



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

NA MODELECH ZALOŽENÝ NÁVRH A REALIZACE SUB-SYSTÉMU SMART HOME

MODEL-BASED DESIGN AND DEPLOYMENT OF A SMART HOME SUBSYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ STANĚK

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. VLADIMÍR JANOUŠEK, Ph.D.

BRNO 2022

Zadání bakalářské práce



Student: **Staněk Tomáš**
Program: Informační technologie
Název: **Na modelech založený návrh a realizace subsystému Smart Home**
Model-Based Design and Deployment of a Smart Home Subsystem
Kategorie: Softwarové inženýrství

Zadání:

1. Prostudujte problematiku modelem řízeného návrhu a modelovací jazyky SysML a IEC 61499. Seznamte se s vhodnými nástroji pro SysML a také s nástroji 4diac a Node-RED.
2. Seznamte se s dostupnými hardwarovými prostředky pro realizaci Smart Home, s protokolem MQTT, ZigBee prvky a s dostupným opensource softwarem, jako je Home Assistant a Domoticz.
3. Stanovte požadavky na řízení chytré domácnosti. Zaměřte se na řízení topení, ale vezměte v úvahu i řízení osvětlení, ventilace a kamerový dohled.
4. Navrhněte řídicí systém pro vybrané subsystémy z bodu 3 s využitím modelů za pomoci vybraných nástrojů z bodu 1. Pro potřeby průběžného ověřování návrhu vytvořte i vhodně abstrahovaný model řízeného prostředí včetně senzorů a aktuátorů a propojení s dohledovým systémem, jako je např. Home Assistant.
5. Po vygenerování kódu pro řídicí systém a jeho zprovoznění otestujte systém v reálném provozu a proveďte vyhodnocení zvoleného postupu a dosaženého výsledku.

Literatura:

- 4diac/FORTE. URL: <https://www.eclipse.org/4diac/>
- Node-RED. URL: <https://nodered.org/>
- Dle pokynů vedoucího.

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

Vedoucí práce: **Janoušek Vladimír, doc. Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 3. listopadu 2021

Abstrakt

Tato bakalářská práce pojednává o možnostech a metodách vytvoření smart home, za použití běžně dostupných prvků pro každého uživatele. Bakalářská práce je zaměřena především na problematiku regulace vytápění domácnosti, ventilaci, řízení osvětlení, či kamerový dohled. Pro ovládání jednotlivých prvků se využívá systému Home Assistant běžícího na Raspberry Pi.

Abstract

This bachelor thesis discusses about the possibilities and methods of creating smart home, using commonly available elements fore each user. The bachelor´s thesis is focused primarily on the issues of home heating control, ventilation, lighting control, camera surveillance. For control individual elements is used Home assistant running on the Raspberry Pi

Klíčová slova

Internet věcí, Home Assistant, Inteligentní dům, Regulace vytápění, Raspberry Pi, Node-Red, ZigBee, IEC 61499, SysML

Keywords

Internet of Things, IoT, Home Assistant, Smart Home, Regulation of heating, Raspberry Pi, Node-Red, ZigBee, IEC 61499, SysML

Citace

STANĚK, Tomáš. *Na modelech založený návrh a realizace subsystému Smart Home*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Doc. Ing. Vladimír Janoušek, Ph.D.

Na modelech založený návrh a realizace subsys- tému Smart Home

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Doc. Ing. Vladimíra Janouška Ph.D. a uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Tomáš Staněk
10. května 2022

Poděkování

Rád bych poděkoval panu Doc. Ing. Vladimíru Janouškovi Ph.D. za poskytnutý čas, odbornou pomoc a konzultaci při vytváření řešení. Dále bych rád poděkoval své rodině, která mě podporovala v průběhu vypracování bakalářské práce.

Obsah

1	Úvod	3
1.1	Motivace	3
1.2	Kapitoly	3
2	Důležité pojmy	4
2.1	Internet věcí - IoT	4
2.1.1	Smart Home	4
2.2	MQTT	4
2.2.1	Základní koncept MQTT	5
2.2.2	Kvalita Služeb	6
2.3	Řídící jednotka	8
2.3.1	Raspberry Pi	8
2.3.2	Banana Pi	8
2.3.3	Orange Pi	8
2.3.4	Porovnání řídicích jednotek	8
2.4	Komunikační technologie	9
2.4.1	Z-wave	9
2.4.2	ZigBee	10
2.4.3	Ostatní technologie	12
2.4.4	LoRaWan	12
3	Software	13
3.1	Home Assistant	13
3.1.1	Instalace	14
3.2	Domoticz	14
3.2.1	Instalace	16
3.3	OpenHAB	16
3.3.1	Instalace	17
4	Modelovací jazyky	18
4.1	SysML	18
4.1.1	Modelio	19
4.1.2	Papyrus	20
4.2	IEC 61499	21
4.2.1	Eclipse 4diac	22
4.2.2	Node-RED	23
4.2.3	Open Modelica	24

5	Požadavky na řídicí systém	27
5.1	Požadavky na systém	27
5.2	Požadavky na Hardwarová zařízení	28
5.3	Požadavky na software řídicího systému	28
6	Návrh a Realizace Řešení	30
6.1	Server	31
6.2	Hardware	31
6.2.1	Koordinátor	31
6.2.2	Relé	32
6.2.3	Termostatické hlavice	33
6.2.4	Detektory pohybu	36
6.2.5	Osvětlení	37
6.2.6	Detektory dveří a oken	37
6.2.7	Tlačítko	38
6.2.8	Kamera	39
6.3	Software	39
6.3.1	Návrh subsystémů	39
6.3.2	Realizace řešení v Home Assistant	45
7	Závěr	54
	Literatura	55
A	Vytvořené SysML diagramy	58
B	Node-RED modely	60
C	Obsah příloženého paměťového média	62

Kapitola 1

Úvod

V dnešní době se stále častěji můžeme setkávat s pojmy jako je Internet věcí (dále jen IoT) nebo Smart Home. Dnes přibývá stále více domácností, které využívají nějaké chytré zařízení jako je např. ovládání spotřebičů, ovládání světel, kamerové systémy, regulace vytápění, kontrola zamčených nebo zavřených dveří či oken a mnohé další. Pokrok v oblasti vestavěných systému je aktuálně velmi rychlý. Každý měsíc se objevují stále nové prvky IoT, které by nám měli usnadnit život.

Za velmi zajímavou a důležitou součást chytrých domácností se považuje chytré vytápění. Nejen z důvodu možnosti nastavit si požadovanou teplotu v jednotlivých místnostech, ale i kvůli úspoře obrovského množství energie a tím i úspoře peněz. V tomto směru je daleko více možností, jakým způsobem může k úspoře dojít. Nejprve je důležité zvolit vhodné tepelné zařízení. Správným výběrem dochází k úspoře. Po výměně kotle za nový ekologický může dojít až k 20 % úspoře paliva ročně.

Využití kamerových systému případně osvětlení podle pohybu osob je dnes již běžnou záležitostí. Většinou jsou ovšem tyto prvky ovládány manuálně, a tak málo lidí využívá připojení přes mobil, kde mohou sledovat aktuální dění.

1.1 Motivace

V současné době představujeme tepelný systém v našem rodinném domě. Řešení takového systému představuje poměrně velmi nákladnou věc. Proto se chci zaměřit na to, aby cena za takový systém byla daleko nižší, než nabízí komerční firmy. Vzhledem k možnosti využití jakýchkoliv prostředků se mohu dostat k finanční úspoře až desítek tisíc korun ročně.

1.2 Kapitoly

Práce je rozdělena do několika věcných kapitol, které by postupně měli vysvětlit jednotlivé části daného projektu. V 2. kapitole Důležité pojmy je rozebíráno důležité názvosloví pro seznámení se s částmi bakalářské práce. Ve 3. kapitole Software je popsán modelovací jazyk SysML. 4 kapitola Modelovací jazyky pojednává o jednotlivých prostředcích pro vytvoření funkčního Smart Home. V 5. kapitole Požadavky na řídicí systém jsou probrány požadavky na projekt v rámci zadání i požadavky samotného uživatele. V 6. kapitole Návrh a Realizace řešení je rozebrán postup při realizaci systému a komponenty, které byly pro řešení použity.

Kapitola 2

Důležité pojmy

2.1 Internet věcí - IoT

Zkratka IoT znamená Internet of Things neboli v překladu Internet věcí. Jedná se o síť fyzických zařízení, vozidel, domácích spotřebičů a dalších zařízení, která jsou vybavena elektronikou, softwarem, senzory, pohyblivými částmi a síťovou konektivitou, která umožňuje těmto zařízením se propojit a vyměňovat si data.[26]

2.1.1 Smart Home

Poprvé se pojem smart home objevuje v 80. letech a dává tak základ následujícím letům rozvoje domácností a automatizací.

Myšlenka smart home se vážněji začala brát až na přelomu 20. a 21. století, kde některé firmy představují svoje vize inteligentních domů. Překvapivě se tehdejší myšlenka neliší tolik od současnosti. Tehdejší systémy měli ovládat zámky, televizory, osvětlení, vytápění, komunikaci s návštěvníky a ke všemu i vytvářet nákupní list dle naskenovaného čárového kódu. Tím byl dán základ pro dnešní inteligentní domy.[2]

V dnešní době existuje několik způsobů, jak inteligentní dům definovat. Všechny tyto definice vedou, ale vždy ke stejným závěrům. Inteligentní budova musí splňovat:

- Zvýšení komfortu osob využívající služby a možnosti Smart Home
- Snížení nákladů na bydlení
- Lepší zabezpečení bydlení
- Prodloužení životnosti domácnosti či domácích spotřebičů

2.2 MQTT

MQTT neboli message queuing telemetry je extrémně lehký protokol, který obvykle běží přes TCP/IP a funguje na principu odběru a publikování (subscribe and publish). MQTT je určen pro sítě s malou šířkou pásma, vysokou latencí nebo u nespolehlivých sítí. Je navržen tak, aby snížil šířku pásma a zajistil bezpečný přenos informací. Umožňuje například odesílat příkazy pro kontrolu výstupů, případně publikovat a číst data ze senzorů.[6]

2.2.1 Základní koncept MQTT

Broker

Prvním konceptem, kterému je potřeba porozumět je Broker. Broker je zodpovědný za příjem všech zpráv, filtrování zpráv, určuje, kdo se přihlásil k odběru zprávy a odesílá zprávy těm, kdo jsou k odběru přihlášení. Dalšími úkoly brokera je autentizace a autorizace klientů. Je to tedy takový centrální rozbočovač, přes kterého musí být směrována každá zpráva. Je proto velmi důležité, aby byl broker vysoce škálovatelný. [6]

MQTT Broker používá TLS¹ šifrování s uživatelským jménem, heslem a volitelnými certifikacemi. Tyto certifikace poskytné klient a musí se shodovat s certifikacemi na serveru.

Broker může filtrovat zprávy podle několika způsobů:[23]

1. Filtrování podle tématu

- Filtruje se podle předmětu nebo podle tématu zprávy
- Broker zajistí, aby přijímající klient získal vše k odebíraným tématům

2. Filtrování podle obsahu

- Broker filtruje na základě konkrétního obsahu filtru
- Odběratelé se přihlásí k odběru zpráv, o které mají zájem
- Nevýhodou je, že obsah musí být známý a nemůže být šifrován

3. Filtrování podle typu

- Nastává pouze u objektově orientovaných jazyků
- Odběratel může například poslouchat všechny zprávy typu výjimka nebo jakýkoli podtyp

Publish/Subscribe

System Pub/Sub odděluje klienta, který zprávu odesílá od klienta nebo klientů, kteří zprávu přijímají. Komunikace však nikdy neprobíhá přímo. Ve skutečnosti ani jeden neví, že ten druhý existuje. Komunikaci mezi nimi zpracovává výše zmíněný broker.

Nejdůležitějším aspektem pub/sub je oddělení vydavatele zprávy od příjemce. Toto oddělení má několik rozměrů:[23]

- **Oddělení prostoru:** Vydavatel a příjemce se nemusí navzájem znát
- **Oddělení času:** Vydavatel a příjemce nemusí pracovat současně
- **Oddělení synchronizace:** Operace na příjemci nebo na vydavateli není nutné přerušovat během přijímání nebo vydávání

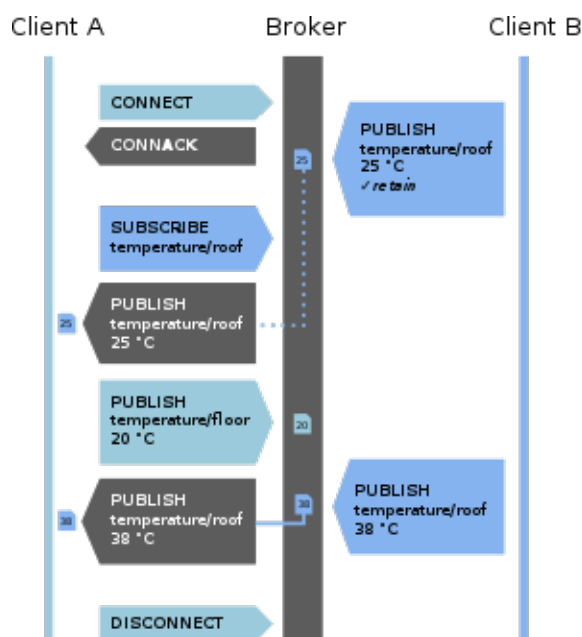
Zprávy

Jsou to informace, u kterých je požadováno, aby si je mezi sebou jednotlivá zařízení vyměňovala. Ať už se jedná o příkaz, například ať se rozsvítí světla v kuchyni, protože do ní někdo právě přišel. Nebo se může jednat o data typu, v ložnici je 23 stupňů a to je teplota při, které nemá docházet k žádné akci.[1]

¹TLS - Transport Layer Security

Typy zpráv

- **Connect:** navázání připojení
- **Connack:** Odezva od brokeru s návratovou zprávou zda spojení je otevřené.
- **Subscribe:** Klient pošle brokeru informaci, kterou zprávu bude chtít přijímat. Tato zpráva má jednoznačný identifikátor.
- **Publish:** Slouží k publikování zpráv.
- **Disconnect:** Odhlašuje klienta z procesu.



Obrázek 2.1: Příklad komunikace MQTT.[28]

Témata

Posledním konceptem jsou témata. Témata jsou reprezentována jako řetězce, které je třeba oddělit lomítkem. Každé lomítko pak představuje úroveň tématu. Za pomoci témat lze registrovat zájem o příchozí zprávy, nebo určit, kde mají být jednotlivé zprávy publikovány.[23]

Zahrada/pergola/kamera

Jak je vidět na příkladu, tak zahrada a pergola představují úroveň tématu. Samotná kamera pak představuje zařízení, které může vysílat, nebo přijímat zprávy.

2.2.2 Kvalita Služeb

Kvalita služeb definuje záruku doručení konkrétní zprávy mezi odesílatelem a příjemcem zprávy. Součástí MQTT jsou 3 úrovně kvality služeb:

- Maximálně jednou

- Alespoň jednou
- Přesně jednou

Maximálně jednou

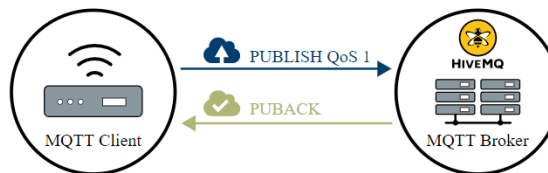
V tomto případě je zpráva odeslána maximálně jednou. Příjemce ale nepotvrdí, zda zprávu přijal. Odesílatel zprávu již znovu nepošle. Není zde tedy garantováno, že byla zpráva doručena a v pořádku přijata.



Obrázek 2.2: Úroveň kvality služeb - maximálně jednou.[23]

Alespoň jednou

V této úrovni je zaručeno, že zpráva je příjemci doručena alespoň 1x. Odesílatel uloží zprávu, dokud od příjemce nedostane potvrzení o přijetí zprávy. V případě, že po uplynutí nastaveného času potvrzení nedostane, odešle zprávu znovu. Při znovu odeslání zprávy se nastaví příznak DUP. Tento příznak nemá na nic vliv. Jedná se pouze o to, aby bylo známo, že se jedná o duplicitní zprávu.

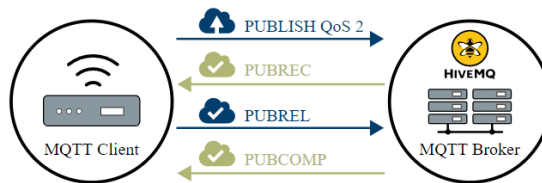


Obrázek 2.3: Úroveň kvality služeb - alespoň jednou.[23]

Přesně jednou

Poslední úroveň je zároveň i nejbezpečnější úroveň. Důkladné zabezpečení ale dělá z této úrovně také nejpomalejší službu. Tato úroveň zaručuje, že zpráva bude příjemci přijata právě jednou. Záruka je poskytována nejméně dvěma toky požadavků.

Nejprve je odeslána původní zpráva. V případě, že ji příjemce přijme odešle zpět zprávu o přijetí. V případě, že zpráva přijata nebude, odesílatel odešle po určitém čase zprávu znovu s příznakem DUP. Odesílatel čeká na zprávu od příjemce, že k němu zpráva přišla. V případě, že potvrzení přijde. Odesílatel maže původně odeslanou zprávu.



Obrázek 2.4: Úroveň kvality služeb - přesně jednou.[23]

2.3 Řídící jednotka

Řídící jednotky jsou zařízení, která ovládají a řídí celý systém. Samozřejmě je několik možností, jak zvolit vhodnou řídicí jednotku. Jako první se nabízí desktopový počítač případně notebook. Toto řešení je pro účely inteligentních domů velmi nevhodné. Důvod je především obrovská spotřeba, která může dosahovat až několika Wattů.

Proto bylo důležité se zaměřit na alternativní řešení, které svou spotřebou a celkovými náklady budou systému vyhovovat. Jako ideální řešení se nabízí některý z mikropočítačů, které se v současné době na trhu nachází.

Jeden z nejznámějších, v současné době poměrně snadno dostupných mikropočítačů, je Raspberry Pi. Ale i produkty Raspberry mají již několik konkurentů. Mezi nejvážnější konkurenty se aktuálně řadí například Orange Pi nebo Banana Pi.

2.3.1 Raspberry Pi

Raspberry Pi je malý jednodeskový počítač s deskou plošných spojů. Byl vyvinut již v roce 2012. Od té doby bylo vyvinuto 9 modelů Raspberry. Poslední vydanou verzí je Raspberry Pi Pico, která byla vydána v lednu 2021.[34]

2.3.2 Banana Pi

Banana Pi vznikl v roce 2013 a měl být jednou z hlavních konkurencí vznikající komunity Raspberry Pi. Dnes je Banana Pi stále velmi využíváný, i přestože už má svého následovníka Banana PRO.[32]

2.3.3 Orange Pi

Orange Pi je taktéž jednodeskový počítač s otevřeným zdrojovým kódem. Je velmi podobný své konkurenci, tj. mikropočítačům Banana Pi nebo Raspberry Pi. Základem všech mikropočítačů Orange Pi je procesor ARM Cortex-A7. Oproti své konkurenci má především velkou výhodu v připojení dalších periférií, jako jsou pevné disky nebo nejrůznější redukce.[33]

2.3.4 Porovnání řídicích jednotek

Pro porovnání mezi řídicími jednotkami bylo vybráno Raspberry Pi model 4, který je aktuálně snadno dostupný a plně nahradil předchozí model Raspberry Pi model 3 B.

Dále Banana Pi m5 a Orange Pi Plus, které jsou cenově srovnatelné s Raspberry Pi 4.

	Raspberry Pi 4 Model B	Banana Pi M5	Orange Pi Plus 2E
Procesor	Čtyřjádrový procesor Broadcom BCM2711, Cortex-A72	Čtyřjádrový procesor Alogic S905X Cortex a-55	Čtyřjádrový procesor Allwinner H3 ARM Cortex-A7
RAM	1/2/4/8 GB LPDDR4	4 GB LPDDR4	2GB DDR3
Úložný prostor	Micro SD	Micro SD	Micro SD
USB porty	2x USB 3.0, 2x USB 2.0	4 xUSB 3.0	3x USB 2.0
GPIO	40 Pinů	40 Pinů	40 Pinů

Tabulka 2.1: Porovnání řídicích systémů.[33],[25]

2.4 Komunikační technologie

Komunikační technologie jsou buď drátové nebo bezdrátové. Způsob řešení drátovou cestou je vhodný v momentě, kdy je stavba rozestavěná a není problém v ní dělat úpravy. V případě, řešení inteligentní domácnosti v již dostavěném domě je lepší využít komunikaci bezdrátovou. Díky tomu se nemusí řešit složitá úprava domácnosti. Není to však jediný důvod, v čem je bezdrátová síť napřed oproti mechanické. Dalšími důvody jsou škálovatelnost, cena, flexibilita a další.

