



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

**SPRÁVA A AUTOMATIZACE SYSTÉMU
VYTÁPĚNÍ PODNIKU**

MANAGEMENT AND AUTOMATION OF ENTERPRISE FACILITY HEATING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JIŘÍ HARTMANN

VEDOUcí PRÁCE

SUPERVISOR

Ing. VÁCLAV ŠIMEK

BRNO 2021

Zadání diplomové práce



Student: **Hartmann Jiří, Bc.**
Program: Informační technologie
Obor: Počítačové sítě a komunikace
Název: **Správa a automatizace systému vytápění podniku**
Management and Automation of Enterprise Facility Heating System
Kategorie: Vestavěné systémy
Zadání:

1. Seznamte se s existujícími systémy pro řízení vytápění a obslužných procedur v daném podniku.
2. Prostudujte možnosti komunikace řídicích prvků a navrhnete vhodné připojení k systému řízení.
3. Vytvořte návrh architektury systému, který bude zajišťovat řízení vytápění a chlazení. Systém bude zahrnovat webové uživatelské rozhraní se zobrazením stavových hodnot, možností řízení aktivních prvků systému a konfiguraci hodnot vyžadovaných systémem řízení.
4. Proveďte realizaci uživatelského rozhraní a připojení jednotlivých funkčních prvků do systému.
5. Demonstrujte funkčnost vytvořeného systému a zhodnoťte dosažené výsledky. Pokuste se navrhnout případná rozšíření či vylepšení.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

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

Vedoucí práce: **Šimek Václav, Ing.**
Vedoucí ústavu: Sekanina Lukáš, prof. Ing., Ph.D.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 19. května 2021
Datum schválení: 30. října 2020

Abstrakt

Cílem této práce je vytvořit jednotný řídicí a dohledový systém, SCADA, pro řízení vytápění a chlazení podniku. Systém je ovládán pomocí PLC jednotek UniPi. Jako řídicí software je použit Node-RED. Uživatelské rozhraní tvoří rozšiřující modul dashboard. Komunikace prvků probíhá pomocí protokolu MQTT. Pro ukládání dat je využita databáze InfluxDB. Pro vizualizaci historických dat je použit nástroj Grafana. Vytvořený systém je z drtivé většiny tvořen otevřeným softwarem. Systém je univerzální, rozšiřitelný a je možná propojit jej s jiným systémem. Systém je možné využít nebo upravit na podobné problémy pokročilého řízení vytápění.

Abstract

The aim of this work is to create a unified control and supervision system, SCADA, for controlling the heating and cooling of the company. The system is controlled by UniPi PLC units. Node-RED is used as control software. The user interface is created by an extension "dashboard". For communication between elements using the MQTT protocol. The InfluxDB database is used for data storage. The Grafana tool is used to visualize historical data. The majority of the created system consists of open software. The system is universal, expandable and it is possible to connect it with another system. The system can be used or adapted to similar problems of advanced heating control.

Klíčová slova

Systém vytápění, SCADA systém, Node-RED, UniPi Technology, 1-Wire, MQTT protokol, Grafana, InfluxDB, Docker, Raspberry Pi.

Keywords

HVAC system, SCADA system, Node-RED, UniPi Technology, 1-Wire, MQTT protocol, Grafana, InfluxDB, Docker, Raspberry Pi.

Citace

HARTMANN, Jiří. *Správa a automatizace systému vytápění podniku*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Václav Šimek

Správa a automatizace systému vytápění podniku

Prohlášení

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

.....
Jiří Hartmann
18. května 2021

Poděkování

Chtěl bych poděkovat za cenné rady a jejich trpělivost pánům Ing. Václavu Šimkovi, Ing. Filipu Poláčkovi a Ing. Tomáši Hinkovi.

Obsah

1 Úvod	2
2 Teoretická východiska	4
2.1 Tepelné čerpadlo	4
2.2 Rosný bod	5
2.3 Termohlavice – ventily – čerpadla	6
2.4 Ekvitermní regulace	9
2.5 Jističe a stykače	10
2.6 Ethernet - oddělení sítě	11
2.7 UniPi PLC a Mervis	13
2.8 1-Wire sběrnice – zařízení a řídicí software	15
2.9 MQTT protocol	19
2.10 Node-RED	20
2.11 Virtuální server a Docker kontejner	22
3 Analýza současného stavu	24
4 Vlastní návrh řešení	27
5 Realizace řešení	31
5.1 Řízení tepelných čerpadel	33
5.2 Hlavní ovládací panel	33
5.3 InfluxDB a Grafana	34
5.4 Node-RED osvědčené postupy	35
6 Testování a možnosti rozšíření	38
7 Závěr	39
Literatura	40
A Webové rozhraní – nastavení	43
B Webové rozhraní – diagnostika	44

Kapitola 1

Úvod

Každá dnešní budova je vybavena systémem vytápění. Pokročilejší budovy obsahují systém ventilace za pomoci automatického otevírání oken nebo zabudované komplexnější vzduchotechniky. V dnešních sociálních standardech bývá v letních měsících potřeba chlazení. Chlazení v kancelářích typicky bývá řešeno pomocí nezávislých stropních jednotek, protože jsou nejjednodušší na instalaci. Dále následují systémy s vnitřní a vnější jednotkou, které jsou typicky nainstalovány na rodinných domech a panelácích. Vrchol toho všeho je klimatický systém, který kombinuje vzduchotechniku, topení a chlazení. Typický bývá navržen rovnou v plánech budovy a instalován při stavbě, případně kompletní rekonstrukci.

Dalším důležitým aspektem je spotřeba energie a k tomu odpovídající zdroje energie. Už jen samotné zjištění spotřeby energie pro vytápění, vyžaduje alespoň základní model konstrukce budovy, aby mohl být vypočten alespoň odhad spotřeby energie, případně vycházet z dřívějších zkušeností. V současnosti se jeví tepelné čerpadlo jako dobrý zdroj tepelné energie. Svoji vysokou pořizovací cenu kompenzuje nižší spotřebou energie oproti ostatním zdrojům. Jeho nevýhoda spočívá v klesání účinnosti při zvětšujícím se rozdílu teplot, tudíž delší souvislé úseky velmi chladných dnů mohou tepelné čerpadlo dostat na limit svých možností. Proto bývá tepelných čerpadel nainstalováno více nebo jsou nainstalovány pomocné zdroje na plyn, elektřinu, atd. Pokud budete vytápění ve firmě, můžete si typicky dovolit vyšší pořizovací náklady, protože se náklady rozloží na následném úsporném provozu. Pokud máte rodinný domek postavený před rokem 1989, tak už jste pravděpodobně zažili minimálně jednu výměnu zdroje tepelné energie. Například z uhelného kalu na elektřinu, z elektřiny na plyn, z plynu na tepelné čerpadlo, dřevo, dřevěné pelety, uhelné pelety, uhlí, znova uhelné kaly [3]. Pokud máte nový úsporný domek, pravděpodobně zvolíte tepelné čerpadlo nebo dřevěné pelety, ovšem při pohledu na minulost lze někdy v budoucnu předpokládat změnu zdroje tepelné energie.

Ať už máte cokoli z výše zmíněného, tak všechny tyto věci vyžadují systém řízení. Některé z nich obsahují již svůj vlastní vestavěný systém řízení, kterému se zadají požadované parametry. Případně si pořídíte jeden termostat, který postavíte do jedné místnosti a připojíte ke kotli, čímž vytvoříte primitivní systém řízení. Máte více nezávislých systémů například pro vzduchotechniku a topení. Máte v provozu jednu část v jednom řídicím systému a potřebujete připojit jinou část, která je od jiného výrobce. Tato práce vám ukáže možné řešení výše zmíněných problémů a možnosti integrace jednotlivých systémů do funkčního celku.

V teoretické části jsou popsány pojmy a zařízení se kterými jsem se při tvorbě této práce zabýval. Větší důraz jsem logicky kladl na věci spojené s informačními technologiemi. Tuto práci je možno použít jako ukázkou možného řešení, které si následně můžete upravit pro vaše

potřeby. V této práci je využito otevřeného systému Node-RED. Ten lze využít pro různé účely. Je tedy možné toto řešení upravit a použít pro jiné podobné účely. Navíc je možné systém dále rozšiřovat o moduly a tím dosáhnout kompatibility teoreticky s jakýmkoli zařízením.

Cíle práce Mým úkolem je zefektivnit ovládání kotelny ve společnosti SPOLEČNOST-24 s.r.o. Hlavním cílem této práce je vytvořit systém pro řízení a obsluhu zařízení kotelny tak, aby bylo možné pomocí jednotného ovládacího panelu nastavit a řídit všechny zařízení v kotelně s ohledem na jejich návaznost. Jde o vytvoření grafického uživatelského rozhraní, které bude zobrazovat schématické znázornění celé soustavy, aby bylo možno se relativně snadno orientovat. Do tohoto schématu budou promítnuty stavy jednotlivých zařízení, hodnoty ze senzorů a možnost měnit nastavení akčních členů a prvků systému. Uspořádání grafického uživatelského rozhraní by uživateli mělo umožnit snadné a rychle zorientování se a tím i možnost adekvátního zásahu do systému. V podstatě se jedná o vytvoření SCADA systému pro konkrétní zařízení v kotelně společnosti SPOLEČNOST-24 s.r.o.

Požadavky řešení:

- Ovládání tepelných čerpadel a vzduchotechniky
- Zónová regulace na úrovni jednotlivých místností
- Implementace režimů Topení, Chlazení a Vypnuto
- Automatické zajištění funkčnosti režimů
- Ovládání z centrálního panelu
- Jednoduché ovládání systému údržbářem společnosti
- Důraz na spolehlivost řešení
- Zajištění bezpečnosti řešení
- Ošetření neočekávaných a mezních stavů

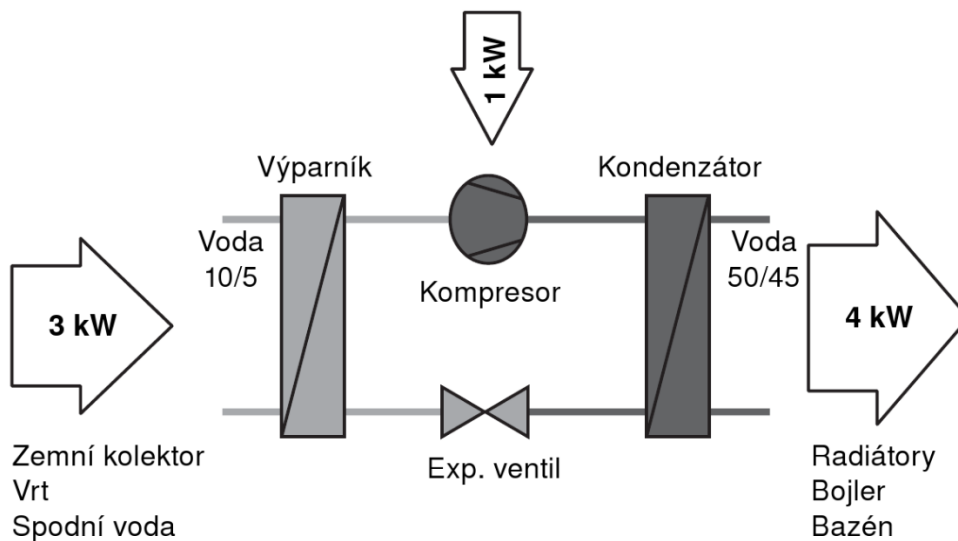
Kapitola 2

Teoretická východiska

2.1 Tepelné čerpadlo

Tepelné čerpadlo je v současné době považováno za nejlepší možný zdroj tepelné energie. Je to dáno tím, že příkon nebo-li elektrická energie dodaná tepelnému čerpadlu je v ideálním případě menší, než výstupní tepelná energie. V případě, že příkon tepelného čerpadla je větší, než výstupní tepelný výkon je tepelné čerpadlo v podstatě degradováno na konvenční zdroj tepelné energie [20].

Princip činnosti tepelného čerpadla spočívá v čerpání tepla z chladnějšího místa do teplejšího 2.1. Této činnosti, kdy chladnější prostor ještě více ochlazujeme a teplejší prostor ještě více zahříváme může být v současnosti dosaženo různými technologiemi. Namátkou lze zmínit Peltierův článek.



Obrázek 2.1: Princip tepelného čerpadla [19]

2.2 Rosný bod

Rosný bod je teplota, při které je vzduch maximálně nasycen vodními parami. Při této teplotě je relativní vlhkost vzduchu 100 %. Při poklesu teploty pod tento bod dojde ke kondenzaci vodních par, neboli vysrážení vody. Tento jev můžeme pozorovat na plechovce piva vytaženého z lednice, nebo na zrcadle v koupelně po koupeli. Pokud ovšem ochladíme samotný vzduch pod rosný bod, nemusí ke kondenzaci dojít a to z důvodu, že není přítomen dostatek kondenzačních jader. Kondenzační jádra jsou aerosolové částice přítomné ve vzduchu. V podstatě jde o jakékoli částice přítomné ve vzduchu například prach, kouř. Bez těchto částí dojde ke kondenzaci, až se vzduch přesytí [32].

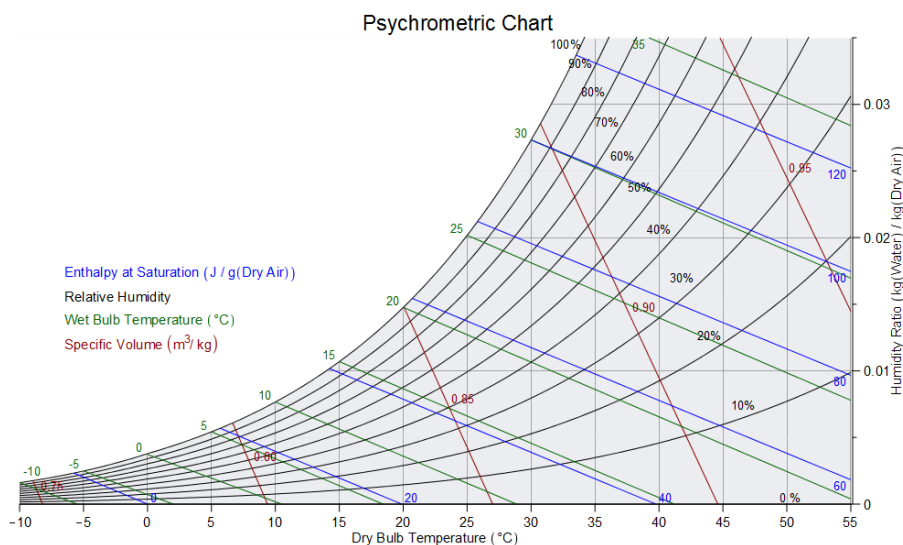
Vlhkost vzduchu udává kolik množství vody obsahuje dané množství vzduchu. Pro vyjádření vlhkosti vzduchu existuje několik možností. Nám bude stačit Absolutní vlhkost a Relativní vlhkost. Absolutní vlhkost vyjadřuje hmotnost vodní páry obsažené v objemu vzduchu. Nejčastěji se vyjadřuje v gramech vodní páry na metr krychlový vzduchu. Výpočet absolutní vlhkosti lze provést vzorcem 2.1.

$$\Phi = \frac{m}{V} [g \cdot m^{-3}] \quad (2.1)$$

Relativní vlhkost vzduchu vyjadřuje poměr mezi množstvím vodních par aktuálně obsažených ve vzduchu proti vzduchu o stejném tlaku a teplotě při plném nasycení. Udává se v procentech. Φ_n označuje absolutní vlhkost vzduchu nasyceného vodními párami. Hodnotu lze vypočítat vzorcem 2.2. Hodnotu lze najít v tabulkách pod pojmem hustota sytých par.

$$\phi = \frac{\Phi}{\Phi_n} [\%] \quad (2.2)$$

Hustota syté páry tedy udává množství vodní páry obsažené ve vzduchu při plném nasycení. Množství vodní páry, které může vzduch pojmout je závislé na teplotě vzduchu. Čím více je vzduch teplejší tím více může pojmout vodní páry. Vzhledem k tomu, že maximální množství vodní páry obsažené ve vzduchu se mění v závislosti na teplotě, mění se tedy i relativní vlhkost v závislosti na teplotě aniž by došlo ke změně množství vodní páry obsažené ve vzduchu.



Obrázek 2.2: Psychrometrický diagram [24]

Psychrometrický diagram je na obrázku 2.2, jedná se o otočený Moliérův diagram. Na ose X je teplota změřená suchým teploměrem, veličina je označena jako Dry Bulb Temperature. Na ose Y je poměr váhy vodních par ku váze suchého vzduchu, což je absolutní vlhkost vzduchu vyjádřená v jiných jednotkách. Relative humidity označuje relativní vlhkost. Wet Bulb Temperature je teplota naměřená teploměrem omotaný navlhčenou tkaninou. Pokud je relativní vlhkost menší než 100 %, tak dochází k vypařování vody z tkaniny. Vypařování vody zároveň odebírá teplo z teploměru. Teplota navlhčeného teploměru zůstává stejná, pokud stoupá teplota suchého teploměru a zároveň se nezvyšuje množství vodní páry ve vzduchu. Specific Volume vyjadřuje poměr objemu vzduchu k váze vzduchu. Enthalpy at Saturation je multiplikativní konstanta. Slouží k výpočtu kolik tepelné energie je třeba dodat, aby došlo k přeměně vody na vodní páru.

Pro potřeby chlazení je nutné zjistit správnou hodnotu rosného bodu. Při překročení teploty pod rosný bod by došlo ke kondenzaci vody na všech částech chladicího systému včetně stěn, což je nežádoucí. Rosný bod je možné vypočítat pomocí jednoduchého vzorce 2.3, který však není příliš přesný. T je teplota vzduchu v °C, V je relativní vlhkost v %.

$$T_{rosnybod} = T - \frac{100 - V}{5} \quad (2.3)$$

Existuje však i přesnější vzorec 2.4. Vstup je teplota vzduchu T °C, V je relativní vlhkost v %. Vzorec obsahuje empiricky stanovené konstanty.

$$T_{rosnybod} = \frac{243,5 \cdot \ln\left(\frac{V}{100} \cdot e^{\frac{17,67 \cdot T}{243,5+T}}\right)}{17,67 - \ln\left(\frac{V}{100} \cdot e^{\frac{17,67 \cdot T}{243,5+T}}\right)} \quad (2.4)$$

Jako další možnost se jeví vzít teplotu rosného bodu z meteorologické stanice poblíž a předpokládat, že vevnitř je vzduch s podobným množstvím vodních par jako venku. Možný problém by mohla představovat kuchyně, kde vznikají vodní páry při vaření. Nejbližší meteorologická stanice je Brno-Tuřany, je vzdálená 6 kilometrů vzdušnou čarou. Meteorologická stanice poskytuje informace o rosném bodu každou hodinu.

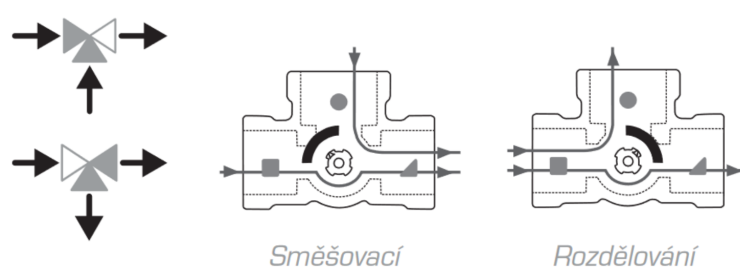
Podle informací od jednoho topenáře se optimální teplota chlazení vypočítá následovně. Teplotu rosného bodu zaokrouhlíme nahoru na celé stupně a přičteme jeden stupeň. Nastavení takto vypočtené teploty chlazení by mělo zajistit aby na soustavě nekondenzovala voda [13].

