Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Decentralizovaný IoT systém topení

Diplomová práce

Autor: Bc. David Česák
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Hradec Králové srpen 2024

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 6.8.2024 Bc. David Česák

Poděkování:

Děkuji vedoucí diplomové práce Mgr. Daniele Ponce, Ph.D. za metodické vedení práce a odborné rady.

Anotace

Tato diplomová práce popisuje návrh, sestrojení, vývoj a realizaci systému pro decentralizované vytápění libovolné bytové jednotky. Decentralizací je myšlen takový přístup k návrhu architektury, že v systému nebude žádný centrální prvek, který by při výpadku vyřadil veškerou funkcionalitu. Tímto přístupem je zajištěno, že systém bude odolný vůči několika druhům výpadku. Mezi tyto druhy lze zařadit výpadek nebo úplná nedostupnost internetu případně výpadek jednoho nebo více zařízení. V systému byly navrhnuty a sestrojeny dva druhy zařízení, měřicí a spínací. Zároveň byla vyvinuta i vlastní aplikace, kterou lze nasadit do serveru přístupného z internetu. Tím pádem se zajistí možnost nastavit teplotu i z libovolného místa, kde je dostupný internet. Zařízení jsou složená z běžně dostupných součástek a jsou použity volně dostupné nástroje pro jejich vývoj. Funkčnost systému byla demonstrována na reálném nasazení, kde se prokázala funkčnost systému. Zároveň byly zjištěny nedostatky, které mohou být námětem pro další závěrečné práce.

Klíčová slova: decentralizace, IoT, měření, systém, vytápění

Annotation

Title: Decentralized IoT heating system

This thesis describes the design, construction, development and implementation of a system for decentralized heating of any housing unit. Decentralization means an approach to architectural design such that there will be no central element in the system that would disable all functionality in the event of a failure. This approach ensures that the system is resilient to several types of failure. Among these types can be included an outage or complete unavailability of the Internet or the failure of one or more devices. In the system, two types of devices, measuring and switching, have been designed and constructed. At the same time, a custom application has been developed that can be deployed in a server accessible from the Internet. Thus, it is possible to set the temperature from any place where the Internet is available. The devices are composed of commonly available components and freely available development tools are used. The functionality of the system has been demonstrated and proven in a real deployment. At the same time, shortcomings were identified which can be the subject of further thesis work.

Keywords: decentralization, IoT, metering, system, heating

Obsah

1 Úvod ... 1

2 Cíl práce ... 3

3 Metodika zpracování ... 4

4 Decentralizované systémy v kontextu vytápění ... 5

4.1 Vytápění obecně ... 5

4.1.1 Dělení vytápěcích systémů ... 5

4.1.2 Modelový příklad ... 6

4.2 Decentralizované systémy a distribuované výpočty ... 6

4.2.1 Rozdíl mezi centralizovaným a decentralizovaným systémem ... 7

4.2.2 Příklady decentralizovaných systémů ... 7

4.3 Existující chytrá řešení ... 8

5 Identifikace požadavků pro decentralizovaný systém topení ... 10

5.1 Požadavky na systém ... 10

5.2 Požadavky na prvky systému ... 11

5.2.1 Požadavky na měřicí zařízení ... 12

5.2.2 Požadavky na spínací zařízení ... 14

5.2.3 Požadavky na cloud ... 16

6 Charakterizace dílčích prvků systému podle požadavků ... 17

6.1 Části systému (podsystémy) ... 17

6.1.1 Společné části měřicího a spínacího zařízení ... 17

6.1.2 Měřicí zařízení ... 20

6.1.3 Spínací zařízení ... 21

6.1.4 Cloud ... 23

7 Návrh technického řešení ... 24

7.1 Systém topení ... 24

7.2 Společný základ pro zařízení 26
7.2.1 Porovnání podporovaných platforem pro ESP8266EX 26
7.3 Výběr rozdělení firmware 28
7.3.1 Společný firmware 29
7.3.2 Rozdělený firmware 29
7.3.3 Shrnutí 29
7.4 Komunikace v systému 29
8 Postup realizace navrženého technického řešení 32
8.1 Navržený systém topení 32
8.2 Výroba zařízení 33
8.3 Vývoj firmware pro měřicí a spínací zařízení 34
8.3.1 Popis složek ve zdrojovém kódu firmware 38
8.3.2 Složky projektu společného firmware 38
8.4 Realizace webového rozhraní 41
8.5 Realizace cloudové části systému 45
9 Demonstrace funkčnosti systému 48
9.1 Základní parametry demonstrace systému 48
9.2 Test připojení nového zařízení 49
9.3 Test lokální synchronizace zařízení 51
9.4 Test nastavení konfigurace v cloudu 52
9.5 Test synchronizace konfigurace s cloudem 54
9.6 Test funkčnosti topení 56
9.7 Test systému bez připojení k internetu 57
9.8 Test odpojení jednoho měřicího modulu 59
9.9 Shrnutí všech testů 60
9.10 Zopakování požadavků a jejich splnění 60

10 Závěr 64

11 Shrnutí a doporučení 65

12 Seznam použité literatury 66

13 Přílohy 70

1 Úvod

Vytápění obytných prostor je klíčovou součástí moderního bydlení a s rozvojem technologií přicházejí stále nová chytrá řešení, která přinášejí nejen větší pohodlí, ale i potenciální úspory energie. Pokud bychom vůbec neregulovali teplotu, s velmi vysokou pravděpodobností by došlo ke snížení kvality ve všech směrech. Ať už by to bylo sníženým komfortem, zhoršenou produktivitou, častějším nemocem, zmrzlým přívodem vody, ale i hygienou. To souvisí i s tím, že v měsících s vyšší vlhkostí by došlo i ke vzniku plísní, což je naprosto nežádoucí.

Chytrá zařízení jako jsou termostaty a další IoT (Internet of Things) zařízení, nabízejí pokročilé funkce. Klasická jednoduchá zařízení tyto funkce nabídnout nemohou. Nemají totiž přístup k rozšířeným informacím. Mezi hlavní výhody těchto chytrých řešení patří efektivita a úspora energie. Hlavními důvody využití chytrých termostatů jsou možnost optimalizovat vytápění na základě denní doby, předpovědi počasí a přítomnosti osob v domě. To vede k nižší spotřebě energie. Dále nabízejí zvýšený komfort, neboť možnost vzdáleného ovládání pomocí aplikací na mobilních zařízeních umožňuje regulovat vytápění odkudkoli.

Na druhou stranu tato chytrá řešení přinášejí i určitá rizika a nevýhody. Mnohá chytrá zařízení vyžadují trvalé připojení k internetu, bez kterého mohou ztratit část nebo i úplnou funkcionalitu. Což by v krajním případě znamenalo, že se všechny ovládané místnosti stanou neobyvatelnými. Připojení do cizího cloudu znamená, že data jsou odesílána na servery třetích stran, což může představovat riziko pro bezpečnost a soukromí uživatelů. Zvláště pokud není jasně specifikováno, jaká data jsou shromažďována a jak jsou využívána. Uživatelé zároveň mají často omezenou kontrolu nad svými zařízeními a k tomu jsou závislí na výrobcích, pokud jde o aktualizace, opravy a další podporu.

Existují zde však alternativní řešení, která mohou tyto nevýhody odstranit nebo alespoň částečně omezit. Jedním z nich je možnost lokální kontroly nad zařízeními bez nutnosti odesílat data na vzdálené servery. Tato řešení často využívají lokální servery, přímo integrovanou logiku v zařízení nebo vlastní servery umístěné v internetu.

Pro uživatele je důležité mít přehled o tom, jaká data jsou jejich zařízeními shromažďována a jak jsou tato data využívána. Zabezpečení by mělo být prioritou, což zahrnuje pravidelné aktualizace firmware a zabezpečené připojení. Při výběru chytrých zařízení je důležité zvážit nejen funkce a cenu, ale také aspekty bezpečnosti a soukromí.

2 Cíl práce

Cílem práce je návrh a realizace decentralizovaného systému topení alespoň pro tři místnosti. Decentralizace zajistí, že při výpadku jednoho prvku bude zbytek systému plně funkční i při nedostupnosti internetu. Systém bude složen ze dvou typů zařízení: měřicí a spínací. Zařízení si budou mezi sebou synchronizovat konfiguraci v rámci lokální wifi sítě, aby z libovolného místa bylo možné systém plně nakonfigurovat. Návrh a výroba zařízení je součástí práce.

3 Metodika zpracování

Metodika této práce se skládá z identifikace požadavků, následné charakterizace dílčích prvků, návrhu systému, jeho realizace a ověření funkčnosti za pomocí demonstrace.

4 Decentralizované systémy v kontextu vytápění

Tato kapitola popisuje různorodé teoretické předpoklady, ze kterých vychází návrh práce.

4.1 Vytápění obecně

Topení, jak již bylo zmíněno v úvodu, je velice podstatnou a nedílnou součástí našich každodenních životů ať si to uvědomujeme či nikoliv. Je důležité zvážit, jaký systém vytápění se v obydlí nachází a tím pádem, jak se ovlivňuje možnost jeho ovládání ať už manuální nebo automatizované. Z toho důvodu je důležité mít představu, jaké možnosti vytápění jsou, aby bylo možné zvážit, jak je ovládat.

4.1.1 Dělení vytápěcích systémů

Vytápěcí systémy lze rozdělit dle lokality zdroje tepla. Tím se myslí samotný zdroj tepla.

Lokální vytápění

U lokálního vytápění se používají zdroje tepla umístěné přímo v místnostech. Nejčastější využití je v obydlí s malým počtem místností nebo takové obydlí, do kterého ústřední vytápění není možné zavést. Mezi příklady lze zařadit elektrické konvektory (přímotopy), plynová topidla (např. propan-butanová) nebo kachlová kamna. [1]

Jedná se o zdroje tepla, které jsou vzájemně izolovány a často bývají omezeny na manuální ovládání pomocí ručních ventilů nebo ještě jednodušším způsobem, jako je přikládání dřeva. Výjimkou jsou zde chytré konvektory, které mají ovládání, ale často jsou limitovány proprietárním řešením připojené do proprietárního cloudu. Pokud by tato řešení měla možnost ovládání skrze programové rozhraní, dala by se integrovat do chytré domácnosti pod vlastní správou.

Ústřední vytápění

Opačným příkladem je topení ústřední. Je charakteristické tím, že zdroj tepla je na jednom místě, například kotelna. Následně z tohoto místa jsou rozvedeny

do všech dalších místností, kde je třeba regulovat teplotu. Typickým příkladem jsou radiátory v místnostech. [1] Samotným zdrojem může být například kotel. Pokud se jedná o kotel elektrický nebo plynový, bude jistě mít i automatickou vnitřní regulaci teploty. Pokud by se jednalo o kotel na tuhá paliva, tak je nutné kotel manuálně udržovat v optimální teplotě. Další možností je místo kotle mít přivedenou páru z tepelné elektrárny, tak jako je to typické u bytových domů. Výhodou těchto řešení je fakt, že rozvod tepla začíná na jednom místě, což lze využít při automatizaci.

4.1.2 Modelový příklad

K zasazení vytápění do konkrétního modulu, je zvolen rodinný dům. Tento dům je patrový a obsahuje tři místnosti, ve kterých je potřeba aktivně regulovat teplotu. Zdroj tepla je centrální s ventily umístěnými u bojleru. Ve vytápěných místnostech jsou mechanické hlavice, které lze pomocí přívodu napětí regulovat. Je to tedy vhodný kandidát na automatizaci pomocí chytrého řešení.

4.2 Decentralizované systémy a distribuované výpočty

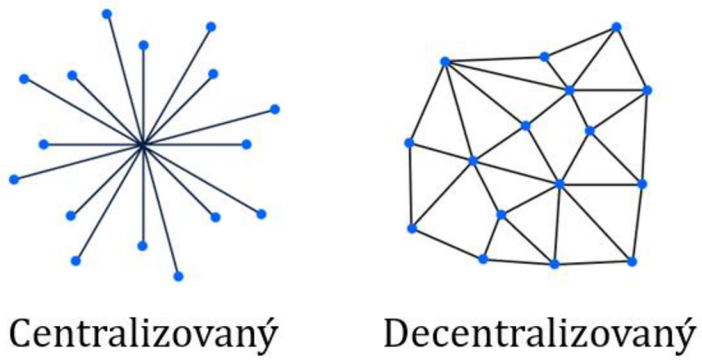
Pokud vezmeme v potaz návrh systému, nabízí se zde několik možných rozdělení. Distribuované a decentralizované přístupy jsou si podobné. Každý decentralizovaný systém je zároveň i distribuovaný. Zásadní rozdíl v decentralizovaném řešení oproti distribuovanému je takový, že v distribuovaném systému může stále být centrální prvek rozdělující úkoly.

Je dobré zmínit, že decentralizované systémy nejsou jediným řešením návrhu systémů. Existuje i přístup opačný, tedy systémy centralizované. Distribuovaný systém si lze představit jako kolekce spolupracujících jednotek, která dohromady usiluje o splnění společného cíle. Decentralizovaný systém to posune o úroveň výš a ze všech udělá samostatně fungující jednotky. Pokud bychom to vzali v případě distribuovaného a zároveň decentralizovaného topícího systému, tak společný cíl by bylo udržení správné teploty v místnostech a každé zařízení by bylo zodpovědné za svou část. Například měřením teploty, spínáním ventilů, komunikací.

Decentralizovaný systém spoléhá na rozdělení se do několika uzlů utvářející jeden velký celek. Uzly, které se v systému nacházejí, mezi sebou komunikují a synchronizují si data. Běžným příkladem může být například fyzické zařízení jako je počítač, chytrý telefon aj. V některých případech se může jednat pouze o programy běžící na totožném počítači. Mezi hlavní cíle distribuovaných systémů lze zařadit odstranění tzv. bottlenecků (úzkých hrdel), tedy míst, kde by mohlo dojít k omezení výkonu. Nejdůležitějším cílem je odstranění centrálního bodu, který by při výpadku vyřadil celý systém, což je vlastně jeho samostatná podstata. [2]

4.2.1 Rozdíl mezi centralizovaným a decentralizovaným systémem

Když vezmeme centralizovaný systém, tak jeho podstata spočívá v tom, že všechny úkoly jsou obstarávány a zpracovávány jedním uzlem. Tedy výpadek tohoto uzlu automaticky znamená, že systém selže. Což může být u některých řešení, která nejsou tak kritická, dostatečné řešení a přijatelný kompromis. Zároveň veškerá komunikace probíhá skrze tento uzel, což může mít za následek, že některá data mohou být opožděna a v horším případě i ztracena, pokud by byl uzel přehlcen. [2]



Obrázek 1 – Grafické znázornění centralizovaného a decentralizovaného systému (zdroj: [2])

4.2.2 Příklady decentralizovaných systémů

K podpoření decentralizovaných systémů je dobré zmínit několik příkladů z reálného světa, které tímto způsobem fungují. Jedním z příkladů distribuovaných

systémů jsou takzvané Torrenty, což je způsob a technika sdílení souborů. V kostce funguje tak, že daný soubor není uložený na žádném serveru, ale je distribuován mezi lidmi. Pokud bychom ho chtěli dostat do našeho počítače, tak stahování bude probíhat od všech připojených lidí, kteří mají tento soubor u sebe. Všichni, kteří soubor mají se mohou podílet na jeho šíření dál.

V dnešní době asi nejznámějším příkladem decentralizovaného systému je digitální měna Bitcoin. Bitcoin je založen na blockchainu, což je databáze distribuována mezi všemi uživateli. V této databázi jsou uloženy veškeré transakce, které se odehrály. Co tuto síť dělá decentralizovanou je, že pro všechny uzly v síti platí stejná pravidla. Zároveň se může kdokoliv připojit i odpojit. Aby byl do blockchainu přidán záznam, tedy nedávné transakce, musí s ním souhlasit většina sítě. [3]

4.3 Existující chytrá řešení

Vytvoření a realizace takového systému není převratný nápad a na světě existuje velké množství komerčních řešení. Když si vezmeme běžně komerčně dostupné termostaty, příkladem může být například chytrá termostatická hlavice KE100 od TP-Linku. Samostatně nefunguje a je nutné ji koupit v balení spolu se základnou. K ovládání a spárování je nutné si stáhnout mobilní aplikaci, do které je nutné se nejdříve zaregistrovat a přihlásit. Poté lze vše konfigurovat z mobilní aplikace odkudkoliv. Všechna data jsou uložena na cizím cloudu bez možnosti ho odebrat, a to ani pro lokální použití bez internetu. Úplně stejným problémem trpí i řešení od společnosti IMMAX. Tato firma nabízí také chytrou hlavici „Immax NEO Smart Termostatická hlavice Zigbee“, k ní je nutné zakoupit prostředníka, který zprostředkovává veškerou komunikaci „Immax NEO MULTI BRIDGE PRO“. Bez aplikace a připojení k internetu do cizího cloudu je řešení nepoužitelné. Při hledání dalších řešení, tentokrát od společnosti AVATTO, je přístup výrobce stejný jako u předchozích. Je tedy pravděpodobné, že pokud si uživatel bude chtít zavést do svého obydlí chytrou domácnost s vytápěním, tak si pravděpodobně vybere jedno z těchto řešení. Tato řešení jsou uživatelsky přívětivá a díky nim často uživatelé zapomínají, že mohou mít i svá rizika. Na internetu nebyla nalezena taková řešení, která by mezi své hlavní výhody a funkčnosti zařadila právě

možnost fungovat bez připojení ke cloudu nebo případně možnosti funkčnosti kompletně bez internetu.

