičů k zařízením.

Aby byly jednotlivá zařízení krytá, tak budou umístována do plastových krabic. V případě komponenty pro ovládání garážových vrat a měření teploty, kde se pracuje se stejnosměrným napětím do 24 V, nejsou na vlastnosti krabiček kladeny žádné speciální požadavky. V případě ovládání střešního okna, kde se pracuje mimo malého bezpečného napětí i s napětím síťovým o hodnotě 230 VAC, bude použita samozhášivá instalační krabice. V případě, že by došlo k hoření, oheň by se poté neměl šířit mimo prostor krabice.

3.5 Shrnutí navrženého řešení

V této kapitole byl vytvářen návrh infrastruktury pro osvětlení a zásuvkové okruhy, kde jednotlivá světla a zásuvkové okruhy jsou spínány pomocí IoT prvků. Dále byla navržena síťová infrastruktura s využitím nízko příkonového serveru. Na tomto serveru budou nastaveny služby pro správu sítě a bude zde možnost zálohovat disk mimo lokální síť. WiFi zařízení v režimu AP s nastavenými stejnými přístupovými údaji zajišťují konektivitu bezdrátovým zařízením.

Pro správu IoT zařízení domácnosti je v návrhu uvažováno s platformou *Hass.io*, která je spouštěná na lokálním serveru. Programování, respektive popis chování komponent a propojení s platformou poté návrh řeší pomocí *esphome*. Funkce jednotlivých důležitých komponent, jakými jsou spínače světel, spínače elektrických okruhů a automatizace střešního okna, nebudou závislé na ovládací platformě ani na datové síti.

Kapitola 4

Implementace a testování

Řešení chytré domácnosti bylo implementováno dle návrhu vytvořeném v kapitole 3. V rámci implementace bylo třeba vyřešit několik konstrukčních problémů a zvolit optimální součástky pro vytvoření komponent založených na deskách *WEMOS*. Implementace se ve většině situací drží návrhu, nicméně v případě ovládání garážových vrat a ovládání okna byly přidány magnetické senzory pro signalizaci stavu otevřeno/zavřeno. Dále bylo zkonstruováno upravené čidlo *DS18B20* pro měření teploty. Platforma *Hass.io* je poté místo fyzického serveru nasazena na virtuálním počítači v rámci virtualizační platformy *VirtualBox* a jednotlivá zařízení se připojují na hostující počítač.

4.1 Společná část implementace vlastních komponent

Konstrukce HW Při konstruování HW byly používány především široce prodávané součásti sestavené z menších celků. Byly využity shieldy pro desku *WEMOS* a jiné sestavené DPS. Celé řešení bylo konstruováno tak, aby ho následně bylo možné nasadit v rámci stávající domácnosti.

YAML soubory Pro programování IoT zařízení bylo využito open-source doplňku do platformy *Hassio*, *ESPHome*. Popis chování jednotlivých komponent je prováděn pomocí YAML souborů. Tyto YAML soubory obsahují název zařízení, informace o čipu a desce, údaje pro připojení k WiFi, povolení logování, přístupové údaje pro komunikaci s platformou a dále popis chování specifický pro danou komponentu. YAML soubory jsou následně doplňkem (automatizovaně) překládány do C++, dále do binárního kódu a skrze OTA přenášeny a nahrávány do paměti dané komponenty. Rozdílem je pouze první programování zařízení, kde je využíváno aplikace *ESPEasy Flasher* pro nahrání prvotního kódu do paměti desky *WEMOS* za pomoci microUSB.

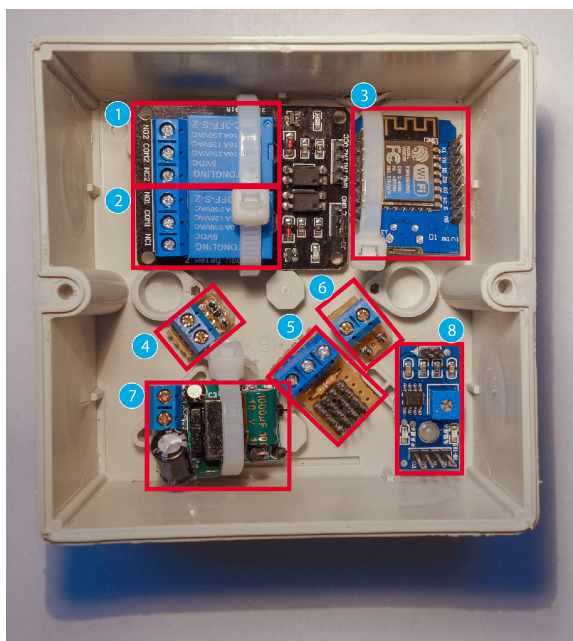
4.2 Automatizace střešního okna

Pro konstrukci této komponenty chytré domácnosti byly využity součásti vyobrazené na obrázku 4.1. Jedná se o obrázek komponenty před propojením jednotlivých součástí.

Pro spínání motoru je využito dvou relé (čísla 1 a 2). Číslo 3 je deska *WEMOS*, ve které je řídicí kód. Pod číslem 4 je DPS (deska plošných spojů) se svorkovnicí, na kterou se připojuje magnetický senzor. Číslo 5 je DPS pro rozvod napájení mezi jednotlivé součásti, dále je zde i pull-up rezistor kvůli připojení čidla *DS18B20* pomocí I²C sběrnice. Čidlo se

připojuje na svorky této DPS. Pod číslem 6 je DPS se svorkovnicí pro připojení dešťového senzoru. Vlevo dole pod číslem 7 je vidět převodník napětí z 230 VAC na 5 VDC pro napájení jednotlivých součástí. Tímto je zajištěna potřeba pouze jednoho přívodního kabelu namísto dvou pro napájení elektroniky napětím 5 VDC a motoru napětím 230 VAC. Pod číslem 8 vpravo dole se poté nachází kontrolér pro dešťový senzor.

Komponenta byla konstruována do instalační krabice *Kopos KSK 100_KA* ze samozhášivého materiálu¹. U krabice výrobce udává testování žhavou smyčkou o teplotě 850 °C, povolené jmenovité napětí až 400 V, proud 16 A a stupeň krytí odpovídá IP 66 [11]. Uchyacení jednotlivých komponent do krabice je zajištěno pomocí plastových stahovacích pásek, které byly ke krabici připevněny pomocí tavné pistole.



Obrázek 4.1: Krabice se součástmi komponenty pro ovládání střešního okna. 1) Otevírací relé. 2) Zavírací relé. 3) Deska *WEMOS*. 4) DPS pro připojení magnetického senzoru. 5) DPS pro rozvod napájení a připojení teplotního čidla. 6) DPS pro připojení dešťového senzoru. 7) Konvertor napětí 230 VAC/5 VDC. 8) Kontrolér dešťového senzoru.

Kompletní komponenta se zapojenými vodiči je na obrázku 4.2. Zde už jsou vidět i pomocí kabelů připojené jednotlivé senzory. Je zde dešťový senzor, který byl připojen ke kabelu a připojen na svorkovnici uvnitř krabice. Šedý kabel vede od svorkovnice v krabici a na druhém konci je připojen k magnetickému spínači, tento je použit ke zjišťování stavu okna (otevřeno/zavřeno).

