



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**CHYTRÁ DOMÁCNOST: DIGITÁLNÍ TERMOSTATICKÁ
HLAVICE ŘÍZENÁ PŘES WIFI**

SMART HOME: DIGITAL THERMOSTATIC VALVE HEAD CONTROLLED VIA WIFI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN SMEJKAL

VEDOUcí PRÁCE

SUPERVISOR

Ing. MATERNA ZDENĚK, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Smejkal Jan**
Program: Informační technologie
Název: **Chytrá domácnost: Digitální termostatická hlavice řízená přes WiFi**
Smart Home: Digital Thermostatic Valve Controlled via WiFi
Kategorie: Vestavěné systémy

Zadání:

1. Proveďte rešerši existujících řešení modifikujících levné digitální radiátorové hlavice pro jejich řízení přes WiFi, za pomoci modulů založených na ESP8266.
2. Vyberte vhodnou hlavici pro modifikaci.
3. Navrhněte úpravu, včetně napájení, způsobu řízení a odesílání naměřených dat.
4. Realizujte modifikaci hlavice a implementujte řídicí systém.
5. Ověřte dlouhodobým testem funkčnost řešení.
6. Zdrojové kódy a dokumentaci publikujte na GitHubu.
7. Vytvořte stručný plakát nebo video prezentující vaši práci, její cíle a výsledky.

Literatura:

- Dle doporučení vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1-3.

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

Vedoucí práce: **Materna Zdeněk, Ing., Ph.D.**

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

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Tato práce se zabývá chytrým vytápěním v domácnostech s radiátory. Cílem je vytvořit levnou hlavici pro vytápění a pomocí WiFi a MQTT ji integrovat do systému Home Assistant. Zaměřil jsem se na zachování vzhledu, funkcí hlavice a uživatelskou přívětivost. Home Assistant umožňuje zobrazování grafů díky InfluxDB a Grafaně. Dále dokáže díky programu v nástroji Node-RED ztlumit vytápění v době nepřítomnosti uživatele a plánovat časový rozvrh ovládání ventilů. Celková cena hlavice je pod 400 Kč. Regulaci teploty zajišťoval predikční kontrolér, který byl nahrazen PID kontrolérem. Ten je schopen přesnější regulace s rozdílem maximální a minimální teploty okolo 2 °C. Režim boost hlavice se dá ovládat i z Home Assistanta. Výsledná práce je dobrým základem pro nízkonákladový chytrý systém vytápění.

Abstract

This bachelor thesis deals with smart home heating in households with radiators. The goal is to design a cheap digital heating head and to integrate it via WiFi and MQTT into the Home Assistant system. I have focused on keeping good design, head functions and a good user experience. Home Assistant enables showing graphs using InfluxDB and Grafana. Using a program in tool Node-RED enables it to lower heating temperature at the times of user absence and to schedule heating temperatures. The price of the complete unit is under 400 CZK. Temperature regulation was provided by a prediction controller, but it was replaced by a PID controller. The controller can regulate with the difference of maximal and minimal temperature of around 2 °C. The user can control the boost mode of the head via a physical button or using the Home Assistant system. The resulting product is a good basis for a low-cost smart heating system.

Klíčová slova

regulace vytápění, ESP8266, WiFi, chytrá domácnost, Eqiva EQ-3 N, MQTT, Home Assistant, InfluxDB, Grafana, Node-RED

Keywords

heating control, ESP8266, WiFi, smart home, Eqiva EQ-3 N, MQTT, Home Assistant, InfluxDB, Grafana, Node-RED

Citace

SMEJKAL, Jan. *Chytrá domácnost: Digitální termostatická hlavice řízená přes WiFi*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Materna Zdeněk, Ph.D.

Chytrá domácnost: Digitální termostatická hlavice řízená přes WiFi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zdeňka Materny, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Jan Smejkal
10. května 2021

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Zdeňkovi Maternovi, Ph.D. za užitečné rady, nasměrování a ochotu pomoci s kontrolou této práce. Další poděkování patří rodině Gerzičákových, která mi umožnila nainstalovat a testovat hlavici pro vytápění v jejich domácnosti.

Obsah

1	Úvod	2
2	Regulace vytápění	3
2.1	Systém vytápění v budovách s radiátory	3
2.2	Vytápění radiátory	5
2.3	Algoritmy pro vytápění	7
3	Výběr technologií chytrého vytápění	15
3.1	Komunikace s centrální jednotkou	15
3.2	ESP8266	19
3.3	H můstek	21
3.4	Teplotní senzor DS18B20	22
3.5	Světelný senzor	23
3.6	DHCP server a statický záznam	23
3.7	Home Assistant	24
3.8	Node-RED	26
3.9	InfluxDB a Grafana	27
4	Návrh úpravy hlavice	28
4.1	Předpokládané vlastnosti upravené hlavice	28
4.2	Ovládací jednotka	29
4.3	Ovládání motoru hlavice	30
4.4	Logika vytápění	30
5	Návrh systému vytápění	31
5.1	Centrální jednotka	31
6	Implementace	32
6.1	Úprava hlavice	32
6.2	Řídící software hlavice	36
6.3	Centrální jednotka	37
7	Testování	41
7.1	Kontrolér s predikcí	42
7.2	PID kontrolér	44
8	Závěr	48
	Literatura	49

Kapitola 1

Úvod

Technologie pro chytřejší ovládání zařízení v domácnosti se vylepšují neúnavným tempem. Na trhu existuje široký výběr chytrých zařízení a sad zařízení. Motivací výrobců je zisk, a tak se často snaží svého zákazníka udržet ve svém ekosystému navrhnutím vlastních centrálních systémů domácí automatizace.

Poslední dobou se objevují chytrá zařízení firem, které používají pro komunikaci Zigbee. To s sebou přináší spoustu výhod, ale i nevýhod. Využití jednoho standardu komunikace odstraňuje nutnost kupovat nový hub pro připojení nových zařízení do domácí sítě. Problémem zůstává to, že využití Zigbee pro komunikaci neznamena kompatibilitu se stejnými centrálními systémy domácnosti. Může tak nastat situace, kdy pro ovládání zařízení tří různých výrobců bude potřeba více než jeden systém ovládání. U proprietárních řešení mohou být pádnější obavy o bezpečnost a dobu poskytování aktualizací (bezpečnostních i funkčních). U řešení velkých firem jako je Google Home může uživatelům vadit, že má poskytovatel možnost využít informace ze senzorů pro reklamní účely.

V současné době jsou na trhu dostupné sady domácí automatizace, které zahrnují i hlavice pro vytápění termostatickými ventily. Jsou však velmi drahé a možnosti propojení s jinými systémy domácí automatizace jsou velmi omezené [40].

Cílem této práce je vyrobit prototyp upravené levné digitální hlavice, která bude zajišťovat kvalitnější regulaci vytápění a připojení do systému chytré domácnosti. Toto připojení zajistí pomocí otevřených běžně používaných standardů, aby uživatel nemusel platit měsíční poplatky ani vysoké vstupní poplatky a nebyl uvázan k ekosystému jednoho či několika výrobců.

Dalším cílem je zajištění a konfigurace centrálního systému, jenž umožňuje ovládání hlavic. Centrální jednotka zajistí i zaznamenávání a zobrazování dat z jednotlivých hlavic a umožní nastavit časový plán vytápění včetně snížení teploty vytápění v momentě, kdy uživatel není přítomen v místnosti či domácnosti. Cílem je vybrat bezpečný systém otevřený k zařízením různých výrobců s odlišnými způsoby komunikace.

Kapitola 2

Regulace vytápění

V následující kapitole je shrnuta problematika vytápění a regulace v budovách s radiátory z pohledu systému vytápění jako celku. Další kapitola se zabývá regulací jednotlivých radiátorů.

2.1 Systém vytápění v budovách s radiátory

Na základě [54] vytápění v budovách s radiátory představuje systém s jedním či více zdroji tepla. Přenos tepla od zdroje či zdrojů tepla do radiátorů zajišťují rozvody, kterými cirkuluje topné médium (obvykle voda).

Systém rozvodů tepla má dvě části: teplou a studenou. Teplá část je rozvedena od zdrojů tepla k radiátorům, kde se teplo předá do vytápěných místností a ochlazené topné médium se vrací druhou částí rozvodů zpět.

Pokud zdroj tepla slouží pouze pro byt či patro budovy, jedná se o etážové vytápění. Pokud je však zdroj tepla sdílený více patry budov či dokonce celými budovami, jedná se o dálkové vytápění. U dálkového vytápění může být zdroj umístěn například v kotelně panelového domu nebo i ve vzdálené teplárně.

Řízení tohoto systému má v případě etážového vytápění obvykle na starosti centrální termostat. Termostatů může být více nebo jejich funkci může nahradit systém chytrých hlavic. Termostat na základě teploty v místnosti, kde je umístěn, zapne centrální zdroj tepla, který vytápí celou budovu. Pokud jsou na radiátorech osazeny termostatické hlavice, regulují teplo, které jednotlivé radiátory vydávají.

Problém může nastat v situaci, kdy některé z hlavic budou potřebovat vytápět, ale centrální termostat či termostaty tuto potřebu nebudou mít. Jako příklad může posloužit otevření a následné zavření okna v jedné z místností bez centrálního termostatu, odlišné časy vytápění různých místností, místnost s většími tepelnými ztrátami, příliš malé radiátory, vytápění z jiného zdroje (pečení či vaření, zapnutá sušička) a mnoho dalších.

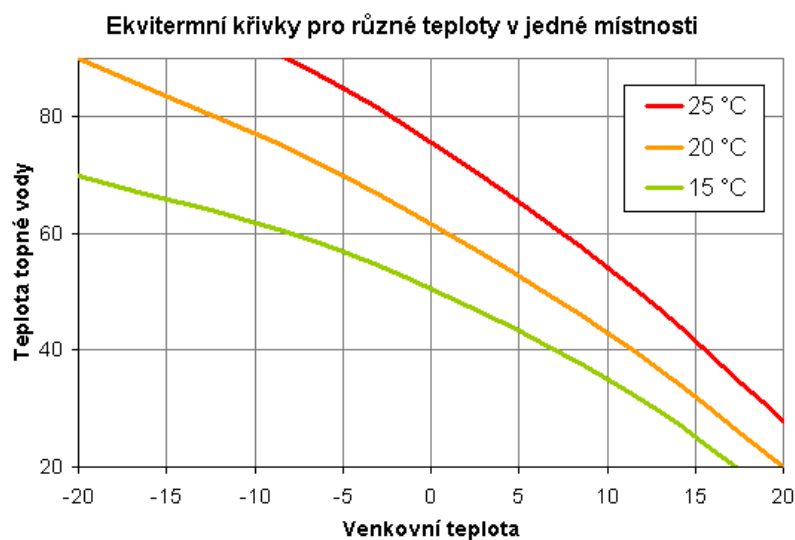
Tuto situaci pomůže vyřešit připojení ventilů k termostatům či napřímo ke zdroji tepla. Když některý z ventilů bude potřebovat vytápět, odešle zprávu centrálnímu termostatu či kotli, aby se zapnul. Tomuto řešení se anglicky říká „call for heat“.

2.1.1 Ekvitermní regulace

Některé systémy vytápění obsahují senzory venkovní teploty. Na základě této teploty upravují teplotu topného média, tím se vytápění stává ekvitermním. Ekvitermní vytápění zajišťuje vyšší účinnost vytápění v případech, kde není potřeba vytápět plným výkonem.

Průchod teplé vody rozvody provází tepelné ztráty závislé na teplotě vody. Kondenzační kotle dosahují velmi vysoké účinnosti při teplotě topného média okolo 40 až 60 °C, takže nižší teplota topného média dále napomáhá k vyšší účinnosti. Kromě vyšší účinnosti nabízí i vyšší tepelný komfort, jelikož dochází k menšímu kolísání teplot v místnosti.

Ekvitermní vytápění pro svoji funkci využívá ekvitermních křivek, které jsou závislé na venkovní teplotě a požadované teplotě v místnostech. Na obrázku níže je k nahlédnutí příklad ekvitermní křivky [51].



Obrázek 2.1: Příklad ekvitermních křivek. Vodorovná osa představuje venkovní teplotu, svislá osa teplotu topné vody a barevné křivky představují požadovanou teplotu v místnosti či budově. Zdroj: [51].

Starší radiátory mívají klasický kohoutkový ventil, který umožňuje regulovat pouze intenzitu ohřívání jednotlivých místností, a tak zhruba kompenzovat rozdíly v úbytku tepla z místnosti vůči velikosti radiátoru. Toto řešení je pro mnohé uživatele nedostačující a bývá nahrazeno ventilem termostatickým. U nových vytápěcích systémů je použití termostatických ventilů povinné.

Klasický kohoutkový ventil je nekompatibilní s ventilem pro termostatické hlavice. Pokud má uživatel staré kohoutkové ventily a chce na ně připevnit termostatické hlavice, bude je muset nejprve nechat vyměnit topenářem za termostatické.

2.2 Vytápění radiátory

Základní hlavice pro termostatický ventil uživateli umožňuje pomocí otáčení hlavice nastavit požadovanou teplotu. Pro orientaci uživatele slouží čísla 1 až 6, kde 3 obvykle znamená teplotu 20 či 21 °C [16].

Regulace vytápění tedy probíhá pomocí rozpínání tepelně citlivého vosku či kapaliny na základě rozdílu aktuální teploty od cílové. Rozpínání této kapaliny otevírá či zavírá ventil.



Obrázek 2.2: Termostatický ventil. Zdroj: [33].

Toto řešení neumožňuje automaticky měnit cílovou teplotu v noci a přes den či například v době, kdy místnost nebývá využívána.

Má však své výhody. Mezi hlavní výhody mechanických hlavic patří jednoduchost použití, bezhlučnost a absence potřeby napájení ať už ze sítě či bateriemi, které potřebují jednou za čas vyměnit.



Obrázek 2.3: Fotografie klasické mechanické termostatické hlavice. Zdroj: [16].

Další nevýhodou je, že klasická hlavice si neporadí se zbytečným vytápěním v případě, kdy se uživatelé rozhodnou otevřít v místnosti okno. Ventil také kvůli ochraně proti zanesení vodním kamenem potřebuje jednou za čas plně otevřít a zavřít.

Na trhu se však již několik let objevují digitální hlavice, které se dají jednoduše vyměnit místo klasických termostatických hlavice. Ty s sebou přinášejí mnoho řešení dříve zmíněných nedostatků a k tomu navíc mohou nabízet i funkce navíc. Mezi ně patří možnost ovládání telefonem či napojení do centrálního systému chytré domácnosti, s jehož využitím se může vytápění jednotlivých místností řídit například přítomností telefonů v domácí WiFi síti či jinými vlivy.

Digitální hlavice však mají svá negativa. Hlavním je hlučnost běžně prodávaných jednotek: když hlavice zavírá či otevírá ventil, dělá při tom hluk a vibrace, které se přenášejí do samotného radiátoru. Přes den tato skutečnost obvykle nevádí, avšak večer může tento hluk vadit citlivějším obyvatelům v situaci, kdy by hlavice byla nainstalována v místnosti, kde se snaží usnout.

Základní digitální hlavice jsou finančně dostupné: jedna hlavice EQ-3 N je dostupná za cenu nižší než 300 Kč [10]. Pomocí tlačítek či otočných koleček umožňují nastavit požadovanou teplotu a plánovat časový rozvrh vytápění. Často se u těchto hlavice objevují další dvě funkce. První z nich je ochrana ventilu přes zanesení vodním kamenem. Druhá spočívá v detekci otevřeného okna tak, aby hlavice zbytečně neotevírala ventil a tím pádem nevytápěla.

Problém však nastává, pokud by si uživatel chtěl pořídit sadu chytrých termostatických hlavice. To znamená, že hlavice umožňují připojení do systému chytré domácnosti, který umožňuje jednoduché ovládání z počítače či přes mobilní aplikaci s chytrým nastavováním a plánováním vytápění.

Využitím nabídky [40] obchodu CZC.CZ lze pořídit sadu výrobce Honeywell, která obsahuje tři hlavice, centrální jednotku, ovládací panel a bezdrátovou relé jednotku. To vše za cenu okolo 9000 Kč. Umožňuje ovládat vytápění i pomocí aplikace pro telefon. Ovládací

system této sady je zaměřen primárně na vytápění, jeho kompatibilita se zařízeními třetí strany je mizivá.



Obrázek 2.4: Fotografie sady Honeywell Evohome. Zdroj: [37].

Tato práce se snaží vyrobit hlavici se systémem, který je široce kompatibilní s dalšími zařízeními a zároveň je výrazně levnější než hotová sada.

2.3 Algoritmy pro vytápění

V následujících podkapitolách se nachází příklady implementace regulace vytápění jednotlivých radiátorů pomocí digitálních hlavice a jejich srovnání.

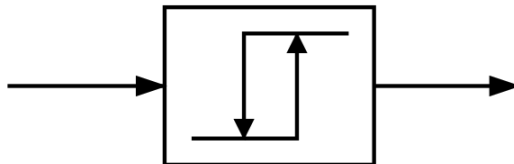
2.3.1 Bang-bang kontrolér

Dle [52] Bang-bang kontrolér, známý též jako on-off kontrolér, je velmi jednoduchý na výrobu, není třeba žádné elektroniky.

Běžně se objevuje například v zásobníkových ohřivačích vody, rychlovarných konvicích, starších termostatech pro vytápění či klimatizaci, u plynových topidel do místnosti (laickým výrazem též známé jako vafky) a v mnoha dalších zařízeních.