2.3 Termohlavice – ventily – čerpadla

Oběhové čerpadlo slouží k čerpání vody v topném okruhu. Jedná se o okruh s nuceným oběhem. Existují však i okruhy bez nuceného oběhu. Okruh bez nuceného oběhu nebo-li okruh s přirozeným oběhem topné vody funguje na principu rozdílné hustoty topné vody. Teplejší topná voda má menší hustotu, chladnější naopak vyšší hustotu. V kotli pak nastává situace, kdy teplá (řidší) voda stoupá nahoru. Chladnější (hustější) voda přivedená vratnou trubkou tak vytváří větší hydrostatický tlak. Přetlak vody způsobí pohyb vody v topném okruhu. V takovém případě však musí být zdroj tepla umístěn níž, než instalovaná otopná tělesa. Nevýhoda přirozeného oběhu je, že voda musí mít dostatečný rozdíl teplot, aby docházelo k oběhu. Dále je pak třeba použít trubky větších průměrů, aby byl zajištěn dostatečný průtok vody pro přenos tepelné energie. Výhoda přirozeného oběhu je funkčnost bez pomoci čerpadla a elektrické energie použité pro pohon čerpadla.

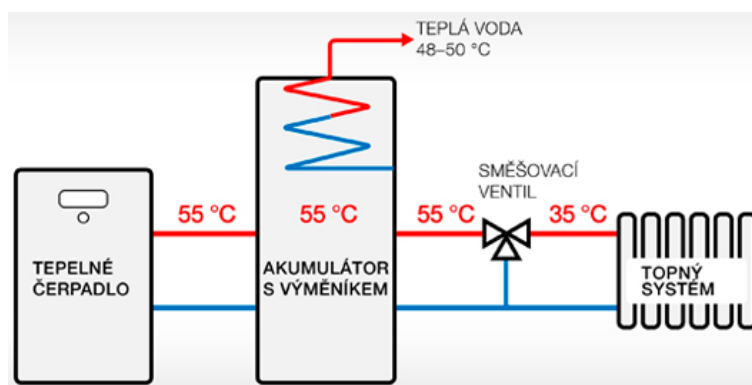
Soustava s oběhovým čerpadlem nebo-li s nuceným oběhem odstraňuje problém s umístěním tepelného zdroje. Dále je pak možné použít trubky menších průměrů, protože průtok je zajištěn činností čerpadla. Uzavřená soustava s nuceným oběhem dnes patří k nejrozšířenějším [30].

Pro regulaci teploty mezi kotlem a otopnými tělesy se často používá trojcestný ventil. Trojcestný ventil je ventil, do kterého jsou přivedeny tři potrubí. V případě, kdy dvě potrubí slouží pro přívod kapaliny a jedno pro odvod kapaliny, jedná se o směšovací ventil. V případě, kdy jedno potrubí slouží pro přívod kapaliny a zbylé dvě pro odvod kapaliny, jedná se o rozdělovací ventil. Ventil můžeme provozovat jak v krajních polohách, kdy teče kapalina pouze první nebo druhou trubkou, tím pádem se jedná pouze o přepínač mezi trubkami. Zajímavější a praktičtější je používání ventilu kdekoli na celé pracovní škále mezi krajními polohami. To nám umožňuje mísit nebo rozdělovat kapaliny v poměru nastaveném na trojcestném ventilu 2.3 [10].



Obrázek 2.3: Příklady trojcestných ventilů [10]

Teoreticky mezi rozdělovacím a směšovacím ventilem není vidět žádný rozdíl, s výjimkou přívodu a odvodu kapaliny. V praxi však může být rozdíl ve vnitřním tvarování ventilu tak, aby byl zajištěn hladký průchod kapaliny ve všech polohách ventilu a potlačeny nežádoucí efekty, například bublání či hučení při průtoku. Trojcestné ventily se běžně používají pro mísení teplé vody z primárního okruhu topné soustavy do sekundárního okruhu, například jako na obrázku 2.4.



Obrázek 2.4: Příklad zapojení trojcestného ventilu [6]

Pro ovládání jednotlivých otopných těles se používají takzvané termohlavice. Termohlavice slouží k ovládání ventilu. Termohlavice má standardní rozměr $M30 \times 1,5$. Termohlavice je přišroubovaná na ventil. Každý ventil pouští vodu do jednoho otopného tělesa. Ventil je ovládán pomocí osy ve středu ventilu. Osa ventilu je ve výchozím stavu tlačena pružinou

ven, ventil je otevřen. Při zatlačení osy dojde k zavření ventilu. V případě novějších nástěnných otopných těles nebo-li radiátorů bývá ventil umístěn přímo v topném tělese a je jeho součástí. Na starších radiátorech je ventil našroubován mezi radiátorem a přívodní trubkou. V případě podlahového otopného tělesa je ventil umístěn v přípojném místě. Ventil má za úkol omezit nebo úplně zastavit průtok topné vody a tím ovlivnit výkon otopného tělesa.

Termohlavice tedy reguluje výkon otopného tělesa. Termohlavice je několik typů podle druhu regulace. Základ lze rozdělit na plynulou regulaci, od úplně zavřeno až po úplně otevřeno, pak na jednodušší typ, který lze plně otevřít nebo plně zavřít. Nejjednodušší typ je manuální. Ručně si nastavíte výkon topení. Manuální nemá žádnou automatickou zpětnou vazbu, kromě vás, tudíž tento způsob není vhodný. V dnešní době, pravděpodobně nejpoužívanější je typ s termostatem. Kolečkem lze nastavit požadovanou teplotu a termohlavice zajistí otevírání a uzavírání ventilu a tím i výkon topení. Jedná se o plně pasivní systém, pravděpodobně tvořený pokročilým typem bimetalického teploměru. V dnešním trendu internetu věcí (IoT) vznikla také skupina elektronicky řízených a programovatelných termohlavic. Programovatelné mají na sobě umístěno ovládání případně i displej. Lze naprogramovat, různou teplotu pro různé časové období, například teplotu přes den a jinou přes noc. Nejnovější termohlavice obsahují bezdrátové technologie například Wifi, Bluetooth, Zigbee, které umožňují ovládání pomocí mobilního telefonu. Některé také umožňují připojení na domácí inteligenci Google Home, Alexa Home Connect a další asistenty, které se objevují na trhu. Domácí asistenti jsou v dnešní době na vzestupu. Umožňují hlasové ovládání a správu zařízení v moderních grafických uživatelských rozhraních. Tyto ventily jsou přímo určeny pro automatizaci. Lze je samozřejmě ovládat centrálně i vzdáleně.

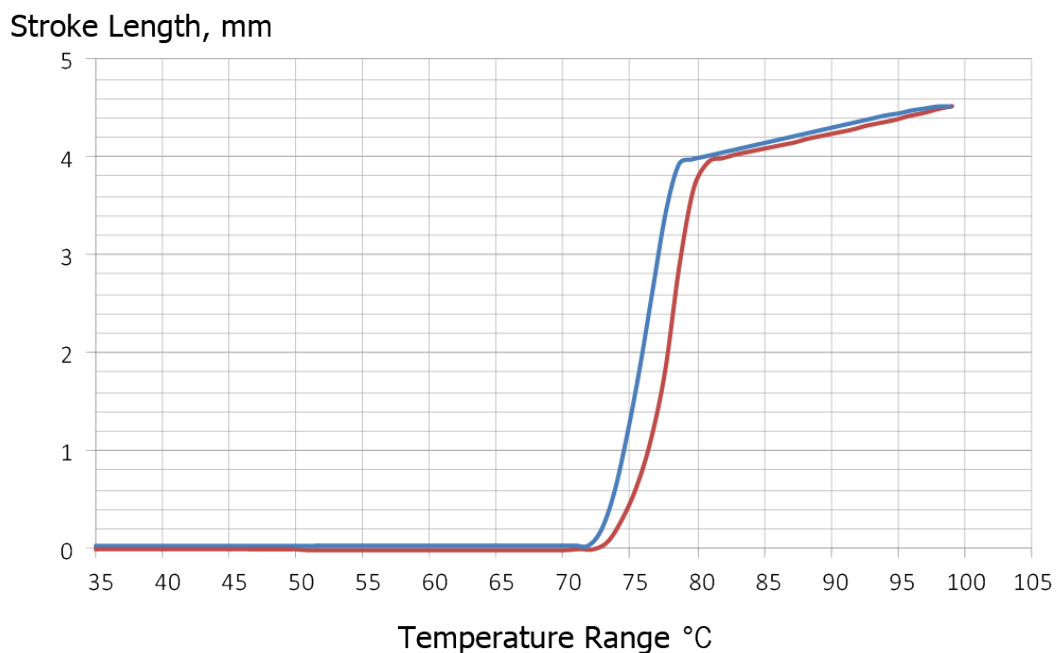
Pak existuje skupina termohlavic určených pouze pro centrální ovládání. Umožňují pouze dvě polohy, úplně otevřeno a úplně zavřeno. Pohon je tvořen voskovým motorem [23]. Voskový motor je tvořen pístem s malým prostorem pro vosk. Ve spodní části se nachází termistor typu PTC (positive temperature coefficient). Termistor je rezistor, jehož hodnota je závislá na teplotě. U termistorů s pozitivním koeficientem (PTC) při zahřívání roste odpor, naopak u termistorů s negativním koeficientem (NTC) při zahřívání odpor klesá.

Princip činnosti je následující. Na termistor je typu PTC je přivedeno elektrické napětí. Termistor se začne zahřívát a zároveň ohřívá vosk pod pístem. Vosk s postupným ohřevem zvětšuje svůj objem a vytlačuje píst směrem ven. Píst přemůže zpětnou pružinu a zvedne ovládací mechanismus ventilu. Po odpojení elektrického napětí vosk postupně chladne a zmenšuje svůj objem. Zpětná pružina zatlačí píst i ovládací mechanismus ventilu zpět do výchozí polohy. Maximální teplota termistoru je omezena jeho vlastností změny odporu. Čím větší je teplota, tím větší je odpor, tím pádem podle Ohmova zákona protéká menší proud a tím klesá tvorba tepla a termistor chladne. Při chladnutí dojde k opačnému chodu. Odpor termistoru klesne, zvětší se protékající proud a tím i tvorba tepla.

Teplota vosku, pístu, termistoru, bývá v zapnutém stavu okolo 85 až 90 °C [40]. Spotřeba termistoru bývá v zapnutém stavu kolem dvou wattů. Při zapínání má však termohlavice teplotu okolí, tudíž odpor termistoru je malý. To způsobuje násobně větší odběr elektrické energie při zapnutí, což zapínání více termohlavic v jeden okamžik způsobuje proudovou špičku, která může způsobit shození jističe.

Pracovní napětí závisí na hodnotě odporu termistoru. Vyrábí se na napětí 24 V, 110 V, 230 V. Dále je třeba určit zda má být termohlavice ve výchozím stavu otevřená (NO – normally open) nebo zavřená (NC – normally closed), vyrábí se obě varianty.

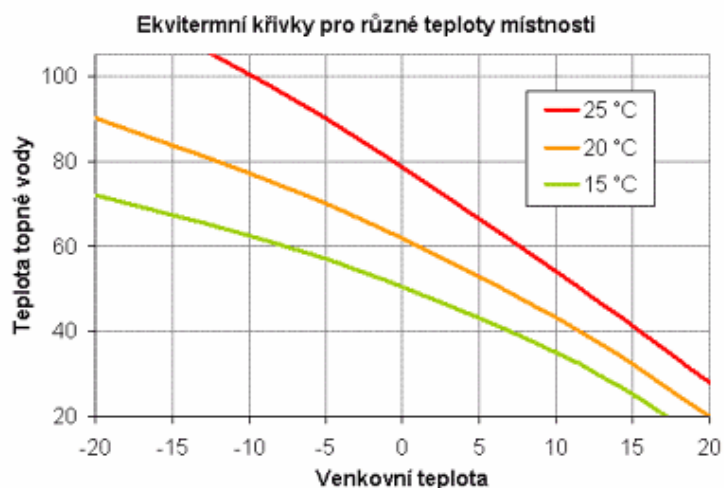
Na obrázku 2.5 je graf vysunutí pístu v závislosti na teplotě vosku. Červená čára zobrazuje ohřev a modrá chladnutí vosku.



Obrázek 2.5: Průběh vysunutí pístu v závislosti na teplotě [40]

2.4 Ekvitermní regulace

Ekvitermní regulace spočívá v regulaci topné vody v závislosti na venkovní teplotě. Při nižší venkovní teplotě je nastavena vyšší teplota topné vody a naopak při vyšší venkovní teplotě nastaví nižší teplotu topné vody. V podstatě jde o dorovnání tepelné energie dodané topením, proti odchozí tepelné energii do okolního prostoru [14] Příklad ekvitermní funkce je na obrázku 2.6.



Obrázek 2.6: Příklad ekvitermní regulace [14]

2.5 Jističe a stykače

Elektrický jistič slouží k rozpojení elektrického okruhu v případě přetížení. Přetížení označuje stav, kdy elektrickým obvodem teče větší proud, než na který byl dimenzován. Při průtoku proudu dochází k ohřevu vodiče, obzvláště při přetížení dochází k nadměrnému ohřevu vodiče, což může způsobit požár. Jistič na rozdíl od pojistky obsahuje mechanismy, které je možné po aktivaci vrátit do původního stavu a tím obnovit původní funkci jističe. V jističích se používá kombinace tepelné a elektromagnetické spouště.

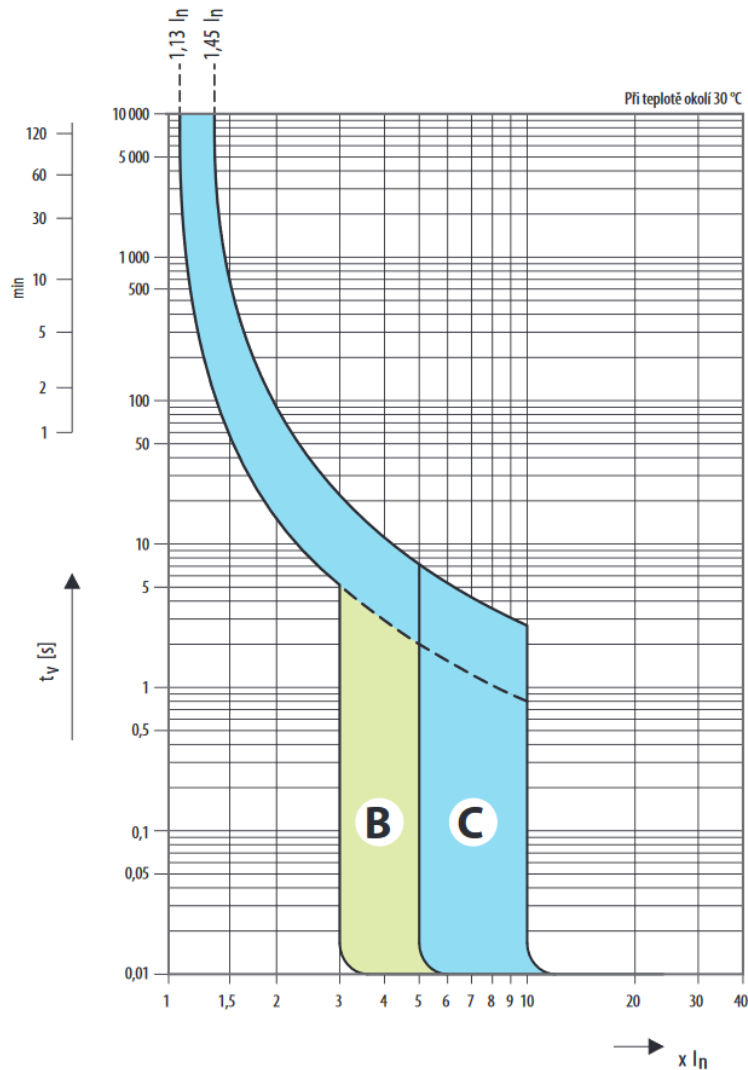
Tepelnou spoušť tvoří bimetal. Jedná se o dva rozdílné kovy spojené dohromady. Rozdíl musí v teplotní roztažnosti kovů. Bimetal připojen sériově do elektrického obvodu. Při průchodu elektrického proudu dochází k zahřívání obou kovů. Vlivem jejich spojení a rozdílné roztažnosti dojde k ohybu celku. Tento pohyb aktivuje rozpojovací mechanismus. Tento princip je využit u spoušty dalších zařízení, většinou jako jednoduchý regulátor teploty. Tato spoušť však není dostatečně rychlá pro nárazové přetížení. Naopak je dostatečně přesná na nastavení jmenovitého proudu.

Elektromagnetická spoušť je tvořena permanentním magnetem a cívkou. Cívka je připojena sériově do elektrického obvodu. Průchodem elektrického proudu vzniká v cívce magnetické pole. Magnetické pole způsobí pohyb magnetu, který aktivuje rozpojovací mechanismus. Tato spoušť podstatně rychlejší při nárazovém proudu, protože magnetické pole vznikne podstatně rychleji než ohřátí bimetalu. Naopak krátké odběrové špičky mohou způsobit její aktivaci. Z toho důvodu je typicky elektromagnetická spoušť nastavena na o něco větší jmenovitý proud, než tepelná spoušť.

Podstatný parametr jističe je jmenovitý proud. Je to maximální proud, který může trvale procházet jističem, aniž by jistič rozpojil elektrický obvod. Dále je pak důležitá vypínací charakteristika jističe. Vypínací charakteristika většinou určuje způsob použití jističe. Běžně používané jsou charakteristiky B, C, D. Charakteristika B slouží pro jištění elektrického vedení. Jistič rychle vypíná malé přetížení, typicky 3 až 5-ti násobek jmenovitého proudu v čase do 0,1 s. Používá se pro jištění světelných a zásuvkových okruhů jak v doměch, tak ve firmách, což z něj dělá nejpoužívanější jistič, který najdete opravdu všude. Jistič s charakteristikou C je označován jako motorový jistič. Jistič typicky vypíná 5 až 10-ti násobek jmenovitého proudu v čase do 0,1 s. Jak název napovídá používá se pro jištění větších motorů, které při rozběhu způsobují proudovou špičku. Dnes se hodí také na jištění výpočetní techniky. Dnešní elektronika v drtivé většině používá spínané napájecí zdroje. Spínané zdroje mají v primárním obvodu velký vyrovnávací kondenzátor, který se při zapnutí musí nabít. V kvalitněji provedených zdrojích bývá omezovač tohoto nabíjecího proudu, avšak pokud je zdrojů větší množství, tak ani tento omezovač nestačí. Jistič s charakteristikou B se nepodaří zapnout napoprvé [42] [25].

Na obrázku 2.7 jsou znázorněny vypínací charakteristiky B a C. Na ose X je vyneseno jmenovitý proud I_n vyjádřený v násobcích I_n . Na ose Y je vyneseno vypínací čas T_v . Na grafu lze vidět zvyšující se rychlost tepelné spouště od $1,13I_n$, až po spuštění elektromagnetické spouště při $3I_n$ v charakteristice B nebo $5I_n$ v charakteristice C.

Stykače slouží pro spínání a rozpojování elektrického obvodu. Oproti relé jsou navrženy na výrazně vyšší proudy. Základním principem je elektromagnet, který je ovládán přivedením elektrické energie na cívku. Elektromagnet následně ovládá spínání a rozepínání silových kontaktů. Stykače mohou být dvou druhů. Běžně bývá stykač rozepnut a až po přivedení elektrické energie na cívku dojde k sepnutí kontaktů. Po odpojení elektrické energie dojde k rozepnutí kontaktů. Opačný druh má silové kontakty sepnuty a až po přivedení



Obrázek 2.7: Vypínací charakteristiky jističů [25]

elektrické energie na cívku dojde k rozepnutí kontaktů. Po odpojení elektrické energie dojde opět dojde k sepnutí kontaktů [26].

