

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

REALIZACE MODELU S PRVKY CHYTRÉ DOMÁCNOSTI

REALIZATION OF A MODEL WITH A SMART HOME COMPONENTS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Dominik Fuxa

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Miroslav Jirgl, Ph.D.

BRNO 2022



Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Dominik Fuxa

ID: 220977

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Realizace modelu s prvky chytré domácnosti

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je realizace reálného modelu domácnosti pro účely demonstrace implementace standardních technologií využívajících prvky chytré domácnosti.

1. Seznamte se s konceptem chytré domácnosti a provedte rešerši dostupných technologií.
2. Navrhněte model domácnosti a realizujte jej.
3. Vyberte technologie vhodné pro demonstraci jejich řízení a implementujte je do modelu.
4. Definujte požadavky na řídicí systém a navrhněte koncept řízení vybraných technologií.
5. Implementujte řízení pro vybrané technologie.
6. Demonstrujte funkčnost Vašeho řešení.

DOPORUČENÁ LITERATURA:

ZIELONKA, Adam, Marcin WOZNIAK, et al. Smart Homes: How Much Will They Support Us? A Research on Recent Trends and Advances. IEEE Access [online]. 2021, 9(5), 26388-26419 [cit. 2021-9-7]. ISSN 2169-3536. Dostupné z: doi:10.1109/ACCESS.2021.3054575

Termín zadání: 7.2.2022

Termín odevzdání: 23.5.2022

Vedoucí práce: Ing. Miroslav Jirgl, Ph.D.

doc. Ing. Václav Jirsík, CSc.

předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Bakalářská práce na začátku seznamuje s možnými řešeními chytré domácnosti, které již byly navrženy a odzkoušeny, ať už od různých firem nebo volně dostupných open-source řešeních. Následně se práce věnuje výběru komponent do chytré domácnosti, naprogramování vlastního softwaru pro řízení chytré domácnosti a také navržení kompletního funkčního modelu.

KLÍČOVÁ SLOVA

Chytrá domácnost, Raspberry PI, Python

ABSTRACT

At the beginning, Bachelor thesis introduces possible smart home solutions that have already been designed and tested, whether from different companies or freely available open-source solutions. Subsequently, the work is devoted to selecting components for the smart home, programming its own software for managing the smart home, as well as designing a complete functional model.

KEYWORDS

Smart home, Raspberry PI, Python

FUXA, Dominik. *Realizace modelu s prvky chytré domácnosti*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2022, 63 s. Bakalářská práce. Vedoucí práce: Ing. Miroslav Jirgl, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Dominik Fuxa
VUT ID autora:	220977
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Realizace modelu s prvky chytré domácnosti

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno
.....
podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval svému vedoucímu bakalářské práce panu Ing. Miroslavu Jirglovi Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	13
1 Koncept chytré domácnosti a její možná řešení	14
1.1 Chytrá domácnost	14
1.2 Dostupná možná řešení	15
1.2.1 Komerční řešení	15
1.2.2 Open-source	15
2 Model bytové jednotky	17
2.1 Jaké technologie bude model obsahovat	17
2.2 Návrh bytového modelu	17
2.3 Vytváření fyzického modelu	19
3 Výběr použitých technologií a následná implementace	20
3.1 Výběr použitých technologií	20
3.1.1 Senzor na měření teploty v domácnosti - DHT22	20
3.1.2 Vytápění bytové jednotky	20
3.1.3 Technologie na řízení osvětlení a vytápění	22
3.2 Implementace vybraných technologií do modelu	24
3.2.1 Implementace topné fólie	24
3.2.2 Implementace osvětlení	28
3.2.3 Implementace teplotních senzorů	28
3.2.4 Zapojení svorkovnice	29
4 Definice požadavků na řídící systém a navržení konceptu řízení vybraných technologií	30
4.1 Požadavek na hlavní řídící jednotku	30
4.2 Požadavek na databázi	30
4.3 Požadavek na regulaci teploty	30
4.4 Spínání osvětlení	31
4.5 Zapojení vypínače	31
5 Implementace řízení pro vybrané technologie	32
5.1 Server na chytrou domácnost	32
5.1.1 Využití VPS popřípadě VDS	32
5.1.2 Vlastní server	33
5.1.3 Raspberry PI	33
5.2 Databázové systémy	34

5.2.1	Relační databázový model	34
5.2.2	Objektový databázový model	35
5.2.3	MySQL	35
5.2.4	PostgreSQL	35
5.3	Řízení osvětlení	36
5.3.1	Zapojení vypínače	36
5.3.2	Spínání žárovek	37
5.4	Regulace teploty v místnosti	38
5.4.1	Zapojení teplotního senzoru	38
5.4.2	Zapojení relé	38
5.4.3	Regulace teploty	39
6	Demonstrace funkčnosti	43
6.1	Test topných folií	43
6.2	Regulace teploty v místnosti	44
6.3	Test osvětlení	46
6.4	Test Telegram bota	46
6.5	Google tabulky	47
Závěr		49
Literatura		50
Seznam symbolů a zkratek		53
Seznam příloh		54
A Model domu		55
A.1	Plánek	55
B Měření vytopení místnosti pomocí topné folie		56
B.1	RPI 4B GPIO Header	56
B.2	Program použitý při měření přechodové charakteristiky	57
B.3	Výkyvy venkovní teploty při měření	58
C Programy		59
C.1	Program pro řízení osvětlení	59
C.2	Práce s databází na serveru	60
C.3	Program pro vypínače	61
C.4	Program pro teplotní senzory	62
D Obsah elektronické přílohy		63

Seznam obrázků

1.1	Ukázka chytré domácnosti	14
1.2	Ukázka overlaye z Home Assistanta	16
2.1	Hrubý nákres rozvržené plochy	18
2.2	3D model domu	18
2.3	Fyzický model domu	19
3.1	Senzor DHT22	20
3.2	Ukázka maximální teploty topné fólie pro různé výkony	21
3.3	Ukázka elektromagnetického relé	23
3.4	Model pro měření přechodové charakteristiky	24
3.5	Naměřená data pro fólii s výkonem 20W	25
3.6	Naměřená data pro fólii s výkonem 30W	25
3.7	Ukázka implementace topné fólie pod plovoucí podlahu	26
3.8	Hrubý základ modelu	27
3.9	Kompletní model	27
3.10	Model osvětlení před osazením DIN lišty	28
3.11	Kompletní osvětlení společně se svorkami	28
3.12	Umístění senzorů v modelu	29
3.13	Zapojení svorkovnice	29
5.1	Raspberry PI 4B	33
5.2	Ukázka relační databáze	34
5.3	Ukázka rozdílu relační a objektové databáze	35
5.4	Zapojení vypínače	36
5.5	Zapojení relé ovládající osvětlení	37
5.6	Zapojení senzoru DHT22	38
5.7	Zapojení relé pro jednu teplotní folii	38
5.8	Zapojení regulační smyčky pro demonstraci	39
5.9	Regulace soustavy prvního řádu	40
5.10	Regulace soustavy druhého řádu	41
6.1	Měření přechodové charakteristiky v ložnici	43
6.2	Měření přechodové charakteristiky v obýváku	44
6.3	Regulace teploty v ložnici	45
6.4	Regulace teploty v obývací místnosti	45
6.5	Zpráva poslaná botem na telegram skupinu	47
6.6	Ukázka google tabulek ovládané pythonem	48
A.1	Plánek 3D modelu	55
B.1	PINOUT u RPI 4B	56
B.2	Výkyny venkovní teploty při měření 30W folie	58

B.3 Výkyvy venkovní teploty při měření 20W folie	58
--	----

Seznam tabulek

3.1	Druhy topných folií pro model HD310G	22
3.2	Druhy topných folií pro model HD3025G	22

Seznam výpisů

5.1	Řízení osvětlení ze dvou míst	37
5.2	Regulaci teploty v místnosti	42
6.1	Obecná funkce pro posílání telegram zpráv	46
6.2	Ukázka programu pro přepisování dat v google tabulkách	48
B.1	Program použitý pro měření vytopení místnosti pro danou folii	57
C.1	Řízení osvětlení	59
C.2	Práce s databází	60
C.3	Vypínač	61
C.4	Teplotní senzory	62

Úvod

Žijeme v době, kde se naše možnosti neustále rozšiřují a stále více věcí se plně automatizuje. Jedním takovým příkladem může být vývoj v letectví kde byla rozvinuta možnost autopilota. Ale ani vybavení domácností nezůstává pozadu a v dnešní době je čím dále populárnější koncepce chytré domácnosti (anglicky Smart-Home). Celá tato koncepce spočívá v propojení aktorů pomocí serveru se senzory, popřípadě s uživatelem, a vykonávání různých úkonů, jako může být zapnutí osvětlení při příchodu do dané místnosti, sepnutí vytápění domu pokud teplota klesne pod žádanou hodnotu a mnoho dalšího.

V dnešní době mnoho firem nabízí vlastní řešení chytré domácnosti. Většinou se ale u těchto řešení střetáváme s vyšší cenou a nebo s malou kompatibilitou. Existuje i mnoho open-source projektů, které ale většinou vyžadují určité znalosti a nebo hodně strávených hodin při snaze pochopit samotný systém a jeho fungování.

Cílem této práce je návrh a implementace jednoduchého modelu chytré domácnosti ze získaných poznatků z rozboru konkurenčních řešení. Poté budou vybrány technologie, které budou poté implementovány do modelu a budou pomocí řídící jednotky řízeny.

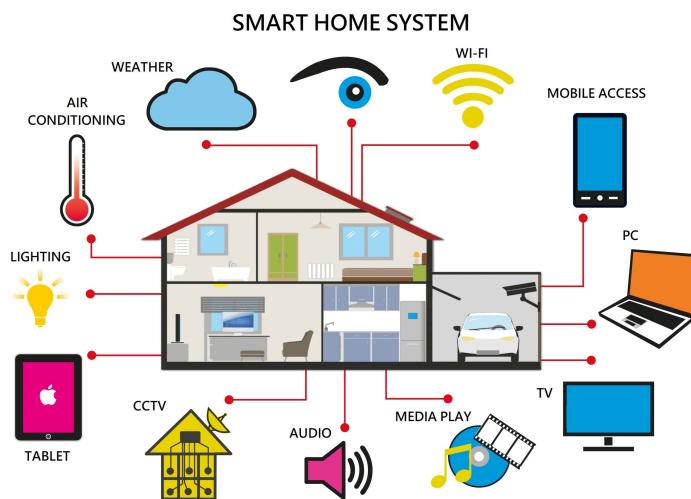
1 Koncept chytré domácnosti a její možná řešení

Tato kapitola se věnuje popisu možných řešení chytrých domácností ať už od firem, tak i opensource řešení. Bude zde probrána samotná podstata chytré domácnosti a zhodnocení možných konkurenčních řešení.

1.1 Chytrá domácnost

Historicky první chytrou domácností se stal dům v Jacksonu v Michagenu pod jménem Push-Button Manor, jenž patřil vynálezci Emilu Mathiasovi. Tento dům byl vybaven automatickým zamykáním dvěří, zatahováním rolet, zavíráním oken při dešti, atd.. Všechny tyto akce byly ovládány přímo z Mathiasovi ložnice a to konkrétně ze skříně, kde byly na stěnách připevněny různé termostaty, vypínače, hodiny a mnoho dalších věcí. [1]

V dnešní době už je většina chytrých domácností řízena jedním hlavním kontrolérem (systémem), který na všechno dohlíží a dává možnost uživateli ovládání pomocí mobilu nebo tabletu, i když není dotyčná osoba doma. Chytré domácnosti samotné nám přináší nadstandardní komfort vůči velké pořizovací ceně. Hlavním účelem chytré domácnosti proto musí být, aby dokázala maximálně šetřit náš čas a zároveň šetřit energie, aby se nám brzy investice do takové chytré domácnosti vrátila.



Obr. 1.1: Ukázka chytré domácnosti [5]

1.2 Dostupná možná řešení

1.2.1 Komerční řešení

Bohužel v tomto odvětví lze nalézt jen jedinou firmu, jelikož každé ostatní firmy řeší chytrou domácnost přímo na míru a neudávají o tom další informace, jako je například použité technologie, konektivita, cena, modulárnost řešení, možnost následného rozšíření, atd..

Firma Loxone

Firma Loxone se zaměřuje na kompletní sestavení chytré domácnosti a nabízí pro toto řešení pouze svoje omezené prvky, se kterými server komunikuje. Spočítání přibližné cenové relace pro jednu místnost (data odečtena dne 27.10.2021):

- Základní miniserver 14 000,-Kč
- Čtyři kusy LED světla 8 000,-Kč
- Dva kusy spínačů 4 000,-Kč

Celkem tedy vychází jedna místnost při zařízení pouze osvětlění na 26 000,-Kč. Všechna data jsou brána ze stránek [2]

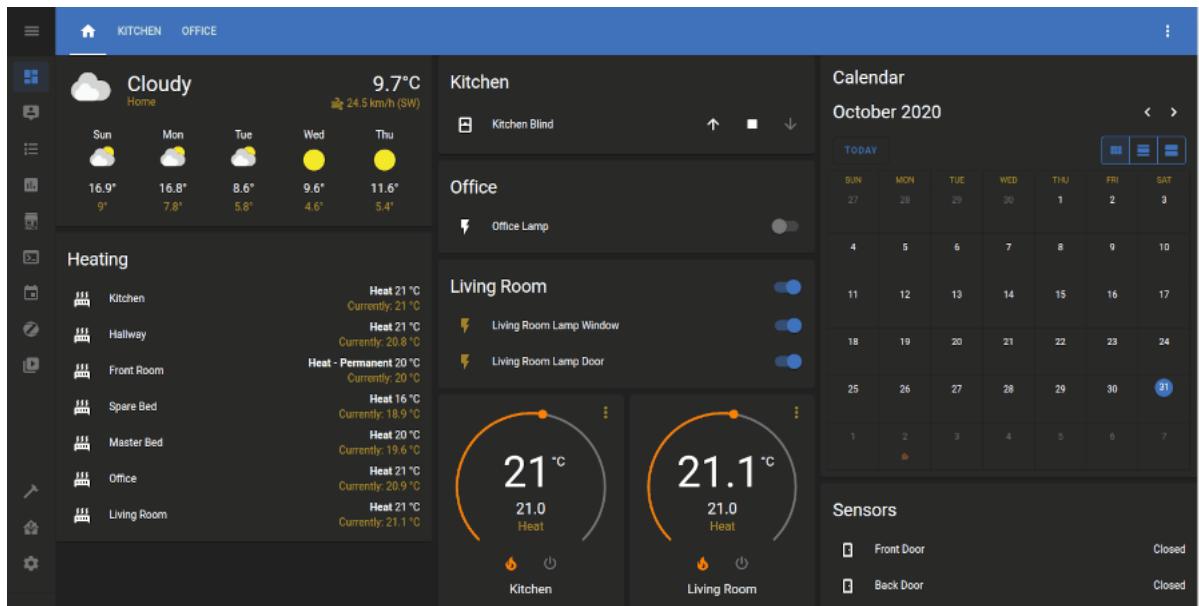
1.2.2 Open-source

Home Assistant

Jedná se o open-source systém pro domácí automatizaci. Při používání tohoto softwaru je potřeba si pořídit hardware, na kterém samotný systém poběží. Mnoho návodů doporučuje RPI, ale je možné samotný systém nainstalovat i na určitá NAS zařízení pomocí dockeru a nebo je přímo nainstalovat na virtuální stroje.