Alternativní řešení, které má schopnost fungovat lokálně je použití Home Assistantu. Home Assistant je dnes již známý software s otevřeným zdrojovým kódem pro automatizaci domácnosti. Výhodou tohoto řešení je jeho vysoká škála podporovaných zařízení, která lze přes něj spravovat. Zároveň je možné, respektive i nutné ho provozovat na vlastním zařízení pod vlastní kontrolou, což je osvědčený postup, jak zvýšit bezpečnost takovýchto řešení. K tomu je nutné si poradit s nákupem vhodného hardware, na který je následně potřeba samotný Home Assistant nainstalovat. Po instalaci samozřejmě uživatele čeká přidání zařízení a samotné nastavování plánů. Toto řešení je vhodné pouze pro uživatele, kteří v oboru mají přehled. V základním nastavení Home Assistant nepodporuje přístup z internetu. Nejpohodlnějším řešením je Home Assistant Cloud, což je zpoplatněná služba umístěná v cizím cloudu. Bohužel tato varianta je bez možnosti vlastního nasazení. Samozřejmě lze Home Assistant zpřístupnit i pomocí VPN, ale to není pro běžného uživatele jednoduchý úkol. Tedy ačkoliv Home Assistant je výborný nástroj pro tvorbu vlastní chytré domácnosti, je zároveň i technicky náročný na nastavení a správu. Není tedy vhodný pro každého.

5 Identifikace požadavků pro decentralizovaný systém topení

Decentralizovaný systém topení se bude skládat ze tří dílčích částí. Jedna bude samostatná, odlišná – synchronizační cloud a dvě budou podobné. Konkrétně se bude jednat o fyzická zařízení. Každé bude plnit svou roli v systému. Pro zjednodušení orientace v požadavcích je zavedena jmenovací konvence **POZ-ZAR-XX**, kde **POZ** je zkratka pro požadavek, **ZAR** je konkrétní zařízení a **XX** je unikátní identifikátor požadavku.

5.1 Požadavky na systém

Z cíle práce vycházejí nejobecnější požadavky na celý systém. Jsou shrnuty v bodech v tabulce - Tabulka 1. První odvozený požadavek (POZ-SYS-01) je decentralizace. Decentralizace celého systému zajistí, že v případě jednoho prvku systému se celý systém nezhroutí a nepřestane fungovat. To znamená, že nebude mít centrální prvek. Díky tomu se zvedne celková odolnost a spolehlivost systému. Druhým požadavkem (POZ-SYS-02) je nutnost, aby systém fungoval alespoň pro tři místnosti. Tímto požadavkem se zajistí, že bude možné systém použít v praxi. Pokud by tomu tak nebylo, systém by byl použitelný jenom v omezené míře [4]. Třetí požadavek (POZ-SYS-03) souvisí s decentralizací, vlastně jej obecně popisuje. Je to požadavek na odolnost vůči výpadku jednoho prvku. Příkladem z tohoto systému může být, že při výpadku jednoho měřicího zařízení nedojde k ovlivnění funkce dalších měřicích zařízení. V případě, že by se jednalo o síť typu mesh, upravila by se tak, aby fungovala bez problému, pokud by to bylo topologicky možné. Čtvrtým požadavkem (POZ-SYS-04) je odolnost vůči nedostupnosti internetu. To zajistí plnou a samostatnou funkčnost i v případě, kdy by došlo k jeho výpadku během běhu, tak i k výpadku před startem celého systému. Při obnovení připojení dojde k návratu do běžného režimu. Pátým požadavkem (POZ-SYS-05) je nutnost dvou typů zařízení. V tomto systému se budou vyskytovat dva druhy: měřicí a spínací. Každé bude plnit svoji funkci,

i kdyby měly být zabudovány v jednom společně. Šestým požadavkem (POZ-SYS-06) je synchronizace konfigurace. Ta zajistí, že všechna zařízení, která se nacházejí v systému, budou mít identickou konfiguraci, a to i v případě, že bude některé z nich nedostupné. Pokud by došlo k připojení nového zařízení, tak zajistí, že globální konfigurace bude distribuována i jemu. Druhým způsobem, jak lze tento požadavek splnit je možnost synchronizace konfigurace s cloudem v případě, že je dostupný internet. Sedmým požadavkem (POZ-SYS-07) je komunikace pomocí wifi. Díky tomuto bezdrátovému protokolu bude možnost zajistit, že zařízení spolu nebudou muset být fyzicky propojená a tím pádem se zjednoduší jejich instalace. Posledním, osmým požadavkem (POZ-SYS-08) je konfigurace z libovolného místa. Tím se rozumí, že bude v systému možné nastavit všechny parametry odkudkoli ať už lokálně nebo vzdáleně právě pomocí cloudu.

Tabulka 1 – Požadavky na systém (zdroj: autor)

Id požadavku	Popis požadavku
POZ-SYS-01	systém musí být decentralizovaný
POZ-SYS-02	systém musí pokrýt dostatečný počet místností
POZ-SYS-03	systém musí být odolný vůči výpadku jednoho prvku
POZ-SYS-04	systém musí fungovat i bez funkčního připojení k internetu
POZ-SYS-05	systém musí obsahovat dva typy zařízení – měřicí a spínací
POZ-SYS-06	systém musí být schopen si synchronizovat konfiguraci v rámci zařízení
POZ-SYS-07	komunikace v systému může být přes protokol wifi
POZ-SYS-08	systém musí být konfigurovatelný z libovolného místa

5.2 Požadavky na prvky systému

V cíli práce nejsou určeny specifické požadavky na jednotlivé prvky systému, které jsou stanoveny v této kapitole vycházející z předchozích požadavků. Požadavky na prvky systému jsou rozděleny do třech skupin, pro každý typ zařízení a pak pro samotný cloud.

5.2.1 Požadavky na měřicí zařízení

Vzhledem k účelu zařízení (měření), jsou stanoveny a odvozeny požadavky. Prvním a zásadním požadavkem je, aby samotné zařízení umělo měřit teplotu (POZ-MER-01). To zajistí, že splní svůj účel měřicího zařízení. Druhým požadavkem (POZ-MER-02) je rozlišení teploty. Nominální rozlišení pokojové teploty je v tomto případě stanoveno od 5 °C do 30 °C. Z vlastní zkušenosti by tento rozsah měl pokrýt pokojovou teplotu s dostatečnou rezervou, a to i pro neobydlené místnosti. V krajním případě, že by teplota byla mimo rozsah, musí naměřit hodnotu odpovídající krajním hodnotám. To znamená, že pokud by teplota byla nižší než minimum, musí naměřit alespoň minimální hodnotu. Pro maximální hodnotu platí to samé, ale v opačném směru. Třetí požadavek (POZ-MER-03) říká, že měřená teplota musí být v dostatečném rozlišení. Zde bylo rozlišení nastaveno na maximální odchylku 0,2 °C. Takový rozdíl od teploty člověk není v pokojové teplotě schopen rozpoznat. Četnost měření dat (POZ-MER-04) je stanovena na interval každých pět minut. Důvodem je to, že teplota se nedokáže měnit tak rychle, aby bylo nutné měřit v reálném čase. Zároveň aby výsledná teplota nemohla být ovlivněna náhodnou změnou, bude pracovat s klouzavým průměrem o pěti hodnotách.

Pátým požadavkem (POZ-MER-05) je napájení. Jelikož se jedná o aktivní zařízení, které bude měřit a používat protokol wifi, bude muset být aktivně napájeno. Zde se nabízí několik možností, jak lze napájení realizovat. První je za použití akumulátoru, druhá možnost je napájení z elektrické sítě. Požadavek s pořadovým číslem šest (POZ-MER-06) je umístění zařízení. Jelikož se jedná o zařízení měřicí teplotu, je zde několik věcí, které se musí zohlednit tak, aby měření bylo co nejméně ovlivněno. To znamená, že měřicí zařízení nemůže být blízko oken, kde by mohl být průvan, což by negativně ovlivnilo měření. Stejný případ je i umístění blízko dveří, kde by mohl nastat identický problém. Ještě jedna z věcí, které by mohly ovlivnit měřenou teplotu je stěna, na kterou se umístí, a to v případě, že by se jednalo o stěnu, která je zároveň i stěnou venkovní.

Co se týče výšky (POZ-MER-07), zařízení bude muset být umístěno v úrovni metr šedesát. Jelikož se jedná o zařízení umístěné v místnosti na stěně, musí být ideálně co nejmenší, tak aby co nejméně překáželo. Maximální velikost, respektive objem, je stanovena na 300 cm^3. S rozměry se také váže i hmotnost samotného zařízení (POS-MER-08). Jelikož se jedná o zařízení, které bude umístěné na stěně, je nutné, aby bylo co nejlehčí. To znamená, aby nebylo potřeba zásahu do zdi, kde bude zařízení umístěno. Je tedy potřeba, aby bylo dostatečně lehké. Požadovaná hmotnost je do 150 gramů. Jelikož v požadavcích na systém byl uveden požadavek POZ-SYS-08, je nutné, aby jej splnilo i měřicí zařízení.

Požadavek POZ-MER-09 říká, že půjde konfigurovat systém právě i z měřicího zařízení. Ke konfiguraci se váže i další požadavek (POZ-MER-10), který stanovuje, že konfigurace bude persistentně ukládána, tedy že i po restartu nebo odpojení a připojení zařízení se konfigurace opět načte. K tomu, aby bylo možné přenést data a splnit další požadavek celého systému (POZ-SYS-07) je stanoven požadavek POZ-MER-11, který garantuje wifi konektivitu měřicího zařízení. S ním se váže i další požadavek POZ-MER-12. Zařízení bude vybavené wifi schopné přenášet data systému.

Tabulka 2 – Požadavky na měřicí zařízení (zdroj: autor)

Id požadavku	Popis požadavku
POZ-MER-01	měřicí zařízení musí měřit teplotu místnosti
POZ-MER-02	měřicí zařízení musí mít dostatečně velké rozlišení měřené teploty
POZ-MER-03	měřená teplota se nesmí lišit od reálné v rozmezí 0,2 °C
POZ-MER-04	měřená teplota musí být získávána v intervalu kratším než 1 hodina
POZ-MER-05	měřicí zařízení musí být napájeno buď z baterie, akumulátoru nebo ze sítě
POZ-MER-06	měřicí zařízení musí být umístěno na optimálním místě v místnosti
POZ-MER-07	měřicí zařízení musí být malé, aby se bezpečně dalo umístit na stěnu bez nutnosti složité montáže

Id požadavku	Popis požadavku
POZ-MER-08	měřicí zařízení by mělo být lehké, optimálně do 150 gramů
POZ-MER-09	měřicí zařízení musí umožnit konfiguraci ze sebe samotného pro celý systém
POZ-MER-10	měřicí zařízení musí perzistentně ukládat data o systému
POZ-MER-11	měřicí zařízení může komunikovat pomocí wifi
POZ-MER-12	měřicí zařízení musí umět přenášet data sebe sama a ostatních zařízení v systému

5.2.2 Požadavky na spínací zařízení

Druhým typem zařízení je zařízení spínací. Jedná se o zařízení, které bude schopno ovládat teplotu v místnostech. Cíl práce konkrétně neurčil přesný druh, ani žádné specifické detaily takového zařízení. Je tedy definováno tak, aby jej bylo možné použít pro domácí použití. V tomto systému se bude pracovat s takovým zařízením, které je schopné ovládat všechny místnosti naráz pro zjednodušení komplexity.

Prvním požadavkem (POZ-SPI-01) je důležité určit, jaké druhy vybavení topného systému se budou spínat. Zde se počítá s několika možnostmi. Může jít buď o dálkové ovládání topných hlavic nebo ventily, případně i topná tělesa. Pracovat se bude v binárním režimu, tedy zapnuto / vypnuto. (POZ-SPI-02) Dle původních požadavků je nutné ovládat alespoň tři místnosti, a z toho pro optimální ovládání vyplývá, že pro každou místnost bude jeden spínací člen. (POZ-SPI-03 a POZ-SPI-04) Aby je bylo možné ovládat, je potřeba aktivní prvek, který bude schopen měnit jejich stavy. Nabízí se dvě možnosti, buď relé nebo tranzistory. V tomto případě je vyžadováno relé, kde je sice potřeba vyšší spínací napětí a proud, ale jeho zásadní výhodou je galvanické oddělení. Zároveň je schopné spínat větší proudy v případě nutnosti, to by ale v tomto případě nemělo být potřeba. Při spínání prvků ovládající topení se dá předpokládat, že odezva nebude instantní, ale mezi kompletním otevřením a zavřením bude prodleva. Je tedy nutné navrhnout zařízení tak, aby spínalo maximálně jednou za pět minut nebo dle parametrů měřicího zařízení (POZ-SPI-05). Samotné rozhodování o spínání nebude v roli tohoto zařízení. Bude ale v roli jednotlivých měřicích zařízení přiřazených pro jednotlivé místnosti (POZ-SPI-06).

Další požadavky jsou sdílené s měřicím zařízením, ale jelikož se jedná o jiný druh, je potřeba je zopakovat a upravit.

Co se týče napájení, je zde jediné optimální řešení, a to napájení ze sítě. Důvodem je, že spínání relátek je energeticky náročné a pro dlouhodobý provoz by byly akumulátory nevhodné, protože by to znamenalo nutnost mít velkou kapacitu a tím pádem zabírat i velkou plochu. Tato možnost je tedy vyloučena. (POZ-SPI-07) Rozměry zařízení zde nejsou tak důležité, protože bude umístěno v místě spínaných ventilů, a tedy nebude běžně vidět. Zároveň by ale nemělo být přes celou místnost. Jeho velikost by tedy měla být co nejkompaktnější vzhledem k použitým komponentům. (POZ-SPI-08) Hmotnost zde bude také vyšší než u měřicího zařízení. Přidaná váha je hlavně způsobena relátky, která budou vážit několik desítek až stovek gramů. (POZ-SPI-09). Tak jako u měřicího zařízení je zde cena velice důležitá a neměla by být několikanásobná oproti němu (POZ-SPI-10). Další tři požadavky (POZ-SPI-11, POZ-SPI-12, POZ-SPI-13) jsou identické s měřicím zařízením.

Tabulka 3 – Požadavky na spínací zařízení (zdroj: autor)

Id požadavku	Popis požadavku
POZ-SPI-01	spínací zařízení musí umět ovládat topící elementy
POZ-SPI-02	spínací zařízení musí mít prvek, se kterým bude spínat koncové prvky
POZ-SPI-03	spínací zařízení musí umět spínat alespoň tolik prvků, kolik má být ovládáno místností, minimálně však 3
POZ-SPI-04	spínané prvky musí být podporované spínacím zařízením
POZ-SPI-05	spínací zařízení musí mít minimální prodlevu jednu minutu
POZ-SPI-06	spínací zařízení bude spínat na základě pokynů měřicího zařízení
POZ-SPI-07	spínací zařízení musí být napájeno ze sítě
POZ-SPI-08	spínací zařízení by mělo být co nejmenší na základě použitých komponent
POZ-SPI-09	měřicí zařízení by mělo být lehké, tak aby šlo jednoduše umístit ke spínaným prvkům

Id požadavku	Popis požadavku
POZ-SPI-10	měřicí zařízení musí být co nejlevnější při splnění všech požadavků
POZ-SPI-11	spínací zařízení musí umožnit konfiguraci ze sebe samotného pro celý systém
POZ-SPI-12	spínací zařízení může komunikovat pomocí wifi
POZ-SPI-13	spínací zařízení musí umět přenášet data systému

5.2.3 Požadavky na cloud

Třetím a posledním prvkem je cloud. Přesněji řečeno cloud ve vlastní správě bez zásahu třetí strany. První požadavek (POZ-CLD-01) zaručí, že půjde z cloudu konfigurovat celý systém a tím pádem se splní jeden z původních základních požadavků. Druhým, zásadním požadavkem (POZ-CLD-02) je dostupnost z internetu. Ačkoliv se může zdát, že je to požadavek zbytečný, tak cloud, který by nebyl z internetu dostupný, by vlastně ani cloudem nebyl. Třetí požadavek (POZ-CLD-03) zajistí, že po konfiguraci systému se data uloží a zůstanou perzistentní i v případě, že by došlo k výpadku spojení nebo restartu cloudu z jakéhokoli důvodu. Čtvrtý požadavek (POZ-CLD-04) dává povinnost cloudu umět synchronizovat data se zbytkem systému. Není zde určeno, jakým způsobem by tato akce měla být prováděna. Jako jeden ze způsobů se nabízí pomocí HTTP protokolu, který je bezstavový, rozšířený, a hlavně i relativně jednoduchý na implementaci. V případě využití stavového protokolu by se zde nabízela možnost použití WebSocketů.

Tabulka 4 – Požadavky na cloud (zdroj: autor)

Id Požadavku	Popis požadavku
POZ-CLD-01	cloud musí umožnit nakonfigurovat celý systém
POZ-CLD-02	cloud musí být dostupný z internetu
POZ-CLD-03	cloud musí trvale ukládat systémová data
POZ-CLD-04	cloud musí umět synchronizovat data se zbytkem systému, tedy zařízeními

6 Charakterizace dílčích prvků systému podle požadavků

Dle požadavků z předchozí kapitoly se systém bude skládat ze tří typů základních prvků, a to ze dvou druhů zařízení (měřicí a spínací) a třetím typem bude cloud. Počet jednotlivých prvků se bude lišit. Cloud bude jenom jeden, společný. Ačkoliv by jich mohlo být více, tak je zde zvolena cesta jednoho. To z toho důvodu, že dostupnost cloudů je vysoká a šance na výpadek statisticky malá. U spínacího zařízení je to identické, tedy jedno, které bude schopno ovládat akční prvky topení. U měřicího zařízení je to zajímavější, tam se počet bude lišit, a to od jednoho až po maximální počet, který je dán jak limitací hardware (ta je ale vyšší), ale hlavně počtem místností. Počítá se s jedním měřicím zařízením na jednu vytápěnou (kontrolovanou) místnost. V podkapitolách jsou rozepsány jednotlivé použité hardwarové a softwarové prvky, které jsou nezbytné pro návrh celého systému.