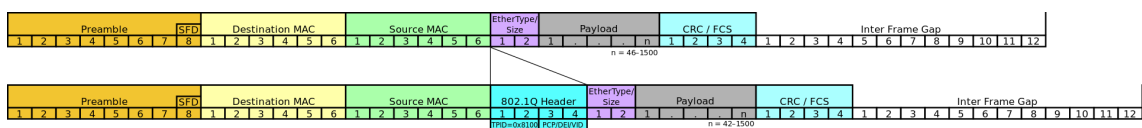
2.6 Ethernet - oddělení sítě

Ethernet je skupina drátových technologií používaných pro lokální počítačovou síť. Jedná se o skupinu standardů IEEE 802.3. Jedná se o přepínanou síť, to znamená, že data se přenáší v jednotlivých na sobě nezávislých rámcích. Jednotlivé body sítě jsou adresovány pomocí takzvaných MAC (media access control) adres. Ty jsou zapsány do síťového zařízení již při výrobě. Adresa je dlouhá 48 bitů, z toho prvních 24 bitů je identifikátor výrobce OUI (organizationally unique identifier), druhých 24 bitů je unikátních pro každé zařízení. Ethernet obsahuje i vyhrazené MAC adresy pro zaslání rámce všem (broadcast) a nebo skupině zařízení (multicast). Zaslání rámce jednomu konkrétnímu zařízení se nazývá unicast.

Pro fyzické propojení se drtivě většině případů používá kabel se čtyřmi páry kroucené dvojlinky kategorie minimálně 5e. Síť je propojená pomocí přepínačů (switch). Do přepínače jsou připojené jednotlivé koncové zařízení. Síť tak tvoří tvar hvězdy. V případě potřeby více zařízení lze propojit více přepínačů mezi sebou, čímž vznikne propojení několika hvězd. Při zapojování je potřeba dát pozor, aby nevznikl kruh. Ten způsobí broadcastovou bouři neboli cyklení broadcast rámců. To následně zahltní linku, čímž znemožní doručování ostatních rámců [33].

Ethernet obsahuje i spoustu užitečných rozšíření například 802.1Q, 802.1s, 802.3af. Například 802.1s obsahuje MSTP (multiple spanning tree protocol). Ten umožňuje detekovat a rozpojit výše zmíněný kruh a tím zamezit broadcastové bouři.

802.1Q umožňuje vytvářet oddělené sítě takzvané VLAN (virtual local area network). Například potřebujeme mít dvě a více od sebe oddělených sítí. Ty můžeme vybudovat buď pomocí fyzicky separátních přepínačů. Lepší varianta je na přepínači nastavit VLANy a přiřadit jednotlivé porty do jednotlivých VLAN. To nám umožňuje lepší nasazení v případě více sítí, jednodušší změny pouhou změnou konfigurace a menší počet potřebných zařízení. Největší výhodou této technologie je přenos více VLAN mezi jednotlivými přepínači. Toho je dosaženo pomocí vložení značky (tag) do ethernetového rámce, která identifikuje, do které VLANy daný rámec patří. Mezi přepínači pak probíhá takzvaný značkový provoz, v Cisco terminologii je označen názvem trunk. VLANy je potřeba nakonfigurovat do všech zúčastněných přepínačů stejně. Porty na přepínači pro značkový provoz je potřeba nakonfigurovat. Porty pro koncová zařízení neposílají značky VLAN. Jednak je koncová zařízení nemusí rozpoznat a za druhé by koncové zařízení mohlo ovlivnit do které VLAN bude rámec zaslán, což by byl bezpečnostní problém. Počet VLAN je omezen na 4094. Tato technologie je převážně využita v budovách ve kterých sídlí více nezávislých firem, které mohou tímto způsobem sdílet infrastrukturu. Ethernetový rámec se značkou i bez je na obrázku 2.8. Ethernetový rámec začíná synchronizační sekvencí označovanou preamble. Jedná se o periodické střídání jedničky a nuly pro vytvoření obdélníkového signálu. Tento signál slouží k synchronizaci hodin přijímací strany pro správné vzorkování následujících dat. Poslední bajt označovaný SFD (start frame delimiter) má poslední bit stejný jako předchozí, tím signalizuje začátek přenášených dat. Pak následuje cílová MAC adresa, zdrojová MAC adresa, EtherType označuje typ dat, který následuje, payload jsou nesená data a CRC je kontrolní součet, který slouží pro detekci možné chyby při přenosu. Všimněte si, že VLAN tag zabírá celou pozici EtherType, přepínač pomocí tohoto pole detekuje, zda se jedná o 802.1Q rámec [9].



Obrázek 2.8: Ethernetový rámec s a bez VLAN značky [34]

802.3af nebo-li PoE (power over ethernet) umožňuje napájet koncová zařízení přes ethernet kabel. To je praktické pro připojení IP kamer a telefonů. Jednak se k nim nemusí tahat napájení zvlášť a při výpadku napájení stačí mít zálohované napájení přepínače a prvky na něj připojené jsou tím pádem taky zálohované. Ve výchozím stavu port pouze detekuje zda koncové zařízení požaduje napájení. Teprve po požadavku je napájení zapnuto, aby nedošlo k poškození portu na koncovém zařízení. Napájení probíhá stejnosměrným proudem kolem 48 V po datových vodičích současně při přenosu dat. Střídavá a stejnosměrná složka

je v zařízení separována. 802.3af umožňuje přenést 12 W, 802.3at umožňuje přenést 25 W, 802.3bt 51 W. Dále pak existuje nestandardizovaná varianta označovaná pasivní PoE. Pasivní PoE využívá staršího ethernet standardu 100BASE-TX, které pro přenos dat používá pouze dva páry. Zbylé dva páry jsou přímo využity pro napájení. Tento způsob je často využíván u levnějších zařízení, nemusí se separovat stejnosměrná složka od dat a nemusí proběhnout domluva na napájení. Tu variantu lze také provozovat pomocí pasivních kabelových redukci. Výhoda spočívá v tom, že napájení je plně odděleno od dat a lze si jej přizpůsobit podle potřeb. Nevýhoda je připojení pasivního PoE do normálního portu, který PoE neočekává. Typicky dojde k trvalému poškození portu.

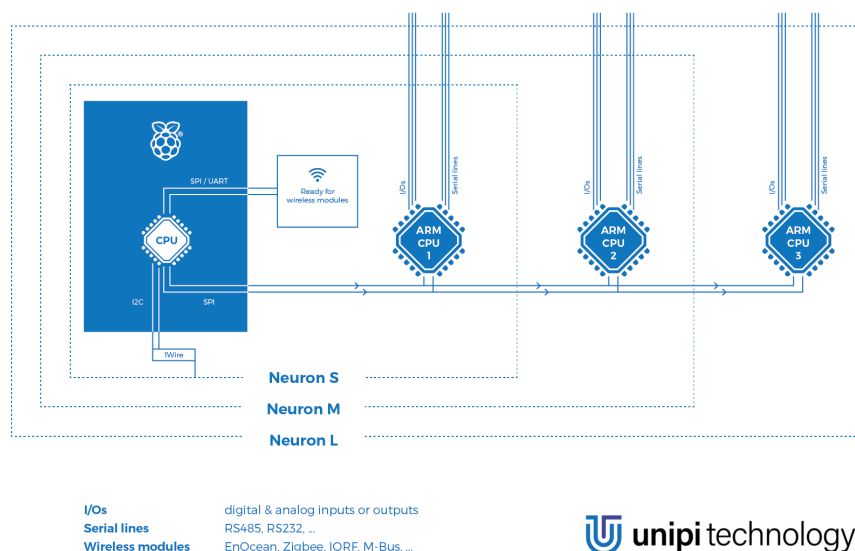
Ethernet obsahuje i možnosti, které v dnešní době již nemá smysl používat a zabývat se jimi. K nim patří použití hloupého propojení přes HUB. HUB propojí zařízení pouze fyzicky, což vyžaduje koordinovat přístup ke sdílenému mediu pomocí CSMA/CD (carrier sense multiple access / collision detection). Na sdíleném mediu může vždy vysílat jen jedno zařízení, to způsobuje, že je možné buď vysílat nebo přijímat a tedy poloviční duplex. Při připojení do přepínače je na lince vždy jedno zařízení, tím vznikne pouze point-to-point linka, která má plný duplex, může tedy přijímat i vysílat zároveň. Dalším archaismem jsou křížené kabely. Ty zajišťují správné propojení vysílače k přijímači a naopak. Prakticky všechny dnešní zařízení obsahují technologii Auto MDI-X, která zajistí automatické spárování vysílací a přijímací strany. Stačí tedy používat pouze přímé kabely.

2.7 UniPi PLC a Mervis

Jednotky UniPi jsou ve společnosti SPOLEČNOST-24 s.r.o. používány již delší dobu. Mají s nimi již zkušenosti a taky jich mají dost k dispozici. Jednotky navrhuje a vyrábí brněnská firma Faster CZ spol. s r.o.. V jejich nabídce jsou v současnosti tři hlavní produktové řady. UniPi 1.1 je jejich první PLC jednotka. Je postavená na RaspberryPi a k němu je připojena rozšiřující deska se vstupními a výstupními porty. Dalším produktem je UniPi Neuron. Jedná se o vylepšený model předchozího 1.1. Vylepšení lze vidět na první pohled, modrá krabička, konektory na všech portech, ty umožňují snadnou manipulaci s jednotkou, držák na DIN lištu. Dále pak příprava pro průmyslové použití, například standardní napájení 24 V.

Uvnitř UniPi Neuron se nachází standardní jednodeskový počítač RaspberryPi 3B. V současnosti je možné nainstalovat dovnitř i RaspberryPi 4. K počítači je připojena deska s se vstupními a výstupními porty. Obecnou architekturu popisuje obrázek 2.9.

Pro UniPi Neuron lze stáhnout před chystaný obraz, který se nahraje na SD kartu. Obraz je založen na distribuci Debian. Současně je v něm nachystán i ovladač periferií. Pro ovládání lze využít jedno z před chystaných API. Na nejzákladnější úrovni je API SysFS. Jedná se o standardní linuxové využití souborového systému. Pro každý port je vytvořen soubor. Například `/sys/devices/platform/unipi_plc/io_group1/do_1_01/do_value` je port z první skupiny, digitální výstup 1 (DO – digital output). Při zapsání znaku 1 do souboru dojde k zapnutí výstupu, při zapsání 0 dojde k vypnutí výstupu. Vstup se obsluhuje čtením ze souboru. Dále je k dispozici protokol ModbusTCP. Jak název napovídá, protokol využívá TCP spojení a je jej možné provozovat vzdáleně. Tento protokol využívá Mervis. Pak je k dispozici API Evok, které umožňuje použití websokets. Na něj lze napojit Node-RED, kde jsou před definované moduly pro ovládání portů. Z Node-RED pak lze hodnoty poslat po MQTT. Každá jednotlivá skupina portů je obsluhována drobným mikrokontrolérem s ARM jádrem. Jejich firmware lze zkontrolovat i aktualizovat programem `fwspi`.



Obrázek 2.9: Vnitřní struktura UniPi Neuron [38]

S jednotkami UniPi je dodáván systém Mervis. Mervis je software od české firmy Energo centrum plus. Je to kompletní softwarová platforma pro profesionální řešení vhodné pro širokou škálu nasazení, od řízení chytré domácnosti až po průmyslovou automatizaci. Platforma zahrnuje následující produkty [29].

- Mervis IDE – software pro konfiguraci, programování, ladění a návrh vizualizace
- Mervis RT – software běžící přímo na PLC, který provádí program vytvořený v Mervis IDE
- Mervis DB – databáze pro ukládání dat zaznamenaných v rámci automatizace
- Mervis Proxy – služba pro zabezpečený vzdálený přístup k PLC přes internet
- Mervis SCADA – dohledový a vizualizační nástroj pro dispečink celé technologie

Mervis splňuje mezinárodní standard IEC 61131-3, který definuje základní programovací režimy pro PLC. Standard je známější pod názvem CoDeSys (Controlled Development System), což je univerzální vývojové prostředí pro aplikační programy řídicích systémů PLC. K dispozici jsou následující programovací metody [39].

- IL – Instruction List (posloupnost instrukcí)
- ST – Structured Text (vyšší programovací jazyk - obdoba Pascalu)
- LD – Ladder Diagram (kontaktní plán / liniové či reléové schéma)
- SFC – Sequential Function Chart (vývojové schéma)
- FBD – Function Block Diagram (schéma funkčních bloků)

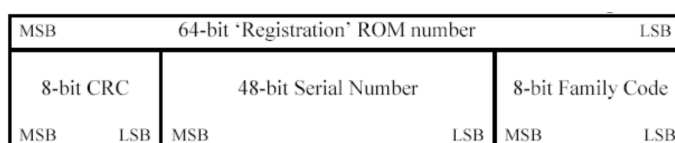
- CFC – Continuous Function Chart (volně propojované bloky)

V Mervisu je tedy možné vystavět celý systém. Mervis, na rozdíl od Node-RED, je uzavřená a placená technologie. Licencí pro provoz je několik. Mervis IDE a RT se váže na konkrétní jedno PLC zařízení. K PLC UniPi Neuron je licence dodávána automaticky. Mervis DB, Proxy a SCADA jsou cloudové aplikace, tudíž se za ně platí měsíční předplatné. Výše předplatného se odvíjí od Počtu datových bodů. Datový bod proměnná ukládána do Mervis DB nebo použita v projektu Mervis SCADA. Do 25 bodů je použití zdarma. Předpokládám však, že počet bodů bude minimálně 500, což vychází podle aktuálního ceníku 625 korun bez DPH za měsíc [37].

2.8 1-Wire sběrnice – zařízení a řídicí software

1-Wire je sběrnice navržena firmou Dallas Semiconductor. 1-Wire je relativně pomalá sběrnice, maximálně dosahuje rychlosti 16,3 kbit/s. Název napovídá, že sběrnice využívá pouze jeden drát, ve skutečnosti musí být vodiče minimálně dva. Druhý vodič je takzvaná společná zem, proti které se vztahuje napětí na signálovém drátu 1-Wire. Přenos po 1-wire je řízen pouze jedním masterem. Ostatní zařízení (slave) jsou dotazovány masterem. Přenos informací probíhá tedy polovičním duplexem, to znamená buď od masteru k jednomu vybranému slavu nebo slave odpovídá na dotaz mastera. Technicky je sběrnice tvořena tak zvaným otevřeným kolektorem a pull-up rezistorem. Pull-up rezistor je připojen na napájecí napětí a zajišťuje tím klidovou úroveň sběrnice. Zařízení na sběrnici komunikují připojováním a odpojováním sběrnice na společnou zem. Jelikož sběrnice je relativně pomalá a klidová úroveň sběrnice je napájecí napětí, tak je možné odebírat trochu proudu pro vlastní potřeby zařízení. Napájecí napětí může být 1,5 V, 3,3 V i 5 V. Běžně se používá 5 V. Všechny zařízení na sběrnici musí mít stejnou úroveň napájecího napětí [21].

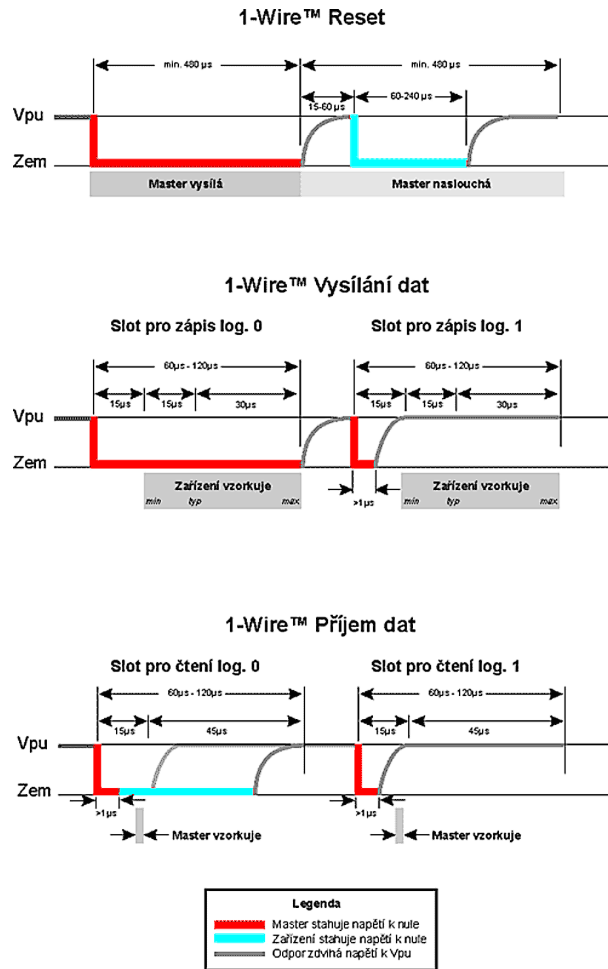
Zařízení na sběrnici jsou odlišeny unikátním identifikátorem, nazvaným též Registration ROM. Identifikátor vypálen do zařízení při výrobě. Identifikátor má 64 bitů a je vyobrazen na obrázku 2.10. 48 bitů Serial Number je unikátní pro každé zařízení. 8 bitů Family Code označuje třídu zařízení, určuje o jaký druh zařízení se jedná například teploměr, tlačítko [27].



Obrázek 2.10: Identifikátor 1-Wire slave zařízení [8]

Přenos po sběrnici probíhá v timeslotech nebo jinak řečeno časových oknech. Časový slot je dlouhý minimálně 60 μ s a maximálně 120 μ s. Druhů časových slotů je celkově pět. Vysílání nuly, vysílání jedničky, příjem nuly, příjem jedničky a reset sběrnice. Pro lepší představu jsou timesloty na obrázku 2.11.

Reset sběrnice uvede všechny slave zařízení do výchozího stavu. Pak master vysílá příkaz nazvaný command code. Příkaz je dlouhý dva bajty. V tabulce 2.1 je uveden příklad podstatných příkazů na 1-Wire sběrnici. Pokud je na sběrnici pouze jedno zařízení lze využít příkazy Skip ROM a Read ROM. Skip ROM přeskočí vysílání identifikátoru zařízení, tím je možné ušetřit čas při komunikaci. Read ROM umožňuje rovnou přečíst identifikátor, aniž by muselo proběhnout prohledávání. Po odvysílání příkazu Match ROM proběhne



Obrázek 2.11: 1-Wire timesloty [21]

vysílání identifikátoru, čímž se vybere konkrétní zařízení, se kterým bude dále probíhat komunikace. Pro vybrání jiného zařízení musí master opět resetovat sběrnici. Pouhá detekce připojeného zařízení na sběrnici je obsažena již v resetovacím timeslotu. Abychom určili, zda se jedná o jedno nebo více zařízení je potřeba prohledat sběrnici. Příkaz Search ROM zahájí prohledávací algoritmus. Algoritmus umožňuje zjistit identifikátory všech zařízení připojených na sběrnici. Algoritmus je založen na principu prohledávání binárního stromu [2] [15].

1-Wire příkaz	Hodnota	Popis
Skip ROM	0xCC	Ignoruje ID zařízení
Read ROM	0x33	Přečte ID zařízení
Search ROM	0xF0	Spuštění prohledávání ID
Match ROM	0x55	Výběr zařízení pomocí ID

Tabulka 2.1: Příklad 1-Wire příkazů.

Po odvysílání příkazu Search ROM dojde u všech slave zařízení k odvysílání bitu identifikátoru. Začíná se bitem s nejnižším významem (LSB). Vzhledem ke konstrukci 1-Wire

sběrnice dojde k promíchání signálu od všech zařízení. Jakmile jedno zařízení stáhne sběrnici na nulu bude na sběrnici nula, což tvoří logickou funkci AND. V dalším timeslotu všechny zařízení odvysílají negaci tohoto bitu. Master tedy obdrží dvě hodnoty a vyhodnotí situaci podle následující tabulky 2.2.