Hlavní nevýhodou je složitost tohoto systému. Díky tomu, že si autor musí vše připravit sám, sám to propojit a vytvořit si vlastní automatizaci pro daný senzor, popřípadě nastavit spuštění určitého prvku na aktivování daného senzoru. Díky tomu se stává tento systém příliš složitým pro úplné nováčky a je potřeba u prozkoumávání všech možností strávit spoustu hodin.

Aby jsme nezůstali jen u nevýhod, tak hlavní výhodou je modulovatelnost. Díky mnohým rozšířením, co software nabízí (například ESPHome) který umožní propojit ESP desky přímo s Home Assistentem a vzdáleně je programovat, čist hodnoty ze senzorů, popřípadě spouštět výstupy samotného mikrokontroléru. Obrovskou výhodou je i možnost využití cloudu, díky kterému se můžete připojit a sledovat stav svého domu i mimo lokální síť bez nutnosti veřejné IP adresy.[3]



Obr. 1.2: Ukázka overlaye z Home Assistanta [4]

Home Assistant Core

Ve všem se podobá Home Assistantu, ale hlavním rozdílem je, že zde nenaleznete add-on store, neboli nainstalování možných rozšíření. Jedná se tedy čistě o klasické jádro napsané v jazyce Python. [3]

OpenHab

OpenHab je ve srovnání s Home Asistantem postavený na programovacím jazyce Java a nabízí podobné možnosti (bohatý add-on store, ...).

Pro jednotlivé aktualizace je potřeba využívat příkazový řádek, oproti tomu Home Assistant to má vestavěné přímo ve webové aplikaci. Napříč tomu ale nabízí lepší podporu IOT protokolů jako je Bluetooth, Z-wave nebo Zigbee. [6][7]

2 Model bytové jednotky

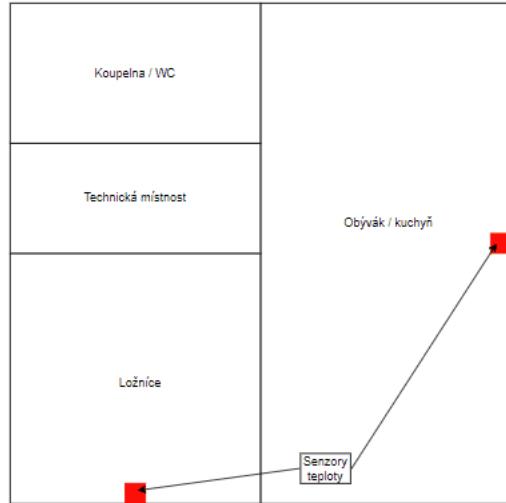
Tato kapitola je zaměřena především na obecný a teoretický návrh bytové jednotky a řešení možné její problematiky.

2.1 Jaké technologie bude model obsahovat

V modelu bude implementováno vytápění pomocí topné fólie, která bude řízena pomocí relé. Následně bude pomocí relé ovládáno i osvětlení. Pro řízení teploty bude nadále potřeba přidat do modelu i snímače teploty. Všechno bude napojené na řídící jednotku RPI 4B, která bude sloužit jako hlavní řídící jednotka a bude všechna data změřená ze senzorů ukládat do databáze běžící na serveru.

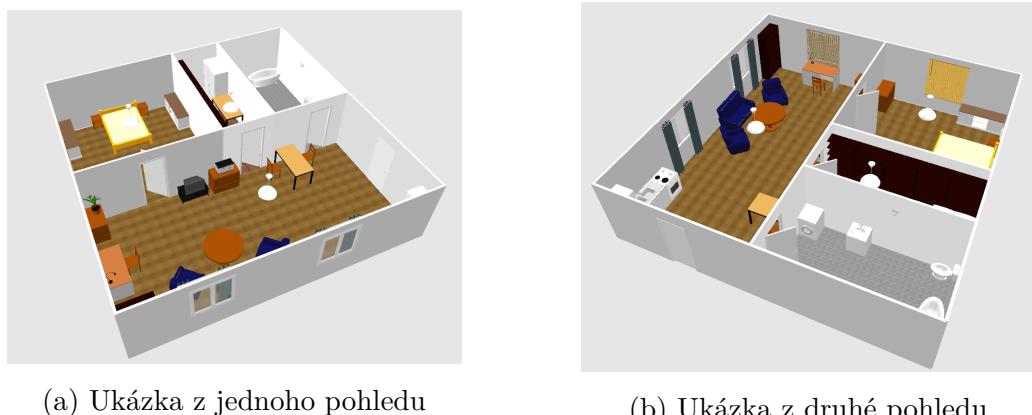
2.2 Návrh bytového modelu

Při návrhu je potřeba se zaměřit na otázku jaký model máme v plánu stavět a také zda budeme řídit místnosti centrálně a nebo každou samostatně. Při centrálním řízení je koncept směrován tak, že jsou všechny vytápěcí elementy řízeny z jedné místnosti, jedním termostatem. Pokud ale chceme mít rozdělené teploty v místnostech, například v koupelně $25^{\circ}C$ a v obývacím pokoji $20^{\circ}C$, musíme sáhnout po řešení, kdy v každé místnosti budeme měřit teplotu a ovládat každou vytápěnou jednotku zvlášť. Samozřejmě je možné tyto jednotky řídit centrálně od serveru. Tato bakalářská práce se bude zabývat problematikou řízení každé místnosti zvlášť, aby bylo možné si jednotlivě nastavovat požadované teploty. Z toho důvodu a zároveň pro minimalizování bude model realizován pomocí dvou místností ve tvaru bytové jednotky, například bytu v panelovém domě.



Obr. 2.1: Hrubý nákres rozvržené plochy

Rozvržení senzoru je koncipováno tak, aby bylo ideálně ve středu stěny místnosti a co nejdále od samotných oken. Bohužel i toto řešení, které je vidět na obrázku 2.1, má své nevýhody a tou hlavní je ovlivnění senzoru teplotou stěny. Pokud by venku bylo pod bodem mrazu, tak nám stěny, pokud nejsou dobře izolované, dokáží tuto teplotu přenášet do prostor místnosti a díky tomu může poté docházet ke zmatení senzoru, který si bude myslet, že daná místnost není stále dobře vytopená a přitom můžeme v jiné části naměřit mnohem vyšší teplotu, než kterou nastavíme.



Obr. 2.2: 3D model domu

V příloze A.1 lze vidět kompletní plánek návrhu bytu i se samotnými rozměry. Samotný nákres 3D modelu probíhal v programu Sweet Home 3D, který je zdarma stažitelný.

2.3 Vytváření fyzického modelu

Z důvodu možné veliké náročnosti tvoření reálného modelu domu a také k ušetření co možná největší plochy je základ modelu vytvořen z polystyrénových desek (viz 2.3a). Následně na tyto polystyrenové desky přijde beton, jenž tvoří úplný základ podlahy, na kterou poté přijde implementování samotné topné folie, která je popsána v kapitole 3.2.1.



(a) Pouze polystyrenové stěny

(b) Betonová podlaha

Obr. 2.3: Fyzický model domu

Jelikož se jedná o relativně malý model domu, tak samotné osvětlení bude implementováno mimo samostatný model na samostatnou desku společně s vypínači a svorkami pro uchycení kabeláže (viz. kapitola 3.2.2).

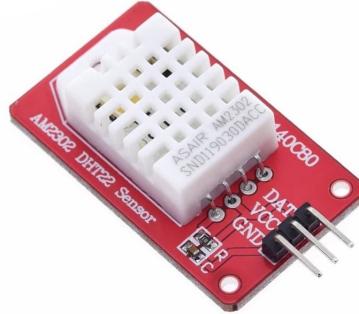
3 Výběr použitých technologií a následná implementace

3.1 Výběr použitých technologií

Problematikou této kapitoly bude výběr teploměru pro měření teploty v bytové jednotce, dále výběr samotných relé pro řízení osvětlení a vytápění bytové jednotky. Budou zde rozebrány druhy topných fólií a na co je při jejich výběru nutné si dát pozor.

3.1.1 Senzor na měření teploty v domácnosti - DHT22

Jedná se o přesnější variantu senzoru DHT11. Oproti DHT11 měří vlhkost v plném rozmezí, kde bylo možné přesně měřit jen od 20% do 80%. Při měření teploty je zde přesnost $\pm 0.5^{\circ}\text{C}$ oproti $\pm 2^{\circ}\text{C}$ u DHT11. Jednou z nevýhod ve srovnání se starším modelem je větší pořizovací cena, která je přibližně o 80Kč vyšší. Druhou a zároveň poslední nevýhodou je, že novější model má vzorkovací frekvenci 0.5Hz, kdežto starší model má vzorkovací frekvenci 2Hz. [8]



Obr. 3.1: Senzor DHT22 [9]

Tento senzor byl do práce vybrán z důvodu jeho přesnosti, která je pro tuto práci dostačující, a také díky jeho malé pořizovací ceně.

3.1.2 Vytápění bytové jednotky

Samotné vytápění bytové jednotky je možné řešit mnoha způsoby. Jednou z nich je přímo zakoupení vyhřívaných panelů, které se umístí na zeď místo stěny, a k nim

připojíme přímo termostat. Dalším možným řešením je návrh teplovodního podlahového topení, kde do celé podlahy je implementováno vodovodní potrubí, kterým protéká horká voda a díky tomu ohřívá plovoucí podlahu. Poslední možností, co se v dnešní době dostala do popředí, je zavedení topné folie přímo pod plovoucí podlahu. Tato poslední metoda bude implementována v bakalářské práci. Samozřejmě při výběru podlahového vytápění nesmíme zapomenout, aby maximální teplota podlahy nepřesáhla ideálně 28°C . Tato mez je limitována mnoha firmami zabývajícími se výrobou podlahových kritin. Pokud by topná folie překračovala ve svém maximální teplotě při plném výkonu tuto teplotu, je potřeba zavedení podlahového snímače a navržení dalšího regulátoru pro hlídání této dané maximální teploty, aby nedošlo k jejímu překročení.

Výkon topné fólie	Délka nahřívání topné fólie	Mezní hodnota topné fólie při instalaci přímo pod plovoucí podlahu	Max. teplota, kterou dosáhne topná fólie o daném výkonu
60 W/m ²	● ● ● ● ● ●	28°C	28°C
80 W/m ²	● ● ● ● ● ●	28°C	32°C
140 W/m ²	● ● ● ● ● ●	28°C	40°C

Obr. 3.2: Ukázka maximální teploty topné fólie pro různé výkony [20]

Na tomto obrázku (3.2) lze vidět různé doby nahřívání pro různé výkony. Pokud zavedeme silnější topnou fólii, sice docílíme rychlejšího ohřátí daného objektu na danou teplotu, ale také zde budeme muset maximální teplotu hlídat. Kdežto když vybereme méně výkonou fólii, tak nám tyto starosti odpadnou a stačí nám pouze řešit samotnou regulaci v místnosti.

Topné fólie Heat Decor

Firma je specializovaná na vyrábění infračervených topných folií o různých výkonech a o různých maximálních teplotách. Tyto topné folie je možné instalovat jak pod plovoucí podlahu, tak pod dlažbu, tak i také pod potěr. Folie fungují na principu elektrického ohříváče s grafitovým (uhlíkovým) jádrem, který vlivem elektrického odporu generuje teplo ve formě dlouhého infračerveného záření. Takto vytvořené teplo z topné fólie, formou infračerveného záření ohřívá předměty, které jsou s ním v kontaktu nebo v dosahu jeho záření. Firma samotná vyrábí několik řad topných folií,

ale zde budou uvedeny jen dva druhy. Všechny fólie pracují se střídavým napětím 230V. [20]

První model HD310G:

Výkon [W/m]	Maximální teplota [$^{\circ}\text{C}$]	Maximální výkon jednoho pásu fólie [W]
220	50	1100
140	40	1120
80	32	960
60	28	840

Tab. 3.1: Možnosti topných folií pro model HD310G

Druhý model HD3025G:

Výkon [W/m]	Maximální teplota [$^{\circ}\text{C}$]	Maximální výkon jednoho pásu fólie [W]
55	50	990
35	40	910
20	32	640
15	28	540

Tab. 3.2: Možnosti topných folií pro model HD3025G

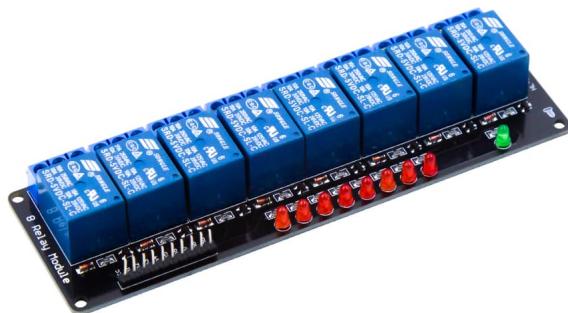
Hlavním rozdílem jak je vidět z tabulek 3.1 a 3.2, je jejich výkon, kde HD3025G dosahuje 4x menšího výkonu než HD310G. Díky tomu HD3025G může nabývat délky na jeden pás až 36m, kdežto HD310G pouze 14m. Záleží tedy na tom co přesně preferujeme, jestli topný výkon nebo možnou vzdálenost natažení jednoho pásu. V tomto projektu budou použity folie typu HD3025G s výkonem 20W a 35W, další druh folie je zde proto, pokud by dané vytápění nestačilo a bylo by nutné použít silnější model a zároveň udržovat stejnou maximální teplotu [20]. Hodnoty do tabulek jsou brány ze zdroje [19].