6.1 Části systému (podsystémy)

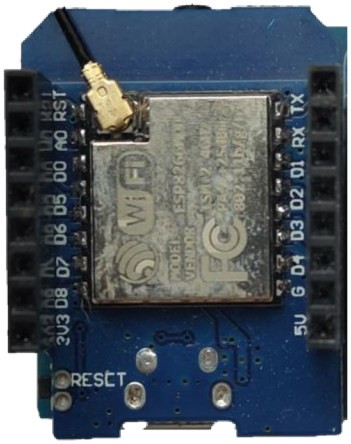
Jelikož je systém jako celek složen z několika částí, je nutné jej rozdělit na několik menších částí a ty následně popsat.

6.1.1 Společné části měřicího a spínacího zařízení

Na základě požadavků na měřicí a spínací zařízení je zvolena společná komponenta. Tou je systém na čipu od společnosti Espressif – ESP8266EX. Jedná se o malé, přenosné zařízení vhodné právě pro tuto IoT aplikaci. Mezi jeho přednosti patří tyto věci [5]:

- nízká pořizovací cena,
- dostatečný výpočetní výkon,
- nativní podpora wifi (2,4 GHz),
- nativní podpora TCP/IP,
- velmi dobrá podpora vývojových platforem,
- velké množství komunitních knihoven,
- rozmanitost vstupů a výstupů.

Samotný modul ovšem nenabízí žádnou perzistentní paměť a ani jednoduchost v tom, jak jej naprogramovat. To přináší možnost, aby si další firmy přišly s vlastní implementací. Právě jedna z nich je zvolena, a tou je deska LOLIN D1 mini od společnosti WEMOS. Ta k samotnému ESP8266EX modulu přidává USB rozhraní, integrovanou anténu pro wifi, externí 4 MiB flash paměť. Hlavně také vstupy a výstupy ve formě špiček, které se dají zapojit do patice pro připojení dalších, rozšiřujících periférií. Díky všem těmto vlastnostem zajistí kvalitní základ pro oba druhy zařízení.



Obrázek 2 – Kompletní modul LOLIN D1 mini (zdroj: autor)

Co se týče klíčové bezdrátové konektivity, tedy wifi, je zde podpora třech standardů. IEEE 802.11b/g/n, dnes moderně a zjednodušeně označované jako wifi 1, 3 a 4. Všechny tyto standardy pracují v ISM pásmu 2,4 GHz. Níže v tabulce jsou jednotlivé standardy vypsány s podstatnými parametry. Konkrétně to jsou teoretická propustnost, která udává maximální bitový průtok. Vysílací výkon antény a její citlivost při příjmu určují nejnižší možnou sílu signálu, při které lze ještě úspěšně zpracovávat přijatá data. Z tabulky je jasně vidět, že čím vyšší propustnost chceme, tím citlivější a náchylnější na chybovost je její příjem. Zároveň klesá i maximální vysílací výkon. Jelikož se nepředpokládá u měřicího systému velký přenos dat a zároveň je preferován co nejvyšší dosah, tj. nejnižší citlivost. Je proto zvolen a v rámci sítě zamčen na IEEE 802.11b. Jedinou potencionální limitací je, že přístupové body wifi sítě pro konektivitu ke cloudu by mohly mít vypnutý takzvaný mixed mode, který umožňuje použít i tento starší režim.

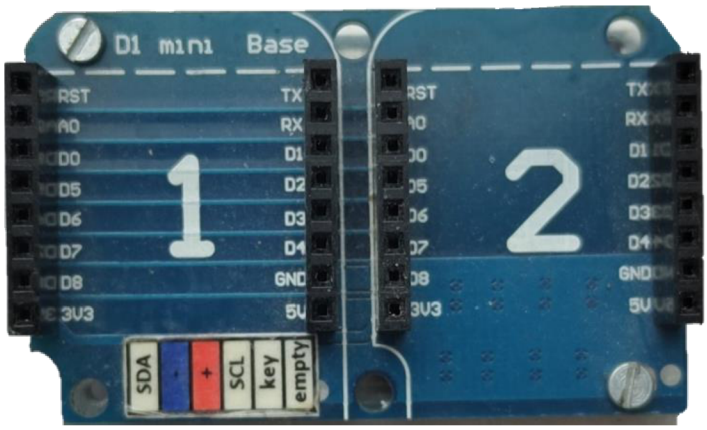
K tomu, aby komunikace byla při přenosu chráněna proti odposlechu je zde podpora zabezpečení sítí. Ta se skládá z WEP (dnes již zastaralé), WPA a WPA2 s podporou šifrování TKIP nebo AES. Bohužel nejnovější a nejbezpečnější podpora pro WPA3 zde chybí, a to pravděpodobně z toho důvodu, že se jedná o zařízení, které je na trhu už několik let a pro podporu by bylo nutné využít novější, výkonnější, ale zároveň i dražší řadu jako např. ESP32.

Tabulka 5 – Podporované wifi parametry ESP8266 (zdroj: [5])

Wifi standard	Propustnost	Vysílací výkon	Citlivost
IEEE 802.11b	11 Mbit/s	20 dBm (100 mW)	-91 dBm
IEEE 802.11g	54 Mbit/s	17 dBm (50 mW)	-75 dBm
IEEE 802.11n	72,2 Mbit/s	14 dBm (25 mW)	-72 dBm

Aby mohl modul komunikovat s okolím, nabízí širokou škálu vstupních a výstupních rozhraní a špiček k nim přiřazených. Jako příklad lze uvést UART, SPI, I^2C (sériová rozhraní) nebo také GPIO (univerzální špička), ADC (analogovo číslicový převodník), PWM (ovládání pomocí střídy). [5] Konkrétní vyvedení špiček není v tomto případě podstatné zmínit, protože u každého modulu se může lišit a je vždy nutné je dohledat u konkrétního datasheetu výrobce. Celkový počet vyvedených špiček je u samotného ESP8266EX třicet dva.

K propojení hlavního modulu a dalších komponent je použita pomocná základna, která umožní jejich snadné připojení. Další výhodou je, že základna má otvory pro šroubky, díky které ji lze připevnit např. do vnitřku krabičky nebo kamkoliv jinam tak, aby byla pevně připevněna u tohoto prototypu. Pro finální produkt by toto řešení nebylo vhodné, jelikož by bez použití šlo zařízení ještě zmenšit.



Obrázek 3 – Modulární základna k propojení (zdroj: autor)

6.1.2 Měřicí zařízení

Měřicí zařízení se bude skládat z několika dílčích částí. Hlavní částí je samotný modul s ESP8266EX a základna pro umístění na pevnou podložku. Jelikož sám o sobě neobsahuje čidlo pro měření, je nutné ho dodat.

K měření je zvoleno čidlo z řady SHT 3X od společnosti Sensirion, konkrétně verze SHT30. Jedná se o levná, moderní, a hlavně dostatečně přesná čidla založená na polovodičové technologii. Další výhodou je, že jsou z výroby plně kalibrovaná. Zároveň je zde tepelná kompenzace spolu s kompenzací dle napájecího napětí. To znamená, že výstup není ovlivněn provozní charakteristikou součástky. Teplota, při které je čidlo schopné pracovat správně je od -40 do 125 °C. Tento rozsah je dostatečný, aby pokryl běžné pokojové teploty. Výrobce udává, že v rozsahu teplot od 0 do 65 °C je garantovaná odchylka naměřené od skutečné hodnoty 0,2 °C. Opakovatelnost je další popsanou vlastností čidla. Zde se nabízí tři úrovně od nejnižší, po nejvyšší. Čím vyšší požadujeme opakovatelnost (přesnost), tím je měření samozřejmě delší. Jelikož měření teploty není v systému tak časté a zároveň délka měření v nejpřesnějším režimu je průměrně 12,5 milisekund je zvolen právě tento režim. Ke komunikaci používá sériové rozhraní I^2C, což je kompatibilní s ESP8266EX, které je možné použít. [6]

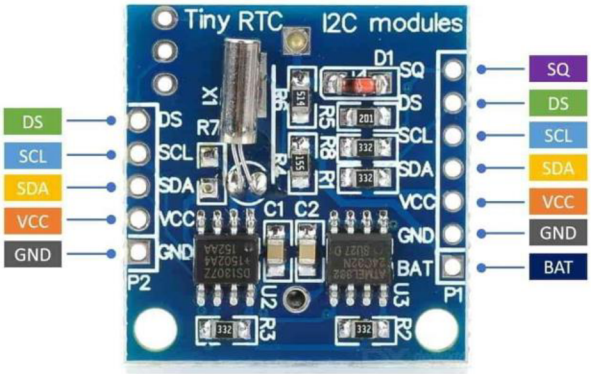
Obrázek 4 – Deska s čidlem SHT 30 od společnosti Wemos (zdroj: autor)

6.1.3 Spínací zařízení

Spínací zařízení podobně jako měřicí zařízení, vychází z modulu ESP8266EX, ale dle požadavků musí umět jiné věci než měřicí. To znamená, že teplotní čidlo nebude obsaženo. Toto zařízení má spínat, je tedy zvolena hotová deska se čtyřmi relátky. To umožní kontrolu až čtyř ventilů, což bohatě pokryje požadavky stanovené v předchozích kapitolách. Jelikož se jedná o relátka, není zde žádná komunikační sběrnice, která by zbytečně přidávala komplexitu. Je tedy použito jednoduché binární ovládání zapnuto / vypnuto pomocí napětí na jednotlivých pinech. K udržení zapnutí každého relátka si vyžádá proud 15 mA až 20 mA, což je na nejvyšší maximální hranici, co ESP8266EX poskytne. Maximální výkon přenášený skrze relátka v stejnosměrném režimu je 30 voltů při 10 ampérech a při střídavém zvládne až 250 voltů při 10 ampérech. Relátka jsou samozřejmě galvanicky oddělena, a to za použití optočlenů. Typ použitých relátek je „SPDT“ (anglicky single pole, double throw). Česky by se to dalo vysvětlit jako dvou stavové relé. Je tedy možné ovládání jednoduše negovat, protože relé má dva výstupy. [7]

Obrázek 5 – Deska se čtyřmi relátky (zdroj: autor)

Vzhledem k tomu, že má systém fungovat bez připojení k internetu, je důležité mít někde přesnou časovou referenci. Bez ní by nebylo možné přesně čas ovládat, jelikož by nebyl znám ani den a ani čas, tedy topení by bylo čistě náhodné. Proto je zvolena komponenta s hodinami reálného času, která tuto funkčnost zajistí. Limitací je, že je potřeba alespoň jedno připojení k internetu, aby se referenční hodiny nastavily. Případně méně přesnou alternativou by bylo získání systémového času od uživatele v prohlížeči. To ale není předmětem této kapitoly. K ukládání referenčního času je použit Tiny RTC module, který na sobě má integrovaný obvod s hodinami reálného času DS 1307. Jedná se o podobný případ jako u ESP8266EX, kde samotný integrovaný obvod sám o sobě fungovat nemůže. Díky dalším komponentám již zabudované a připravené na Tiny RTC desce je možné tyto hodiny reálného času zapojit a rovnou používat. Pro komunikaci používá protokol I^2C tak, jako například teplotní čidlo, tedy kompatibilita v tomto ohledu je zajištěna. [8]



Obrázek 6 – Deska s hodinami reálného času (zdroj: [8])

6.1.4 Cloud

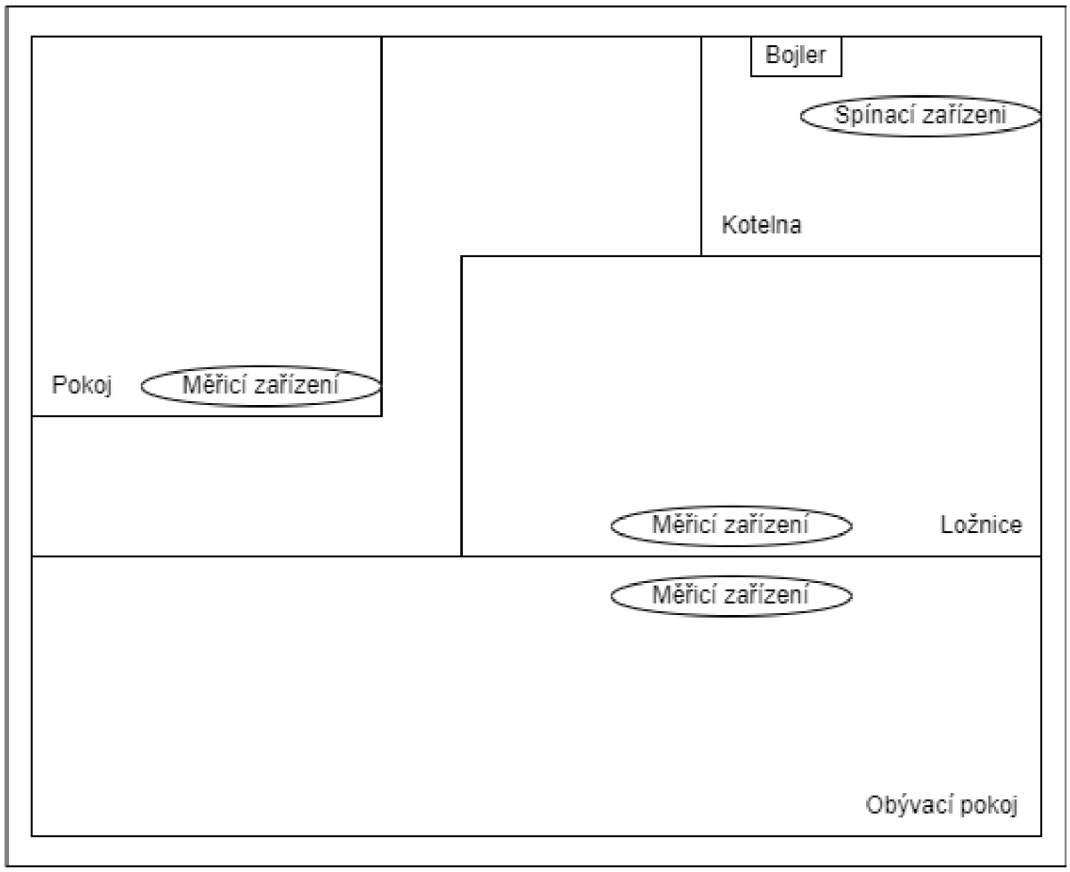
Cloud je v požadavcích, respektive použitých technologií, volný. Jediné, co vyžaduje je funkčnost vůči systému. Z logiky věci zde dává použít nějaké open sourcové (s otevřeným zdrojovým kódem) řešení, což v dnešní době znamená GNU/Linux, které se těší čím dál větší popularitě díky například i možností kontejnerizace.

7 Návrh technického řešení

Tato kapitola popisuje využití navrženého hardware k návrhu samotného řešení.

7.1 Systém topení

Návrh systému je takový, že bude jedno spínací zařízení umístěné u ventilu topení, což s největší pravděpodobností bude v místnosti s kotlem. Umístění měřicích zařízení bude tak, jak bylo v minulých kapitolách zmíněno ve výšce zhruba 160 cm v místě s minimálním průvanem. Na obrázku níže lze nalézt možné umístění zařízení v teoretickém patrovém domě. Velikost zařízení není ve stejné škále jako zbytek domu.



Obrázek 7 – Možné umístění zařízení v domě (zdroj: autor)

Systém nebude mít centrální uložiště, nýbrž celá konfigurace bude uložená v perzistentní paměti každého aktivního prvku (tj. flash paměť u ESP8266EX a u cloudu to bude disk a data v plaintext formě, protože systém není tak rozsáhlý, aby potřeboval databázi), a to i včetně cloudu. Každá konfigurace bude mít vlastní unikátní identifikátor, aby bylo možné rozhodnout, která je novější. Řešením je epocha, což je počet sekund od 1. 1. 1970, tedy přesnost je sekundová a nepředpokládá se, že by se konfigurace tak často měnila, aby mohlo dojít ke konfliktu. Každé zařízení v systému bude umět upravit konfiguraci jak svoji, tak i ostatních zařízení, a to i když jsou offline. Synchronizace bude automatická a nebude vyžadována žádná interakce s uživatelem. Konfigurace bude minimálně obsahovat data o relátkách, teplotu vysokou a nízkou (mezi lidmi známé jako měsíček a sluníčko), zvolené relátko, vytápěcí program s rozlišením na hodiny a uživatelské názvy pro přívětivost při nastavování. Konfigurace bude možná buď lokálně (doma) nebo i vzdáleně z internetu za pomocí cloudu. To umožní nastavování např. když jsme na dovolené, v práci, jednoduše když nejsme doma. Způsob konfigurace je pomocí webového prohlížeče (případně i mobilní aplikace), což umožní přístup z téměř jakéhokoliv zařízení připojeného buď k internetu nebo v lokální síti.

Systém je koncipovaný tak, že pro každou místnost je jedno měřicí zařízení, a to bude ovládat právě jedno relátko. Měřicí zařízení má svůj topící program a podle naměřené hodnoty je bude spínat. Frekvence spínání je stanovena na interval patnáct minut, interval měření teploty na pět minut.

Systém musí mít platný čas, jinak není možné ovládat topení. Platný čas lze v systému získat třemi způsoby:

- z internetu pomocí NTP,
- z hodin reálného času na spínacím zařízení,
- z ostatních zařízení v případě, že mají platný čas.