Timeslot 1	Timeslot 2	Informace
0	0	Na bitové pozici je 0 i 1
0	1	Na bitové pozici je pouze 0
1	0	Na bitové pozici je pouze 1
1	1	Žádné zařízení

Tabulka 2.2: Vyhodnocení prohledávaného bitu [15]

Poté master odvysílá bit. Tento bit si zařízení porovnají se svým bitem v identifikátoru. Pokud se bit shoduje, zařízení se dál účastní vyhledávání. Pokud se bit neshoduje, zařízení se uspí a dál se neúčastní vyhledávání a čeká na reset sběrnice. Zařízení, které se dál účastní se posunou na další bit. Tímto krokem master rozhodne, kterou větev stromu bude prohledávat.

Postup se opakuje na dalších bitových pozicích až je zjištěn identifikátor jednoho zařízení na sběrnici. Zařízení tvoří na binárním stromě list a identifikátor je popis cesty k listu. Pokud během vyhledávání dojde na některé bitové pozici ke stavu na prvním řádku tabulky, kdy může existovat nula i jednička, tak master si tuto pozici uloží a následně zahájí nové vyhledávání, při kterém se vydá opačným směrem. Postup se opakuje do doby, než jsou objeveny všechny listy v binárním stromu a tím i všechny zařízení na sběrnici.

Další podstatnou částí je nasazení 1-Wire sběrnice do praxe. Teoreticky je možné ke sběrnici připojit až 75 zařízení. Prakticky se však doporučuje maximálně dvacet zařízení. Samozřejmě záleží na vzdálenosti a kvalitě kabeláže. Dokumentace od firmy UniPi doporučuje maximálně 13 zařízení pro bezproblémový provoz [36]. Důležitý je také způsob propojení. Ideální je samozřejmě zapojení do sběrnice. Další varianta je sběrnice a jednotlivé zařízení nejsou přímo na sběrnici, ale jsou připojeny pomocí kabelu maximálně délky dvou metrů, ideálně pro všechny zařízení stejná délka. Nejhorší varianta je zapojení do hvězdy. Při zapojení do hvězdy dochází k mnoha odrazům od konců vedení, které způsobují rušení signálu [17] [36] [8].

Na sběrnici 1-Wire lze připojit různé druhy zařízení. Nejčastěji se připojují různé druhy senzorů. Ve Společnosti-24 je využit senzor Unica 1-wire typ U1WTD [31]. Senzor je určen pro zabudování do vypínačů. Senzor se prodává ve více variantách. Všechny varianty umožňují měření teploty. Další dostupné schopnosti jsou měřič vlhkosti, měřič intenzity světla, snímání jednoho digitálního vstupu, snímání jednoho analogového vstupu od 0 V do 5 V a indikaci rozsvěcováním LED diody. Zajímavým řešením je použití čipu DS2438, který původně určen pro měření a dohled akumulátorů [16]. Teploměr je integrován přímo na čipu. Senzor Unica má doporučené napájení 5 V. Dokumentace k čipu dovoluje rozsah napájecího napětí od 2,4 V až po 10 V. Experimentálně jsem ověřil, že senzor je možné provozovat i bez přídavného napájení a tedy si vystačí s parazitním napájením ze sběrnice 1-Wire. Teplotu je možné měřit od -55°C až do $+125^{\circ}\text{C}$ v krocích po $0,03125^{\circ}\text{C}$. Hodnota je uložena na třinácti bitech. Prakticky je však rozdělena do dvou registrů o osmi bitech ve formátu dvojkového doplňku. Maximální rozsah měřených hodnot je však velmi zavádějící vzhledem k provozním teplotám čipu -40°C až $+85^{\circ}\text{C}$. Dokumentace k Unica senzoru zmiňuje dokonce menší rozsah od -20°C do $+40^{\circ}\text{C}$.

Podstatným prvkem je obsluha 1-Wire sběrnice. 1-Wire mastera je možné vytvořit z mikrokontroléru. Mikrokontroléry mají buď již vytvořené 1-Wire rozhraní zabudované přímo v hardwaru nebo předem připravenou softwarovou implementaci. Pokud ne, tak si musí uživatel vytvořit vlastní implementaci 1-Wire protokolu a vysílat ji na některém GPIO pinu. Přímochařejší je připojit 1-Wire sběrnici přímo do počítače, to je možné vícero způsoby. Nejjednodušší varianta je použití převodníku na USB, například DS2490. Rozhraní USB je v dnešní době skoro na každém počítači. Další možností je využít klasikou sériovou linku, ať už v podobě RS232, případně odlehčené varianty v podobě USB UART převodníku například CP2102. Samozřejmě nelze připojit 1-Wire na sériovou linku přímo. Musí se připojit přes obvod s pasivními součástkami, aby byly dodrženy napěťové úrovně signálu. Obvodů je více druhů podle způsobu řízení. Některé obvody využívají pro řízení signál DTR. Signál DTR je následně obsluhován softwarem tak aby tvořil odpovídající 1-Wire signál. Další způsob je využití pinu TXD a RXD. Ty se využívají pro normální sériovou komunikaci. Komunikace se nastaví na jeden stop bit a žádná parita. Na sériové lince se posílá po znacích. Znak má délku 8 bitů. Na začátku je uveden jeden start bit a na konci jeden stop bit. Sekvence osmi bitů se nazývá bajt. Sekvence osmi bitů se start a stop bitem se nazývá baud. Řídící software odesílá buď samé nuly nebo samé jedničky. Změnou rychlosti přenosu, udávanou v baudech za sekundu, lze dosáhnout předepsaných hodnot pro sběrnici 1-Wire. Tato možnost je oblíbená mezi kutily. Sériovou linku většinou znají a mají ji a k tomu si postaví vlastní obvod [35].

Další možností je využití převodníku mezi 1-Wire a I2C například DS2482. Výhodou tohoto řešení je možnost připojení k mikrokontroléru tak jednodeskovým počítačům. Nemusíme se starat o 1-Wire signál. Komunikace přes I2C je většinou již ověřená a funkční. Následně je třeba naprogramovat správné ovládání pomocí registru v převodníku. Nemusíme je již starat o generování 1-Wire signálu. Konkrétní obvod DS2482-800 obsahuje 1-Wire master portů. Převodník má nastavitelnou adresu I2C pomocí tří pinů na čipu. To umožňuje připojit až osm těchto převodníků paralelně na I2C sběrnici a vytvořit tím až 64 1-Wire master portů. Při tomto masivním nasazení je však třeba počítat s omezenou propustností sériové sběrnice I2C [18].

Na straně počítače je třeba vybrat software, který se bude starat o 1-Wire komunikaci a čtení hodnot z čidel a senzorů. Softwaru je potřeba vybrat také podle připojení sběrnice do počítače, zda se má starat o generování 1-Wire signálu a jakým způsobem, podporovat USB převodníky, I2C převodníky a podobně.

Jako schopný software pro Linux se jeví OWFS. OWFS je balíček, který se skládá ze čtyř programů. Jsou to `owserver`, `owhttpd`, `owftpd`, `owfs-fuse`. Program `owserver` je základem OWFS, tvoří takzvaný backend. Ostatní programy (`owhttpd`, `owftpd`, `owfs-fuse`) se na něj připojují. `owserver` umožňuje různé druhy připojení 1-Wire sběrnice zmíněné výše. Podrobné nastavení se nachází v manuálové stránce [1]. K připojení frontend programů k `owserver` probíhá pomocí vlastního vytvořeného protokolu. Protokol se přenáší pomocí TCP spojení a poslouchá na portu 4304.

Program `owfs-fuse` se připojí k `owserver` a vytvoří v souborovém systému virtuální soubory pro jednotlivé zařízení. Například k senzoru teploty a vlhkosti vytvoří složku s názvem identifikátoru zařízení. Ve složce vytvoří soubor s teplotou a soubor s vlhkostí. Následně stačí číst pouze číst soubor a automaticky je v něm aktuální hodnota ze senzoru.

Program `owftpd` je podobný jako výše zmíněný `owfs-fuse`. Rozdíl je v přístupu k souborům. Program `owftpd` vytvoří také virtuální soubory, ale zpřístupní je pomocí FTP protokolu. K FTP lze samozřejmě přistupovat i vzdáleně.

Program `owhttpd` je webový server a zároveň generuje webovou stránku s informacemi ze senzorů. Webová stránka je přístupna na portu 2121. Webové rozhraní je přehlednější, lze si rychle udělat přehled o všech hodnotách senzoru.

V Linuxu existuje subsystém pro obsluhu 1-Wire zařízení nazvaný `w1`. Subsystém umožňuje obsluhovat 1-Wire zařízení jako ostatní zařízení v Linuxu. Například USB převodník DS2490 po připojení je vzhledem k vlastnostem automaticky rozpoznán a přiřazen do `w1` subsystému. Subsystém `w1` vytvoří virtuální soubor, který reprezentuje 1-Wire master zařízení. Subsystém automaticky detekuje zařízení na 1-Wire sběrnici a vytvoří virtuální soubory s názvem podle identifikátoru zařízení. Jelikož subsystém `w1` je univerzální lze do něj připojit i I2C převodník. I2C subsystém však neumožňuje automatickou detekci zařízení na I2C sběrnici, proto je nutné zařízení přiřadit ručně. Subsystém `w1` je v Linuxu v podstatě nativní rozhraní pro připojování 1-Wire zařízení [7].

2.9 MQTT protocol

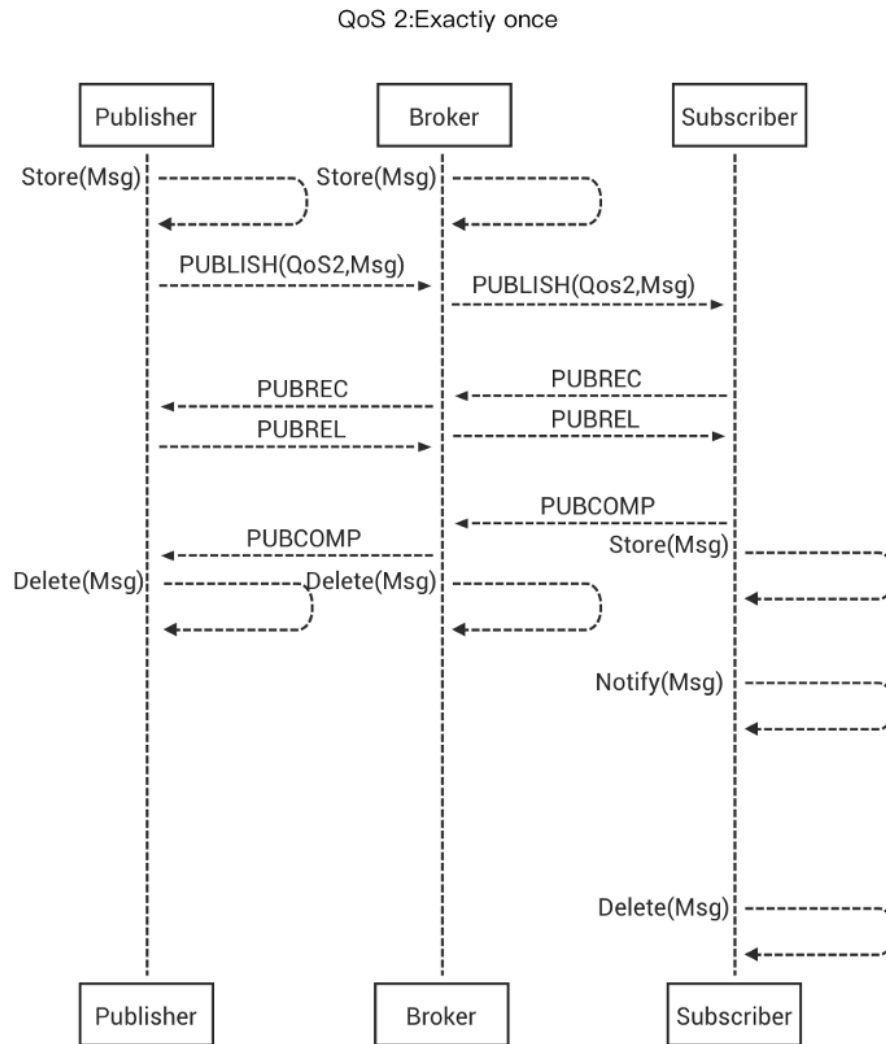
MQTT (Message Queuing Telemetry Transport) je jednoduchý a nenáročný protokol pro předávání zpráv. Funguje na principu klient-server. Server je označován jako broker. Zprávy si předávají klienti mezi sebou pomocí brokeru. Pro transport je využit protokol TCP ve výchozím stavu na portu 1883, avšak není na TCP závislý a lze jej provozovat na jakémkoli jiném transportu, který splňuje bezztrátovost, doručení ve stejném pořadí a obousměrnou komunikaci. Protokol standardizuje organizace OASIS, patří tedy mezi otevřený standard. V současnosti existují dvě verze protokolu. MQTT 3.1.1 je standard z roku 2014 [4], také je standardizován organizací ISO pod kódem ISO/IEC 20922:2016. MQTT 5.0 je standard z roku 2019. Obsahuje různé vylepšení ohledně typů zpráv, velikosti paketu a další detaily [5].

Pro předání zpráv je použit vzor publisher-subscriber. Zprávy jsou zasílány do témat (topic). Subscriber si nahlásí u brokeru odběr z vybraných témat. Publisher odešle zprávu brokeru s vybraným tématem a broker ji automaticky doručí všem subscriberům, kteří mají nahlášen odběr tématu. Broker nemusí téma dopředu znát, podle požadavků ho dynamicky založí. Klient může být najednou v některých tématech publisher i subscriber. Zprávy mohou nést libovolný obsah, ten je nazýván payload. Maximální velikost zprávy je necelých 256 MB.

Témata mohou být hierarchicky rozdělena pomocí lomítek, například `budova1/patro5/místnost505/světlo`. Hierarchii je třeba dopředu promyslet, aby jednotlivé celky dávaly smysl a bylo je možné hromadně přihlásit. Klient může při odběru zpráv použít zástupné znaky `#` a `+`. Znak `+` nahrazuje jednu úroveň, například `budova1/patro5/+světlo` nahlásí odběr zpráv světel ze všech místností v pátém patře. Znak `#` nahrazuje jednu nebo více úrovní, proto musí být vždy jako poslední. Například `budova1/patro5/#` nahlásí odběr ze všech témat v pátém patře.

MQTT obsahuje i takzvané QoS (Quality of Service), rozdělené do tří úrovní 0, 1, 2. Úroveň 0 nezajišťuje žádné potvrzování zpráv. Publisher pošle zprávu typu `publish` brokeru a dál se nestará. Broker přeposílá zprávu se stejným QoS, tudíž pošle zprávu subscriberům, také typu `publish` a nestará se. Úroveň 1 zajišťuje doručení minimálně jednou. Publisher pošle zprávu `publish` brokeru a čeká. Broker pošle zprávu `publish` subscriberovi, taky s QoS 1. Subscriber odpoví zprávou `puback`, broker zprávu s payloadem odstraní a pošle publisherovi zprávu `puback`. Broker může i nemusí čekat na všechny subscribery, záleží na implementaci. Zpráva však může být přijata více krát, než dojde přijetí `puback`. Aby zpráva došla právě jednou, slouží úroveň 2. Na úrovni 2 posílá publisher brokeru zprávu `publish`.

Ten ji opět přepošle se stejným QoS subscriberem a publisherovi pošle zprávu `pubrec`. Publisher odpoví zprávou `pubrel`, broker zprávu s payloadem smaže a potvrdí zprávou `pubcomp` [22]. QoS úrovně 2 je na obrázku 2.12.



Obrázek 2.12: MQTT QoS 2 komunikace [41]

Mezi TCP a MQTT lze vložit TLS mezivrstvou, která zajistí šifrovaný tunel mezi klientem a brokerem. Pro šifrované MQTT spojení je vyhrazen port 8883. Je však nutné si zajistit potřebné certifikáty, případně vygenerovat si vlastní certifikační autoritu podepsanou sama sebou a tu následně bezpečným způsobem roz distribuovat do všech zařízení.

2.10 Node-RED

Node-RED je grafický nástroj, založený na grafu toku, za jehož vznikem stojí IBM. Jednotlivé programovací celky existují v podobě uzlů – nodů, které si programátor poskládá za sebe. Node-RED je napsaný v javascriptu a běží na platformě Node.js. Node.js je otevřená platforma pro tvorbu webových aplikací na straně serveru. Node.js obsahuje také systém

repositáře a balíčků, k instalaci a správě slouží příkaz `npm`. Pomocí `npm` lze nainstalovat celé Node-RED a případné přídatné moduly.

Node.js je napsaný v mixu C, C++, Javascript, takže umožňuje běh na jakékoliv procesorové architektuře. Nároky na HW počítače se liší podle zátěže a složitosti jednotlivých aplikací. Node-RED sám o sobě si vystačí s hardwarem RaspberryPi. Náročnost dále závisí na počtu nodů a počtu zpráv co jimi prochází. Bez konkrétní aplikace se nedá tedy náročnost příliš odhadnout.

Popularita nástroje v posledních letech roste. Podle průzkumu odpovědí uživatelů z roku 2019 má nejméně polovina dotázaných Node-RED nasazený v komerčním provozu, ať už samostatně nebo jako součást produktu [28].

Node-RED lze spustit přímo v terminálu, ovšem vhodnější je jej spouštět jako službu na pozadí. Node-RED se ovládá pomocí webového rozhraní, lze jej tedy obsluhovat odkudkoli. Jednotlivé grafy toků je možné exportovat i importovat ve formátu JSON, což umožňuje snadné zálohování práce a jednoduchou přenositelnost.

Node-RED není klasické programování. Jeho hlavní náplní je předávání zpráv z uzlu do uzlu. Formát a architektura zpráv je velmi podobná MQTT, což vlastně byla původní myšlenka vzniku Node-RED. Jednotlivé uzly přijmou zprávu, tu následně transformují a předají na výstup. V grafickém editoru pospojujeme jednotlivé uzly čarami tak, jak chceme aby zprávy proudily. Tím vznikne graf toku zpráv. Programování tedy není klasické, spustím program, zkontroluji vstupy, vytvořím výstupy a program následně spustím periodicky časovačem. Programování je orientováno na obsluhu událostí, které reprezentují jednotlivé příchody zpráv. Tento způsob vyžaduje přemýšlet nad problémem trochu jinak, než je běžné. Některé běžně řešitelné problémy jsou najednou komplikovaně řešitelné, například výpadek čidla. Výpadek čidla znamená, že zprávy od čidla přestanou přicházet. To způsobí zastavení provádění uzlů v grafu a tím znemožní reakci na tuto chybovou událost. Další překážka je s uchováváním stavu. Jednotlivé uzly pracují nezávisle na sobě. Jednoduchým příkladem je tvorba termostatu.

Na jeho vstup přichází zprávy o aktuální teplotě a na výstupu vychází zprávy zda vypnout nebo zapnout topení. Na první pohled elegantní provedení. Problém nastává jak z něj dostat parametry o teplotě při které má topení zapnout a při které vypnout. To lze provést samozřejmě zprávami s jiným tématem, na které uzel reaguje tak, že si hodnotu uloží k sobě dovnitř do proměnné. Pokud chceme stav zjistit, musíme opět zaslat zprávu s požadavkem na zjištění stavu, uzel následně vytvoří zprávu s jiným tématem, než jde do topení a s obsahem svého stavu. Překážka s uchováním stavu je tedy řešitelná, avšak řešení není tolik elegantní.