3.1.3 Technologie na řízení osvětlení a vytápění

U řízení osvětlení máme hned jasno, jelikož se jedná o spínání pouze typu ON/OFF (dvoustavová regulace), je zde ideální relé. U řízení samotného vytápění by bylo lepší navrhnut samotný PID regulátor a na jeho základě spínat výkonový tranzistor, ale bohužel by se muselo jednat o dynamický PID regulátor, jelikož by se soustava pro každou okolní teplotu chovala jinak. Proto bude jednodušší použití principu termostatu, kde se používají relé k řízení teploty místonosti.

Elektromagnetické relé

Jak už je z názvu patrné jedná se o relé, které obsahuje elektromagneticky ovládané kontakty. Samotná úloha relé je spínání výkonové části obvodu. Pro model je vybrána sada s celkem osmi kusy relé na jednom plošném spoji. Hlavním důvodem je, že tato sada je cenově velmi dostupná a zároveň kdysi byla použita v jiném projektu a poté zůstala nevyužita. Samotná relé nesou označení SRD-05VDC-SL-C. Jedná se o relé, které je stavěné do maximálního napětí 230V a procházející proud může činit maximálně 10A. To je pro potřeby samotného projektu dostačující. [17]



Obr. 3.3: Ukázka elektromagnetického relé [18]

3.2 Implementace vybraných technologií do modelu

Tato kapitola se zabývá tématem implementace technologií (osvětlení, vytápění, senzory,...) do modelu domu.

3.2.1 Implementace topné fólie

Výběr topné folie

Jelikož při návrhu byly zakoupeny dva druhy topné fólie, tak je potřeba nejdříve otestovat, která fólie bude dostatečná pro aplikaci. Proto bude nutné nejprve vytvořit pomocí polystyrenu místo a do ní vložit danou fólii, kterou chceme měřit, a následně měřit pomocí teploměru přechodovou charakteristiku topné fólie na jednotkový skok (v tomto případě na zapnutí topné folie do zásuvky na 230V). Díky tomu dostaneme představu o tom, jak dlouho bude muset být daná fólie sepnutá, aby nám dostatečně vytopila místo, popřípadě dokážeme zjistit, že nám daný typ fólie nemusí vůbec vyhovovat.



(a) Ukázka struktury

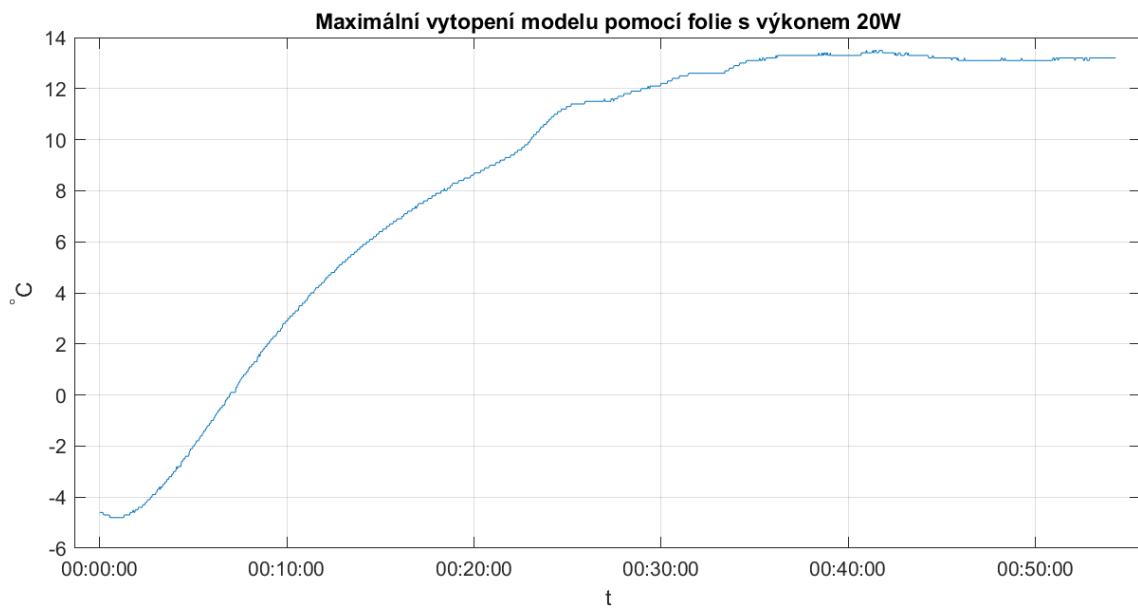


(b) Při běhu programu pro měření

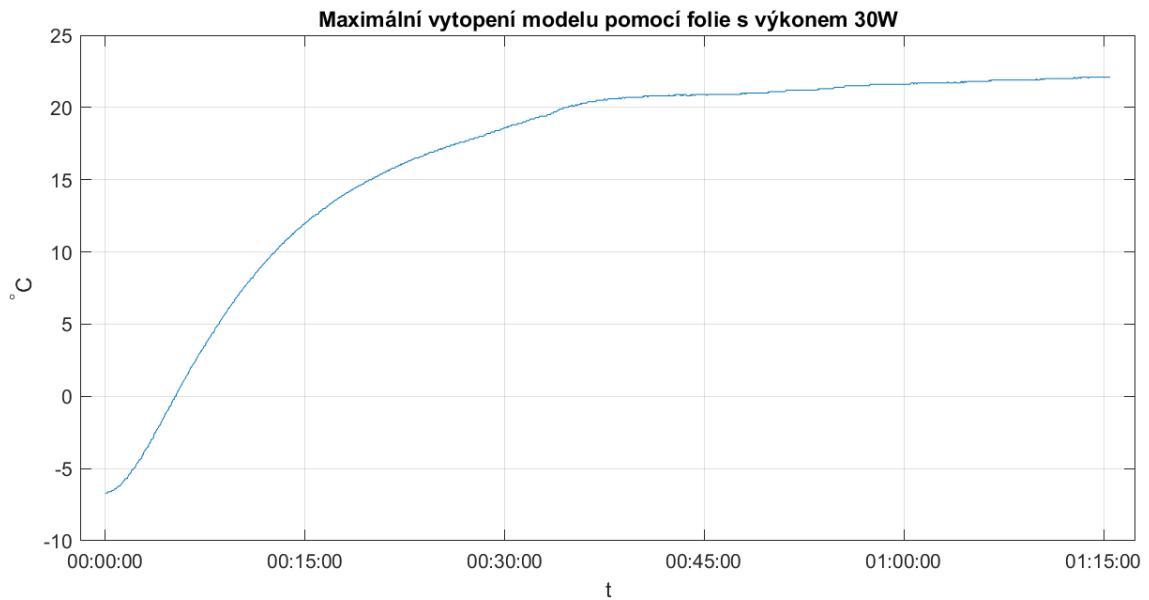
Obr. 3.4: Model pro měření přechodové charakteristiky

Výsledkem měření přechodové charakteristiky pro fólie vybrané v kapitole 3.1.2, jsou grafy 3.5 a 3.6. S těmito změřenými daty byly následně změřeny i výkyvy venkovní teploty, které lze nalézt v příloze: B.2 a B.3, jelikož měření probíhalo ve venkovním prostředí a ne ve stálých podmínkách. Z grafů je patrné, že i když venkovní teplota při měření fólie o výkonu 30W měla nižší teplotu, tak stále je mnohem více schopná daný model vytopit. Z tohoto důvodu bude poté v návrhu bytové jednotky vybrána folie o výkonu 30W.

Samotné měření probíhalo se senzory DHT22, které budou použity i v projektu pro měření teploty. Tyto senzory byly použity s mikrokontrolérem RPI. Na tomto RPI byl nadále spuštěn program, který lze nalézt v příloze B.2, jenž běžel po dobu, dokud se teplota ukazující se ve výpisu z konzole neustálila nebo se nepohybovala v okolí ustálené hodnoty.



Obr. 3.5: Naměřená data pro fólii s výkonem 20W

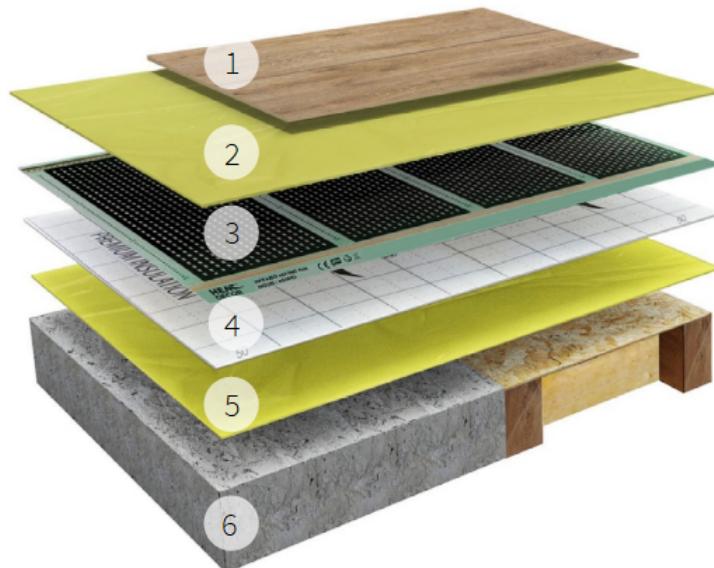


Obr. 3.6: Naměřená data pro fólii s výkonem 30W

Z důvodu obrovského objemu dat nejsou tato data, co jsou použita v grafech 3.5 a 3.6, vypsána v tabulkách, ale jsou dostupná online na odkazu [22], popřípadě v samotném CD, které bylo odevzdáno s bakalářskou prací.

Implementace do modelu

Samotná implementace proběhne pod plovoucí podlahu. V tomto případě je doporučeno se držet návodu od výrobce dané folie, doporučená implementace je vidět na obrázku 3.7.



Obr. 3.7: Ukázka implementace topné folie pod plovoucí podlahu [20]

Kde:

1. Konečná krytina
2. Parotěsná fólie
3. Topná fólie Heat Decor
4. Izolační podložka
5. Parotěsná fólie
6. Podlaha (betonový potér, ...)

Hlavní na co je potřeba si dát pozor, je, aby byla správně odizolována topná folie od betonové podlahy. Čím lépe tato izolace bude provedena tím dosáhneme mnohem lepších vlastností. Dále je potřeba dbát na správnou izolaci přilehlých stěn bytu, aby mezi nimi nevznikaly nikde žádné mezery, které by nám samotné izolační vlastnosti snížily. Toto samotné rozpoložení, co je na obrázku 3.2.1, bylo konzultováno s lidmi, kteří s takovými druhy folií pracují, a po konzultaci byl ze samotné implementace odebrán bod 2 a 4. Díky tomu, že se jedná o model, který nebude zabudován v zemi, tak nemusíme řešit izolační podložku, která zabraňuje průniku vlhkosti z půdy směrem k podlahové krytině.xf

Jako první na betonovou podlahu (obr. 2.3b) byla umístěna paroizolační fólie, která nám usměrňuje teplo vzhůru, aby se nevstřebávalo do betonové podlahy. Po umístění této fólie byly dodělány polystyrenové stěny (obr. 3.8b) a následně přilepeny ke zbytku modelu pomocí lepidla nesoucí název Mamut.



(a) Paroizolační fólie

(b) Oddělení jednotlivých místností

Obr. 3.8: Hrubý základ modelu

Nyní je potřeba rozmístit fólie do svých míst a na stranách vytvořit malé otvory, pro vytažení kabeláže mimo model (obr. 3.9a). Topná folie umístěná v ložnici, na obrázku 3.9a v pravé části, zabírá skoro 80% místnosti, kdežto fólie umístěna v obývací místnosti, na obrázku 3.9a v levé části, zabírá pouze okolo 60%. Tyto velké rozdíly pokrytí se výsledně hodně podepíší na rychlosti vytopení a také na maximální možné teplotě, které je možné s folií v dané místnosti dosáhnout.



(a) Umístění topných fólií

(b) Položená plovoucí podlaha

Obr. 3.9: Kompletní model

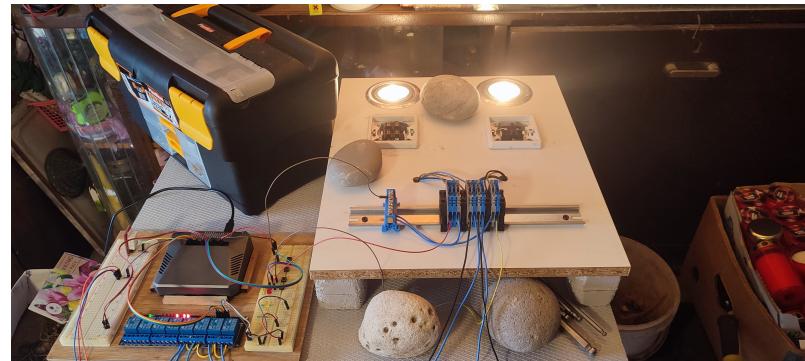
Po umístění folií na svá místa již stačí položit plovoucí podlahu. Na obrázku 3.9b je možné vidět že jedna z místností zůstává nevyplněna. To je z důvodu, že v této místnosti nebudeme řešit regulaci ani osvětlení, tudíž je zbytečné zde pokládat podlahu. A slouží díky tomu, jako ukázka co se nachází pod samotnou plovoucí podlahou a topnou fólií.

3.2.2 Implementace osvětlení

Jako osvětlení byly vybrány klasické 230V žárovky, aby byl model co nejvíce podobný reálnému bytu. Dále byl následně vybrán jeden klasický vypínač a jeden dvojitý vypínač, který bude umístěn v obývacím pokoji, jako demonstrace možného řízení osvětlení ze dvou míst, kde každá půlka vypínače představuje jeden vchod do místnosti, jak je možné vidět na plánu modelu A.1.



Obr. 3.10: Model osvětlení před osazením DIN lišty



Obr. 3.11: Kompletní osvětlení společně se svorkami

Zde na obrázku 3.11 je vidět kompletní uspořádání i se svorkami, které bude následně popsáno detailně v kapitole 4.5 a prakticky v kapitole 5.3.1. Samotné zapojení je plánováno s již hotovou kompletací jističů a chráničů v rozvodové skříně.