Je tedy nutné, aby systém byl alespoň jednou připojen k internetu a zároveň byl tento čas uložen do hodin reálného času spínacího zařízení. Díky tomu by při potencionálním výpadku celého systému a restartu nebyl potřeba znovu přístup

k internetu. Pokud je systém připojen k internetu, synchronizace času z NTP bude pravidelná v intervalu několika minut. Šířený čas bude mít různou prioritu na základě toho, z jakého zdroje pochází. Pracovně je tato priorita nazvaná stratum stejně tak, jako to má samotný NTP protokol. Priorita je nejvyšší u času z internetu, poté z hodin reálného času a nejnižší prioritu má synchronizovaná lokálně z operační paměti.

V případě nedostupnosti měřicího prvku bude topení vypnuté pro dané relátko (místnost). Jelikož není možné odhadnout, jaká je aktuální teplota v místnosti, tak zde se jako menší riziko předpokládá vypnutá a podchlazená místnost než taková, ve které se neustále topí a tím se zvyšují náklady za topení. Toto je jedna věc, kterou by toto řešení mělo co nejvíce omezit a spíše na druhou stranu by mělo náklady snížit.

7.2 Společný základ pro zařízení

Návrh samotného měřicího a spínacího zařízení začíná u firmware, tedy běhovém kódu. Nejdříve je nutné vybrat vývojovou platformu na základě, které se bude všechno odvíjet. U ESP8266 je díky tomu, že je v produkci již několik let, rozmanité množství platforem. Mezi nejznámější může patřit Arduino, PlatformIO, ESP-IDF, NodeMCU Lua nebo také MicroPython. Každá z těchto platforem má své výhody a nevýhody, které budou rozebrány v podkapitole a následně bude jedna vybrána a pomocí ní realizován firmware.

7.2.1 Porovnání podporovaných platforem pro ESP8266EX

Dle textu výše byly vyjmenovány různé platformy pro vývoj na této desce. Před výběrem konkrétní platformy následuje jejich popis.

Arduino

Platforma Arduino je velice známou platformou mezi lidmi, kteří se zajímají o prototypování hardware. V základu sice Arduino nepodporuje žádné desky od Espressif, ale díky komunitě je podpora přidána i právě pro zvolenou ESP8266. Díky těmto vlastnostem je také dostupné mnoho rozšiřujících knihoven, které zpravidla bývají kompatibilní skrze všechny platformy, pokud používají společnou

sběrnici. Mnoho výrobců různých senzorů, čidel a aktuátorů má knihovny vlastní a jsou dobře podporovány. Zároveň je také možné napsat vlastní knihovnu, kterou lze obohatit celou komunitu vývojářů. Na internetu jsou dostupné návody a příklady, které mohou vývoj usnadnit právě pro nováčky.

Arduino je také multiplatformní, takže se vývojář nemusí omezovat na jeden konkrétní operační systém, ale může použít kterýkoliv populární, jako např. Windows, Linux/GNU nebo i macOS. Jako nevýhodu je dobré zmínit, že podpora ladění je zde velice omezená. Naštěstí existuje možnost, jak tento nedostatek odstranit. Jedná se o rozšíření do populárního vývojářského prostředí Visual Studio s názvem VisualMicro. Jedinou nevýhodou Visual Studia je možnost ho mít pouze na operačním systému MS Windows. Rozšíření je placené, ale pro studenty a nekomerční projekty je zlevněné. Toto rozšíření poskytuje ladění i s podporou „zarážek“ (breakpoints), kvalitního našeptávání, refaktoringu a spoustu dalších vlastností, které samotné Arduino IDE nenabízí. Tím se ulehčí vývoj a zkrátí se čas strávený na projektu.

PlatformIO

PlatformIO je další z platforem, která se nabízí k vývoji. Je podobná s Arduino platformou, ale s tím rozdílem, že má lepší vývojové prostředí, jelikož se jedná o plugin pro Visual Studio Code, které je multiplatformní. Celá platforma je zdarma pro vlastní vývoj a knihovny jsou také sdílené s Arduinem. Zároveň podporuje i ladění. Bohužel ale s jednou výjimkou a tou je ESP8266, u které dle dokumentace není možné ladit žádnou desku. [9] Jelikož ladění je u vývoje důležitý faktor, je tato platforma vyřazena.

ESP-IDF (nativní SDK)

Pokud bychom chtěli použít nástroje přímo od tvůrců ESP8266, je zde možnost zvolit ESP-IDF. Výhodou tohoto řešení je, že vývojář má absolutní kontrolu nad vším, co za kód poběží, protože je zde všechno implementované na nejnižší možné úrovni v jazyce C. Z těchto skutečností vyplývá, že výkon tohoto řešení má potenciál být největší oproti ostatním. Zásadní nevýhodou je, že křivka učení je díky tomuto přístupu nejstrmější ze všech. K vývoji se používá IDE přímo

od Espressif, ale to je založené na Eclipse, což v porovnání s předchozími prostředími je s funkcemi dnes již zastaralé. Navíc je nutné se naučit ovládat složitější kompilační nástroje jako CMake. Zásadní výhodou, co ESP-IDF nabízí, je implementace operačního systému reálného času (RTOS). Díky tomuto přístupu je zase možný jiný způsob vývoje než u předchozích platforem, které tuto možnost nenabízí.

NodeMCU Lua a MicroPython

Pokud není vývojář seznámen s nízko úrovňovými programovacími jazyky, jako jsou C a C++, lze si vybrat dvě platformy poskytující abstrakci v jazycích vyšší úrovně. Prvním je Lua a druhým je Python. Obě řešení mají zásadní problém. Jedná se o jazyky interpretované, tedy jejich zdrojový kód je i na zařízení lehce čitelný a zároveň interpretace stojí procesorový čas, tedy jejich finální výkon bude nižší než u kompilovaných.

Firmware podporující jazyk Lua se nazývá NodeMCU, což je společnost, která vyvíjí i své vlastní moduly, ale tento firmware běží bez limitací na všech druzích. Výhodou u tohoto řešení je na druhou stranu méně strmá křivka učení a jednodušší syntax. Zároveň nemusíme řešit manuálně správu paměti, u které by mohlo dojít k přetečení, přístupu do špatné části paměti a podobným nedefinovaným chováním. Nevýhodou ovšem je, že je zde limitovaná podpora knihoven, vývojové prostředí je velice limitované a nepodporuje žádné ladění v klasické formě. Komunitně také není rozšířena tak jako ostatní a je tedy tato platforma vyřazena.

U MicroPythonu je to velice podobná situace s tím rozdílem, že počet podporovaných knihoven je ještě menší, a tudíž není vhodný pro vývoj tohoto řešení.

7.3 Výběr rozdělení firmware

Jako programovací jazyk je vybrán C++ spolu s velice známou komunitní platformou s otevřeným zdrojovým kódem Arduino. Zde se nabízí dvě možnosti řešení. Jednotný firmware pro obě zařízení nebo oddělený software pro každé zařízení zvlášť. Každé z těchto řešení má své výhody a nevýhody.

7.3.1 Společný firmware

Společný firmware je takový firmware, ve kterém bude obsažena logika obou zařízení, a to spínacího a měřicího. Toto řešení má výhodu v tom, že je jednodušší na údržbu a aktualizaci, protože pro všechny zařízení šíříme jenom jeden a ten samý kus binárního kódu. Není tedy nutné rozlišovat typ zařízení při aktualizaci. Zároveň se zde zamezí i tzv. mokrému kódu (wet code), kde by byla část kódu duplikovaná „a dělala by jednu a tu samou věc. Další výhodou jsou také pravděpodobně nižší náklady na vývoj a případně testování, protože vše je na jednom místě a chyby, případně opravy chyb budou sjednocené.

Mezi nevýhody lze zařadit větší komplexitu, protože se musí zahrnout logika obou zařízení do jednoho kusu software. Tímto stylem se zase může zvýšit chybovost kódu. Přidaným kódem se také může snížit výkon, pokud by běžela část, která nemá.

7.3.2 Rozdělený firmware

U rozděleného firmware jsou výhody a nevýhody přesně opačné.

7.3.3 Shrnutí

Po uvážení všech výhod a nevýhod byl pro vývoj z logičtějšího hlediska zvolen společný (sdílený) firmware.

7.4 Komunikace v systému

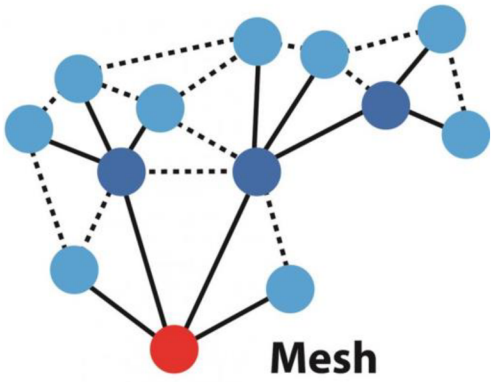
Data budou přenášena každým koncovým zařízením v systému, respektive takovým, které bude online a připojeno. Tím se rozumí měřicí i spínací zařízení, to zaručí propojenost mezi sebou.

Aby mezi sebou zařízení mohla komunikovat, je využito pro přenos dat v druhé ISO/OSI vrstvě protokolu wifi, který byl jako jeden z požadavků. Verze protokolu bude pevně uzamčena na 802.11b z důvodu nejlepší konektivity i za nejhorších podmínek. Wifi kanál bude variabilní a bude sdílen s wifi přístupovým bodem v síti LAN. Důvod je takový, že pokud chceme, aby ESP8266 fungovalo jako stanice a zároveň klient, kanály musí být identické. Může to mít jednu nevýhodu, a to rušení, pokud by byl přenos dat na daném kanálu moc

vysoký. Řešením je používat 5 GHz wifi pro všechny ostatní zařízení, tím se šance na zarušení ze strany klienta sníží, ale nemusí se vyřešit, pokud by existovala i jiná zařízení v okolí, která by tento kanál používala. V tom případě je řešení změnit kanál na vlastním přístupovém bodu.

Pro přenos vlastních dat v rámci sítě není použit žádný běžně dostupný protokol, jako např. TCP/IP. Místo toho se používá vlastní protokol, který je založený na serializačním textovém formátu JSON.

Pro co největší dostupnost všech zařízení v lokální síti a zároveň i snaha o vytvoření redundantních spojení je pro síť zvolena smíšená topologie (mesh). V této topologii není žádný master (hlavní) prvek a síť je plně reorganizovatelná. Na obrázku níže je možné příklad takové sítě vidět. Plnou čárou jsou vyobrazena aktivní spojení mezi uzly a čárkovaně jsou znázorněna spojení, která mohou v případě výpadku nějakého jiného uzlu vzniknout. To zároveň znamená i to, že struktura sítě nemusí být vždy stejná a může se měnit v čase.



Obrázek 8 – Ukázka sítě mesh (zdroj: [10])

Co se týče přenášených dat, tak navrch k protokolu na úrovni zařízení je implementován protokol na aplikační vrstvě. Je zde několik druhů zpráv, které se po síti přenáší. Mezi některé vybrané lze zmínit:

- NodeTypeBroadcast,
- MeshWifiSignalStrenthToAP,
- NameBroadcast,
- TimeSyncResponse,
- MeasuredTemperatureResponse.

Každý typ zprávy má svůj vlastní a unikátní osmi bitový identifikátor, který je ke zprávě přirazen. Zároveň každá zpráva má svoji vlastní velikost. Ukázku definice typu zpráv lze nalézt v tabulce níže.

Pro zpřehlednění je vybráno několik ukázkových typů zpráv, jejich popis a zároveň jejich velikost před serializací. Tyto zprávy jsou přenášeny lokálně v rámci mesh sítě mezi všemi zařízeními. Všechny zprávy jsou typu broadcast, to znamená, že všechna zařízení dostanou všechny zprávy.

Tabulka 6 – Vybraný přehled typu zpráv a jejich velikost (zdroj: autor)

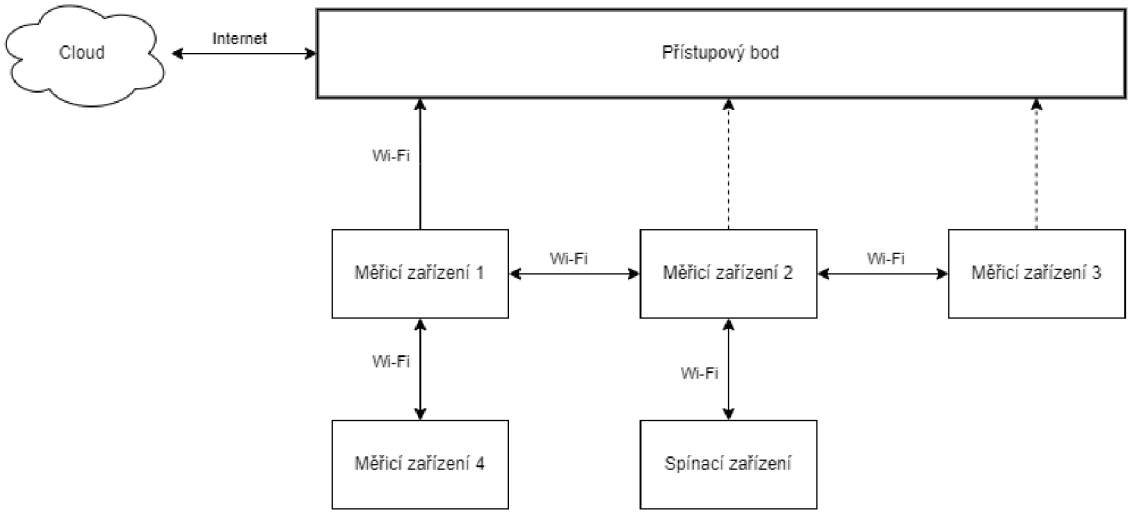
Typ zprávy	Popis	Velikost
PersistentNodeInfoResponse	informace o jednom uzlu	64 bajtů
RelayInformationResponse	kompletní informace o relátkách v systému	116 bajtů
RequestSelectRelayReponse	informace o změně vybraného relé	8 bajtů
HeatingRequestResponse	informace o změně topení	2 bajty
NodeTypeBroadcast	informace o typu uzlu (měřicí / spínací)	1 bajt
MeshWiFiSignalStrenghthToAP	informace o síle k vybranému přístupovému bodu	1 bajt
HeatingProgramResponse	informace o topném programu	40 bajtů
MeasuredTemperatureReponse	informace o naměřené teplotě	4 bajty
TimeSyncReponse	informace o aktuálním čase	12 bajtů

8 Postup realizace navrženého technického řešení

V této kapitole jsou popsány kompletní implementační kroky.

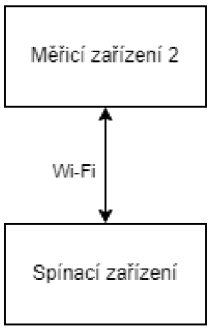
8.1 Navržený systém topení

Blokový diagram systému je znázorněn na obrázku níže. Systém je sestaven ze čtyř měřicích zařízení, jednoho spínacího zařízení a společného cloudu. Přístup ke cloudu (internet) zajišťuje přístupový bod. Komunikace mezi zařízeními je zajišťována prostřednictvím wifi. Pro spojení mezi zařízeními je zvolena smíšená topologie (mesh). Topologie je vytvořena na základě síly signálu. Výpadkem jednoho zařízení dojde k automatické aktualizaci topologie. Přístup ke cloudu prostřednictvím přístupového bodu zajišťuje pouze jedno zařízení, které má ze všech zařízení nejsilnější signál. Na obrázku tomu odpovídá Měřicí zařízení 1. V případě výpadku Měřicího zařízení 1 převezme jeho roli Měřicí zařízení 2 nebo 3.



Obrázek 9 – Blokový diagram propojení systému (zdroj: autor)

Pro ilustraci odolnosti systému lze uvést následující situace, ve kterém je systém ještě schopen plnit funkci regulace teploty. V takovém scénáři postačuje, aby zůstalo v provozu alespoň jedno měřicí zařízení (Měřicí zařízení 2) a jedno spínací zařízení. Předpokladem je, že systém měl alespoň jednou synchronizovaný čas. Situace je vyobrazena na obrázku níže. Nastavení systému lze stále provádět lokálně. K synchronizaci s cloudem dojde po opětovném připojení k internetu.



Obrázek 10 – Nejmenší možná funkční reprezentace systému (zdroj: autor)

Systém byl postupně realizován v následujícím pořadí:

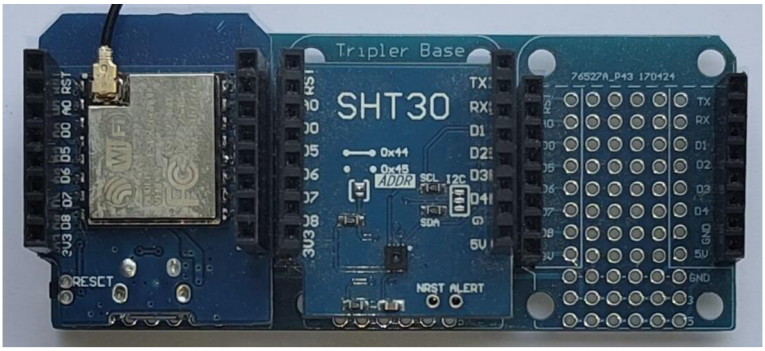
1) výroba zařízení (měřicí a spínací),
2) společný firmware pro měřicí a spínací zařízení,
3) webové rozhraní pro uživatele,
4) softwarové řešení pro cloud.

V následujících podkapitolách jsou jednotlivé části systému popsány detailně.