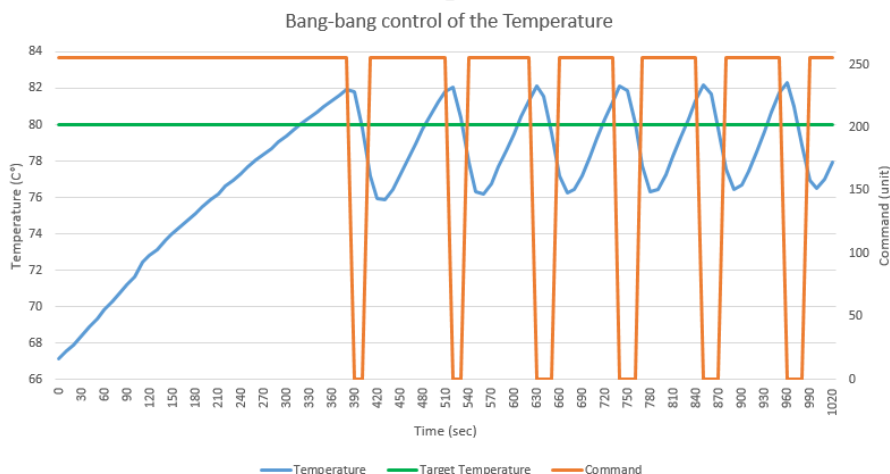
Má dva stavy a přepíná mezi nimi na základě zpětné vazby (také známé jako sledované veličiny). Pokud je zpětná vazba o určité množství nižší než požadovaná, kontrolér přepne na zapnutý stav. Pokud je o určité množství vyšší než zpětná vazba, kontrolér přepne na vypnutý stav. Přechodová množství nad a pod zpětnou vazbou mohou být odlišná. Rozdíl přechodového množství nad zpětnou vazbou vůči množství pod zpětnou vazbou je známý jako hystereze a dále v této práci bude používáno toto označení.

Princip této funkcionality je zobrazen i v symbolu, se kterým se používá v diagramech. Je dostupný pod odstavcem, byl převzat z [52]. Vodorovná osa znázorňuje zpětnou vazbu a svislá osa představuje dvoustavový výstup (zapnuto či vypnuto).



Obrázek 2.5: Symbol pro bang-bang kontrolér. Zdroj: [52].

Na obrázku níže, který byl získán z [25], je vidět zásadní nevýhoda takto jednoduchého systému: pro mnoho případů použití je nedostatečný či nevhodný. Z obrázku níže je vidět časový průběh ovládané veličiny, který připomíná zuby od pily. V případě znázorněném na obrázku tato situace nebyla problémem, cíl byl ohřát pokrm na 80 °C a maximální chyba 6 °C se dala zanedbat.



Obrázek 2.6: Graf časového průběhu zpětné vazby pro bang-bang kontrolér. Proměnné: aktuální teplota (modře), cílová teplota (zeleně), výstup (oranžový, zapnuto odpovídá 255, vypnuto odpovídá 0). Zdroj: [25].

Takto velké kolísání teploty by v případě vytápění domácnosti mohlo být nepříjemné. V jednom momentě by uživateli byla moc zima, o chvíli později moc teplo a stále dokola. Hystereze se dá snížit pro dosažení menšího kolísání teplot, avšak to s sebou přináší další problémy.

V případě této práce by zmíněné řešení bylo pro regulaci teploty nedostatečné zejména z těchto důvodů: velké výkyvy od cílové teploty a zbytečně vysoká náročnost na energii pro ovládání ventilu. S častými změnami polohy ventilu je také spojená hluchost, jelikož by tento kontrolér prováděl plné otevření či zavření ventilu namísto částečného, které termostatický ventil umí.

Další přidružený problém nastane, když je příliš malý rozsah mezi teplotami, kde se přepíná výstup. Tento rozsah je také známý jako hystereze. To má za následek častější přepínání

výstupu, které může způsobit příliš výrazné opotřebení, hluk a sníženou výdrž na baterii, pokud je využita pro napájení.

2.3.2 Kontrolér inspirovaný klasickým mechanickým

Klasická termostatická hlavice otevírá a zavírá ventil na základě aktuální teploty. Využívá tepelně rozpínavé kapaliny a možnosti nastavit si požadovanou teplotu volbou odpovídajícího čísla.

Implementace takového stylu ovládání je jednoduchá. Nejprve se zjistí aktuální poloha ventilu, následně se ventil otevírá či zavírá dle aktuální teploty. Pokud je aktuální teplota vyšší než cílová, topení je plně zavřeno. Pokud je aktuální teplota nižší než cílová, ventil se otevře do polohy odpovídající tomu, jak velký je rozdíl od požadované teploty. To znamená, že takový způsob ovládání v momentě dosažení cílové teploty má plně zavřený ventil.

Podobá se proporciální části PID kontroléru.

2.3.3 Kontrolér s predikcí budoucí teploty

Tato varianta rozšiřuje předešlý algoritmus inspirovaný klasickým ventilem o predikci budoucích hodnot. Na základě predikované teploty se o trochu změní poloha ventilu pro dosažení a uchování cílové teploty.

Tento princip byl inspirován z [44] pro účely první verze regulace vytápění, která byla nahrazena PID.

Predikce probíhá tak, že kontrolér nejdříve zprůměruje směry změn teploty z posledního určitého počtu záznamů teploty (řekněme N). Následně vynásobí tímto průměrem změn teplot aktuální teplotu a vynásobí ji multiplikační konstantou. Tím vznikne predikovaná teplota. Čím větší tato konstanta je, tím více do budoucna se snaží odhadovat teplotu. Pokud je příliš velká, ztrácí na přesnosti. Pokud je příliš malá, nemusí podchytit situaci, kdy je radiátor rozžhavený a sám bez dalšího přísunu tepla by místnost vytopil o například $1,5\text{ }^{\circ}\text{C}$.

Toto řešení si po správném nastavení ovládacích konstant lépe poradí s udržením požadované teploty. Nepotřebuje znát absolutní pozici ventilu, pouze jím posunuje. Tato vlastnost s sebou přináší problém vyvážení dostatečně rychlého vytápění a přetápění v momentě, kdy radiátor je ohřátý a již nepotřebuje mít otevřený přívodní ventil. Doba, po kterou ohřátý radiátor vydrží dále vytápět bez přívodu horké vody, může být i přes deset minut. Takto dlouho dopředu plánovat může být velmi obtížné.

2.3.4 PID kontrolér

Na základě [13] a [39] je PID složitější než dříve zmíněné kontroléry, přesto je velmi používaný, zejména v industriální sféře. Uplatnění nachází zejména v regulaci teploty, rychlosti, toku kapalin a podobně. Část kódu práce zajišťující PID regulátor byla realizována s využitím knihovny [13].

Hlavní předností je v případě odladěného systému výrazné zvýšení přesnosti dosažení cílového stavu a schopnost samočinně vylepšit ovládací konstanty pro konkrétní zařízení, které PID kontrolér ovládá. K samočinnému vylepšení ovládacích konstant slouží PID autotune, jeho princip funkce je vysvětlen dále v práci.

Následující část o složkách PID kontroléru byla převzata z [41]. Tento kontrolér obsahuje tři hlavní složky, kde správným nastavebním vah jednotlivým složkám vzniká odladěný

system. Tyto složky jsou: proporcionální (P), integrální (I) a derivační (D). První písmena tvoří zkratku PID, pod kterou je tento kontrolér znám.

V některých případech se u některé ze složek nastaví váha rovna nule. Tím se změní druh kontroléru. Například, pokud by byla pro složku D nastavena nula, stane se z PID kontroléru PI kontrolér.

Běžně používaný přístup k regulaci pomocí PID či podobného kontroléru využívá proportional on measurement (česky proporcionální výpočet na základě chyby, dále PonE, jelikož český název se téměř vůbec nepoužívá). V následujících odstavcích se nachází vysvětlení jednotlivých složek PID kontroléru v případě PonE. Srovnání s alternativním způsobem PonM se nachází o několik odstavců níže.

Proporcionální složka (P)

Proporcionální složka vyjadřuje rozdíl mezi cílovou hodnotou a hodnotou zpětné vazby (též známé jako sledované veličiny). Pokud je zpětná vazba nižší než cílová hodnota, bude kladná a naopak. Tento rozdíl hodnot je následně vynásoben vahou pro proporcionální složku a vzniká výsledek P.

Integrální složka (I)

Integrální složka znamená součet rozdílů od cílové hodnoty z minulosti. Funguje tedy podobně jako integrál v matematice. Tato složka reaguje pomaleji na změny, ale postupně se časem výrazně změní a způsobí změnu zpětné vazby blíže k či dokonce přes cílovou hodnotu.

Takto spočítaný integrál je následně pronásoben příslušnou vahou a vzniká výsledná integrální složka I.

Derivační složka (D)

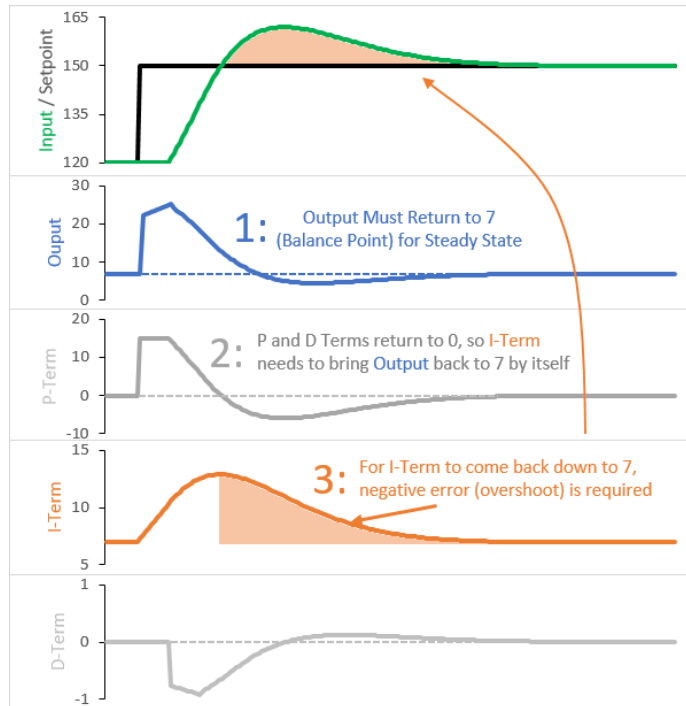
Derivační složka slouží k předpovědi změny sledované veličiny a její predikované hodnotě. Na základě této předpovědi se snaží zabránit změně sledované veličiny pomocí výstupu v opačné hodnotě než je predikovaná hodnota sledované veličiny.

S nulovou nebo příliš nízkou hodnotou této složky by nemuselo dojít k ustálení výkyvů sledované hodnoty, nebo by to alespoň trvalo déle.

Jako poslední krok dochází k pronásobení spočítané derivace k ní příslušnou vahou. Tím vznikne derivační složka D.

PID složky pohromadě

Na obrázku níže, který byl převzat z [13], se nachází grafické znázornění jednotlivých složek PID kontroléru. Jedná se o průběh složek a výstupu v čase, kdy se z ustáleného stavu zvýší cílová hodnota.



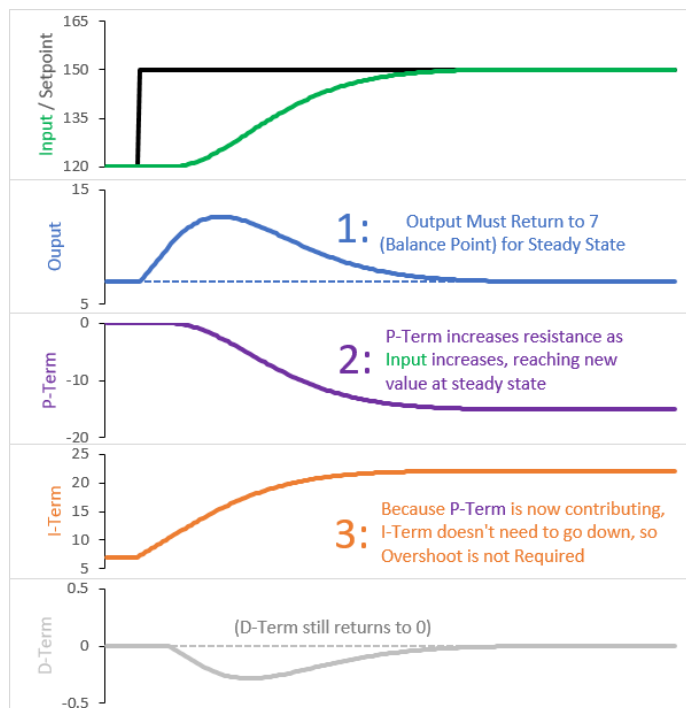
Obrázek 2.7: Znáznornění funkce složek PID kontroléru v čase, varianta PonE. Zdroj: [13].

Na začátku poskytuje výstup složka P, později se připojí složka I a nakonec při přibližování k cílové hodnotě složka D zapůsobí opačným směrem tak, že výsledný průběh překročí cílovou hodnotu. Každopádně, pro dosažení ustáleného stavu je potřeba překonat cílovou hodnotu tak, aby klesla hodnota integrační složky I, která má na ustálený stav významný vliv. V případě dobře vybraných vah jednotlivým složkám však tato situace nemusí být problémem.

Proportional on Measurement (PonM)

Alternativní přístup využívá proportional on Measurement (česky proporcionální výpočet na základě měření, dále anglicky, jelikož český název téměř není využíván).

Stále využívá složek PID, jen z nich jiným způsobem vypočítává hodnotu výstupu. Více k matematické části tohoto přístupu je dostupné na [13]. Níže je k nahlédnutí graf s možným průchodem hodnot, vysvětlení grafu se nachází v odstavcích pod obrázkem.



Obrázek 2.8: Znázornění funkce složek PID kontroléru v čase, varianta PonM. Zdroj: [13].

Tento přístup využívá pro dosažení ustáleného stavu namísto složky integrační (I) složku proporciální (P). To znamená, že aktuální zpětná vazba má větší vliv než integrační. Složka P může nabývat pouze hodnot menších nebo rovno 0, což je změna oproti předešlému přístupu, kde mohly být kladné i záporné hodnoty. Výsledkem tedy je to, že ze složky P se namísto urychlující stane tlumící složka, podobně jako derivační složka D.

V momentě dosažení cílové hodnoty je hodnota složky P nenulová, na rozdíl od předešlého přístupu, kde nuly v tomto případě dosáhne.

Výsledkem tohoto přístupu je to, že PID kontrolér s alternativním způsobem řízení PonM nepotřebuje přesáhnout cílovou hodnotu pro dosažení ustáleného stavu.

V programu je využita knihovna dostupná z [13], která umožňuje měnit poměr mezi klasickým způsobem regulace PonE a alternativním PonM.

PID autotune

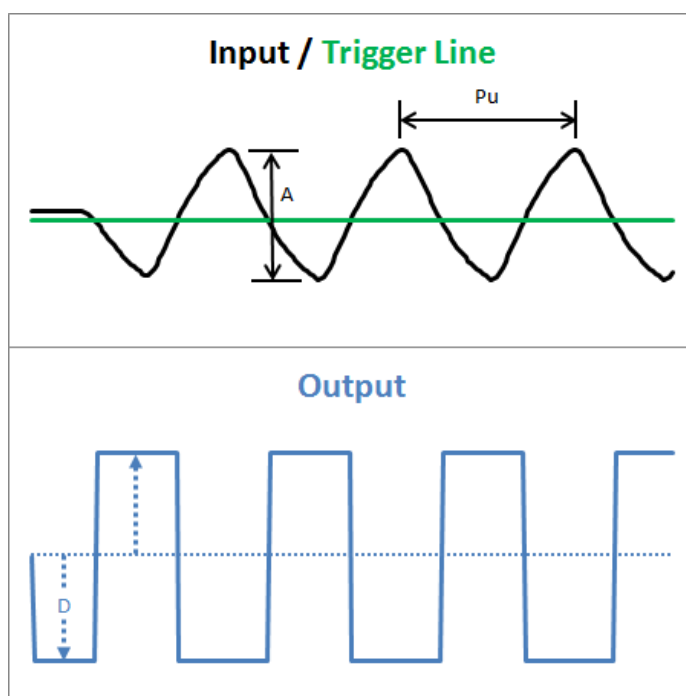
Dle [5] je v systémech regulace vytápění mnoho předem neznámých či v čase se měnících proměnných. Radiátory v jednotlivých místnostech mohou mít odlišný odpor toku vody, velikost radiátoru a být zakryté nábytkem. Místnosti bývají různě velké, v některých z nich mohou být různě poddimenzované či předimenzované radiátory. Vytápěcí systém se postupně může zanést či částečně ucpat vodním kamenem a jinými nečistotami. Senzory teploty a motory pro ovládání ventilů mohou degradovat.

Z tohoto důvodu nemá pro praktické či komerční použití příliš velký smysl ručně ladit řídicí konstanty PID kontroléru. Logičtější je přenechat ladění na automatickém systému, který funguje na míru pro konkrétní umístění využití hlavice. Na základě historie naměřených dat spočítá vhodné hodnoty řídicích konstant pro PID kontrolér. Tomuto systému se říká PID autotune.

Jedna z těchto metod výpočtu se jmenuje metoda relé. V porovnání s ostatními metodami si dokáže lépe poradit s nahodilými výkyvy a šumem zpětné vazby. Tato metoda je taktéž vhodná pro využití systémy s dlouhou periodou výpočtu nových hodnot PID kontroléru (anglicky známo jako time constant). Další z výhod je nízká náročnost matematických výpočtů.

Základní princip funkce metody relé je znázorněn na obrázku níže. Snaží se zjistit periodu opakování výkyvů sledované veličiny a rozdíl maxima a minima sledované veličiny. Tato metoda tedy nebere v potaz přesné hodnoty, ale pouze časový průběh binární informace o pozici nad či pod cílovou hodnotou v kombinaci s rozdílem maximální a minimální dosažené hodnoty.

Během probíhání této metody je potřeba přenechat řízení PID kontroléru na autotune. Ten v případě, že se nachází pod cílovou hodnotou, zapne výstup na maximální možnou hodnotu. V momentě, kdy cílovou hodnotu překročí, přepne výstup na nulu. Tímto způsobem zjistí maximální a minimální hodnoty (a tedy rozptyl) a periodu opakování těchto stavů.