S tím nastává otázka, zda je možné v Node-RED vyřešit všechny možné problémy. Odpověď je samozřejmě ano. Programování uzlů, například výše zmíněného termostatu, se tvoří v jazyku Javascript. Ten je spouštěn každou příchozí zprávou. Tudíž stačí před něj vložit uzel s časovačem, který v nastavený čas odešle zprávu, která způsobí spuštění kódu. Tento postup je však mimořádně neelegantní. Narušuje koncept toku zpráv a tudíž i znepráhlední situaci v grafu, ve kterém pak není jednoduché se zorientovat, tudíž tomuto způsobu je vhodné se vyhnout za každou cenu.

Vlastní uzly je tedy možné si naprogramovat. Pokud bychom chtěli rozšířit funkcionalitu víc, či ji rovnou přidat, nabízí se možnost instalace přídatných modulu. Přídatné moduly je možné vyhledat a instalovat rovnou z webového rozhraní, ovšem tato varianta ne vždy funguje, je to vhodné akorát pro kontrolu co za moduly je nainstalováno. Praktičtější je najít si modul na oficiálních stránkách. Ve většině případů je tam rovnou uveden popis modulu a ukázky k vyzkoušení, které vám pomůžou se zorientovat. Samotný modul pak

nainstalovat pomocí příkazu `npm install <název-modulu>`. Příkaz sice přímo nevyžaduje spuštění s právy roota avšak bez nich většinou selže na přístupu k souborům. Před instalací je vhodné Node-RED vypnout a zapnout až po instalaci.

Mezi moduly lze najít konektory k všelijakým databázím, internetovým úložištím a cloudovým službám. Dále potom implementaci a obsluhu různých komunikačních protokolů. Za zmínku stojí Aedes MQTT broker, který vám vytvoří MQTT broker rovnou v Node-RED a nemusíte jej řešit někde bokem. MQTT klient uzly jsou v Node-RED již nativně vestavěny. Dalším a asi nejpoužívanějším modulem je, v plném názvu, `node-red-dashboard`. Jak název napovídá, modul umožňuje vytvořit tak zvaný dashboard, což je takový uživatelský ovládací panel. Panel je samozřejmě také webový. Běží na stejné adrese jako rozhraní Node-RED, akorát se do adresy doplní `/ui`. Dashboard je možné tvořit z předdefinovaných prvků, jako jsou tlačítka, textová pole, posuvníky, ukazatele a další prvky.

2.11 Virtuální server a Docker kontejner

Pro řídicí systém je dobré zajistit nepřetržitý běh. To zahrnuje vybrat spolehlivý hardware a zajistit zálohované napájení. Jako ideální se jeví využití stávající serverové infrastruktury. Ta je stavěná na nepřetržitý provoz a zahrnuje i zálohované napájení v případě výpadku. Na serverech běží virtualizační nástroj XCP-ng. Ten zajišťuje běh jednotlivých virtuálních počítačů. Stačí tedy vytvořit virtuální stroj a nainstalovat operační systém. Nároky Node-RED nejsou vysoké. Pro začátek by mohly stačit parametry RaspberryPi, tedy jeden procesor a k tomu 1 GB RAM a 10 GB diskového prostoru. Jako podkladový operační systém je ideální Debian v aktuální stabilní verzi. Debian samotný je relativně malý a přehledný linuxový operační systém. Podpora stabilního vydání je tři roky plus další dva roky bezpečnostní záplaty, celkově tedy pět let. Každé tři roky vychází nová stabilní verze. Přesné datum v roce není stanoven, pokaždé záleží na aktuálních problémech, případně se čeká na vydání linuxového jádra dlouhodobou podporou. Další výhodou je jednoduchá záloha celého stroje a případné obnovení při chybě. Jedná se o způsob plně virtualizace.

Pro lehkou virtualizaci je možné použít Docker. Docker je virtualizační technologie na úrovni operačního systému. Umožňuje běh operačních systémů, které sdílí jádro hostujícího systému. Jednotlivé virtualizované operační systémy mají vlastní oddělený diskový prostor. Systémy jsou od sebe odděleny na úrovni systémových procesů. Virtualizovaný operační systém nemá tedy vlastní jádro a jeho procesy běží na jádře podkladového systému. Virtualizovaný operační systém se distribuuje v podobě obrazu, který je již předem nakonfigurovaný a stačí jej pouze spustit. Existuje repositář Docker HUB, kde se nachází obrazy skoro všech používaných operačních systémů. Pro stažení obrazu Debianu stačí příkaz `docker pull debian`. Systém pak spustíme příkazem `docker run debian`. Se spuštěním systému vznikne takzvaný kontejner s vygenerovaným identifikátorem, konfigurací a konkrétním obrazem disku, který už není jako původní stažený, protože běžící stroj má do něj povolený zápis.

Kontejnery lze vypsat příkazem `docker ps` a `docker ps -a`. Do běžícího operačního systému se lze dostat spuštěním příkazového interpretu, například `bash docker exec -u 0 -it <ID nebo název> /bin/bash`. Přepínač `-i` společně s `-t` zajistí interaktivní režim, to znamená, že přepíše současný terminál interpretem z virtuálního stroje. Přepínač `-u 0` vybere ID uživatele ve virtuálním kontejneru, tedy uživatele root.

Sít je standardně překládána NATem s maškarádou, virtuální kontejnery dostanou IP adresu z privátního rozsahu a mají tedy přístup do stejné sítě jako podkladový operační systém. Problém však nastane pokud chcete adresovat službu uvnitř virtuálního kontejneru. Na

to slouží příkaz pro přemostění síťových portů. Například `docker run -p 1880:1880/tcp debian` protuneluje TCP spojení z vnějšího portu 1880 do kontejneru na port 1880. Pokud potřebujeme změnit nastavení portů již existujícího kontejneru, je potřeba kontejner vypnout `docker stop <ID nebo název>`, následně uložit aktuální obraz pod jiným názvem, tím vlastně vznikne záloha aktuálního stavu `docker commit <ID> <název nového obrazu>`. Následně znovu založit nový kontejner s novým nastavením portů, například `docker run -p 8080:80/udp <název nového obrazu>`.

Pokud chceme kontejner aktualizovat, jednoduše stáhneme novou verzi příkazem `pull` a založíme nový kontejner příkazem `run`, tím ovšem přijdeme o konfiguraci a data. K tomu se může hodit tunelování složek. Například

```
docker run -v /home/user/data_z_virtualu:/slozka_ve_virtualu debian
```

protuneluje složku `/home/user/data_z_virtualu` z podkladového systému do kontejneru do složky `/slozka_ve_virtualu`.

Docker se hodí také pro aplikace které je potřeba často aktualizovat nebo jsou závislé na novějších knihovnách než nabízí stabilní operační systém. Tvůrci aplikací tomu jdou naproti a vydávají aplikace ve formě rovnou celého obrazu. Obrazy jsou postaveny na minimálním operačním systému jenom pro potřeby aplikace. Aplikace je naprogramována tak, aby ukládala konfiguraci do jedné složky, která se následně protuneluje výše zmíněným mechanismem. Tím je zajištěna jednoduchá aktualizace aplikace pouhou výměnou obrazu kontejneru.

Kapitola 3

Analýza současného stavu

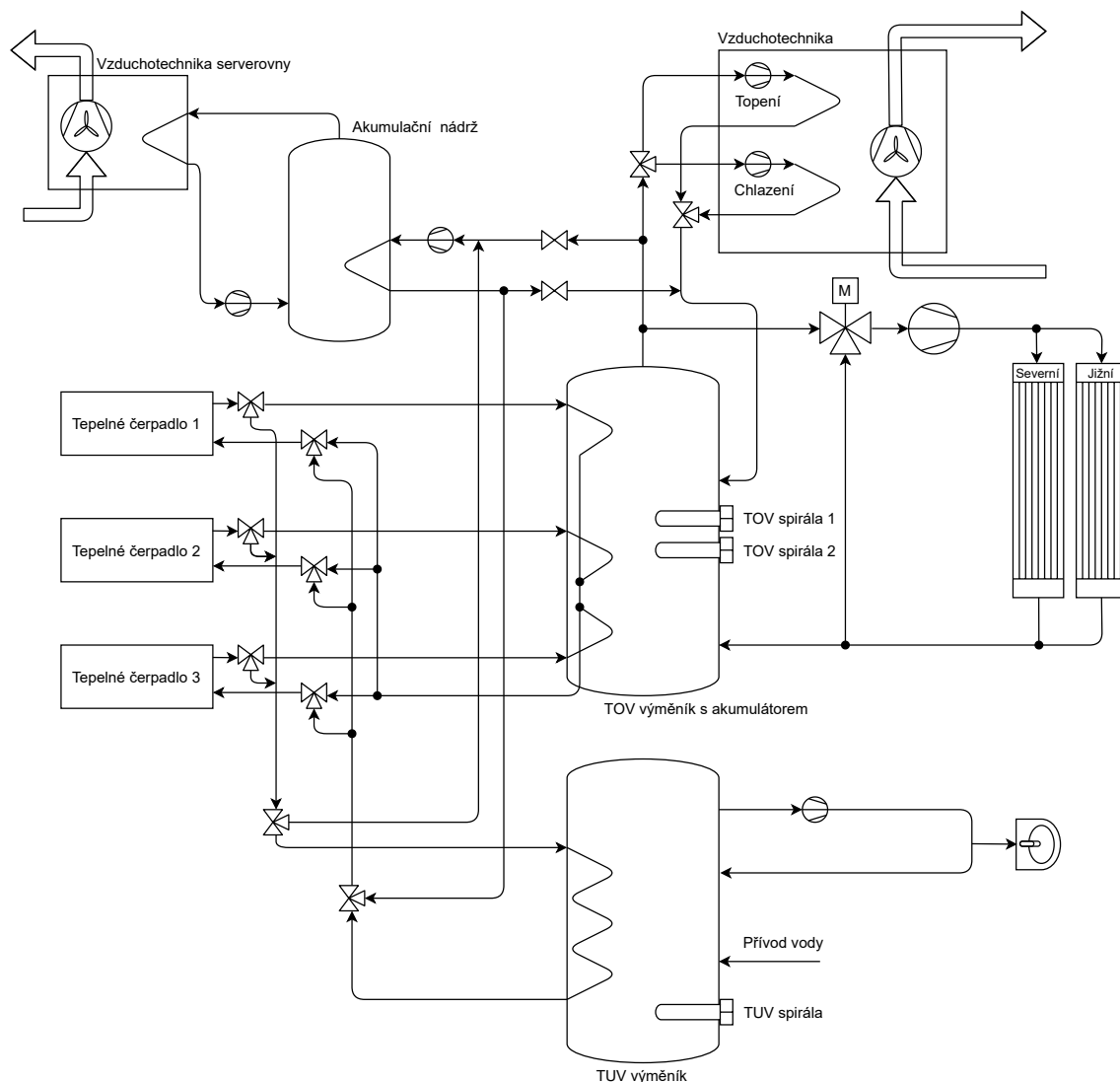
Ve SPOLEČNOST-24 s.r.o. je 32 místností, které vyžadují termoregulaci. Některé z místností například serverovna a sklad jsou obsluhovány separátně, než zbytek obývaných místností. Každá místnost má nějaký počet otopných těles. Každé otopné těleso je ovládáno jedním ventilem. Ventil je umístěn na přípojném místě, stejně jako otopné těleso. Ventil je ovládán sepnutím nebo rozepnutím síťového napětí 230 V. V každém přípojném místě je umístěn malý rozvaděč, do kterého jsou přivedeny všechny ventily v daném přípojném místě. V malém rozvaděči je pouze přepojení na dvanácti-žilový kabel, který vede do hlavního rozvaděče. V hlavním rozvaděči lze tedy ovládat každý konkrétní ventil. Ventily jsou připojeny náhodně do PLC jednotky UniPi. V jednotce UniPi je nahrán program, který je vytvořen v prostředí Mervis IDE. Program přečte jeden digitální vstup a podle něj všechny ventily zapne nebo vypne. Dále pak čte další digitální vstup, který neguje logiku ovládání ventilů. Tímto je dosaženo přepínání režimu léto a zima. První vstup je připojen na přijímač bezdrátového teploměru s termostatem. Druhý vstup je ovládán manuálně přepínačem. V zimě při dosažení horní hranice všechny ventily vypne a při nízké hranici zapne. Naopak v létě při nízké hranici všechny ventily vypne a při vysoké zapne. Všechny ventily jsou tedy ovládány jedním teploměrem s jedním termostatem.

Jako zdroj tepla jsou využity tři tepelné čerpadla Sinclair SHP-180IRC. Ty pracují nezávisle na sobě. Přenos tepla je rozdělen na dva okruhy. Okruhy jsou odděleny výměníkem. Výměníky jsou dva. Jeden je určen pro ohřev teplé užitkové vody (TUV), která je napojená do umyvadel a sprch. Druhý je určen pro teplou oběžnou vodu (TOV), která je připojená do otopných těles. Výměník TOV je větší, aby zajišťoval schopnost akumulace tepla. Každé tepelné čerpadlo je připojeno do obou výměníků a každé tepelné čerpadlo je tedy schopno topit nebo chladit vybraný výměník nezávisle na sobě. Tím je zajištěna rezerva i redundance v případě výpadku. Například první tepelné čerpadlo topí TUV, druhé topí TOV, třetí je rezerva. V případě větší zimy třetí čerpadlo pomáhá druhému v topení TOV. Pro případ poruchy lze funkcionalitu zpřeházet dle potřeby. V případě ještě větších poruch lze využít elektrické topné spirály. Každá má výkon 3 kW, jedna je ve výměníku TUV a dvě ve výměníku TOV. Příkon jednoho tepelného čerpadla je 4 kW, dále je schopno dodat od 4 až do 16 kW tepelné energie. V závislosti na venkovní teplotě a výstupní teplotě. Čím větší je rozdíl teplot tím je horší účinnost. Ovšem i při nejhorších podmínkách je účinnost pořád lepší než elektrické spirály.

Systém je tedy rozdělen na primární a sekundární okruh. Primární okruh je mezi tepelnými čerpadly a výměníky. Sekundární okruhy jsou mezi výměníky a otopnými tělesy, umyvadly. V sekundárním okruhu TOV je zařazen trojcestný ventil v režimu směšování. Ten reguluje teplotu vody dodávanou do otopných těles a tím i odběr tepla z výměníku. Ventil

je v manuálním režimu a je občas ručně doladěn. Teplota výstupní vody je kontrolována jedním teploměrem na jednom přípojném místě v kanceláři. Rozvod na jednotlivá přípojná místa je proveden trubkami s velkým průměrem, jedna část vede po severní straně budovy, druhá část vede po jižní straně budovy.

Nastavení tepelných čerpadel je prováděno manuálně, pomocí cloudového řešení firmy Sinclair. Tepelné čerpadla jsou tedy připojena do ethernetové sítě a do internetu. Sít má svou vlastní oddělenou VLANu, kontakt s internetem filtruje firewall.



Obrázek 3.1: Schéma kotelny

Schéma kotelny je na obrázku 3.1. Trojcestné ventily u jednotlivých tepelných čerpadel jsou ovládány řídicí logikou tepelného čerpadla. Každé čerpadlo tedy ovládá dva své ventily, jeden rozdělovací pro výstupní vodu a druhý směšovací pro vstupní vodu. Teoreticky by se dalo ventily nastavit doprostřed, aby jedno tepelné čerpadlo topilo oba výměníky zároveň. Prakticky jsou ventily vždy v jedné nebo v druhé poloze, takže jedno tepelné čerpadlo topí v jednu chvíli pouze jeden výměník. Řízení se provádí nastavením parametrů tepelného čerpadla. Lze nastavit sedm parametrů. Zapínací a vypínací teplota topení TUV. Zapínací

a vypínací teplota topení TOV. Zapínací a vypínací teplota chlazení TOV. Nakonec priorita mezi TOV a TUV, ta určí, která se bude obsluhovat dřív.

Dále jsou ve společnosti dvě vzduchotechniky. Jedna obstarává výměnu vzduchu v místnostech. V zimním období vzduchotechnika pomáhá s topením. V letním období je využita také pro chlazení vzduchu. Teplo nebo chlad je odebírán z výměníku TOV. Druhá vzduchotechnika je použita pro obsluhu serverovny. Je napojená na vlastní akumulční nádrž. Tu je dále možné připojit na TUV výměník nebo na tepelná čerpadla. V současné době je vzduchotechnika a její akumulční nádrž odstavená a nepoužívá se.

V jednotlivých místnostech jsou ve vypínačích nainstalovány teploměry. Od každého vypínače i teploměru je natažený kabel do hlavního rozvaděče. V rozvaděči nejsou kabely nijak ukončeny, jsou zaizolovány v tašce a nepoužívají se.

Ve společnosti se v serverovně používají PLC jednotky od UniPi, jsou k dispozici i starší kusy, které se již nepoužívají a je možnost je využít. Dále je zde problém s vypadáváním elektrického proudu v kotelně. Kotelna je napájena jiným separátním přívodem, než serverovna a zbytek místností. Podezření padá na některý ze stovky ventilů. Při zapnutých tepelných čerpadlech nelze přesně určit co tento problém způsobuje. Rozvod je trochu poddimenzován a při vytížení spadnou jističe po celé cestě až k přívodu.

Kapitola 4

Vlastní návrh řešení

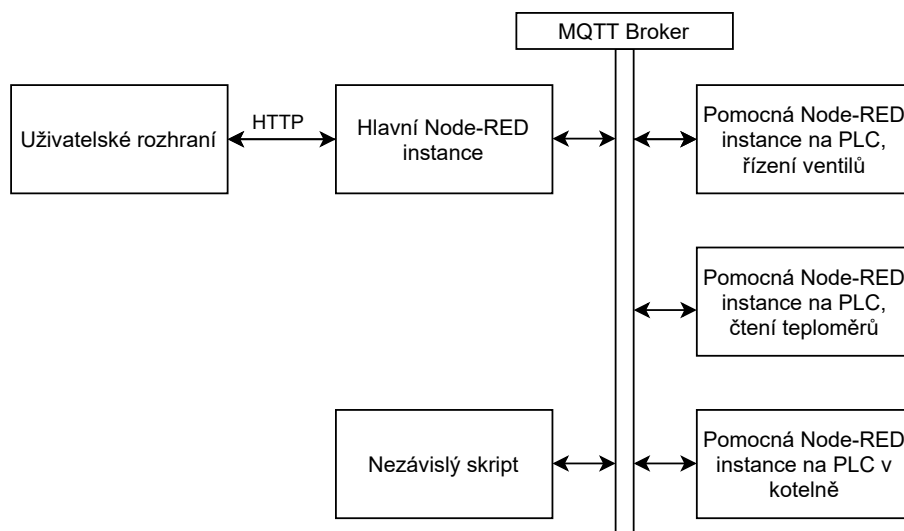
Nejpodstatnější rozhodnutí na začátku je zvolit softwarovou platformu, na které se následně postaví systém. Možnosti jsou dvě. První možnost je použít Mervis. Mervis IDE umožňuje naprogramování logiky do PLC, řízení PLC. Mervis SCADA umožňuje vytvořit uživatelské rozhraní a nakonec použít Mervis DB pro ukládání historie dat. Druhá možnost je použít Node-RED a do něj přidat moduly, které umožní požadovanou funkcionalitu. Mervis se jeví jako profesionální a spolehlivá varianta. Zkoušel jsem chvíli pracovat s Mervis IDE, který je současně nasazen a usoudil jsem, že systém je příliš komplikovaný pro jednoduché použití, nedá se v něm snadno orientovat a vyžaduje určité zkušenosti. V neprospěch Mervisu dále hraje roli cena. Mervis IDE/RT je k jednotkám UniPi již v ceně. Mervis SCADA už vyžaduje měsíční předplatné podle počtu datových jednotek. Do 25 datových bodů předplatné není potřeba, avšak podle velikosti projektu předpokládám, že datových bodů bude přes 100. Další nevýhoda je rozšiřitelnost. Během projektu může nastat problém, kdy bude potřeba nějakou funkčnost doprogramovat. Mervis je uzavřená platforma, tudíž bychom museli požádat výrobce o přidání funkcionality. S tím souvisí připojení externích zařízení, které nejsou určeny pro komunikaci s Mervisem.