V dnešní době existuje v IoT několik bezdrátových technologií, které mají vždy stejný cíl, tj. přijímat a odesílat data mezi sebou. Každá z těchto technologií jsou pro různá zařízení lepší než jiná. Faktory, které ovlivňují výběr technologie jsou :[14]

- nároky na zdroj energie
- dosah
- požadavky na výkon
- rychlost přenosu dat

2.4.1 Z-wave

Z-wave je bezdrátový komunikační protokol, který vznikl speciálně pro řešení domácí automatizace. Poprvé se tento protokol objevuje již roce 1999 v Kodani. Vážněji se technologie prosazuje až v roce 2005, kdy je Z-Wave přijat v Severní Americe několika firmami.[29]

Technologie Z-wave používá nelicencované průmyslové, vědecké a zdravotní pásmo. Pracuje na frekvenci 868,42 MHz. Díky této frekvenci není v domácnosti rušena jinými bezdrátovými technologiemi, které většinou pracují na frekvenci 2,4GHz. V tomto pásmu lze najít například Wi-Fi, bluetooth, NFC a další. Rychlost přenosu dat je 9600 bit/s - 40 kbit/s. Ve volném prostoru je dosah 90 metrů. V případě uzavřených prostorů je dosah asi 30 metrů. [29]

Topologie

Z-wave využívá architekturu tzv. mesh sítí, které jsou známy také jako bezdrátové ad hoc sítě. V těchto sítích mohou mezi sebou veškerá zařízení sdílet informace o tom, co se na nich právě odehrává. Tyto informace mohou být dále předány uzlům, ke kterým jsou přímo připojeny a jsou tak v dosahu daných zařízení. V případě, že zařízení potřebuje předat

informace uzlu, ke kterému není přímo připojeno (např. z důvodu vzdálenosti od uzlu), může využít ostatní uzly k přenosu informací na dané zařízení. Díky této možnosti se může Z-wave síť rozšířit do poměrně velkých rozměrů. Navíc je tak zaručeno, že v případě překážky v komunikaci mezi dvěma zařízeními mohou být informace předány přes zařízení jiné.[29]

Zabezpečení

Zabezpečení Z-wave je velmi specifické. V jedné síti mohou být zařízení, která jsou zabezpečena, ale mohou se v ní nacházet zařízení, která zabezpečena nejsou. Je to z toho důvodu, že ne všechna zařízení je možné zabezpečit. Díky tomu se dají využít v síti i produkty, které zabezpečit nelze. Je poté na implementaci každé aplikace a každého zařízení, aby se rozhodli, zda využijí bezpečnostní prvky.[16]

Bezpečnostní vrstva Z-wave poskytuje 3 funkce zabezpečení:

- **Integrita zpráv:** příjemce si může být jistý, že odeslaná zpráva byla doručena zabezpečeným uzlem a nebyla nijak změněna
- **Důvěrnost:** data mohou číst pouze zabezpečené uzly v síti. Ostatním se zobrazují jako náhodná data
- **Aktuálnost:** příchozí data jsou aktuální, tzn. nejsou zastaralá

2.4.2 ZigBee

Zigbee je bezdrátový komunikační protokol, který je založený na standardu IEEE 802.15.4². Poprvé se Zigbee objevuje již v roce 1998. Platným standardem je až od roku 2004.

Technologie byla vyvinuta za účelem vytvoření nízko-energetické a nízko-nákladové domácí sítě. Je možné ji využít k domácí automatizaci shromažďování údajů o zdravotnických či průmyslových zařízeních a jiné projekty menšího rozsahu.[31]

Technologie Zigbee může pracovat na 4 rádiových pásmech:

- pásmo ISM 2,4 GHz, přenosová rychlost 250 kb/s, definováno celosvětově
- pásmo 915 MHz, přenosová rychlost 40 kb/s, definováno pro Americký kontinent
- pásmo 868 MHz, přenosová rychlost 20 kb/s, definováno pro Evropu
- pásmo 784 MHz, definováno pro Čínu

Vzhledem k tomu, že pásma 915 MHz a 868 MHz nejsou celosvětově pokryta, je nejčastěji využívaným pásmem pro ZigBee pásmo 2,4 GHz.[31]

Topologie

Struktura Zigbee sítě se skládá ze 3 druhů zařízení:

- **Koordinátor:** Tvoří kořen síťového stromu, může se přemísťovat do jiných sítí a v každé síti se nachází pouze jeden. Ukládá informace o síti, včetně fungování a adresáře pro bezpečnostní klíče.

²IEEE 802.15.4 - https://en.wikipedia.org/wiki/IEEE_802.15.4

- **Router:** Směřuje data na jiná zařízení. Spouští aplikační funkce. Jedná se o zařízení s plnou funkčností (FFD³).
- **Koncová zařízení:** Obsahuje pouze funkce nutné pro komunikaci s routerem nebo koordinátorem. Lze konstatovat, že se jedná o zařízení s omezenou funkčností (RFD⁴).



Obrázek 2.5: ZigBee Koordinátor CC2531.[7]

Síťová vrstva

Síťová vrstva má za úkol správné použití podvrstvy MAC a poskytnutí vhodného rozhraní pro další vrstvu. Následujícím úkolem je připojení, odpojení a nastavení sítě, a také přidávání a odebírání určitých zařízení.

Síťová vrstva používá k směrování protokol AODV. Ten odesílá požadavek za účelem nalezení nového zařízení všem svým sousedem a ti pokračují stejným způsobem dokud zařízení nebude nalezeno. Jakmile přijde zdroji od zařízení odpověď, aktualizuje směrovací tabulku.[31]

Aplikační vrstva

Jedná se o vrstvu nejvyšší úrovně. Je zodpovědná za správu zařízení, bezpečnost klíčů a politiku sítě. Dále odesílá zprávy mezi zařízeními, spravuje skupinové adresy a přenáší data.[31]

Zabezpečení

Jako základní zabezpečení ZigBee se používá AES⁵ s klíčem o délce 128 bitů, jež je implementován v síťové vrstvě. Pokud je požadováno i zabezpečení MAC Command Frame, Beacon Frame a Acknowledgement Frame je realizováno již v MAC vrstvě pomocí AES. Díky tomu je možné ověřit autenticitu a integritu MAC rámce a zajistit jeho důvěrnost. Při požadavku na ověření integrity je vytvořen MIC⁶ o délce 4, 8 či 16 oktetů a je vyslán společně s MAC rámcem. V tomto případě je použit AES algoritmus v CTR (Counter) módu. Pokud je nutné zajistit důvěrnost MAC rámce je k němu přidána informace o pořadí rámce a klíče. Na vysílací a přijímací straně je udržována aktuální informace o čísle rámce. Pokud obdrží přijímací zařízení rámec s neplatným číslem je detekováno narušení bezpečnosti. AES je použit v CBC-MAC⁷ módu. Při implementaci, jak ověřování integrity, tak šifrování je použit AES v módu, jež je nazýván CCM.

Síťová vrstva používá k zabezpečení SSP⁸. Tato vrstva zajišťuje zabezpečení odchozích rámců, dekodování a ověřování pravosti příchozích rámců. Jako zabezpečovací algoritmus je

³FFD - Full Functional Device

⁴RFD - Reduced Functionality Device

⁵AES - Advanced Encryption Standard

⁶MIC - Message Integrity Code

⁷CBC - Cipher Block Chaining

⁸SSP - Security Services Provider

použit AES v mírně modifikovaném módu CCM. Síťová vrstva je zodpovědná za realizaci zabezpečení. Vyšší vrstvy se starají o nastavení SSP.[31]

2.4.3 Ostatní technologie

Wifi

Pravděpodobně nejznámější bezdrátovou komunikační technologií je wifi. Tato technologie je především spojena s připojením počítačů, tabletů, mobilů a dalších zařízení k internetu. Centrálním bodem, každé wifi sítě je router. Ten podle konfigurace může připojit stovky klientů najednou, maximálně cca 250. Výhodou wifi sítě je snadná integrace a je velmi snadno rozšiřitelná při přidávání dalších klientů.

V dnešní době mnoho IoT zařízení využívá pro připojení právě technologii wifi. Ovšem i u této technologie může nastat problém. Software pro wifi a TCP/IP je poměrně komplexní. Tento fakt může znamenat problém u zařízeních, která nedisponují tak velkým výkonem, jako je například počítač. Tento problém je možné vyřešit pomocí modulů s embedded wifi a TCP/IP softwarem.[14]

Bluetooth a Bluetooth Low Energy

Bluetooth a Bluetooth Low Energy (dále jen BLE) jsou bezdrátové technologie pro přenos dat na krátkou vzdálenost. Jejich využití je nejčastěji spjato s mobilními telefony. Využívá se pro připojení zařízení na velmi krátkou vzdálenost (maximálně 10 metrů). Navíc k technologii lze připojit standardně pouze 8 zařízení. V chytré domácnosti je to prakticky nevyužitelné. Ovšem často se objevují jiná zařízení, která přes technologii Bluetooth spojujeme, jako jsou chytré hodinky, bezdrátová sluchátka a jiné.

BLE potřebuje k přenosu dat méně energie než Bluetooth. Využívá se například u průmyslových snímačů nebo lékařských přístrojů. U chytré domácnosti je Bluetooth i BLE prakticky nepoužitelné.[14]

2.4.4 LoRaWan

Jedná se o protokol, který vznikl záměrně pro IoT aplikace v rozsáhlých sítích. Používá malé rychlosti přenosu od 0,3 do 50 kbps. Využívá nelicencované pásmo pod 1 GHz. Velkou výhodou je dosah, který činí až 20 km.

V porovnání s ostatními bezdrátovými technikami je bezpečnější. Přenáší šifrovaná data na různých kmitočtech a s různými přenosovými rychlostmi.⁹[14]

⁹LoRaWAN - Long Range Area-Wide Networks

Kapitola 3

Software

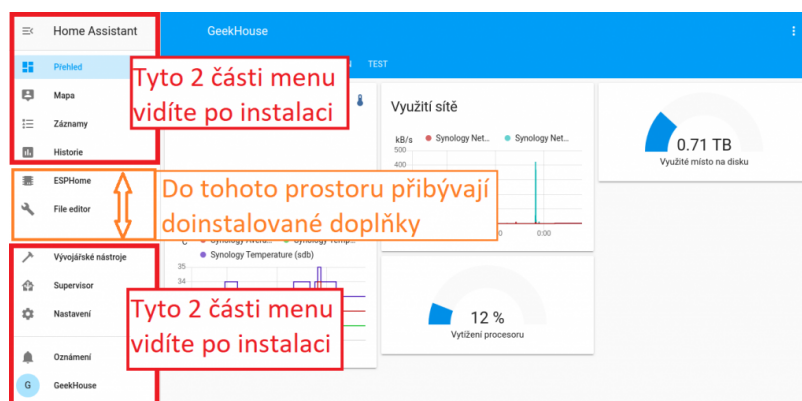
Home Automation Software neboli volně přeloženo do češtiny software domácí automatizace slouží ke sdružení veškerých chytrých zařízení v jedné aplikaci. Díky tomuto sjednocení se pak dají všechna zařízení ovládat a automatizovat z jedné aplikace.

Takovýchto softwarů je na trhu aktuálně mnoho. Ovšem za proprietární softwarové aplikace si uživatelé musí platit i několik tisíc ročně. Přesto existují open source produkty, které se proprietárním produktům dokáží vyrovnat. Mezi ty nejznámější a nejpoužívanější patří Home Assistant, Domoticz, OpenHab.

3.1 Home Assistant

Tento projekt vznikl v roce 2013, když byl poprvé veřejně publikován na Githubu. Následně se začali objevovat lidé, kteří chtěli Home Assistant dále rozvíjet. V prosinci 2020 má tak tento open-source skoro 2300 vývojářů, kteří pro Home Assistant dělají nová vydání prakticky každý týden. Celý Home Assistant je napsaný v Pythonu.[30]

Uživatel, který používá Home Assistant využívá takzvaný lovelace. Lovelace je řídicí panel, na kterém si uživatel může spravovat svoji domácnost. Na tento panel si může umístit až 29 různých karet, které mohou být libovolně umístěny a konfigurovány.



Obrázek 3.1: HomeAssistant Grafické rozhraní.[13]

Velkou výhodou Home Assistant je, že mnoho zařízení rozpozná sám automaticky. Stačí je pouze synchronizovat. Zařízení, která Home Assistant sám nerozpozná, musí být přidána ručně. Automatizaci zařízení lze zapsat do souboru YAML. To se ale využívá čím dál méně.

Další možností je provedení automatizací v Pythonu. V roce 2019 vyšel doplněk v podobě Visual Studio Code v Home Assistant. Od té doby je možné provést ovládání syntaxe pomocí Visual Studia. V neposlední řadě je možnost přidat přímo v Home Assistant některou z integrací pro automatizaci jako je Node-RED. Ten má možnost propojit snadno jednotlivá zařízení prostřednictvím uzlů.[12]

3.1.1 Instalace

Home Assistant postupným vývojem získal několik verzí instalací. Častokrát to bylo pro zjednodušení práce pro uživatele. Některé verze se v dnešní době již skoro nepoužívají. Velkou výhodou je, že je multiplatformní. Může být spuštěn jak na Windows, MacOS i Linuxu. Nejčastěji využívaný systém, na kterém je Home Assistant spuštěn je ale Raspberry Pi. [8]

Home Assistant Core

Dříve se nazýval Home Assistant. Je to hlavní program napsaný v pythonu pro domácí automatizaci. Jedná se o hlavní části složené z uživatelského rozhraní, jádra, komponent atd. Bohužel v této instalaci nenajdeme "Add-on-store". Bez toho nelze přidat rozšíření pro Home Assistant, a tak se tento instalační balíček již téměř nevyužívá.

Home Assistant

Dříve se nazýval Hass.IO. Jedná se o docker. Přidává nové možnosti do uživatelského rozhraní. Přidává možnost Add-on-store. Aktualizaci celého systému a mnoho dalšího. Právě tato instalace je nejčastěji pro Home Assistant používaná.

HassOS

Jedná se o minimální distribuci linuxu, na které je možné spustit HomeAssistant. Uživatel má minimální kontrolu nad hostitelským systémem. Nenachází se zde například ani balíček správce. Přesto je HassOS často využívám na zařízeních Raspberry Pi. Je neustále zdokonalován a aktualizován.

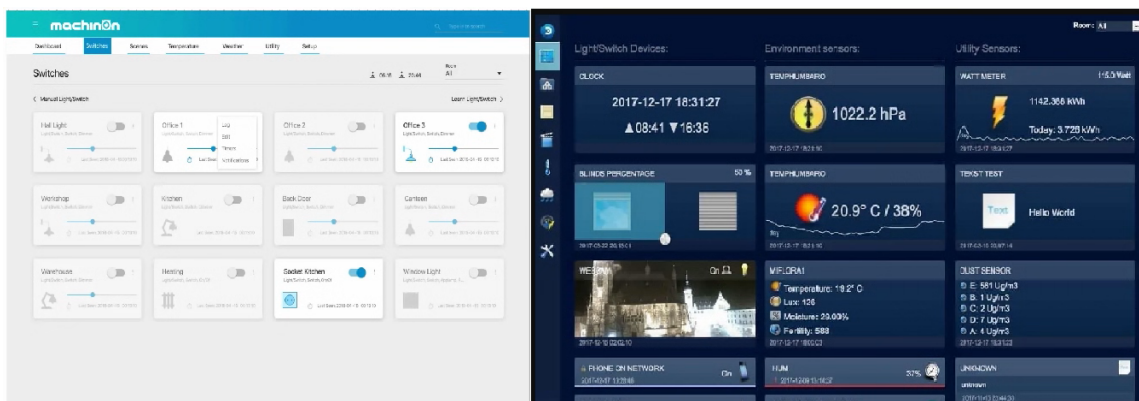
Hassbian

Již podle názvu lze odvodit, že se jedná o operační systém Raspbian. Jen je zde již předinstalovaný Home Assistant v Pythonu. Přestože systém umožňuje aktualizace, instalace dalšího softwaru, podporuje Z-wave a mnoho dalšího tak již není podporován a ani se neudržuje.

3.2 Domoticz

Domoticz vznikl v roce 2012 původně jako projekt zaměřený na RFXCOM. Velmi rychle ale získal podporu i ostatních zařízení. Projekt začal jako práce jednoho vývojáře, který na Domoticz pracuje dodnes. Za tu dobu se k němu přidalo mnoho dalších, kteří systém rozvíjeli, ale v projektu vše musí schválit původní zakladatel. Aktualizace u Domoticz jsou trošku pomalejší oproti konkurenci. Obzvláště v přidávání nových zařízení je Domoticz pozadu. Domoticz je psán v jazyce c++.

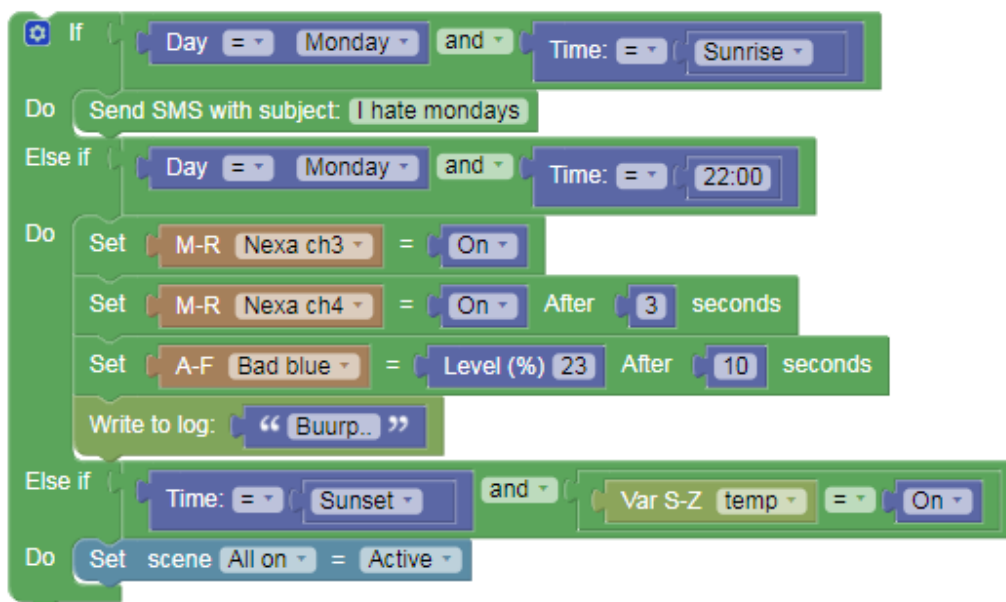
Domoticz má výchozí téma s lehce zastaralejším vzhledem. Existuje ale spousta motivů, jak si vzhled upravit a vylepšit jej. Uživatel si tedy téma může přizpůsobit více sám sobě.[12]



Obrázek 3.2: Nalevo úvodní rozhraní Domoticz. Napravo jiný motiv v Domoticz.[12]

V Domoticzu se vždy přidává hardware pomocí uživatelského rozhraní. V případě, že je zařízení podporované, tak se pouze přidá vybere se typ zařízení a pojmenuje se. Když zařízení není podporované je možnost, že bude k dispozici Python Plugin. V případě, že by nepomohl ani on, musel by se napsat skript a vytvořit fiktivní hardwarové zařízení.

K vytvoření skriptů a automatizace se nejčastěji využívá komponenta Blockly. Za pomoci této komponenty je možné vytvářet toky s několika kroky a několika podmínkami. V případě vytvoření složitějšího konceptu, by bylo nutné použít skriptování místo bloků. K tomu slouží v Domoticz programovací jazyk LUA nebo DzVentz. Všeobecně se častěji využívá v Domoticz DzVentz. Ten je prakticky stejný jako Lua, ale nabízí několik rozšíření, a tak jsou v něm složitější skripty mnohem menší a čitelnější.[12]



Obrázek 3.3: Blockly v Domoticz.[12]

3.2.1 Instalace

V minulosti měl Domoticz také více obrazů instalací. Po několika problémech tyto systémy přestal podporovat. V současné době lze Domoticz stáhnout pomocí jediného příkazu. Nejprve ale musí být nainstalovaný Linux. K dispozici jsou také instalační soubory pro Windows a MacOSx[12]

3.3 OpenHAB

OpenHab je z výše uvedených systémů nejstarší. První vydání bylo již v roce 2010. Zatím poslední stabilní verze byla vydána v lednu 2022. OpenHab je napsaný v Javě. Na vývoji se opětovně podílí velká část vývojářů. Ovšem schvalovací proces nových aktualizací a přidávání nových zařízení trvá OpenHab velmi dlouho. Je pro něj důležitá stabilita zařízení.[20]

Uživatelské rozhraní v OpenHab se nazývá HABPanel. Je standardně nainstalované v každém z počátečních instalačních balíčků. Poté je na každém uživateli, jaké si zvolí dashboardy. Je pouze na uživateli, jak si dashboardy pojmenuje. Po prvním spuštění je na každém dashboardu uvedena sekce pro dané odvětví, jako jsou například světla, kamery, dveře atd. [19]



Obrázek 3.4: HabPanel.[19]

OpenHab je pro automatizaci vybaven řadou nástrojů. Nejznámějším nástrojem je Xtend. Xtend je výkonný skriptovací jazyk, používající mnoho struktur a funkcí. Bohužel dokumentace jazyka není na moc velké úrovni, a tak je poměrně složité se jazyk naučit. Mimo Xtend je v OpenHab stejně jako v Domoticz Blockly. Díky Blockly se dají vytvářet v OpenHab opravdu propracované automatizace. V neposlední řadě je součástí OpenHab i Node-RED, který není tak propracovaný jako v Home Assistant ale je možné ho taky použít. Samozřejmě k automatizaci lze použít i čistý javascript. Tato možnost dává sice větší flexibilitu, ovšem není uživatelsky přívětivá.[20]

3.3.1 Instalace

Stejně jako Home Assistant i OpenHab má více instalačních balíčků. Vždy záleží na systému, pro který se OpenHab instaluje. Velice oblíbený je openHABian, který se často využívá na mikropočítačích Raspberry Pi. Výsledek instalace je vždy stejný, a to je uživatelské rozhraní v podobě HabPanelu.