8.2 Výroba zařízení

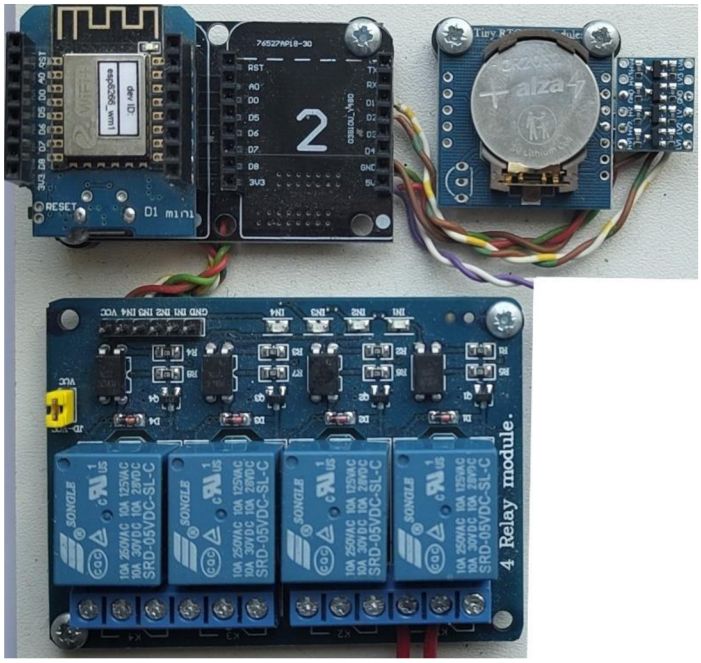
Realizace systému začala výrobou zařízení. Nejdříve bylo nutné vybrat veškeré komponenty k jejich výrobě. Následně byly zakoupeny všechny elektronické součástky na zahraničním e-shopu z důvodu nižší pořizovací ceny. Po obdržení byl proveden funkční test všech součástek. Po úspěšném otestování byly sestaveny prototypy třech měřicích zařízení a jednoho spínacího zařízení. Ukázky realizovaných zařízení jsou na následujících dvou obrázcích.

Měřicí zařízení obsahuje tepelné čidlo SHT30 spolu s ESP8266EX na společné základně.



Obrázek 11 – Prototyp měřicího zařízení (zdroj: autor)

Spínací zařízení obsahuje desku se čtyřmi relátky, hodiny reálného času s baterií a ESP8266EX.



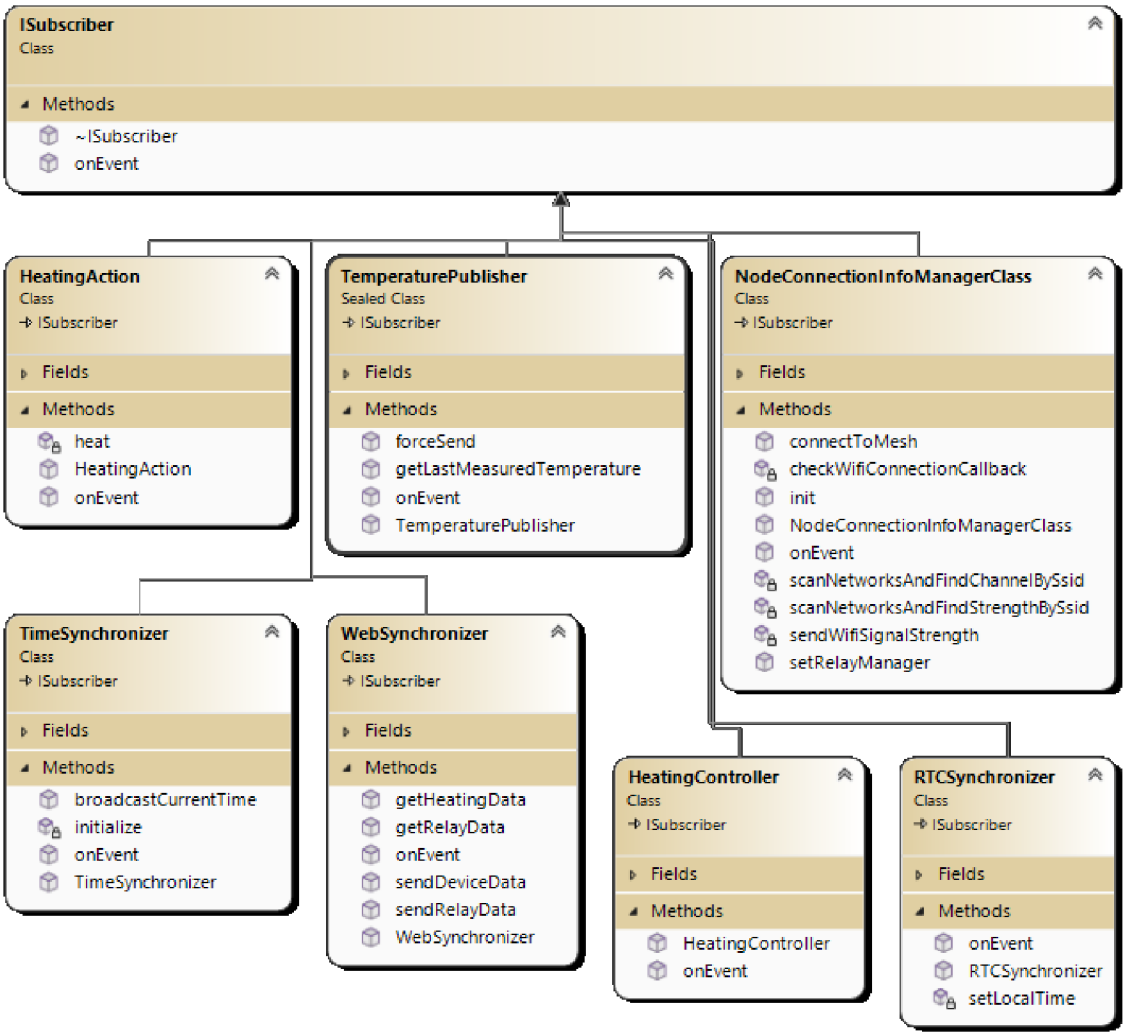
Obrázek 12 – Prototyp spínacího zařízení (zdroj: autor)

8.3 Vývoj firmware pro měřicí a spínací zařízení

Do měřicího a spínacího zařízení byl realizován společný firmware. Jeho vývoj probíhal postupnými iteračními kroky:

1) implementace datových struktur,
2) komunikace pomocí smíšené topologie,
3) připojení k wifi,
4) časová synchronizace pomocí NTP,
5) implementace měření,
6) implementace spínaní,
7) uložení konfigurace
8) synchronizace konfigurace,
9) webové rozhraní,
10) synchronizace s cloudem,
11) časová synchronizace pomocí RTC.

V rámci celého firmware byly vybrány nejdůležitější třídy a jsou vyobrazeny formou diagramu tříd na obrázku níže. Všechny třídy dědí z ISubscriber, která slouží jako rozhraní pro přijímání zpráv. Provozní informace se stavem všech zařízení v síti zajišťuje NodeConnectionInfoManagerClass. Synchronizaci mezi lokální sítí a cloudem zajišťuje WebSynchronizer. Klíčové třídy pro systém topení jsou HeatingAction, TemperaturePublisher a HeatingController. Synchronizaci času zabezpečují třídy TimeSynchronizer a RTCSynchronizer.



Obrázek 13 – Diagram tříd s jednotlivými metodami (zdroj: autor)

Třída NodeConnectionInfoManager uchovává časový plán topení. Časový plán obsahuje v datové struktuře HeatingProgram atributy dle následující tabulky, které jsou platné po dobu sedmi dní (týden).

Tabulka 7 – Definice časového plánu topení (zdroj: autor)

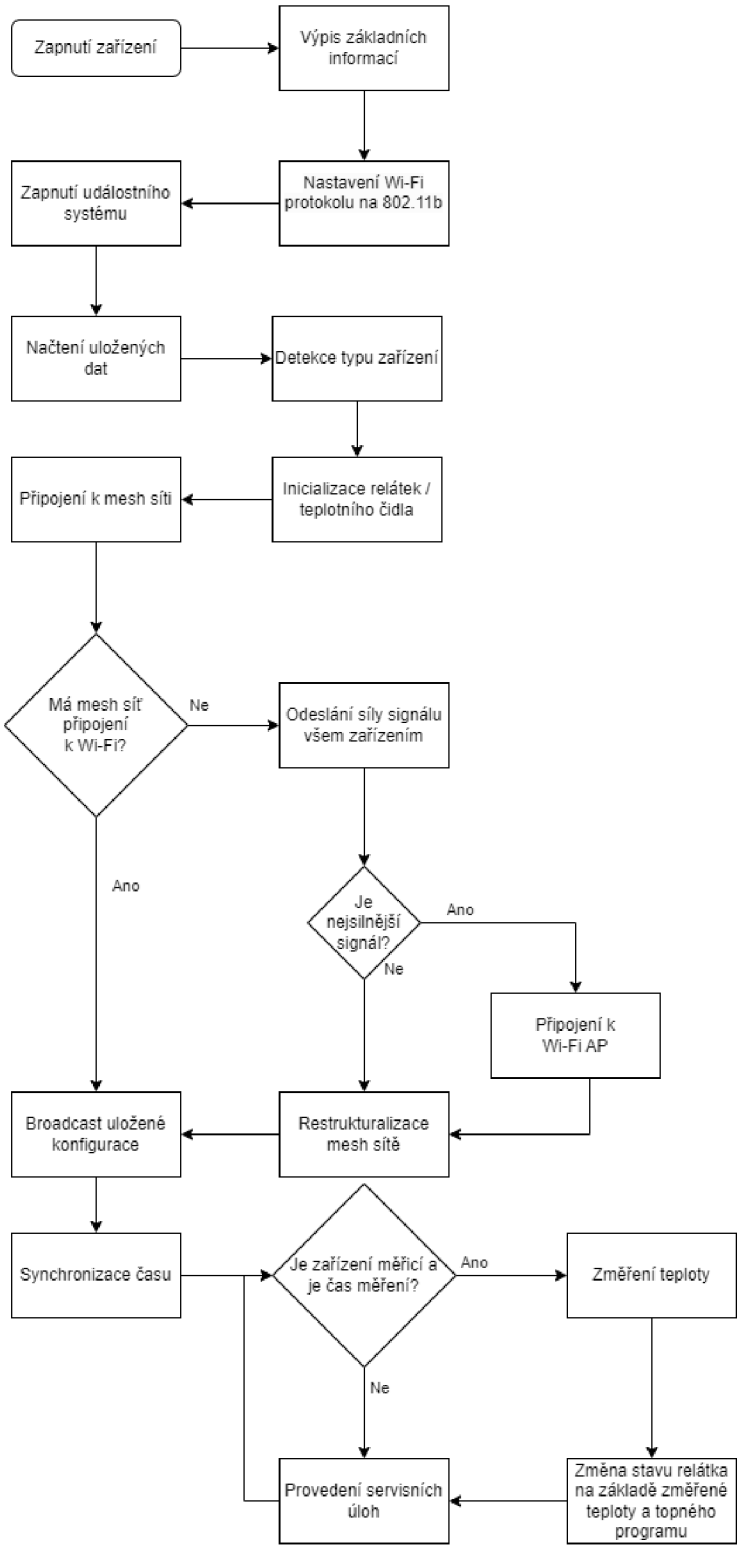
Název atributu	Datový typ	Velikost datového typu
highTemperature	float_t	32 bitů
lowTemperature	float_t	32 bitů
dayBytes	uint32_t [7]	32 bitů * 7

V celém projektu jsou využity následující knihovny, které jsou sepsané pro přehled v následující tabulce. Všechny knihovny jsou veřejně dostupné s otevřeným zdrojovým kódem, je tedy možné si je stáhnout a libovolně upravit. Je však nutné dávat pozor na jejich licence, aby nedošlo k jejímu porušení při publikaci úpravy nebo například komerčnímu využití. Celkově se jedná pouze o pět rozšiřujících knihoven.

Tabulka 8 – Použité knihovny u firmware (zdroj: autor)

Knihovna	Popis	Odkaz
WEMOS_SHT3X	knihovna pro obsluhu teplotního čidla	https://github.com/wemos/WEMOS_SHT3x_Arduino_Library
painlessMesh	knihovna pro správu mesh sítě	https://gitlab.com/painlessMesh/painlessMesh
TaskScheduler	knihovna pro kooperativní multitasking	https://github.com/arkhipenko/TaskScheduler
NTPClient	knihovna pro dotazování NTP času	https://github.com/arduino-libraries/NTPClient
RTClib	knihovna pro práci s hodinami reálného času	https://github.com/adafruit/RTClib

Celý firmware se skládá z jednotlivých menších modulů, kde každý plní svůj účel dle typu zařízení. Graficky a sekvenčním diagramem není vhodný pro znázornění, proto byl zvolen zjednodušený vývojový diagram. Detaily lze získat ze zdrojového kódu.



Obrázek 14 – Zjednodušený sekvenční diagram IoT zařízení v systému (zdroj: autor)

Ve firmware je aplikován kooperativní (nepreemptivní) multitasking. Kontext je předáván přes moduly a obslužnými rutinami.

Nepreemptivní plánování je jeden ze dvou způsobů plánování úloh. Je založeno na principu, že úloha, která aktuálně na procesoru vykonává svůj kód, tak má nad ním plnou kontrolu a nemůže být sama od sebe přerušena. K běhu další úlohy může dojít pouze ve dvou případech. V prvním případě, když úloha úspěšně ukončí svůj běh. V druhém případě, když úloha dobrovolně přeruší a odevzdá kontrolu, například z důvodu čekání na pomalá data. Ovšem nechtěným následkem může být, že pokud běžící úloha selže, může dojít k úplnému zastavení celého systému. Výhodou tohoto plánování také je, že se zde nemusí řešit synchronizace zdrojů, protože může běžet pouze právě jedna úloha v jeden moment. Nevýhodou také je, že pokud chceme, aby využití systémových zdrojů bylo lepší, tím je myšleno třeba běh dvou a více úloh současně, bylo by nutné použít preemptivní plánovač. Ten s sebou přináší ale i zvýšení komplexity a zároveň i režie, protože časté přepínání kontextů mezi úlohami je zdrojově a časově náročná aktivita. Čím častěji by se tato změna kontextu prováděla, tím méně času by zbylo na samotný běh aplikačního kódu. [11] Vzhledem k tomu, že měření a odesílání teploty není kriticky náročná aktivita a zpoždění v nejhorším případě i několika sekund se nejeví jako zásadní problém, tedy preemptivní plánování by přidalo zbytečnou komplexitu navíc.

8.3.1 Popis složek ve zdrojovém kódu firmware

Jelikož se jedná o složitější firmware, je proto rozdělen do několika složek (v C++ filtrů), které je opticky i logicky rozlišují pro zlepšení přehledu programátora.

8.3.2 Složky projektu společného firmware

Celkově je osm složek, které rozdělují projekt do svých podružných částí. Následuje popis jednotlivých složek.

Složka helpers

Složka helpers obsahuje pomocné třídy. V tomto projektu je pouze jedna třída, kterou je RollingAverage. Přeloženo do češtiny se jedná o klouzavý průměr.

Tato třída byla implementována tak, aby při měření nedošlo k nechtěnému chování v případě, že by bylo teplotní čidlo dočasně negativně ovlivněno neplatnou teplotou (např. průvan) zrovna, když by došlo k měření v ten daný moment. Parametrově lze zvolit, kolik hodnot si bude třída pamatovat pro výpočet průměru. Limitem je velikost volné paměti, ale vysoké hodnoty by v tomto případě neměly smysl. Zároveň nejmenší hodnota klouzavého průměru dává smysl až od dvou hodnot. V projektu je nastavena hodnota pět. Výsledek se tedy bude skládat z průměru posledních pěti hodnot. Pokud dojde k výpadku čidla a hodnota se nezměří, tak se z průměru nejstarší hodnota neodebírá. Interval, z jak dlouhé doby budou tyto hodnoty počítány je určen četností měření v jiné třídě. Žádná obrana proti výpadku není implementována. Pokud by se v praxi a při testování prokázalo, že četnost výpadků je vysoká, bude nutné čidla vyměnit nebo upravit algoritmus tak, aby po několika výpadcích topení vypnul a nahlásil chybu.

Složka IO

Složka IO, jak již z názvu vyplývá, obstarává vstupy a výstupy. V aktuálním stavu projektu je zde jedna třída. Flash Manager zaručuje, že je možné číst a psát do flash paměti zařízení. Firmware využívá pro čtení a zápis souborový systém LittleFS. Zároveň je k němu implementovaná logika přímo pro dva druhy ukládaných a načítaných dat – data o relátkách a data o všech členech systému. Pro zajímavost informace o jednom zařízení zabírá 64 bajtů. To obsahuje jeho unikátní id, název, vybrané relátko, plán vytápění a časovou značku vzniku této informace (epocha, tj. počet sekund od začátku roku 1970). Informace o relátkách zabírá 116 bajtů. Obsahuje počet relátek, časovou značku vytvoření a informaci až o šesti relátkách.

Složka heating

Složka heating se stará o dvě zásadní věci. Jedním je měření samotné teploty a následné odeslání hodnoty do systému.

Složka system

Ve složce system jsou třídy starající se o udržování aktuálních provozních dat celého systému za jeho běhu. Příkladem mohou být informace o relátkách nebo

kompletní informace o topných programech. Zároveň jsou zodpovědné za jejich správnou aktualizaci v případě příchodu nových dat. Také je zde uložena statická konfigurace celé mesh sítě.

Složka protocol

Složka protocol obsahuje z hlediska celého systému nejzajímavější zdrojový kód. Jednou z částí je definice lokální publish-subscribe (publikuj-odebírej) návrhového vzoru pro komunikaci mezi třídami. To umožní jednoduchou architekturu tříd, protože nebude nutné do tříd předávat mnoho závislostí, ale jenom pouze třídu umožňující odebírat zprávy. Aby vůbec bylo možné nějaké zprávy posílat, je také nutné navrhnout a implementovat jejich definici, což je další z věcí, kterou lze zde nalézt. A asi nejdůležitější částí je definice přenášených síťových zpráv včetně serializace a deserializace. Vzhledem k limitaci přenášených dat, kdy je obsahem zprávy textový řetězec zakončený prázdným znakem, používá se kódování Base64, což má bohužel za následek zvětšení přenášené zprávy. Ale jelikož nejsou zprávy nikdy větší než nižší stovky bajtů, není to zásadní problém.