Pro měření teploty byl zakoupen teplotní senzor DS18B20 ve vodotěsném pouzdře. Tento senzor je rovněž připojen na svorky v instalační krabici, kde na DPS, na kterou je připojen je navíc pull-up rezistor, z důvodu existence I²C sběrnice.

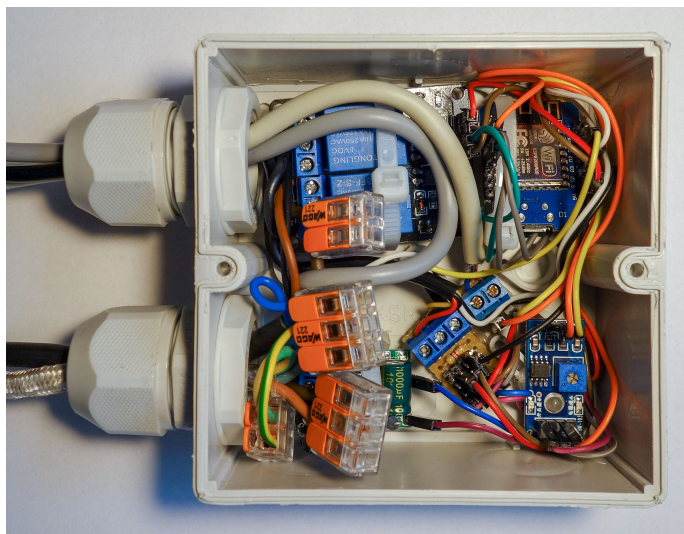
Komponenta je napájena kabelem, který se připojuje do běžné elektrické sítě na napětí 230 VAC. Ve finální implementaci se však počítá s napojením přímo na svorkovnici. Pomocí tohoto přívodu je napájen i motor.

¹Krabice Kopos KSK 100_KA: <https://www.kopos.cz/cs/produkt/2884-krabice-s-ip-krytim>



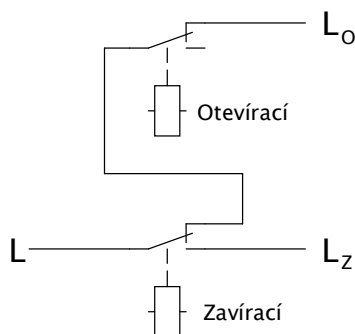
Obrázek 4.2: Kompletní komponenta pro ovládání střešního okna. Instalační krabice, dešťový senzor, magnetický senzor/spínač, teplotní čidlo DS18B20, přívodní napájecí kabel a motor pro manipulaci s oknem.

Na obrázku 4.3 je detail instalační krabice. Pro pospojování a napájení součástí pomocí 5 VDC je využito vodičů, které se připojují na piny součástí. Pro rozvod napětí 230 VAC se využívají *WAGO* svorky. Vodiče k externím senzorům, napájecímu kabelu a kabelu pro připojení motoru jsou poté vyvedeny z instalační krabice pomocí dvou vývodů.



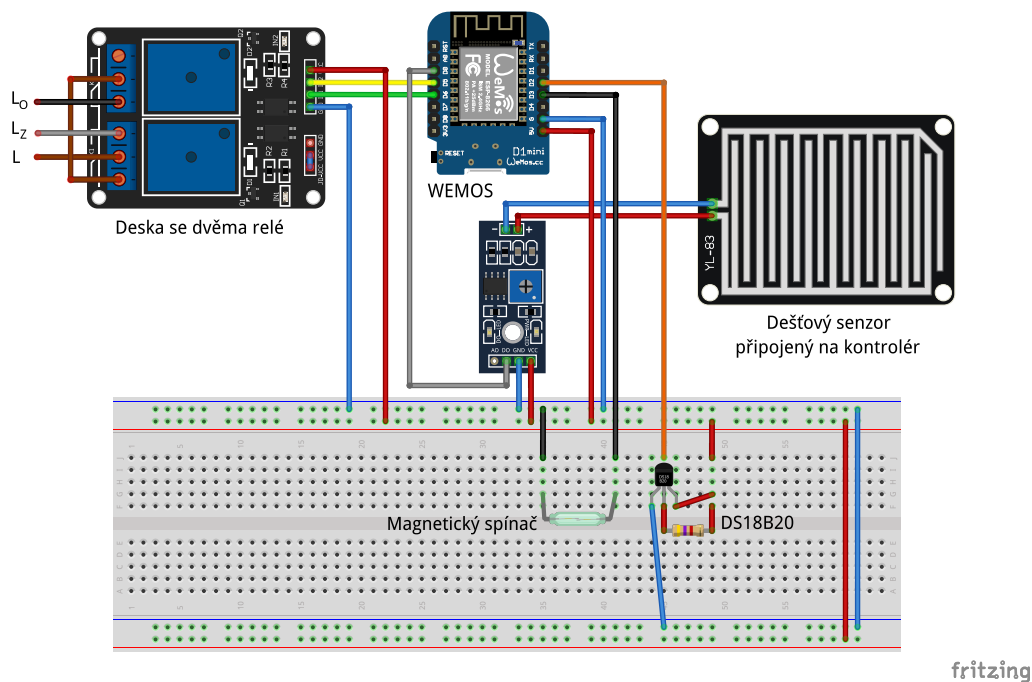
Obrázek 4.3: Detail instalační krabice komponenty pro ovládání střešního okna. Pospojované vnitřní součásti a připojené externí části.

Pospojování relé Další věcí, kterou bylo nutné zajistit na HW úrovni je způsob připojení motoru na napájení. K motoru vedou čtyři vodiče. Těmito vodiči jsou vodič PE, N a otevírací a zavírací fáze. Aby se nemohlo stát, že bude například při poruše na řídicím členu na obě tyto fáze přivedeno napětí současně, je toto zabezpečeno pomocí vhodného pospojování spínacích a rozpínacích kontaktů dvou relé, jak je patrné ze schématu na obrázku 4.4. Na vstup druhého relé je přiveden vodič L (vstupní fáze), skrze rozpínací kontakt je poté propojeno se vstupem prvního relé. Fáze pro otevření (vysunutí tyče) je označená L_0 a fáze pro zavření (zasunutí tyče) je označená L_Z .



Obrázek 4.4: Schéma zapojení relé nakreslené v programu ProfiCAD. Využívá se pospojování spínacích a rozpínacích kontaktů obou relé, tak aby bylo zajištěno přivedení napětí na motor pouze v jednom směru. L_0 značí otevírací fázi a L_Z zavírací fázi motoru.

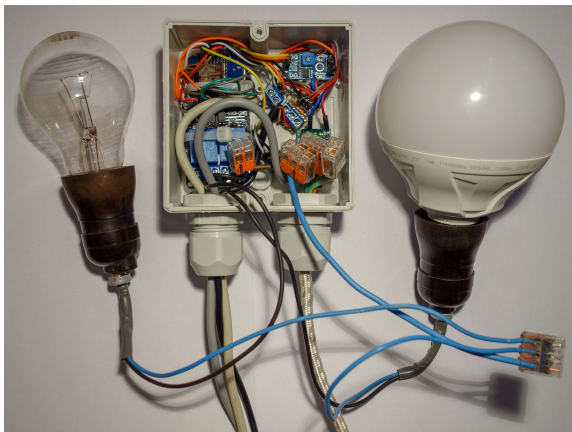
Schéma celé komponenty Na obrázku 4.5 je schéma zapojení jednotlivých komponent včetně vzájemného pospojování relé, které je podrobněji popsáno výše. Zapojení na schématu odpovídá konečnému zapojení součástí v rámci hotové komponenty.