Kapitola 4

Modelovací jazyky

Modelovací jazyk je jakýkoliv umělý jazyk, který může být použitý k vyjádření informací, znalostí či systému ve struktuře. Struktura je soubor pravidel, které se používají pro interpretaci významu komponent.[27]

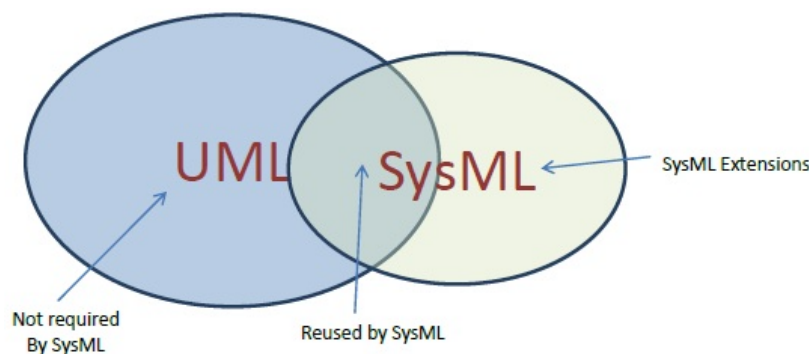
Modelovací jazyky jsou buď textové nebo grafické:

- **Grafické:** používají techniku diagramů s pojmenovanými symboly. Tyto symboly jsou spojeny čarami, které představují vazby mezi nimi.
- **Textové:** používají klíčová slova s parametry nebo výrazy. Dále fráze v přirozeném jazyku, k vytvoření výrazů interpretovatelným počítačem.

4.1 SysML

SysML je univerzální jazyk pro modelování architektury systémového inženýrství. Vznikl již v roce 2003 v rámci projektu společnosti SysML Partners.

SysML podporuje analýzu, návrh a ověřování komplexních systému včetně hardwaru, softwaru, informací, personálu, procedur a vybavení v grafické notaci. SysML je definován jako rozšíření podmnožiny jazyka UML.[24]

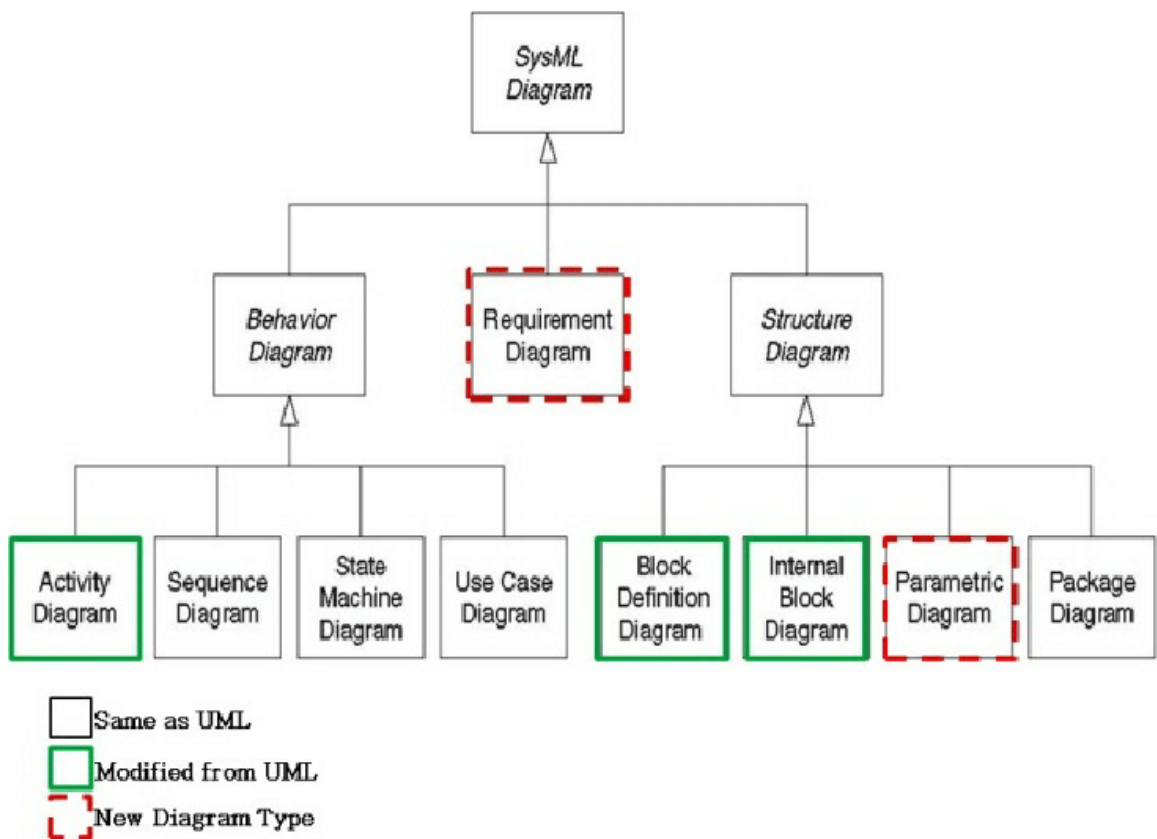


Obrázek 4.1: Vztah mezi SysML a UML.[24]

V SysML se používá několik diagramů, které reprezentují struktury systému.

- **Blokový definiční diagram:** ukazuje systémové komponenty ať už hardwarové nebo softwarové.

- **Vnitřní blokový diagram:** ukazuje obsah konkrétních bloků.
- **Diagram případu užití:** popisuje funkce na vysoké úrovni poskytované prostřednictvím interakcí mezi částmi systému.
- **Diagram aktivit SysML:** ilustruje tok dat a kontrolu mezi aktivitami.
- **Sekvenční diagram:** zobrazuje interakce mezi uživatelem, obrazovkou, objekty a entitami v systému.
- **State Machine Diagram:** identifikuje přechody, ke kterým dochází v reakci na události v systému.
- **Diagram požadavků:** identifikuje hierarchii požadavků.
- **Parametrický diagram:** identifikuje omezení hodnot systémových vlastností.



Obrázek 4.2: Vztah mezi SysML a UML.[24]

4.1.1 Modelio

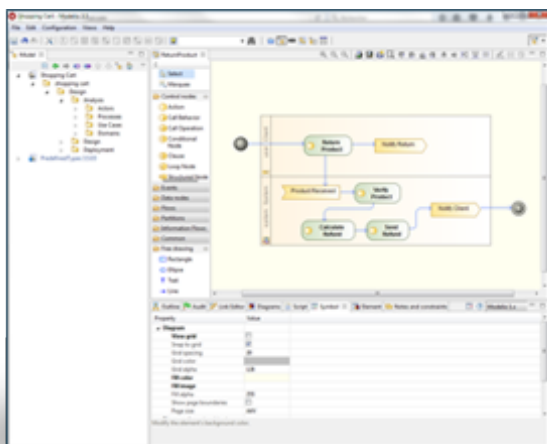
Modelio, je open-source nástroj vyvinutý stejnojmennou společností. Podporuje širokou škálu modelů a digramů. Mimo jiné poskytuje funkce pro kontrolu konzistence. Modelio je napsané v Javě až na malou část kódu, která je v c++.

Základní architektura je založena na meta-meta infrastruktuře. Aktuální metamodel je přeložen v jednotné sadě služeb a APIS¹, která poskytuje modelovací služby.

Velkou výhodou Modelia je rozšíření o kterýkoliv jazyk. Toho lze dosáhnout pomocí přidání modulu do konfigurace. V případě, že modul pro daný jazyk neexistuje je možnost vytvořit nový modul. Zvláštností Modelia je podpora jazyka Jython. Tento jazyk představuje implementaci jazyka Python v jazyce Java.

V průběhu let Modelio začalo podporovat několik standardů. Původně podporované standardy byly UML2 a BPMN2. V průběhu let se ale podpora modelia rozšířila i na XMI, MDA, SysML, TOGAF, SoaML a jazyk pro testování UML.

Ke generování kódu lze použít modul Java Designer. Tento modul podporuje generování a reverzní kódování Java, generování Javadoc a automatizaci Java.[18]



Obrázek 4.3: Vývojové prostředí Modelio.[18]

V současnosti neexistuje mnoho open-source nástrojů pro SysML. V porovnání s konkurencí je ale Modelio jeden z lepších systémů.

4.1.2 Papyrus

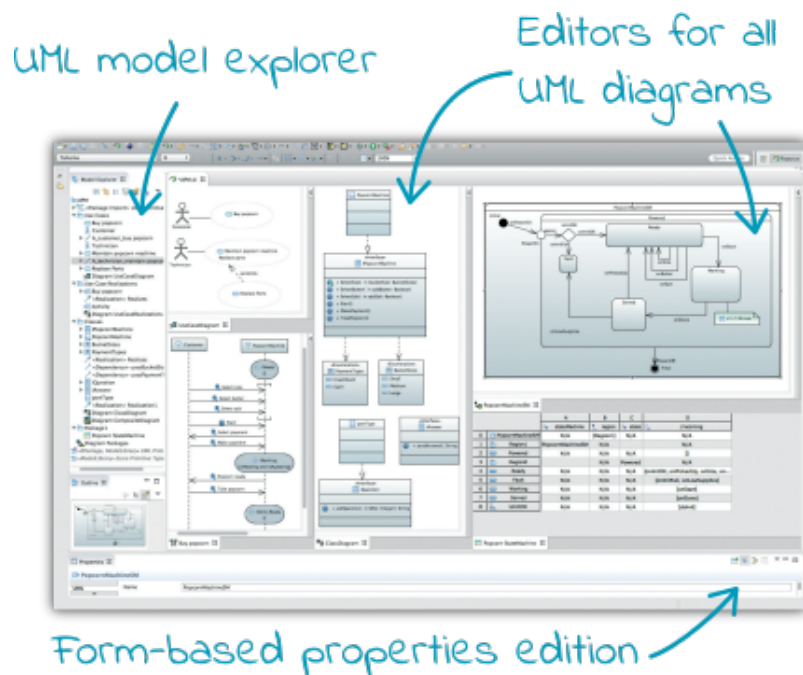
Papyrus je po Modeliu dalším velmi často používaným open-sourcem. Nejčastěji je využíván v průmyslových projektech, a je základní platformou pro několik nástrojů průmyslového modelování. Napsaný je v Javě.

Papyrus je také velmi snadno rozšiřitelný, a to díky založení na principech profilů UML. A protože grafické modelování není vždy tím nejlepším způsobem, jak určit chování spustitelných modelů, poskytuje Papyrus vydání textové notace se zvýrazněním syntaxe, dokončením a podporou obsahu.

Papyrus v současné době podporuje tři jazyky, které slouží ke generování kódu.. Jedná se o podporu C++ Java a C. Systém podporuje i reverzní generaci.

Veškeré modelovací funkce v Papyrusu jsou designově přizpůsobitelné. To zaručuje, že si uživatel může přizpůsobit konfiguraci přesně pro činnost, kterou právě dělá. Papyrus byl původně zaměřen pouze na UML a SysML. Časem svoji podporu ale rozšířil o fUML, ALF, MARTE , BPMNProfile, BMM, SMM, PSCS, PSSM, FMI a ISO/IEC.[9]

¹APIS - integrovaná sada nástrojů



Obrázek 4.4: Vývojové prostředí Papyrus.[9]

Přestože se Papyrus snaží Modelio zdatně konkurovat, tak při vyzkoušení a porovnání obou systémů Modelio vychází jako vítěz. Především v oblasti tvorby diagramů, která je pro uživatele příjemnější a přehlednější.

4.2 IEC 61499

Architektura IEC 61499 byla publikována již v roce 2005. Původně je založena na jiné normě a to IEC 61131, která je standard pro programovatelné ovladače. Norma IEC 61499 poskytuje obecný model pro distribuované systémy průmyslové automatizace zaměřené na přenositelnost, interoperabilitu opětovného použití a rekonfiguraci distribuovaných aplikací. Tento model zahrnuje procesy a komunikační sítě jako prostředí pro vložená zařízení, prostředky a aplikace. Aplikace jsou následně vytvářeny sítěmi funkčních bloků.

Funkční blok je základní model normy IEC 61499. Poskytuje rozhraní pro vstup a výstup událostí a dat. Jsou dva typy funkčních bloků a to jsou základní funkční bloky a složené funkční bloky. Složené funkční bloky se mohou skládat buď z dalších složených funkčních bloků nebo pouze ze základních funkčních bloků. Mezi nejzákladnější funkční bloky patří událostmi řízený přístup. To jsou prakticky stavové automaty.[3]

IEC 61499 poskytuje:

- Kombinaci distribuovaného programovacího jazyka a programovatelného PLC s IEC 61131-2.
- Obecný přístup modelování pro distribuované řídicí aplikace.
- Koncept funkčního bloku.
- Oddělení dat a toku událostí.

4.2.1 Eclipse 4diac

Eclipse 4diac poskytuje open-source infrastrukturu pro distribuované systémy a řízení průmyslových procesů na základě normy IEC 61499.

Prostředí běhu

4diac FORTE je malá přenositelná implementace, zaměřující se na malá integrovaná kontrolní zařízení. Podporuje online rekonfiguraci svých aplikací a úpravu funkčních bloků poskytovaných normou IEC 61449.

4diac FORTE podporuje všechny datové typy, struktury a pole, která jsou obsažena v normě 61131-3, což je předchůdce normy 61499. Velkou výhodou je škálovatelná architektura, díky které se 4diac přizpůsobuje aplikacím. Aplikace se skládají z libovolných prvků IEC 61499, které představují funkční bloky. Ve 4diacu jsou tři druhy funkčních bloků.

- BFB - základní funkční bloky
- CFB - složené funkční bloky
- SIFB - bloky servisního rozhraní

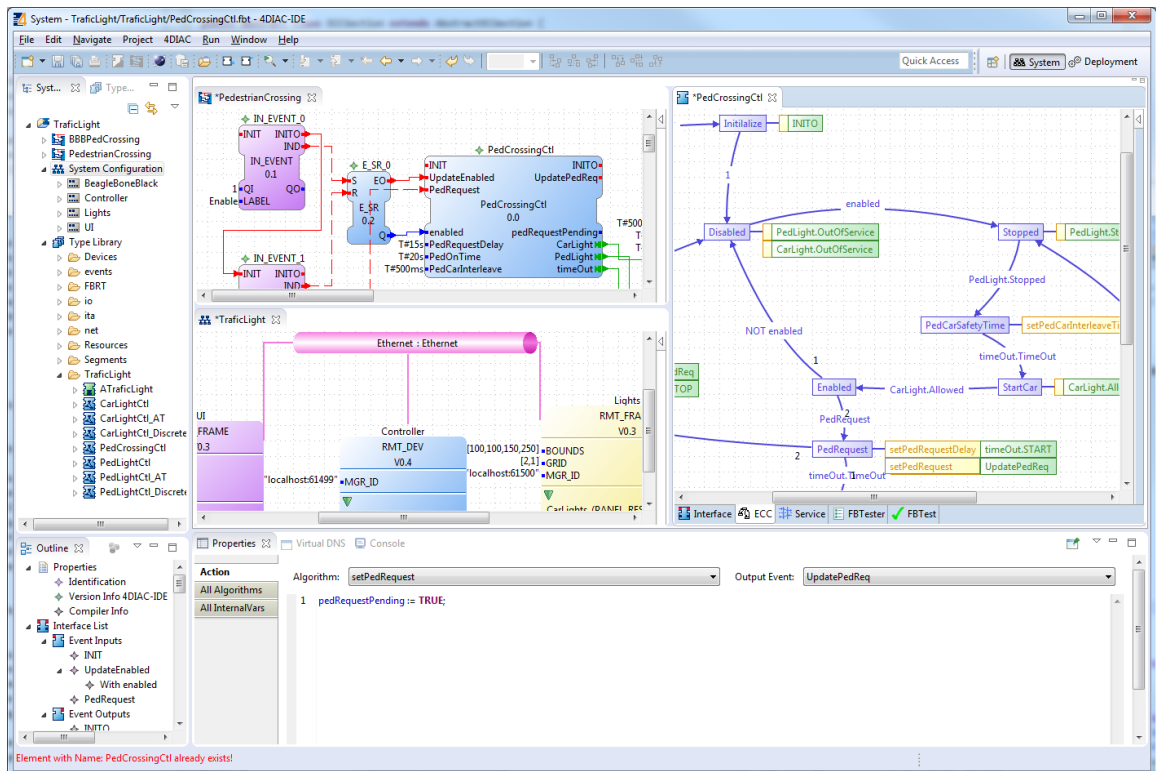
Každý z těchto funkčních bloků obsahuje rozhraní a tělo. Rozhraní poskytuje spojovací body pro přenos dat a spouštěče událostí. V těle je popsáno chování funkčního bloku.

Mimo uvedené typy bloků lze dále použít 2 seskupovací prvky. Jsou to adaptéry a pod aplikace. Adaptéry umožňují seskupit prvky rozhraní v rámci vlastního typu adaptéru. Díky tomu se sníží množství připojení. Pod aplikace seskupují funkční bloky a vytvářejí síť funkčních bloků (FBN). Ty jsou potom součástí těla u složených funkčních bloků.[10]

Vývojové prostředí

4diac IDE je založen na Eclipse². Ten umožňuje snadnou integraci dalších zásuvných modulů a poskytuje nové nebo rozšířené funkce. Systémy založené na IEC 61499 se řídí designem zaměřeným na aplikace. Což znamená, že je nejprve vytvořena aplikace celého systému. Ta se vytvoří za pomoci funkčních bloků. Po dokončení hardwarové části, lze přidat systém do konfigurace projektu.[10]

²Eclipse - vývojová platforma určená pro programování v jazyce Java, rozšiřitelná o pluginy dalších jazyků



Obrázek 4.5: Vývojové prostředí 4diac.[10]

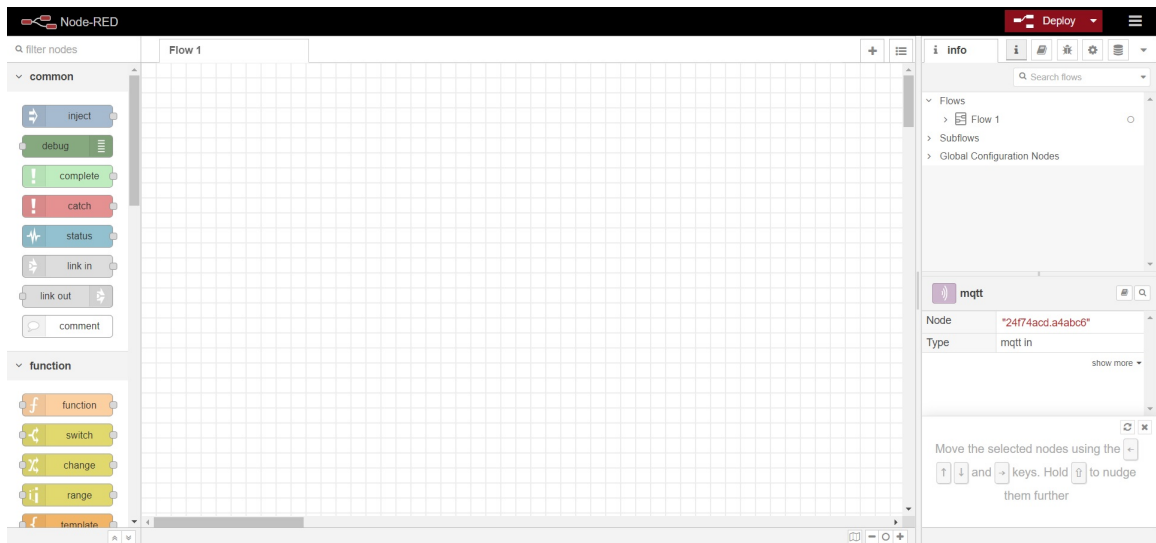
4.2.2 Node-RED

Node-RED je vizuální nástroj, který slouží k propojení zařízení internetu věcí. Může být ovšem využit i pro jiné aplikace, kde je potřeba sestavení toků. Základní myšlenkou vývoje Node-RED je vybudování sady opakovaně použitelných kódů. Ty by měli být použity k propojení hardwarových a softwarových systémů. Jedná se o engine pro zpracování událostí postavený na Node.js. Toky se v něm vytvářejí pomocí jednoduchých uzlů. Uzel představuje část kódu, která je opakovaně použitelná a logiku usnadňující vývoj aplikační cesty.[21]

Vývojové prostředí

Vývojové prostředí se v Node-RED dělí na 3 části:

- Panel uzlů
- Panel Listů
- Informační a ladící panel



Obrázek 4.6: Vývojové prostředí Node-RED.
[21]

Uživatel si může vybrat a použít libovolný uzel pro činnost, kterou potřebuje. Po zvolení uzlu se jednoduše přetáhne uzel na panel listů. Tam se může specifikovat konfigurace daného uzlu nebo ho lze propojit s jiným uzlem.