Obrázek 2.9: Znázornění PID autotune metody relé. Cílová hodnota představuje zelenou čáru (na obrázku Trigger Line), zpětná vazba černou křivku (na obrázku Input) a výstup pro výpočet je o graf níže modrou křivkou (na obrázku Output). Zdroj: [5].

Na základě těchto informací vypočítá nové hodnoty s využitím vzorců, o kterých je psáno například v [5]. Cílem by měl být PID kontrolér s řídicími konstantami upravenými na míru konkrétnímu využití hlavice. To znamená vytápění s nižšími výkyvy teplot či rychlejším vytápěním. Uživatel však musí vydržet v diskomfortu po dobu, kdy probíhá PID autotune. Uživatel tedy nemůže ovlivňovat průběh teploty například tím, že otevře okno, dveře či jiný významný zdroj tepla.

Je otázkou, zda-li je takové řešení přijatelné pro použití výrobku vycházejícího z této práce běžnými zákazníky, kteří si nemusí přečíst návod k použití od hlavice. Může dojít k situaci, že v době, kdy probíhá PID autotune hlavice, bude hlavice způsobovat velké kolísání teplot. To může vést k tomu, že na základě první hodiny používání si uživatel udělá o výrobku špatný dojem a přestane jej používat.

Dle [15] alternativou k autotune je self-tune. Ten se liší tím, že ladí řídicí konstanty PID kontroléru za jeho běhu. Tím odpadá nutnost odstavit PID kontrolér a předat řízení autotune. Takové řešení by mohlo být zajímavým rozšířením implementace této práce.

Kapitola 3

Výběr technologií chytrého vytápění

3.1 Komunikace s centrální jednotkou

3.1.1 Síťové protokoly

WiFi

Na základě [58] WiFi je rodina bezdrátových protokolů založených na standardu IEEE 802.11. Bývá využívána pro domácí sítě, často s přístupem k Internetu. Sítě založené na této technologii využívající 2.4 GHz mohou mít dosah mezi 20 a 150 metry v závislosti na množství a složení zdí a překážek mezi připojenými zařízeními.

Standardy WiFi byly poprvé vydány před 20 lety a umožňují širokou zpětnou kompatibilitu zařízení využívajících starší verzi standardu se zařízeními využívajícími novější verzi.

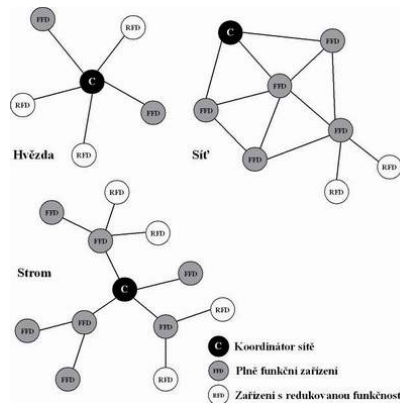
Komunikace probíhá přes centrální jednotku, známou jako přístupový bod (známo pod anglickou zkratkou AP, což znamená Access Point).

WiFi při umožňuje zabezpečit přenesená data proti odposlechu. K tomu je potřeba správně nastavit zabezpečení sítě pomocí hesla.

Tento standard je primárně určen pro rychlejší přenos dat v řádu desítek či stovek mb/s za cenu vyšší energetické náročnosti, v případě využití starších verzí protokolu umožňuje snížit spotřebu energie výměnou za výrazné snížení rychlosti přenosu. Pro zařízení chytré domácnosti může být využití starší verze protokolu přijatelnějším kompromisem.

Zigbee

Jak je psáno v [59], Zigbee je specifikace pro komunikaci mezi zařízeními v lokálním měřítku, založená na standardu IEEE 802.15.4. Jeho zásadní výhodou oproti WiFi je přenos dat ve stylu skoků, kde klientská zařízení mohou sloužit jako prostředník pro komunikaci mezi vzdálenými zařízeními bez přímého dosahu signálu. Další výhodou je za určitých podmínek vyšší dosah a nižší spotřeba energie.



Obrázek 3.1: Varianty topologie Zigbee, znázorňují, že koncová zařízení nemusí mít přímé spojení s koordinátorem (černě), stačí využít jiné zařízení s funkcí preposílání (šedě) jako prostředníka spojení. Zdroj: [50].

Každá síť se zařízeními běžícími na Zigbee potřebuje centrální jednotku, jejíž funkcí je koordinátor sítě. Rychlost přenosu přes Zigbee se pohybuje od 20kb/s do 250 kb/s. To znamená, že je vhodné pro menší datové toky, které se mohou a nemusí často opakovat.

Zigbee je mimo jiné vhodný pro IoT díky svému návrhu, který zajišťuje nízkou nákupní cenu, nízkou spotřebu a nízkou latenci.

Centrální jednotka se dá získat okolo 1000 Kč. Informace k Zigbee hubům, které se dají připojit do systému Home Assistant (tomu je věnována pozdější část práce) lze najít na [22]. Vývojové klientské moduly se pohybují cenou okolo 200-300 Kč a většinou slouží pouze ke komunikaci. Oproti vývojovým modulům založeným na WiFi je viditelný rozdíl v ceně a rozšířenosti a s tím spojené komunitní podpoře ve prospěch WiFi.

Pro účely bakalářské práce by vývojové moduly často potřebovaly další desku z důvodu omezeného množství GPIO pinů. Mají též složitější možnosti naprogramování oproti WiFi modulům, které jsou často kompatibilní s Arduino IDE a podobnými rozšířenými nástroji.

Aktuálně tedy Zigbee tíží nedostatky vycházející z výrazně nižší rozšířenosti oproti WiFi. To se může časem změnit.

3.1.2 Síťová komunikace

V následujících kapitolách jsou shrnuty podstatné informace o některých používaných technologiích, které se používají pro komunikaci zařízení chytré domácnosti. Poslední kapitola se věnuje zdůvodnění výběru technologie pro tuto práci.

MQTT

Dle [36] je MQTT lehký, rychlý a flexibilní komunikační protokol určený primárně do Internetu věcí (běžně známý pod anglickou zkratkou IoT) a komunikaci mezi zařízeními (Machine to Machine communication, nebo také M2M).

V MQTT komunikaci figurují tyto tři role: broker, publisher a subscriber. Broker je centrální jednotka, přes kterou prochází zprávy od publisherů k subscriberům. Publisher představuje roli koncového zařízení, které odesílá zprávu. Subscriber je role koncového zařízení, které se přihlásí (anglicky subscribe) k odebírání témat. Zprávy obsahují informaci o tématu, ke kterému se vztahují a hodnotu. Tato hodnota může být velmi krátká i relativně

dlouhá, až 256 MB. Jedno zařízení může publikovat i přijímat informace k různým tématům, která se nemusí překrývat.

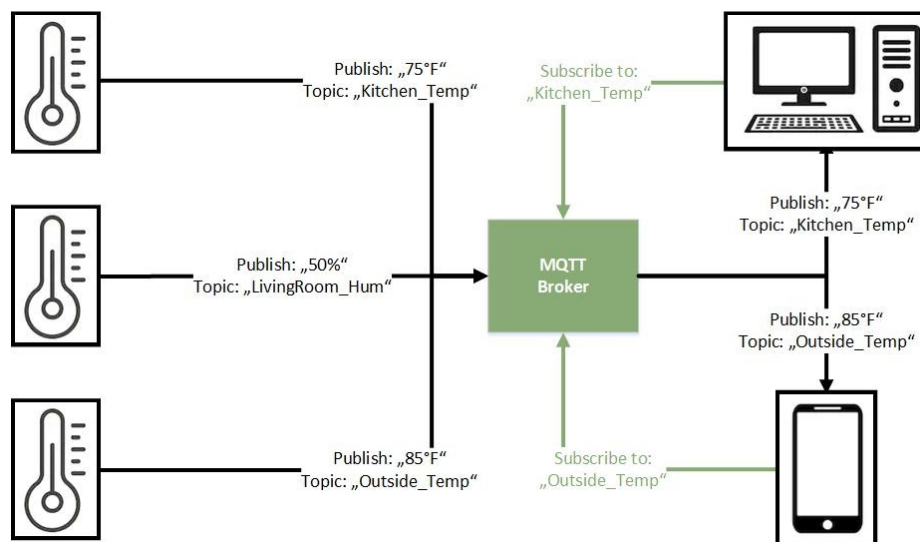
Pro zahájení MQTT komunikace je třeba znát pouze IP adresu brokera a název tématu, ke kterému chce zařízení odeslat informaci. Znamená to tedy, že odesílatel nepotřebuje znát IP adresy cílových zařízení, která se zajímají o informace k danému tématu. Přijímající strana potřebuje znát stejné informace, to znamená pouze IP adresu brokera a název témat o které má zájem.

Na základě [11] mohou být sezení mezi brokerem a zařízeními perzistentní, díky tomu se klientské zařízení nemusí při každém novém připojení k brokerovi znovu přihlašovat k odběru témat. Namísto toho stačí klientovi si zapamatovat jeho unikátní identifikátor, pomocí kterého se rozliší od ostatních klientů. Po výpadku spojení tak stačí pouze sdělit brokerovi identifikátor, na základě kterého klient přijme promeškané zprávy k tématům, ke kterým byl přihlášen.

Kromě toho zaručuje doručení zpráv od brokera klientům k přihlášeným tématům tak, jak je přijal. To znamená, že nenastane situace, kde by při výpadku spojení ať už na straně odesílatele či příjemce dorazily zprávy ve špatném pořadí. Pokud je spojení s brokerem perzistentní, klient dostane zprávy i z doby, kdy nebyl připojen. Díky tomu není potřeba synchronizovat čas u koncových zařízení pro zajištění seřazení přijatých zpráv.

Dle [14] je samotné spojení mezi brokerem a zařízeními perzistentní. To znamená, že pokud nenastane výpadek spojení, zařízení jsou neustále připojená. Pro udržení připojení si posílají keep-alive ping a kterákoli strana může v libovolný moment odeslat zprávu druhé straně. Tento styl spojení je vhodný pro časté odesílání dat zejména malého objemu.

Zdroj [3] říká, že MQTT také umožňuje nastavit dle priority zpráv záruku doručení. Nastavuje se pomocí hodnoty QoS. Pokud je QoS nastaveno na 0, zpráva se přihlášeným zařízením odešle jednou a broker nekontroluje, jestli zpráva opravdu dorazila. U QoS 1 a 2 je doručení zprávy zaručeno, pouze s rozdílem, že u QoS 2 zpráva dorazí právě jednou a ne vícekrát.



Obrázek 3.2: Schéma s členy MQTT komunikace. Publisherem je zde teploměr, který publikuje zprávy brokerovi (zelený obdélník). Broker následně rozešle zprávy těm zařízením, která jsou ke zprávám konkrétního tématu přihlášená (vpravo). Zdroj: [12].

REST

Dle [14] a [26] je REST styl architektury často využívaný pro příjem a odesílání dat interaktivními webovými aplikacemi. RESTful znamená, že daná webová aplikace či služba využívá tohoto stylu architektury. Pro účely této práce se dá tohoto stylu využít pro komunikaci mezi centrální jednotkou a jinými připojenými zařízeními chytré domácnosti.

Umožňuje odesílat informace v podobě HTML, XML, JSON či v jiném téměř libovolném formátu. Je bezstavový, díky čemuž jeho implementace nemusí být příliš složitá a umožňuje vyšší univerzálnost. Znamená to, že server i klient nepotřebují pro pochopení příchozí zprávy znát předešlé zprávy. Tato vlastnost může být v některých případech nevhodná.

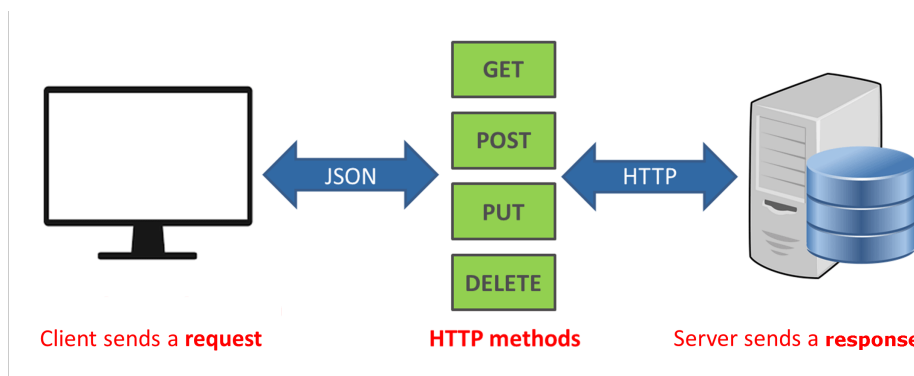
Implementuje tyto stavební metody známé pod označením CRUD: Create (česky vytvořit data na druhé straně), Retrieve (získat určitá data), Update (aktualizovat neboli změnit uložená data), Delete (smazat data). Jedná se o velice univerzální a i díky své jednoduchosti široce používaný systém.

Využívá síťové architektury klient-server. To znamená, že klient žádá server o nějaké informace či služby a server v reakci na tuto žádost odešle odpověď. Pokud by server chtěl odeslat nějakou informaci směrem ke klientovi, musí vyčkat na příchozí požadavek od klienta a až v tomto momentě může klientovi tuto informaci předat.

Pokaždé, co chce klient komunikovat se serverem, musí znovu zahájit spojení a až poté odeslat samotná data, počkat na odpověď a poté spojení ukončit. V případě častého odesílání menšího objemu dat tato režie může tvořit i větší podíl přenesených dat než samotná informace, která se přenáší.

Tato skutečnost je jednou z příčin vyšší náročnosti na procesor i síť. U běžných webových aplikací bývá tato skutečnost zanedbatelná. Avšak v případě zařízení chytré domácnosti, která jsou často napájena bateriemi s omezenou kapacitou, už zanedbatelná nebývají. Další z nevýhod je relativně vysoké množství dat v hlavičkách požadavků a odpovědí.

Ačkoliv REST je z hlediska rychlosti a jednoduchosti použití lepší než mnohé jiné architektury, jeho univerzálnost se projevuje jako nevýhoda pro systém chytré domácnosti, kde je kladen vyšší důraz na nízkou náročnost na baterii a procesor.



Obrázek 3.3: Schéma s členy REST komunikace. Komunikaci vždy započne klient (obrazovka), který odešle požadavek s informacemi (třeba o naměřené teplotě). Server informaci zpracuje a odešle odpověď. Pokud server potřebuje odeslat zprávu klientovi, musí vyčkat, až mu přijde nějaký dotaz od klienta. Zdroj: [6].

3.2 ESP8266

3.2.1 Základní informace

Na základě [56] ESP8266 je levný mikročip s integrovanou WiFi konektivitou a relativně výkonným procesorem běžícím na frekvenci 80 Mhz.

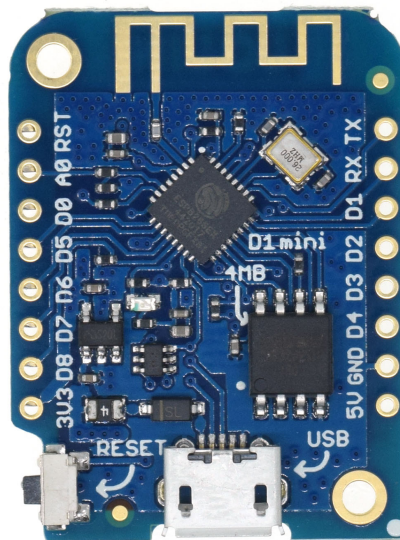
Napájí se pomocí 3,3V. Má poměrně vysoké množství 16 GPIO pinů (GPIO piny jsou digitální vstupně-výstupní kontakty pro komunikaci a ovládání připojených zařízení, dále bude v práci používána tato zkratka, protože je běžně využívána). S těmito piny zvládne komunikovat pomocí SPI například s nevolatilní pamětí flash, dále pomocí I2C, I2S a UART. Dále umožňuje na jednom pinu čtení analogového vstupu v rozsahu 0 až 3,3 V.

Původně byl tento mikročip určen pro čínský trh, díky své nízké ceně se začal objevovat po celém světě v kutilských projektech, kde byla přidána WiFi konektivita k existujícímu zařízení či pro prototypování celých projektů. Později byla pro tento čip vytvořena podpora pro programovací platformy a jazyky jako například Arduino IDE, ESPHome a Micropython.

3.2.2 NodeMCU D1 Mini

Dle [49] je D1 Mini malou levnou deskou založenou na ESP8266. Tato deska umožňuje WiFi konektivitu, využití 11 GPIO pinů a k tomu přidává napájení pomocí micro USB či 5V díky integrovanému napěťovému převodníku. Ačkoliv lze pro napájení použít 5V, veškeré GPIO piny komunikují pomocí 3,3V.

Pro komunikaci a nahrávání programů mezi počítačem a ESP8266 se dá využít integrovaného převodníku CH340, který zajišťuje konverzi USB komunikace na sériovou komunikaci mikročipu. Na desce se také nachází integrovaná 4MB flash paměť připojená pomocí SPI k mikročipu. Ta se dá využít například pro ukládání uživatelem nastavené konfigurace.