3.2.3 Implementace teplotních senzorů

Umístění senzorů je již známé z obrázku 2.1, kde je vidět umístění senzorů na stěnách modelu. Senzory z důvodů možné výměny nejsou na stěně nějak přilepeny. Jsou pomocí kabelů zavěšeny lehce dále od okraje, aby nebyly tak ovlivněny pohlcenou teplotou od samotných polystyrenových stěn. Další výhodou tohoto umístění je, že

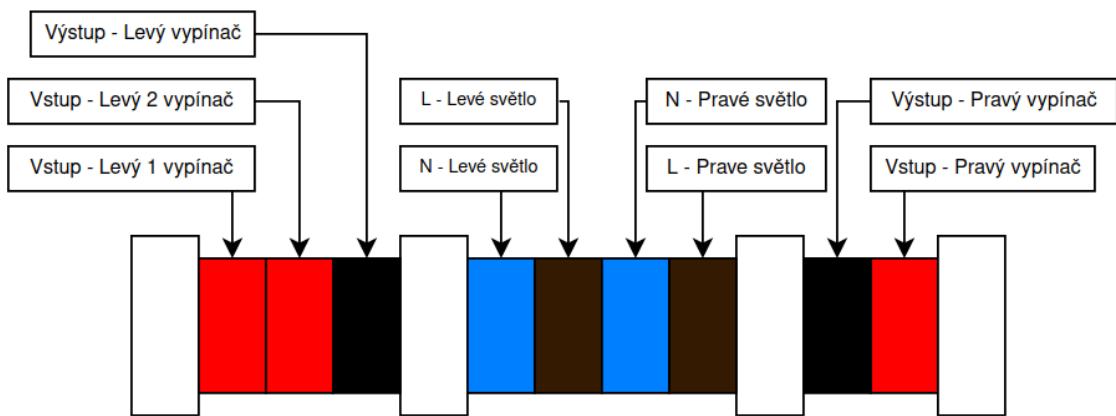
pokud by se náhodou senzor pokazil, tak je možné ihned rychlá výměna a není potřeba složitě řešit odmontování. A hlavním důvodem pro takovýto druh umístění a uchycení je, že se jedná pouze o model, tudíž není přesně nutné využívat umístění do nějaké krabičky, která by byla následně přidělána na stěnu. Díky vynechání této implementace je mnohem jednodušší poté měnit různě umístění senzorů.



Obr. 3.12: Umístění senzorů v modelu

3.2.4 Zapojení svorkovnice

Jelikož se na samotné desce nachází svorkovnice (obr. 3.11), která je umístěna na DIN liště, tak pro lepší přehlednost zde níže lze nalézt obrázek, který přesně popisuje, jak jsou jednotlivé svorky využity a zapojeny.



Obr. 3.13: Zapojení svorkovnice

Jak je možné vidět samotné svorky jsou rozděleny celkem do tří segmentů. Kde dva segmenty jsou pro samotné vypínače a poslední segment pro osvětlení.

4 Definice požadavků na řídící systém a navržení konceptu řízení vybraných technologií

Tato část je věnována teoretickému rozboru zapojení, popřípadě možných vzniklých problémů při implementaci samotných technologií a řídícího systému.

4.1 Požadavek na hlavní řídící jednotku

Jak je možné vidět z grafů 3.6, doba náběhu samotného systému se pohybuje v řádech minut. Tudíž není potřeba mít řídící jednotku s obrovským výpočetním výkonem, ale bude nám stačit jednotka, kde se bude provádět program například každých 100ms, popřípadě jak si sami již upravíme. Díky tomuto poznatku nám bohatě postačí RPI 4B a ještě nám většina výkonu zbyde pro další různé aplikace popřípadě programy. Jedním z dalších požadavků je možnost fungování samotné jednotky bez nutnosti připojení internetu. To se dá docílit buď detekcí připojení, popřípadě lze v částech kódu, kde se má komunikovat pomocí internetu, použít výjimky (try, except), tato skutečnost zapříčiní, že pokud primární kód nelze provést, tak se provede sekundární kód. Tato nakonec druhá možnost pomocí výjímek bude v programu pro řídící jednotku použita.

4.2 Požadavek na databázi

Databáze samotná bude sloužit pouze pro ukládání dat a nastavení. Takže je potřeba, aby běžela neustále společně se samotnou řídící jednotkou. Ideálním případem je vytvoření speciálního uživatelského jména a hesla, kterému určíme, do jaké sekce databáze bude mít přístup a zároveň jaké změny bude moci v dané databázi provádět. Ideální je také, pokud by databáze měla být zpřístupněna mimo lokální síť, nastavit silná hesla pro jednotlivé uživatele v databázi, aby se nám někdo do té databáze tak snadno nedostal. Poté už jen stačí nastavit samotný firewall a přesměrování přímo na náš server v lokální síti.

4.3 Požadavek na regulaci teploty

Jelikož budeme používat dvoustavovou regulaci, je potřeba myslit na to, že se musí použít relé, které zvládne 230V a také dokáže být ovládané pomocí 5V logiky. Na toto byl brán ohled při výběru samotných relé v kapitole 3.1.3. Pokud by jsme chtěli

použit tato relé pro vytápění klasické podlahy, je potřeba si dávat pozor na výkon folíí. Díky tomu, že se zde jedná pouze o model, tak samotný výkon je v jednotkách watů. Kdežto při klasickém vytápění podlahy je nutné na to brát ohled, aby nebyla překročena maximální hranice samotného relé, které ovládá danou fólii. Posledním velkým požadavkem na regulaci teploty je, aby se teplota pohybovala maximálně v rozmezí $\pm 1C^\circ$ od požadované hodnoty. Tento požadavek se dá popřípadě dodlat nastavením velikosti hystereze v programu. Samotné topné fólie nemají nějak předem danou orientaci. Díky tomu je jen na nás co si zvolíme jako fázi a co jako nulovací vodič.

4.4 Spínání osvětlení

Po výběru klasických 230V žárovek bude na řízení použito samotné relé, které díky svému maximálnímu možnému výkonu bohatě postačí na tento účel. Šlo by zde samozřejmě i použít například 24V nebo 12V osvětlení (LED pásky,...), ale to už by se musel řešit další možný napájecí zdroj. Jak topné fólie, tak žárovky jsou na 230V to nám usnadňuje zapojení a zároveň nemusíme řešit přídavný zdroj.

4.5 Zapojení vypínače

Samotné vypínače je potřeba snímat pomocí RPI. K tomuto účelu lze využít 5V zdroj, kterým samotné RPI na GPIO pinech disponuje a to poté nám stačí jen snímat, a následně na základě snímání dvoustavové hodnoty (0,1) rozhodovat co se provede. Zároveň zde nejsou potřeba nějak extrémně rychlé reakce na sepnutí nebo odepnutí vypínače. Ideální stav by byl reakce do maximálně 100ms. Tato hodnota je volena tak, aby uživatel nevnímal příliš velkou prodlevu mezi sepnutím vypínače a rozsvícením osvětlení. A zároveň aby nemusel být samotný procesor vytížen na svojí maximální zátěž. Nesmíme také při teoretickém návrhu zapomenout na ošetření samotných zákmitů, které můžou být vyvolány.

5 Implementace řízení pro vybrané technologie

Kapitola je zaměřena na serverovou část a také na část řídící. Z části serverové je to výběr vhodného serveru, popřípadě hostingu. A řídící část je zaměřená na zapojení jednotlivých komponent použitých v modelu (osvětlení, vytápění, vypínače). A v poslední řadě také ukázka on-off regulace za použití programu Matlab.

5.1 Server na chytrou domácnost

Hlavní a nedílnou součástí každé chytré domácnosti je server, na kterém běží všechny potřebné technologie. Výkonnost serveru se musí odhadnout podle náročnosti chytré domácnosti. Pokud by jsme chtěli mít po domě např. několik kamer a zaznamenávat obraz a rozpoznávat obličeje, zda se nám nepohybuje někdo cizí po pozemku a do toho mít i zakomponované další senzory a akční členy, tak jsme nuceni sáhnout po silnějších strojích.

Pokud ale již kupujete chytrou domácnost přímo od firmy, tak se touto problematikou nemusíte zabývat. Ale díky tomu, že tato práce je o vlastním návrhu chytré domácnosti, je dobré zde všechny tyto možnosti, co jsou níže, zmínit.

Pro bakalářskou práci byla vybrána konfigurace RPI 4B, který slouží jako hlavní mozek celé chytré domácnosti, následně vlastní server, na kterém běží databáze pro ukládání dat. A jako databázový systém je použita MYSQL databáze.

5.1.1 Využití VPS popřípadě VDS

VPS

VPS je virtuální privátní server, kde při zařízení VPS dostane uživatel předem přidělený výkon. Na jednom fyzickém serveru může běžet více virtuálních privátních serverů. Většinou sdílený mezi těmito jednotlivými VPS na stejném fyzickém zařízení bývá disk a procesor, kde díky většímu vytížení jednoho uživatele se může zpomalit čtení u dalších uživatelů a zároveň snížit maximální výkon, který můžou ostatní uživatelé využít. VPS bývá hostován přímo hostingovou službou.

VDS

VDS neboli virtuální dedikovaný server je server, který je umístěný v datacentru a máte k dispozici kompletní jeho výkon a celý tento server si pronajímáte. Jedná se o mnohem dražší službu než VPS, ale cena je dána tím, jaký výkonný stroj si

pronajímáte. Díky tomu, že se jedná o dražší řešení, tak se spíše hodí do firmy, kde se nechťejí starat o vlastní servery a potřebují velký výkon a dobrou spolehlivost.

5.1.2 Vlastní server

Člověk si pod slovem server hned představí stroj za desítky tisíc korun českých. Ale tak to být nemusí, přímo serverem se může stát i starý stolní počítač, popřípadě nepoužívaný notebook. Hlavní přitom je výběr správného systému, který na tomto stroji poběží. Pokud uživatel nemá zkušenosti s linuxem, může využít systém Windows. Ale pokud už nějaké zkušenosti uživatel má, tak je ideálnější použití systému na bázi linuxu (Ubuntu, Fedora,...). Pokud se ale nemůže rozhodnout nebo by chtěl používat více různých systémů, je možné využít systému pro virtualizaci serverů. Jedním z těchto možných systémů, co lze využít, je Proxmox. Jedná se open-source řešení a díky tomu není nutné za něj platit. Proxmox je více specializovaný pro firmy. Pro domácí použití je lepší využít systému Unraid. Bohužel tento systém je již placený a to od počtu připojených disků, kde nejlevnější doživotní licence stojí 60\$ za maximálně 6 možných připojených počítačových disků.

5.1.3 Raspberry PI

Nejlevnějším možným řešením je pořízení RPI. Jedná se o jednodeskový počítač, který díky novějším modelům může mít až 8GB RAM. Ideálním hlavním systémem pro tento počítač je systém Raspbian, který je přímo vytvořený pro tyto jednodeskové počítače. Pokud by tento systém nevyhovoval uživatelským požadavkům, tak je možné využít i jiného operačního systému, jako je například Ubuntu. Popřípadě možná instalace LITE verze Windowsů. Základní cena RPI se pohybuje nejlevněji okolo 1000Kč (data z 13.10.2021) za nejnovější modely s minimální konfigurací.



Obr. 5.1: Raspberry PI 4B [10]

5.2 Databázové systémy

V chytré domácnosti je posíláno mnoho dat a je potřeba na základě jejich hodnot vyhodnocovat nějakou činnost. Proto je potřeba zařídit databázi, aby mezi sebou jednotlivé senzory a akční členy mohly komunikovat. Ať už prostřednictvím posílaných hodnot, například teploty, vlhkosti atd., tak i uložením možných propojení (například propojení relé ovládající osvětlení a vypínače v místnosti). Databázi je možné si zařídit ihned pomocí několika možností. První možností je pronajmutí nějakého hostingového serveru. Bohužel u této možnosti je pravděpodobné, že se jí někdo pokusí napadnout zvenku a může získat možná data (například zda se někdo doma nachází za pomocí stavu osvětlení v databázi). Proto je zde lepší využít lokálního serveru a na něj nainstalovat požadovanou službu. Docílíme to tím, že je naše databáze přístupná pouze po lokální síti. A pokud máme doma zprovozněnou službu VPN a veřejnou IP adresu, můžeme se k tomuto serveru připojovat i vzdáleně pomocí právě VPN. Nebo druhou možností je samotné přesměrování samotných paketů přicházející na daný port přímo k našemu lokálnímu serveru. Ale tím akorát zvýšíme šanci získání dat z databáze do rukou hackerů při špatném zvolení hesla a nastavení firewallu.

5.2.1 Relační databázový model

Jak je již z názvu patrné, jedná se o model používající relace. Pod pojmem relace si můžeme představit databázovou tabulku (například tabulka s názvem "switches"). Každá relace se pak skládá z atributů (v podobě sloupců) a záznamů (v podobě řádků).

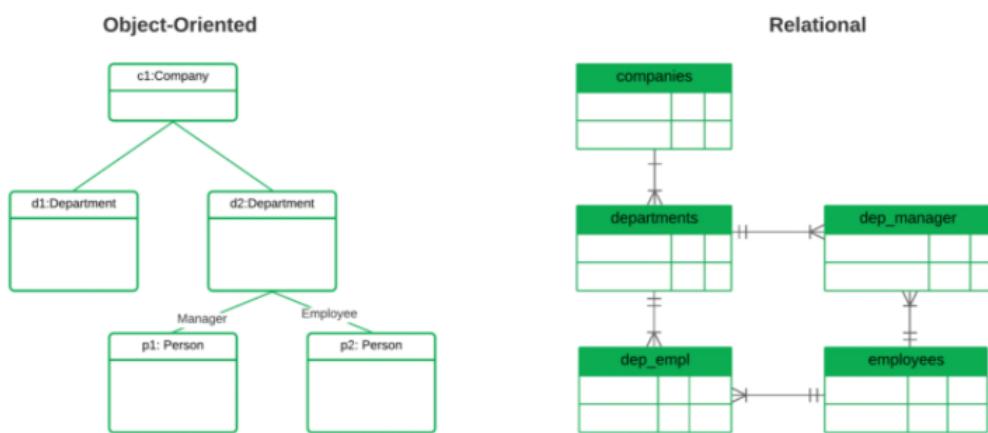
	id	switch_location	light_location
▶	1	chodba1	chodba
	2	chodba2	chodba
	3	kuchyn	kuchyn
*	NULL	NULL	NULL

Obr. 5.2: Ukázka relační databáze

V tomto případě (obr. 5.2) je relace s názvem "connection_light_swatches" (bohužel z tohoto obrázku toto není patrné) a nacházejí se v ní atributy: id, switch_location, light_location. Poté už je možné vidět jednotlivé záznamy, které jsou seřazeny podle "id".[11],[12]

5.2.2 Objektový databázový model

Základním rozdílem od relačního databázového modelu, kde se využívaly tabulky, se zde využívají objekty k ukládání informací. Obecně se jedná o kombinaci databázových možností a objektově orientovaného programování. Zde by mohlo dojít k mylce, že pokud používám objektově orientované programování a pracuji tam s nějakými daty jako je třída "Student", který má atributy "třídu" a "věk", tak vlastně již s takovou databází pracuji, ale to je veliký omyl. Hlavním rozdílem zde je, že pokud nám program skončí, tak objekt zmizí, kdežto u objektového databázového modelu se data neztratí, ale uloží do databáze, kde při příštém spuštění programu mohu dále s těmito daty pracovat.