Data jsou mezi uzly předávána za pomoci JSON objektů. Ten se v Node-RED nazývá msg. Data jsou umístěna do sekce vlastnosti->obsah. Tato hodnota ale přepisuje vždy všechny předchozí hodnoty. Je tedy potřeba dbát na to, aby nedošlo k přepisování dat, která mohou být užitečná.[21]

Při porovnání 4diacu a Node-RED je možné dojít k několika závěrům. 4diac je z hlediska kvality a propracovanosti práce s bloky před Node-RED. Co se týče přehlednosti IDE jsou na tom velmi srovnatelně. Dokumentace je jak u Node-RED tak u 4diacu přehledná a usnadní vývojáři úvod do práce s nimi. Díky tomu, že Node-RED využívá pro svoji práci uzly, a ne bloky působí jednodušeji. Nelze s přesností určit, které prostředí se pro vývoj smart home hodí více. Velká výhoda je ale v tom, že open-source systémy jako je OpenHab nebo Home Assistant mají Node-RED v rámci modulu, který je možné snadno doinstalovat.

4.2.3 Open Modelica

Open Modelica je open-source modelového a simulačního prostředí určeného pro průmyslové akademické využití založeného na jazyku Modelica. Ten slouží k simulaci, optimalizaci a analýze složitých dynamických systémů. Cílem OpenModelica je vytvořit komplexní prostředí pro modelování, kompilaci a simulaci.

OpenModelica využívá několik nástrojů a aplikací pro svoji funkčnost. Každý z těchto nástrojů je určený pro něco jiného. Nejznámější a nejvyužívanější z těchto nástrojů či aplikací jsou:[4]

- OMC - OpenModelica Compiler - Kompiluje Modelica kód do jazyka C
- OMEdit - OpenModelica Connection Editor - vytváří, upravuje a simuluje modely v grafickém režimu

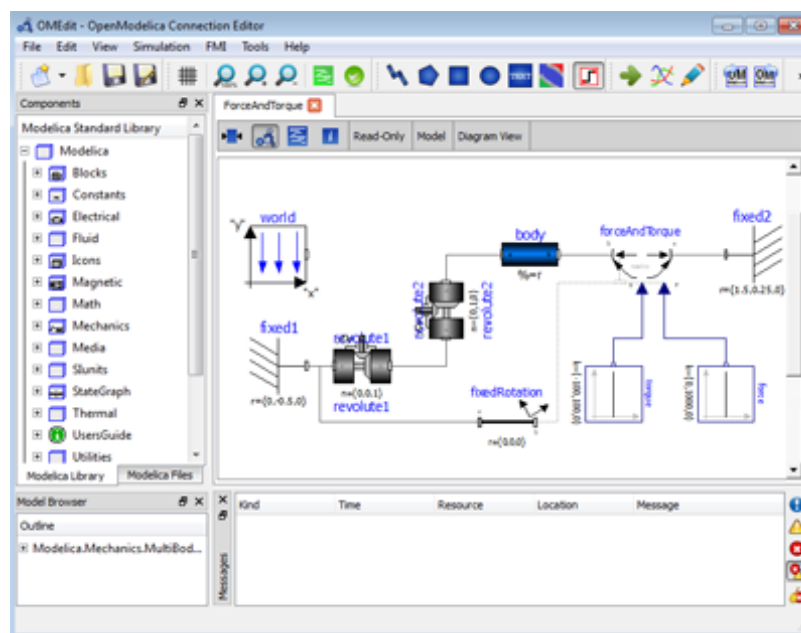
- OMShell - Interactive OpenModelica Shell - analyzuje a interpretuje výrazy pro vykreslování a simulaci.
- OMNotebook - Účel OMNotebooku je poskytnout pokročilý výukový nástroj Modelica.
- OMOptim - Optimalizace práce ve strojírenství.
- ModelicaML - Grafické modelování pro popis na událostech založené dynamiky systému
- OMEdit, MDT, OMPython, OMSimulator a mnohé další

Pro využití v oblasti chytrých domů se nejčastěji používá nástroj OMEdit.

OMEdit - OpenModelica Connection Editor

OMEdit je uživatelsky přívětivé grafické rozhraní, které poskytuje uživatelům snadné vytváření modelů, úpravu připojení, simulaci modelů a vykreslování výsledků. Rozhraní je dostatečně rozšiřitelné, aby podporovalo uživatelsky definovaná rozšíření. Modely mohou být jak textové, tak grafické.

Editor připojení bude fungovat jako front-end a OMC jako back-end. Front-end a back-end mezi sebou komunikují pomocí interaktivního rozhraní API. Díky využití OMC může být výstup modelace kód jazyka C.



Obrázek 4.7: Vývojové prostředí Modelica.

Důležitou součástí pro modelaci v OpenModelica je správně zvolená knihovna. Open Modelica využívá pro modelaci budov knihovnu Modelica Buildings library. Knihovna se primárně využívá pro flexibilní a rychlé modelování, k urychlení inovací a k nákladově efektivním velmi nízkoenergetickým systémům pro nové i stávající budovy. Obzvláště vhodná je pro:[4]

- Prototypování nových energetických systémů budov.

- Návrh a analýzu nekonvenčních energetických a řídicích systémů.
- Analýza provozu stávajících stavebních systémů
- Vývoj, specifikace, ověřování a nasazení ovládacích prvků budovy v rámci procesu návrhu založeného na modelu.
- Opětovné použití modelů během provozu pro funkční testování.

Kapitola 5

Požadavky na řídicí systém

V této kapitole bude rozebráno, jaké požadavky byly stanoveny na řízení chytré domácnosti. Jaká zařízení byla pro realizaci chytré domácnosti zvolena, a co bylo při jejich výběru nejdůležitější. V další části bude probráno, jaké požadavky jsou kladeny na softwarovou část, a který software je pro řešení využitý. Všechny tyto části byly konzultovány s uživatelem domu, který systém bude v budoucnu používat.

5.1 Požadavky na systém

Tato část bude zaměřena na požadavky, které stanovil uživatel na systém inteligentního domu. Vzhledem k tomu, že systém bude reálně využit bylo s uživatelem vše konzultováno. Hledalo se řešení, která vyhovuje nárokům na projekt a současným možnostem uživatele.

Požadavky uživatele

Celý návrh i realizace systému se odvíjela od možností uživatele. V současné době je běžné, že se chytrá zařízení zabudovávají přímo do domu. Často se přímo napojí na elektrické obvody v domě a na první pohled není ani vidět, že dům využívá chytrá zařízení. Toto je běžná záležitost při stavbě nových domů anebo při rozsáhlejších rekonstrukcích. Takto sestavený inteligentní dům je praktičtější, ale výsledná realizace vždy záleží na požadavcích uživatele a možnostech domu.

V tomto případě bylo při implementaci pro uživatele nejpodstatnější, aby se do konstrukce domů žádným způsobem nezasahovalo. Poslední rozsáhlejší rekonstrukce domu probíhala v roce 2018 a uživatel neměl zájem o zařízení, která by přímo zasahovala do stavby nemovitosti. Druhým podstatným faktorem, který hrál roli, byla celková cena projektu. Maximální částka, která pro projekt mohla být využita byla 15000 Kč.

Po dohodě s uživatelem byly požadavky pro vytvoření chytré domácnosti stanoveny v tomto pořadí:

1. **Zásah do současného stavu**
2. **Cena**
3. **Ovládání kdykoli a odkudkoli**
4. **Řízení vytápění**
5. **Rozšířitelnost**

6. Počet zařízení

5.2 Požadavky na Hardwarová zařízení

Jak bylo zmíněno dříve, uživatel si nepřál zasahovat do konstrukce domu. Z tohoto důvodu se hledalo alternativní řešení, jak chytrá zařízení zprovoznit. Elektrický zdroj byl nahrazen bateriemi. Nevýhodou tohoto řešení je, že baterie občas potřebují vyměnit. I při výběru zařízení bylo třeba dbát na to, aby baterie v zařízení vydržela dlouho a nemuseli být často měněny.

Hlavním hardwarovým požadavkem při sestavení chytré domácnosti byla regulace vytápění. Ta měla být řízena za pomoci termostatických hlavice a relé pro spínání kotle. S regulací topení je spjat i dozor nad ventilací. Jedno bez druhého je velmi neefektivním řešením.

Všechna zařízení, která byla pro práci vybrána splňují základní požadavky pro ovládání inteligentní domácnosti. Zde je seznam komponent, které jsou v domácnosti zapojeny:

- Termostatické hlavice
- Relé pro ovládání kotle
- Senzory pro kontrolu otevřených a zavřených dveří
- Pohybové senzory
- Osvětlení v kombinaci s pohybovými a dveřními senzory
- Senzor vytopení
- Alarm
- Teploměry pro kontrolu teploty termostatických hlavice.
- Vlhkoměr
- Chytrá zásuvka pro řízení a kontrolu spotřeby

5.3 Požadavky na software řídicího systému

Vhodný výběr aplikace pro řídicí systém se zakládá na přívětivosti pro samotného uživatele. Systém by měl být snadno ovladatelný, aby ho dokázal ovládat i technicky méně zdatný uživatel. Důležitou součástí je přehlednost systému. Na první pohled musí být zjevné, jaká data jsou uživateli interpretována. Ten se musí v aplikaci snadno vyznat a bez problémů procházet jednotlivé subsystemy domácnosti. V kapitole 3 jsou rozebrány systémy, které jsou z volně dostupných softwarů jedny z nejpoužívanějších.

Požadavky na přehlednost a ovladatelnost splňuje kterýkoliv z těchto softwarů. Když porovnáme všechny 3 systémy dojdeme k závěru, že Home Assistant a OpenHab jsou, jak po uživatelské, tak po administrátorské stránce před Domoticz. Přehlednější obzvláště pro začínající uživatele je ale Home Assistant, který byl také vybrán.

Připojení k Home Assistant

V rámci Home Assistant je největší problém s ovládáním kdykoli a odkudkoli. V případě, že uživatel není ochoten zaplatit za cloudové připojení, je prakticky nemožné se vzdáleně připojit. V Home Assistant je za cloudové připojení účtováno 6,5 USD měsíčně.

U každého systému tedy i u Home Assistant se může stát, že nebude fungovat správně. To může znamenat, že při pokusech o připojení se nezobrazí grafické rozhraní a uživatel tak nebude moci svoji domácnost řídit. Proto je důležitá zajistit přístup více způsoby. To vše Home Assistant, umožňuje a to pomocí doplňků Samba Share a Terminal SSH. Více o funkčnosti obou doplňku v části [6.3.2](#).

Kapitola 6

Návrh a Realizace Řešení

Tato kapitola bude zaměřena na návrh a realizaci vlastního řešení.

První část bude zaměřena na serverové řešení projektu. Jaký hardware byl vybrán pro řízení domácnosti a způsob připojení k domácí síti.

V druhé části kapitoly bude popsán hardware, který byl využitý pro řešení chytré domácnosti. Přehled senzorů, aktuátorů a koncových zařízení a zprávy, které tyto komponenty přeposílají.

V další části bude popsáno řešení v modelovacím jazyku SysML a jeho realizace v Node-RED a Home Assistant.

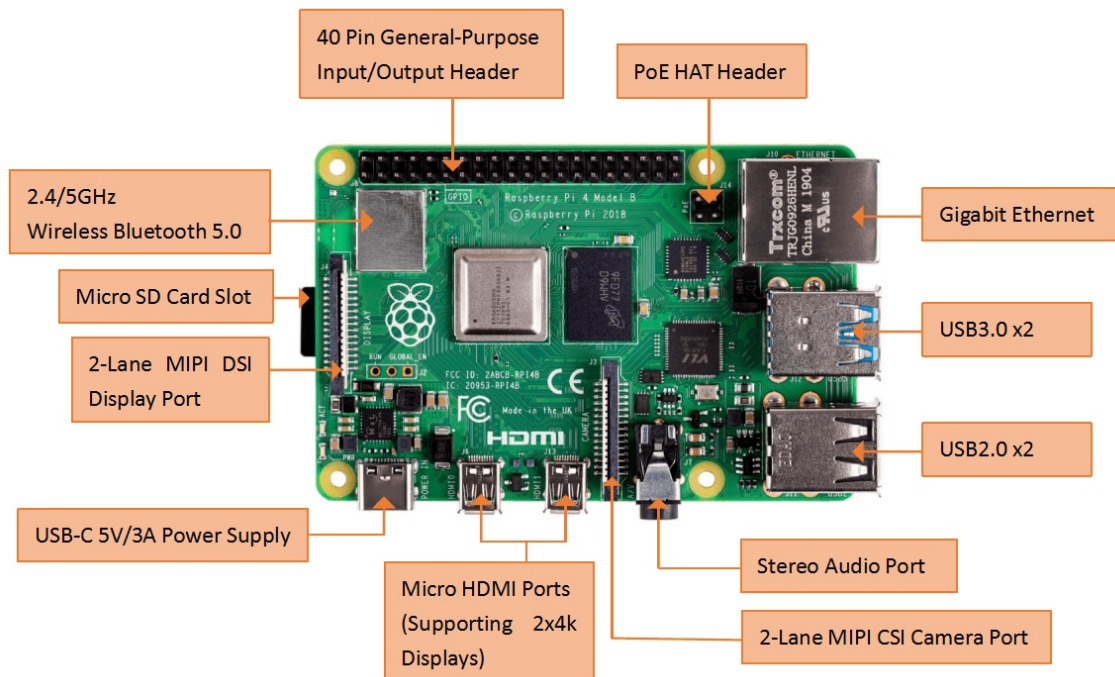
Řešení bylo realizováno ve dvouposchodovém rekonstruovaném domě, který je z části podsklepený. Ve sklepě se nachází pouze kotel, který je zapotřebí spínat. Patra jsou téměř totožná. Rozměry nemovitosti jsou 12x8 metrů.



Obrázek 6.1: Pohled z vrchu na přízemí nemovitosti.

6.1 Server

Jako serverové řešení projektu mohlo být využito několik zařízení, jak bylo rozebráno v kapitole 2.3. Z těchto variant byl zvolen jednodeskový počítač Raspberry Pi model 4. Hlavním důvodem je podstatně širší komunita oproti konkurenčním jednodeskovým počítačům a předchozí dobré zkušenosti s tímto jednodeskovým počítačem.



Obrázek 6.2: Raspberry Pi 4 (model B) [11]

Na Raspberry byl nainstalován operační systém Home Assistant. I přes doporučení zvolit připojení k síti pomocí kabelu, bylo zvoleno připojení přes wifi. V Raspberry Pi je zapojen USB flash-disk se souborem. Na souboru je uložena konfigurace a přístup k internetovému připojení. Po spuštění se Raspberry Pi samo připojí k síti wifi.

6.2 Hardware

V této části bude probrán hardware, který byl skutečně použit v rámci řešení Smart Home. Vzhledem k zaměření práce zde budou rozebrány senzory zaměřené na ventilaci, žárovky, spínače, tlačítka a především termostatické hlavice. V rámci projektu bylo využito více termostatických hlavice od různých firem. Především z důvodu porovnání kvality aktuálně dostupných termostatických hlavice na trhu.

6.2.1 Koordinátor

Koordinátor neboli kontrolér slouží k připojení různých zařízení k síti. Je schopen nahradit brány od jednotlivých výrobců jako je Xiamo, Aqara, Imax, Tuya a další. V případě této BP se bude jednat o Zigbee síť, jejíž zařízení spolu poté komunikují pomocí MQTT protokolu.

Jako koordinátor byl vybrán Sonoff Zigbee 3.0 USB Dongle Plus založený na CC2652P + CP2102N. Oproti konkurenci je velkou výhodou jeho stabilita. U konkurenčních kontrolérů se stává, že po zakoupení často nefungují, nejdou nainstalovat do aplikace a jsou s nimi další problémy. Díky přídavné anténě, kterou lze na kontrolér připevnit se výrazně zvyšuje dosah zigbee sítě. Navíc velkou výhodou je zvládnání větších zigbee sítí (100 zařízení).



Obrázek 6.3: Sonoff Zigbee 3.0 USB Dongle Plus. Foto autor

Samotný koordinátor nemusí fungovat správně. Často je to zapříčiněno zastaralým firmwarem. Ten se musí přehrát nejnovější verzí.

Existují 2 způsoby, jak nahrát do USB Dongle nový firmware:

- Flash Programmer - po připojení USB Donglu je zapotřebí stáhnout firmware. Po držením tlačítka restart vymazat starý firmware a pomocí Flash Programmeru se nahraje nový
- Python script - jednodušší varianta, při které se pouze připojí USB Dongle a Python scriptem se přehraje starý firmware za nový

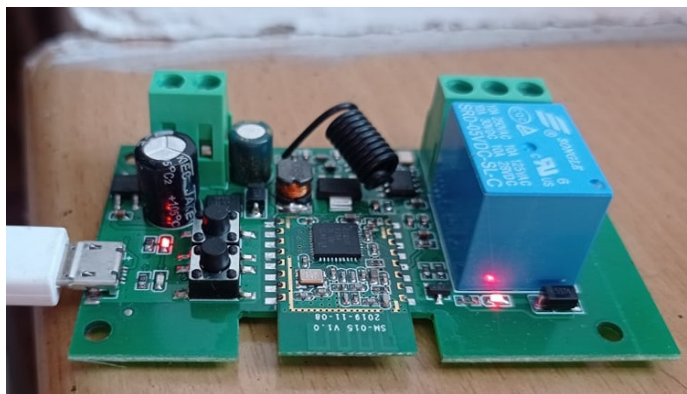
Verze firmwaru jsou vydávány pravidelně. Z tohoto důvodu se může stát, že zařízení připojená ke koordinátoru nemusí fungovat správně. To z důvodu očekávání jiného firmwaru. Je potřeba pravidelně sledovat nově vydané firmwary a případně starý firmware přehrát.[15]

6.2.2 Relé

Relé je elektromechanické zařízení sloužící ke spínání signálu. Prakticky plní funkci klasického spínače. V rámci projektu bylo důležité vybrat relé, které komunikuje přes zigbee2MQTT a bude schopno ovládat tepelné čerpadlo. Nakonec bylo zvoleno relé ZigBee 5V/7-32V + RF. Tento typ relé byl zvolen i z důvodu, že v rámci zigbee sítě slouží jako směrovač. Díky tomu zesílí a prodlouží signál do některých místností, kde může být signál koordinátoru slabý anebo nemusí dosáhnout vůbec.

Relé je možné nastavit do více funkčních módů:

- Inching mode - relé sepne ale pouze na 1s.



Obrázek 6.4: ZigBee 5V/7-32V + RF. Foto autor

- Selflocking - relé zůstane trvale sepnuté dokud nebude zaslán další signál pro rozepnutí.

Pomocí módu inching je možné relé využít nejen k spínání kotle ale i například otvírání garážových vrat. V módu inching nelze zjistit, v jakém stavu se nachází ovládané zařízení. Uvidíme pouze, že se relé na vteřinu sepnulo.

MQTT komunikace

Relé je schopno odesílat v jakém stavu se zrovna nachází (Sepnuto/Rozepnuto). Je schopné reagovat na zprávy, které jsou na něj zaslány. V případě projektu relé funguje tak, že když se topí alespoň v jedné z místností je relé sepnuto. Jakmile tato podmínka neplatí pošle se na relé MQTT zpráva s požadavkem o rozepnutí.

6.2.3 Termostatické hlavice

Způsob, jak regulovat teplotu v domácnosti je nastavení termostatických hlavice. Ty v sobě mají zpravidla zabudované teplotní senzory. Na základě okolní teploty se buď uživatel nebo samotná hlavice dle nastavení rozhodne, zda má být otevřená, zavřená, případně částečně otevřená. Toho lze dosáhnout pouze stisknutím tlačítek přidat nebo ubrat teplotu v domě, zpravidla po půl stupních.