Obrázek 3.4: Deska NodeMCU D1 Mini. Zdroj: [48]

3.2.3 Software pro ovládání ESP8266

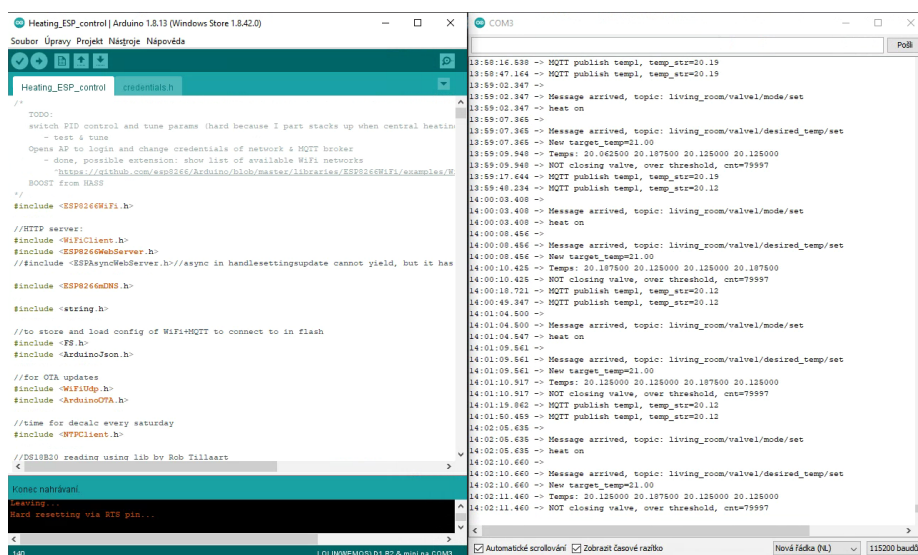
V následujících podkapitolách se nachází výčet a srovnání některých softwarových řešení, která by se mohla využít pro nahrání programu ovládajícího ESP8266.

Arduino IDE

Dle [2] je Arduino IDE integrované prostředí pro vývoj, kompilování, nahrávání a ladění programů určených primárně pro jednočipové desky této firmy jako je Arduino Uno. Existují rozšíření pro mnoho dalších zařízení, mezi nimi i ESP8266 a na tomto čipu postavené vývojové desky.

Programy vytvořené v Arduino IDE jsou napsané v Embedded C. K programům lze jednoduše přidat knihovny vytvořené jinými uživateli díky existenci správce knihoven. Správce při instalaci knihoven upozorní na případné závislosti na jiných knihovnách a pomůže je vyřešit. Dále zajišťuje správu instalovaných knihoven tak, aby byly aktuální.

Pro Arduino IDE existuje knihovna, která po nahrání na zařízení umožňuje vzdáleně pomocí OTA aktualizací nahrát novou verzi ovládacího softwaru. Způsob ovládání na dálku je též známý pod zkratkou OTA, z anglického over the air, což v českém jazyce znamená přes vzduch.



Obrázek 3.5: Ukázka vývojového prostředí Arduino IDE, vlevo prostředí pro vývoj, vpravo výstup z konzole připojeného zařízení.

ESPHome

Dle zdroje [30] ESPHome představuje systém ovládání zařízení založených na čípech ESP8266 či ESP32. Tento systém je primárně určen pro senzory různého typu či jednoduché přepínače. Ovládání probíhá pomocí systému domácí automatizace.

Nastavení jednotlivých zařízení probíhá pomocí konfiguračních souborů YAML. Pro složitější senzory existuje možnost rozšíření YAML o část kódu napsanou v C++.

Správa zařízení a jejich konfigurace je velmi jednoduchá a přehledná. Systém umožňuje jednoduše na dálku spravovat a aktualizovat připojená zařízení. Tento způsob je také znám jako OTA.

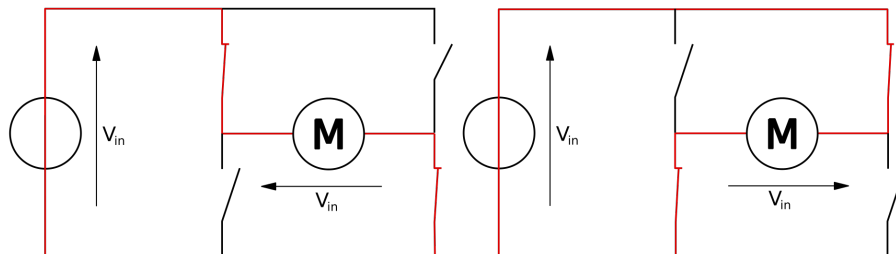
3.3 H můstek

H můstek je dle [55] elektronický obvod běžně dostupný jako jedna součástka. Zajišťuje ovládání zátěže a pomocí přepínání polarity výstupu umožňuje otočit směr proudu v zátěži. Další funkcí je to, že při vypnutém stavu umí držet zátěž ve zkratu či otevřenou.

Zátěž velmi často představuje elektrický motor na stejnosměrný proud. U toho lze pomocí zkratu či otevření kontrolovat brzdění pohyblivého se motoru či vytvoření odporu vůči pohybu, nebo nebrzdit a nechat motor bez omezení.

Běžně se tento obvod skládá ze čtyř tranzistorů, kde jejich zapnutí či vypnutí v určité konfiguraci vede k požadovaným funkcím. Pro účely této práce bude stačit otáčet motorem v obou směrech, řídit brzdění či nebrzdění nebude potřeba.

V obrázku pod tímto odstavcem jsou k nahlédnutí schémata znázorňující ovládání motoru, druhé schéma v opačném směru. Ve schématech se na levé straně nachází zdroj, na pravé straně H můstek, kde značky vypínačů mají stejnou funkci jako tranzistory. Kolečko s M uprostřed značí zátěž v podobě motoru.



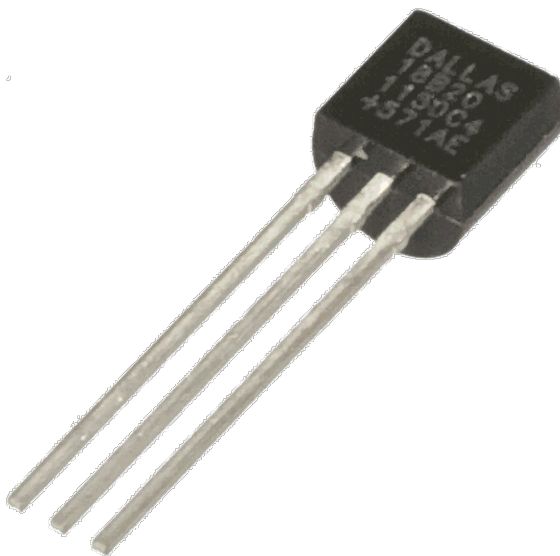
Obrázek 3.6: Schéma znázorňující základní funkce H můstku. Zdroj: [55].

3.4 Teplotní senzor DS18B20

Dle [28] DS18B20 je senzor teploty, který pro svou funkci potřebuje pouze 2 až 3 vodiče v závislosti na režimu zapojení. Umožňuje nastavit rozlišení měření od 9 do 12 bitů s přesností $\pm 0,5$ °C.

Komunikace probíhá pomocí 1-Wire rozhraní, což znamená, že pro obousměrnou komunikaci stačí jeden datový vodič. Každý teplotní senzor má svoji unikátní 64bitovou adresu, díky čemuž pomocí jediného datového vodiče může komunikovat více senzorů.

Senzor se nabízí ve více podobách, voděodolné s kabelem, pro připájení na desky plošných spojů a v pouzdru TO-92 pro všestranné využití, kde není třeba voděodolnost.



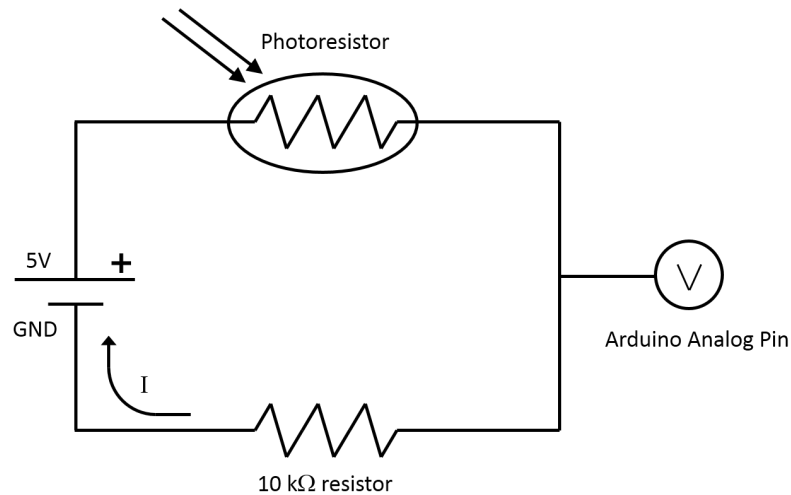
Obrázek 3.7: Teplotní senzor DS18B20 v pouzdře TO92. Zdroj: [32]

3.5 Světelný senzor

Pro detekci intenzity světla existuje mnoho druhů senzorů. Mohou být založeny na fotorezistoru, fotodiodách a dalších. Mohou k nim být přidány další součástky pro zlepšení přesnosti měření či jednoduchost použití.

Fotorezistor je pasivní elektronický prvek. Funguje podobně jako rezistor, ale s tím rozdílem, že jeho odpor se mění v závislosti na intenzitě dopadajícího světla.

Obvyklé zapojení pro Arduino a jiné malé desky s ADC převodníkem (analog to digital converter, převádí vstupní napětí v daném rozsahu na číslo v daném rozsahu, se kterým může pracovat program) využívá napěťové děličky. Tato dělička umožní čtení analogového napětí na vstupním pinu v závislosti na měnícím se poměru odporu světelného senzoru vůči $10\text{k}\Omega$ rezistoru. Toto zapojení je vyobrazeno na schématu níže.



Obrázek 3.8: Schéma zapojení fotorezistoru a $10\text{k}\Omega$ rezistoru pro čtení napětí. Zdroj: [27].

3.6 DHCP server a statický záznam

Na základě [53] DHCP je protokol používaný v lokálních sítích (LAN) pro přidělení unikátních IPv4 adres sloužících pro komunikaci po síti. Je rozšířen v téměř každé domácnosti a každém zařízení s připojením k Internetu, ale i ve velkých lokálních sítích firem a univerzit.

Klientská zařízení komunikují s DHCP serverem pomocí zpráv discovery, offer, request a acknowledge. Pomocí těchto zpráv si klientské zařízení zajistí od DHCP serveru unikátní adresu, výchozí bránu a další informace, které jsou potřeba pro komunikaci s ostatními zařízeními na síti. Díky tomu není potřeba ručně konfigurovat klientské zařízení v každý moment, kdy se připojí do jiné sítě.

3.7 Home Assistant

Dle [57] a [21] představuje Home Assistant program domácí automatizace pro centrální ovládání jiných systémů a prvků domácí automatizace. Tento program je založen na otevřeném kódu (anglicky open source code) a umožňuje širokou škálu rozšíření. Zaměřuje se na lokální ovládání a ochranu soukromí uživatelů.

Umožňuje přístup přes webové rozhraní, aplikaci pro Android či iOS nebo dokonce pomocí hlasových povelů přes virtuální asistenty, jako je třeba Google Assistant a jiné. Z aplikace umožňuje sdílet informaci o poloze vevnitř či vně určité oblasti do centrální jednotky, která na základě těchto dat může například začít vytápět, když se jeden z uživatelů blíží domů. Možností je nespočet.

Na základě [29] Home Assistant umožňuje za měsíční poplatek 5 USD provozovat server v cloudu. Tato služba se jmenuje Home Assistant Cloud. Provozuje ji společnost Nabu Casa, Inc., která z příjmů financuje programátory, kteří se na vývoji programu Home Assistant podílejí. Díky tomu má tento program dobrou perspektivu dalších aktualizací jak zabezpečení, tak i funkčních rozšíření.

Uživatelé tato služba umožňuje odlehčit od rizik a potřebného nastavování, které je spojeno s nastavením zabezpečeného přístupu k programu Home Assistant z vnější sítě. Dále výrazně zjednoduší spojení s externí službou Google Assistant či Amazon Alexa.

3.7.1 Operační systém HASS.io

Home Assistant je od roku 2017 nabízen také jako kompletní operační systém HASS.io. Tento systém je dostupný pro jednodeskové počítače jako je například Raspberry Pi, Intel NUC či ODROID. Pro nepříliš zdatné uživatele existuje hotový balíček počítače ODROID N2+ se základním příslušenstvím a předinstalovaným operačním systémem HASS.io.

Existuje i snímek virtuálního stroje s Home Assistantem, který mohou uživatelé spouštět na zařízení běžících na klasických operačních systémech jako je Microsoft Windows, Linux a jiné. Stačí jim k tomu hostitelský program, například Oracle VM VirtualBox, VMware a podobné.

Hlavní výhodou tohoto operačního systému je jednoduchost použití pro začátečníky i široké možnosti využití pro zkušené uživatele. Pokud by si uživatel chtěl nainstalovat rozšíření, stačí po přihlášení jako administrátor pár kliknutí ve webovém rozhraní. Následná konfigurace rozšíření je díky jednoduše dostupným konfiguračním souborům a dokumentaci rozšíření velmi jednoduchá.

3.7.2 Telefon s Android jako host Home Assistant

Nemalý počet lidí má doma starý telefon, případně nový s rozbitým sklem displeje, který mají schovaný v šuplíku.

Dle [7] je zprovoznění Home Assistanta na zařízení se systémem Android možné, avšak není oficiálně podporováno vývojáři Home Assistanta. S tím je spojena nižší spolehlivost a vyšší náročnost na zprovoznění.

Nejznámější řešení využívá program Termux, což je nástroj, který poskytuje příkazový řádek společně s nejrůznějšími knihovnami. Potřebná verze systému Android je 5.0 a vyšší. Tento nástroj není třeba instalovat s právy root. Umožňuje nastavit vzdálené připojení ke konzoli (textové rozhraní známé jako příkazový řádek) pomocí SSH, takže jej lze ovládat z pohodlí klávesnice počítače.

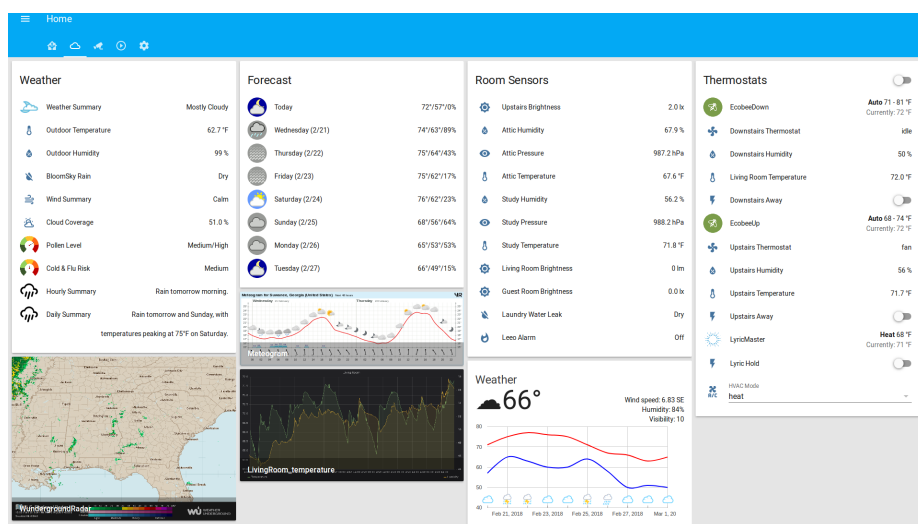
Pro instalaci rozšíření Home Assistanta nelze využít webového rozhraní, ale místo toho je třeba zadat příkazy do konzole Termuxu. Některá rozšíření nemusí být dostupná či kompatibilní pro konkrétní hostující zařízení.

3.7.3 Ovládací panel Lovelace

K ovládání a zobrazování informací o zařízeních slouží ovládací panel Lovelace. Využívá systému karet, díky kterým se dá vybrat umístění ovládacích či informativních prvků jednotlivých zařízení. Karty se skládají do svislých pásů vedle sebe dle šířky obrazovky zařízení.

Lovelace umožňuje ručně rozmístit karty do svislých a vertikálních pásů karet. Výhodou je jasné rozmístění prvků na různě velkých obrazovkách počítače, problémem však je, že prvky se automaticky nenaskládají pod sebe ale zůstanou v nastavených pásech.

Nastavení lze provést kromě uživatelsky přívětivého webového klikacího rozhraní i pomocí kódu v jazyce YAML. Karty mohou obsahovat jinou webovou stránku (element `iframe`).



Obrázek 3.9: Příklad ovládacího panelu Lovelace. Zdroj: [34]

3.8 Node-RED

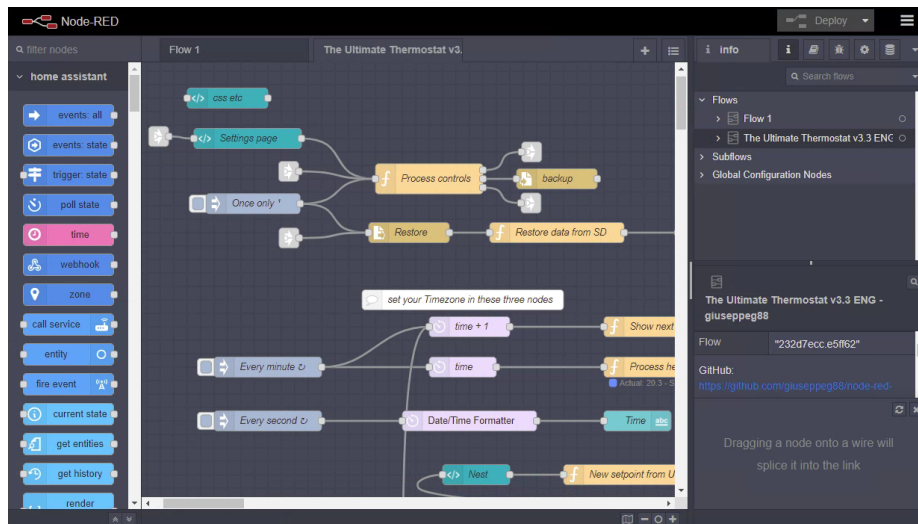
Na základě [31] je Node-RED vizuální programovací nástroj pro spojení nejrůznějších API (česky aplikační programové rozhraní, slouží pro zajištění standardu komunikace programu s okolím, dále pouze jako API, protože je běžně používáno). Dále umí propojit lokální či online služby a zařízení.

Jedná se o nástroj vhodný jak pro začínající uživatele, tak i zkušené. Díky grafickému pojetí funkčních částí jsou programy jednodušejší pochopitelné než například v JavaScriptu, v kterém je Node-RED napsán.