Složka relay

Složka relay obsahuje logiku ke spínání relátek. To znamená definici jednotlivých relátek, tj. jejich identifikátor, příslušný pin, název, inverzní chování a aktuální status. Tyto vlastnosti jsou přebrány z konfigurace celého systému. Zároveň je zde obslužná procedura, která poslouchá příkazy na změnu stavu relátka (zapnuto, vypnuto) a podle toho jednotlivá relátka spíná (pokud se jedná o validní identifikátor). Tato část je aktivní pouze na spínacím zařízení, jelikož v případě měřicího zařízení nedává smysl mít tuto část aktivní.

Složka time

Udržení správného času napříč celým systémem je jedna z těch nejdůležitějších částí. Synchronizace času je rozdělena do tří částí:

- NTP synchronizace,
- RTC synchronizace,
- lokální synchronizace.

Každá z těchto tříd má na starost jednu z dílčích synchronizací. V případě dostupnosti internetu na jednom ze zařízení, je čas pravidelně synchronizován právě přes něj. Pokud internet není vůbec v celém systému dostupný, je synchronizováno pomocí RTC (hodin reálného času). Když nejsou dostupné hodiny reálného času „a ani není dostupné spínací zařízení, tak je celý systém nefunkční.

Jednotlivé části jsou aktivní dle typu zařízení a zároveň podle toho, jestli je aktuální zařízení připojené do lokální sítě s přístupem k internetu.

Složka web

Ve webové složce se nachází dvě velmi důležité funkce z hlediska splnění požadavků. Jednou z nich je webové rozhraní umožňující konfiguraci celé sítě z jednoho bodu. Zároveň toto webové rozhraní je společné pro celou síť ve všech verzí firmware, tedy uživatel není nikdy překvapen a zároveň ani nemusí vědět, přímo z jakého zařízení konfiguruje.

Druhou, také velmi podstatnou částí je synchronizace konfigurace z cloudu. Jelikož je zde riziko, že cloud nemusí být dostupný, není proto použito plně duplexní trvalé komunikace. Místo toho je zvolena jednodušší a to half-duplex za použití http protokolu. Je zde zároveň veliká pravděpodobnost, že koncová zařízení budou schovaná v intranetu za technologií NAT, nebude tedy možné vytvořit připojení z internetu do sítě k zařízení. K obejití tohoto problému je použito takzvaného opakovaného dotazování v pevných intervalech (anglicky http polling). Propagace konfigurace je pak automatická v obou případech.

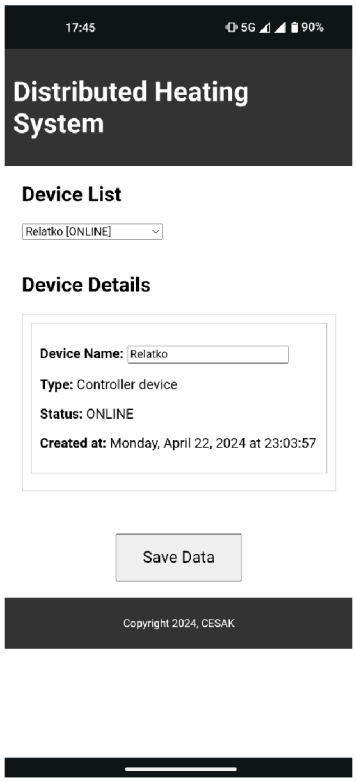
8.4 Realizace webového rozhraní

Úspěšnost a použitelnost systému topení závisí na kvalitě UI a UX části pro konfiguraci topení. Implementace byla vedena návrhovým stylem „mobile first“ s tím, že systém má být co nejjednodušší na obsluhu a co nejpřehlednější. Výsledná konfigurační stránka je vyobrazena na snímcích obrazovky:

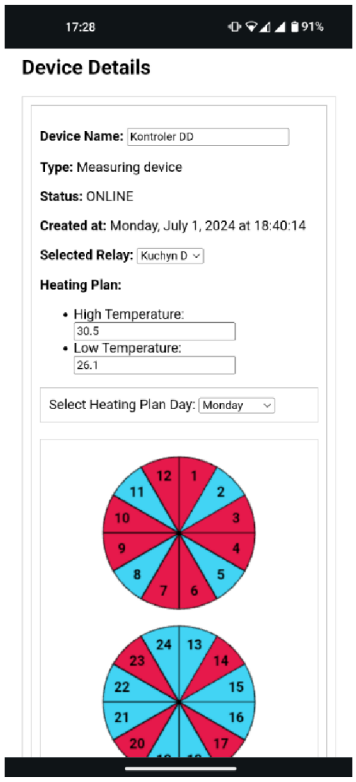
- Obrázek 15 – Nastavení spínacího zařízení v systému na mobilním zařízení (zdroj: autor),
- Obrázek 16 – Nastavení měřicího zařízení v systému na mobilním zařízení (zdroj: autor),
- Obrázek 17 – Nastavení měřicího zařízení v systému na stolním zařízení (zdroj: autor),
- Obrázek 18 – Nastavení zařízení v systému, jiný den (zdroj: autor).

Hodnoty nastavené na snímcích jsou pouze pro účely k otestování systému. Nejsou vhodné pro reálný provoz.

Konfigurační stránka funguje následujícím způsobem. Nejdříve se uživatel přihlásí (přes basic auth ze strany webového serveru). Následně si z rozbalovacího seznamu uživatel vybere zařízení. Už přímo v seznamu je možné vidět, která zařízení jsou online a která ne. Je zde prodleva několika minut, způsobeno stylem synchronizace. Není implementována žádná oboustranná komunikace v reálném čase pro minimalizaci přenosu dat. Po vybrání zařízení se stránka rozšíří a vyplní informace o konkrétním zařízení. V případě, že se jedná o spínací zařízení, lze zde nastavit pouze název zařízení, stav (online, office) a čas, kdy tato konfigurace byla naposledy upravena. Pokud se jedná o zařízení měřicí, načte se komplexnější stránka, ve které lze nastavit dvě teploty: vysoká a nízká, podle kterých se bude určovat topení. Dále se objeví rozbalovací nabídka s výběrem dní v týdnu. Po načtení se automaticky vybere pondělí. Pod touto nabídkou se objeví dvoje analogové hodiny, každé pro dvanáct hodin dne, 1–12 a 13–24. Každá hodina značí sektor, u kterého lze ovlivnit teplotu topení. Pokud je sektor modrý, topí se podle nízké teploty, pokud je červený topí se podle vysoké teploty. Každý den a každé zařízení je rozlišeno. Na konci stránky je pak už jenom tlačítko, které slouží k uložení dat. To ukládá všechny upravené hodnoty od posledního načtení, to znamená, že lze mezi zařízeními a dny libovolně přepínat (pokud se neobnoví stránka) a uložit vše najednou.

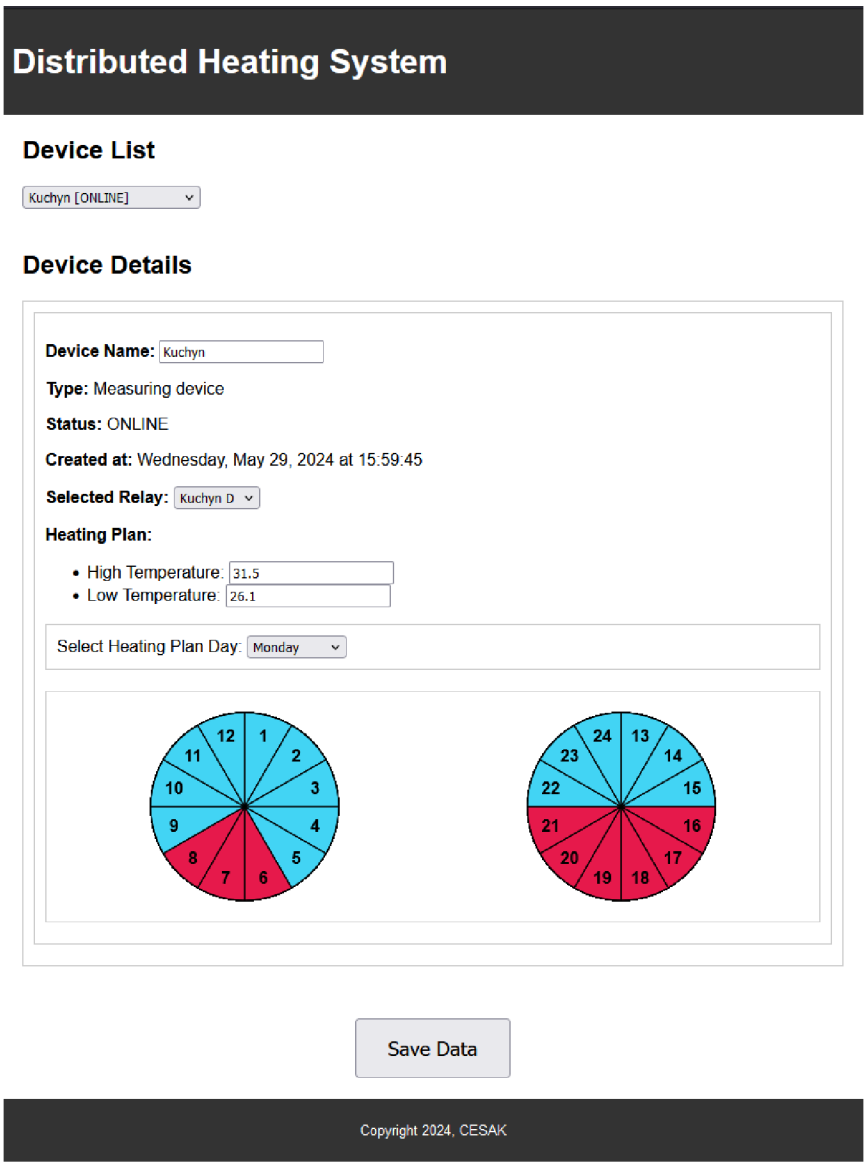


Obrázek 15 – Nastavení spínacího zařízení v systému na mobilním zařízení (zdroj: autor)



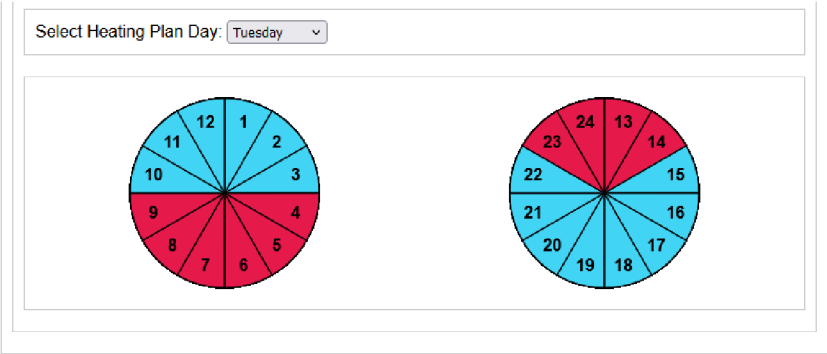
Obrázek 16 – Nastavení měřicího zařízení v systému na mobilním zařízení (zdroj: autor)

Na obrázku níže je zobrazen časový plán pro kuchyň. Na vysokou teplotu 31,5 °C se bude topit v pondělí v ranních hodinách od šesti do osmi hodin a odpoledne od čtyř do devíti hodin. Na nízkou teplotu 26,1 °C se bude topit v ostatních hodinách.



Obrázek 17 – Nastavení měřicího zařízení v systému na stolním zařízení (zdroj: autor)

Při volbě jiného dne v týdnu se načte nastavení zvoleného dne. Na níže uvedeném obrázku je zvoleno úterý, kdy na vysokou teplotu se bude topit v čase od čtyř do devíti hodin ráno, od třinácti do čtrnácti hodin odpoledne a večer od jedenácti hodin do půlnoci.



Obrázek 18 – Nastavení zařízení v systému, jiný den (zdroj: autor)

8.5 Realizace cloudové části systému

Nejdříve je nutné si zvolit cílový server, kam se nainstaluje operační systém. Operační systém byl zvolen Debian ve své nejnovější verzi 12. Poskytovatelem virtuálního serveru byla zvolena společnost Orelsoft s jedním z jejich nejlevnější VPS. Jeho parametry jsou:

- 1 sdílené CPU jádro,
- 1024 MiB operační paměti,
- 10 GiB perzistentního uložiště na rotačním disku,
- 1 veřejná IPv4 adresa.

Celý projekt je postaven na Javascriptové platformě Node.js, díky které byl vývoj jednoduchý a hlavně rychlý. Jelikož není dobrý způsob zveřejňovat takto běžící služby přímo do internetu, je použitý webový server Apache HTTP Server jako reverzní proxy pro větší zabezpečení a zároveň i částečné oddělení. Z důvodu jednoduchosti aplikace není použita kontejnerizační platforma, např. Docker.

Mezi použité knihovny (závislosti) pro projekt v rámci Node.js jsou:

- express – webový server,
- fs – modul obsluhující souborový systém,
- path – modul zjednodušující práci s cestami v souborovém systému.

V projektu je implementován jednoduchý REST model pro získávání a publikování dat z cloudu. Konkrétně se jedná o čtyři obslužné metody. Dvě metody pro získávání dat (metoda GET) a dvě metody pro aktualizaci (metoda POST). Každé pro relátka a zbytek vytápěcího systému. Zabezpečení jako takové není v aplikaci implementováno a spoléhá se na reverzní proxy.

Analýzou struktury dat uložených v cloudu lze nalézt dva jednoduché JSON soubory. Ve standardním režimu nejsou tak dobře čitelná, protože se nachází na jednom řádku, ale z důvodu zlepšení čitelnosti byl kód upraven. V prvním souboru jsou uloženy tak, jako v samotných zařízeních.


```
{
    "nodeId": 540981609,
    "isOnline": true,
    "nodeType": 1,
    "persistentInfo": {
        "name": "Ovladac dvojky",
        "createdAt": 1714043457,
        "selectedRelay": 2,
        "heatingPlan": {
            "highTemperature": 29.10000038,
            "lowTemperature": 20.10000038,
            "dayBytes": [
                983152,
                4294967295,
                4294967295,
                4294967295,
                4294967295,
                4294967295,
                4294967295
            ]
        }
    }
}
```


Obrázek 19 – Struktura dat v JSON pro jedno měřicí zařízení (zdroj: autor)

Druhým souborem jsou informace o relátkách, jejich názvech a přiřazených špičkách. V ukázce níže lze vidět čtyři relátka z pokoje, ložnice, obýváku a kuchyně.

Každé z nich má unikátní identifikátor a svoji špičku, která odpovídá relátku. Zároveň je zde i časová značka 1714048181, která odpovídá 25. dubnu 2024 14:29:41. Atribut pin, který je u každého záznamu uveden říká, jaký z univerzálních vstupů a výstupů na modulu je přiřazen k právě dané špičce.


```
{
  "createdAt": 1714048181,
  "relays": [
    {
      "id": 1,
      "name": "Pokoj A",
      "pin": 13
    },
    {
      "id": 2,
      "name": "Loznice B",
      "pin": 12
    },
    {
      "id": 3,
      "name": "Obyvak C",
      "pin": 14
    },
    {
      "id": 4,
      "name": "Kuchyn D",
      "pin": 16
    }
  ]
}
```


Obrázek 20 – Struktura dat v JSON pro jedno všechna relátka v systému (zdroj: autor)

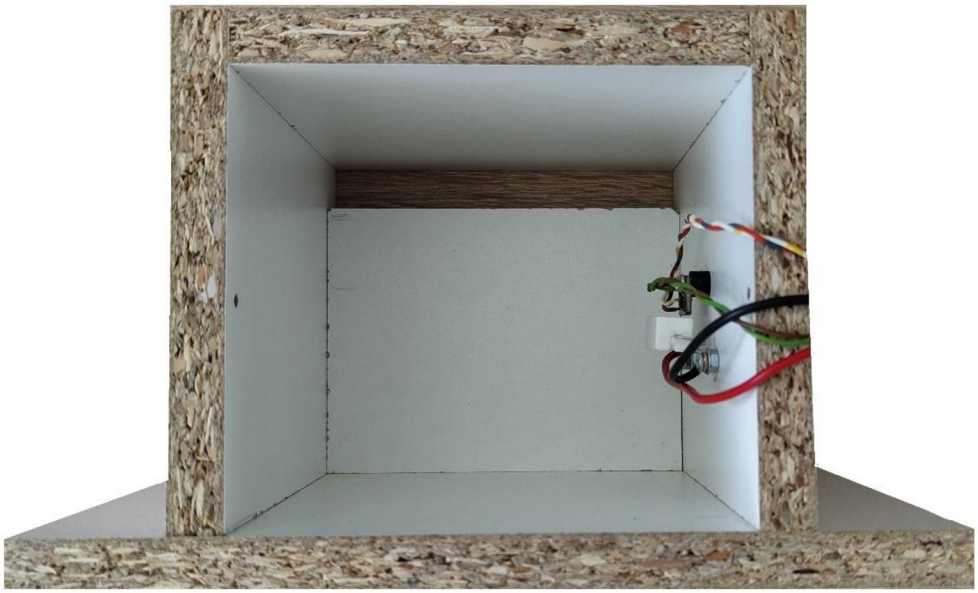
9 Demonstrace funkčnosti systému

Postup demonstrace funkčnosti systému je složen z jeho sestavení, konfigurace a následného ověření funkčnosti. Úspěšnost demonstrace systému je prokázána v případě, že systém je funkční dle konfigurace. Navíc musí být systém v souladu s dříve stanovenými požadavky, kterými se potvrdí splnění cíle práce.