Obrázek 4.5: Schéma zapojení komponenty ovládání okna. Jedná se o pospojování jednotlivých součástí, tak jak jsou zapojeny ve finální podobě zařízení.

Testování

Aby byla ověřena funkčnost komponenty, tak byly místo motoru připojeny dvě žárovky. Levá žárovka měla sloužit jako vizuální podoba pohybu tyče motoru směrem nahoru (otevření okna) a druhá měla posloužit jako vizuální znázornění pohybu tyče směrem dolů. Testovalo se především, aby v žádném případě nesvítily žárovky současně a ověřovala se funkčnost celého zapojení a SW.



Obrázek 4.6: Zapojení při jedné z forem testování. Levá žárovka vizualizuje pohyb nahoru, pravá pohyb dolů.

Implementovaný kód ve formátu YAML

Příloha B obsahuje text ve formátu YAML s popisem chování celé komponenty. Tento kód umožňuje v platformě zobrazit teplotu, zda prší a stav okna. Dále je možné skrze platformu zapnout automatizaci pro automatickou manipulaci s oknem na základě venkovních klimatických podmínek (v souboru `configuration.yaml` musí být kód pro definici přepínače, který je rovněž uveden v kódu v příloze B).

Automatizace otevírá okno, pokud venku neprší a teplota je větší nebo rovna 27 °C. Tato automatizace je nezávislá na platformě, protože je kód pro její vykonávání uložen přímo v zařízení pro ovládání okna. Bude tedy fungovat i v případě výpadku ovládací platformy.

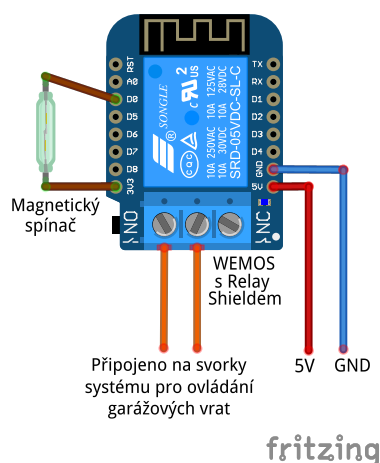
4.3 Ovládání garážových vrat

Na obrázku 4.7 je vidět sestavená komponenta pro ovládání garážových vrat. Tato komponenta sestává z desky *WEMOS*, na které je připojen shield s relé. Spínací kontakt relé spíná externí kontakt systému garážových vrat. Pro napájení se využívá adaptéru s výstupním napětím 5 VDC, který je připojen na desku *WEMOS*. Součástí komponenty je oproti návrhu navíc i magnetický senzor pro zjišťování stavu garážových vrat, tedy zavřeno/otevřeno.



Obrázek 4.7: Zkonstruované zařízení pro ovládání garážových vrat. Deska *WEMOS* a na ní nasazený spínač v podobě relé. Dále DPS připojená na piny desky *WEMOS*, kde na plošném spoji je vidět připájená svorkovnice a do ní připojený magnetický senzor.

Schéma celé komponenty Na obrázku 4.8 je schéma finální podoby zapojení celé komponenty. Magnetický senzor/spínač je připojen na pin D0, na kterém je povolený pull-down rezistor.



Obrázek 4.8: Schéma zapojení komponenty ovládání garážových vrat. Jedná se o pospojování jednotlivých součástí, tak jak jsou zapojeny ve finální podobě zařízení.

Testování

Při testování bylo zjištěno, že ačkoliv je v současném systému existence napětí 24 VDC, tak napájení pomocí napájecího shieldu nebylo vhodné, protože zařízení zřejmě kvůli nedostatku proudu nefungovalo správně. Z toho důvodu došlo ke změně, napájecí shield byl odstraněn a nyní se napájí pomocí adaptéru, který dodává napětí 5 VDC.



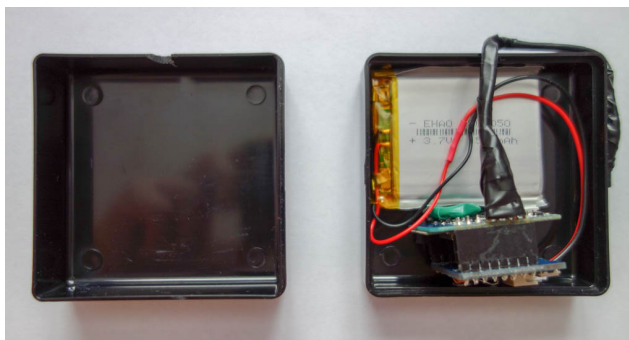
Obrázek 4.9: Ukázka z testování komponenty. Na obrázku je vidět komponenta připojená na svorky současného systému. Vyobrazená komponenta obsahuje zatím ještě Power Shield.

Implementovaný kód ve formátu YAML

Implementovaný kód pro tuto komponentu je rovněž uveden v příloze B. Úkolem je sepnutí relé, které je nastaveno na dobu 0,5 s. Tato doba je dostatečná na to, aby systém ovládání garážových vrat zaznamenal tuto změnu a změnil svůj vnější stav, tzn. otevřel nebo zavřel garážová vrata nebo zastavil jejich pohyb.

4.4 Měření teploty

Obrázek 4.10 znázorňuje komponentu pro měření teploty. Spodní díl komponenty je tvořen deskou *WEMOS*, na kterou je připojen Battery Shield. Na Battery Shield byl připájen rezistor pro měření napětí na baterii. Tento rezistor má jeden vývod připájený na kladný pól samičího konektoru baterie a druhý vývod na pin A0, oba vývody rezistoru jsou připájeny na Battery Shieldu. Do konektoru Battery Shieldu je dále připojena baterie o kapacitě 1500 mAh.



Obrázek 4.10: Zkonstruované zařízení pro měření teploty. Deska *WEMOS* a na ni připojený Battery Shield s měřicím rezistorem a do něj připojenou baterií. Dále teplotní čidlo DS18B20 a jeho vodiče připájené na desku *WEMOS*.

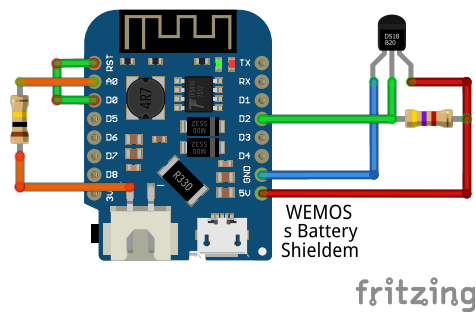
Jak je vidět z obrázku 4.11, vzhledem k tomu, že senzor je připojován k MCU pomocí *1-Wire* sběrnice, bylo nutné k senzoru ještě přidat 4,7 k Ω pull up rezistor. Tento rezistor je připájen přímo na straně čidla mezi kladný pól zdroje a pin 2 (datový pin) čidla. Vodiče byly následně řádně zaizolovány pomocí smršťovacích bužírek a elektroizolační pásky.



Obrázek 4.11: Upravené teplotní čidlo DS18B20. DS18B20, připájený 4,7 k Ω pull-up rezistor mezi datový pin 2 a kladný pól napájení, dále tři vodiče pro připojení k desce *WEMOS*.