Teplotní senzory v termostatických hlaviciích mohou mít z několika důvodů zkreslené výsledky. Jedním z nejpodstatnějších vlivů, který má vliv na senzor je umístění samotné hlavice v místnosti. V případě, že jsou radiátor a s ním i termostatická hlavice umístěni za překážkou, je možné očekávat velké zkreslení výsledků. Pravděpodobně bude na senzoru při topení teplota výrazně vyšší, než skutečně v místnosti je. Dalšími vlivy, které mohou výslednou teplotu ovlivnit jsou venkovní teplota, větrání v místnosti, sluneční záření a další. Na tyto vlivy je nezbytné přihlídnout při výběru hlavice. Možným řešením je použití nezávislého teplotního senzoru a jeho správným umístěním v místnosti. Na základě přijímaných dat z toho senzoru poté ovládat hlavici v místnosti.

Při realizaci byly využity oba způsoby získání teploty. Především z důvodu vlivů, které byly popsány výše. Výhodou použití nezávislého teplotního senzoru může být získání dat, které termostatická hlavice neposkytne. Další data získaná nezávislými teplotními senzory může být vlhkost v místnosti. I tyto data mohou pomoci v následném řízení teploty.

Pro regulaci topení v místnostech byly využity 3 různé termostatické hlavice od 2 výrobců.

Immax NEO Smart Termostatická hlavice Zigbee 3.0

Hlavice má oproti konkurenci výrazně nižší možnosti. Její použití bez aplikace je složité a nepraktické. Především přednastavení topení v jednotlivých dnech je velmi obtížné a v případě jedné změny je zapotřebí předělat všechny dny.

Po přidání do Home Assistant je možné provést základní nastavení cílové teploty a možnost zvolit si, zda má být hlavice otevřena, zavřena nebo má běžet v přednastaveném režimu. V kombinaci s Node-RED vzniká možnost nastavit si funkčnost v jednotlivých hodinách a dnech snadněji než při manuálním nastavení.



Obrázek 6.5: Termostatická hlavice Immax. Foto autor

V případě, že dojde na hlavici ke změně jakéhokoliv stavu dojde k přijmutí zprávy MQTT. Tyto zprávy mohou být i publikovány, díky čemuž probíhá komunikace pomocí Node-RED.

- Received 19:58:44
 - QoS: 0
 - away_mode: 'OFF'
 - battery_low: false
 - child_lock: UNLOCK
 - current_heating_setpoint: '27.5'
 - device_offline: 'OFF'
 - external_sensor_error: 'OFF'
 - high_temperature: 'OFF'
 - internal_sensor_error: 'OFF'
 - linkquality: 54
 - local_temperature: '22.5'
 - low_temperature: 'OFF'
 - preset: none
 - running_state: heat
 - Payload: system_mode: heat

Obrázek 6.6: Immax MQTT zpráva. Foto autor

Z MQTT zprávy je možné vyčíst čas příchodu zprávy, aktuální stav topení, kvalita připojení, teplota v místnosti, stav baterie a mnohé další.

ZigBee Termostatická Hlavice - Tuya

Od společnosti Tuya byly využity dva typy termostatických hlavic a to hlavice Hy368 a HY369.

Obě dvě hlavice mají prakticky stejnou funkčnost. Za největší rozdíl lze považovat vzhled obou hlavic. Použití bez aplikace je komplikované. V případě spojení hlavic s aplikací je ale použití snadné. Veškeré nastavení, které jde provést fyzicky, lze provést i přes aplikaci. To se může hodit především, když využijeme jeden z přednastavených režimů topení (schedule, manual, boost, complex, comfort, eco). V případě manuálního nastavení musíme využít rozdělení na pracovní dny a víkend, anebo nastavení v aplikaci.

Porovnání Tuya a Immax

Oproti termostatickým hlavicím Immax, je zda řada funkcí navíc. Za nejvíce užitečnou je možné považovat nastavení detekce otevřených oken. V případě, že výrazně poklesne teplota v místnosti například o 5 stupňů během předem nastavené doby. Hlavice rozpozná, že v místnosti jsou pravděpodobně otevřená okna, a tak se sama zavře.

Díky obsáhlejší funkčnosti termostatických hlavic jsou i MQTT zprávy podstatně delší a nabízí rozsáhlejší možnosti přijímání a odesílání dat. Je zde obsažena veškerá funkčnost, a tak je MQTT zpráva asi 4x delší než v případě hlavice Immax.



Obrázek 6.7: vlevo HY368 a vpravo HY369. Foto autor

MQTT komunikace

Hlavice v případě, že na nich dojde k jakékoliv změně odesílají MQTT zprávu, o jaké dění se jedná. Nejčastější zprávou bývá změna teploty v místnosti, ale může se jednat o cokoli jiného. Na základě této změny hlavice přijme MQTT zprávu a nastaví se buď do režimu heat nebo do režimu off. Tedy, buď aby byly otevřené a topili, případně aby byly zavřené. V tomto případě by ale mohlo dojít k zacyklení systému, protože hlavice zprávu odešle a hned zase přijme. Z tohoto důvodu je v Node-RED nastaveno, aby hlavice mohli přijmout 1 zprávu za 5 minut, čímž se zabrání zacyklení, a také neustálým změnám zapnuto, vypnuto.

6.2.4 Detektory pohybu

Obvykle jde o tzv. PIR senzory. Tyto senzory měří infračervené světlo vyzařované z objektů, které mají nějakou teplotu. To je schopen detekovat pomocí pyroelektrického senzoru. V případě, že poté kolem projde člověk nebo zvíře detektor vygeneruje signál. Samotný detektor žádné záření nevysílá.[17]

Využití

Tyto detektory mají obsáhle využití především v zabezpečení domácnosti, nejčastěji v alarmech a kamerách. Mohou být použity ale například i v rámci topného systému, kde může dojít v případě přítomnosti osoby k zahájení vytápění. Běžné použití je v kombinaci s osvětlením, přičemž se osvětlení zapne při zaznamenání pohybu.

ZigBee PIR Senzor Pohybu

Pro připojení byl využit PIR Senzor pohybu, který je možné připojit taktéž k zigbee2MQTT síti. V případě, že je zaznamenán pohyb je odeslána zpráva přes MQTT. Vzhledem k využití PIR senzoru v místnostech domu by se baterie vybíjela velmi rychle, a tak PIR senzor zaznamenává pohyb v intervalu 3 minut.

MQTT zpráva se neodesílá pouze při zaznamenání pohybu. Zpráva je odeslána i v případě neoprávněné manipulace se zařízením, což senzor dokáže taktéž rozpoznat. Je to

další bezpečnostní prvek, díky kterému se nemůže stát, že by nebyl zaznamenán pohyb a někdo by se PIR senzor pokusil odstranit. V tomto případě se zpráva odesílá ihned při pokusu o manipulaci.

6.2.5 Osvětlení

Základní komponentou v chytré domácnosti jsou světla. V rámci světel už se neřeší pouze zapnutí a vypnutí, ale je tu velké množství dalších možností co na světlech lze nastavit.

Další částí nastavení světel je jejich jas. Toho je možno využít především v noci, kdy může být lepší pro uživatele přítmí. U některých světel je dále možné využít teplotu a barvu světla. Nastavení barev v rámci klasické chytré domácnosti standardně nemá využití.

MQTT komunikace

Světla jsou koncová zařízení, která posílají MQTT zprávu v případě, že na nich dojde ke změně stavu. Ta ale nemůže nastat sama od sebe. Aby k tomu došlo je třeba dalších zařízení. Může to být výše uvedený PIR senzor. Poté je možné použít například postupné rozsvícení světel v domě tak, aby si světla po zaznamenání pohybu poslala MQTT zprávu.

Veškeré výše uvedené vlastnosti lze nastavit u světel pomocí MQTT zpráv, ale i obvykle pomocí dashboardu v Home Assistant.

6.2.6 Detektory dveří a oken

Detektory otevření dveří či oken fungují na principu vzdálení se jedné části detektoru od druhé části detektoru. Z toho plyne že obě části musí být umístěny těsně vedle sebe. Jakmile se jedna část vzdálí od druhé části, dojde k detekci otevření dveří či okna.

Nejčastějším využitím těchto detektorů je v bezpečnosti domu. Nevýhodou tohoto použití je nutnost instalace na veškerá okna a dveře objektu. V případě, že by došlo k neoprávněnému vniknutí do objektu, mohlo by se stát, že by byly využity dveře či okna bez detektorů. V tomto případě by celý bezpečnostní systém selhal.

V případě, že se použije detektor na baterie je potřeba si navíc pohlídat vybití baterií. Toto hlídání je možno zajistit případnými notifikacemi, kdyby k něčemu podobnému mělo dojít.

Detektory nemusí být použity pouze k zabezpečení domu tak jak bylo zmíněno výše. Všechny termostatické hlavice nemají funkci detekci otevřených oken. V tu chvíli nejsou hlavice schopné zareagovat na náhlou a výraznou změnu teploty. Tento problém může detektor otevřených oken vyřešit, ba dokonce tento systém vylepšuje. Samotná hlavice nemusí vždy zareagovat na změnu teploty v případě, že neklesne o určitý počet stupňů. To se u senzorů nemůže stát. V případě, že je automatizace nastavena správně, se termostatická hlavice uzavře a v místnosti se tak netopí zbytečně.

Při realizaci byly využity 2 různé typy těchto senzorů. Sensory byly od společností Tuya a IMMAX. Typ senzorů od firmy IMMAX fungoval bezproblémově. Po měsíci testování byla baterie stále 100 %. V systému je tak jejich použití snadné a uživatel se nemusí obávat neustálého vybíjení.

Druhý senzor od společnosti Tuya byl taktéž testován měsíc. Po uplynutí měsíce byla baterie na 52 %. Senzor bylo tak při realizaci velmi složité využít vzhledem k rychlosti vybíjení.

MQTT komunikace

Vzhledem k tomu, že se jedná o senzor, na něm při přijmutí MQTT zprávy nelze nic změnit. Naopak tento senzor publikuje zprávu o tom, zda je otevřeno případně zavřeno. Těchto zpráv se následně využívá pro automatizaci systému.

6.2.7 Tlačítko

Tlačítko je jednou z nejběžnějších komponent chytrých domácností. Detekuje sepnutí dvou kontaktů. Na základě tohoto sepnutí je možné domácnost bezdrátově automatizovat bez použití klasických vypínačů.



Obrázek 6.8: Použité MQTT REVOGI tlačítko. Foto autor

Běžně se tak používá v kombinaci s chytrými žárovkami případně led pásky. Dále může být použito například pro otevírání či zavírání garážových vrat nebo k propojení zvonku se sirénou. Tlačítko je častokrát využito i pro případ topení. Tato možnost je při použití nepraktická vzhledem k tomu, že se může zatopit i v místnostech, kde to není potřeba.

Zakoupené tlačítko Revogi funguje při stlačení jak má a zprávy odesílá. Při detailnějším prozkoumání byla ale objevena chyba v odesílání MQTT zpráv. Při stisknutí tlačítka se odešle několik MQTT zpráv po sobě. To může zapříčinit například blikání světel. Nebo jiné problémy v automatizaci domácnosti. Při automatizaci je k této chybě potřeba přihlídnout.

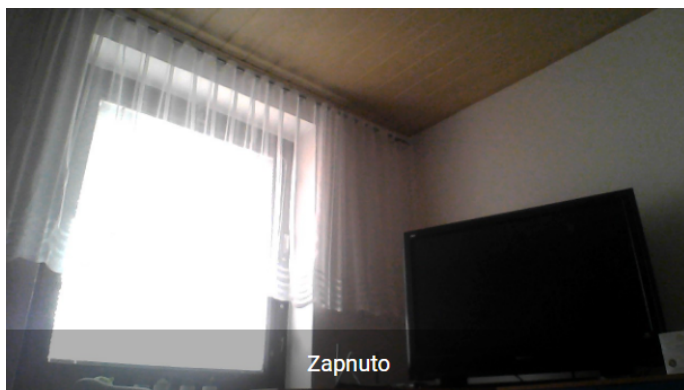
MQTT komunikace

Tlačítko komunikuje pouze přes MQTT protokol. Není na něm možné nic nastavovat vzhledem k tomu, že se jedná o aktuátor. Tlačítko při stisknutí posílá jedinou akční zprávu a tou je stisknuto. Při realizaci chytré domácnosti je použito pro spínání světel přes den. Přes noc se osvětlení spíná samo a tlačítko není potřeba. Ovšem na jeho použití noc nemá vliv.

6.2.8 Kamera

Kamery jsou používány pro zajištění dohledu a bezpečnosti nad objektem. Zpravidla je možné je použít k zobrazení aktuálního stavu. Případně se dají použít také k zobrazení historických záznamů. K tomu je zpravidla zapotřebí server, kde se data shromažďují.

V projektu byla použita kamera pouze v rámci python skriptu, který udělal dočasně MQTT kameru z PC. Ta již dále nebyla modifikována v Node-RED vzhledem k nepoužití v rámci realizace chytré domácnosti. Slouží tak pouze jako ukázka pro simulaci.[22]



Obrázek 6.9: Výstup kamery v aplikaci Home Assistant. Foto autor

6.3 Software

Hlavní náplní softwarové části bylo vytvoření blokových SysML diagramů (blokový definiční diagram a vnitřní blokový diagram). Ty byly vytvořeny pomocí nástroje Modelio, který byl vybrán podle porovnání v kapitole 4. Následné vytvoření systémů v Node-RED, který bude s modely v izomorfním vztahu. Celý systém je vytvořen pomocí aplikace Home Assistant, která byla vybrána na základě porovnání uvedeném v kapitole 3.

6.3.1 Návrh subsystémů

Celý systém je rozdělen do menších subsystémů. Každý z těchto subsystémů je modelován pomocí SysML v programu Modelio.

Nejpodstatnějším návrhem je regulace topení, ve kterém již je zahrnuta ventilace jednotlivých místností. Ne v každé místnosti byly senzory pro ventilaci potřeba. I toto je zahrnuto v rámci jednotlivých návrhů. Dalším subsystém, pro který byl model vytvořen je řízení osvětlení. Pro vytvoření modelů byly využity 2 typy diagramů.

Blokový definiční diagram

Blokový definiční diagram (dále BDD), vyjadřuje vztahy mezi jednotlivými bloky. Vazby, které je BDD schopen vyjádřit mezi jednotlivými bloky jsou agregace, kompozice, asociace, generalizace a závislost. Nejčastěji jsou pro návrh blokového diagramu využívány agregace, kompozice případně asociace. [5]

- **Asociace** - Základní vztah mezi bloky. Původní vztah je na obě strany stejný, přidáním šipky lze zajistit odkaz jednoho bloku na druhý.

- **Kompozice** - Jedná se o vztah mezi bloky, který neumožňuje samostatnou existenci bloku, aniž by byl připojený k nějakému celku. Kompozice je vyjádřena plným kosočtvercem na straně celku.
- **Agregace** - Na rozdíl od kompozice mohou jednotlivé bloky existovat samostatně. To jest když se rozpojí nic to pro ně neznamena a mohou fungovat samostatně. Agregace je vyjádřena prázdným kosočtvercem taktéž na straně celku.
- **Generalizace** - Neboli zobecnění lze použít pro bloky s podobnými vlastnostmi a vytvořit tak jeden blok pro skupinu.
- **Závislost** - Změna v jedné entitě způsobí změnu v druhé entitě.

V rámci projektu byly využity vazby kompozice a agregace.

Vnitřní blokový diagram

Vnitřní blokový diagram (dále IBD) zachycuje vnitřní strukturu jednotlivých bloků, jako jsou vlastnosti jednotlivých bloků a jejich porty. Porty se pokládají za speciální třídu vlastností. Za pomoci portů lze definovat, jaké interakce mezi jednotlivými bloky probíhají. Nemusí se jednat pouze o fyzický port na zařízení. Jejich cílem je vyjádřit nějaký vstup či výstup ze zařízení.[5]

V rámci projektu je hlavním záměrem při modelaci IBD vyjádření, jaké toky mezi jednotlivými bloky probíhají. V některých blocích jsou doplněny i jejich vlastnosti, především pro lepší srozumitelnost, co se v bloku při přijmutí zprávy odehrává. IBD je s BDD provázán. V IBD modelu nesmí být použité bloky, které BDD neobsahuje. Kdyby tomu tak bylo, systém by nebyl konzistentní, obsahoval by hrubé chyby a nemohl by tak být s reálným systémem ve vztahu 1:1. Velkou výhodou Modelia je, že při modelování diagramů tomuto zabráňuje. Při modelaci tak nemůže v tomto směru dojít k chybě a při vytváření IBD modelu nabízí pouze bloky, které byly vytvořeny pro BDD model.

BDD modely

Blokové definiční diagramy systému pro osvětlení a vytápění si jsou velmi podobné. Vrchní blok, u kterého se vše sbíhá, určuje systém, ve kterém se právě nacházíme. Bloky pod ním jsou s ním v agregátním vztahu, protože všechny části mohou fungovat samostatně. Pro jejich společné využití je ale potřeba sloučení do jednoho celku. Tak jak to jde vidět v blokovém schématu [6.10](#)

Většina bloků pod vytápěním či osvětlením je hardwarových. Jedná se o všechny prvky, ze kterých je systém složen. Ať už to jsou termostatické hlavice, kotel, či jiné. Výjimkou toho je blok řídicího systému.

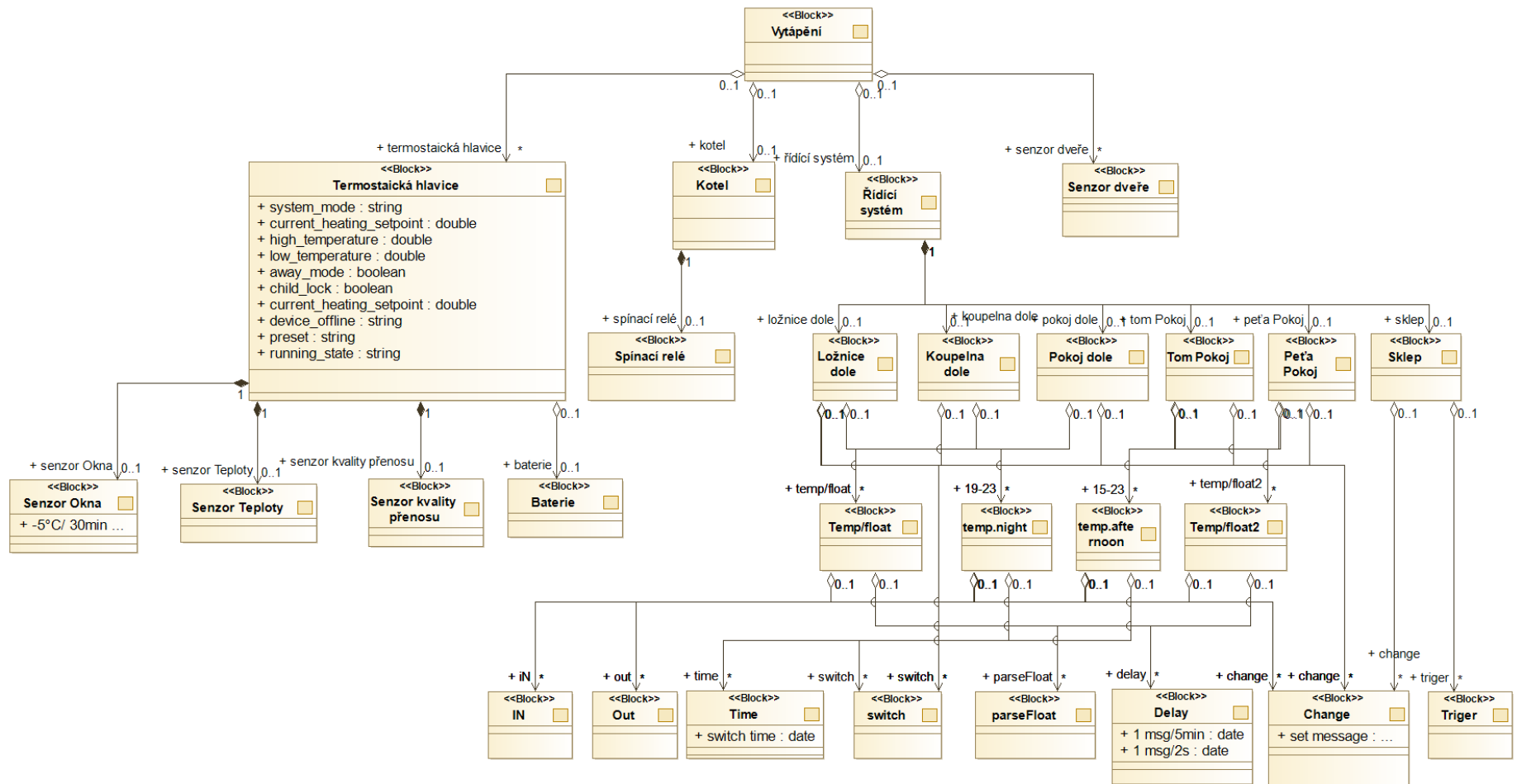
Blok řídicí systém se nachází v obou modelech jak pro vytápění, tak pro osvětlení. V obou případech se jedná o ten samý blok. Součástí řídicího systému je Home Assistant, Node-RED a mqtt broker. Bloky pro tyto části není potřeba vytvářet zvlášť. V modelu tyto části nejsou vidět a při vytváření následujících diagramů by neměli využít. Z tohoto důvodu byl vytvořen jeden blok (Řídicí systém), jehož součástí jsou všechny tyto podsystémy.