Základní jednotkou jsou uzly, které tvoří jednotlivé toky.

Uzlem může být vstupní, výstupní, logická či funkční činnost. Jednotlivé uzly mohou v závislosti na konkrétním druhu uzlu obsahovat vstup či výstup pro hodnoty. Tyto vstupy a výstupy se mohou připojit k jiným uzlům. Všechny vstupy a výstupy nemusí být využity.

Toky tvoří jednotlivé části – funkční celky programu, které se mohou samostatně spouštět a vypínat. Samozřejmostí je, že tyto toky si mohou předávat informace.



Obrázek 3.10: Ukázka kódu programu vytvořeného v Node-RED. Zdroj: [20].

3.9 InfluxDB a Grafana

3.9.1 InfluxDB

Dle [23] je InfluxDB databázový systém pro ukládání a základní zobrazování dat ve stylu časových sérií. Nejedná se o častěji známý relační databázový systém, který je vhodný spíše pro více provázané skupiny informací, kde je menší množství záznamů.

Zdrojový kód tohoto systému v jazyce Go je veřejně publikován pod licencí MIT.

Tento systém je tedy velice vhodný pro zaznamenávání dlouhé (ve smyslu množství položek) historie naměřených dat ze senzorů a zařízení.

InfluxDB nachází využití například v monitorování systémů či aplikací. Další využití je pro zaznamenávání naměřených dat z Internetu věcí (IoT).

3.9.2 Grafana

Na základě [24] k zobrazování dat uložených například v databázovém systému InfluxDB může posloužit Grafana.

Vizualizace probíhá pomocí grafů, schémat, tabulek a upozornění. Grafy umožňují zobrazit minima a maxima, různými způsoby vyplnit mezery mezi datovými položkami.

Profesionální využití

Pro uživatele z firemní sféry mohou zakoupit licenci vylepšené verze Grafany o další funkcionality, případně si platit předplatné za produkt Grafana Cloud, který nabízí kombinaci vylepšené verze společně s poskytnutím výpočetního výkonu vzdálených serverů pro zajištění chodu Grafany.



Obrázek 3.11: Ukázka zobrazení dat v Grafaně. Zdroj: [24].

Kapitola 4

Návrh úpravy hlavice

Tato kapitola se zabývá návrhem úpravy základní digitální termostatické hlavice pro připojení do systému chytré domácnosti. Pro tyto účely byla zvolena hlavice Eqiva EQ-3 N pro svou nízkou cenu a již zdokumentovaný návod na rozložení, který je dostupný z [45].



Obrázek 4.1: Fotografie digitální hlavice Eqiva EQ3-N. Zdroj: [17].

4.1 Předpokládané vlastnosti upravené hlavice

Tato podkapitola zmiňuje vlastnosti, které by úprava měla zajistit či pokud možno zachovat.

První z nich je napájení pomocí baterií, pokud se společně s řídicí jednotkou vejdou dovnitř těla hlavice. Pokud se baterie nevejdou, bylo by vhodné umožnit napájení standardizovaným způsobem a konektorem, který nepředstavuje riziko usmrcení vysokým napětím. Pro tyto účely by mohl sloužit microUSB či USB-C konektor vhodně umístěný v těle hla-

vice. Pokud bude napájení řešeno bez využití baterií a bude využívána některá z komponent na původní řídicí desce hlavice, bude potřeba zajistit převodník napětí z 5V, které nabízí USB, na 3,3V, které potřebuje původní deska.

Hlavice by měla být schopna komunikovat s centrální jednotkou, která může být provozována na lokální síti nebo i vzdáleně v cloudu. Způsob komunikace by měl být v určitém standardu tak, aby byla zajištěna možnost připojení do systému chytré domácnosti, kde se mohou objevovat zařízení jiného typu (světlo, chytrá zásuvka, ...). Dosah komunikace by měl být dostatečně velký, aby dosáhl všude, kde se budou objevovat ventily.

Bylo by vhodné se vyhnout nutnosti pořizovat další zařízení sloužící jako hub pro speciální standard komunikace ve prospěch sítě WiFi, která je dnes běžně rozšířená v každé domácnosti.

Bluetooth by pro účely centrálního systému chytré domácnosti nestačil. Mezi hlavní důvody patří omezené možnosti připojení k centrální jednotce v síti (pravděpodobně s nutností zakoupení dalšího zařízení) a slabý dosah signálu, zejména přes zeď.

Ve výčtu vlastností by nemělo chybět zachování a případné vylepšení funkcí pro detekci otevřeného okna a režim boost. Režim boost by bylo vhodné umožnit ovládat na dálku. U ventilů by se hodila detekce přítomnosti osob v místnosti. Detekce může být provedena pomocí nějakého senzoru (například světelného, jak je využit v projektu Radbot, více vizte [46]) nebo napojením na centrální jednotku, která tuto informaci zjistí například díky detekci polohy telefonu uživatele.

4.2 Ovládací jednotka

Pro komunikaci zařízení chytré domácnosti existuje velké množství technologií, vybrané z nich byly popsány v předešlých kapitolách. Následující odstavce procházejí návrhem řešení vytápění.

Jak již bylo zmíněno, jako základ pro úpravy byla vybrána digitální termostatická hlavice Eqiva EQ-3 N především díky své nízké ceně.

Ovládací jednotka byla vybrána NodeMCU D1 mini založená na ESP8266. Hlavními důvody jsou tyto: dostatek GPIO pinů, WiFi připojení součástí desky, napájení pomocí 5V microUSB a pro účely této práce dostatečně malé rozměry desky. Obsahuje integrovaný převodník z 5V na 3,3V, což potřebuje jak samotné ESP8266, tak i původní ovládací deska hlavice, kterou je v plánu využít. To vše za cenu do 50 Kč za kus.

Jako senzor teploty byl navržen DS18B20 v základním provedení TO-92 pro svou jednoduchost použití a dostačující přesnost měření. Balení 5 ks z čínského e-shopu stojí do 15 Kč za kus.

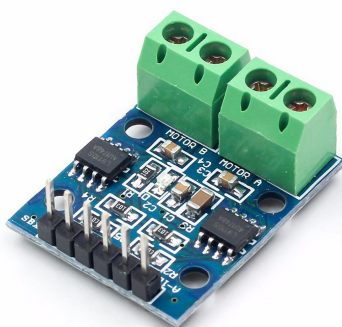
Pro napájení bylo vybráno USB, nakonec však byla zvolena varianta přímého připojení kabelu k modulu NodeMCU D1 mini. Na druhé straně kabelu se nacházel USB-A samčí konektor, který se dal přímo zasunout například do staré nabíječky na telefon.

Mezi variantami byl i USB-C port, vzhledem k vysoké ceně varianty s malými rozměry, jednoduše připajitelnými kontakty a dalším nevýhodám sdílenými i s microUSB portem, byly tyto varianty vyloučeny již při návrhu a objednávání komponentů. Společnými nedostatky byla nutnost vytvářet přesně velký otvor zespod hlavice pro USB port a přidaná složitost upevnění portu uvnitř hlavice tak, aby se při nešetrném zacházení neuvolnil. Dalším společným nedostatkem byla dle názoru autora práce horší estetická hodnota tohoto řešení.

4.3 Ovládání motoru hlavice

Pro otáčení motoru hlavice bylo třeba dodat dostatečný stejnosměrný proud a pomocí změny polarity měnit směr otáčení.

Tuto funkci bylo v plánu přenechat H můstku. Návrh počítal s variantou využití H můstku integrovaného na původní ovládací desce od hlavice, na který by se připojely kontrolní piny z NodeMCU. Pravděpodobné schéma ovládací desky hlavice je dostupné z [9]. Náhradní variantou bylo využití H můstku na samostatné desce, jako je L9110S [43] či L298N.



Obrázek 4.2: Fotografie desky s H můstkem L9110S. Zdroj: [43].

4.4 Logika vytápění

Pro ovládání vytápění byla zvolena logika s predikcí budoucí teploty pro svou relativně jednoduchou implementaci a její jednoduše pochopitelný způsob funkce. Tento způsob regulace byl inspirován z [44] a upraven a byl popsán v kapitole 2.3.3.

Kapitola 5

Návrh systému vytápění

Tato kapitola se zabývá návrhem systému chytré domácnosti s ohledem na možnosti ovládní vytápění, otevřenost k různým zařízením a pravděpodobnost budoucích aktualizací.

5.1 Centrální jednotka

Pro ovládní ventilů a následné shromažďování, zaznamenávání a vizualizaci dat bylo navrženo využít systému chytré domácnosti Home Assistant 3.7 běžící na Raspberry Pi Zero W či Raspberry Pi B+.

5.1.1 Systém centrální jednotky

Varianta vlastního řešení nebyla vybrána kvůli většímu rozsahu práce potřebné k zprovoznění systému postaveného například v Pythonu či Javascriptu s využitím frameworků. Home Assistant je poměrně rozšířený a stojí za ním solidní komunitní podpora i tým vývojářů. Pro představu, ovládací aplikace pro Home Assistant na Google Play Store má přes 100 000 stažení s hodnocením aplikace 4,5 hvězdy [4].

U vlastního řešení založeného na využití knihoven jako ReactJS a podobnými by práce navíc příliš nepomohla k přidání uživatelské hodnoty řešení, když Home Assistant umožňuje přidat různá oficiální rozšíření a upravit si již hotová komunitně vyrobená řešení pro své účely, systém ovládní vytápění nevyjímaje.

5.1.2 Hostující zařízení

Pro provoz tohoto systému byl vybrán telefon Oneplus 3T s procesorem Snapdragon 821, 6GB RAM a na něm běžícím systémem Android 10. Byl vybrán z jednoduchého důvodu: měl prasklé krycí sklo a tak zůstal nevyužitý v šuplíku tvůrce práce. Cílem bylo demonstrovat možnost znovuvyužití starého telefonu a tak oddálit jeho prodej či odvoz do odpadu.

Této variantě se věnovala kapitola 3.7.2. Na telefon bylo v plánu nainstalovat aplikaci Termux s potřebnými rozšířeními a provést instalaci Home Assistanta a všech potřebných rozšíření do této aplikace.

Náhradní variantu představoval jednodeskový počítač Raspberry Pi s 16GB pamětovou kartou, na které by běžel operační systém založený na Home Assistantovi. V plánu bylo využít již starší Raspberry Pi B+, případně novější podobně výkonou variantu s bezdrátovým připojením namísto drátového v podobě Raspberry Pi Zero W. Jako záložní plán sloužilo případné pořízení nového Raspberry Pi 3 nebo 4.

Kapitola 6

Implementace

6.1 Úprava hlavice

K hlavici byl přidán WiFi modul NodeMCU D1 mini, jak bylo navrženo. Napájení bylo vyřešeno pomocí USB-A kabelu připojeného k modulu. Schéma zapojení následuje po vysvětlení zapojení níže. Zdrojový kód a další informace k úpravě hlavice jsou dostupné v repozitáři dostupném z [38].

K modulu byl připojen senzor teploty DS18B20 v pouzdře TO-92 včetně zhruba $5\text{k}\Omega$ rezistoru v konfiguraci pull-up. Později byl zjištěn nedostatek v měření příliš vysokých teplot senzorem, zhruba o $2\text{ }^\circ\text{C}$. V rámci řešení problému bylo vyzkoušeno snížení frekvence měření na jednu za 30 vteřin. Hlavním zdrojem problému však byly příliš krátké vodiče mezi senzorem a modulem NodeMCU D1 mini, které přenášely teplo z modulu do senzoru. Stačilo tedy namísto zhruba 7 cm dlouhých vodičů použít vodiče 15-20 cm dlouhé a senzor nechat vyčnívat dírou zespod hlavice.

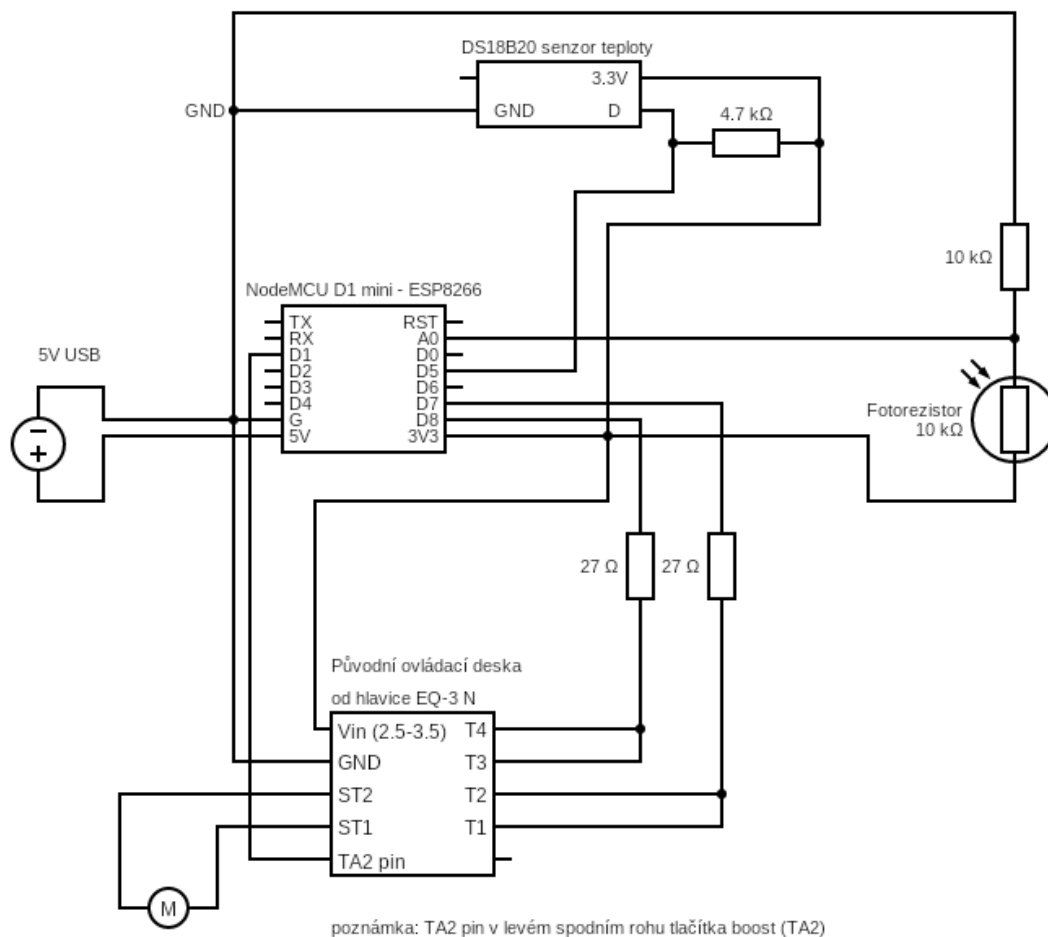
Modul NodeMCU byl připojen k původní ovládací desce od hlavice. Napájel ji pomocí 3,3V získaného z integrovaného převodníku v modulu. Využíval tranzistorů tvořících H můstek pro ovládání motoru, kde pinem D7 ovládá tranzistory T1 a T2 naráz. Obdobně s pinem D8 ovládá tranzistory T3 a T4. Mezi piny a tranzistory je po jednom rezistoru o hodnotě zhruba $20\ \Omega$. Slouží preventivně k ochraně GPIO pinů před případným přetížením. Experimentem bez využití zmíněných rezistorů taktéž nedošlo k známým poškozením pinu.

Motor hlavice při svých pohybech vyrábí hluk a vibrace přenášející se do radiátoru. Tento hluk není zanedbatelný a může rušit v případě, kdy uživatelé chtějí ve stejné místnosti usnout. Pro snížení hluku vyrobeného motorem slouží nastavení PWM výstupu z motoru. Na základě [18] je známo, že snížení PWM má pro DC motor stejný efekt jako snížení napětí. Například tedy 12V 50% PWM je pro DC motor stejné jako 6V 100% PWM. To znamená, že s tímto parametrem je potřeba nakládat opatrně, aby motor vykonával dostatečný kroutící moment. V práci bylo využito 70% PWM, kde nebyl zaznamenán problém s ovládáním motoru na testovaných ventilech.

PWM s sebou v případě základních nastavení nese nepříjemnost v podobě 1 kHz zvuku z motoru. Tento problém byl z části eliminován zvýšením frekvence PWM na 30 KHz a dále pomocí využití klasického stejnosměrného 0V výstupu při zastavení motoru namísto 0% PWM.