Z výše zmíněných důvodů jsem rozhodl pro použití Node-RED. Node-RED je primárně určen pro propojování komponent a případně nějakou jednoduchou logiku. Funkcionalitu je možno rozšířit pomocí modulů. Node-RED je otevřený software, to přináší výhody i nevýhody. Nevýhoda je, že za software nikdo neručí a tím i kvalita by nemusela vždy být na profesionální úrovni. Výhoda je, že software je plně pod lokální kontrolou a můžeme s ním nakládat libovolně, tedy i doprogramovat moduly, které rozšíří funkcionalitu.

S Node-RED se nutně pojí protokol MQTT, který je ideální pro spolehlivý přenos zpráv mezi všemi prvky systému. Pro tvorbu uživatelského rozhraní bude využit modul `dashboard`. Jelikož jednotky UniPi Neuron obsahují RaspberryPi, lze Node-RED spustit přímo na nich. Tím nastává otázka kam umístit hlavní Node-RED instanci na které bude modul s uživatelským rozhraním. Hlavní instance bude rovněž obsahovat pokud možno veškerou rozhodovací logiku. Vzhledem k problému vypadávání elektřiny v kotelně, jsem se rozhodl využít firemní infrastruktury [11] a vytvořit malý virtuální server hostovaný na serveru v serverovně. Tím je zajištěn spolehlivý a nepřetržitý provoz. Parametry virtuálního serveru jsou 1 GB RAM, jedno jádro procesoru Xeon 2,6 GHz a 10 GB diskového prostoru. V případě zvyšujících se nároků je možné parametry navýšit.

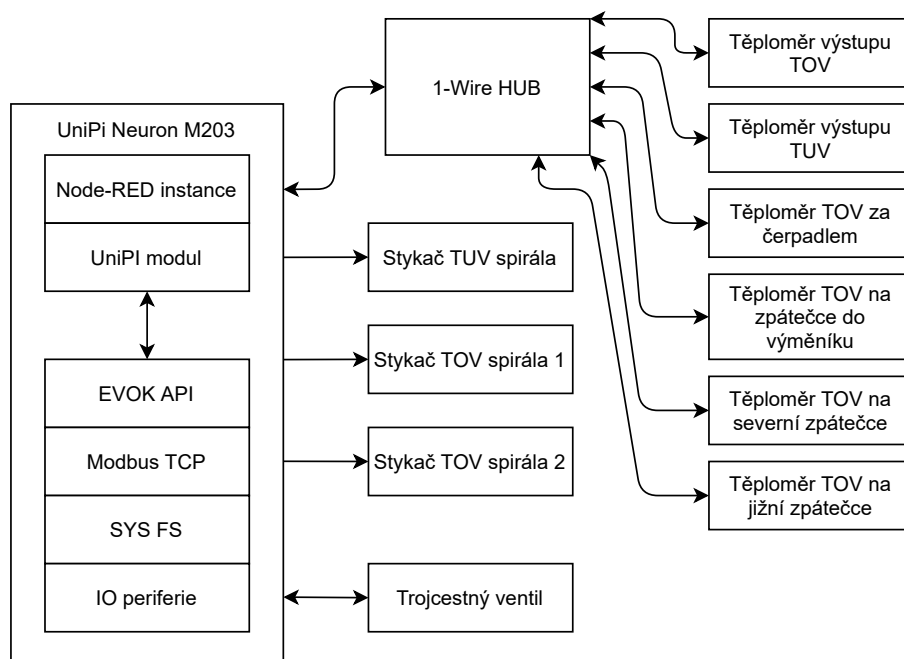
Na virtuální server jsem vybral distribuci Debian, stabilní větev v současné verzi 10. Software v Debianu je sice stabilní, avšak rychle se vyvíjející software jako je Node-RED bychom měli pořád starou verzi. Proto jsem se rozhodl využít předchystaného docker obrazu s Node-RED, tím lze snadno zajistit aktualizaci.

Na obrázku 4.1 je hlavní projení systému. Hlavní instance Node-RED společně s MQTT brokerem běží na výše zmíněném virtuálním serveru. K němu se připojují ostatní jednotky pomocí MQTT protokolu. Přes MQTT je připojen i externí skript, který obstarává komunikaci s tepelnými čerpadly. Ta je zajištěna automatickou komunikací s lokálním webovým rozhraním tepelného čerpadla.



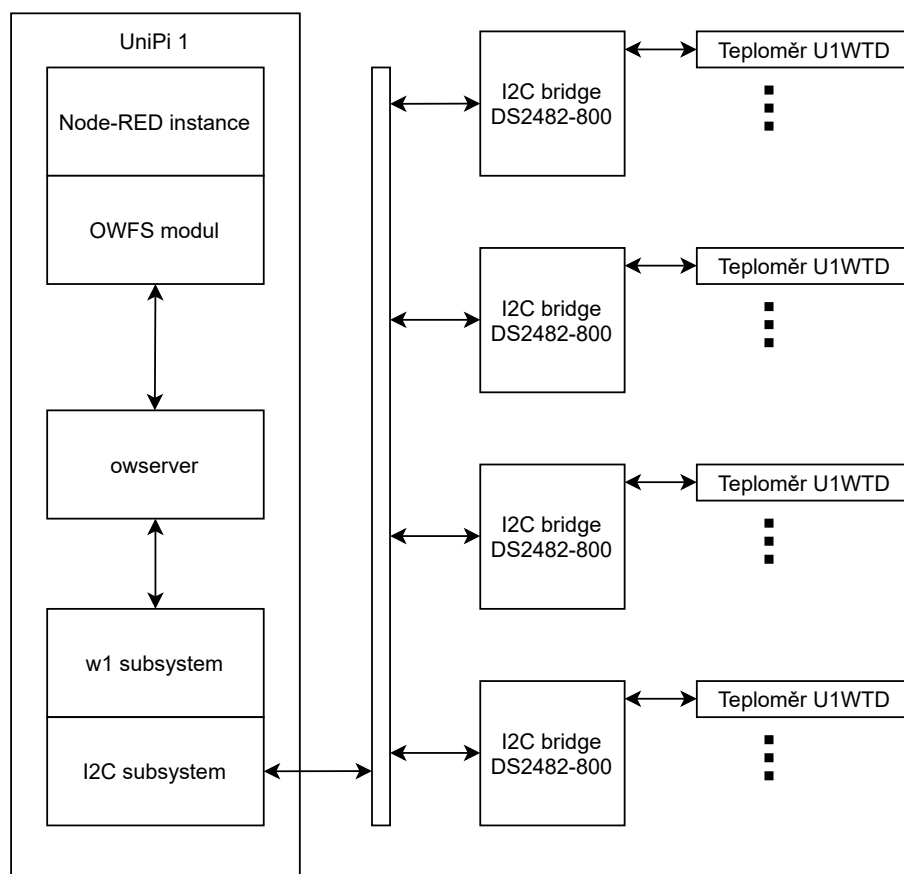
Obrázek 4.1: Hlavní propojení systému

Na obrázku 4.2 je schéma zapojení prvků v kotelně. Teploměry jsou zapojeny pomocí 1-Wire HUBu. Ten zajistí propojení senzorů do topologie sběrnice.



Obrázek 4.2: Schéma ovládání kotelny

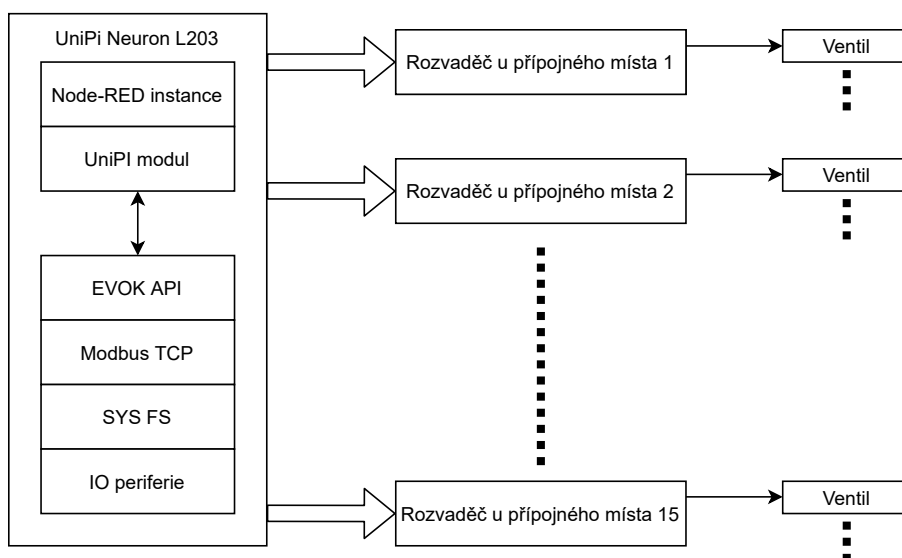
Pro připojení teploměrů jsem zvolil přímočarou variantu, která bude spolehlivá. Dokumentace od Maxim Integrated nedoporučuje zapojení 1-Wire zařízení do topologie hvězda, dokumentace od UniPi tento způsob dokonce zakazuje. Použití 1-Wire HUBu není přímo možné, ke každému teploměru by se musel připojit drát navíc. Proto jsem se rozhodl využít starších rozšiřujících desek k UniPi 1. Každá deska obsahuje osm 1-Wire master portů. Do každého portu se připojí jeden senzor. Tím odstraníme problémy s odrazy signálu a rušením od ostatních zařízení. Navíc v případě problému na jednom teploměru by problém neměl ovlivnit ostatní senzory. Struktura zapojení je na obrázku 4.3.



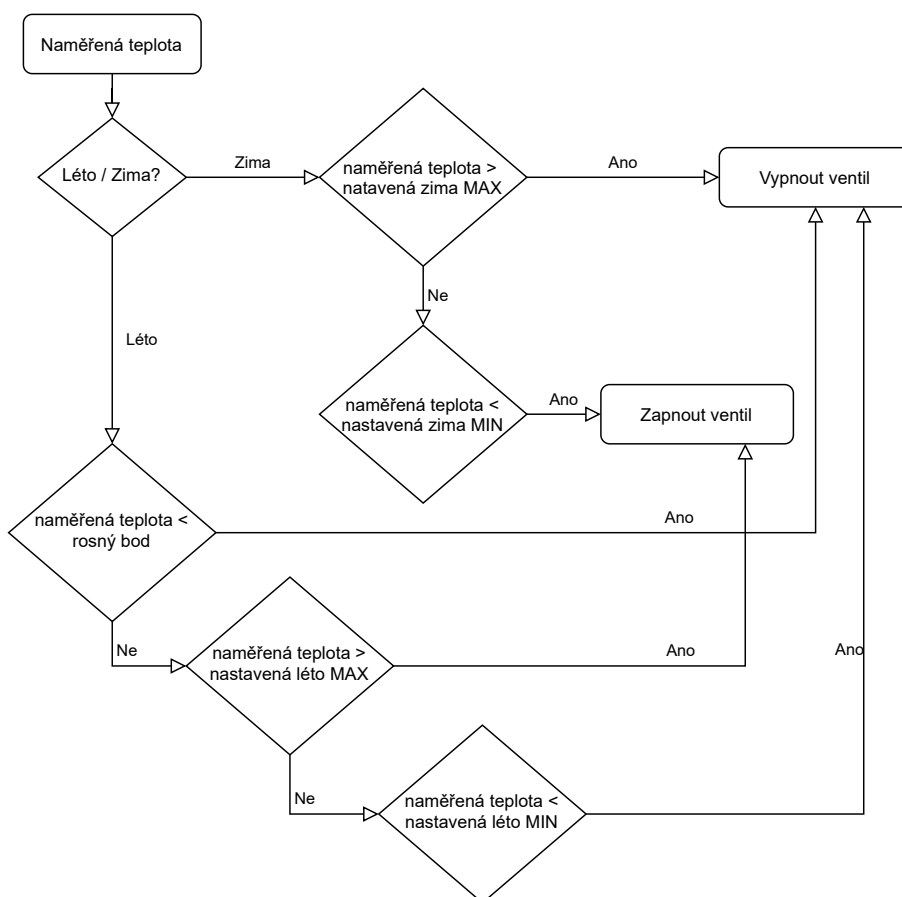
Obrázek 4.3: Schéma připojení teploměrů

Ventily otopných těles jsou přivedeny do hlavního rozvaděče. Ventily se připojí na porty tak, aby bylo možno ovládat každou místnost separátně. Jelikož je ventilů kolem sto kusů je potřeba využít jednotky UniPi Neuron L203. Ta má nejvíc relé portů. Ty spínají síťové napětí 230 V a tím otevřou ventil. Nepřipojený ventil je ve výchozí poloze zavřený. Nákres zapojení je na obrázku 4.4.

Nyní máme k dispozici ovládání ventilů pro každou místnost a k tomu teploměr pro každou místnost. Každá místnost bude mít svůj termostat. Ten bude na základě dat z teploměru ovládat ventily pro danou místnost. Rozhodování termostatu je na diagramu 4.5. Termostat má zimní i letní režim. Pro zimní a letní režim je nastavení teplot oddělené. Letní režim musí při chlazení zohlednit teplotu rosného bodu, aby nedošlo ke srážení vody na stěnách.



Obrázek 4.4: Schéma ovládání ventilů otopných těles



Obrázek 4.5: Diagram obsluhy ventilů pro jednu místnost

Kapitola 5

Realizace řešení

Nejprve byla zvolena výsledná vývojová platforma Node-RED a komunikace mezi prvky pomocí MQTT. Veškerá řídicí logika je naprogramovaná v hlavní instanci Node-RED. Hlavní instance je tedy nezbytně nutná pro provoz a funkčnost celého systému. Pro tento účel bylo výhodné využít současné firemní infrastruktury, která zahrnuje několik serverů umístěných v klimatizované místnosti, včetně záložního napájení. Na jednom serveru běží virtualizační nástroj Xen Orchestra. V něm byl vytvořen virtuální stroj s parametry 1x CPU, 1 GB RAM a 10 GB diskového prostoru. Do něj byl nainstalován současný Debian 10 stabilní verze. Následně byla nainstalována virtualizační aplikace Docker z oficiálních repositářů Debianu. Do dockeru byl stažen oficiální obraz `node-red`. MQTT broker byl nasazen rovnou v Node-RED, instalací balíčku `node-red-contrib-aedes`. Ostatní zařízení se tedy připojují na adresu virtuálního stroje. Webové uživatelské rozhraní je vytvořeno také v Node-RED pomocí balíčku `node-red-dashboard`.

Tímto byl položen základ realizace. Následná realizace probíhala přírůstkovou metodou a nasazována rovnou do praxe. Nejdříve bylo zapotřebí rozběhnout sběr teplot z jednotlivých místností. Kabely od jednotlivých teploměrů již byly přivedeny do jednoho místa za instalační skříň. I2C bridge je z výroby osazen konektory RJ45. Toho jsem se rozhodl využít a jednotlivé kabely byly připojeny do standardního patch panelu, který je používá pro ethernetové rozvody. Patch panel je propojen standardními patch kabely s jednotlivými I2C bridge. Všechny I2C bridge jsou propojeny průběžnou I2C sběrnici společně s UniPi1. Pro I2C jsou použity konektory RJ11. Společně s I2C je vedeno také napájení 5 V, které je společně s UniPi1, potažmo s RaspberryPi na něm osazené. Toto napájení je využito pouze pro napájení převodního čipu. Připojené senzory sice fungují, ale jenom z parazitního odběru 1-Wire. Proto každý I2C bridge požaduje přídavné napájení 5 V, které slouží pro napájení sběrnice. Toto napájení zajišťuje přídavný zdroj.

Před zapojením byly zkontrolovány volné adresy příkazem `i2cdetect -y 1`. Poté byla na převodnicích nastavena volná adresa pomocí přepínačů. Připojení do systému a zavedení ovladače `w1` bylo provedeno příkazem

```
echo ds2482 0x1a > /sys/bus/i2c/devices/i2c-1/new_device.
```

Typ převodníku je `ds2482` a adresa na I2C je `1a`. Ovladač `w1` založí soubor pro každé připojené zařízení na 1-Wire ve složce `/sys/bus/w1/devices`. Odpojení převodníku ze systému lze provést příkazem `echo 0x1a > /sys/bus/i2c/devices/i2c-1/delete_device`. Při přepojování 1-Wire zařízení byl objeven drobný problém, že soubor se stejným názvem již existuje. Soubory se pojmenovávají podle ID zařízení. Možností je buď počkat až ovladač prohlásí zařízení za odpojené nebo připojit odpojit převodník podle výše zmíněných příkazů. Dále byla nainstalována aplikace `owserver` a `owhttpd` z oficiálních repositářů. Do konfigurace byl

přidán modul w1. Zařízení byla zkontrolována ve webovém rozhraní na portu 2121. Rovněž byly zapsány identifikátory jednotlivých teploměrů, aby šlo určit, kde se daný teploměr nachází.

Do Node-RED byl nainstalován modul `node-red-contrib-owfs`. V Node-RED byl vytvořen seznam identifikátorů zařízení, tedy teploměrů. Podle seznamu se cyklicky se posílají dotazy na `owserver`. Zprávy s naměřenou teplotou se následně odesílají přes MQTT do hlavní instance. Topic je pojmenovaný `/místnosti/<číslo místnosti>/temperature`. Přečtení všech 32 teploměrů trvá kolem 5 minut.

Následovala implementace ekvitermní křivky a regulace hlavního trojcestného ventilu. Pro výpočet ekvitermní křivky je potřeba venkovní teplota. Zpočátku byla venkovní teplota čtena ze senzoru tepelného čerpadla. Senzor je umístěn na noze venkovní jednotky tepelného čerpadla. Jakmile dorazí zpráva s venkovní teplotou je vypočítána ideální teplota oběžné vody. Trojcestný ventil následně reguluje výstupní teplotu vody tak, aby byla co nejbližší ideální teplotě.

Pro ovládání trojcestného ventilu a sběr teplot z jednotlivých částí systému je v kotelně umístěn UniPi Neuron M203. Ten obsahuje analogový výstup a vstup, který je připojen do trojcestného ventilu. Trojcestný ventil je tak možné regulovat plynule po celé škále. Při hodnotě 0 V protéká voda vratnou trubkou, při 10 V protéká voda TOV výměníkem. Analogový vstup slouží pro čtení aktuální pozice ventilu. Pro jednoduchou regulaci jsou zvoleny kroky po 0,1 V. Pro zpětnou vazbu je využit teploměr umístěný za čerpadlem teplé oběžné vody.

Aby mohl trojcestný ventil ovlivnit teplotu výstupní vody, potřebuje mít ve výměníku TOV vyšší (při chlazení nižší) teplotu vody. Teplota vody v TOV výměníku je tedy také regulována podle ekvitermní křivky. Vypínací a zapínací teplota je tedy také ovlivněna ekvitermní křivkou. Rozdíl mezi zapínací a vypínací teplotou je zvolen tak, aby nedocházelo k příliš častému vypínání a zapínání kompresoru tepelného čerpadla, zároveň však tak, aby vypínací teplota nebyla příliš vysoko. Je to z důvodu klesající účinnosti při zvyšující se teplotě.

Pro správný chod systému je tedy venkovní teplota důležitá. Proto bylo nutné zajistit její spolehlivost. Nakonec je venkovní teplota čtena ze všech tří tepelných čerpadel. K tomu je čtena venkovní teplota z webu nejbližší meteostanice Brno–Tuřany. Výsledné teploty jsou následně průměrovány. Venkovní teplota je tedy dostupná i při výpadku všech tří tepelných čerpadel.