Celá komponenta byla následně umístěna do krabičky. Na vlastnosti krabičky zde nebyly kladeny žádné speciální požadavky z pohledu hořlavosti. Teplotní čidlo bylo vyvedeno mimo krabičku.

Schéma celé komponenty Na obrázku 4.12 je schéma finální podoby zapojení celé komponenty (bez připojené baterie). Z důvodu SW uspání procesoru je propojen vodičem pin D0 a RST. Rezistor s odporem 100 k Ω je připojen mezi kladný pól baterie a pin A0. Na pin D2 je poté připojeno teplotní čidlo DS18B20, ke kterému je připojen 4,7 k Ω pull up rezistor.



Obrázek 4.12: Schéma zapojení komponenty pro měření teploty. Jedná se o pospojování jednotlivých součástí, tak jak jsou zapojeny ve finální podobě zařízení.

Testování

Oproti návrhu došlo ke změně v připojení teplotního čidla. V původním návrhu měl být použit shield pro *WEMOS* s čidlem *DS18B20* a tento shield měl být připojen navrch Battery Shieldu. Při prvotním testování bylo ale zjištěno, že teplota vznikající provozem mikrokontroleru nezanedbatelně ovlivňuje teplotu svého okolí, a tedy i shield s teplotním čidlem. Z toho důvodu byl senzor vyveden pomocí tří cca 10 cm dlouhých vodičů dál od čipu. Délka vodičů byla volena tak, aby dostačovala k vyvedení čidla mimo konstrukční krabičku.

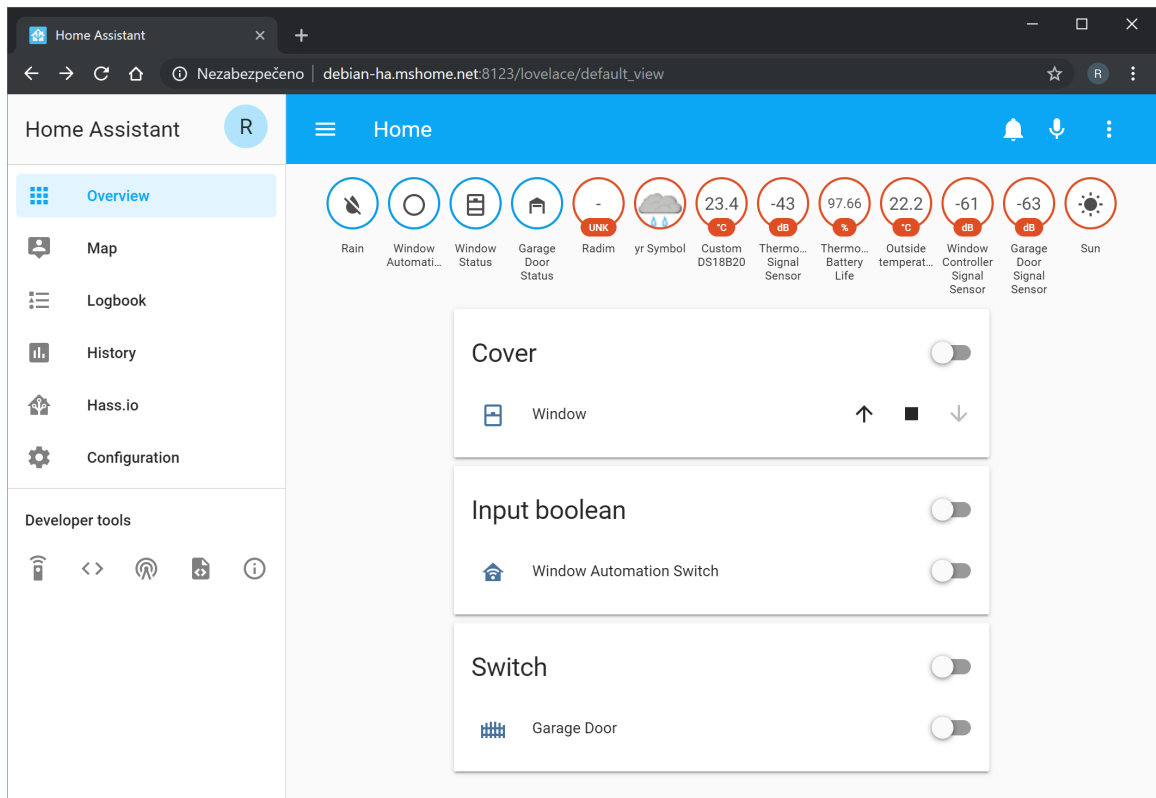
Implementovaný kód ve formátu YAML

Implementovaný kód pro tuto komponentu je opět v příloze **B**. Úkolem je měření teploty po nastaveném intervalu. Kód dále periodicky uspává komponentu. Doba běhu je nastavena na 75 s z důvodu připojení platformy *Home Assistant*, která se pokouší připojit jednou za 60 s. Doba uspaní je nastavena na krátký časový interval, z důvodu lepší ukázky řešení (muselo by se čekat na hodnoty). Součástí je dále kód pro zjištění napětí na baterii a jeho přepočtení na procenta nabití.

4.5 Integrace komponent do platformy

Pro integraci komponent je potřeba počítat se spuštěnou platformou, připojený ke stejné síti jako jednotlivé komponenty. Díky automatickému zjišťování *ESPHome* zařízení připojených v síti lze poté tato zařízení do platformy přidat skrze *Nastavení* → *Integrace* v platformě *Hass.io*. Pokud se zařízení nevyhledají automaticky, tak je lze přidat do platformy pomocí jejich doménového jména, sestávajícího z názvu zařízení určeném v YAML souboru v doplňku *ESPHome* a domény lokální sítě (např. *thermometer.local*). Přidaná zařízení je poté možné zobrazit na hlavní stránce platformy. Podrobnější postup konfigurace je popsán na přiloženém médiu.

Na obrázku **4.13** jsou vidět tři vlastní komponenty po integraci do platformy. Skrze zobrazenou obrazovku je lze ovládat a sledovat hodnoty získané ze senzorů. Je zde i přepínač vytvořený za účelem možnosti zapnutí/vypnutí automatizace.



Obrázek 4.13: Vlastní zařízení integrovaná do platformy. Pomocí této obrazovky lze provádět ovládání komponent a sledování dat ze senzorů.

Kapitola 5

Zhodnocení vlastního řešení a srovnání s komerčními řešeními

V této kapitole je vlastní řešení rozebíráno z pohledu finanční náročnosti. Jsou zde vypsány nejdůležitější součásti jednotlivých vlastních komponent a jejich orientační ceny. Dále jsou zde orientační ceny za ostatní použité komponenty. Ve shrnutí je pak proveden celkový cenový odhad za navrženou chytrou domácnost. Odhad ceny jednotlivých vlastních komponent je proveden na základě skutečně použitých součástí a materiálu při konstrukci.

Text této kapitoly dále srovnává navržené řešení z kapitoly 3 s existujícími komerčními řešeními z kapitoly 2. Je zde probíráno, jak by bylo možné navrženou chytrou domácnost řešit za pomoci vybraných komerčních řešení. Následně kapitola obsahuje shrnutí srovnání, možné rozšíření a závěrečné zamyšlení s využitím poznatků získaných při tvorbě vlastního řešení.