Pod blokem řídicího systému jsou jednotlivá flow nacházející se v Node-RED. V Node-RED jsou flow rozdělena po místnostech, proto názvy sklep, Tom pokoj atd. Při modelování BDD diagramů je toto rozdělení velmi důležité, protože musí být v izomorfním vztahu vzhledem k systému Node-RED. Tam je rozdělení podle flow zcela běžnou záležitostí pro

zpřehlednění a rozdělení jednotlivých částí systému. Flow jsou s řídicím systémem ve vztahu kompozitním. Z toho plyne, že bez řídicího systému by flow fungovat nemohla.

V jednotlivých flow se pak už mohou nacházet přímo bloky, které se v Node-RED dají použít pro automatizaci. Tato situace nastává u bloků switch a change, které jsou použity přímo ve flow. Případně je takto strukturalizován systém osvětlení, u něhož nebylo potřeba dalších speciální bloků. Zde jsou použity agregátní vztahy, protože jednotlivé části jsou schopny fungovat sami. U všech těchto bloků také platí, že v rámci flow jich může být použito neurčité množství.

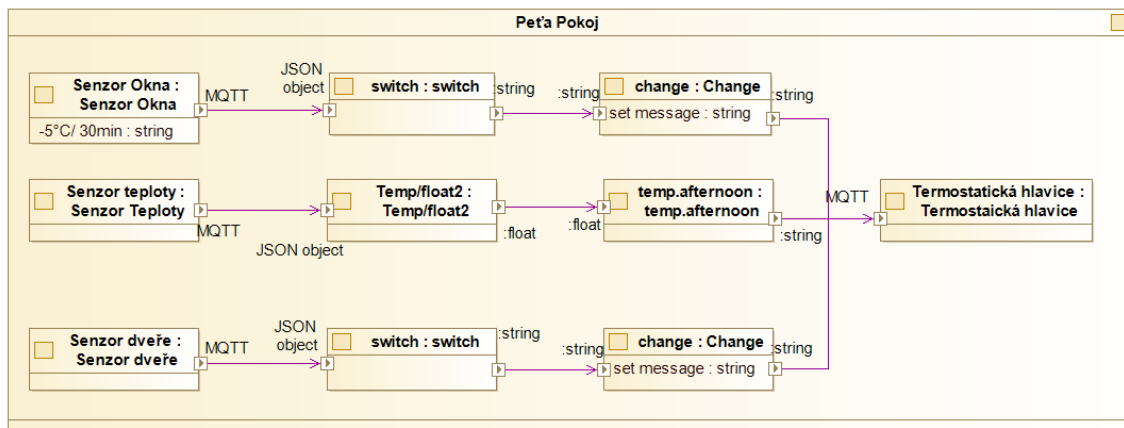
U vytápění pro zjednodušení práce bylo využito subflow. Způsob jak subflow práci zjednoduší a proč je dobré jej používat je popsán v podkapitole 6.3.2. V modelu BDD vytápění se subflow nachází hned pod jednotlivými flow, ve kterých jsou tyto subflow využity. Subflow jsou s flow v agregátním vztahu, neboť jsou schopny fungovat sami a v rámci flow jich může být opět použito neurčité množství. Pod jednotlivými subflow už se nachází bloky použitelné v node-RED. Těch může být stejně jako ve flow použito neurčité množství.



Obrázek 6.10: Blokový definiční diagram pro systém topení. Foto autor

IBD model vytápění

Jak již bylo napsáno výše, IBD model vyjadřuje vnitřní strukturu a toky v systému. V každé místnosti topný systém funguje jinak. Každá místnost obsahuje jiná zařízení, a proto je pro topný systém vytvořeno několik vnitřních diagramů. Diagramy byly vytvářeny podle struktury zapojení v jednotlivých místnostech.



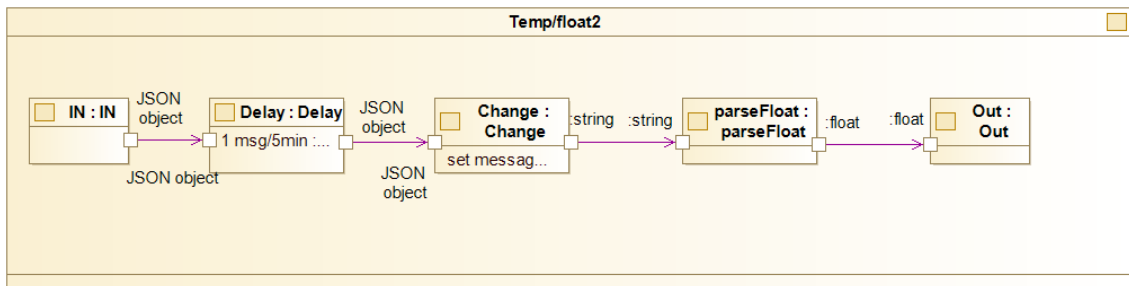
Obrázek 6.11: Vnitřní blokový diagram pro jednu z místností. Foto autor

Na obrázku 6.11 je vidět jakým způsobem probíhá komunikace v jedné z místností domu.

Vrchní část schématu obsahuje senzor oken, z něho v případě změny stavu odchází MQTT zpráva. Switch zprávu přijímá ve formátu JSON, do které lze zprávu předělat. Následně už je poslána pouze jako string, přičemž switch zkontroloval, zda je senzor aktivován. V change dochází k nastavení zprávy do podoby, kterou může termostatická hlavice přijmout. Do termostatické hlavice už zpráva opět přichází v MQTT podobě. Velmi podobně funguje i senzor u dveří, zde je jediná změna a to v bloku switch, kdy dochází ke zjištění zda je kontakt u dveří sepnut.

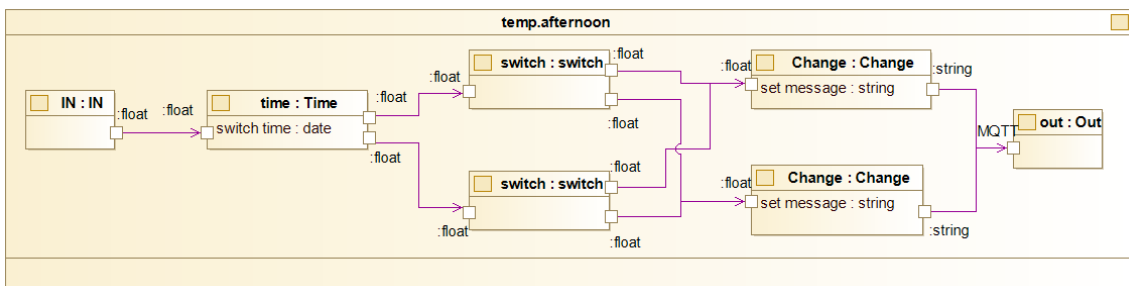
Prostřední část je zaměřena přímo na zahájení topení v místnosti. Ze senzoru teploty jsou odesílány MQTT zprávy o tom, v jakém stavu se hlavice nachází. Následující bloky Temp/float2 a temp.afternoon jsou subflow. Mají v sobě vnitřní strukturu zapojení. Do Temp/float2 přichází zpráva ve formátu JSON a dále je odeslána ve formátu float.

Zpráva je odeslána pouze v případě, že během uplynulých pěti minut nebyla poslána žádná další, jinak zpráva čeká na vstupu. Toto zapříčiňuje blok Delay. Díky tomuto bloku nedochází k zacyklení subsystému. Kdyby byl blok vynechán přicházelo by do hlavice několik zpráv za vteřinu. To z důvodu, že hlavice odesílá zprávu vždy, když se pokusí o změnu stavu. Stalo by se tedy, že by se snažila hlavice otevřít pro zatopení i přesto, že by otevřená byla. Blok change nastaví obsah zprávy pouze na teplotu v aktuální místnosti. ParseFloat tuto teplotu převede z formátu string do formátu float aby s ní bylo možné dále pracovat.



Obrázek 6.12: Vnitřní blokový diagram blok Temp/float2. Foto autor

Blok temp.afternoon zpracovává zprávu pro výstup. Uvnitř bloku probíhá rozhodovací proces, kdy a zda vůbec se má zatopit. Po vstupu do bloku rozhodne blok time podle aktuální denní doby, kterým způsobem se bude v domě topit. V případě, že je odpoledne a v domě se někdo nachází, bude místnost vytopena na předem nastavenou hodnotu pro pohodlí uživatelů. Jestli je ale jiná denní doba, než předem zvolená je sepnuta pouze udržovací teplota, aby nedošlo k poklesu teploty v místnostech na moc nízkou hodnotu. Podle toho poté dochází v blocích change k nastavení zprávy aby se hlavice buď zapnula nebo vypnula. Následně termostatická hlavice v pokoji obdrží MQTT zprávu o tom v jakém stavu se má nacházet.



Obrázek 6.13: Vnitřní blokový diagram temp/afternoon. Foto autor

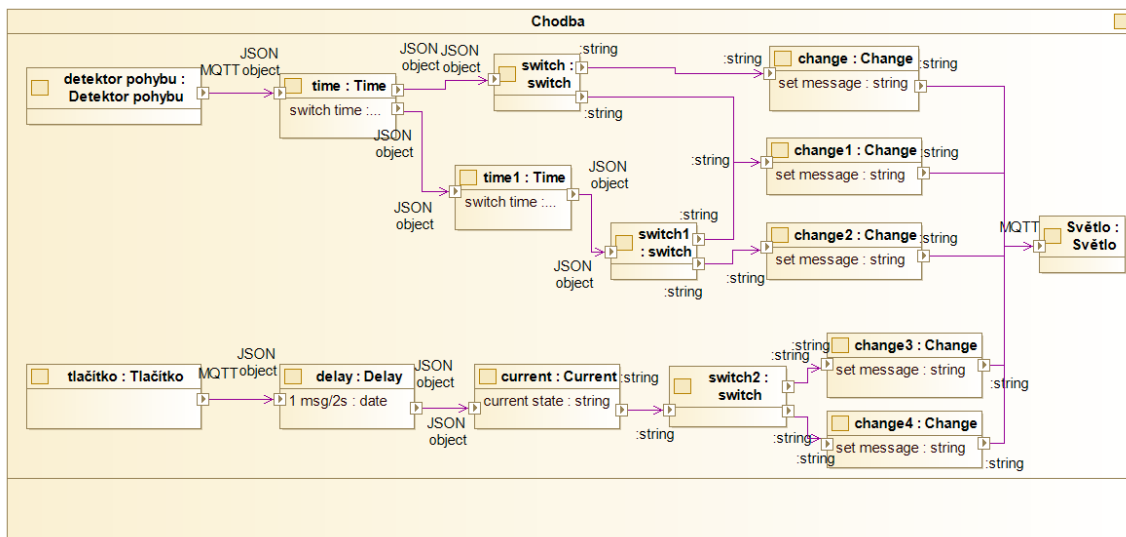
IBD model osvětlení

U systému osvětlení nebylo potřeba žádných subflow. Osvětlení je řízeno pouze v jedné části domu, a tak subflow není zapotřebí.

Světlo reaguje na detektor pohybu, který v případě změny stavu na zařízení odesílá MQTT zprávu. V bloku time dochází k rozdělení podle času. Zpráva pokračuje dále za jakýchkoliv okolností. V případě, že se čas nenachází v rozmezí od západu slunce do 23:59, dojde ke kontrole v dalším bloku time, zda čas není v rozmezí od 00:00 do východu slunce. Když ani tato podmínka neplatí a čas není v daném rozmezí, zpráva dál není poslána. Ať se čas nachází v jakémkoliv rozmezí, tak se v bloku switch zkontroluje, zda je ve zprávě posláno zaznamenání pohybu. V případě zprávy, kde pohyb zaznamenán není se v bloku change1 nastaví vypnutí světel a po přijmutí MQTT zprávy se světla vypnou. Jestliže se čas nachází v rozmezí od západu slunce do 23:59 a pohyb byl zaznamenán, v bloku change se nastaví nejen zpráva pro zapnutí světel, ale i velký jas světla. Tato zpráva se po půlnoci mění a do východu slunce se v bloku change2 zpráva nastavuje se sníženým jasnem.

Světlo je možné zapnout i mimo výše uvedenou denní dobu, a to za pomoci tlačítka. To odesílá opět MQTT zprávu. Vzhledem k chybě tlačítka, která je popsána v kapitole 6.2.7, byl využit blok delay s propouštěním 1 zprávy za 2 vteřiny. Blok current má uložený aktuální stav světel. Ten je poslán bloku switch2, ten vybere, zda se mají světla vypnout nebo zapnout podle aktuálního stavu.

Obě části spolu při zapínání a vypínání světel spolupracují. Je možné, že se tlačítkem světla rozsvítí, ale protože detektor pohybu nezaznamenal delší dobu žádný pohyb světla vypne.



Obrázek 6.14: Vnitřní blokový diagram ovládání světel. Foto autor

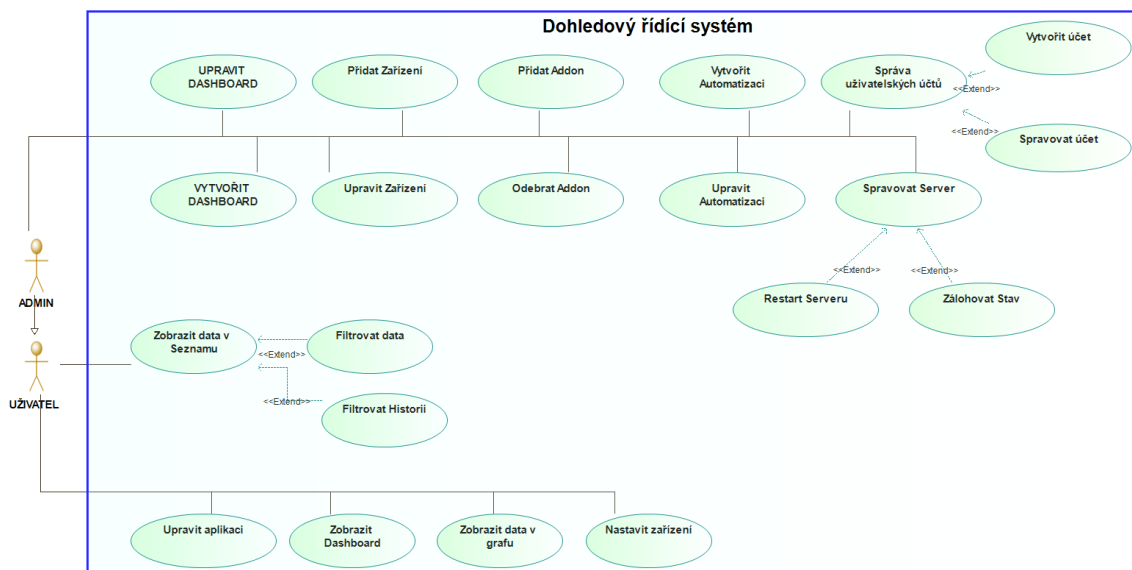
6.3.2 Realizace řešení v Home Assistant

Pro instalaci Home Assistant byl vybrán image operačního systému Home Assistant. Jak bylo popsáno v kapitole 3.1.1. Jedná se o nejčastěji používaný typ instalace. Po dokončení instalace je třeba znát nastavení routeru. V případě, že router podporuje mDNS¹ stačí do webového prohlížeče zadat `http://hassio:8123`. Jestliže router tuto podporu nemá, je třeba zadat formát adresy `http://<host>:8123` kde host je IP adresa. Ta lze zjistit v nastavení routeru.

Při prvním spuštění je potřeba provést základní nastavení systému. Vytvoření účtu, zadání systémových jednotek případně možnost nastavení, kde se bude chytrá domácnost nacházet. Po nastavení sám Home Assistant automaticky nalezne zařízení využitá v domácnosti a nabídne jejich integraci. Nenalezne všechna zařízení pouze ty, která zvládne načíst s počáteční konfigurací.

Po počátečním spuštění a instalaci je aplikace ovládána z pohledu administrátora. Ten má oproti uživateli podstatně více možností, jak aplikaci spravovat. Jedním z nich je právě vytváření nových uživatelů. Navíc dědí všechny možnosti spravování aplikace uživatelem. Všechny tyto informace je možné vidět v Use Case diagramu na obrázku 6.15.

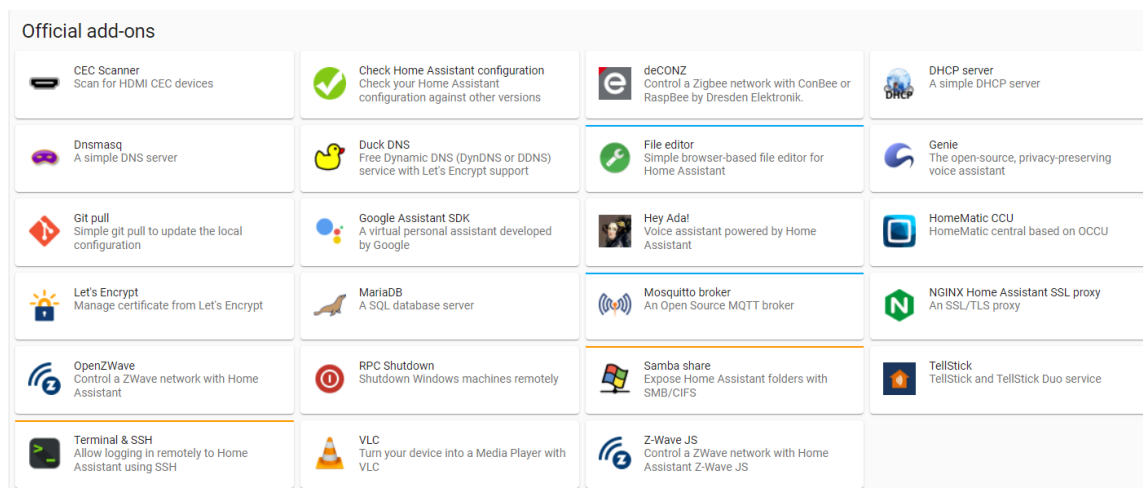
¹mDNS - multicast DNS



Obrázek 6.15: Use Case diagram Home Assistant. Foto autor

Nainstalované doplňky

Samotný Home Assistant může integrovat spoustu zařízení. To, co ho ale dělá jednou z nejpoužívanějších aplikací pro automatizaci je velké množství doplňků a integrací, které rozšiřují nejen možnosti přidání nových zařízení od různých firem. Jednou z možností je například i sledování vývoje pandemie covid-19 a na základě vývoje dělat automatizace. Dále existuje několik oficiálních doplňků nebo jsou zde doplňky, které vytvořila komunita. Existují ovšem doplňky, které nejsou uživateli nabízeny. Tyto doplňky je možné do Home Assistant přidat za pomoci repozitáře.



Obrázek 6.16: Oficiální doplňky Home Assistant. Foto autor

File editor

Nezákladnějším doplňkem je file editor. Dříve byl označován jako konfigurační editor. File editor poskytuje textový editor pro úpravu souborů běžících na počítači. Je zde možnost upravovat veškeré konfigurační soubory Home Assistant. V případě takovéto úpravy souboru typu YAML, je zde kontrola automatických chyb. V konfiguračních souborech je možnost i integrace některých nových zařízení například komunikujících přes MQTT protokol.

```
16 camera:~
17   platform: mqtt
18   topic: home/living-room/camera
19   name: "Kamera ložnice"
```

Obrázek 6.17: Příklad kamery v konfiguračním souboru. Foto autor

Samba Share

Doplňek, díky němuž je možno přistoupit ke konfiguračním souborům přímo ze zařízení Windows nebo macOS. Díky doplňku se v případě chyby v konfiguračních souborech nemůže stát, že by se do aplikace nedalo dostat přes grafické rozhraní (může nastat například po aktualizaci). Díky Samba Share lze přistoupit alespoň k opravě konfiguračních souborů, nebo v případě zálohování systému k vytvořeným zálohám, které je možné pro opravu využít.

Terminal & SSH

Nastavení serveru SSH umožní přístup ke složkám Home Assistant z libovolného SSH klienta. Dále obsahuje také nástroj příkazového řádku pro přístup k API. Smysl doplňku je stejný jako u Samba Share a to zajištění přístupu v případě selhání systému.

Mosquitto broker

Doplňek, který je nezbytný v případě využití komunikace přes MQTT protokol. Umožňuje přenos zpráv pro publikování a odběr. Jakékoliv zařízení, které komunikuje přes MQTT, je potřeba k broker připojit, aby mohlo komunikovat s ostatními zařízeními.

Struktura MQTT zprávy

Je podstatné v rámci komunikace určit jakou strukturu MQTT zpráv používat. Tuto strukturu je potřeba sjednotit pro všechny senzory, aktuátory a koncová zařízení, která spolu komunikovat mají. Díky využití Node-RED je možné veškeré odchozí MQTT zprávy formátovat do JSON object. Díky tomu mají odchozí zprávy ze zařízení stejný tvar.