Obr. 5.3: Ukázka rozdílu relační a objektové databáze [13]

5.2.3 MySQL

Tento druh databáze uplatňuje, jako většina, relační databázový model (kap.5.2.1). Komunikace s touto databází probíhá pomocí SQL syntaxů. Jedním ze syntaxů například může být: "SELECT * FROM data;". Tento syntax vezme všechny atributy a všechny data z tabulky jménem "data" a pošle je uživateli, který tento syntax zadal. [15]

5.2.4 PostgreSQL

Oproti MySQL, který vytvořila firma MySQLAB a nyní jej vlastní Oracle, Postgres je vytvořen komunitou vývojářů a firem. Dalším rozdílem oproti MySQL je že PostgreSQL využívá objektově relační databázový model. Jde o hybrid mezi relačním modelem a objektovým modelem. [16]

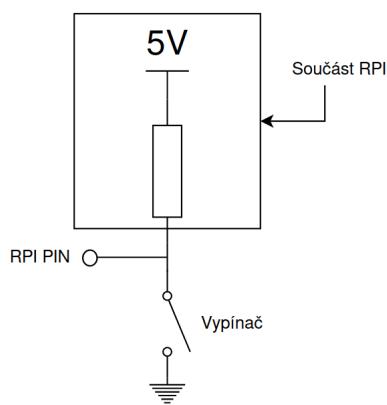
5.3 Řízení osvětlení

5.3.1 Zapojení vypínače

Samotné zapojení vypínačů je řešeno přes PULL-UP rezistor. Jedná se o systém, jenž lze vidět na obrázku 5.4, který při sepnutí vypínače přivede 5V přes odpor na zem, a my díky tomu ztratíme měřené napětí na RPI PINU, to se projeví jako detekce hodnoty "LOW". Naopak, když je vypínač vypnutý, tak na RPI PINu naměříme 5V a to RPI detekuje v programu jako hodnotu "HIGH". Pro tuto detekci byly vyhrazeny GPIO PINY GPIO-23,GPIO-27,GPIO-17 (obr. B.1). Kde GPIO17 je pro ukončení programu na RPI.

Část kódu pro nastavení GPIO do módu PULL_UP:

```
GPIO.setup(button_1, GPIO.IN, pull_up_down=GPIO.PUD_UP)
```



Obr. 5.4: Zapojení vypínače

Zabránění zákmitům

Nedílnou součástí, na co je potřeba myslet při návrhu vypínače, je že při sepnutí nebo odepnutí vypínače může dojít k zákmitům. Toto je řešeno, zde v bakalářské práci, softwarově a to pomocí jednoduché podmínky a cyklování. Jde hlavně o to, že při sepnutí spínače se načítá hodnota proměnné až do nějaké maximální hodnoty (kterou si samy v programu nastavíme), a jakmile této hodnoty dosáhne tak, dojde k sepnutí výstupu, jenž sepne relé, které má na starosti jednotlivé žárovky.

Naprogramování dvojitého vypínače

Pro správné fungování ovládání osvětlení ze dvou míst, bylo navrženo softwarové řešení, které pouze kontroluje stavy samotných vypínačů a pokud jsou stavy rozdílné

tak sepne relé v obývacím pokoji a pokud ne tak naopak jej vypne. Jedná se o velmi jednoduchou a účinnou metodu, jak tuto problematiku řešit. Další možným nápadem je samotná úprava hardwarevého zapojení samotných vypínačů.

Výpis 5.1: Řízení osvětlení ze dvou míst

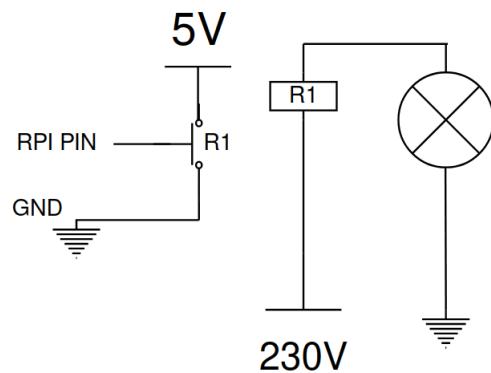
```

1  if living_room_switch_1.actual_state != 1
2      living_room_switch_2.actual_state: 2
3          if not relay[1]: 3
4              living_room_light.actual_state = 1 4
5              living_room_light.update_datab_state() 5
6              relay[1] = True 6
7      else: 7
8          if relay[1]: 8
9              living_room_light.actual_state = 0 9
10             living_room_light.update_datab_state() 10
11             relay[1] = False 11

```

5.3.2 Spínání žárovek

Spínání žárovek bylo v tomto případě vyřešeno pomocí elektromagnetického relé, které je softwarově řízeno. Jak již bylo naznačeno, ke sepnutí dojde po na-cyklování hodnoty proměnné. Pro spínání relé je potřeba dát GPIO piny do stavu **OUTPUT** a poté přepínat mezi HIGH a LOW anebo je možné využít přepínání mezi módy **INPUT** a **OUTPUT**, které nakonec bylo zabudováno do programu. Jako piny pro spínání relé byly vybrány GPIO-6 a GPIO-13 (B.1). Spínací prvek relé je již na vstupu, v samotném modulu (plošném spoji), chráněn ochrannými diodami proti přepětí.

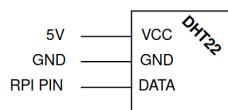


Obr. 5.5: Zapojení relé ovládající osvětlení

5.4 Regulace teploty v místnosti

5.4.1 Zapojení teplotního senzoru

První, co je potřeba při řízení teploty, je čist správně data ze senzorů. K tomu byl napsán program, který lze nalézt v příloze C.4. Program pro teploměr (ale i vypínače, ...) je napsán pomocí třídy, a to z důvodu dědění ze třídy, která pracuje s databází. To pokud je vše správně napsáno a zároveň děděno nám zjednodušuje dílkí programy už jen kvůli tomu že si můžeme vytvořit libovolný počet senzorů, kolik budeme chtít, a jen díky jejich jménům k nim přistupovat a čist z nich data, popřípadě s nimi dál nějak pracovat. Níže na obrázku 5.6 je vidět jak je senzor v této bakalářské práci zapojen.

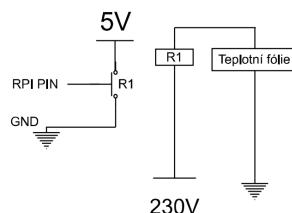


Obr. 5.6: Zapojení senzoru DHT22

Jako piny pro teplotní senzory jsou používány GPIO-09,GPIO-11 (B.1). Nadále pro zachování jednoduchosti regulace není zde v práci navrhován PID regulátor, nýbrž pouze on-off regulátor. Proto jako další krok je potřeba si zapojit relé, aby nám napájelo teplotní fólie. V poslední řadě nám stačí vytvořit takový program, který nám podle teploty bude spínat výstupní piny a díky tomu dojde k sepnutí relé, které spustí napájení samotné topné fólie.

5.4.2 Zapojení relé

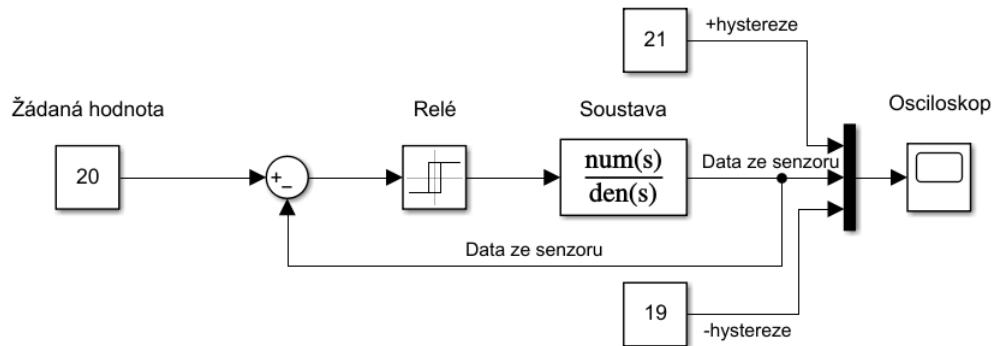
Spínání a vypínání relé v této úloze má úplně stejný princip jako v kapitole 5.3.2. Zde pro ovládání relé slouží piny GPIO-5 a GPIO-6 (B.1). Spínací prvek relé je již na vstupu, v samotném modulu (plošném spoji), chráněn ochrannými diodami proti přepětí.



Obr. 5.7: Zapojení relé pro jednu teplotní folii

5.4.3 Regulace teploty

Pro regulaci je využíváno relé, tím pádem se jedná o dvoustavovou regulaci (on-off regulace). Nevýhodou této regulace je, že nikdy nedosáhneme nulové ustálené odchylky, vždy nám bude naše reálná hodnota kmitat okolo žádané hodnoty. Odchylka se dá snižovat velikostí hystereze, kterou v programu nastavíme. Například pokud nastavíme velikost hystereze na $\pm 1^{\circ}\text{C}$ a naše žádáná hodnota bude 20°C , tak nám bude výsledná teplota kmitat v rozmezí $19^{\circ}\text{C} - 21^{\circ}\text{C}$ (platí pro systém prvního řádu). Čím více tímto hysterezi zmenšíme tím razantně ovlivníme životnost samotného relé. Pokud by hystereze nebyla žádná relé bude neustále spínat a rozepínat, a díky tomu budeme muset mnohem dříve řešit jeho výměnu z důvodu opotřebení. Velkou výhodou on-off regulace je, že je velmi jednoduchá na naprogramování a na obsluhu. Proto je velmi využívána hlavně pro regulaci teplot, jelikož zde působí hodně okolních vlivů, se kterými by bylo potřeba při návrhu PID regulátoru počítat. Zároveň je potřeba počítat s tím, že samotné složení podlahy (akumulátory energie) je systém vyššího jak prvního řádu, tudíž regulace nebude úplně přesná, tím je myšleno, že teplota podkmitne anebo překmitne přes hysterezi. To je zapříčiněno setrvačností samotné soustavy. Rozdíl je možné vidět na porovnání grafů 5.9 a 5.10. Všechny demonstrace v Matlabu jsou prováděny pro žádanou hodnotu 20°C a pro hysterezi $\pm 1^{\circ}\text{C}$. Hystereze je nastavená vyšší jak v modelu a to z důvodu lepší viditelnosti překmitů a podkmitů.

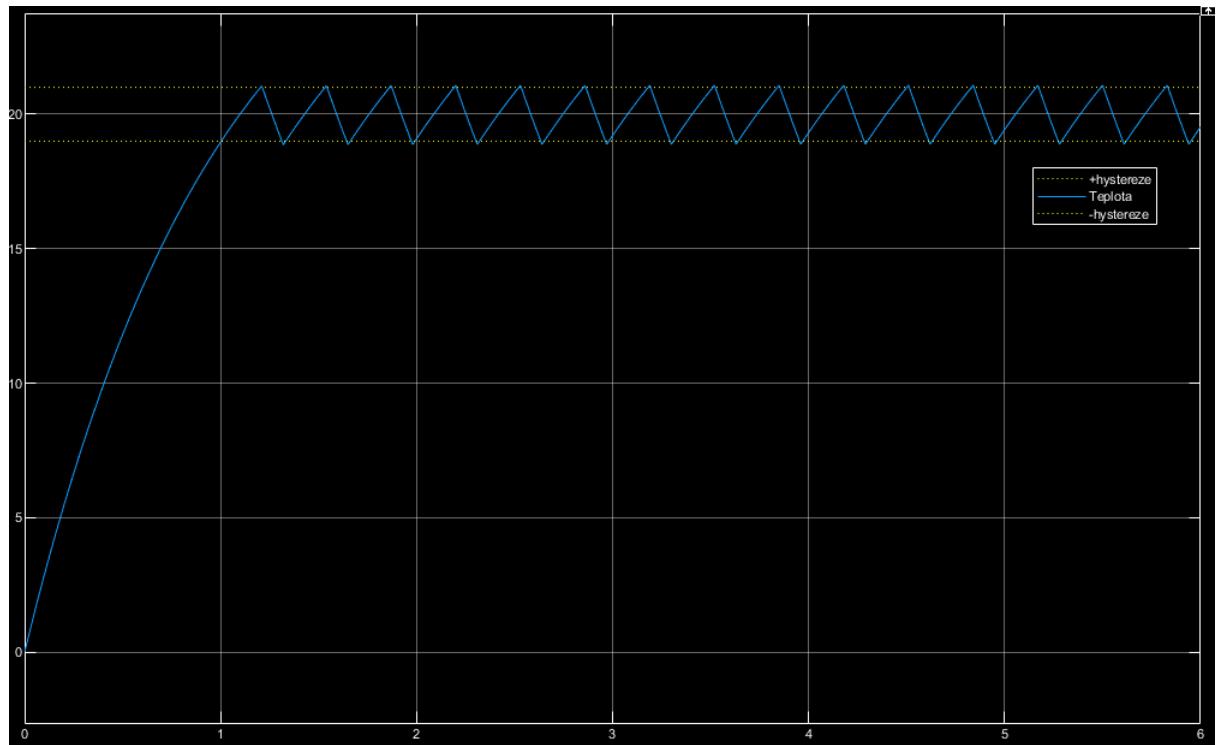


Obr. 5.8: Zapojení regulační smyčky pro demonstraci

Demonstrace regulace soustavy prvního řádu

Pro demonstraci soustavy prvního řádu byla použita rovnice soustavy, které nemá žádnou souvislost se samotným modelem a slouží zde pouze k demonstraci.