9.1 Základní parametry demonstrace systému

Test byl proveden v reálně nasaditelném režimu s tím rozdílem, že zařízení byla rozmístěna v menším počtu místností. K simulaci topení v místnosti je sestavena miniatura místnosti z dřevotřísky, do které je umístěn odpor spínaný relátkem ze spínacího zařízení. Dále je instalováno v této miniatuře i teplotní čidlo napojené k měřicímu zařízení. Aby systém nebyl tak malý, tak do dalších dvou místností byly náhodně přidány ještě další dvě měřicí zařízení. Ty budou měřit, ale topení budou mít nastavené tak, že ke spínaní žádného relé nedojde. Jde o to si vyzkoušet, jak se systém bude chovat s optimálním počtem zařízení v téměř reálné podobě (až na vzdálenost testovaného měřicího zařízení). Pokud by mělo dojít k ostrému testování, byla by potřeba topná sezóna. Případně se testy mohou v té době nezávisle zopakovat.

V obrázku přiloženém níže je možné vidět model místnosti, na které byl systém otestován. Při testování byla přední část podobně zakryta, tak jako její zadní část, ale pro pořízení fotografie byla odebrána. Vnitřní rozměry jsou 15 x 15 x 15 cm s dvěma otvory pro zrychlení změny teploty. Všechny důležité komponenty jsou na pravé vnitřní stěně. K odporu jsou přivedeny dva dráty, černý a červený. Připojeny jsou k relátku na spínacím zařízení. U měřicího čidla jsou kabely čtyři. Dva napájecí a dva datové pro I^2C sběrnici. Ty jsou připojeny k měřicímu zařízení.



Obrázek 21 – Testovací model místnosti (zdroj: autor)

9.2 Test připojení nového zařízení

U testu připojení spočívá princip testu v tom, že se připraví a naprogramuje úplně nové zařízení (ať už měřicí nebo spínací). To se následně zapne a sleduje se, jak se zařízení bude v systému chovat a jestli dojde „k synchronizaci konfigurace, časů a jestli se projeví změna i v cloudu.

Tento test byl proveden tak, že byl vybrán náhodný ESP8266EX modul bez žádných periférií, naprogramován a spuštěn. Jelikož na něm není připojeno čidlo teploty, je detekováno jako zařízení spínací. Záznam chování je možné sledovat v textu níže, který je výpisem do sériové konzole.

Z výpisu lze vidět, že nejdříve dojde k vypsání základních informací. Nejdříve verze firmware (což je hash – česky otisk firmware pomocí algoritmu SHA-1). Zde nevadí, že se jedná o již prolomenou hashovací funkci, protože zde nezajišťuje žádné zabezpečení. Dále vidíme samotné datum sestavení firmware, unikátní identifikátor v rámci lokální mesh sítě a fyzickou (MAC) adresu modulu. Vypíše se druh zařízení, počet uložených zařízení (zde je správně nula), nakonec IP adresy na wifi rozhraní (klient a stanice) a k tomu i DNS. Zde jsou prázdné, protože zařízení není nikam připojené. Poté je možné vidět připojení se k již běžícímu zařízení, kdy dojde k automatické synchronizaci všech uložených zařízení a relátek. Zároveň lze na konci vidět, že došlo i k synchronizaci času ze sítě.

```
Version: 06be85d2993d805238b125cdd4cbe242e9172e75
Build date: 2024-07-13 11:21:09
Mesh ID: 2827250357
MAC Address: 84:CC:A8:84:6A:B5
Mode: Control (Relays)
Found 0 of saved nodes
Executing routine tasks
Soft AP IP:10.106.181.1
Client IP:(IP unset)
DNS:(IP unset)
--> New Connection, nodeId = 3157253988
--> Number of connected nodes: 2
Received PersistentNodeInformation for 3157253988: Success? 1 It was from
3157253988
NodeID: 3157253988
Saved data to flash Success? 1 count:3
Received PersistentNodeInformation for 540981609: Success? 1 It was from
3157253988
NodeID: 540981609
Saved data to flash Success? 1 count:3
Received PersistentNodeInformation for 3208229378: Success? 1 It was from
3157253988
NodeID: 3208229378
Saved data to flash Success? 1 count:4
Received PersistentNodeInformation for 1418599901: Success? 1 It was from
3157253988
NodeID: 1418599901
Saved data to flash Success? 1 count:5
Writing file: /relay.dat
File written
Synced time from MESH: Sat Jul 13 17:13:41 2024
```

Obrázek 22 – Záznam z připojení nového zařízení do systému (zdroj: autor)

9.3 Test lokální synchronizace zařízení

Test lokální synchronizace zařízení probíhal ve stejném nastavení, jako skončil v předchozím testu. Jeho cílem bylo zjistit, jestli se změna konfigurace zpropaguje z jednoho zařízení do celé sítě. Jelikož je konfigurace již synchronizovaná, nedochází k jejímu přepsání. Proto byl použit webový prohlížeč na počítači v lokální síti. Byla zadána IP adresa zařízení připojeného do lokální sítě. Jedná se o jiné než to nově připojené. K výběru dochází dle síly signálu. Následně došlo k úpravě topícího programu u zařízení č. 540981609 (uložen jako LozniceOvladac). Po potvrzení došlo k uložení dané konfigurace a následně bylo zpropagováno do celé sítě, což lze vidět z výpisu níže. Skutečně byla upravena a zapsána jenom jedna změna, a to právě pro upravené zařízení. To je z toho důvodu, aby došlo k co nejmenšímu opotřebení perzistentní paměti.

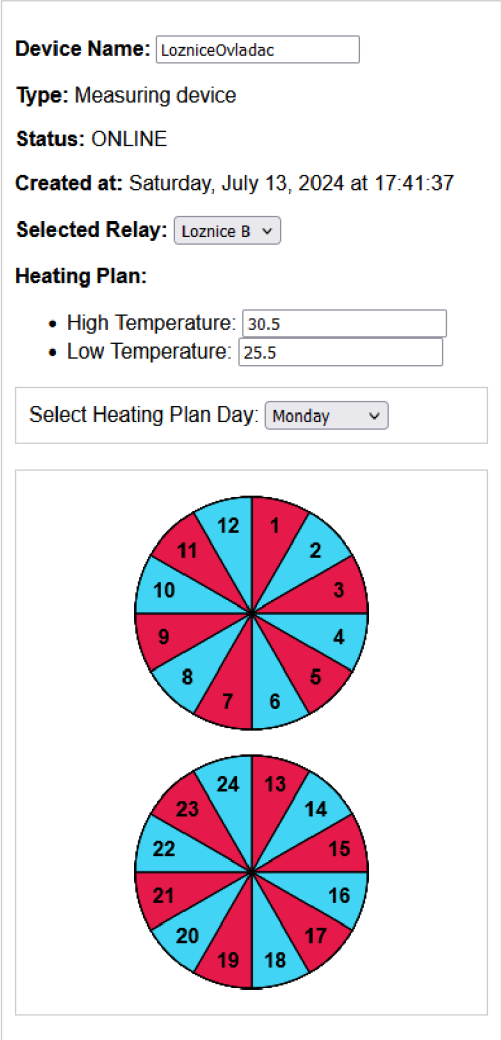
Test je tedy možné z hlediska jeho popisu prohlásit za úspěšný, protože došlo k synchronizaci konfigurace, a to právě jenom toho zařízení, které bylo upravené.

```
Got heating information
Processing node 2827250357
Received PersistentNodeInformation for 2827250357: Success? 0 It was from
2827250357
Processing node 540981609
Sending heating program
Received PersistentNodeInformation for 540981609: Success? 1 It was from
2827250357
NodeID: 540981609
Writing file: /persistent.dat
File written
Writing file: /length.dat
File written
Saved data to flash Success? 1 count:5
Processing node 3157253988
Received PersistentNodeInformation for 3157253988: Success? 0 It was from
2827250357
Processing node 3208229378
Sending heating program
Received PersistentNodeInformation for 3208229378: Success? 0 It was from
2827250357
Processing node 1418599901
Sending heating program
Received PersistentNodeInformation for 1418599901: Success? 0 It was from
2827250357
DONE
```

Obrázek 23 – Záznam o změně konfigurace zařízení z lokální sítě (zdroj: autor)

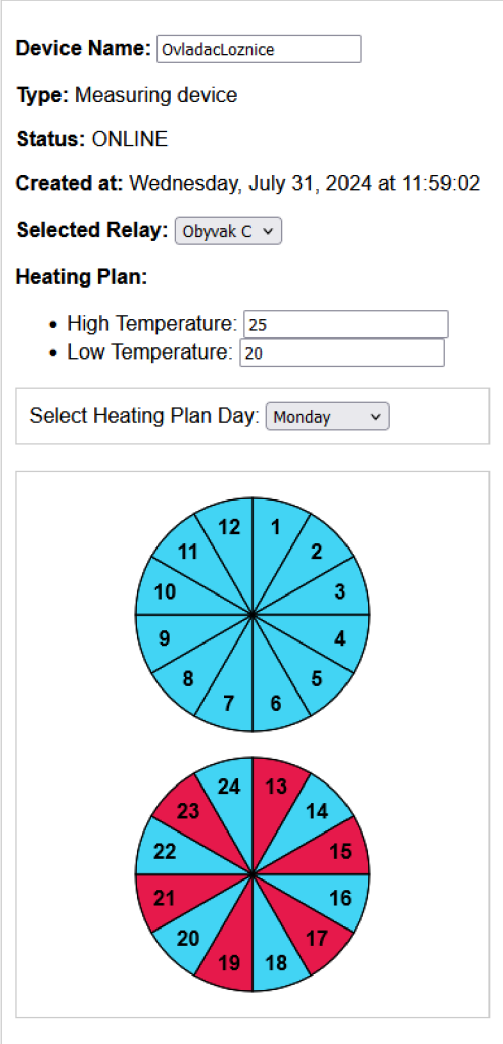
9.4 Test nastavení konfigurace v cloudu

Test nastavení konfigurace v cloudu je složen z několika jednoduchých kroků. Předpokladem je nasazená aplikace v cloudu, která je přístupná z internetu. K otestování vybereme zařízení s názvem „LozniceOvladac“. Jeho aktuální konfigurace je následující.



Obrázek 24 – Původní konfigurace „LozniceOvladac“ v testu (zdroj: autor)

Následně je provedena změna všech parametrů. Název je změněn na „OvladacLoznice“, relátko vyměněno za libovolné jiné. Vysoká teplota je nastavena na 25 °C a nízká na 20 °C. U časového plánu je provedena změna jenom v pondělí v prvních dvanácti hodinách. Všechny hodnoty jsou nastaveny na požadovanou nízkou teplotu, která odpovídá modré barvě. Po stisknutí tlačítka Save Data je požadavek odeslán na server. To lze sledovat z vývojářské konzole prohlížeče. Odpověď na požadavek je 200 (OK). Po obnovení stránky bez mezipaměti je opět vybráno to samé zařízení s novým názvem „OvladacLoznice“, které již obsahuje upravené hodnoty.



Obrázek 25 - Nová konfigurace „OvladacLoznice“ v testu (zdroj: autor)

Po analýze stránky lze vyhodnotit, že test byl úspěšný. Za povšimnutí stojí, že se změnila hodnota „Created at“, která opravdu souhlasí s časem, kdy došlo k uložení nové konfigurace.

9.5 Test synchronizace konfigurace s cloudem

Test synchronizace spočívá v tom, že po úspěšném předchozím testu, kdy došlo ke změně konfigurace v cloudu u jednoho zařízení. Je otestováno, zdali se propíše i do lokálních zařízení. Předpokladem samozřejmě je, že internet i cloud jsou při testování dostupné.

Synchronizace u zařízení probíhá automaticky a pravidelně. Z výpisu zařízení připojeného k internetu lze zjistit, že se nejdříve zařízení pokusí data

odeslat, což projde úspěšně vzhledem k tomu, že stavový kód byl 200. Po úspěšném odeslání dat dochází také ke stáhnutí nejnovějších dat. Z výpisu je patrné, že došlo k úpravě a aktualizaci zařízení s id 2827250357, u ostatních zůstala konfigurace stejná. Výpis byl zkrácen tak, aby se lépe vešel do textu. U relátek nedošlo k žádné změně. Lze tedy vidět, že test proběhl úspěšně.

```
Sending current data to cloud
Tried to connect to http://hydra.cesak.com/relays. Success? TRUE
Status code: 200
Tried to connect to http://hydra.cesak.com/heating. Success? TRUE
Status code: 200
Receiving data from cloud
Tried to connect to http://hydra.cesak.com/heating. Success? TRUE
Status code: 200
Got heating information
---- data zkracena ---
Processing node 2827250357
Received PersistentNodeInformation for 2827250357: Success? 1 It was from
2827250357
NodeID: 2827250357
Saved data to flash Success? 1 count:5
Processing node 3208229378
Sending heating program
Received PersistentNodeInformation for 3208229378: Success? 0 It was from
2827250357
DONE
Tried to connect to http://hydra.cesak.com/heating. Success? TRUE
Status code: 200
Got relay data
Created at: 1714048181
Array length: 4
```

Obrázek 26 – Záznam ze synchronizace zařízení a cloudu (zdroj: autor)

9.6 Test funkčnosti topení

K ověření celkové funkčnosti topení je provedeno nastavením plánu topení, při kterém dojde k otestování funkčnosti spínání. Nastavením se rozumí úprava požadované teploty, která je vyšší než aktuální teplota prostředí o pár stupňů Celsia. Spínací element je v tomto případě elektrický odpor, v jehož těsné blízkosti je umístěno teplotní čidlo. Výsledná teplota v tomto testu je klouzavý průměr pěti posledních naměřených hodnot. Firmware byl pro potřeby testu upraven tak, že teplota je měřena každých třicet sekund a spínání je aktivováno každou minutu. Níže jsou přiloženy dva výpisy. Jeden z měřicího zařízení, druhý ze spínacího zařízení. Nastavená nízká teplota pro vytápění byla rovných 34 °C pro všechny hodiny v týdnu. Zároveň všechny hodiny byly nastavené tak, aby topily na nízkou teplotu. To znamená, že se bude topit, pokud naměřená průměrná teplota bude menší než nastavených 34 °C.

```
Can't control. Time is not synced or temperature is not measured yet
Executing routine tasks
Synced time from MESH: Mon Jul 15 20:23:04 2024
Executing routine tasks
Avg. Temp: 33.994198, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting HEAT
Avg. Temp: 35.945145, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting COLD
Executing routine tasks
Avg. Temp: 40.796593, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting COLD
Avg. Temp: 37.341724, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting COLD
Avg. Temp: 33.028076, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting HEAT
Executing routine tasks
Avg. Temp: 32.928204, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting HEAT
Avg. Temp: 35.265282, DoW: 0, CuH: 20, RS: 0, RT: 34.000000
Requesting COLD
```

Obrázek 27 – Záznam měřicího zařízení ovládající topení (zdroj: autor)

```
Requested relay status change from 2827250357 for relay 1 (Pokoj A).
Setting it to: 1
Requested relay status change from 2827250357 for relay 1 (Pokoj A).
Setting it to: 0
Requested relay status change from 2827250357 for relay 1 (Pokoj A).
Setting it to: 1
Requested relay status change from 2827250357 for relay 1 (Pokoj A).
Setting it to: 0
```

Obrázek 28 – Záznam ze spínacího zařízení ovládající relátka (zdroj: autor)

Na výpisu spínacího zařízení je vidět, že se to projeví těmito hláškami. Co se týče výsledků, tak měřicí zařízení reaguje správně na topící program. Spínací zařízení správně reaguje na požadavky měřicího zařízení. Tedy tento test je úspěšný.

9.7 Test systému bez připojení k internetu

Test systému bez připojení k internetu je podmíněn tím, aby byl systém připojen k internetu alespoň jednou a zároveň, aby bylo v síti přítomné zařízení s hodinami reálného času. Tato část je splněna. Dále je nutné vyzkoušet, jestli systém naběhne a topení bude fungovat. Nejdříve se ověří funkčnost na zařízení s hodinami reálného času, které je zároveň spínací.

```
Version: 49f2a25ad1b718ded2443a1cf235a48393969e87
Build date: 2024-07-17 13:41:03
Mesh ID: 3157253988
MAC Address: EC:FA:BC:2F:DF:64
Mode: Control (Relays)
... vynechané řádky ...
Syncing time from RTC
current time: Wed Jul 17 13:47:34 2024
Executing routine tasks
Soft AP IP:10.223.100.1
Client IP:(IP unset)
DNS:(IP unset)
--> New Connection, nodeId = 2827250357
... vynechané řádky ...
--> Number of connected nodes: 1
... vynechané řádky ...
Broadcasting local time
--> Number of connected nodes: 2
--> New Connection, nodeId = 540981609
... vynechané řádky ...
Requested relay status change from 2827250357 for relay 1 (Pokoj A).
Setting it to: 0
Requested relay status change from 540981609 for relay 2 (Loznice B).
Setting it to: 1
```

Obrázek 29 – Záznam synchronizace času z RTC na spínacím zařízení (zdroj: autor)

Ze záznamu je patrné, že ihned po zapnutí dochází k synchronizaci času z RTC, i když není síť připojená k wifi. Po napojení dalších dvou měřicích zařízení je čas s těmito zařízeními sdílen. Následně po desítkách sekund dojde k ovládání topení. Dále je analyzován výpis z druhého zařízení, tentokrát měřicího.

```
Version: 49f2a25ad1b718ded2443a1cf235a48393969e87
Build date: 2024-07-17 13:41:03
Mesh ID: 2827250357
MAC Address: 84:CC:A8:84:6A:B5
Mode: Measurement (SHT3X)
... vynechané řádky ...
Can't control. Time is not synced or temperature is not measured yet
Executing routine tasks
... vynechané řádky ...
--> New Connection, nodeId = 3157253988
--> Number of connected nodes: 1
... vynechané řádky ...
Synced time from MESH: Wed Jul 17 13:47:39 2024
... vynechané řádky ...
--> Number of connected nodes: 2
... vynechané řádky ...
Executing routine tasks
Avg. Temp: 33.014545, DoW: 2, CuH: 13, RS: 0, RT: 34.000000
Requesting HEAT
```

Obrázek 30 – Záznam synchronizace a topení měřicího zařízení (zdroj: autor)

Po startu zařízení vypisuje, že nemůže ovládat topení, protože nemá synchronizovaný čas. Hned po připojení prvního zařízení (zde podle identifikátoru je možné říci, že se jedná o spínací zařízení, tedy zařízení se synchronizovaným časem) dojde k okamžité synchronizaci času. Prodleva je několik málo sekund. Od načtení informací z hodin reálného času na druhém zařízení až po připojení a synchronizaci zde uběhlo zhruba pět sekund. Zároveň zařízení začalo ovládat topení, jakmile došlo k synchronizaci času. Tedy tento test je také úspěšný.