5.1 Celková finanční náročnost

Součástí zhodnocení navrženého řešení je i jeho finanční náročnost, to povede k lepšímu přehledu o tom, zda se vyplatí stavět chytrou domácnost založenou na open-source prvcích. Dále lze díky tomu provést srovnání s komerčním řešením založené na dalším kritériu, kterým je cena.

Automatizace střešního okna

V tabulce 5.1 jsou uvedeny ceny za jednotlivé součásti. Nejsou započítány například ceny za vývodky, svorkovnice, kabely, propojovací vodiče, DPS, lámací lišty. K práci byly dále potřeba nástroje jako pájecí stanice, multimetr, cín, tavná pistole, stahovací pásky apod. Dále se nepočítá s cenou za dopravu. Uvedený výčet je pouze orientačním výčtem s cenami za jednotlivé nejdůležitější součásti. Celková cena s dopravou a započítáním ostatního spotřebního materiálu potřebného ke konstrukci může být odhadem do **600 Kč** za ovládací zařízení. Cena použitého motoru MFAST300H0B00 pro ovládání okna byla při koupi v roce 2014 asi 4000 Kč bez dopravy.

Součást	Obchod	Orientační cena
Klon WEMOS D1 mini	<i>AliExpress</i>	45 Kč
Deska o dvou relé	<i>AliExpress</i>	25 Kč
Zdroj 5 V 700 mA	<i>AliExpress</i>	16 Kč
Dešťový senzor a kontrolér	<i>AliExpress</i>	13 Kč
Vodotěsné DS18B20, 3 m	<i>AliExpress</i>	34 Kč
Magnetický senzor	<i>Banggood</i>	28 Kč
Krabice Kopus KSK 100_KA	<i>Elima</i>	72 Kč
Celkem		233 Kč (600 Kč)

Tabulka 5.1: Ceny za jednotlivé součásti komponenty pro ovládání střešního okna. Jedná se pouze o výčet nejdůležitějších položek. Celková výsledná cena za ovládací zařízení je odhadem do **600 Kč**.

Ovládání garážových vrat

Tabulka 5.2 zobrazuje ceny za jednotlivé součásti použité při konstrukci zařízení pro ovládání garážových vrat. Tabulka opět neukazuje ceny za vývodky, kabely, vodiče, DPS, svorku, lámací lištu a spotřební materiál. Pro konstrukci byl sice použit nevyužívaný zdroj od mobilního telefonu, ale vzhledem k jeho stále dostupnosti v internetových obchodech je jeho cena započítána do celkové ceny komponenty.

Součást	Obchod	Orientační cena
Klon WEMOS D1 mini	<i>AliExpress</i>	45 Kč
Relay shield	<i>AliExpress</i>	19 Kč
Magnetický senzor	<i>Banggood</i>	28 Kč
Sítový adaptér Samsung ETA0U10E	<i>CZC.cz</i>	89 Kč
Krabice KM55 ABS BLACK ST	<i>GM electronic</i>	84 Kč
Celkem		265 Kč (300 Kč)

Tabulka 5.2: Ceny za jednotlivé součásti komponenty pro ovládání garážových vrat. Jedná se pouze o výčet nejdůležitějších položek. Celková výsledná cena za ovládací zařízení je odhadem okolo **300 Kč**.

Měření teploty

V tabulce 5.3 lze nalézt ceny za jednotlivé nejdůležitější součásti potřebné k sestavení komponenty pro měření teploty. Znovu nejsou uvedeny ceny za spotřební materiál a cena za 4,7 k Ω rezistor.

Součást	Obchod	Orientační cena
Klon WEMOS D1 mini	<i>AliExpress</i>	45 Kč
Battery shield	<i>AliExpress</i>	18 Kč
Baterie 1500 mAh	<i>AliExpress</i>	67 Kč
DS18B20	<i>AliExpress</i>	15 Kč
Krabice U-MINI, ABS HF 380 černý	<i>GM electronic</i>	36 Kč
Celkem		181 Kč (200 Kč)

Tabulka 5.3: Ceny za jednotlivé součásti pro komponentu měření teploty. Jedná se pouze o výčet nejdůležitějších položek. Celková výsledná cena za ovládací zařízení je odhadem do **200 Kč**.

Shrnutí

V tabulce 5.4 jsou orientační ceny za ostatní komponenty použité v rámci navržené chytré domácnosti. Zařízení *Sonoff Touch* je v návrhu z kapitoly 3 použito celkem 10, výsledná cena za použitá zařízení je tedy asi **2870 Kč**. Zařízení *Sonoff Pow* je v návrhu použito celkem 7, výsledná cena za tato zařízení je tedy **1988 Kč**.

Další komponentou, která by zde měla být uvedena je server pro správu sítě a chytré domácnosti. V návrhu byl zvolen server *Alix*, který se prodává jako samostatná deska. Při zakoupení zdroje k tomuto počítači, SSD a montážní krabice, může být cena za tento server odhadem okolo **5000 Kč**.

Pro úplnost se předpokládá existence dvou WiFi přístupových bodů. Cena za jeden bude do 1000 Kč, za oba tedy **2000 Kč**.

Komponenta	Obchod	Orientační cena
Itead Sonoff Touch EU	<i>AliExpress</i>	287 Kč
Sonoff Pow	<i>AliExpress</i>	284 Kč
BlitzWolf BW-SHP6	<i>AliExpress</i>	313 Kč

Tabulka 5.4: Ceny za ostatní komponenty. Jedná se o orientační ceny za komponenty, se kterými se počítá v návrhu.

Pokud člověk vlastní potřebné vybavení, tak se dají tři vlastní komponenty postavit za částku asi **1100 Kč** bez započtené práce. V praxi, kdy člověk navrhuje nějakou komponentu od začátku je to ale více, protože experimentuje i s jinými součástkami, které by mohly být vhodné, ale pak se přijde na to, že z nějakého důvodu nevyhovují. Díky tomu se zvyšuje jak cena, tak čas, za který je výsledná komponenta dokončena.

Celková výsledná cena za zde navrženou chytrou domácnost je odhadem **13000 Kč**. Do této ceny není započítána například cena za Ethernetové kabely a cena práce.

5.2 Srovnání návrhu s komerčními řešeními

Vzhledem k existenci odlišných zařízení v komerčních řešeních, než jsou zařízení zvolené v rámci vlastního návrhu, je obtížnější provést srovnání a výsledná chytrá domácnost může vypadat zcela jinak. Pro podrobné srovnání by byl třeba ještě návrh chytré domácnosti pomocí srovnávaného komerčního systému. V ideálním případě by měl tento návrh být sestaven člověkem, který se v dané komerční sféře pohybuje. Poté by bylo možné provést

plnohodnotné srovnání komerčního a vlastního řešení. Vzhledem k tomu, že komerční systémy nebylo možné v rámci práce vyzkoušet, tak jsou srovnávány jen na základě textů dostupných z webových stránek jednotlivých systémů.