$$Fs = \frac{30}{s+1} \quad (5.1)$$

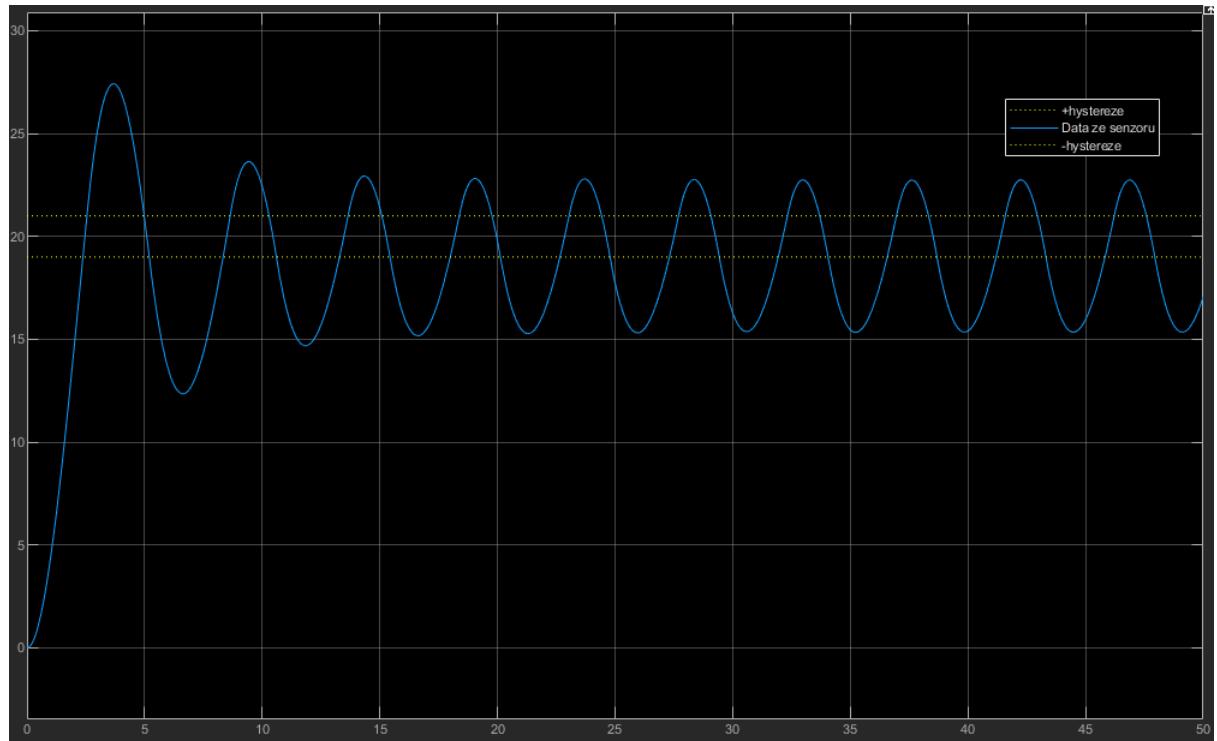


Obr. 5.9: Regulace soustavy prvního řádu

Demonstrace regulace soustavy druhého řádu

Pro demonstraci soustavy druhého řádu byla použita rovnice soustavy, které nemá žádnou souvislost se samotným modelem a slouží zde pouze k demonstraci.

$$Fs = \frac{30}{3 \cdot s^2 + s + 1} \quad (5.2)$$



Obr. 5.10: Regulace soustavy druhého řádu

Program pro řízení relé

Jelikož se jedná o jednoduchou regulaci, tak nám pouze stačí číst teplotu a na základě rovnice vyhodnotíme, zda relé bude spuštěno nebo vypnuto.

Ukázka rovnice pro sepnutí relé:

$$stav = (\text{merenaTeplota} < (\text{pozadovana} - \text{delta})) \quad (5.3)$$

Výpis 5.2: Regulaci teploty v místnosti

```
#Temperature measurement  
DhtSensor1.mess_temperature()  
DhtSensor2.mess_temperature()  
# Living room switch  
if DhtSensor2.temperature > set_temp_liv + delta_temp:  
    GPIO.setup(heat_foil_liv, GPIO.IN)  
    relay[3] = False  
elif DhtSensor2.temperature < set_temp_liv - delta_temp:  
    GPIO.setup(heat_foil_liv, GPIO.OUT)  
    relay[3] = True  
# Bedroom switch  
if DhtSensor1.temperature > set_temp_bed + delta_temp:  
    GPIO.setup(heat_foil_bed, GPIO.IN)  
    relay[2] = False  
elif DhtSensor1.temperature < set_temp_bed - delta_temp:  
    GPIO.setup(heat_foil_bed, GPIO.OUT)  
    relay[2] = True
```

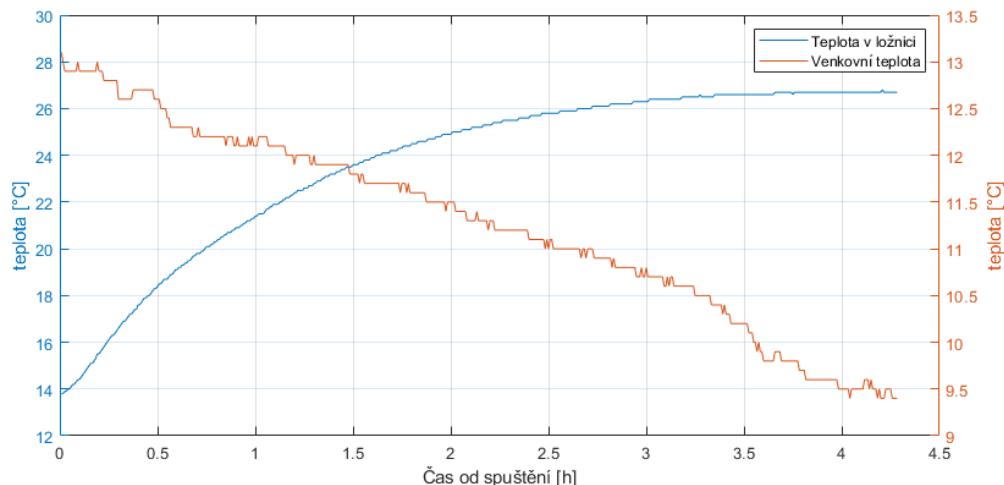
Jak je v kódu patrné, dochází zde nejprve k měření teploty, k tomu slouží funkce, která je implementovaná ve třídě DhtSesnor a její kód je možné nalézt zde C.4.

6 Demontrace funkčnosti

Poslední kapitola bakalářské práce je věnována kompletní demonstraci funkčnosti samotného modelu a k němu navržených řídících algoritmů, senzorů, aktuátorů. Samotná naměřená data, co jsou použita v této kapitole lze nalézt v přiloženém CD, které bylo odevzdáno s bakalářskou prací.

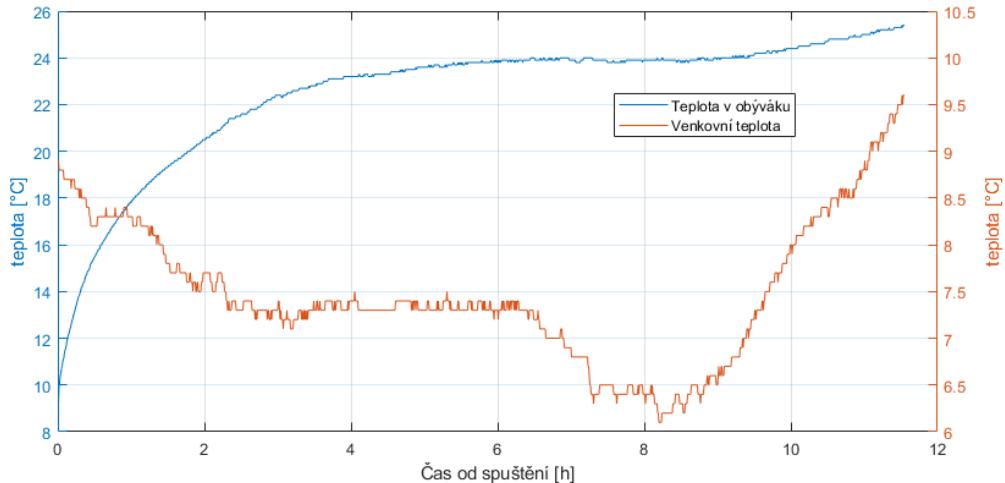
6.1 Test topných folií

Jako první testem, co vůbec byl na modelu proveden, byl test, do jaké maximální teploty jsou schopny jednotlivé fólie samotný model vytopit. Z důvodu provozování samotného měření v noci a v březnovém měsíci, se samotný model nacházel umístěný v garáži. Ale pro lepší představu byla měřena i venkovní teplota.



Obr. 6.1: Měření přechodové charakteristiky v ložnici

Hned u grafu 6.1 je možné si všimnout, jak samotné vyvíjení teploty připomíná skoro dokonale model druhého a vyššího řádu, o čemž bylo mluveno v kapitole 5.4.3. Jehož následky se nám projeví i při regulaci.



Obr. 6.2: Měření přechodové charakteristiky v obýváku

Další měření proběhlo v obývací místnosti 6.2, kde samotná fólie zabírá přibližně 60% plochy, kdežto v ložnici je to okolo 90%. Toto se projevilo i menší možnou maximální vytopenou teplotou samotné místnosti, a to přibližně o 3°C . Zároveň je možné vidět u konce grafu, že okolní teplota vzrostla, a s tím se pohnula i teplota v místnosti. To dokazuje, že samotná topná folie při takovéto okolní teplotě nedokáže model již více vytopit. To může být zapříčiněno nedokonalou izolací a také nedokonalým pokrytím celé plochy.

6.2 Regulace teploty v místnosti

Díky poznatku z minulé kapitoly byla regulace nastavena na mnohem nižší teploty, aby bylo dosaženo rychlejší dynamiky systému a nemuselo se dlouho čekat na změřená data. Samotné nastavení regulace bylo:

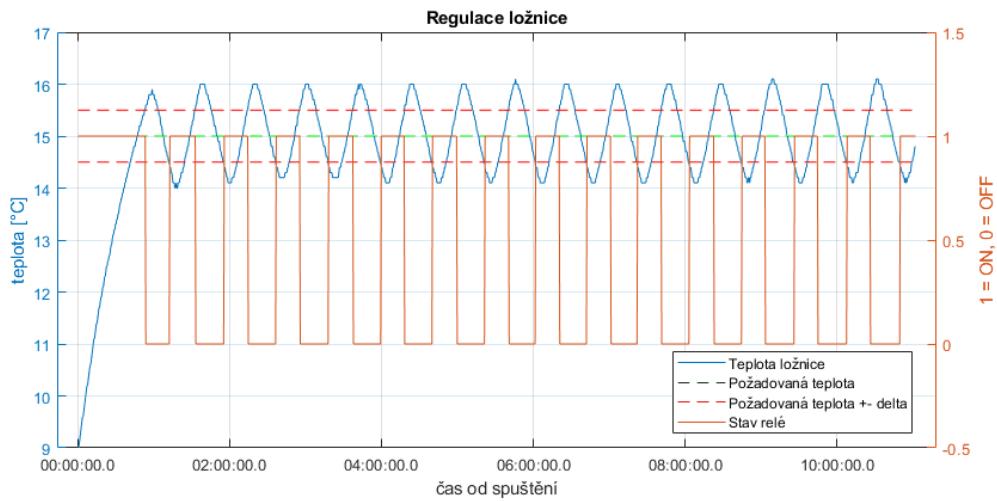
Ložnice

1. Teplota: 15°C
2. Delta: $\pm 0.5^{\circ}\text{C}$

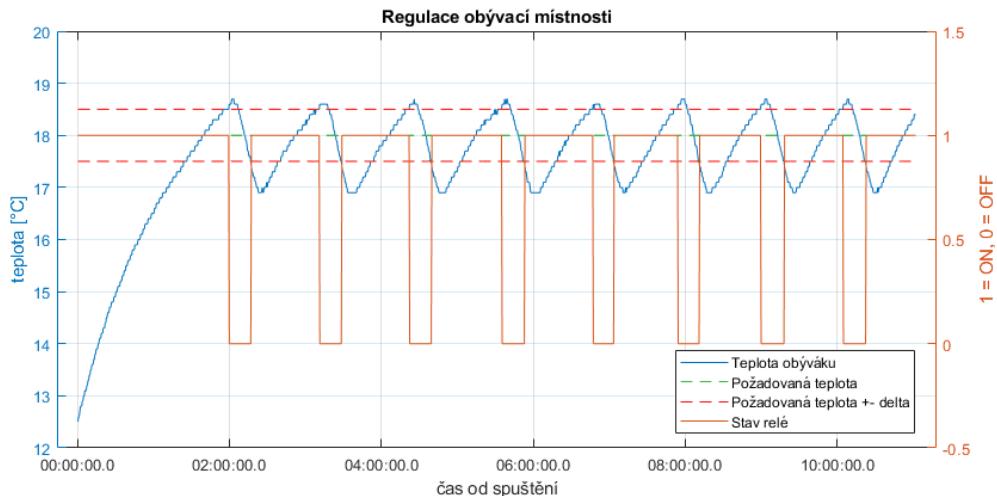
Obývací místnost

1. Teplota: 18°C
2. Delta: $\pm 0.5^{\circ}\text{C}$

Úmyslně bylo nastaveno, aby požadovaná teplota v ložnici byla menší jak v obývací místnosti. Je to z toho důvodu, že při reálné regulaci je nastavena nižší teplota v místnostech, kde se spí, než v místnostech, kde se přes den buď pracuje nebo relaxuje.



Obr. 6.3: Regulace teploty v ložnici



Obr. 6.4: Regulace teploty v obývací místnosti

Jak je patrné při samotném porovnání grafů 6.3 a 6.4, je vidět, že ložnice má mnohem rychlejší dynamiku. Jak bylo avizováno v kapitole 5.4.3, je na grafech možné vidět překmit, který byl předtím pouze simulován, a že předpoklad byl správný. Na grafu je i vyobrazené relé, v jakých stavech se nacházelo (konkrétně označeno oranžovou barvou).

6.3 Test osvětlení

Pro demonstraci osvětlení bylo místo fotek nahráno kompletní video.

Odkazy je možné nalézt zde: Dropbox, Google drive

Popřípadě samotné video se nachází v přiloženém CD s bakalářskou prací.

Video ukázka obsahuje test obou osvětlení a zároveň je na něm vidět i sepnuté relé pro regulaci. Je to kvůli tomu že natáčení ukázky probíhalo v době kdy se zrovna měřily přechodové charakteristiky fólií v modelu.