Příchozí zpráva na zařízení, která dává vědět o změně stavu už může být ve formátu string. To z toho důvodu, že zařízení použitá v projektu mají topic pro nastavení zařízení. Na ten stačí poslat zprávu typu string. Zařízení si upraví zprávu do formátu MQTT a změnu provede.

Zigbee2mqtt

Existují tři způsoby, jak integrovat zigbee zařízení do Home Assistant.

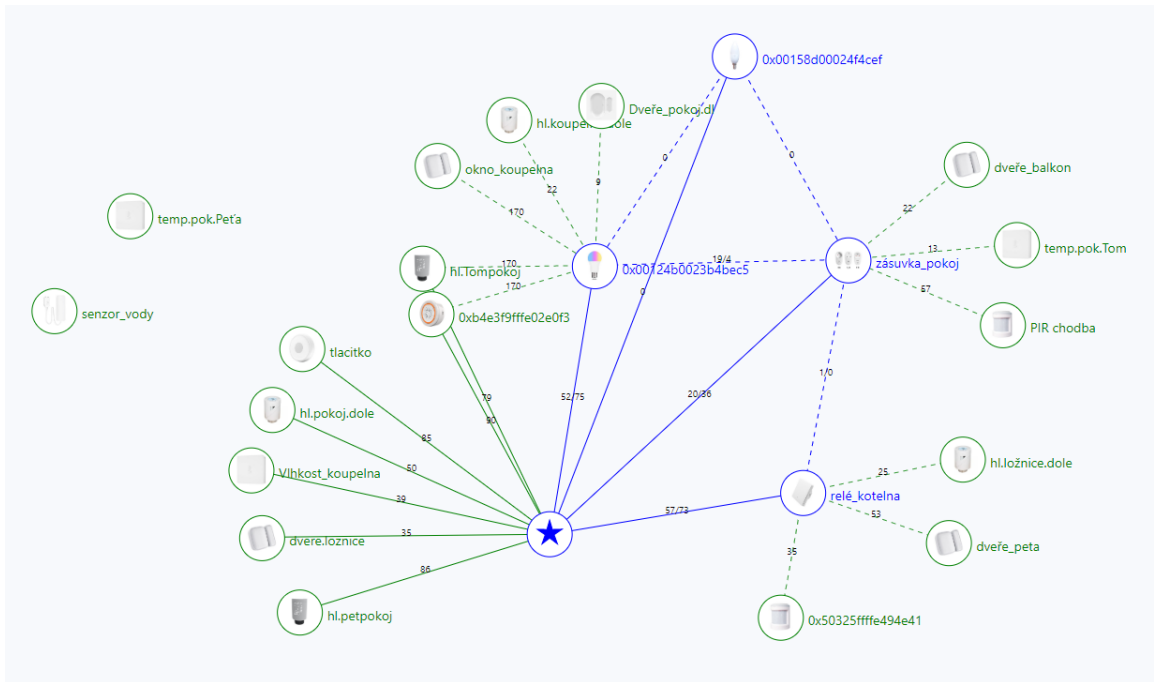
- **Zigbee2mqtt (dále Z2M)** - Schopnost ovládat zařízení přes MQTT. Vyžaduje koordinátor.
- **Zigbee Home Automation** - Umožňuje připojit běžná zařízení na bázi Zigbee přímo k Home Assistant. Nutné je použití koordinátoru.
- **deCONZ** - Ovládá zigbee síť pomocí komunikační brány ConBee II.

Pro účely projektu byl vybrán doplněk Z2M. Ten se musí pro instalaci přidat pomocí repozitáře. Po instalaci doplňku se mohou přidat zařízení, která jsou schopna komunikovat přes Z2M. Pro přidání zařízení je zapotřebí spustit párování nejen na zařízení, ale i v doplňku samotném. Jediné zařízení, které se nemusí párovat ale není vidět ani v seznamu zařízeních je koordinátor. Jeho funkce v síti je popsána v kapitole 6.2.1. Zařízení v seznamu nemusí být pouze koncová. Může se jednat i o zařízení, které v sobě mají směrovač. Díky této schopnosti jsou se na ně schopna připojit další zařízení. Často touto schopností disponují komponenty, které jsou připojeny k elektrické síti. Jako příklad se dají uvést žárovky nebo zásuvky. Po přidání jakéhokoliv zařízení do systému je možné provést základní nastavení. To standardně obsahuje stejné funkce jako by bylo zařízení nastavováno manuálně.

V doplňku Z2M je možnost zobrazení mapy zigbee sítě. Na obrázku 6.18 je vidět aktuální mapa Zigbee sítě v uživatelské domácnosti. Při detailnějším prohlédnutí mapy se dají odhalit slabé a silné stránky sítě. Každé zařízení, které je k síti připojeno nebo se k síti připojí, se snaží nalézt směrovač s nejlepší kvalitou signálu. Kvalita signálu se pohybuje v hodnotách od 0 do 255. Zařízení se připojí ke směrovači s nejlepší kvalitou spojení. Díky tomu se nedá určit přesný dosah sítě, protože jednotlivá zařízení se na sebe mohou neustále napojovat a síť rozšiřovat. V případě, že je některý ze směrovačů odstraněn zařízení se pokusí připojit na jiný směrovač.

Při testování mapy v doplňku Z2M bylo zjištěno, že mapa nefunguje vždy jak má. Na níže uvedené mapě je vidět vlevo zařízení temp.pok.Peta. Při kontrole zařízení bylo zjištěno, že je připojeno ke koordinátoru a zprávy odesílá. Na mapě přesto není připojeno k žádnému zařízení. Mapa je proto orientační a nelze na ni 100 % spoléhat.

Při delším sledování mapy v aplikaci je možno pozorovat jaká zařízení zrovna reagují a odesílají MQTT zprávy. Takovéto zařízení se na mapě krátce rozsvítí. Díky tomu je možno rychleji odhalit funkčnost zařízení a není potřeba sledovat MQTT zprávy přímo na zařízení.



Obrázek 6.18: Mapa Zigbee sítě. Foto autor

Vysvětlivky pro mapu:

- Hvězda - koordinátor
- Modrá - směrovače
- Zelená - koncová zařízení
- Plná čára - zařízení spojené s koordinátorem
- Přerušovaná čára - zařízení spojené se směrovačem

Node-RED

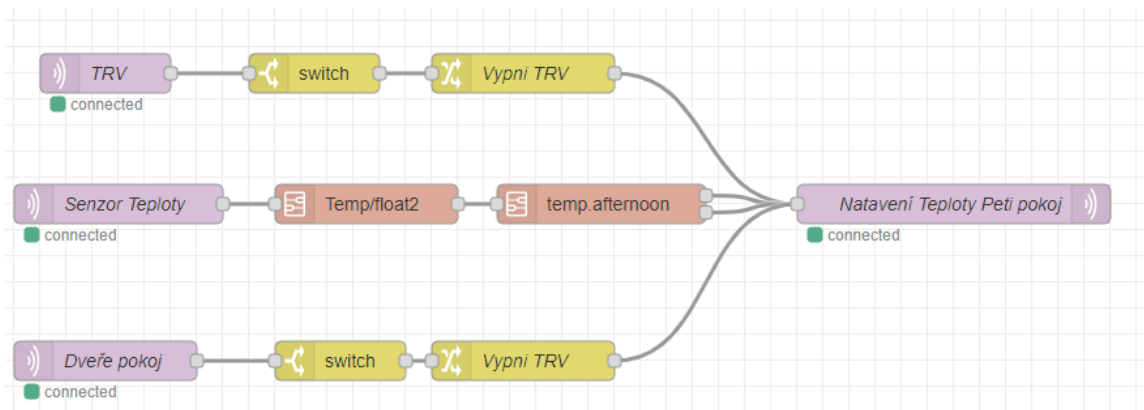
Poslední doplněk, který bylo nutné pro zprovoznění systému nainstalovat je Node-RED. Ten se pro programování používá i bez výchozí aplikace. Díky jeho doinstalování do aplikace se dají dělat automatizace, které by za normálních okolností bylo velmi obtížné naprogramovat. Node-RED je součástí komunitních doplňků Home Assistant, kde ho lze nalézt pro instalaci.

SysML a Node-RED

Celá inteligentní domácnost vytvořená pro uživatele byla automatizována pomocí Node-RED. Vývoj automatizace byl závislý na SysML diagramech. Ty byly vytvářeny současně s vývojem automatizace v Node-RED. Pro vytvoření SysML diagramu bylo nutné zjistit, jaké uzly se budou používat a jakým způsobem bude systém rozčleněn. Po získání těchto informací byly vyhotoveny modely popsané v předchozí kapitole. Podle těchto modelů pak byla vytvořena automatizace v Node-RED.

Původní modely byly vytvořeny bez subflow. Při realizaci automatizace v systému node-RED bylo zjištěno, že některé posloupnosti uzlů se často opakují. Na základě této skutečnosti byly tyto kolekce uzlů vloženy do subflow. To zapříčinilo i změnu v blokovém schématu

SysML. Do modelů se přidala nově vzniklá subflow a vznikly nové diagramy pro vnitřní zapojení bloků. Díky této úpravě byl zachován izomorfní vztah mezi Node-RED a SysML. Jediný rozdíl, který vzniká mezi modely a realitou jsou notifikační zprávy. Ty slouží uživateli případně administrátorovi pro kontrolu, zda je vše v pořádku, a tak v diagramech zahrnuté nejsou.



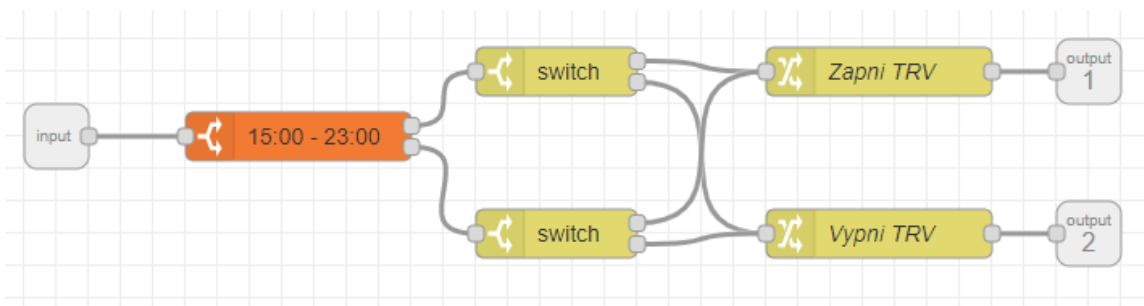
Obrázek 6.19: Automatizace v Node-RED odpovídající modelu 6.11. Foto autor

Subflow

Účelem subflow je zabalení kolekce uzlů do jednoho uzlu. Ten se poté dá využít v jakémkoliv flow. Velkou výhodou tedy je zpřehlednění a zjednodušení návrhu. Samotný Node-RED nedovoluje subflow více než 1 vstup. Výstupů může být pro subflow libovolný počet. Obzvláště na omezení 1 vstupu je třeba při použití subflow dbát.

V Subflow je možno navíc využít dalšího nastavení. Ve vlastnostech je možnost využít enviroment variables. Zde lze nastavit hodnotu na řetězec. Když bude hodnota použita v uzlu, nahradí se hodnota enviroment variable vždy před průchodem toku.

Další možností, co ve vlastnostech lze nastavit jsou vlastnosti modulu. To je možno použít k nastavení metadat o dílčích tocích. Dál zde najdeme úpravu vzhledu uzlu a případně část pro popsání co má daný uzel vykonávat.



Obrázek 6.20: Subflow odpovídající modelu 6.13. Foto autor

Dashboard

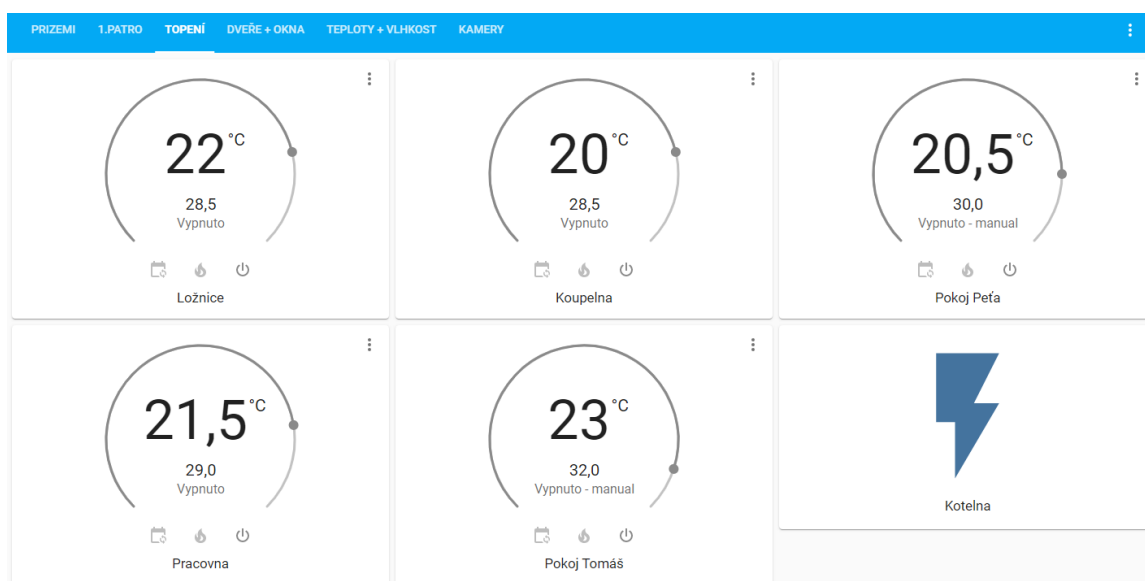
Jsou 2 způsoby jak vytvořit dashboard pro uživatele:

1. Node-RED

2. Home Assistant

Vytvoření dashboardu pomocí Node-RED by bylo vzhledem k použití aplikace Home Assistant značně nepraktické. Dashboard Home Assistant využívá, že zařízení jsou integrována přímo v něm. V tomto případě je schopen aktualizovat hodnoty v dashboardu mnohem častěji nežli Node-RED.

Původní Dashboard v Home Assistantovi se po přidání nových zařízení stává nepřehledný. Panel se sám od sebe aktualizuje ale v případě použití většího množství zařízení se v něm není možné vyznat. Proto je vhodné zvolit cestu rozdělení zařízení podle nějakých parametrů. V tomto případě bylo zvoleno rozdělení podle použitých zařízení. Ty jsou tedy rozdělena do dashboardů pro topení, senzory oken a dveří atd. Další rozdělení, které se zde dá nalézt je podle podlaží domu.



Obrázek 6.21: Dashboard pro topný systém. Foto autor

V případě, že uživatel chce rozdělení po patrech domu, je potřeba vytvořit náčrtek jednotlivých podlaží. Tento náčrtek je potřeba do Home Assistant nahrát. Jednotlivá zařízení poté lze do patra jednoduše přidat. Z tohoto dashboardu uživatel na první pohled vidí, pouze zda je zařízení aktivní nebo ne. Pro případ detailnějšího pohledu musí ikonu zařízení rozkliknout nebo použít jiný z dashboardů.



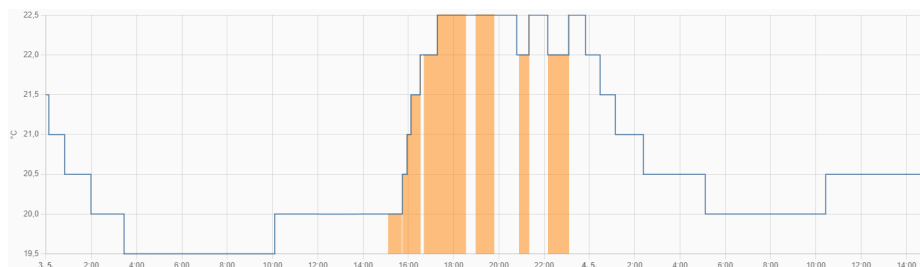
Obrázek 6.22: Vytváření dashboardu pro jedno z pater. Foto autor

Historie

Důležitou integrací pro uživatele je historie. Ta je součástí Home Assistant hned po zapnutí. Kdyby uživatel nechtěl sledovat historii jednotlivých zařízení integraci je možné oddělat v konfiguračním souboru `configuration.yaml`.

Historie je jednou z mála slabín Home Assistant. Ten je schopen historii ukládat pouze 10 dní zpětně. Jakákoliv další historie zařízení nejde zobrazit. Jedinou možností je ukládat historii do databáze.

V historii si uživatel může zobrazit data k jakékoliv entitě na zařízení. Z počátku jsou zobrazena všechna zařízení. Data je možné filtrovat přímo podle zařízení u kterého historii chceme zobrazit.



Obrázek 6.23: Graf historie topení za 24 hodin. Foto autor

Na obrázku 6.23 je názorně vidět jakým způsobem bylo topeno v jedné z místnosti domu. Modrý průběh je aktuální teplota dle termostatické hlavice v pokoji. Oranžová znázorňuje sepnutí topení. Topení se mělo sepnout v 15:00. V případě, že budou splněny podmínky o zavření dveří a oken. Bílé mezery následně znázorňují, kdy se topit přestalo, ať už z důvodu vytopení místnosti na požadovanou teplotu nebo otevření oken.

Kapitola 7

Závěr

Cílem práce bylo seznámení se s problematikou modelovacích jazyků a pomocí open source systémů navrhnout a realizovat inteligentní domácnost. Návrh celé domácnosti byl závislý nejen na zadání práce, ale i na požadavcích uživatele, pro kterého byl systém tvořen. Požadavky uživatele na domácnost se s požadavky zadání překrývaly, případně se vhodně doplňily. Povedlo se dodržet i cenové rozmezí uživatele domácnosti v rámci automatizace.

Modely jednotlivých subsystémů byly tvořeny pomocí programu Modelio, i přestože byly vyzkoušeny další programy pro vytváření modelů. Tento program mi přijde ve více ohledech nedokonalý a uživatelsky není úplně přívětivý. Spousta programů pro modelovací jazyky existuje, ale pouze pro modelovací jazyk UML. SysML je jazyk, který není zdaleka tak používán a podporován. Přesto UML v mnohých ohledech rozšiřuje a jeho použití při modelování je praktičtější obzvláště v systémech využívajících hardwarová zařízení. V projektu se při modelování muselo hodně spolupracovat s aplikací Node-RED. Bylo to zapříčiněno především velkými možnostmi Node-RED a několika uzlů, u kterých byla potřeba zjistit funkčnost pro zasazení do modelu.

V rámci této práce byly vyzkoušeny všechny zmíněné dohledové a řídicí aplikace. Home Assistant byl vybrán na základě několika kritérií, jako jsou přístup ke konfiguraci, podpora integrace zařízení a doplňků, nastavení, rozšíření aplikace a mnohé další, jako nejvhodnější aplikace pro řízení chytré domácnosti. V aplikaci by měl doinstalován doplněk Node-RED, který slouží k řízení celého systému. Node-RED je pak v izomorfním vztahu s navrženým modelem SysML.

Teoretická funkčnost vytvořeného systému byla ověřena v reálném systému domácnosti. Během realizace byla veškerá zařízení zprovozněna a byla integrována do automatizace domu. Díky zaznamenávání a nastavení jednotlivých zařízení se povedlo dosáhnout k úspoře spotřeby elektřiny. Topení se zapíná jen, když je to nutné, nebo když o to uživatel požádá. Vzhledem k omezení teploty nejsou jednotlivé místnosti zbytečně přetopené, navíc se topení vždy při větrání vypne. Podstatnou součástí je i řízení osvětlení. V zimě při odchodu z domácnosti nebylo vidět při průchodu garáží. Díky sensorům pohybu byl tento problém vyřešen. Za jediný problém tak lze považovat vhodný výběr zařízení, která ne vždy fungovala správně.

Projekt má spoustu možností rozšíření. Ať už se jedná o rozšíření pomocí SysML diagramů nebo rozšíření samotné realizace projektu. Vzhledem k možnostem dnešních inteligentních domů je možné projekt rozšířit například o kontrolu pH v bazénu domácnosti, vyřešení lepšího zabezpečení nebo pořízení inteligentních žaluzií pro lepší stínění v domě.