Loxone

V případě systému *Loxone* by bylo možné podobnou chytrou domácnost sestavit s pomocí bezdrátových prvků systému, tedy s využitím technologie *Loxone Air*. Jako řídicí jednotka by se použil *Loxone Miniserver*, ke kterému by se připojil *Air Base Extension* pro připojení bezdrátových zařízení. Pro spínání světelných okruhů by se mohl využít bateriový bezdrátový spínač *Loxone Touch Air*, kde spínání světel by prováděl přímo *Loxone Miniserver* pomocí výstupů relé. Spínač *Loxone Touch Air* obsahuje navíc senzor pro měření teploty a vlhkosti, který by mohl být ekvivalentem k vlastní komponentě pro měření teploty. Pro spínání spotřebičů by poté bylo možné využít chytrou zásuvku *Smart Socket Air*, která umožňuje měřit spotřebu elektrické energie i teplotu.

Dvě navržené komponenty pro ovládání okna a garážových vrat by se daly vyřešit každá s využitím jednoho zařízení *Nano IO Air*, které obsahuje mimo dvou 5 A relé i několik digitálních vstupů. Pro měření srážek se dá využít digitální dešťový senzor připojitelný do *Nano IO Air* a pro měření ostatních klimatických podmínek pak venkovní senzor teploty a vlhkosti. Tento senzor by se dal připojit do systému přímo na analogové vstupy zařízení *Loxone Miniserver*. Pro zjištění stavu okna nebo garážových vrat se pak dá využít drátový okenní kontakt, rovněž připojitelný do *Loxone Miniserver* nebo bezdrátový okenní kontakt.

Fibaro

V systému *Fibaro* by mohla být navržená chytrá domácnost řešena například následujícím způsobem. Jako řídicí jednotka by se využil *FIBARO Home Center*, který, stejně jako ostatní komponenty tohoto systému, používá technologii *Z-Wave*. Pro spínání několika okruhů je nabízen *HeatIt Z-Din Spínací modul*. Jako alternativu je možné využít *FIBARO Spínací modul* pro spínání jednoho nebo dvou výstupů. Pomocí druhého zařízení lze navíc měřit spotřebu. Jako ovládací spínač pro světla je poté nabízen například *NodOn Wall Switch*. Pro spínání spotřebičů dále tento systém poskytuje bezdrátovou chytrou zásuvku.

Pro řešení ovládání garážových vrat se dá využít rovněž *FIBARO Spínací modul* s jedním výstupem. Pro ovládání motoru střešního okna pak obdobné zařízení s dvěma výstupy. Pro připojení senzorů lze využít univerzální moduly. Pro zjišťování stavu otevřeno/zavřeno může být využit samostatný bezdrátový senzor *FIBARO Senzor na okna a dveře*.

Control4

Návrh ukázkové infrastruktury řešené pomocí tohoto systému byl již zobrazen na obrázku 2.6. Například pro spínání světel a zásuvkových okruhů se dají využít elektroinstalační relé moduly, dále pak spínače komunikující po sběrnici RS-485. Pro režii chytré domácnosti poté existuje řídicí jednotka.

Shrnutí

Komerční systémy jsou často uzavřená řešení, o kterých je obtížnější zjistit podrobnější informace pouze z webových stránek. Při pokusu o provedení obdobného návrhu, jako je vlastní návrh, za pomoci komerčních řešení ze sekce 2.4 lze říci, že pořizovací cena by

byla mnohem vyšší, než cena za navržené řešení za pomoci open-source prvků. Nicméně lze očekávat vyšší odladěnost těchto zařízení, protože jsou na rozdíl od vlastních zařízení prověřeny praxí.

5.3 Možná rozšíření

Mezi možná SW rozšíření by se mohlo zařadit například zvýšení počtu automatizací, kde se může pracovat se stavem uživatelského telefonu. Například při odchodu z domu (odpojení od WiFi) může automatizace v platformě zhasnout světla nebo odpojit určitý elektrický okruh. Další možností je naopak rozsvícení světel při příchodu nebo při času nastaveného budíku.

Co se týče HW rozšíření, tak zde by se dala chytrá domácnost rozšířit například o bezpečnostní kamery nebo magnetické senzory na dveře a okna. Případně o chytré termostatické hlavice k radiátorům, za účelem ekonomicky výhodnějšího vytápění.

5.4 Závěrečné zamyšlení autora

V této sekci bych rád shrnul poznatky, které jsem získal při návrhu chytré domácnosti a tvorbě vlastních komponent. Myslím si, že pokud si člověk bude navrhovat chytrou domácnost založenou na open-source HW a SW, a to třeba až do takové míry, že bude tvořit jednotlivé komponenty, tak musí být opravdový nadšenec.

Navrhnout chytrou domácnost od začátku vyžaduje hodně času stráveného průzkumem existujících řešení, experimentování se součástkami, a tedy občas nákup součástek, které se ve výsledku vůbec nepoužijí, čímž se zvyšují pořizovací náklady. Dalším tématem, které je třeba řešit, je otázka bezpečnosti jednotlivých zařízení a celého návrhu, s tím poté souvisí průzkum aktuálních norem a jejich aplikace. Dá se tedy říct, že nevýhodou může být čas strávený nad zjišťováním informací a čas strávený nad implementací. Výhodou naopak může být téměř naprostá volnost tvorby.

Kapitola 6

Závěr

Cílem této práce bylo prozkoumat trh s komerčními řešeními elektrifikace chytré domácnosti a navrhnout vlastní infrastrukturu chytré domácnosti založenou na open-source software a/nebo hardware. Záměr práce byl splněn, a to včetně návržení několika vlastních komponent, jejich konstrukce a integrace do zvolené platformy pro domácí automatizaci.

Jako zástupci komerčních systémů byly vybrány tři systémy. Systém *Control4*, reprezentující drátové řešení, systém *Fibaro*, reprezentující bezdrátové řešení a systém *Loxone*, reprezentující kombinaci drátového a bezdrátového řešení chytré domácnosti.

Dále byl proveden návrh tří vlastních komponent pro ovládání střešního okna, ovládání garážových vrat a měření teploty. Současně byl s návrhem komponent proveden návrh infrastruktur osvětlení, zásuvkových okruhů, síťové infrastruktury a dále integrace IoT prvků do zvolené platformy. Zkoumanými volně dostupnými platformami byly platformy *ThingSpeak*, *ThingsBoard*, *Domoticz*, *openHAB* a *Home Assistant*. U těchto platform byly v textu práce popsány možnosti jejich instalace, podporované protokoly a technologie, možnosti jejich použití na jednotlivých zařízeních (telefon, web, desktop) a jimi nabízené funkcionality. Z těchto platform byla zvolena platforma *Home Assistant*, z důvodu širokých možností použití. Robustnost systému je dále zajištěna mimo jiné vhodným návrhem komponent a software, kde žádná významná komponenta není přímo závislá na ovládací platformě a dokáže fungovat i sama o sobě.

V rámci implementace byly poté vytvořeny vlastní komponenty, které jsou určeny k plnohodnotnému nasazení v rámci navržené chytré domácnosti. Tyto komponenty byly integrovány do platformy *Home Assistant*. Dále bylo provedeno zhodnocení implementovaného, respektive navrženého řešení a toto řešení bylo srovnáno s vybranými komerčními řešeními.

Práce mi zlepšila schopnost orientovat se v textu a vybrat z něj podstatné informace. Při provádění průzkumu jsem se dozvěděl o několika zajímavých technologiích, jakou je například *EnOcean* nebo o možnostech současného open-source internetu věcí. Dále jsem získal přehled o náročnosti tvorby vlastní chytré domácnosti a získal srovnání s domácnostmi komerčními.