V režimu chlazení je potřeba brát ohled na teplotu rosného bodu. Ta je získána ze senzoru teploty a relativní vlhkosti a následně vypočítána podle vzorce 2.4. Venkovní teplota rosného bodu je opět brána z nejbližší meteostanice Brno–Tuřany. Výsledná teplota rosného bodu je vybrána vždy ta vyšší. Chlazení tedy nechladí pod tuto teplotu.

Následovalo zprovoznění ovládaní jednotlivých ventilů k otopným tělesům. Ventily jsou ovládány reléovými výstupy UniPi Neuron L203. Každá místnost obsahuje několik ventilů, které se jsou řízeny současně najednou. Tím může být několik ventilů připojených na jedno výstupní relé a tím ušetřit počet výstupů na Neuronu. Jednotlivé relé výstupy jsou v hlavní instanci Node-RED přemapovány na čísla místností. Jednou zprávou lze vypnout a zapnout ventily pro celou konkrétní místnost.

Nyní máme k dispozici, ke každé konkrétní místnosti, teplotu a ventily k otopným tělesům, přichází na řadu termostaty. Pro každou místnost jeden termostat, místnosti jsou tedy řízeny separátně. Termostat je mírně odlišný od navrhovaného řešení. Je v něm navíc kontrola teploty otopné vody. Pokud je otopná voda chladnější (teplejší), než požadovaná teplota, je ventil uzavřen. Termostat obsahuje separátní nastavení hodnot pro režim topení

a chlazení. Hysterezi lze také nastavit zvlášť na každém termostatu. Navíc termostat zobrazuje zda je ventil zrovna otevřen či nikoli. Vyhodnocovací logika termostatu se spouští s každým příchodem zprávy s teplotou místnosti. Nastavené teploty lze uložit i načíst ze souboru ve formátu JSON. Načtení hodnot ze souboru umožňuje případné hromadné změny hodnot.

5.1 Řízení tepelných čerpadel

Tepelná čerpadla obsahují vlastní logiku řízení. Každé tepelné čerpadlo má vlastní senzor teploty v TUV i TOV akumulární nádrži. Nastavení teplot lze provádět přímo na fyzickém ovládacím panelu tepelného čerpadla. Každé tepelné čerpadlo rovněž obsahuje ethernetový modul. Ten slouží pro komunikaci s cloudovou aplikací a k odesílání diagnostických dat o provozních parametrech na server výrobce. Následně bylo zjištěno, že ethernetový modul obsahuje také lokální webové rozhraní. S pomocí interního nástroje se podařilo komunikaci přeložit do MQTT protokolu. Tento nástroj spadá do trendu posledních let robotické automatizace procesů [12]. Jedná se o pokročilý skript v jazyce Python. Skript je spouštěn periodicky každou minutu. Skript přečte hodnoty z webového rozhraní a odešle je do patřičných MQTT témat. Následně v jiném MQTT tématu obdrží, zda má kliknout na vypnout, topení nebo chlazení.

Jelikož tepelná čerpadla nemají přehled o celkovém stavu systému, tudíž jejich logika neřídí čerpadlo optimálně. Logika řízení tedy byla implementována v hlavní instanci Node-RED. Logika v Node-RED dává pouze příkazy vypnout, topení a chlazení. Rozhodování závisí na aktuální vystupující teplotě vody z výměníku a požadované teplotě podle ekvitermní křivky. Do tepelných čerpadel byly nastaveny vysoké, při chlazení nízké, teploty tak, aby vnitřní logika tepelného čerpadla neovlivnila rozhodování nadřazené logiky.

5.2 Hlavní ovládací panel

Hlavní ovládací panel (dashboard) nebo-li uživatelské rozhraní vznikalo také přírůstkově s příchodem jednotlivých prvků. Panel je rozdělen do několika karet, podle účelu funkce. Karty lze přepínat v levém vyjíždějícím seznamu. Obsah karty je rozdělen do sekcí. Sekce obsahují ovládací prvky, které spolu souvisí a tím tvoří celek ovládání například jednoho zařízení. V sekcích se již nachází zmíněné ovládací prvky typu tlačítko, přepínač, posuvník, výsuvné seznamy, měřidla, a tak dále. Karty rozdělují hlavní logické celky. Karta Teploměry slouží pro zobrazení stavu všech teploměrů v místnostech. Zobrazuje také obnovování jednotlivých teploměrů, tím lze rychle poznat zda systém pracuje správně. Karta Ventily slouží k přímému ovládání reléových výstupů Neuronu. Byla užitečná v případě hledání chyb ve ventilech k otopným tělesům. V průběhu realizace se objevovaly chyby v zapínání ventilů v podobě padajícího 16 A jističe. Bylo potřeba jednoduše vypínat a zapínat jednotlivé výstupy, aby bylo možné odhalit problém. Karta obsahuje také přepínač automatické regulace, kterým je možné odstavit ovládání výstupů z předřazené logiky. Mezitím je karta Termostaty. Jedna sekce je vždy jeden termostat. Termostat obsahuje název místnosti, měřidlo s aktuální teplotou, čtyři číselné vstupy pro nastavení minimální a maximální hodnoty pro topení a chlazení, a přepínač pro indikaci stavu ventilu. V první speciální sekci se nachází tlačítka pro ukládání a načítání nastavených hodnot v termostatech. Dále se zobrazuje teplota otopné vody a režim chlazení nebo topení. Režim je pro všechny termostaty stejný.

Hlavní karta s názvem *Kotelna* obsahuje hodnoty ze všech teploměrů rozmístěných v kotelně, nastavení a aktuální stav trojcestného ventilu, nastavení zapínací a vypínací teploty elektrických spirál, nastavení vzduchotechniky serverovny, aktuální hodnoty ekvitermní křivky, stavy tepelných čerpadel, nastavení tepelných čerpadel. Nastavení zapínací a vypínací teploty elektrických spirál je také řízeno ekvitermní křivkou. Je zde také globální nastavení režimu chlazení nebo topení. Následují už jen karty se stavy a hodnotami senzorů tepelných čerpadel. Každé tepelné čerpadlo má svoji kartu.

Jelikož měřidla a textové výstupy a vstupy ukazují pouze současný stav, bylo potřeba přidat ukazatele, které budou zobrazovat hodnoty za delší časový úsek. K tomuto účelu nejlépe slouží grafy. Komponenta graf je také součástí modulu *dashboard*. Časový úsek lze nastavit časem, například poslední dva dny nebo počtem zobrazených bodů grafu. Při počtu zobrazených bodů je pak časový úsek ovlivněn jednotlivými příchody zpráv s hodnotou, která reprezentuje jeden bod grafu. Grafy tedy byly implementovány s nastavením časového úseku tři dny. Nicméně následující den bylo uživatelské rozhraní výrazně pomalé. Další den nefunkční. Grafy tedy musely být upraveny. Při upravení nastavení se již nahraná data smažou, což není ideální. Při průzkumu slabého místa bylo zjištěno, že problém nastává v při načítání velkého počtu grafických prvků, tedy i bodů grafu, na straně webového prohlížeče klienta. Heuristickou metodou bylo zjištěno, že pro ještě rozumnou rychlost může mít graf maximálně tisíc bodů na jedné kartě. Problém se projevuje při načítání karty, ať už počátečním nebo při přepnutí. Jedná se tedy o součet zátěže všech grafických prvků na jedné kartě. Tím že se jedná o součet, tak je možné zátěž dělit, například dva grafy po 500 bodů, čtyři po 250 bodů a podobně. S tímto řešením navíc není možné se dívat dál do historie, případně z něj vybírat data. Bylo tedy rozhodnuto nasadit InfluxDB a Grafanu.

5.3 InfluxDB a Grafana

Databáze InfluxDB je spuštěná také ve virtualizovaném prostředí Docker, nebo-li v kontejneru, z oficiálního obrazu. Nastavení a data jsou, podobně jako Node-RED, vytažena ven do adresáře hosta. InfluxDB je v současné době v přechodu mezi verzemi 1.6, 1.8 a 2.0. Pro komunikaci je použit starší protokol 1.x, který je kompatibilní i s novými verzemi. V Node-RED probíhá komunikace pomocí modulu `node-red-contrib-influxdb`. V konektoru je nastaveno připojení na `localhost` a název databáze `kotelna`. Každý uzel (node) má v parametru `measurement` zadán název tabulky, do které je hodnota ukládána. Ve výchozím stavu je název sloupce (atribut) nazván `value`. Ke každému řádku je vždy uložena časová značka. Název sloupce lze ovlivnit v názvem vlastnosti objektu ve zprávě. Při vytvoření uzlu s parametrem „`measurement`“ je potřeba dbát na datový typ příchozích zpráv. Tabulka v databázi se založí s příchodem první zprávy. Podle datového typu hodnoty ve zprávě se založí i datový typ v tabulce. Například číslo v datovém typu textového řetězce. Problém nastane při čtení dat Grafanou, případně dojde k chybě při matematických operacích. Při změně datového typu ve zprávě však dojde k chybě při ukládání do databáze. Bylo tedy nutné tabulku smazat a založit novou.

Grafana je taktéž spuštěna ve virtualizovaném prostředí Docker. Nastavení a data rovněž uložená v adresáři hosta. Grafana se připojuje na databázi InfluxDB. Z ní vybírá data pomocí SQL dotazu a vykresluje do grafu. Grafy jsou uspořádány a zobrazovány na takzvaném „`dashboardu`“. Je vytvořeno šest `dashboardů`. Tři zobrazují stavy a hodnoty senzorů z tepelných čerpadel, každý pro jedno čerpadlo. Lze na něm kontrolovat správný chod tepelného čerpadla. Dva `dashboards` zobrazují hodnoty ze senzorů v kotelně. Jeden je určen pro režim chlazení a druhý pro režim topení. Zobrazují v podstatě ty stejné hodnoty. Rozdíl

je v jiném grafickém uspořádání. Je to z důvodu lepší vizualizace dat. Poslední dashboard zobrazuje teplotu v jednotlivých místnostech. Současně s teplotou je zobrazen také stav otevření nebo zavření ventilu. Grafana rovněž umožňuje zobrazovat data za zvolené časové období. Dále pak obsahuje grafický editor SQL dotazu.

5.4 Node-RED osvědčené postupy

Tvorba logiky v Node-RED vyžaduje přemýšlet trochu odlišně. Hlavní koncept je předávání mezi uzly (node). Uzly jsou propojeny do toku (flow). Zprávy tedy procházejí tokem a jsou zpracovávány jednotlivými uzly. Zprávy mohou mít libovolný obsah. Standardně se očekává objekt `msg` s vlastností `payload`, ve které se nachází obsah zprávy. Velice používaná vlastnost je také `topic`. V něm je obsažena informace o jakou zprávu se jedná, obdobně jako v protokolu MQTT. Běžné uzly provádí činnost podle svého určení. Pro naprogramování vlastní činnosti slouží uzel `function`. Do něj lze napsat kód v Javascriptu. Kód je spuštěn s příchodem každé zprávy. Standardně se tedy program píše jako obsluha události na příchod zprávy. Potud je programování relativně jednoduché, navíc v grafické podobě uzlů i přehledné. V případě, kdy je potřeba se rozhodovat i na základě předcházejících zpráv je potřeba si informaci uložit. Například termostat dostává na vstup zprávu s hodnotou aktuální teploty místnosti. S příchodem zprávy je spuštěn vnitřní program. Ten přečte teplotu a podle naprogramované logiky a nastavených teplot vytvoří zprávu pro ventil a pošle na výstup. Nastavené teploty nebo-li vnitřní stav je potřeba uložit. K ukládání vnitřních stavu slouží `Context storage`. `Context storage` rozlišuje tři úrovně platnosti proměnné (variable scope).

- Node – Proměnná je platná pouze v konkrétním uzlu (node).
- Flow – Proměnná je platná pro celý tok (flow).
- Global – Proměnná je platná v celém Node-RED.

Platnost proměnné Node je nejjednodušší pro přehlednost. Není potřeba přemýšlet nad ostatními uzly, stačí se soustředit pouze na program v jednom uzlu. V jiném uzlu může klidně být proměnná se stejným názvem. Čtení proměnné je vhodné provádět ihned na začátku přesně takto:

```
var nazev_promenne = context.get('nazev_promenne') || <výchozí hodnota>;
```

Pro případ kdy proměnná ještě neexistuje je využit operátor logický operátor OR, za kterým následuje hodnota, kterou se proměnná inicializuje. Uložení proměnné probíhá příkazem `context.set('nazev_promenne', nazev_promenne);`.

Platnost Flow se vztahuje pro všechny uzly v daném toku. Používá se obdobně jako výše, akorát s použitím metod `flow.get` a `flow.set`. Platnost Global se vztahuje na uzly v celé instanci Node-RED. Používá se metodami `global.get` a `global.set`.

`Context storage`, tak jak je výše popsán, ukládá data pouze do operační paměti. Při restartu Node-RED dojde ke smazání všech proměnných. Možností je použít standardní soubor. Výchozí uzly pro práci se soubory však nejsou příliš vhodné. Při tvorbě ukládání hodnot z termostatu do souboru jsem narazil na problém s vyprazdňováním mezipaměti. Postup byl následující. Načíst soubor se starými hodnotami. Upravit konkrétní hodnotu. Uložit celý soubor s pozměněnou hodnotou. Jelikož zde chybí funkce `flush`, která propíše změny na disk, dojde k poruše konzistence souboru, což rozbije systém. Nepomůže tomu ani nechat víc času, chová se to nespolehlivě.

Řešením je opět použit Context storage, akorát s jiným modulem. Konkrétně je potřeba upravit konfiguraci v souboru `settings.js` podle 5.1.

```
contextStorage: {
  default: "memoryOnly",
  memoryOnly: { module: 'memory' },
  file: { module: 'localfilesystem' }
},
```

Výpis 5.1: Nastavení Context storage.

Na výpisu 5.1 je vidět, že ve výchozím stavu se použije profil `memoryOnly`. Uložení proměnné profilem `file` lze provést příkazem

```
flow.set('nazev_promenne', nazev_promenne, 'file');
```

Načtení se provádí obdobně příkazem

```
flow.get('nazev_promenne', 'file') || <výchozí hodnota>;
```

Pro profil `file` nelze použít úroveň platnosti Node. Je tedy nutné dávat pozor na stejné názvy proměnných v uzlech ve stejném toku.

Ukládání na disk neprobíhá ihned. Probíhá v intervalech po 30 sekundách. Je to z důvodu omezení počtu zápisů. Například pokud Node-RED běží na RaspberryPi, kde je pro ukládání použita SD karta.

Podobně jako v běžném programování je doporučeno v co největší míře se vyhýbat globálním proměnným, tak v Node-RED to platí dvojnásob. Při použití globálních proměnných pro komunikaci mezi uzly dojde poměrně rychle k dezorientaci v činnosti. V extrémním případě nemusí být propojeny žádné uzly typu `function`. Osobně doporučuji používat pouze úroveň Node pro proměnné, co nemusí přežít restart. K tomu úroveň Flow pro proměnné ukládané na disk s tím, že je nutné dávat pozor na název.

Další z možných problémů je tvorba neřízeného cyklu zpráv. Cyklus lze v toku poznat relativně snadno. Složitější pro odhalení je cyklus přes jiné zařízení, to již v toku není přímo vidět. V případě cyklení zpráv přes jiné zařízení lze většinou cyklus rozpojit. V případě lokálního cyklu je vytížení tak vysoké, že webové rozhraní zatuhne. Zbývá pouze restart celého Node-RED. Ovšem propojení uzlů grafických ovládacích prvků s uzly `function` je často zapojeno v cyklu. Například posuvník nastavuje teplotu termostatu. Proměnná je v uzlu `function`. Používá Context storage s profilem `file`, aby nastavená hodnota přežila restart. Po restartu je potřeba hodnotu do grafického prvku poslat aby došlo k zobrazení na panelu. K tomu je využit uzel `inject` s nastavením `inject once`. Tahle konstrukce svádí k řešení aktualizovat grafický prvek při jakékoli změně proměnné. Uzel grafického prvku, ve výchozím nastavení, vzniku cyklu zabraňuje. Zábрана funguje následovně. Pokud dojde na vstup zpráva se stejnou hodnotou, jakou má grafický prvek již nastavenou, tak ji na výstup nepošle.

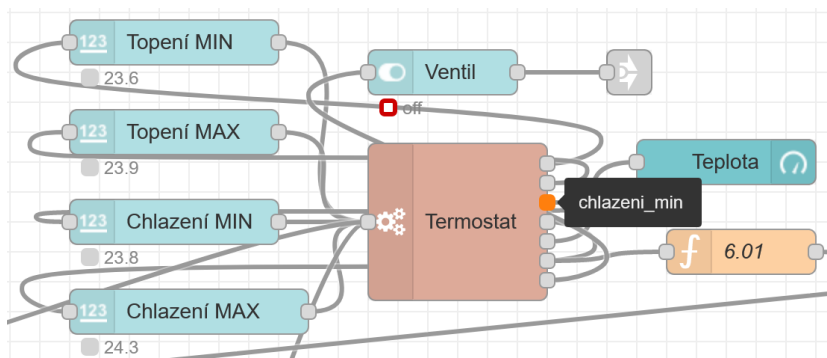
V případě zmíněné primitivní implementace probíhají zprávy následovně. Uživatel použije grafický prvek, z jeho uzlu vypadne zpráva s novou hodnotou, která dojde na vstup uzlu `function`. Ten aktualizuje proměnnou a pošle tuto hodnotu do grafického prvku. Ten ji již má nastavenou, tudíž další zprávu nepošle. V případě aktualizace proměnné zprávou z jiného uzlu, odešle uzel `function` zprávu s aktualizací do grafického uzlu. Grafický prvek se aktualizuje, změní se, tudíž pošle tuto hodnotu na výstup. To vyvolá další aktualizaci ovšem již na stejnou hodnotu, kterou grafický uzel znovu neodešle.

Tato primitivní implementace vypadá funkčně a je i funkční do doby, než uživatel v uživatelském rozhraní začne zběsile klikat. Případně použijete prvek `numeric input`, který ke zběsilému klikání přímo vybízí. Než zpráva s aktualizací nestihne projít celý cyklus, vznikne zpráva s další aktualizací. Stará aktualizace dorazí do grafického uzlu, kde způsobí aktuali-

zaci na starou hodnotu. Grafický uzel pak znovu pošle zprávu s aktualizací. Tím je lokální cyklus uzavřen a musí proběhnout restart systému.

Správné řešení je programově oddělit aktualizací zprávy z grafického uzlu. Hodnotu ze zprávy z grafického uzlu uložit pouze do proměnné a neposílat ji znovu do grafického uzlu. Hodnota z jiného zdroje nejprve aktualizuje proměnnou. Poté odešle hodnotu do grafického prvku. Grafický prvek zašle hodnotu zpět, proměnná se nastaví znovu a však již na stejnou hodnotu a cyklus nevznikne.

Pro přehlednost je praktické v uzlu `function` vytvořit separátní výstupy pro jednotlivé grafické uzly. Výstupy je samozřejmě vhodné pojmenovat, například jako na obrázku 5.1.