6.4 Test Telegram bota

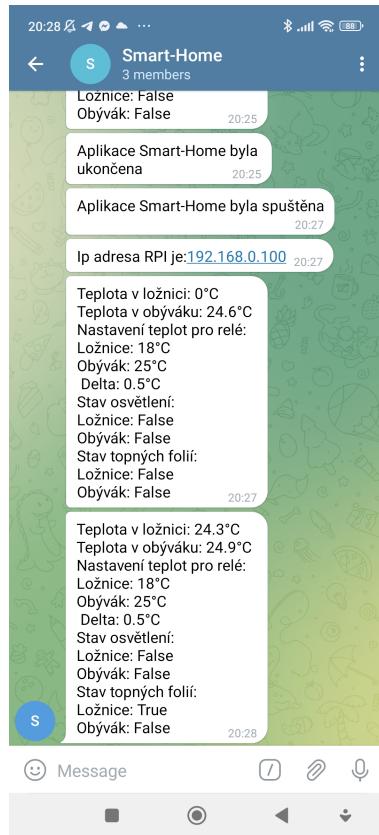
Pro lepší komunikaci s uživatelem byl navíc přidán bot do aplikace Telegram, který vždy jednou denný čas, který samy v programu nastavíme, vypíše všechna důležitá data do Telegram skupiny. Samotný bot je dělaný pomocí requestu na webu, kde se dá pomocí připojení bota do správné skupiny posílat zprávy.

Výpis 6.1: Obecná funkce pro posílání telegram zpráv

```
def TelMess(text):
    try:
        base_url = 'https://api.telegram.org/bot5189477795:
AAEVYv_VOPWOicis7RtdYsNIQZFNOMHx1Jk/
sendMessage?chat_id=-682759305&text=
{text}'.format(text=text)
        requests.get(base_url)
    except:
        return 0
```

1
2
3
4
5
6
7
8
9

Poté je tato funkce volána a pomocíní posílána data. Výsledná zpráva poté vypadá takto.



Obr. 6.5: Zpráva poslaná botem na telegram skupinu

Při pořízení tohoto snímku 6.5 bylo nastaveno posílání každých 30 sekund. Hlavním důvodem samotného vytvoření bylo to, aby bylo možné kontrolovat samotnou regulaci i na dálku, kdyby byl člověk mimo svůj domov. Je zde ze začátku vypsána i samotná IP adresa RPI, kterou mu DHCP server přiřadil. To je především hlavně kvůli možnému následnému snažšímu připojení, jelikož samotné RPI bylo programován vzdáleně pomocí SSH protokolu v aplikaci PyCharm. Bohužel samotný bot nebyl doprogramován do fáze, kdy by dokázal i přijímat zprávy a nějak na ně reagovat. Kvůli tomu slouží pouze jako informativní bot.

6.5 Google tabulky

Díky tomu že telegram bot slouží pouze pro vypisování hodnot, tak díky tomu bylo rozhodnuto o dalším možném propojení a to přímo s google tabulkami, ve kterých lze jednoduše nastavovat parametry regulace a číst aktuální hodnoty. Samotné propojení google tabulek a pythonu je přes python knihovnu gspread, a také za pomocí Google Cloud Platform, přes které je celá komunikace zprostředkována.

	A	B	C	D	E	F	G	H
1	Základní nastavení regulace				Základní data			
2					Teplota ložnice	23.8 °C	Relé topné folie v ložnici	True
3	Hystereze		0.5 °C		Teplota obývací místnosti	23.9 °C	Relé topné folie v obývací místnosti	True
4	Teplota ložnice		30 °C		Osvětlení ložnice	False		
5	Teplota obývací místnosti		38 °C		Osvětlení obývací místnosti	True	Data změřena v	Sun May 22 00:49:16 2022
6	Prodleva měření		30 s				IP adresa	192.168.1.171

Obr. 6.6: Ukázka google tabulek ovládané pythonem

Jak je z obrázku 6.6, je zde možné nastavit jak samotnou prodlevu měření, tak i jednotlivé teploty místností a i samotnou velikost hystereze. Druhou částí tabulek jsou aktuální data změřena ze senzorů. Ty jsou pokaždé s novým měřením přepsány, aby uživatel měl stále aktuální hodnoty.

Výpis 6.2: Ukázka programu pro přepisování dat v google tabulkách

```
wks_sett.update('F2', '{teplota}°C'.format(teplota =
                                             DhtSensor1.temperature))
wks_sett.update('F3', '{teplota}°C'.format(teplota =
                                             DhtSensor2.temperature))
wks_sett.update('F4', '{stav}'.format(stav = relay[0]))
wks_sett.update('F5', '{stav}'.format(stav = relay[1]))
wks_sett.update('H2', '{stav}'.format(stav = relay[3]))
wks_sett.update('H3', '{stav}'.format(stav = relay[2]))
wks_sett.update('H5', '{time}'.format(time = local_time))
wks_sett.update('H6', '{IP}'.format(IP = getip()))
```

Závěr

Hlavním cílem práce bylo navržení modelu s prvky chytré domácnosti a následně pro řídící jednotku naprogramovat program, který tyto všechny prvky společně propojí a následně je bude řídit, popřípadě číst data ze senzorů. Toto všechno se podařilo implementovat a následně byla ještě odzkoušena komunikace pomocí internetu a to přes aplikaci telegram, kde samotný přenos je pouze od řídící jednotky k uživateli, nikoli naopak. V neposlední řadě byla také vytvořena obousměrná komunikace a to konkrétněji s google tabulkami, ze kterých byla umožněna i samotná úprava nastavených parametrů.

Celý projekt je samostatně funkční a to i bez připojení k internetu. Z důvodu pomalejší dynamiky systému byla zvolena žádaná hodnota teploty tak, aby bylo rychlejší pozorovat přepínání relé a samotný děj uvnitř modelu. Samotné senzory o dané přesnosti nám bohatě poslouží. Maximální možný předkmit a podkmit je odhadem pod hranicí $\pm 0.5C^\circ$, což ve výsledku dává kmitání teploty v rozmezí žádané hodnoty $\pm 1C^\circ$, o které bylo pojednáváno v kapitole 4.3.

Bohužel model nebyl odzkoušen pro zimní teploty ale pouze pro ty jarní, díky tomu, že byl vyroben až v letním semestru. To může mít za následek, že při zimních teplotách, se může výrazně maximální výtopná teplota dosti lišit od aktuální. Jelikož se jedná o model, kde střecha je libovolně odnímatelná tak zde vzniká plno možných prostor, kudy může teplo unikat.

Důvod vytvoření této práce, a nepoužití komerčního řešení, bylo že mě samotného mě zajímala problematika chytré domácnosti a zároveň prohloubit znalosti programování v jazyce Python a programování RPI. Samotný projekt aktuálně není až tak hluboko propracovaný jako například komerční řešení Home Assistant z kapitoly 1.2.2. S čímž se musí počítat, jelikož se nejedná o plnohodnotnou chytrou domácnost, ale pouze o model s prvky chytré domácnosti.

Literatura

- [1] Push-Button Manor: The Original Smart Home - Hotfoot Design. Specialist Web Design and Brand Agency in Lancaster, Lancashire, Cumbria Hotfoot Design [online]. Copyright ©2021 Hotfoot Design Limited, [cit. 27.10.2021]. Dostupné z URL:
<https://www.hotfootdesign.co.uk/white-space/push-button-manor-original-smart-home/>.
- [2] Profesionální chytrá domácnost i komerční objekt | Loxone. [online]. Copyright © 2021 Loxone Electronics GmbH. All rights reserved. [cit. 27.10.2021]. Dostupné z URL:
<https://www.loxone.com/cscz/>
- [3] Co je Home Assistant - TátaGeek.blog Vysvětlení od nováčka, který je o krok před vámi. TátaGeek.blog - Osobní blog jednoho taty [online]. Copyright © 2021 TátaGeek.blog [cit. 29.10.2021]. Dostupné z URL:
<https://tatageek.blog/2020/05/26/co-je-home-assistant/>
- [4] Getting Started with Home Automation and Home Assistant - James Ridgway. James Ridgway - James Ridgway [online]. Copyright © James Ridgway 2020 [cit. 28.10.2021]. Dostupné z URL:
<https://www.jamesridgway.co.uk/home-assistant/>
- [5] Visioforce Automation Hong Kong - Smart Home, home automation. Visioforce Automation Hong Kong - Home [online]. Copyright © Visioforce Hong Kong, All Rights Reserved. [cit. 28.10.2021]. Dostupné z URL:
<http://visioforce.com/smarthome.html>
- [6] Openhab vs Home Assistant - Find the Difference?. Electronics Hub - Tech Reviews | Guides & How-to | Latest Trends [online]. Copyright © 2021 ElectronicsHub.org [cit. 29.10.2021]. Dostupné z URL:
<https://www.electronicshub.org/openhab-vs-home-assistant/>
- [7] Home Assistant Vs Openhab 2019 - Home Automation. Home Automation - All Smart Home System and Gadgets [online]. Copyright © 2018. [cit. 29.10.2021]. Dostupné z URL:
<https://home-automations.net/home-assistant-vs-openhab-2019/>
- [8] Blocked. DigiKey Electronics - Electronic Components Distributor [online]. Dostupné z URL:
<https://www.digikey.com/catalog/en/partgroup/dht22-temperature-and-humidity-sensor/58084>

- [9] Tenis-cernovice.cz Bestsellery [online]. Copyright © [cit. 31.10.2021]. Dostupné z URL:
[<https://tenis-cernovice.cz/uploads/Dht22-digit%C3%A1ln%C3%AD-teplota-and-vlhkost-senzor-modul/1_43332.jpg>](https://tenis-cernovice.cz/uploads/Dht22-digit%C3%A1ln%C3%AD-teplota-and-vlhkost-senzor-modul/1_43332.jpg)
- [10] Raspberry Pi 4 Model B - 4GB RAM. RPishop.cz [online]. Copyright © Copyright 2021 RPishop.cz. [cit. 03.11.2021]. Dostupné z URL: [<https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html>](https://rpishop.cz/raspberry-pi-4b/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html)
- [11] Relační databáze vs. nerelační databáze: Čím se liší? | MasterDC. MasterDC – Specialisté na firemní IT infrastrukturu [online]. Copyright © 2021 Master Internet, s.r.o. [cit. 27.11.2021]. Dostupné z: <<https://www.master.cz/blog/relacni-databaze-nerelacni-databaze-jake-jsou-rozdily/>>
- [12] Relační datový model. Home [online] [cit. 27.11.2021]. Dostupné z: <<http://lucie.zolta.cz/index.php/iformacni-systemy-databaze/42-relacni-datovy-model>>
- [13] MongoDB: the application data platform | MongoDB [online] [cit. 29.11.2021]. Dostupné z: <<https://www.mongodb.com/databases/what-is-an-object-oriented-database>>
- [14] Objektová databáze – Wikipedie. [online]. Dostupné z: <https://cs.wikipedia.org/wiki/Objektov%C3%A1_datab%C3%A1ze>
- [15] MySQL – Wikipedie. [online] [cit. 29.11.2021]. Dostupné z: <<https://cs.wikipedia.org/wiki/MySQL>>
- [16] PostgreSQL – Wikipedie. [online] [cit. 29.11.2021]. Dostupné z: <<https://cs.wikipedia.org/wiki/PostgreSQL>>
- [17] Elektromagnetické relé - co je a jak funguje - Elektrokomponenty AZ. Elektrokomponenty AZ - Úvod [online]. Copyright © Elektrokomponenty AZ [cit. 29.11.2021]. Dostupné z: <<http://www.eshop.elektrokomponenty.cz/novinky/elektromagneticke-rele-co-je-a-jak-funguje/>>
- [18] Eelé modul 8-kanálový 5V pro arduino | dratek.cz. dratek.cz: VELKOOBCHOD, MALOOBCHOD S ARDUINEM [online]. Copyright © Copyright EC-LIPSERA s.r.o. [cit. 14.05.2022]. Dostupné z: <<https://dratek.cz/arduino/7187-rele-modul-8-kanalovy-5v-pro-arduino.html>>

- [19] CZ -HDG-instalace-pod panely-2021 web.pdf - Disk Google. [online] [cit. 29.11.2021]. Dostupné z: <https://drive.google.com/file/d/1NeIzrintj0nf2U_IotZ7gGV0CvTa4CvC/view>
- [20] Podlahové topení pod plovoucí podlahu – Heat Home s.r.o.. Topné fólie a topné rohože HEAT DECOR – ESHOP – Heat Home s.r.o. [online]. Copyright © 2021 Heat Home s.r.o. [cit. 29.11.2021]. Dostupné z: <<https://heat-home.cz/podlahove-topeni-pod-plovouci-podlahu/>>
- [21] Raspberry Pi 4 Model B Default GPIO Pinout with PoE Header - Documents - Raspberry Pi - element14 Community. element14 Community [online]. Copyright © 2009 [cit. 23.12.2021]. Dostupné z: <<https://community.element14.com/products/raspberry-pi/w/documents/4317/raspberry-pi-4-model-b-default-gpio-pinout-with-poe-header>>
- [22] Data – Disk Google. [online][cit. 30.12.2021]. Dostupné z: <https://drive.google.com/drive/folders/1StyZXYPJxtpWjZkDXmo8Qf0rgl_MlhB3?usp=sharing>