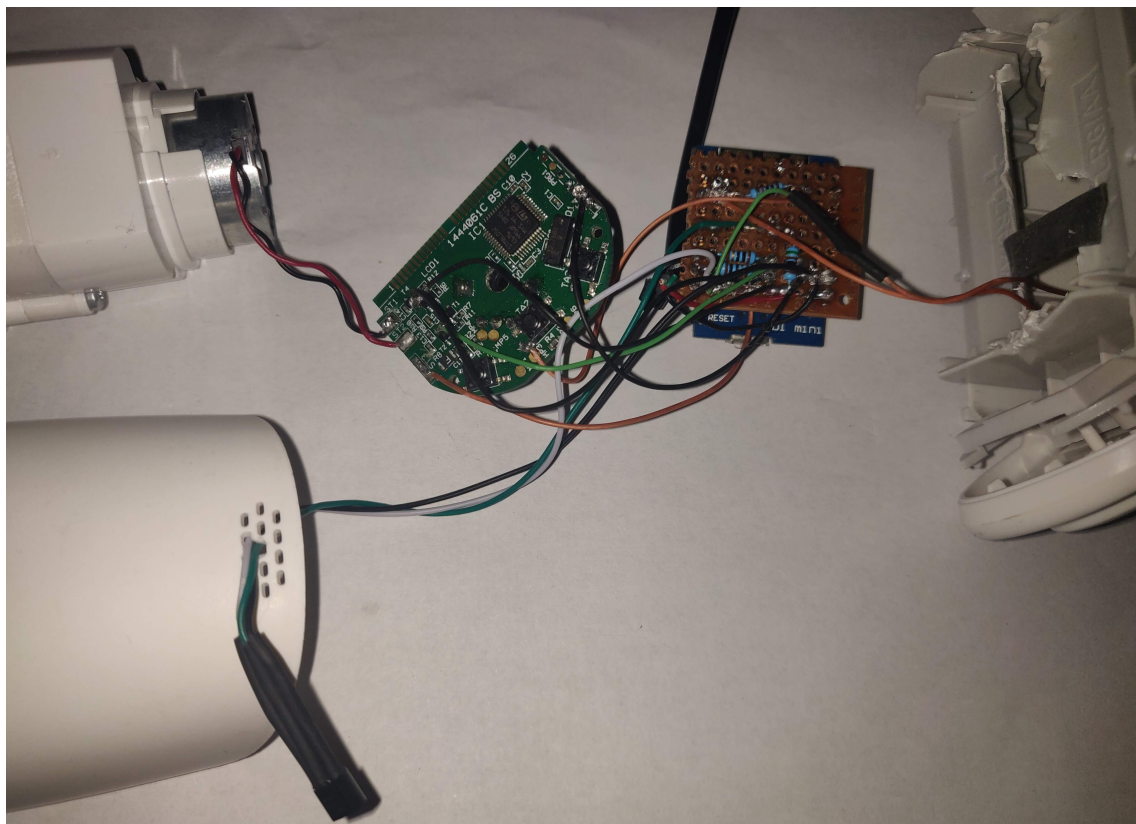
Na horní část hlavice byl umístěn senzor světla. Informace o intenzitě světla se odesílá společně s teplotou, pro regulaci vytápění nebyla využita. Mohla by však být zajímavým rozšířením.

Pro zjišťování stisku tlačítka využívá připojení na levý spodní roh tlačítka TA2. Není třeba pushup či pulldown rezistorů pro správnou detekci, jelikož tento rezistor je součástí zapojení v původní ovládací desce od hlavičky.



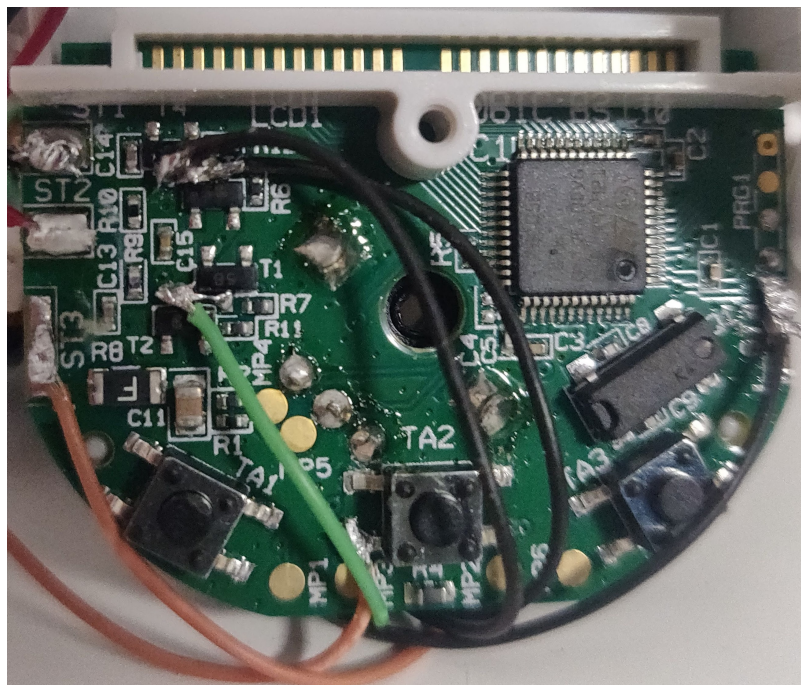
Obrázek 6.1: Schéma zapojení elektroniky v termostatické hlavičce. Vyrobeno v programu: [8].

Na následujících stranách se nacházejí fotografie hotového připojení modulu NodeMCU D1 mini k původní ovládací desce hlavice a přidaným sensorům.



Obrázek 6.2: Fotografie všech úprav komponentů hlavice v rozebraném stavu. Vlevo dole sensor teploty, vlevo nahoře motor, uprostřed původní deska hlavice a vpravo přidaný modul NodeMCU D1 mini, z něj doprava vedoucí vodiče od senzoru světla.

Nakonec se mi podařilo připájet napřímo na řídicí piny tranzistorů H můstku, pomohlo k tomu schéma z [9]. Řídicí piny PNP či NPN tranzistorů, které se v hlavici vyskytují, jsou běžně známy jako bázové. Na následující fotografii se nachází detail úprav původní hlavice včetně připojení.



Obrázek 6.3: Detail připájení na originální řídicí jednotku hlavice. do H můstku jsou připojeny 2 černé vodiče (k T3 a T4) a 1 zelený (k T1 a T2).

6.2 Řídící software hlavice

Logika otevírání a zavírání ventilu pomocí motoru měla několik etap vývoje. Program byl napsán v jazyce Embedded C v prostředí Arduino IDE, které zajišťovalo překlad, kompilaci a nahrání programu do modulu NodeMCU pomocí USB či WiFi. Následující text provádí postupným vývojem a změnami.

Základní systém využíval navržené logiky s predikcí (vizte 2.3.3) a paměť posledních dvou záznamů teploty, včetně aktuálního. Na základě trendu změny teploty a multiplikační konstanty spočítal predikovanou teplotu. Multiplikátor byl roven dvěma, později byl zvýšen na čtyři. Záznamy byly pořizovány každých 30 vteřin.

Pokud predikovaná teplota překročila hysterezi, ventil se o část odpovídající rozdílu predikované od cílové teploty vynásobeného multiplikační konstantou zavřel či otevřel. Multiplikátor pro zavření ventilu byl zhruba dvakrát tak velký, než byl k otevření ventilu.

Následovala vylepšení v podobě zvýšení paměti na posledních 6 záznamů teploty (včetně aktuálního). Rovněž byl snížen multiplikátor pro otevření ventilu a zaveden další velmi nízký multiplikátor pro otevření ventilu v případě, kdy aktuální teplota je vyšší než cílová jako ochrana proti přetápění.

Nakonec bylo toto řešení nahrazeno PID kontrolérem, který byl v této práci založen na knihovně QuickPID.h od uživatele Dloydev [13].

Pro PID kontrolér bylo potřeba najít krajní polohy ventilu. Doba od plného otevření do zavření a naopak byla experimentálně určena jako zhruba 30 vteřin, v programu byla přidána rezerva a doba zavření určena jako 40 vteřin. Po zapnutí napájení hlavice se automaticky plně otevře. Výhodou tohoto řešení je usnadnění případné instalace, jelikož hlavice v zavřeném stavu může znemožnit našroubovat hlavici k termostatickému ventilu.

Pro připojení k existující WiFi síti nebo vytvoření přístupového bodu (anglicky známo jako AP, Access Point) sloužila knihovna ESP8266WiFi.

Pro vzdálenou aktualizaci zařízení pomocí Arduino IDE slouží kód založený na řešení [19] dostupného i v příkladech v Arduino IDE. Jedná se o OTA aktualizace, což vychází z anglického Over The Air, česky přes vzduch.

Funkci mDNS responderu zajistila knihovna ESP8266mDNS.h, která umožňovala zadat namísto IP adresy zařízení adresu ve tvaru "[název WiFi SSID].local", kde WiFi SSID bylo nahrazeno názvem WiFi sítě v případě vytvoření WiFi přístupového bodu.

Na modulu běžel jednoduchý formulář pro nastavení konfiguračních informací, jehož chod zajišťovaly knihovny WiFiClient.h, ESP8266WebServer.h a pro ukládání, zpracování a načítání dat sloužila knihovna FS.h společně s ArduinoJson.h. Byl inspirován zdrojovými kódy řešení dostupných z [42], [47] a [1].

Bylo vyzkoušeno použití asynchronního webového serveru, který však nemohl být použit z důvodu příliš dlouho trvající obsluhy serveru. Přijetí, zpracování a ukládání dat na nevolatilní paměť flash trvalo příliš dlouhou dobu, což způsobilo pád a bootloop (zacyklené restartování) celého modulu s chybou "Soft WDT reset".

Aktualizace řídicího softwaru běžícího na WiFi modulu pomocí WiFi OTA aktualizací zajišťovala knihovna WiFiUdp.h ve spojení s ArduinoOTA.h.

Zajištění synchronizace času pomocí NTP protokolu pro účely automatické ochrany ventilu proti zanesení vodním kamenem měla na starosti knihovna NTPClient.h.

Čtení dat ze senzoru teploty DS18B20 měly na starosti knihovny DS18B20.h od Roba Tillaarta a OneWire.h.

Komunikace pomocí MQTT protokolu byla založena na knihovně PubSubClient.h. Zdrojový kód byl inspirován řešením dostupným z [35].

ESP Heating Valve Credentials Update

WiFi SSID
SSID of WiFi to connect to

WiFi password
password for WiFi to connect to

MQTT IP address
IP address of MQTT broker

MQTT port
port of MQTT broker

MQTT username (optional)
MQTT username

MQTT password (optional)
MQTT password

Save changes

Obrázek 6.4: Screenshot formuláře pro aktualizaci údajů. Inspirováno z [47].

6.3 Centrální jednotka

6.3.1 Hostující zařízení

Zprovoznění hostujícího zařízení pro Home Assistant [3.7](#) bylo složitější.

První snaha byla využít starší Raspberry Pi B+. Dnes dostupnou podobně výkonnou alternativou je Raspberry Pi Zero W, které namísto připojení do Ethernetu nabízí WiFi. To bylo extrémně pomalé. První nastavení zařízení trvalo okolo 8 hodin. Zdrojem těchto problémů se ukázal být nízký výkon procesoru a příliš nízké množství paměti RAM. To mělo za následek přetížení microSD karty. Dalším problémem byla nekompatibilita s Grafnou a mnoha dalšími doplňky.

Následovaly pokusy o zprovoznění na poměrně výkonném telefonu s Android 10, 6 GB RAM a procesorem Snapdragon 821. Bylo využito aplikace Termux, jak je popsáno v [3.7.2](#). Instalace Home Assistanta proběhla bez problému i přes to, že instalace Pythonu nebyla úplná kvůli chybějící podpoře v procesoru. Tato věc se projevila u rozšíření, konkrétně u MQTT brokera Mosquitto. Tomu chyběla implementace funkce `sem_lock` v Pythonu, což způsobilo kritickou chybu bránící spuštění rozšíření. Vzhledem k této skutečnosti nebyly zkoušeny instalace dalších rozšíření.

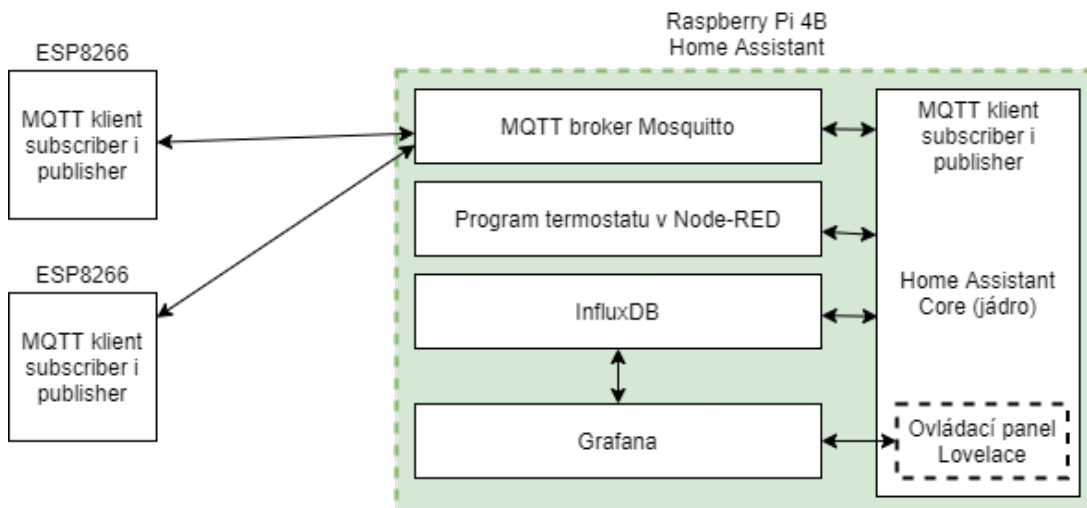
Finální úspěšnou variantou bylo pořízení Raspberry Pi 4B s 2GB RAM a instalace HASS.io. Následně se podařilo jednoduše zprovoznit Home Assistanta včetně všech potřebných rozšíření.

6.3.2 Systém centrální jednotky

Pro účely centra chytré domácnosti slouží Home Assistant. Byla k němu přidána tato rozšíření: Node-RED, Mosquitto MQTT Broker, InfluxDB, Grafana, File Editor a další, která nejsou předmětem této práce.

Základem bylo zprovoznění MQTT brokera, připojení Home Assistanta a integrace MQTT zařízení pomocí konfiguračního souboru uloženého v `/config/configuration.yaml`.

Následovalo zprovoznění základní integrované ovládací komponenty pro klimatizaci či vytápění. Ovládací panel Lovelace kromě této komponenty nabízí mnoho dalších, z nichž ně-

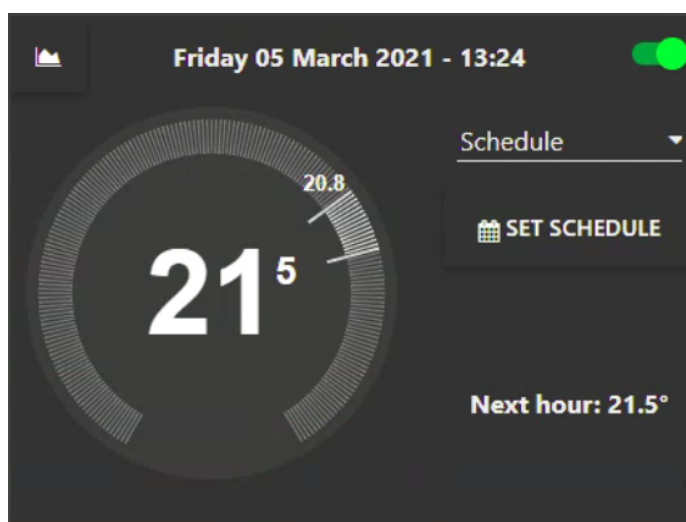


Obrázek 6.5: Schéma propojení komponent systému chytré domácnosti.

kteřé byly také využity, jako třeba graf pro zobrazení teploty. Tato komponenta neměla plánování denních rozvrhů.

Dalším krokem byl přechod na program vytvořený v Node-RED [31], dostupný z [20]. Ten umožňuje plánování denních rozvrhů, dočasných změn teplot na dobu 1 minuty až několika hodin a manuální režim. Společně se zajištěním detekce uživatele v domácnosti např. pomocí detekce telefonu v domácí síti pomocí Home Assistanta umožňuje snížit teplotu vytápění ve vybraných místnostech. Toto nastavení probíhá v zdrojovém kódu programu, nikoliv v uživatelském rozhraní.

Detekce uživatele v domácnosti byla implementována, avšak ne testována, zejména kvůli epidemiologické situaci. Domácnost tvůrce této práce nevyužívá vytápění s radiátory, natož pak termostatickými hlavicemi a tak musela být umístěna do jiné domácnosti.



Obrázek 6.6: Snímek obrazovky programu pro vylepšené ovládání vytápění.



Obrázek 6.7: Snímek obrazovky programu pro vylepšené ovládání vytápění.

Práce pokračovala napojením senzoru teploty hlavice z Home Assistanta do databáze InfluxDB.

K InfluxDB byla napojena Grafana, kde byl vytvořen pohled se zobrazením grafu teploty, senzoru světla a míry otevření ventilu. Jednotlivé veličiny se dají jednoduchým kliknutím myši skrýt. Pohled v Grafaně byl pomocí elementu iframe zobrazen v hlavním panelu a tak nahradil původní jednoduchou komponentu grafu teploty, která byla součástí Lovelace.



Obrázek 6.8: Snímek obrazovky ovládacího centra Lovelace. Vlevo nahoře ovládání termostatu, pod ním tlačítko ovládání režimu boost a pod ním předpověď počasí. Vpravo graf teplot všech místností a pod ním graf teploty, světla a pozice ventilu.

Posledním krokem bylo přidání ovládání režimu boost tlačítkem, kde zobrazuje aktuální stav režimu, jelikož se dá zapnout jak z ovládacího panelu Lovelace přes webové rozhraní, tak i pomocí stisku tlačítka na dotčené hlavici.

Po jedné malé aktualizaci Home Assistanta přestalo fungovat zaznamenávání dat z MQTT připojených zařízení. Tento problém byl po delší době hledání řešení (včetně kompletního přeinstalování systému) vyřešen úpravou syntaxe v konfiguračním souboru Home Assistanta.

Dále byla vytvořena záložka zobrazení pro telefony, kde jsou stejné karty jak v zobrazení pro počítač. Rozdíl je v tom, že jsou vyjmuty z horizontálních či vertikálních pásů karet. Díky tomu se dobře zobrazí na menší obrazovce telefonu.

Kapitola 7

Testování

Probíhalo v podkrovní místnosti bytu velikosti zhruba 4x4 m, ve které pobýval jeden uživatel. Dalšími vlivy na teplotu v místnosti jsou zařízení se spotřebou v zátěži až 300 W. V místnosti jsou dvě okna, kde jak jsem zjistil při testování, uživatel často využíval větrání s lehce pootevřeným oknem. Instalace hlavice byla provedena autorem práce, uživatel byl proškolen, jak se připojit k ovládacímu rozhraní centrálního systému.



Obrázek 7.1: Fotografie nainstalované hlavice na radiátoru.