Obrázek 5.1: Termostat pro jednu místnost

Kapitola 6

Testování a možnosti rozšíření

System byl vyvíjen a testován rovnou na produkčním prostředí. Bloky programu v Node-RED jsou relativně krátké. V případě nejasností probíhal test pomocí inject a debug uzlu. Inject uzel dokáže nasimulovat příchod jakékoli zprávy. Debug uzel dopodrobna vypíše obsah zprávy. Pokud uzel přijal zprávy bez chyby a správně reagoval, tak byl nasazen do provozu. V provozu se několik chyb projevilo. Drtivá většina byla způsobena nepozorností při psaní nebo kopírování v kódu, například špatný název proměnné, špatná podmínka. Obtížnější chyby nastávaly v uzlech s delším kódem. Přírůstková metoda vývoje postupně přidává funkcionalitu. Tím během vývoje některé uzly začnou být příliš složité. Složité uzly je obtížné trasovat. Hledání chyby je poté příliš náročné. Bylo nutné funkcionalitu znovu promyslet a dekomponovat na uzly tak, aby byly samy o sobě jednoduché a tím i jednoduše ověřitelné. System byl rovněž otestován restartem za plného běhu. System úspěšně naběhne a obnoví svou funkcionalitu včetně poslední konfigurace.

Částečná grafická podoba programovacího rozhraní umožňuje ostatním se lépe zorientovat a případně doprogramovat další funkcionalitu. Oproti dříve plánovanému softwaru Mervis je Node-RED více uživatelský přívětivější. Další možnosti rozšiřování jsou limitovány platformou Node-RED, případně komunikačními protokoly. Vzhledem k současnému stavu a výběru přídavných modulů jsou možnosti prakticky neomezené.

Možnosti vylepšení se nabízí v zabezpečení. Nejprve by bylo vhodné zapnout ověřování MQTT komunikace mezi brokerem a připojenými prvky, případně rovnou šifrování. Dále pak povolit přístup do InfluxDB pouze lokálními adresám. Přístup k hlavnímu uživatelskému panelu lze v současnosti nastavit pouze jedno heslo. Do budoucna by bylo vhodné umožnit přístup pomocí více uživatelských hesel.

Kapitola 7

Závěr

Cílem této práce bylo pro řízení a obsluhu kotelny pomocí jednotného ovládacího panelu. Navržené řešení je nasazeno v reálném provozu ve společnosti SPOLEČNOST-24 s.r.o. K dosažení výsledku bylo nutné nastudovat a pochopit technické možnosti již existující techniky a postupně na ni navázat. Byl proveden fyzický průzkum a analýza současného řešení. Výsledkem bylo nejen schéma zapojení přístrojů v kotelně, ale i pochopení původních záměrů pro optimální energetické využití zdrojů. Následně byly prozkoumány komunikační možnosti řídicích prvků. Každá zapojená řídicí jednotka měla své uživatelské rozhraní. Na základě výsledků byl zvolen koncentrační softwarový nástroj Node-RED. Dále byl vytvořen návrh komunikace. Komunikace je převážně založena na protokolu MQTT. Jednotlivé prvky mezi sebou komunikují pomocí odebírání a zasílání zpráv do jednotlivých kanálů. Výsledkem je návrh komunikační struktury v jednotlivých ovládacích prvcích a propojení s řídicím systémem.

Bylo navrženo uživatelsky přívětivé webové rozhraní nad koncentrátorem Node-RED, které umožňuje údržbáři společnosti přepínat mezi předdefinovanými režimy a bylo uzpůsobeno k snadnému ovládní a rychlému přístupu k nejpodstatnějším funkcím. Mezi funkce systému patří řízení stavu trojcestných ventilů tepelných čerpadel vzduchotechniky a monitorování teplot ze senzorů v jednotlivých místnostech. Byly navrženy následující předdefinované režimy: topení, chlazení, vypnuto. Ve všech uvedených režimech probíhá ohřev teplé užitkové vody tepelným čerpadlem, funkcionality režimů spočívá v práci s oběhovou vodou (chlazení nebo topení). Údržbář může nastavovat teploty na úrovni jednotlivých místností, včetně ukládání a nahrávání přednastavených hodnot. Hystereze zónové regulace je na úrovni 0,1 °C.

Celý systém je schopen ošetřit výjimky integrovaného řízení tepelných čerpadel a zotavit jejich funkci z chybových nebo mezních stavů. Systém byl navržen taktéž aby optimalizoval rozložení zátěže mezi jednotlivými tepelnými čerpadly, které se nejvíce opotřebovávají při startu kompresoru.

Systém byl ověřen v praxi a již více než půl roku funguje v produkci ve prostorách společnosti. Řešení bylo navrženo tak aby bylo pro firmu ekonomické. Ekonomičnost spočívá ve výpočtu COP indexu a systém se rozhoduje kdy je výhodnější použít topnou spirálu či tepelné čerpadlo. Systém byl po celou dobu provozu, krom plánovaných servisních zásahů spolehlivě i stabilně v provozu bez výpadku.

Literatura

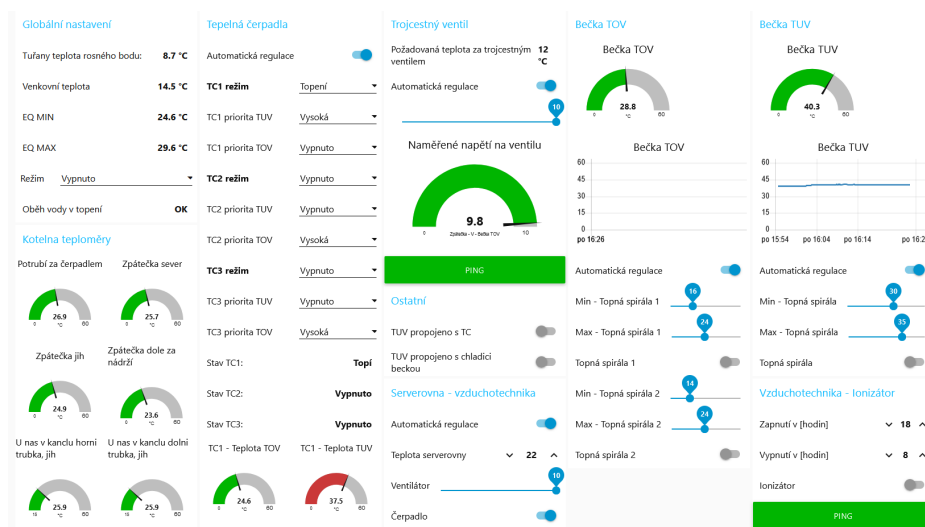
- [1] ALFILLE, P. *Owserver MAN page* [online]. 2003 [cit. 2020-10-15]. Dostupné z: <https://owfs.org/uploads/owserver.html>.
- [2] ALFILLE, P. H. a TEAM, T. owfs. *Complete 1-wire command codes* [online]. 2003 [cit. 2021-01-21]. Dostupné z: <http://owfs.sourceforge.net/commands.html>.
- [3] BAN. *Na Opavsku se dají sehnat uhelné kaly: Zakázaným, ale levným palivem pak lidé topí doma* [online]. 2016 [cit. 2020-10-19]. Dostupné z: <https://ct24.ceskatelevize.cz/regiony/1980075-na-opavsku-se-daji-sehnat-uhelne-kaly-zakazanym-ale-levnym-palivem-pak-lide-topi>.
- [4] BANKS, A. a GUPTA, R. *MQTT Version 3.1.1* [online]. 2014 [cit. 2021-01-27]. Dostupné z: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>.
- [5] BANKS, A., GUPTA, R., BRIGGS, E. a BORGENDALE, K. *MQTT Version 5.0* [online]. 2019 [cit. 2021-01-27]. Dostupné z: <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#AppendixC>.
- [6] CERPADEL.CZ abeceda. *Správné zapojení tepelného čerpadla* [online]. 2010 [cit. 2021-01-14]. Dostupné z: <https://www.abeceda-cerpadel.cz/cz/spravne-zapojeni>.
- [7] COMMUNITY, T. kernel development. *Introduction to the 1-wire (w1) subsystem* [online]. 2009 [cit. 2021-01-22]. Dostupné z: <https://www.kernel.org/doc/html/latest/w1/w1-generic.html>.
- [8] DAN AWTRY, D. L. *SPRINGBOK DIGITRONICS Innovative Hardware and Software for the modern world Understanding 1-Wire Series* [online]. 2004 [cit. 2021-01-20]. Dostupné z: https://dutta.csc.ncsu.edu/csc453_spring16/wrap/1-Wire-Design%20Guide%20v1.0.pdf.
- [9] EASTLAKE, D. *IEEE 802 Numbers* [online]. 2019 [cit. 2021-01-27]. Dostupné z: <https://www.iana.org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml#ieee-802-numbers-1>.
- [10] ESBE. *Směšovací ventily řada VRG130* [online]. 2018 [cit. 2021-01-18]. Dostupné z: https://www.bola.cz/admin/files/e_product_files/0/449/VRG131_132_133_21.pdf.
- [11] HINK, T. *Návrh optimalizace a monitoringu infrastruktury serverovny podniku*. Brno, CZ, 2019. Diplomová práce. Vysoké učení technické v Brně Fakulta podnikatelská. Dostupné z: <https://dspace.vutbr.cz/bitstream/handle/11012/177969/final-thesis.pdf>.

- [12] HINK, T. *Trends in Robotic Process Automation*. 1. vyd. Vysoké učení technické v Brně Fakulta podnikatelská, 2020. 60 s. ISBN 978-80-214-5933-5.
- [13] INFO.CZ izolace. *Rosný bod a kondenzace vodních par ve zdivu* [online]. 2013 [cit. 2021-01-14]. Dostupné z: <https://www.izolace-info.cz/aktuality/9208-rosny-bod-a-kondenzace-vodnich-par-ve-zdivu-a.html>.
- [14] ING. VÁCLAV MATZ, P. *Ekvitermní regulace – princip a využití v systémech regulace vytápění* [online]. 2010 [cit. 2021-01-14]. Dostupné z: <https://vytapani.tzb-info.cz/mereni-a-regulace/6294-ekvitermni-regulace-princip-a-vyuziti-v-systemech-regulace-vytapani>.
- [15] INTEGRATED, M. *1-Wire Search Algorithm* [online]. 2002 [cit. 2021-01-20]. Dostupné z: <https://www.maximintegrated.com/en/design/technical-documents/app-notes/1/187.html>.
- [16] INTEGRATED, M. *DS2438 Smart Battery Monitor* [online]. 2005 [cit. 2021-01-21]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS2438.pdf>.
- [17] INTEGRATED, M. *Guidelines for Reliable Long Line 1-Wire Networks* [online]. 2008 [cit. 2021-01-20]. Dostupné z: <https://www.maximintegrated.com/en/design/technical-documents/tutorials/1/148.html>.
- [18] INTEGRATED, M. *8-Channel 1-Wire Master* [online]. 2014 [cit. 2020-10-07]. Dostupné z: <https://datasheets.maximintegrated.com/en/ds/DS2482-800.pdf>.
- [19] KARLÍK, R. *Tepelné čerpadlo pro váš dům*. 1. vyd. Grada Publishing a.s., 2009. ISBN 978-80-247-2720-2.
- [20] KLÍMA, M. *Tepelné čerpadlo*. Brno, CZ, 2010. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství. Dostupné z: <https://dspace.vutbr.cz/bitstream/handle/11012/2331/final-thesis.pdf>.
- [21] MALÝ, M. *Sběrnice 1-WireTM* [online]. 2004 [cit. 2021-01-20]. Dostupné z: <https://vyvoj.hw.cz/navrh-obvodu/rozhrani/sbernice-1-wiretm.html>.
- [22] MALÝ, M. *Protokol MQTT: komunikační standard pro IoT* [online]. 2016 [cit. 2021-01-27]. Dostupné z: <https://www.root.cz/clanky/protokol-mqtt-komunikacni-standard-pro-iot/>.
- [23] MANN, A., BÜRCEL, C. M. a GROCHE, P. *A modeling strategy for predicting the properties of paraffin wax actuators*. 2018.
- [24] MAPLESOFT. *Calculating the Dew-Point and Wet-Bulb Temperature of Air* [online]. 2016 [cit. 2021-01-10]. Dostupné z: <https://www.maplesoft.com/applications/view.aspx?SID=154054&view=html>.
- [25] OEZ. *OEZ Minia modulární přístroje jističe LTE* [online]. 2019 [cit. 2021-01-19]. Dostupné z: http://www.oez.cz/uploads/oez/files/ks/lte-mi01-2019_cz_sk.pdf.
- [26] OEZ. *OEZ Minia modulární přístroje stykače RSI* [online]. 2019 [cit. 2021-01-19]. Dostupné z: http://www.oez.cz/uploads/oez/files/ks/rsi-mi01-2019_cz_sk.pdf.

- [27] OWFS. *Slaves by Family Code* [online]. 2020 [cit. 2021-01-20]. Dostupné z: https://www.owfs.org/index_php_page_family-code-list.html.
- [28] PEŠKA, R. *Použití platformy Node-Red je na vzestupu* [online]. 2019 [cit. 2021-01-28]. Dostupné z: <https://byznys.hw.cz/pouziti-platformy-node-red-je-na-vzestupu.html>.
- [29] PLUS, E. *Mervis* [online]. 2019 [cit. 2021-01-28]. Dostupné z: <https://mervis.info/>.
- [30] POJAR, P. *Topení, ohřev vody, 23. díl seriálu JAK POSTAVIT DŮM krok po kroku* [online]. 2016 [cit. 2021-01-19]. Dostupné z: <https://www.ceskestavby.cz/jak-se-stavi-dum/vytapeni-topeni-ohrev-5451.html>.
- [31] SEDTRONIC, I. P. S. *UNICA 1-wire typ U1WTVS* [online]. 2015 [cit. 2021-01-21]. Dostupné z: <https://sedtronic.eu/cs/unica-1wire/7-28-unica-1-wire-senzorovy-modul-typ-u1wtvs.html>.
- [32] SMOLKA, V. *Rosný bod* [online]. 2015 [cit. 2021-01-14]. Dostupné z: <https://www.in-pocasi.cz/clanky/teorie/rosny-bod-20.2.2015/>.
- [33] SPURGEON, C. E. *Ethernet: The Definitive Guide: The Definitive Guide*. 1. vyd. O'Reilly Media, Inc., 2000. ISBN 9780596552824.
- [34] STAFFORD, B. *Insertion of 802.1Q tag in an Ethernet frame* [online]. 2013 [cit. 2021-01-26]. Dostupné z: https://en.wikipedia.org/wiki/IEEE_802.1Q#/media/File:Ethernet_802.1Q_Insert.svg.
- [35] STEHLÍK, P. *Internetový termostat: stavba sítě teplotních čidel na 1-Wire sběrnici* [online]. 2015 [cit. 2021-01-22]. Dostupné z: <https://www.root.cz/clanky/internetovy-termostat-stavba-site-teplotnich-cidel-na-1-wire-sbernici/>.
- [36] TECHNOLOGY, U. *1-Wire network* [online]. 2016 [cit. 2021-01-20]. Dostupné z: <https://www.unipi.technology/products/1-wire-network-46>.
- [37] UNIPI. *Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys)* [online]. 2020 [cit. 2021-01-29]. Dostupné z: <https://www.unipi.technology/cs/scada-obchodni-podminky>.
- [38] UNIPI.TECHNOLOGY. *Unipi Neuron* [online]. 2016 [cit. 2021-02-04]. Dostupné z: <https://www.abeceda-cerpadel.cz/cz/spravne-zapojeni>.
- [39] VOJÁČEK, A. *Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys)* [online]. 2011 [cit. 2021-01-28]. Dostupné z: <https://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-611313-codesys>.
- [40] WISDOM. *TEA-500 Thermoelectric Actuator* [online]. 2015 [cit. 2020-11-12]. Dostupné z: <https://en.wisdom-thermostats.com/product/thermoelectric-actuator-tea50>.
- [41] X, E. *Introduction to MQTT QoS* [online]. 2019 [cit. 2021-01-27]. Dostupné z: <https://medium.com/@emqtt/introduction-to-mqtt-5-0-protocol-qos-quality-of-service-e6d9b0aaf9fb>.
- [42] ŠTECH, K. *Elektroinstalace doma a na chatě - 3., zcela přepracované vydání*. 3. vyd. Grada Publishing a.s., 2008. ISBN 802472622X, 9788024726229.

Příloha A

Webové rozhraní – nastavení



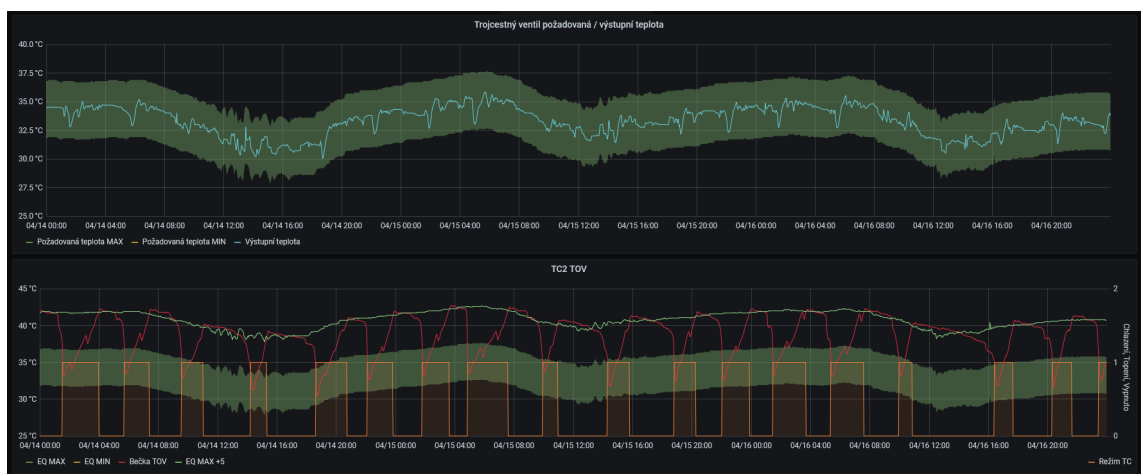
Obrázek A.1: Webové rozhraní nastavení parametrů kotelny



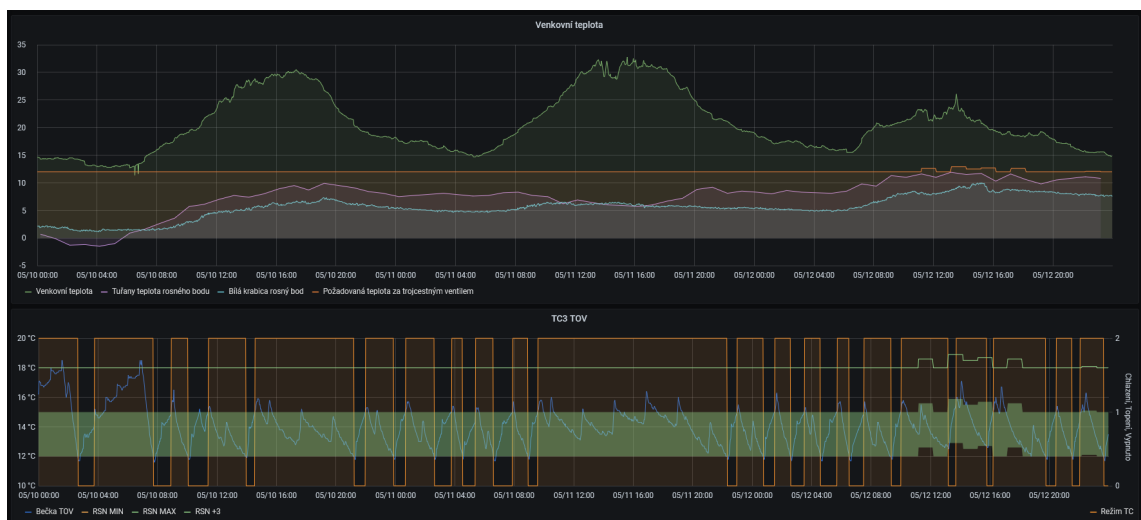
Obrázek A.2: Webové rozhraní nastavení termostátů

Příloha B

Webové rozhraní – diagnostika



Obrázek B.1: Webové rozhraní diagnostika topení



Obrázek B.2: Webové rozhraní diagnostika chlazení