Seznam symbolů a zkratek

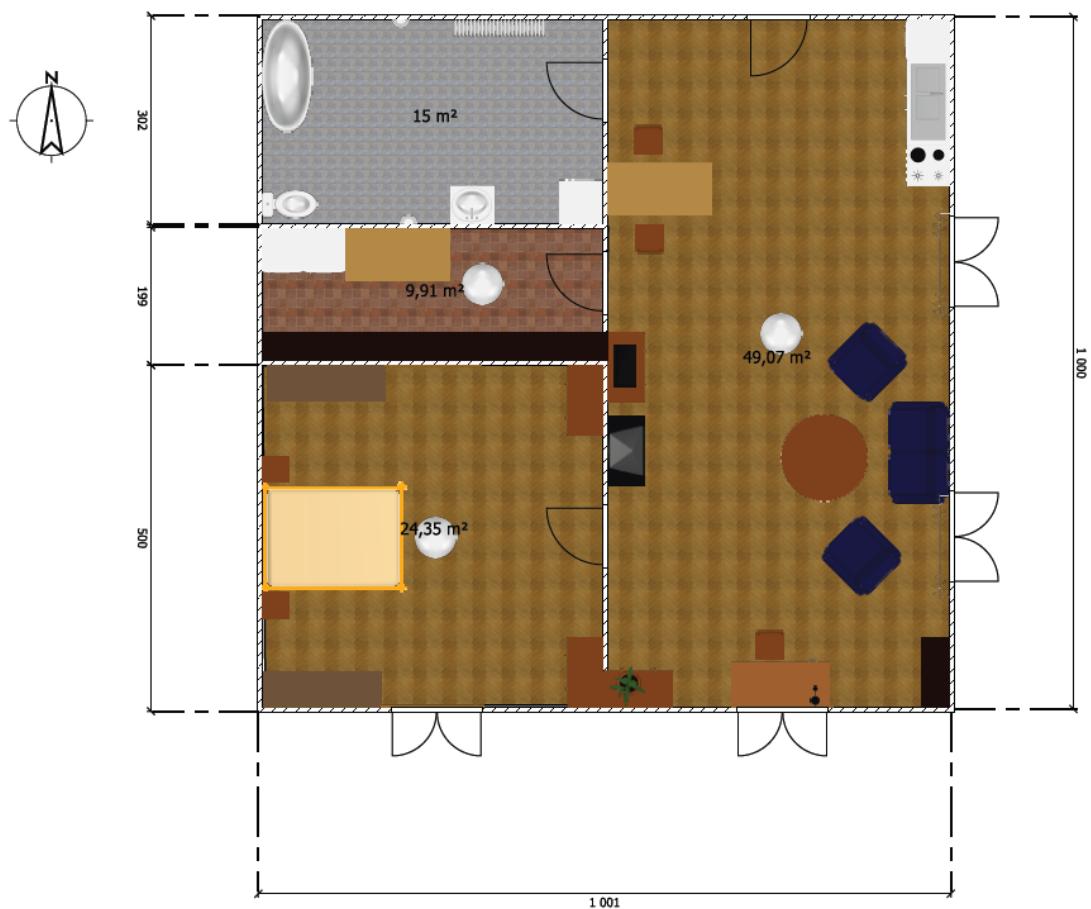
RPI	Raspberry PI
PCB	Printed Circuit Board - Tištěný spoj
NAS	Network Attached Storage - Datové úložiště
VPS	Virtuální privátní server
VDS	Virtuální dedikovaný server
GND	Ground - uzemnění
DHCP	Dynamic Host Configuration Protocol
SSH	Secure Shell
L	Fázový vodič
N	Nulovací vodič

Seznam příloh

A Model domu	55
A.1 Plánek	55
B Měření vytopení místnosti pomocí topné folie	56
B.1 RPI 4B GPIO Header	56
B.2 Program použitý při měření přechodové charakteristiky	57
B.3 Výkyny venkovní teploty při měření	58
C Programy	59
C.1 Program pro řízení osvětlení	59
C.2 Práce s databází na serveru	60
C.3 Program pro vypínače	61
C.4 Program pro teplotní senzory	62
D Obsah elektronické přílohy	63

A Model domu

A.1 Plánek



Obr. A.1: Plánek 3D modelu

B Měření vytopení místnosti pomocí topné folie

B.1 RPI 4B GPIO Header

Raspberry Pi 4 B J8 GPIO Header

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I ² C)		DC Power 5v	04
05	GPIO03 (SCL1, I ² C)		Ground	06
07	GPIO04 (GPCLK0)		(TXD0, UART) GPIO14	08
09	Ground		(RXD0, UART) GPIO15	10
11	GPIO17		(PWM0) GPIO18	12
13	GPIO27		Ground	14
15	GPIO22		GPIO23	16
17	3.3v DC Power		GPIO24	18
19	GPIO10 (SPI0_MOSI)		Ground	20
21	GPIO09 (SPI0_MISO)		GPIO25	22
23	GPIO11 (SPI0_CLK)		(SPI0_CE0_N) GPIO08	24
25	Ground		(SPI0_CE1_N) GPIO07	26
27	GPIO00 (SDA0, I ² C)		(SCL0, I ² C) GPIO01	28
29	GPIO05		Ground	30
31	GPIO06		(PWM0) GPIO12	32
33	GPIO13 (PWM1)		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Raspberry Pi 4 B J14 PoE Header

01	TR01		TR00	02
03	TR03		TR02	04

Pinout Grouping Legend

- Inter-Integrated Circuit Serial Bus Serial Peripheral Interface Bus
- Ungrouped/Un-Allocated GPIO Universal Asynchronous Receiver-Transmitter
- Reserved for EEPROM

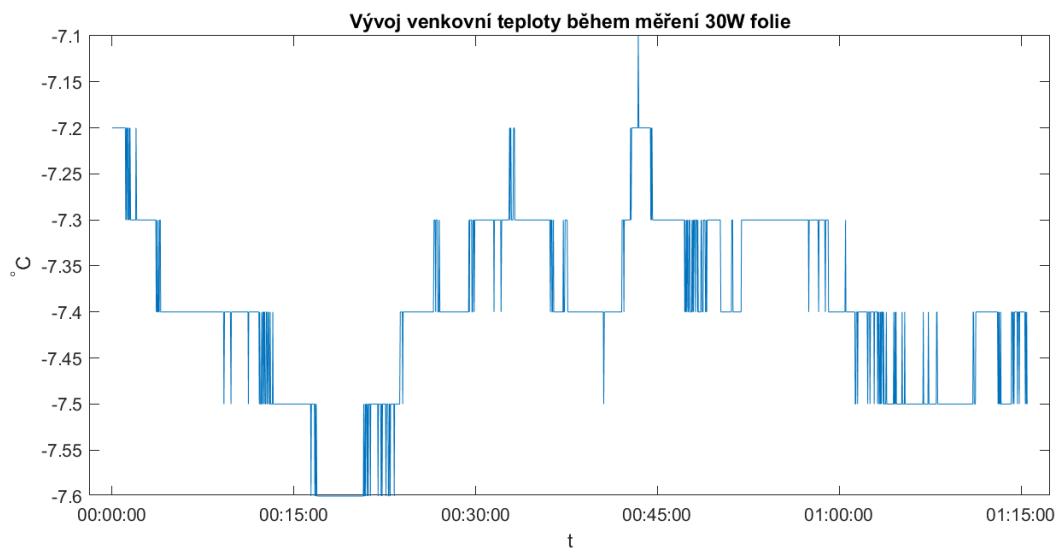
Obr. B.1: PINOUT u RPI 4B [21]

B.2 Program použitý při měření přechodové charakteristiky

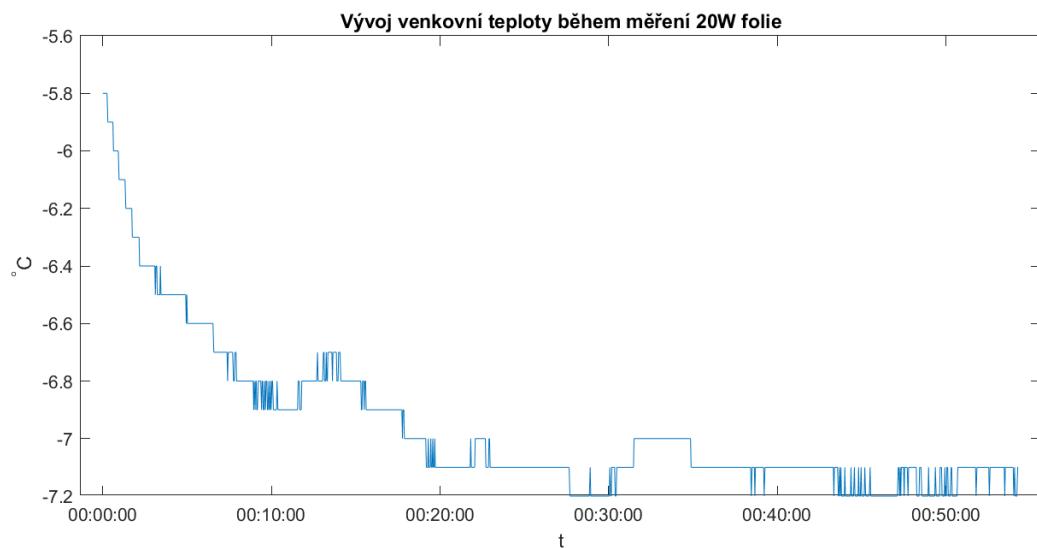
Výpis B.1: Program použitý pro měření vytopení místnosti pro danou folii

```
import csv  
import time  
import board  
import adafruit_dht  
def get_time():  
    seconds = time.time()  
    return time.ctime(seconds)  
dhtDevice1 = adafruit_dht.DHT22(board.D24, use_pulseio=False)  
dhtDevice2 = adafruit_dht.DHT22(board.D23, use_pulseio=False)  
header = ['time', 'temp_1', 'temp_2', 'type_of_heating_foil']  
with open('30w.csv', 'w', encoding='UTF8', newline='') as f:  
    writer = csv.writer(f)  
    writer.writerow(header)  
  
while True:  
    try:  
        temperature_c1 = dhtDevice1.temperature  
        temperature_c2 = dhtDevice2.temperature  
        data = [get_time(), temperature_c1, temperature_c2,  
                'HD3025G_30W']  
        print("Time: {time}, temp_in: {temp}, temp_out:  
               {temp_o}").format(time = get_time(),  
                           temp = temperature_c1,temp_o = temperature_c2))  
        with open('30w.csv', 'a', encoding='UTF8',  
                  newline='') as f:  
            writer = csv.writer(f)  
            time.sleep(1)  
            writer.writerow(data)  
    except RuntimeError as error:  
        print(error.args[0])  
        time.sleep(1)  
        continue  
    except Exception as error:  
        dhtDevice.exit()  
        raise error  
    except KeyboardInterrupt:  
        dhtDevice.exit()  
        print('exiting script')
```

B.3 Výkyvy venkovní teploty při měření



Obr. B.2: Výkyvy venkovní teploty při měření 30W folie



Obr. B.3: Výkyvy venkovní teploty při měření 20W folie

C Programy

C.1 Program pro řízení osvětlení

Výpis C.1: Řízení osvětlení

```
from datab_con import *          1
import random                    2
import board                     3
import RPi.GPIO as GPIO          4
                                5
                                6
class Light(database):          7
    def __init__(self, user, password, host, name_of_database,      8
                  port, location, actual_state, gpio_port):        9
        super().__init__(user, password, host,                      10
                           name_of_database, port)                   11
        self.location = location                         12
        self.actual_state = actual_state                13
        self.gpio_port = gpio_port                      14
                                15
    def update_datab_state(self):                 16
        if self.actual_state == 1:                  17
            GPIO.setup(self.gpio_port, GPIO.OUT)       18
        else:                                     19
            GPIO.setup(self.gpio_port, GPIO.IN)        20
                                21
        sql = "UPDATE `Bakalarka`.`lights` SET state =      22
               {state} WHERE location={light_name}{}".format( 23
                                                       state=self.actual_state, 24
                                                       light_name=self.location) 25
        try:                                     26
            self.insert_to_database(sql)           27
            return 1                            28
        except:                                 29
            return 0                            30
```

C.2 Práce s databází na serveru

Výpis C.2: Práce s databází

```
1 import mysql.connector
2 class database:
3     def __init__(self, user, password, host, database, port):
4         self.user = user ## @var username
5         self.password = password
6         self.host = host
7         self.database = database
8         self.port = port
9     def conn_to_database(self):
10        print("Attempting to connect a user: " + self.user)
11        try:
12            global my_database
13            my_database = mysql.connector.connect(
14                host=str(self.host),
15                user=str(self.user),
16                password=str(self.password),
17                database = str(self.database),
18                port = str(self.port)
19            )
20        except mysql.connector.Error as e:
21            print(e)
22    def insert_to_database(self,sql):
23        cursor = my_database.cursor()
24        cursor.execute(sql)
25        my_database.commit()
26    def select_from_database(self,sql):
27        try:
28            cursor = my_database.cursor()
29            cursor.execute(sql)
30            return cursor.fetchall()
31        except:
32            return 0
33    def close_database(self):
34        try:
35            my_database.close()
36            return 1
37        except:
38            return 0
```

C.3 Program pro vypínače

Výpis C.3: Vypínač

```
from datab_con import *
import socket
import RPi.GPIO as GPIO
class Switch(database):
    def __init__(self, user, password, host, name_of_database,
                 port, location, actual_state, gpio_port):
        super().__init__(user, password, host,
                         name_of_database, port)
        self.location = location
        self.actual_state = actual_state
        self.gpio_port = gpio_port

    def get_actual_state(self):
        sql = "SELECT * FROM switches WHERE location={name_switch}" .format(name_switch=self.location)
        online_state_switch = self.select_from_database(sql)
        self.actual_state = int(online_state_switch[0][2])

    def update_database_state(self):
        var = 0
        sql = "UPDATE switches SET state = {state} WHERE
              location={switch_name}" .format(state=
              str(self.actual_state), switch_name=self.location)
        try:
            self.insert_to_database(sql)
            return 1
        except:
            return 0
```

C.4 Program pro teplotní senzory

Výpis C.4: Teplotní senzory

```
from datab_con import *
import board,adafruit_dht
dhtDevice_obyvak = adafruit_dht.DHT22(board.D20,
                                         use_pulseio=False)
dhtDevice_loznice = adafruit_dht.DHT22(board.D21,
                                         use_pulseio=False)
class DhtSensor(database):
    def __init__(self, user, password, host, name_of_database,
                 port, sensorID, temperature):
        super().__init__(user, password, host,
                         name_of_database, port)
        self.sensorID = sensorID
        self.temperature = temperature
    def mess_temperature(self):
        temp,divide = 0
        for i in range(10):
            if self.sensorID == "Obyvak_DHT" ||
               self.sensorID == "Loznice_DHT":
                try:
                    temp += round(dhtDevice_obyvak.temperature
                                  ,3)
                    divide += 1
                except:
                    temp = 0, divide = 1
            if divide != 0:
                self.temperature = round(temp / divide, 3)
            else:
                self.temperature = round(temp, 3)
    def add_temp_database(self):
        sql = "INSERT INTO `Bakalarka`.'temp_in' ('sensor_id',
          'temp') VALUES ('{}',{})".format(self.sensorID,
                                           self.temperature)
        try:
            self.insert_to_database(sql)
            return 1
        except:
            return 0
```

D Obsah elektronické přílohy

```
/.....kořenový adresář přiloženého archivu
├── programs ..... python programy pro RPI
│   ├── main.py
│   ├── datab_con.py
│   ├── switch.py
│   ├── temperature.py
│   ├── light.py
│   └── service_account_google.json
└── data ..... naměřené charakteristiky a data z regulace
    ├── 20w.csv
    ├── 20w_nova.csv
    ├── 30w.csv
    ├── 30w_nova.csv
    ├── regulace.csv
    ├── obyvak_prechodovka_16-4_v2.csv
    └── loznice_prechod_16-4.csv
├── BP_Fuxa_Dominik_220977_2022.pdf ..... bakalářská práce v elektronické verzi
├── program-report.pdf ..... kompletní programy v elektornické podobě
└── demonstrace_osvetleni.mp4 ..... demostrace osvětlení na 230V
```