Na práci by mělo být možné navázat vylepšením návrhu nebo jeho rozšířením o další komponenty chytré domácnosti. Implementované komponenty lze dále vylepšit tak, aby byla jejich konstrukce pohodlnější, případně lze provést optimalizaci software komponent.

Zkratky

AP Access Point (přístupový bod). 24, 28

DPS Deska plošných spojů. 29, 30, 34, 39, 40

HW Hardware. 29, 32, 43

IoT Internet Of Things (internet věcí). 5, 6, 13, 18, 19, 25, 28, 29

OTA Over-the-Air. 19, 29

SW Software. 5, 18, 22, 33, 36, 43

VAC Volts alternating current (střídavé napětí). 20, 24, 28, 30

VDC Volts direct current (stejnoseměrné napětí). 24, 26, 30, 31, 34, 35

Literatura

- [1] Blum, J.: *Exploring Arduino: tools and techniques for engineering wizardry*. John Wiley & Sons, 2013, ISBN 978-1-118-54936-0.
- [2] Domoticz: *Domoticz*. [Online; navštíveno 6.1.2019].
URL <https://www.domoticz.com/wiki/>
- [3] Domoticz: *Tizen - Domoticz*. [Online; navštíveno 24.1.2019].
URL <https://www.domoticz.com/wiki/Tizen>
- [4] EnOcean GmbH: *EnOcean is the Originator of the Patented Energy Harvesting Wireless Technology / EnOcean - Technology*. © 2019, [Online; navštíveno 24.1.2019].
URL <https://www.enocean.com/en/technology/patents/>
- [5] EnOcean GmbH: *Radio Technology (RF) from EnOcean for Energy Harvesting Wireless Sensor Solutions / EnOcean Technology*. © 2019, [Online; navštíveno 24.1.2019].
URL <https://www.enocean.com/en/technology/radio-technology/>
- [6] EnOcean GmbH: *Ultra-low Power Management / EnOcean - Technology*. © 2019, [Online; navštíveno 24.1.2019].
URL <https://www.enocean.com/technology/energy-harvesting-wireless/>
- [7] Gill, K.; Yang, S.-H.; Yao, F.; aj.: A zigbee-based home automation system. *IEEE Transactions on consumer Electronics*, ročník 55, č. 2, 2009: s. 422–430, ISSN 1558-4127.
- [8] Gomez, C.; Paradells, J.: Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, ročník 48, č. 6, 2010: s. 92–101, ISSN 1558-1896.
- [9] Gubbi, J.; Buyya, R.; Marusic, S.; aj.: Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, ročník 29, č. 7, 2013: s. 1645–1660, ISSN 1872-7115.
- [10] Kamalinejad, P.; Mahapatra, C.; Sheng, Z.; aj.: Wireless energy harvesting for the Internet of Things. *IEEE Communications Magazine*, ročník 53, č. 6, 2015: s. 102–108, ISSN 1558-1896.
- [11] KOPOS KOLÍN a.s.: *KRABICE S IP KRYTÍM*. [Online; navštíveno 25.04.2019].
URL <https://www.kopos.cz/cs/produkt/2884-krabice-s-ip-krytim>
- [12] Loxone Electronics GmbH: *Loxone Air*. [Online; navštíveno 15.12.2018].
URL <https://www.loxone.com/cscz/produkty/loxone-air/>

- [13] Lueth, K. L.: *State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating*. IoT Analytics GmbH, Srpen 2018, [Online; navštíveno 10.12.2018]. URL <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>
- [14] openHAB Community and the openHAB Foundation e.V.: *openHAB*. © 2019, [Online; navštíveno 6.1.2019]. URL <https://www.openhab.org/>
- [15] Schwartz, M.: *Internet of Things with ESP8266*. Packt Publishing Ltd, 2016, ISBN 978-1-78646-802-4.
- [16] The MathWorks, Inc.: *IoT Analytics - ThingSpeak Internet of Things*. © 2019, [Online; navštíveno 6.1.2019]. URL <https://thingspeak.com/>
- [17] wemos: *D1 mini*. [Online; navštíveno 29.04.2019]. URL https://wiki.wemos.cc/products:d1:d1_mini
- [18] Wikipedia contributors: *ESP8266* — *Wikipedia, The Free Encyclopedia*. 2018, [Online; accessed 10-December-2018]. URL <https://en.wikipedia.org/w/index.php?title=ESP8266&oldid=864287862>
- [19] Wikipedia contributors: *Z-Wave* — *Wikipedia, The Free Encyclopedia*. 2018, [Online; accessed 15-December-2018]. URL <https://en.wikipedia.org/w/index.php?title=Z-Wave&oldid=873761281>
- [20] Wikipedie: Energy harvesting — Wikipedie: Otevřená encyklopedie. 2013, [Online; navštíveno 24. 01. 2019]. URL https://cs.wikipedia.org/w/index.php?title=Energy_harvesting&oldid=10894735
- [21] YATUN, s. r. o.: *Technologie Z-Wave / FIBARO*. [Online; navštíveno 24.1.2019]. URL <https://www.mojefibaro.cz/system/technologie-z-wawe/>
- [22] Z-Wave Alliance: *About Z-Wave Technology*. [Online; navštíveno 15.12.2018]. URL https://z-wavealliance.org/about_z-wave_technology/

Přílohy

Seznam příloh

A	Obsah CD	50
B	Zdrojové kódy ve formátu YAML	51
C	Zdroje obrázků	57

Příloha A

Obsah CD

Tato příloha pojednává o obsahu CD přiloženého k výtisku této práce. Médium obsahuje zdrojové kódy vlastních komponent, vlastní část konfiguračního souboru platformy *Home Assistant* a soubor `README` s instrukcemi pro zprovoznění. Na disku je poté video nahrávka s ukázkou funkčnosti komponenty pro ovládání garážových vrat. Dále pak PDF s technickou zprávou a zdrojový text \LaTeX . Struktura média je následující:

`code/` Složka obsahující zdrojové kódy.

`code/configuration-custom.yaml` Vlastní část konfiguračního souboru platformy.

`code/wemos_relay_switch.yaml` Zdrojový kód pro komponentu ovládání garážových vrat.

`code/wemos_thermometer.yaml` Zdrojový kód pro komponentu měření teploty.

`code/wemos_window.yaml` Zdrojový kód pro komponentu ovládání střešního okna.

`latex/` Složka se zdrojovým kódem \LaTeX . Struktura z *Overleaf*.

`README.md` Soubor s postupem integrace komponent.

`video/doors_closing.mp4` Soubor s videm ukazujícím zavírání garážových vrat skrze platformu.

`video/doors_opening.mp4` Soubor s videm ukazujícím otevírání garážových vrat skrze platformu.

`xblaha28.pdf` Technická zpráva.

Příloha B

Zdrojové kódy ve formátu YAML

Tato příloha obsahuje kódy ve formátu YAML. Uvedené kódy obsahují popis chování jednotlivých komponent chytré domácnosti, doplněný komentářem k jednotlivým blokům kódu. Příložená je i vlastní část konfiguračního souboru platformy *Home Assistant*.