9.8 Test odpojení jednoho měřicího modulu

V tomto testu bude otestováno, co se stane, pokud se odpojí měřicí zařízení a bude se zrovna topit. Očekávaný stav je, že je odpojení detekováno v celé lokální síti a spínací prvek následně relátko ovládané tímto vypne a zároveň bude fungovat dál zbytek topícího systému.

```
Requested relay status change from 2827250357 for relay 1 (Pokoj
A). Setting it to: 1
Executing routine tasks
Heating node 2827250357 disconnected. Turning off heating
--> Number of connected nodes: 1
--> Lost Connection, nodeId = 2827250357
Requested relay status change from 540981609 for relay 1 (Pokoj A).
Setting it to: 0
Syncing time from RTC
```

Obrázek 31 – Záznam změny stavu relátek při odpojení měřicího zařízení (zdroj: autor)

Z výpisu spínacího zařízení je jasně vidět, že nejdříve bylo topení sepnuto zařízením s identifikátorem 2827250357, což je zařízení měřicí. Následně bylo manuálně vypnuto. Po několika sekundách se zobrazil řádek „s informací, že se měřicí zařízení odpojilo a že se vypíná topení ze spínacího zařízení, jehož identifikátor je 540981609. Což značí, že test je také úspěšný.

9.9 Shrnutí všech testů

V předchozích podkapitolách 9.2 až 9.8 byly popsány různorodé testy na prokázání funkčnosti systému topení. Všechny testy dopadly pozitivně „a systém je funkční. Pro účely kontroly splnění všech požadavků jsou tyto níže zopakovány a validovány.

9.10 Zopakování požadavků a jejich splnění

V této kapitole jsou zopakovány požadavky vedené na systém, respektive ty, které byly rozvedené na dílčí části a automaticky zahrnují původní celosystémové.

První požadavky jsou na měřicí zařízení. U měřicího zařízení byly implementovány a splněny všechny požadavky bez jediné výjimky.

Tabulka 9 – Splnění požadavků měřicího zařízení (zdroj: autor)

Id požadavku	Popis požadavku	Splněn
POZ-MER-01	měřicí zařízení musí měřit teplotu místnosti	ANO
POZ-MER-02	měřicí zařízení musí mít dostatečně velké rozlišení měřené teploty	ANO
POZ-MER-03	měřená teplota se nesmí lišit od reálné v rozmezí 0,2 °C	ANO
POZ-MER-04	měřená teplota musí být získávána v intervalu kratším než 1 hodina	ANO
POZ-MER-05	měřicí zařízení musí být napájeno buď z baterie, akumulátoru nebo ze sítě	ANO
POZ-MER-06	měřicí zařízení musí být umístěno na nejoptimálnějším místě v místnosti	ANO
POZ-MER-07	měřicí zařízení musí být malé tak, aby se bezpečně dalo umístit na stěnu bez nutnosti složité montáže	ANO
POZ-MER-08	měřicí zařízení by mělo být lehké, optimálně do 150 gramů	ANO
POZ-MER-09	měřicí zařízení musí umožnit konfiguraci ze sebe samotného pro celý systém	ANO
POZ-MER-10	měřicí zařízení musí perzistentně ukládat data o systému	ANO
POZ-MER-11	měřicí zařízení může komunikovat pomocí wifi	ANO
POZ-MER-12	měřicí zařízení musí umět přenášet data sebe sama a ostatních zařízení v systému	ANO

Druhá sada požadavků je na spínací zařízení. Stejně jako u měřicího zařízení, zde byl splněn každý jednotlivý požadavek.

Tabulka 10 – Splnění požadavků spínacího zařízení (zdroj: autor)

Id požadavku	Popis požadavku	Splněn
POZ-SPI-01	spínací zařízení musí umět ovládat topící elementy	ANO
POZ-SPI-02	spínací zařízení musí mít prvek, se kterým bude spínat koncové prvky	ANO
POZ-SPI-03	spínací zařízení musí umět spínat alespoň tolik prvků, kolik má být ovládáno místnosti, minimálně však 3	ANO
POZ-SPI-04	spínané zařízení musí být podporované spínacím zařízením	ANO
POZ-SPI-05	spínací zařízení musí mít minimální prodlevu minimálně jednu minutu	ANO
POZ-SPI-06	spínací zařízení bude spínat na základě pokynů měřicího zařízení	ANO
POZ-SPI-07	spínací zařízení musí být napájeno ze sítě	ANO
POZ-SPI-08	spínací zařízení by mělo být co nejmenší na základě použitých komponent	ANO
POZ-SPI-09	měřicí zařízení by mělo být lehké, tak aby šlo jednoduše umístit ke spínaným prvkům	ANO
POZ-SPI-10	měřicí zařízení musí být co nejlevnější při splnění všech požadavků	ANO
POZ-SPI-11	spínací zařízení musí umožnit konfiguraci ze sebe samotného pro celý systém	ANO
POZ-SPI-12	spínací zařízení může komunikovat pomocí wifi	ANO
POZ-SPI-13	spínací zařízení musí umět přenášet data systému	ANO

Posledním seznamem požadavků jsou požadavky na cloud. Vzhledem k tomu, že se jednalo o velmi jednoduché a obecné požadavky, tak i tyto byly splněny.

Tabulka 11 – Splnění požadavků cloudu (zdroj: autor)

Id Požadavku	Popis požadavku	Splněn
POZ-CLD-01	cloud musí umožnit nakonfigurovat celý systém	ANO
POZ-CLD-02	cloud musí být dostupný z internetu	ANO
POZ-CLD-03	cloud musí trvale ukládat systémová data	ANO
POZ-CLD-04	cloud musí umět synchronizovat data se zbytkem systému, tedy zařízeními	ANO

Navržený a vyvinutý systém splnil všechny na něj kladené požadavky a zároveň byl prakticky otestován. Je možné prohlásit systém za vyhovující a funkční.

10 Závěr

U decentralizovaného IoT systému topení se podařilo identifikovat několik klíčových požadavků vycházejících z cíle práce, které byly rozvedeny do podrobnější podoby. V souladu s požadavky byly navrženy a vytvořeny tři dílčí prvky systému: měřicí zařízení, spínací zařízení a společný cloud. Každý prvek má jinou roli v systému. Měřicí zařízení slouží k měření teploty místnosti. Spínací zařízení slouží ke spínání ventilů. Cloud zajišťuje vzdálené nastavení systému. Komunikace na úrovni lokální sítě byla realizována prostřednictvím wifi s využitím smíšené (mesh) topologie, která zajišťuje vysokou dostupnost. Pro komunikaci s internetem je realizováno cloudové řešení, které umožňuje vzdálenou správu.

Všechna data systému jsou uložena ve všech prvcích systému, kdy výpadek jednoho prvku neovlivní zbývající část systému. Zároveň je možná konfigurace všech zařízení z jakéhokoliv prvku systému.

Demonstrace funkčnosti proběhla na zmenšeném modelu z důvodu rychlejšího ověření systému. Byl vytvořen zmenšený model místnosti, na kterém bylo snadné otestovat regulaci topení. Testováním bylo prokázáno, že systém je plně funkční a splnil všechny požadavky.

11 Shrnutí a doporučení

Při návrhu a vyhotovení bylo nalezeno několik možných podnětů pro zlepšení systému. Tím se otevírá možnost pro budoucí práce, které by mohly na tuto práci navázat a vylepšit řešení.

U obou druhů zařízení (měřicí i spínací) je nutné znát jeho IP adresu, aby bylo možné provádět lokální konfiguraci celého systému.

Nastavení přístupového bodu je pevně nakonfigurováno ve zdrojovém kódu firmware. Vhodnou formou řešení by byla konfigurační stránka pro zadání údajů parametrů přístupovému bodu. Přestože tato funkcionalita nebyla předmětem práce, pomohla by ke zvýšení uživatelské přívětivosti.

U měřicího zařízení by bylo vhodnější navrhnout napájení z baterie nebo akumulátoru, neboť měření teploty je energeticky nenáročné. Překážkou je však velká spotřeba wifi komunikačního modulu, která zejména slouží k nastavování plánu topení.

V cloudové části je webová aplikace, která umožňuje konfigurovat pouze jeden systém. Bylo by vhodné aplikaci upravit tak, aby sloužila více samostatným systémům. Takové řešení bude nutné doplnit o správu uživatelů a řízení přístupu. To by vedlo k možnosti komerčního využití.

12 Seznam použité literatury

[1] UŠETŘENO.CZ S.R.O. *Vytápění bytových domů: Co je nejvýhodnější? | Ušetřeno.cz* [online]. 12. červen 2024 [cit. 2024-08-01]. Dostupné z: https://www.usetreno.cz/clanky/vytapeni-bytovych-domu/

[2] ATLASSIAN. What is „a distributed system? *Atlassian* [online]. 2024 [cit. 2024-08-01]. Dostupné z: https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture

[3] NAKAMOTO, Satoshi. Bitcoin: „A Peer-to-Peer Electronic Cash System [online]. nedatováno [cit. 2024-08-01]. Dostupné z: https://bitcoin.org/bitcoin.pdf

[4] ČSÚ. Počet místností bytu. *Sčítání 2021* [online]. 2021 [cit. 2024-06-02]. Dostupné z: https://scitani.gov.cz/pocet-mistnosti-bytu

[5] ESPRESSIF SYSTEMS. *ESP8266EX Datasheet* [online]. 2023 [cit. 2024-06-11]. Dostupné z: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf

[6] SENSIRION. *Datasheet SHT3x-DIS* [online]. prosinec 2022 [cit. 2024-06-20]. Dostupné z: https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet_SHT3x_DIS.pdf

[7] HANDSON TECHNOLOGY. *„4 Channel 5V Optical Isolated Relay Module* [online]. [cit. 2024-06-23]. Dostupné z: https://www.handsontec.com/dataspecs/4Ch-relay.pdf

[8] RTC DS1307 Real Time Clock Module - Tiny RTC - I2C. *hit.ps* [online]. [cit. 2024-06-20]. Dostupné z: https://hit.ps/product/rtc-ds1307-real-time-clock-module-tiny-rtc-i2c/

[9] *Debugging — PlatformIO latest documentation* [online]. [cit. 2024-07-06]. Dostupné z: https://docs.platformio.org/en/latest/plus/debugging.html#boards

[10] ELECTRONIC, S. O. S. *IQMESH – bezdrátová síť „s topologií MESH* [online]. [cit. 2024-06-28]. Dostupné z: https://www.soselectronic.com/cz/articles/iqrf/iqmesh-bezdratova-sit-s-topologii-mesh-2075

[11] SILBERSCHATZ, Abraham, Peter Baer GALVIN „a Greg GAGNE. *Operating System Concepts* [online]. 2018 [cit. 2024-07-31]. ISBN 978-1-119-32091-3. Dostupné z: https://os.ecci.ucr.ac.cr/slides/Abraham-Silberschatz-Operating-System-Concepts-10th-2018.pdf

Seznam obrázků

Obrázek 1 – Grafické znázornění centralizovaného a decentralizovaného systému (zdroj: [2]) 7
Obrázek 2 – Kompletní modul LOLIN D1 mini (zdroj: autor) 18
Obrázek 3 – Modulární základna k propojení (zdroj: autor) 20
Obrázek 4 – Deska s čidlem SHT 30 od společnosti Wemos (zdroj: autor) 21
Obrázek 5 – Deska se čtyřmi relátky (zdroj: autor) 22
Obrázek 6 – Deska s hodinami reálného času (zdroj: [8]) 23
Obrázek 7 – Možné umístění zařízení v domě (zdroj: autor) 24
Obrázek 8 – Ukázka sítě mesh (zdroj: [10]) 30
Obrázek 9 – Blokový diagram propojení systému (zdroj: autor) 32
Obrázek 10 – Nejmenší možná funkční reprezentace systému (zdroj: autor) 33
Obrázek 11 – Prototyp měřicího zařízení (zdroj: autor) 33
Obrázek 12 – Prototyp spínacího zařízení (zdroj: autor) 34
Obrázek 13 – Diagram tříd s jednotlivými metodami (zdroj: autor) 35
Obrázek 14 – Zjednodušený sekvenční diagram IoT zařízení v systému (zdroj: autor) 37
Obrázek 15 – Nastavení spínacího zařízení v systému na mobilním zařízení (zdroj: autor) 43
Obrázek 16 – Nastavení měřicího zařízení v systému na mobilním zařízení (zdroj: autor) 43
Obrázek 17 – Nastavení měřicího zařízení v systému na stolním zařízení (zdroj: autor) 44
Obrázek 18 – Nastavení zařízení v systému, jiný den (zdroj: autor) 45
Obrázek 19 – Struktura dat v JSON pro jedno měřicí zařízení (zdroj: autor) 46
Obrázek 20 – Struktura dat v JSON pro jedno všechna relátka v systému (zdroj: autor) 47
Obrázek 21 – Testovací model místnosti (zdroj: autor) 49
Obrázek 22 – Záznam z připojení nového zařízení do systému (zdroj: autor) 50
Obrázek 23 – Záznam o změně konfigurace zařízení z lokální sítě (zdroj: autor) ..52

Obrázek 24 – Původní konfigurace „LozniceOvladac“ v testu (zdroj: autor) 53
Obrázek 25 - Nová konfigurace „OvladacLoznice“ v testu (zdroj: autor) 54
Obrázek 26 – Záznam ze synchronizace zařízení a cloudu (zdroj: autor) 55
Obrázek 27 – Záznam měřicího zařízení ovládající topení (zdroj: autor) 56
Obrázek 28 – Záznam ze spínacího zařízení ovládající relátka (zdroj: autor) 57
Obrázek 29 – Záznam synchronizace času z RTC na spínacím zařízení (zdroj: autor) 58
Obrázek 30 – Záznam synchronizace a topení měřicího zařízení (zdroj: autor) 59
Obrázek 31 – Záznam změny stavu relátek při odpojení měřicího zařízení (zdroj: autor) 60

Seznam tabulek

Tabulka 1 – Požadavky na systém (zdroj: autor) 11
Tabulka 2 – Požadavky na měřicí zařízení (zdroj: autor) 13
Tabulka 3 – Požadavky na spínací zařízení (zdroj: autor) 15
Tabulka 4 – Požadavky na cloud (zdroj: autor) 16
Tabulka 5 – Podporované wifi parametry ESP8266 (zdroj: [5]) 19
Tabulka 6 – Vybraný přehled typu zpráv a jejich velikost (zdroj: autor) 31
Tabulka 7 – Definice časového plánu topení (zdroj: autor) 36
Tabulka 8 – Použité knihovny u firmware (zdroj: autor) 36
Tabulka 9 – Splnění požadavků měřicího zařízení (zdroj: autor) 61
Tabulka 10 – Splnění požadavků spínacího zařízení (zdroj: autor) 62
Tabulka 11 – Splnění požadavků cloudu (zdroj: autor) 63

13 Přílohy

1) Zdrojový kód firmware (https://gitlab.com/Najsr/dp-cesak-firmware)
2) Zdrojový kód cloudové aplikace (https://gitlab.com/Najsr/dp-cesak-cloud)

Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor:	**Bc. David Česák**
Studium:	I2200599
Studijní program:	N1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název diplomové práce:	**Decentralizovaný IoT systém topení**
Název diplomové práce AJ:	Decentralized IoT heating system

Cíl, metody, literatura, předpoklady:

Cíl práce

Cílem práce je návrh a realizace decentralizovaného systému topení alespoň pro tři místnosti. Decentralizace zajistí, že při výpadku jednoho prvku bude zbytek systému plně funkční i při nedostupnosti internetu. Systém bude složen ze dvou typů zařízení: měřicí a spínací. Zařízení si budou mezi sebou synchronizovat konfiguraci v rámci lokální Wi-Fi sítě, aby z libovolného místa bylo možné systém plně nakonfigurovat. Návrh a výroba zařízení je součástí práce.

Osnova

1. Úvod
2. Cíl práce
3. Metodika zpracování
4. Identifikace požadavků pro decentralizovaný systém topení
5. Charakterizace dílčích prvků systému podle požadavků
6. Návrh technického řešení
7. Postup realizace navrženého technického řešení
8. Demonstrace funkčnosti systému
9. Závěr
10. Shrnutí a doporučení

Očekávaný přínos

Vlastní funkční vytápěcí systém odolný vůči kybernetickému útoku.

Zadávající pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	Mgr. Daniela Ponce, Ph.D.
Datum zadání závěrečné práce:	15.10.2021