Literatura

- [1] *What is MQTT and How It Works* [online]. 2017 [cit. 2020-12-30]. Dostupné z: <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>.
- [2] *Historie chytré domácnosti část 1. - cesta k samostatnému domu* [online]. 2020 [cit. 2020-12-29]. Dostupné z: <https://www.immax.cz/historie-chytre-domacnosti-cast-1-cesta-k-samostatnemu-domu/>.
- [3] *IEC 61499* [online]. IEC, 2020 [cit. 2021-01-13]. Dostupné z: <https://www.iec61499.de/>.
- [4] *OpenModelica* [online]. OpenModelica, 2020 [cit. 2021-01-14]. Dostupné z: <https://openmodelica.org>.
- [5] *Overview of working with blocks* [online]. PTC, 2020 [cit. 2022-05-02]. Dostupné z: https://support.ptc.com/help/modeler/r9.0/en/index.html#page/Integrity_Modeler%2Fsysml%2F0verview_of_working_with_blocks.html%23.
- [6] *What is MQTT? A practical introduction.* [online]. inray Industriesoftware GmbH, 2020 [cit. 2020-12-30]. Dostupné z: <https://www.opc-router.com/what-is-mqtt/>.
- [7] CHYTREVPINACE. *ZigBee Koordinátor CC2531* [online]. 2020 [cit. 2022-04-29]. Dostupné z: <https://www.chytrevypinace.cz/ZigBee-Koordinator-CC2531-d201.htm>.
- [8] DAVYDOV, M. *HomeAssistant for newcomers: What it is, what is hassio, hassos, hassbian, 101 and cookies* [online]. Home assistant community, 2019 [cit. 2021-01-08]. Dostupné z: <https://community.home-assistant.io/t/homeassistant-for-newcomers-what-it-is-what-is-hassio-hassos-hassbian-101-and-cookies/123004>.
- [9] ECLIPSE. *Eclipse Papyrus Modeling environment* [online]. Eclipse Papyrus™, 2020 [cit. 2021-01-13]. Dostupné z: <https://www.eclipse.org/papyrus/index.php#technologies>.
- [10] ECLIPSE. *What is Eclipse 4diac?* [online]. Eclipse, 2020 [cit. 2021-01-13]. Dostupné z: <https://www.eclipse.org/4diac/>.
- [11] ELEKTRONICS, M. *DFRobot Raspberry Pi 4 Model BI* [online]. Mouser Electronics, 2020 [cit. 2020-12-16]. Dostupné z: <https://cz.mouser.com/new/dfrobot/dfrobot-raspberry-pi-4-model-b>.
- [12] GADGET FREAK, C. *Comparison between Domoticz and Home Assistant in 2019* [online]. gadget-freakz, 2019 [cit. 2021-01-10]. Dostupné z: <https://gadget-freakz.com/domoticz-vs-home-assistant/>.

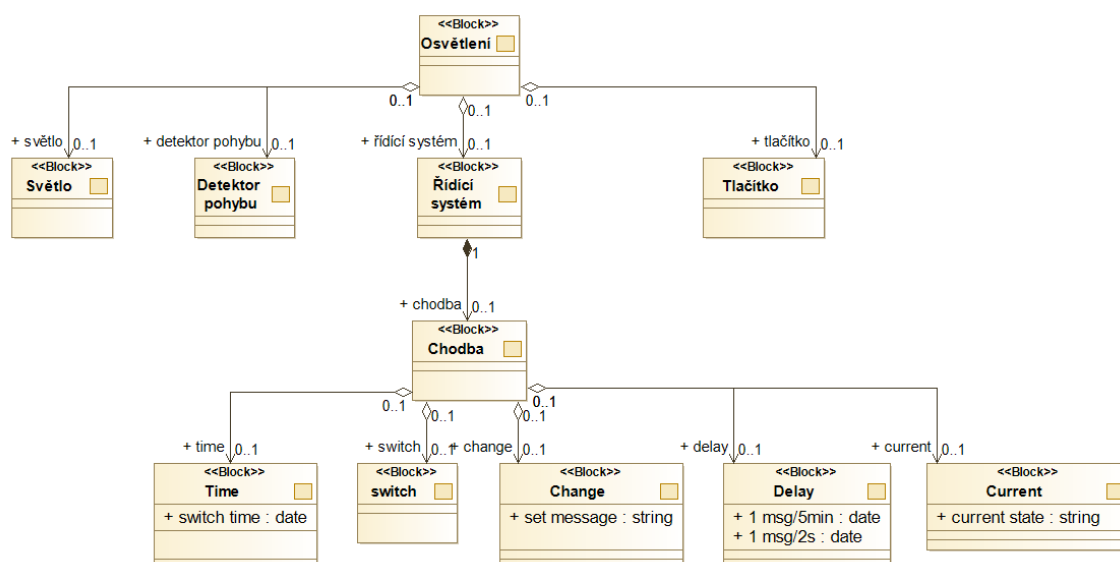
- [13] GEEK, T. *Co je Home Assistant* [online]. 2020 [cit. 2021-01-08]. Dostupné z: <https://tatageek.blog/2020/05/26/co-je-home-assistant/>.
- [14] KLAUZ, I. M. *Bezdrátové technologie pro internet věci* [online]. DPS Elektronika od A do Z, 2018 [cit. 2021-01-02]. Dostupné z: <https://www.dps-az.cz/zajimavosti/id:56560/bezdratove-technologie-pro-internet-veci>.
- [15] KOENKK. *Z-Stack coordinator firmwares* [online]. 2022 [cit. 2022-03-15]. Dostupné z: https://github.com/Koenkk/Z-Stack-firmware/tree/master/coordinator/Z-Stack_3.x.0/bin.
- [16] LABORATORIES, S. *Z-Wave Security Framework* [online]. Silicon Laboratories, 2020 [cit. 2021-01-03]. Dostupné z: <https://en.wikipedia.org/wiki/Z-Wave>.
- [17] MACHALEC, L. *PIR detektor: skvělý sluha, ale zlý pán* [online]. 2013 [cit. 2022-05-01]. Dostupné z: <https://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>.
- [18] MODELIO. *Modelio The open source modeling environment*. [online]. Modeliosoft, 2020 [cit. 2021-01-13]. Dostupné z: <https://www.modelio.org/about-modelio/features.html>.
- [19] OPENHAB. *Designing dashboard interfaces with HABPanel* [online]. openHAB, 2020 [cit. 2021-01-12]. Dostupné z: <https://www.openhab.org/docs/configuration/habpanel.html>.
- [20] PRICE, A. *Best of open source smart home: Home Assistant vs OpenHAB* [online]. 2020 [cit. 2021-01-12]. Dostupné z: https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/#Supported_Devices_and_Pairing.
- [21] SARKAR, A. *Distributed Control System Technologies- NODERED, CODESYS, 4DIAC, DOME* [online]. Institut für Automatisierungstechnik, 2015 [cit. 2021-12-02]. Dostupné z: https://www.researchgate.net/publication/290123072_Distributed_Control_System_Technologies-_NODERED_CODESYS_4DIAC_DOME.
- [22] SMEJKAL, J. *CameraToMQTT.py* [In the bachelor thesis]. 2021 [cit. 2022-03-15].
- [23] TEAM, T. H. *MQTT Essentials* [online]. The HiveMQ Team, 2015 [cit. 2020-12-30]. Dostupné z: <https://www.hivemq.com/tags/mqtt-essentials/>.
- [24] THOMAS, D. *What is SysML? The Systems Modelling Language Explained* [online]. Dunstan Thomas, 2020 [cit. 2021-01-13]. Dostupné z: <http://dthomas-software.co.uk/resources/frequently-asked-questions/what-is-sysml-2/>.
- [25] VERSUS. *Orange Pi Plus 2E vs Raspberry Pi 4 Model B* [online]. 2022 [cit. 2022-04-28]. Dostupné z: <https://versus.com/en/orange-pi-plus-2e-vs-raspberry-pi-4-model-b>.
- [26] WIKIPEDIA. *Internet Věcí* [online]. 2020 [cit. 2020-29-12]. Dostupné z: https://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD.
- [27] WIKIPEDIA. *Modeling language* [online]. Wikipedia, 2020 [cit. 2021-01-13]. Dostupné z: https://en.wikipedia.org/wiki/Modeling_language.

- [28] WIKIPEDIA. *MQTT* [online]. Wikipedia, 2020 [cit. 2020-12-30]. Dostupné z: https://en.wikipedia.org/wiki/MQTT#Message_types.
- [29] WIKIPEDIA. *Z-Wave* [online]. Wikipedia, 2020 [cit. 2021-01-02]. Dostupné z: <https://en.wikipedia.org/wiki/Z-Wave>.
- [30] WIKIPEDIA. *Home Assistant* [online]. Wikipedia, 2021 [cit. 2021-01-08]. Dostupné z: https://en.wikipedia.org/wiki/Home_Assistant.
- [31] WIKIPEDIA. *ZigBee* [online]. Wikipedia, 2021 [cit. 2021-01-08]. Dostupné z: <https://en.wikipedia.org/wiki/ZigBee>.
- [32] WIKIPEDIA. *Banana Pi* [online]. 2022 [cit. 2021-03-15]. Dostupné z: https://cs.wikipedia.org/wiki/Banana_Pi.
- [33] WIKIPEDIA. *Orange Pi* [online]. 2022 [cit. 2021-03-15]. Dostupné z: https://cs.wikipedia.org/wiki/Orange_Pi.
- [34] WIKIPEDIA. *Raspberry Pi* [online]. 2022 [cit. 2021-03-15]. Dostupné z: https://cs.wikipedia.org/wiki/Raspberry_Pi.

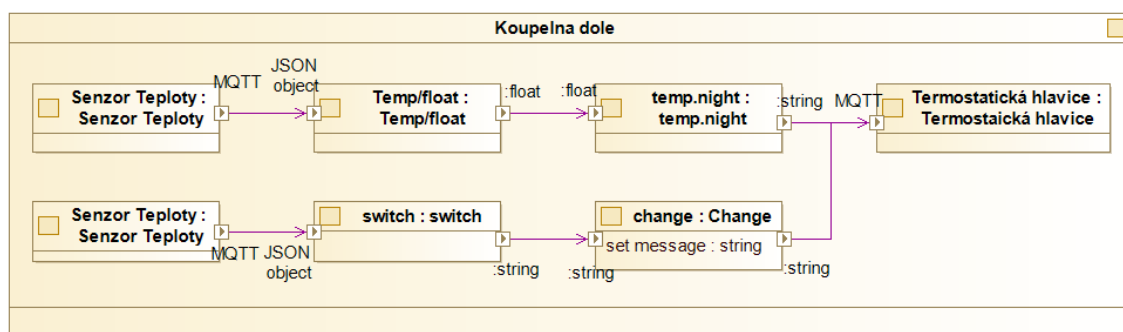
Příloha A

Vytvořené SysML diagramy

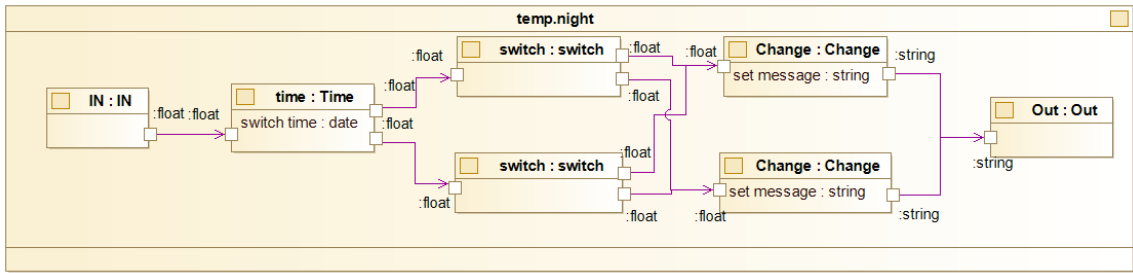
V této příloze jsou ukázány některé další SysML diagramy, které byly vytvořeny v rámci této práce. Všechny tyto diagramy jsou součástí programové přílohy.



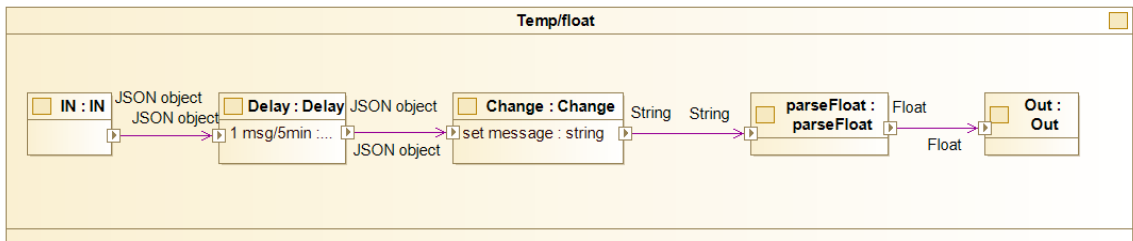
Obrázek A.1: BDD systému osvětlení



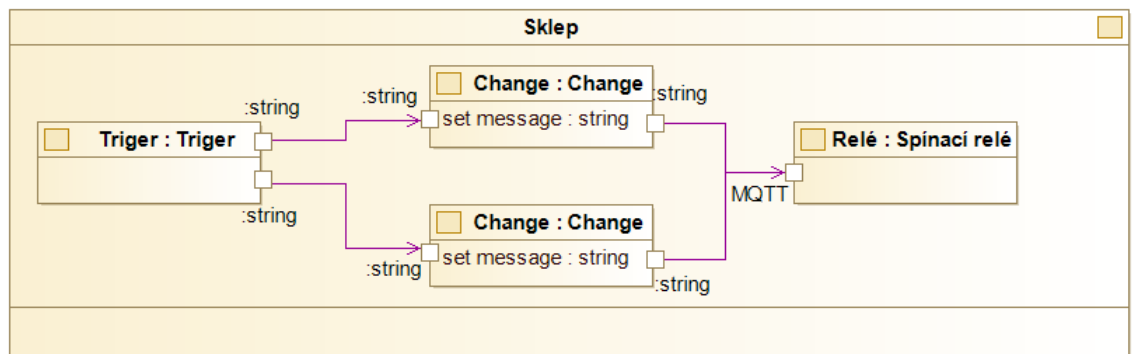
Obrázek A.2: IBD vytápění Koupelny. Foto autor



Obrázek A.3: IBD subflow temp.night. Foto autor



Obrázek A.4: IBD subflow Temp/float. Foto autor

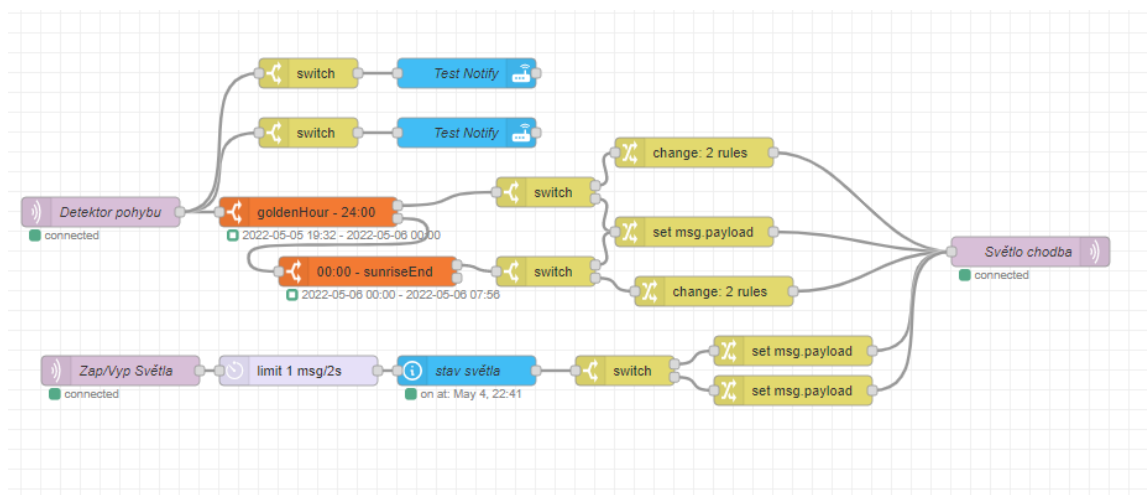


Obrázek A.5: IBD vytápění Sklep. Foto autor

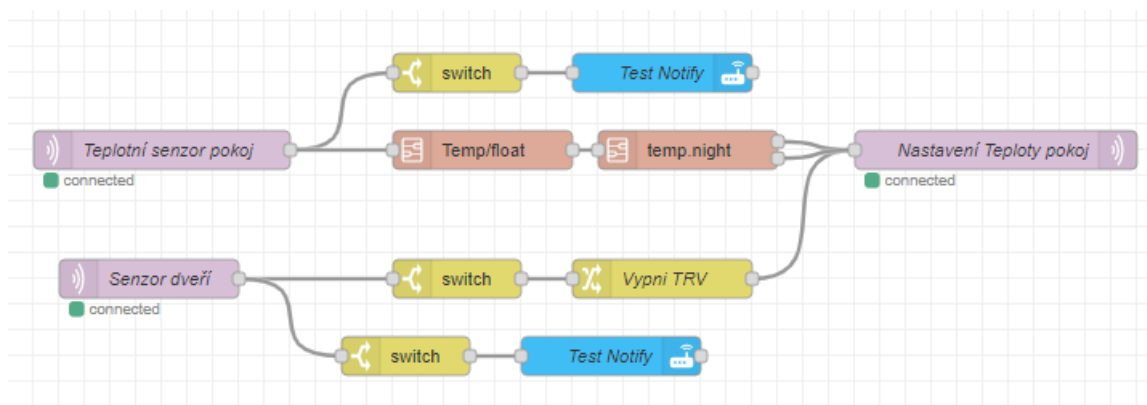
Příloha B

Node-RED modely

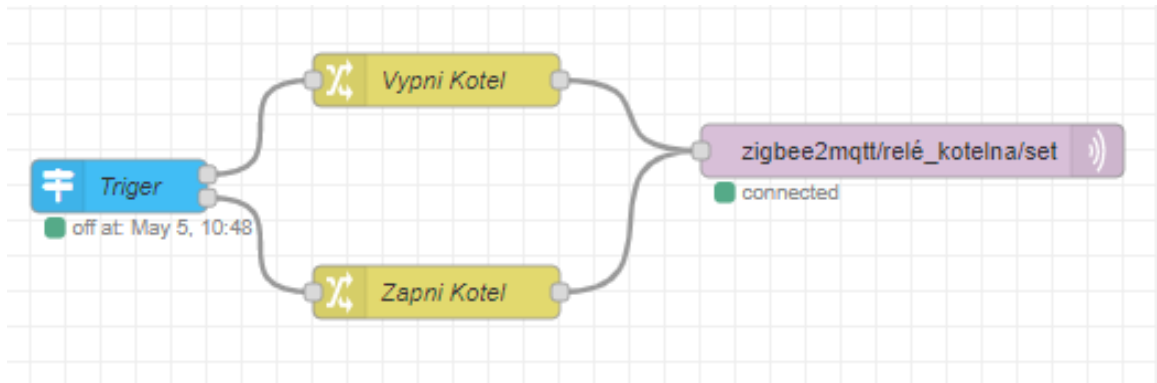
V této Příloze jsou zobrazeny další modely, které vznikly v rámci projektu. Všechny tyto modely jsou součástí programové přílohy.



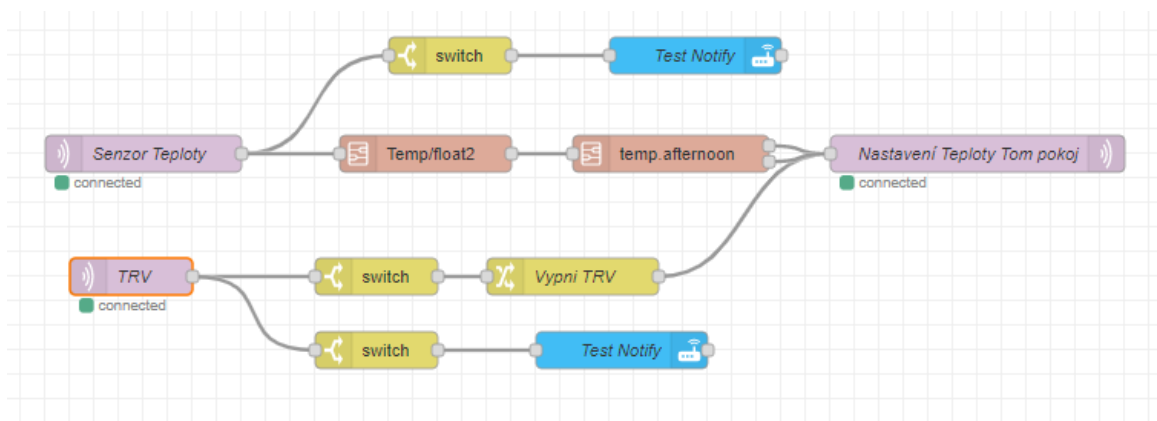
Obrázek B.1: Model Node-RED osvětlení chodba. Foto autor



Obrázek B.2: Model Node-RED topení pokoj přízemí. Foto autor



Obrázek B.3: Model Node-RED sepnutí relé. Foto autor



Obrázek B.4: Model Node-RED topení pokoj 1.patro. Foto autor

Příloha C

Obsah přiloženého paměťového média

V příloze se uvádí, co bylo přiloženo na paměťovém médiu.

- xstane44.pdf - PDF soubor s písemnou zprávou
- xstane44 složka obsahující zdrojové soubory písemné soubory
- src - složka se zdrojovými soubory programů
 - Camera - složka obsahující script ke kameře a README.md
 - SysML - složka obsahující soubor s diagramy a README.md
 - flows NR - složka obsahující soubor formátu JSON, ve kterém jsou uložena Node-RED flows a README.md
 - Coordinator - složka obsahující script ke koordinátoru a README.md