Další veličinou je nastavení vytápění na centrálním termostatu na teplotu o zhruba 1,5 °C vyšší, než je na hlavici. Centrální termostat se nacházel na druhé straně sousední místnosti, asi 10 m daleko. Nedaleko za dveřmi se nachází trouba na pečení.

Postřehy uživatele k teplotnímu komfortu nebyly žádné, jedním z důvodů může být skutečnost, že na radiátoru nebyla dříve využita termostatická hlavice. Uživatel byl proškolen, jak používat ovládací panel Lovelace a nastavit časové plány topení, vše proběhlo bez problému. Uživateli se zalíbily široké možnosti rozšíření umožněné Home Assistantem, kon-

krétně možnost připojení reproduktorů k hostujícímu zařízení a ovládání hudby s využitím rozšíření Spotify Connect. Detekce přítomnosti uživatele nebyla využita.

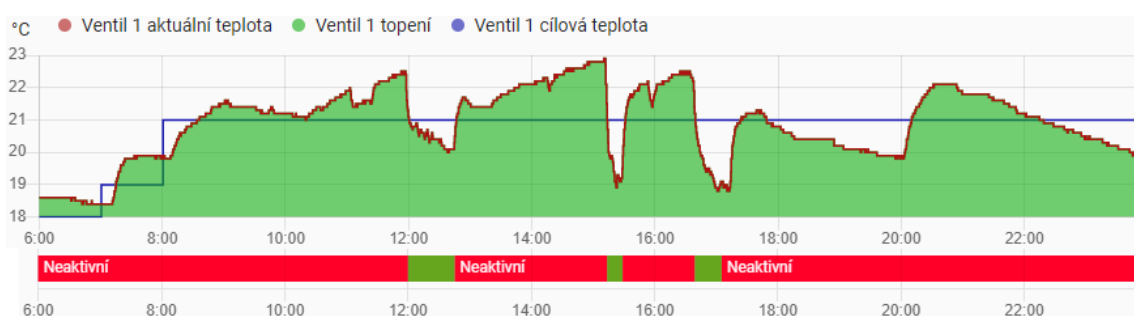
7.1 Kontrolér s predikcí

Tento kontrolér se z hlediska vyvážení všech aspektů naráz ukázal být horší, než jsem očekával. Je potřeba vyvážit dvě situace. V první je třeba výrazně otevřít ventil pro vytopení o několik °C. Druhý případ nastane, když se podaří přiblížit k cílové teplotě a je třeba zavřít ventil tak, aby topení nepřetápělo a následně otevírat ventil jen minimálně pro udržení cílové teploty.

Detekce otevřeného okna má problém s větráním, kde uživatel pouze lehce otevře okno na několik hodin. Tento stav se velmi špatně odlišuje od případu, kdy se otevřou dveře od jiné čerstvě vyvětrané místnosti. To se mnohokrát projevilo v naměřených datech. Pokles teploty je zde malý, avšak stále výrazný. Detekce otevřeného okna by se tedy mohla rozšířit o detekci pootevřeného okna a následné upozornění uživateli. Detekce otevřeného okna celkově fungovala hůře, byla vyladěna později až s příchodem PID regulátoru. Nejlépe si poradí s otevřením okna dokořán tak, jak by se správně mělo větrat.

Dalším problémem je to, že ventily neví, jestli jimi zrovna proudí teplá voda nebo ne. Mohou se tak během pár desítek minut otevřít po malých kouscích až do plně otevřeného stavu, ačkoliv je aktuální teplota pouze o několik desetín stupně nižší než požadovaná. Jakmile začne horká voda proudit, místnost se velmi rychle vytopí a ačkoliv se ventil zavře během několika málo minut, dojde k přetopení o více než 2 °C. V systémech s konstantním přísunem teplé vody nebo alespoň ekvitermní regulací 2.1.1 by k tomuto jevu nemuselo docházet tak často a výrazně. Dalším řešením by bylo propojení systému vytápění s centrálním termostatem. Díky tomu by systém znal stav centrálního zdroje tepla nebo jej mohl ovládat a tím poskytovat lepší regulaci vytápění oproti komerčně dostupným hlavicím.

Dne 14. března 2021 proběhlo měření, kde od 8:00 do půlnoci bylo nastaveno vytápění na 21 °C. Zde docházelo k výraznému přetápění. Průměrná teplota činila 21,56 °C, nejvyšší při otevřeném oknu byla 23,9 °C, nejvyšší při zavřeném oknu byla 19,8 °C.



Obrázek 7.2: Graf testování ze dne 14.3. Všimněte si průběhu po 7:00, kde je vidět opožděný přísun vody do radiátoru a následné přetopení o 1 °C. Červeně průběh aktuální teploty, modře požadovaná (cílová) teplota. Spodní graf vyjadřuje detekci otevřeného okna, červená znamená zavřeno, zelená otevřeno.

V situaci, kdy voda proudí ventilem v době, kdy to hlavice potřebuje, dochází k přetápění do 1 °C. Jak je patrné z obrázku z 12. března 2021 níže, průměrná teplota od 8:00 do půlnoci byla 21,2 °C, přičemž minimální teplota byla 17,7 °C v momentě, kdy bylo otevřeno okno. Bez otevřeného okna byla minimální teplota 20,9 °C. Cílová teplota v této době byla 21 °C.

Z obrázku je také možné zjistit, že vnější vlivy mají významný dopad na teplotu v místnosti. Od 20:02 do 22:43 byla hlavice zavřena z důvodu chybné detekce zavření otevřeného okna. Přesto došlo k přetápění o 1,5 °C. Otázkou je, jestli detekce fungovala správně, nebo bylo pouze přijmutí této zprávy Home Assistantem neúspěšné.



Obrázek 7.3: Graf testování ze dne 12.3. Mezi 20:02 a 22:43 bylo (chybně příliš dlouho) detekováno zavřené okno. Červeně průběh aktuální teploty, zeleně požadované (cílové) teploty. Spodní graf vyjadřuje detekci otevřeného okna, červená znamená zavřeno, zelená otevřeno.

Problém je, že zdaleka ne vždy se ventil a centrální termostat ve stejný čas rozhodnou vytápět a dále se projeví další proměnné. Takový případ z 15.3. je k nahlédnutí níže, došlo zde k přetápění téměř až o 3 °C. Cílová teplota mezi 8:00 a 22:00 byla 21 °C, maximální 23,8 °C, minimální 19,4 °C což bylo způsobeno otevřeným oknem. Bez otevřeného okna byla minimální teplota 20,8 °C. Průměrná teplota činila 21,6 °C.



Obrázek 7.4: Graf testování ze dne 15.3. Červeně průběh aktuální teploty, zeleně požadované (cílové) teploty. Pod grafem je průběh detekce otevřeného okna, červená znamená okno zavřeno, zelená otevřeno (a v reakci na to vypnuté topení). Spodní graf vyjadřuje detekci otevřeného okna, červená znamená zavřeno, zelená otevřeno.

Celkově tato regulace má problémy s výkyvy teploty. Výrazně by jí prospělo ladění řídicích konstant a zvýšení času predikce. Hlavice má zejména problémy s přetápěním, takže by bylo vhodné poblíž cílové teploty výrazně méně otevírat ventil a více jej zavírat. Dalším vylepšením by bylo přidat detekci a možnost zapínání zdroje tepla hlavici. Z těchto důvodů jsem zkusil využít PID kontrolér pro přesnější regulaci teploty.

7.2 PID kontrolér

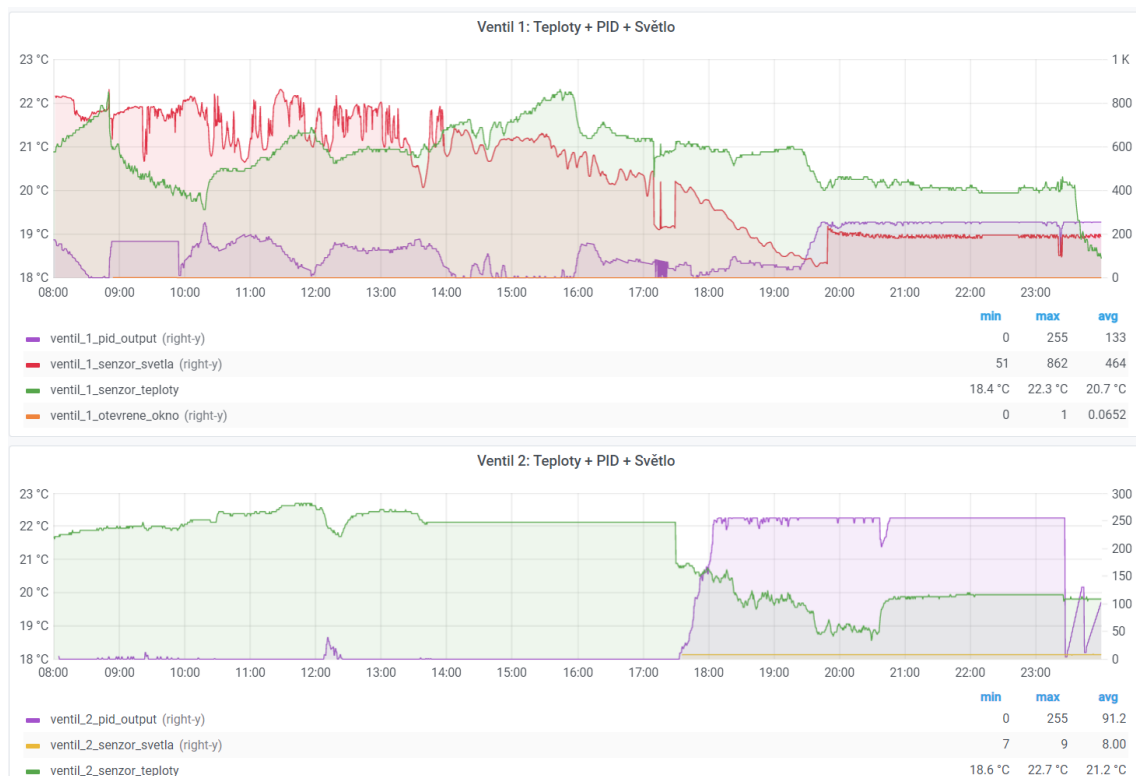
PID kontrolér byl dva měsíce testován a laděn, avšak nebyl jsem s ním schopen dojít k uspokojivému výsledku, měl tendence přetápět, ačkoliv dle výstupu PID kontroléru by měl být ventil hlavice dlouho zavřený. Lehce přes týden před termínem odevzdání této práce jsem odhalil chybu v implementaci logiky otáčení ventilu, kde došlo k přetečení proměnných a neočekávaným pohybům hlavice oproti výstupu PID kontroléru. Od 5. května 2021 večer bylo spuštěno testování opravené verze PID kontroléru.

Níže se nacházejí testovací data od 6. května do 8. května 2021. Byla sbírána ze dvou hlavice ve dvou různých místnostech, kde vytápění z důvodu relativně vysoké teploty a silného slunečního svitu dopadajícího na jihovýchodně orientovaný pokoj bylo třeba pouze ráno. Obě hlavice byly umístěny v podobně velkých i orientovaných místnostech, u druhé hlavice nebyl zapojen senzor světla. Cílová teplota ventilu 1 se v 7:00 změnila z 18 °C na 21 °C a o půlnoci zpět na 18 °C. Ventil 2 byl nastaven na neměnnou cílovou teplotu 21,5 °C.

Grafy jsou vyobrazeny s využitím Grafany [24]. Teplota je zeleně a váže se k levé ose Y. K pravé ose Y je navázán senzor světla s hodnotami 0 až 1023 vyobrazený červeně (vyšší číslo znamená více světla), poměr otevření hlavice byl vykreslen fialově, kde 0 odpovídá plně zavřenému ventilu a 255 plně otevřenému.

První situace z 6. května 2021 je vyobrazena níže. V místnosti, kde byl umístěn ventil 1 (horní graf), bylo od 8:50 občas mírně pootevřeno okno, což mělo za následek postupné snížení teplot pod cílovou teplotu. Od této doby se na základě nulové reakce teploty na otevření ventilu a skutečnosti, že v místnosti s centrálním termostatem nebyla otevřena okna dá usuzovat, že vytápění nebylo použito nebo bylo použito jen na krátké chvíle.

U hlavice 1 byla od 8:00 do půlnoci naměřena průměrná teplota 20,7 °C, standardní odchylka rovna 0,73, maximální teplota 22,3 °C a minimální 18,4 °C s otevřeným oknem, v případě zavřeného 20 °C. U hlavice 2 byla od 8:00 do půlnoci zjištěna průměrná teplota 21,2 °C, standardní odchylka rovna 1,28, maximální teplota 22,7 °C a minimální teplota 18,6 °C v případě otevřeného okna, jinak 20,7 °C.



Obrázek 7.5: Grafy testování ze dne 6.5. Zeleně teplota, červeně intenzita slunečního svitu (bez jednotky, 0 až 1023) a fialově poloha hlavice (0 zavřeno až 255 otevřeno).

Pro srovnání obrázek o stranu níže vyjadřuje situaci ze dne 8. května 2021, kdy topení bylo ručně vypnuto na centrálním termostatu, venkovní teplota dosahovala až 15 °C. Pověšimněte si závislosti teploty místnosti na slunečním svitu, kde došlo k nárůstu teploty v první místnosti z ranních 18,7 °C až k 22,0 °C a v případě druhé místnosti z 19,4 °C na 22,6 °C.

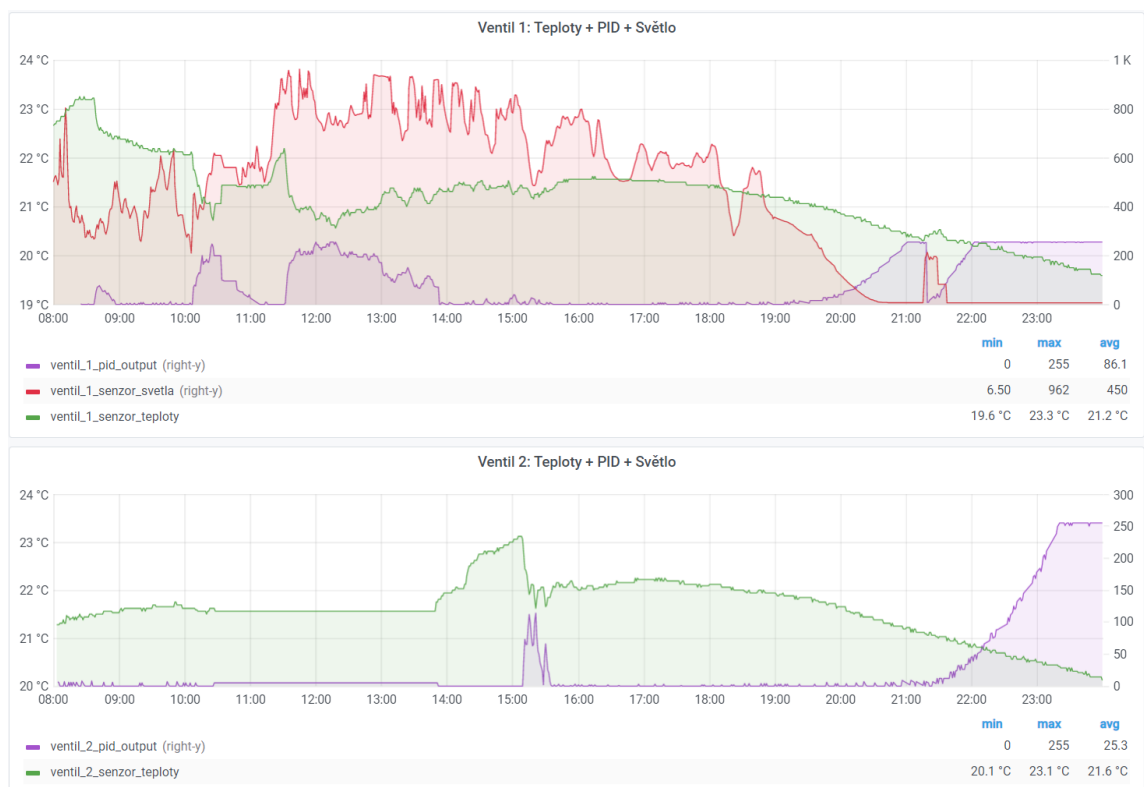


Obrázek 7.6: Grafy testování ze dne 8.5. s vypnutým centrálním termostatem. Zeleně teplota, červeně intenzita slunečního svitu (0 až 1023).

Následující situací je měření z 7. května 2021. Zde z neznámé příčiny i přes zavřený ventil 1 ráno došlo k přetápění až k 23 °C, kdy následně bylo otevřeno okno. To je v rozporu s ventilem 2, kde vytápění ráno fungovalo dle očekávání a došlo k přetopení asi o 0,3 °C, což považuji za skvělý výsledek. Následoval výpadek hlavice mezi 10:30 a 14:00 způsobený vadným napájecím zdrojem. Večer v 21 hodin byla provedena kontrola a restartování první hlavice. Z tohoto důvodu zde byl skokový pád polohy hlavice na nulu.

U hlavice 1 byla od 8:00 do půlnoci naměřena průměrná teplota 21,2 °C, standardní odchylka rovna 0,76, maximální teplota 23,3 °C a minimální 19,6 °C. U hlavice 2 byla od 8:00 do půlnoci zjištěna průměrná teplota 21,6 °C, standardní odchylka rovna 0,65, maximální teplota 23,1 °C a minimální teplota 20,1 °C.

Rozpor mezi těmito dvěma ventily v podobných místnostech byl dle mého názoru způsoben otevřenými dveřmi v místnosti prvního ventilu a zavřenými v místnosti druhého ventilu.



Obrázek 7.7: Grafy testování ze dne 7.5. Zeleně teplota, červeně intenzita slunečního svitu (bez jednotky, 0 až 1023) a fialově poloha hlavice (0 zavřeno až 255 otevřeno).