Ovládání okna

```
# Popis: Konfigurace komponenty ovladani okna.
# Autor: Radim Blaha
# Datum: 2019-05-11

# Nazev zarizeni pro DNS a informace o cipu a desce
esphome:
  name: wemos_window
  platform: ESP8266
  board: d1_mini

# Udaje pro pripojeni k WiFi
wifi:
  ssid: 'IoT_network'
  password: ',Aj9^&mtaEnbn#4R'

# Povoleni logovani
logger:

# Povoleni Home Assistant API
api:
  password: 'ahojsvete'

ota:
  password: 'ahojsvete'

binary_sensor:
  # senzor deste
  - platform: gpio
```

```

    pin: D0
    filters:
      - invert:
    name: "Rain"
    device_class: moisture
    id: rain_sensor

# senzor stavu okna
- platform: gpio
  pin: D3
  name: "Window Status"
  device_class: window

# ziskani stavu prepínace z platformy Home Assistant
- platform: homeassistant
  name: "Window Automation"
  entity_id: input_boolean.window_automation
  id: automation

switch:
  # pin pro zmenu stavu oteviraciho rele
  - platform: gpio
    pin: D5
    id: open
    interlock: [close]

  # pin pro zmenu stavu zaviraciho rele
  - platform: gpio
    pin: D6
    inverted: true
    id: close
    interlock: [open]

# teplotni cidlo DS18B20
dallas:
  - pin: D2
    update_interval: 5s # Z DUVODU UKAZKY

sensor:
  # ukazatel sily signalu
  - platform: wifi_signal
    name: "Window Controller Signal Sensor"
    update_interval: 5s

# teplotni cidlo DS18B20
- platform: dallas
  address: 0xCF051760278EFF28
  name: "Outside temperature"

```



```

id: outside_thermometer
# automatizace provadena pri zmene teploty
on_value:
  if:
    condition:
      binary_sensor.is_on: automation
    then:
      if:
        condition:
          lambda: |-
            if(id(rain_sensor).state ||
              id(outside_thermometer).state < 27)
              return true;
            return false;
        then:
          cover.close: cover
        else:
          cover.open: cover

# komponenta pro ovladani okna
cover:
- platform: template
  name: "Window"
  id: cover
  open_action:
    - switch.turn_off: close
    - delay: 0.5s # zpozdeni k ustaleni napeti
    - switch.turn_on: open
    - delay: 20s
    - switch.turn_off: open
  close_action:
    - switch.turn_off: open
    - delay: 0.5s # zpozdeni k ustaleni napeti
    - switch.turn_on: close
    - delay: 20s
    - switch.turn_off: close
  stop_action:
    - switch.turn_off: open
    - switch.turn_off: close
  optimistic: true

```

Ovládání garážových vrat

```

# Popis: Konfigurace komponenty ovladani garazovych vrat.
# Autor: Radim Blaha
# Datum: 2019-05-11

# Nazev zarizeni pro DNS a informace o cipu a desce

```

```

esphome:
  name: wemos_relay_switch
  platform: ESP8266
  board: d1_mini

# Udaje pro pripojeni k WiFi
wifi:
  ssid: 'IoT_network'
  password: ',Aj9^&mtaEnbn#4R'

# Povoleni logovani
logger:

# Povoleni Home Assistant API
api:
  password: 'ahojsvete'

ota:
  password: 'ahojsvete'

sensor:
  # ukazatel sily signalu
  - platform: wifi_signal
    name: "Garage Door Signal Sensor"
    update_interval: 5s

binary_sensor:
  # senzor stavu vrat
  - platform: gpio
    pin:
      # snimani kladneho polu
      number: D0
      mode: INPUT_PULLDOWN_16
      inverted: True
    name: "Garage Door Status"
    device_class: garage_door

# komponenta pro spinani rele
switch:
  - platform: gpio
    pin: 5
    id: relay

  - platform: template
    name: "Garage Door"
    icon: "mdi:gate"
    optimistic: no
    # sepnuti rele po dobu 0.5s

```

```
turn_on_action:
- switch.turn_on: relay
- delay: 500ms
- switch.turn_off: relay
```

Měření teploty

```
# Popis: Konfigurace komponenty mereni teploty.
# Autor: Radim Blaha
# Datum: 2019-05-11

# Nazev zarizeni pro DNS a informace o cipu a desce
esphome:
  name: wemos_thermometer
  platform: ESP8266
  board: d1_mini

# Udaje pro pripojeni k WiFi
wifi:
  ssid: 'IoT_network'
  password: ',Aj9^&mtaEnbn#4R'

# Povoleni logovani
logger:

# Povoleni Home Assistant API
api:
  password: 'ahojsvete'

ota:
  password: 'ahojsvete'

# Nastaveni doby behu a doby spanku
# Home Assistant se pojí jednou za 60s
deep_sleep:
  run_duration: 75s
  sleep_duration: 30s # Z DUVODU UKAZKY

# teplotni cidlo DS18B20
dallas:
  - pin: D2
    update_interval: 5s

sensor:
  # ukazatel sily signalu
  - platform: wifi_signal
    name: "Thermometer Signal Sensor"
    update_interval: 5s
```

```

# mereni napeti na baterii a prevod na procenta
- platform: adc
  pin: A0
  name: "Thermometer Battery Life"
  update_interval: 5s
  filters:
    # prepocet napeti na procenta
    # 0.666666 je min. namerena hodnota napeti na baterii
    - lambda: |-
        float batteryLife = (x - 0.666666) * 300;
        if(batteryLife < 0)
            return 0;
        else if(batteryLife > 100)
            return 100;
        else
            return batteryLife;
  unit_of_measurement: "%"
  icon: "mdi:battery-outline"

# teplotni cidlo DS18B20
- platform: dallas
  address: 0x9C000006098A2028
  name: "Custom DS18B20"

```

Vlastní část konfiguračního souboru platformy

```

# Popis: Vlastni cast konfigurace platformy.
# Autor: Radim Blaha
# Datum: 2019-05-11

input_boolean:
  window_automation:
    name: Window Automation Switch
    initial: off
    icon: mdi:home-automation

```

Příloha C

Zdroje obrázků

Protože určité dílčí obrázky z kapitoly 3, sekce 3.3 byly převzaty z jiných zdrojů, v této příloze jsou uvedeny zdroje, ze kterých byly získány. Obrázky, které jsou takto složeny jsou obrázek 3.6, obrázek 3.7, obrázek 3.8.

- https://wiki.wemos.cc/products:d1:d1_mini
- https://wiki.wemos.cc/products:d1_mini_shields:relay_shield
- https://wiki.wemos.cc/products:d1_mini_shields:dc_power_shield
- <https://img.staticbg.com/thumb/large/oaupload/banggood/images/OE/2A/ba413450-e2bb-4229-acd6-f63c52f1e5b5.jpg>
- <http://robotstore.cz/wp-content/uploads/2017/01/ds18b20-stit-pro-wemos-d1-mini-wifi-esp8266-nodemcu-arduino-2.jpg>
- <https://laskarduino.cz/vstupni-periferie-cidla/230101-destovy-senzor.html>
- http://www.andromedagie.it/files/Andromeda_Files/Foto/8160_9.JPG
- <https://laskarduino.cz/shield-moduly/230111-wemos-d1-mini-lithium-battery-shield.html>
- <https://www.ges.cz/cz/dratovy-rezistor-3w-draht-33r0-GES05313584.html>