Kapitola 8

Závěr

Cílem této práce bylo upravit levnou digitální hlavici pro vytápění a připojit ji do systému chytré domácnosti. Pro tento účel jsem upravil hlavici Equiva EQ3-N tak, že jsem přidal WiFi modul NodeMCU D1 mini založený na ESP8266. K němu jsem následně připojil senzor tepla a světla. Vytvořil jsem řídicí software hlavice, který pomocí MQTT komunikuje s centrální jednotkou Home Assistant založenou na Raspberry Pi 4B, kterou jsem rovněž zprovoznil a přidal k ní rozšíření pro vizualizaci a příjemné ovládání.

Obě řešení by bylo vhodné vylepšit rozšířením o detekci zapnutého zdroje tepla. Nejlepší možností by bylo pomocí ventilů ovládat zdroj tepla.

Očekával jsem, že se mi pomocí predikce budoucí teploty povede vytápění regulovat přesněji. Udržení teploty se mi podařilo s teplotami lišícími se, v závislosti na mnoha faktorech, s maximálním přesahem o 0,8 °C v jeden den oproti 3 °C v den jiný. Teplota při udržování teploty neklesala pod cílovou, nýbrž ji překračovala. Problém nastal například při vytápění ráno, kdy se hlavice najednou snažila zvýšit teplotu místnosti o 3 °C. Zde hlavice přetápěla až o 1,5 °C. Toto řešení by bylo možné vylepšit pomocí predikce pro delší časové intervaly.

Následně jsem provedl řešení pomocí PID kontroléru. Ten se ukázal být složitějším na nastavení a bylo by vhodné k němu přidat automatickou úpravu řídicích konstant. Toto řešení se jeví funkčněji než předešlé, potřebuje však důkladněji otestovat v reálném provozu zimních teplot. Rozdíl maximální a minimální teploty se pohybuje okolo 2 °C. Řešením by mohlo být PID autotune, které ovšem potřebuje vyvolávat kolísání teploty. Vhodnější by bylo využít PID self-tune namísto autotune, které tento problém odstraňuje.

Zprovoznění ovládacího systému chytré domácnosti bylo složitější, než jsem očekával, pokud však uživatel využije doporučené zařízení, nebude mít problémy. Home Assistant umožňuje velmi jednoduše přidávat mnoho druhů a značek zařízení.

Vyrobený prototyp umožňuje připojit hlavici vytápění do systému chytré domácnosti a po vyladění řízení pro konkrétní instalaci komfortní vytápění. Instalace je stále velmi jednoduchá, pro napájení hlavice stačí starší nabíječka s USB-A portem. Design hlavice je až na drobnosti zachován a jako celek může vytvořit dobrý podklad pro budoucí produkt.

Na práci se dá navázat lepším odladěním PID kontroléru pro konkrétní potřebu s využitím světelného senzoru, senzoru pohybu či dokonce detekcí telefonu v domácí síti pro snížení teploty vytápění v případě nepřítomnosti uživatelů. Senzor světla byl v práci nainstalován na ventil, kde může posloužit jako vstup pro regulaci vytápění.

Literatura

- [1] ACROBOTIC. *Wifi Modes Switch* [online]. Github, říjen 2018 [cit. 2021-05-2]. Dostupné z: https://github.com/acrobotic/Ai_Tips_ESP8266/blob/master/wifi_modes_switch/wifi_modes_switch.ino.
- [2] ARDUINO. *Arduino IDE 1.8.13* [online]. [cit. 2021-04-26]. Dostupné z: <https://www.arduino.cc/en/software>.
- [3] ASSETWOLF. *Quality of Service (QoS)* [online]. Tribal Ltd, duben 2021 [cit. 2021-04-04]. Dostupné z: <https://assetwolf.com/learn/mqtt-qos-understanding-quality-of-service>.
- [4] ASSISTANT, H. *Home Assistant* [online]. Google Play Store, duben 2020 [cit. 2021-04-28]. Dostupné z: https://play.google.com/store/apps/details?id=io.homeassistant.companion.android&hl=en_US&gl=US.
- [5] BEAUREGARD, B. *Arduino PID Autotune Library* [online]. Brettbeauregard.com, leden 2012 [cit. 2021-04-29]. Dostupné z: <http://brettbeauregard.com/blog/2012/01/arduino-pid-autotune-library/>.
- [6] BENHAROSH, J. *What is REST API?* [online]. phpenthusiast, červenec 2018 [cit. 2021-05-07]. Dostupné z: <https://phpenthusiast.com/blog/what-is-rest-api>.
- [7] CESARANO, L. *Install Home Assistant (HASS) on Android (NO ROOT)* [online]. Medium Corporation, září 2020 [cit. 2021-04-16]. Dostupné z: <https://lucacesarano.medium.com/install-home-assistant-hass-on-android-no-root-fb65b2341126>.
- [8] CIRCUIT, D. *Circuit Diagram* [online]. [cit. 2021-05-02]. Dostupné z: <https://www.circuit-diagram.org/editor/>.
- [9] CONFILE. *Modifying a thermostatic radiator valve for micro control* [online]. [cit. 2021-04-27]. Dostupné z: <https://electronics.stackexchange.com/questions/206014/modifying-a-thermostatic-radiator-valve-for-micro-control>.
- [10] CONRAD. *Eqiva CC-RT-N / 132231 N termostatická hlavice elektronický 5 do 29.5 °C* [online]. Conrad Electronic Česká republika, s.r.o., listopad 2020 [cit. 2021-04-04]. Dostupné z: <https://www.conrad.cz/p/eqiva-cc-rt-n-132231-n-termostaticka-hlavice-elektronicky-5-do-295-c-1168832>.
- [11] COPE, S. *MQTT Clean Sessions and QOS Examples* [online]. Steve's internet Guide, březen 2021 [cit. 2021-04-10]. Dostupné z: <http://www.steves-internet-guide.com/mqtt-clean-sessions-example/>.

- [12] DIYIOT. *MQTT Tutorial for Arduino, ESP8266 and ESP32* [online]. DiyIoT, květen 2021 [cit. 2021-05-07]. Dostupné z: <https://diyiot.com/introduction-into-mqtt/>.
- [13] DLLOYDEV. *QuickPID* [online]. [cit. 2021-03-15]. Dostupné z: <https://github.com/Dlloyddev/QuickPID>.
- [14] DSM, R. K. *MQTT vs REST: IoT Implementation* [online]. Bevywise, duben 2018 [cit. 2021-04-04]. Dostupné z: <https://www.bevywise.com/blog/mqtt-vs-rest-iot-implementation/>.
- [15] ENGINEERING, C. *Pros and cons of autotuning control: Part 1* [online]. Control Engineering, červen 2018 [cit. 2021-05-10]. Dostupné z: <https://www.controleng.com/articles/pros-and-cons-of-autotuning-control-part-1/>.
- [16] ENGINEERING, I. H. *Thermostatic head D* [online]. IMI Hydronic Engineering, září 2020 [cit. 2021-04-04]. Dostupné z: https://imihydronic.blob.core.windows.net/resources/Documents/Catalogues/English_International/PDF_low/Thermostatic-head-D_EN_lq.pdf.
- [17] EQ 3. *Radiator Thermostat Model N* [online]. EQ-3, duben 2021 [cit. 2021-05-01]. Dostupné z: <https://www.eq-3.com/products/homematic/detail/radiator-thermostat-model-n.html>.
- [18] GERSHENFELD, P. N. *PWM for DC Motors* [online]. The MIT Center for Bits and Atoms [cit. 2021-05-05]. Dostupné z: https://fab.cba.mit.edu/classes/865.18/power/PWM_motor_control_rev5.pdf.
- [19] GITHUB přispěvovatelé. *BasicOTA* [online]. [cit. 2021-05-08]. Dostupné z: <https://github.com/esp8266/Arduino/blob/master/libraries/ArduinoOTA/examples/BasicOTA/BasicOTA.ino>.
- [20] GIUSEPPEG88. *The Ultimate Node-RED Thermostat for Home Assistant* [online]. [cit. 2021-04-26]. Dostupné z: <https://flows.nodered.org/flow/25857cba1ab7e34309133ac8354004c6>.
- [21] HOME ASSISTANT autoři. *Home Assistant* [online]. Home Assistant, září 2020 [cit. 2021-04-04]. Dostupné z: <https://www.home-assistant.io/>.
- [22] HOME ASSISTANT autoři. *Zigbee Home Automation* [online]. Home Assistant, duben 2021 [cit. 2021-05-01]. Dostupné z: <https://www.home-assistant.io/integrations/zha>.
- [23] INC., I. *InfluxDB* [online]. [cit. 2021-04-17]. Dostupné z: <https://www.influxdata.com/>.
- [24] LABS, G. *Grafana* [online]. [cit. 2021-04-19]. Dostupné z: <https://grafana.com/>.
- [25] LUGAZ, C. *TA(L)KING CONTROL Hibachi self-heating lunchbox* [online]. China Hardware Innovation Camp, září 2020 [cit. 2021-04-20]. Dostupné z: <https://chi.camp/blog/2016/05/26/talking-control/>.
- [26] MALÝ, M. *REST: architektura pro webové API* [online]. Zdroják.cz, srpen 2009 [cit. 2021-04-04]. Dostupné z: <https://zdrojak.cz/clanky/rest-architektura-pro-webove-api/>.

- [27] MATLAB. *Photoresistor Motion Detection* [online]. Matlab, prosinec 2014 [cit. 2021-04-13]. Dostupné z: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/47605/versions/2/previews/html/Photoresistor_Motion_Detection_Article.html?access_key=.
- [28] MAXIM INTEGRATED, I. *DS18B20, Programmable Resolution 1-Wire Digital Thermometer* [online]. Maxim Integrated, Inc, březien 2021 [cit. 2021-04-08]. Dostupné z: <https://www.maximintegrated.com/en/products/sensors/DS18B20.html>.
- [29] NABU CASA, I. *Home Assistant Cloud* [online]. Nabu Casa, Inc., prosinec 2020 [cit. 2021-04-04]. Dostupné z: <https://www.nabucasa.com/>.
- [30] NABUCASA, E. *ESPHome* [online]. [cit. 2021-04-17]. Dostupné z: <https://esphome.io/index.html>.
- [31] NODE RED, A. *Node-RED* [online]. [cit. 2021-04-17]. Dostupné z: <https://nodered.org/>.
- [32] POSTAVROBOTA.CZ. *Digitální teplotní čidlo DS18B20* [online]. FutureTronic spol. s.r.o., duben 2021 [cit. 2021-04-21]. Dostupné z: <https://www.postavrobota.cz/Digitalni-teplotni-cidlo-DS18B20-d320.htm>.
- [33] PRODEJNA, A. *Comap: R855324 R855E priamy termostatický ventil* [online]. Attack prodejna, květen 2021 [cit. 2021-04-28]. Dostupné z: https://attackpredajna.sk/produkty/355_termostaticke-ventily/2041_16448-comap-r855324-r855e-priamy-termostaticky-ventil-s-prednastavenim-1-2-3-4-ek-m30.
- [34] SABOO, A. *Homeassistant-config* [online]. [cit. 2021-04-26]. Dostupné z: <https://github.com/arsaboo/homeassistant-config>.
- [35] SANTOS, N. *ESP8266: Connecting to MQTT broker* [online]. Techtutorialsx, září 2020 [cit. 2021-05-02]. Dostupné z: <https://techtutorialsx.com/2017/04/09/esp8266-connecting-to-mqtt-broker/>.
- [36] SHINJI KIM, R. B. A. *An Introduction to MQTT: Why HTTP isn't the King of the Internet of Things* [online videoprezentace]. Samsung na Samsung developer konferenci 2017, červen 2017 [cit. 2021-04-08]. Dostupné z: <https://www.youtube.com/watch?v=LKz1jYngpcU>.
- [37] SHOP, T. E. *Honeywell Home evohome Starter Kit - UK Version* [online]. The EVOHOME Shop, duben 2021 [cit. 2021-04-25]. Dostupné z: <https://theevohomeshop.co.uk/honeywell-home-evohome-connected-value-packs/301-honeywell-evohome-starter-kit.html>.
- [38] SMEJKAL, J. *Heating radiator valve with WiFi+MQTT* [online]. Github, Inc., květen 2021 [cit. 2021-05-10]. Dostupné z: https://github.com/Smejky338/Heating_ESP_control_with_WiFi_MQTT.
- [39] SOUSA NEVES, M. G. de. *ELAY METHOD ON AUTO-TUNING AUTOMATION SOLUTIONS* [online]. Universidade Técnica de Lisboa, Instituto Superior Técnico, Lisboa, Portugal, září 2020 [cit. 2021-04-17]. Dostupné z: https://www.avrfreaks.net/sites/default/files/forum_attachments/Resumo%20Alargado%20-%20Auto-tuning%20de%20Controladores%20PID%20pe%20m%C3%A9todo%20Relay.pdf.

- [40] S.R.O., C. *Honeywell Evohome Starter Set CZ, Evohome Touch WiFi + 3x termohlavice + BDR91, +5% ErP 8* [online]. CZC.cz s.r.o., říjen 2020 [cit. 2021-04-04]. Dostupné z: <https://www.czc.cz/honeywell-evohome-starter-set-cz-evohome-touch-wifi-3x-termohlavice-bdr91-5-erp-8/192515/produkt>.
- [41] TEAM, P. E. *PID Controller Explained* [online]. Pid Explained Team, březen 2021 [cit. 2021-04-20]. Dostupné z: <https://pidexplained.com/pid-controller-explained/>.
- [42] TECHNOLOGIES, M. *AdvancedWebServer* [online]. Github, březen 2021 [cit. 2021-05-2]. Dostupné z: <https://github.com/espressif/arduino-esp32/blob/master/libraries/WebServer/examples/AdvancedWebServer/AdvancedWebServer.ino>.
- [43] TINDIE. *L9110S Motor Driver Controller Board* [online]. Tindie, květen 2021 [cit. 2021-05-01]. Dostupné z: <https://www.tindie.com/products/blinkingsilver/l9110s-motor-driver-controller-board/>.
- [44] TRAUMFLUG. *ISTAtrol heating valve controller* [online]. Github Inc., březen 2016 [cit. 2021-04-22]. Dostupné z: <https://github.com/Traumflug/ISTAtrol/blob/master/firmware/main.c>.
- [45] TRAUMFLUG. *Hacking an EQ-3 Model N* [online]. Reprap DIY, květen 2018 [cit. 2021-04-26]. Dostupné z: https://www.reprap-diy.com/hacking_an_eq-3_model_n.
- [46] VESTEMI. *Radbot* [online]. [cit. 2021-04-27]. Dostupné z: <https://vestemi.com/products/>.
- [47] W3SCHOOLS. *Try CSS forms* [online]. W3schools, duben 2021 [cit. 2021-05-03]. Dostupné z: https://www.w3schools.com/css/tryit.asp?filename=trycss_forms.
- [48] WEMOS. *LOLIN D1 mini* [online]. Wemos, duben 2019 [cit. 2021-04-13]. Dostupné z: https://www.wemos.cc/en/latest/_static/boards/d1_mini_v3.1.0_1_16x16.jpg.
- [49] WEMOS.CC. *LOLIN D1 mini* [online]. [cit. 2021-04-04]. Dostupné z: https://www.wemos.cc/en/latest/d1/d1_mini.html#lolin-d1-mini.
- [50] WIKIPEDIE přispěvatelé. *Topologie sítě ZigBee* [online]. Wikipedie: Otevřená encyklopedie, listopad 2007 [cit. 2021-04-28]. Dostupné z: https://cs.wikipedia.org/wiki/ZigBee#/media/Soubor:Topologie_siti.jpg.
- [51] WIKIPEDIE přispěvatelé. *Ekvitermní regulace* [online]. Wikipedie: Otevřená encyklopedie, září 2019 [cit. 2021-04-28]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=Ekvitermn%C3%AD_regulace&oldid=17623368.
- [52] WIKIPEDIE přispěvatelé. *Bang–bang control* [online]. Wikipedie: Otevřená encyklopedie, prosinec 2020 [cit. 2021-04-20]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Bang%E2%80%93bang_control&oldid=995103448.
- [53] WIKIPEDIE přispěvatelé. *Dynamic Host Configuration Protocol* [online]. Wikipedie: Otevřená encyklopedie, září 2020 [cit. 2021-04-04]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Dynamic_Host_Configuration_Protocol&oldid=1014205561.

- [54] WIKIPEDIE přispěvatelé. *Ústřední topení* [online]. Wikipedie: Otevřená encyklopedie, červen 2020 [cit. 2021-04-28]. Dostupné z:
https://cs.wikipedia.org/w/index.php?title=%C3%9Ast%C5%99edn%C3%AD_topen%C3%AD&oldid=18773221.
- [55] WIKIPEDIE přispěvatelé. [online]. Wikipedie: Otevřená encyklopedie, duben 2021 [cit. 2021-04-17]. Dostupné z:
<https://en.wikipedia.org/w/index.php?title=H-bridge&oldid=1018895005>.
- [56] WIKIPEDIE přispěvatelé. *ESP8266* [online]. Wikipedie: Otevřená encyklopedie, duben 2021 [cit. 2021-04-04]. Dostupné z:
<https://en.wikipedia.org/w/index.php?title=ESP8266&oldid=1015321424>.
- [57] WIKIPEDIE přispěvatelé. *Home Assistant* [online]. Wikipedie: Otevřená encyklopedie, duben 2021 [cit. 2021-04-15]. Dostupné z:
https://en.wikipedia.org/w/index.php?title=Home_Assistant&oldid=1016313290.
- [58] WIKIPEDIE přispěvatelé. *Wi-Fi* [online]. Wikipedie: Otevřená encyklopedie, květen 2021 [cit. 2021-05-01]. Dostupné z:
<https://en.wikipedia.org/w/index.php?title=Wi-Fi&oldid=1020818217>.
- [59] WIKIPEDIE přispěvatelé. *Zigbee* [online]. Wikipedie: Otevřená encyklopedie, duben 2021 [cit. 2021-05-01]. Dostupné z:
<https://en.wikipedia.org/w/index.php?title=Zigbee&oldid=1020635085>.